ALTAIR

ONLY FORWARD

Altair Feko 2024.1

User Guide

Updated: 07/31/2024

Intellectual Property Rights Notice

This Intellectual Property Rights Notice is exemplary, and therefore not exhaustive, of the intellectual property rights held by Altair Engineering Inc. or its affiliates. Software, other products, and materials of Altair Engineering Inc. or its affiliates are protected under laws of the United States and laws of other jurisdictions.

In addition to intellectual property rights indicated herein, such software, other products, and materials of Altair Engineering Inc. or its affiliates may be further protected by patents, additional copyrights, additional trademarks, trade secrets, and additional other intellectual property rights. For avoidance of doubt, copyright notice does not imply publication. Copyrights in the below are held by Altair Engineering Inc. or its affiliates. Additionally, all non-Altair marks are the property of their respective owners. If you have any questions regarding trademarks or registrations, please contact marketing and legal.

This Intellectual Property Rights Notice does not give you any right to any product, such as software, or underlying intellectual property rights of Altair Engineering Inc. or its affiliates. Usage, for example, of software of Altair Engineering Inc. or its affiliates is governed by and dependent on a valid license agreement.

Altair® HyperWorks®, a Design & Simulation Platform

Altair® AcuSolve® ©1997-2024

Altair® Activate® ©1989-2024

Altair® Automated Reporting Director™ ©2008-2022

Altair® Battery Damage Identifier™ ©2019-2024

Altair® CFD™ ©1990-2024

Altair Compose® ©2007-2024

Altair® ConnectMe™ ©2014-2024

Altair® DesignAI™ ©2022-2024

Altair® EDEM™ ©2005-2024

Altair® EEvision™ ©2018-2024

Altair® ElectroFlo™ ©1992-2024

Altair Embed® ©1989-2024

Altair Embed® SE ©1989-2024

Altair Embed®/Digital Power Designer ©2012-2024

Altair Embed®/eDrives ©2012-2024

Altair Embed® Viewer ©1996-2024

Altair® e-Motor Director™ ©2019-2024

Altair® ESAComp® ©1992-2024

Altair® Silicon Debug Tools™ ©2018-2024

Altair® SimLab® ©2004-2024

Altair® SimLab® ST ©2019-2024

Altair® SimSolid® ©2015-2024

Altair® SpiceVision PRO™ ©2002-2024

Altair® Squeak and Rattle Director™ ©2012-2024

Altair® StarVision PRO™ ©2002-2024

Altair® Structural Office™ ©2022-2024

Altair® Sulis™ ©2018-2024

Altair® Twin Activate® ©1989-2024

Altair® UDE™ ©2015-2024

Altair® ultraFluidX® ©2010-2024

Altair® Virtual Gauge Director™ ©2012-2024

Altair® Virtual Wind Tunnel™ ©2012-2024

Altair® Weight Analytics™ ©2013-2022

Altair® Weld Certification Director™ ©2014-2024

Altair® WinProp™ ©2000-2024

Altair® WRAP™ ©1998-2024

Altair® HPCWorks®, a HPC & Cloud Platform

Altair® Allocator™ ©1995-2024

Altair® Access™ ©2008-2024

Altair® Accelerator™ ©1995-2024

Altair® Accelerator™ Plus ©1995-2024

Altair® Breeze™ ©2022-2024

Altair® Cassini™ ©2015-2024

Altair® Control™ ©2008-2024

Altair® Desktop Software Usage Analytics™ (DSUA) ©2022-2024

Altair® FlowTracer™ ©1995-2024

Altair® Grid Engine® ©2001, 2011-2024

Altair® InsightPro™ ©2023-2024

Altair® Hero™ ©1995-2024

Altair® Liquid Scheduling™ ©2023-2024

Altair® Mistral™ ©2022-2024

May 23, 2024

Technical Support

Altair's support resources include engaging learning materials, vibrant community forums, intuitive online help resources, how-to guides, and a user-friendly support portal.

Visit Customer Support to learn more about our support offerings.

Contents

Introduction to Feko

1

Feko is a comprehensive electromagnetic solver with multiple solution methods that is used for electromagnetic field analyses involving 3D objects of arbitrary shapes.

This chapter covers the following:

1.1 Feko Overview

Feko is a comprehensive computational electromagnetics (CEM) software product used widely in the telecommunications, automotive, aerospace and defense industries.

The name Feko is an abbreviation derived from the German phrase "FEldberechnung bei Körpern mit beliebiger Oberfläche" (field computations involving bodies of arbitrary shape). As the name suggests, Feko can be used for various types of electromagnetic field analyses involving objects of arbitrary shapes. Feko offers several frequency domain electromagnetic (EM) solution methods as well as a time domain method under a single license. Hybridisation of these methods enables efficient analysis of a broad spectrum of EM problems, including antennas, microstrip circuits, radio frequency (RF) components and biomedical systems, the placement of antennas on electrically large structures, the calculation of scattering (RCS), as well as the investigation of electromagnetic compatibility (EMC).

Feko offers tools that are tailored to solve the more challenging EM interactions, including dedicated solvers for characteristic mode analysis (CMA) and bi-directional cables coupling. Special formulations are included for efficient simulation of integrated windscreen antennas and antenna arrays.

Combined with the multilevel fast multipole method (MLFMM), and true hybridisation of the solvers, Feko is considered the global market leader for antenna placement analysis.

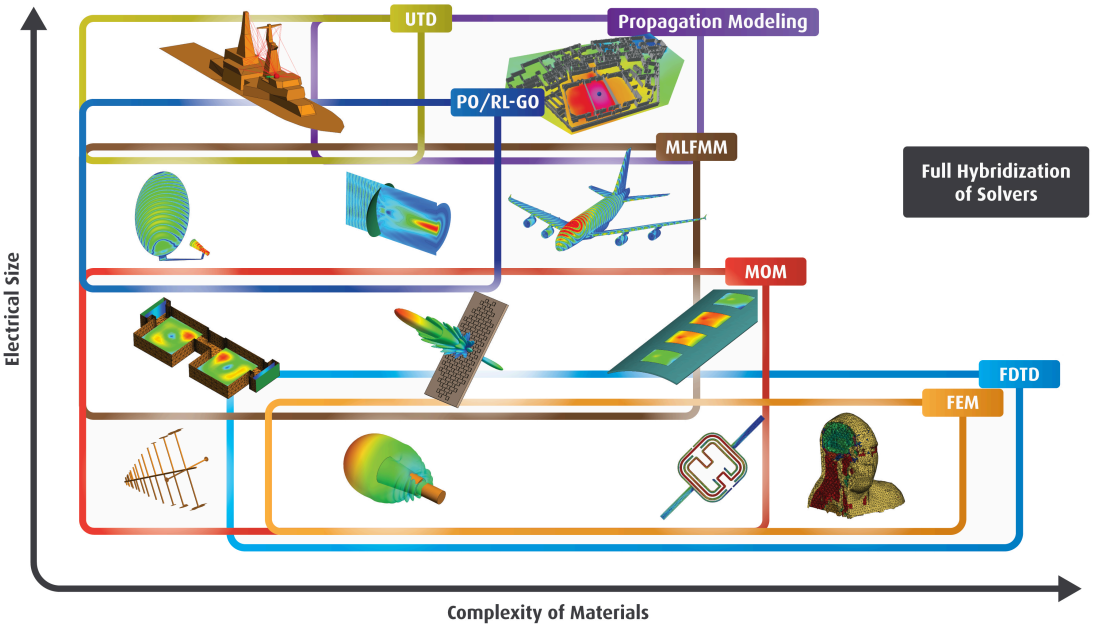


Figure 1: Illustration of the numerical analysis techniques in Feko.

Solver Overview

The Solver supports the following solution methods:

- Full wave frequency domain solution methods:
  - MoM (method of moments)
  - FEM (finite element method)
  - MLFMM (multilevel fast multipole method)

- Full wave time domain solution methods:
  - FDTD (finite difference time domain)
- Asymptotic solution methods:
  - PO (physical optics)
  - LE-PO (large element physical optics)
  - RL-GO (ray launching geometrical optics)
  - UTD (uniform theory of diffraction)

CPU Parallelisation for Shared and Distributed Memory Systems

In Feko, true distributed computing and "farming" parallelisation of simulation are two distinct concepts.

- With true distributed computing, any particular solution (for example, a single frequency) can be parallelised across multiple nodes. This is achieved via rigorous MPI-based parallelisation for clusters and shared memory computers. The efficiency of the parallelisation is improved by limiting the MPI interaction between processes.
- Farming assigns individual optimisation iterations to separate CPU cores, while not distributing the solution of the iterations.

Efficiency in Feko is further boosted by integration for different high speed networking technologies such as Gigabit Ethernet and Infiniband.

- For multiple cores of a single CPU, OpenMP technology is used for parallelisation.
- For clusters or shared memory multi-CPU servers, select memory blocks are copied between processes. Communication is reduced and simulation time decreased, but a penalty is paid in terms of memory efficiency.
- In contrast, the cores of multi-core CPUs address the same memory block much faster than in shared memory multi-CPU systems. OpenMP parallelisation of Feko for multi-core CPUs make use of this fact to reduce memory requirements.

MPI and OpenMP distributed parallelisation methods are hybridised in Feko to harness the strengths of both schemes.

GPU Acceleration

Feko supports the use of multiple GPUs. It uses the GPU cores to accelerate simulation using the unified device architecture (CUDA) framework from NVIDIA. The computational phases targeted for execution on CUDA-based GPUs show a significant speedup when compared to standard CPU-based execution.

Optimisation

Feko offers state-of-the-art optimisation engines based on genetic algorithm (GA) and other methods, which can be used to automatically optimise the design and determine the optimum solution. Furthermore, for advanced design exploration, the interface to Altair HyperStudy offers a comprehensive post-processing functionality (including trade off analysis and stochastics).

User Interface

The Feko components with a graphical user interface (CADFEKO, EDITFEKO and POSTFEKO) make use of a ribbon driven interface that focusses on improved efficiency of workflow. CADFEKO supports parametric model construction. Complex geometry models and mesh models can be imported or

exported in a wide range of industry standard formats. Use the application programming interface (API) to control CADFEKO or POSTFEKO from an external script or to automate repetitive and mundane tasks.

Updater

Feko has an updater utility that allows you the flexibility to install an update containing new features, minor software enhancements and bug fixes on top of an existing base installation.

Altair Units

Feko is part of the Altair Units based licensing system which allows metered usage of the entire Altair suite of products. This value-based licensing model has been extended to Altair's extensive partner network, providing the most comprehensive and dynamic set of solutions to the market.

Related concepts

Which Solution Methods Support GPU Acceleration?

1.2 Feko Applications

Feko is applicable to a wide range of applications in electromagnetic engineering.

The wide range of applications can be attributed to the support for various solution methods and the hybridisation of the methods. No single solution methods is applicable to the full frequency range and model complexity. Feko is uniquely positioned to efficiently solve models with a wide range of complexities and size.

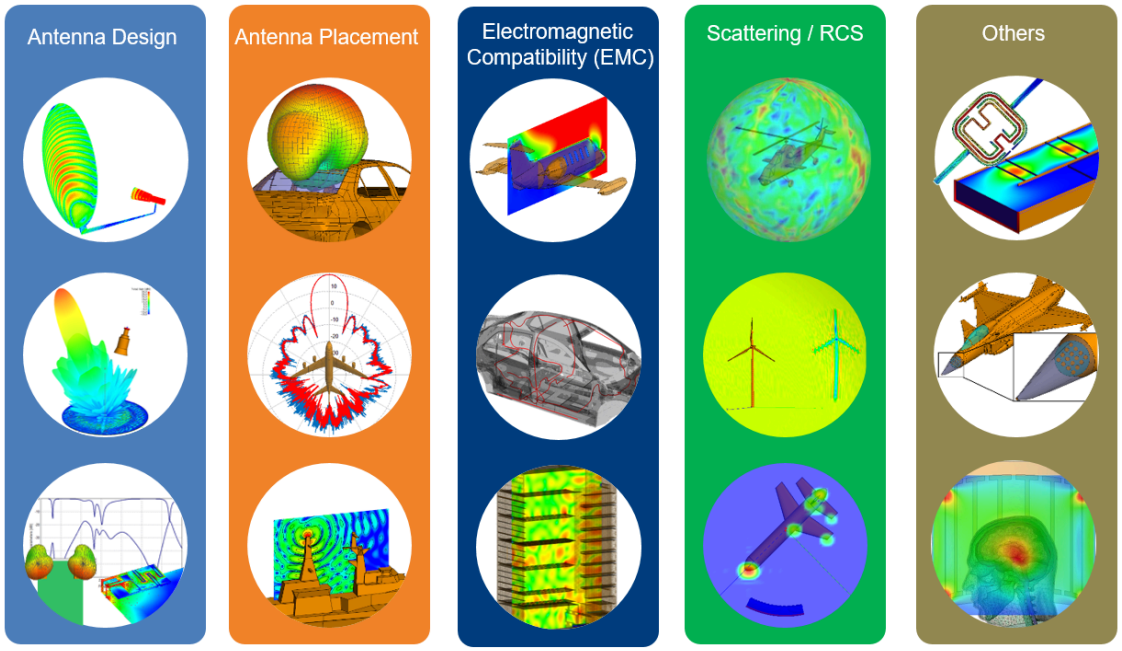


Figure 2: Illustration of the applications of Feko.

Feko is used in, but not limited to, the following applications:

Antenna design
: Analysis of wide-ranging antenna. Examples include wireless communication devices and systems (FM, GPS, 3G, TV, LTE and MIMO), reflector antenna, antennas for radars, antennas with radomes and many more.

Antenna placement
: Analysis of antenna and the interaction with electrically large environments. Examples include antennas on vehicles, aircraft, satellites, ships, cellular base-stations, including radiation patterns, co-site interference and RADHAZ analysis.

Electromagnetic compatibility (EMC) analysis
: Analysis which involve cables, which either radiate through imperfect shields and cause coupling into other cables, devices or antennas, or which receive (irradiation) external electromagnetic fields (radiated from antennas or leaked through other devices) and then cause disturbance voltages and currents potentially resulting in a malfunctioning of the system.

Radar cross section / scattering
: Analysis and scattering of large metallic / dielectric and composite structures, for example, aircraft, vehicles, tanks, ships, buildings and wind turbines.

Radomes
: Analysis of complex shapes, multi-layer and electrically large structures.

Waveguides
: Analysis of complex waveguide components, for example, waveguide filters and couplers.

Bio-electromagnetics
: Analysis of human-structure interaction, for example, hearing aids, active and passive implants using pacemakers, neural implants, stents and microwave imaging technologies.

Microstrip circuits
: Analysis using optimised formulations for layered media, for example, microstrip antenna array and split ring resonators.

Special materials
: Analysis of frequency selective surfaces (FSS), anisotropic materials (for example, carbon fiber) and metamaterials.

1.2.1 Antenna Design

Feko's broad range of solution methods technologies makes it applicable to solve a range of different antenna types.

For example, the method of moments (MoM) solver is well suited to solve metallic antennas while the finite difference time domain (FDTD) method is a better choice for broadband or multi-band antennas.

Typical antennas types including wire antennas, microstrip antennas, horn and aperture antennas with lenses, broadband and multi-band antennas, multiple-input and multiple-output (MIMO) designs for wireless communications, reflectors, phased arrays and conformal antennas.

Key features:

- Variety of accurate, powerful and reliable 3D electromagnetic (EM) solution methods, including dedicated tools for designing windscreen antennas.
- Unique characteristic mode analysis (CMA) solver for intelligent design.
- Dedicated tools for antenna arrays including periodic boundary condition (PBC) for repeating structures and domain Green's function method (DGFM) for large, but finite arrays.
- Parametric modelling and a powerful optimisation engine.

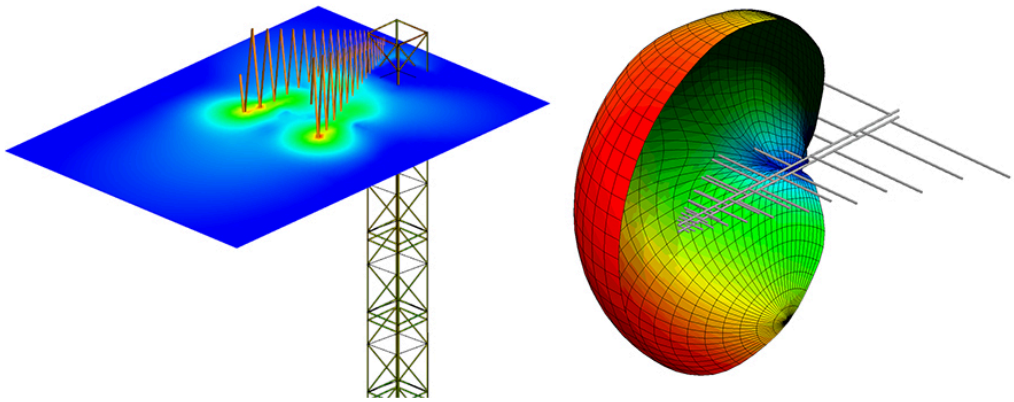


Figure 3: A base station antenna including supporting mast (on the left) and a log periodic antenna (to the right).

1.2.2 Antenna Placement

It is often necessary to understand (and optimise) how an antenna's performance is influenced by the structure (for example, an aeroplane, vehicle or ship) it is mounted on.

The electrically large nature of the structures that need to be considered make them challenging to simulate. Feko's solver offering make it the leading tool for antenna placement and co-site interference analysis.

Key features:

- multilevel fast multipole method (MLFMM) for the efficient simulation of electrically large platforms.
- True hybridised, asymptotic solvers for simulation of electrically very large platforms.
- Model decomposition allows equivalent representation of transmit / receive antenna to reduce computational requirements.
- Unique characteristic mode analysis (CMA) solver for understanding placement aspects.

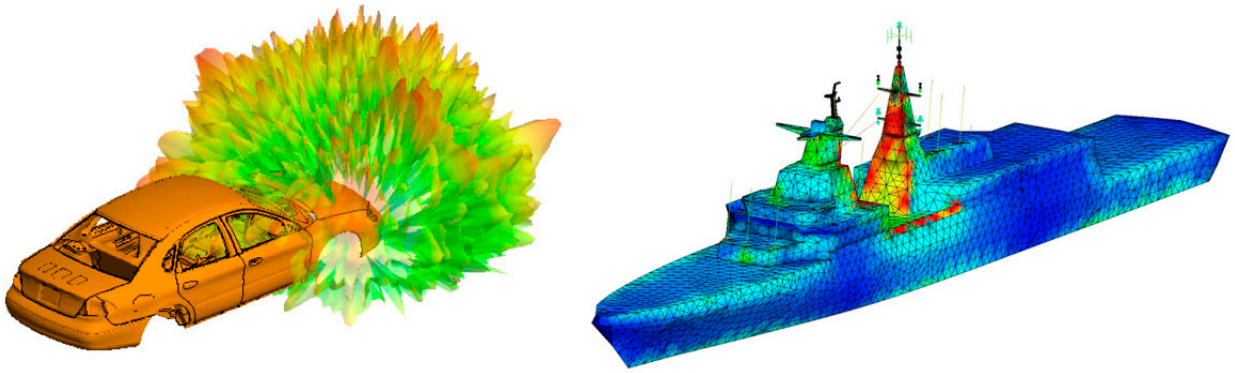


Figure 4: Wireless antenna placement in a vehicle (on the left) and co-site interference analysis for a naval platform (to the right).

1.2.3 Radar Cross Section / Scattering

The scattering performance of an object describes how energy is scattered when the object is exposed to incident electromagnetic fields.

Applications include mono- and bi-static radar cross sections of defence platforms, scattering from wind turbines and optimisation of other radar systems like automotive collision detection systems. Feko's high frequency methods are used to solve these typically electrically large, platforms.

Key features:

- The multilevel fast multipole method (MLFMM) is used to efficiently simulate electrically large platforms.
- True hybridised, asymptotic solvers for simulation of electrically very large platforms.
- Model decomposition allows equivalent representation of transmitter / receiver antennas to reduce computational requirements.

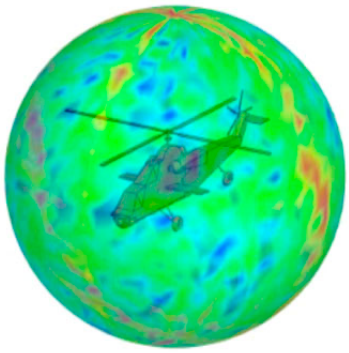


Figure 5: RCS calculation for a helicopter.

1.2.4 Electromagnetic Compatibility (EMC) Analysis

Electromagnetic compatibility (EMC) analysis is not only used to predict emission and immunity performance, but also in the product design phase to mitigate problems due to external or co-site interfaces.

Feko is used extensively for immunity and radiated emissions testing, shielding effectiveness, noise coupling, radiation hazard (RADHAZ) analysis, electromagnetic pulses (EMP), lightning analysis, high intensity radiated fields (HIRF), reverberation and anechoic chamber simulations.

Key features:

- Efficient solvers enable simulation of EMC tests including device under test (DUT) and the test environment.
- Specialised cable modelling and solver tool, to analyse inter-cable coupling, and coupling between cables and antennas or other devices.
- Model decomposition uses equivalent representation of electronic control units in emission tests to reduce computational requirements.
- Time analysis tools to investigate the time domain behaviour in lightning, EMP and noise coupling.

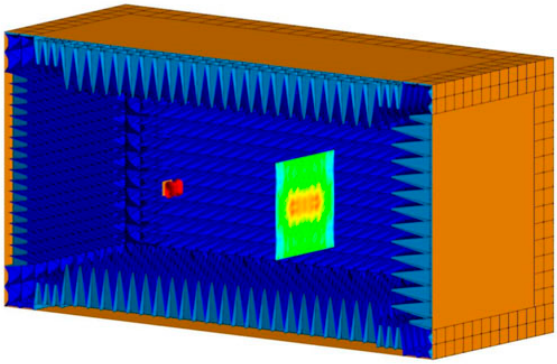


Figure 6: Simulation of an anechoic chamber with antenna under test (AUT).

1.2.5 Waveguides

Feko is well suited to the simulation and optimisation of waveguide components. Due to the typically metallic structures, they can be analysed accurately and efficiently with the method of moments (MoM) and finite element method (FEM) solvers.

Applications include waveguide filters, couplers, circulators, diplexers and multiplexers. Waveguides are also used to feed certain antennas.

Key features:

- Efficient solvers suited to simulating typical waveguide geometries.
- S-parameter analysis, import / export of Touchstone files and general network blocks for circuit representation.
- Parametric model creation using canonical shapes through GUI or scripting, and extensive CAD import support.

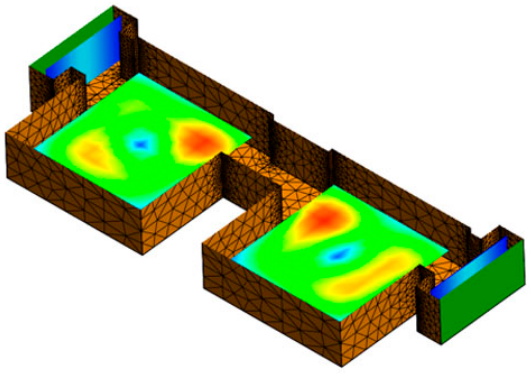


Figure 7: Ku-band waveguide filter.

1.2.6 Radomes

Although radomes are designed to be transparent, they typically have some small effect on the antenna performance which needs to be quantified.

This can be challenging because they are generally electrically large structures, often consisting of multiple thin dielectric layers. Feko offers a range of features and methods that are well suited to simulation of radomes.

Key features:

- The multilevel fast multipole method (MLFMM) for the efficient simulation of electrically large platforms.
- True hybridised, asymptotic solvers for simulation of electrically very large platforms.
- Efficient treatment of thin dielectric layers and coatings.
- Model decomposition for equivalent representation of antennas to reduce computational requirements.

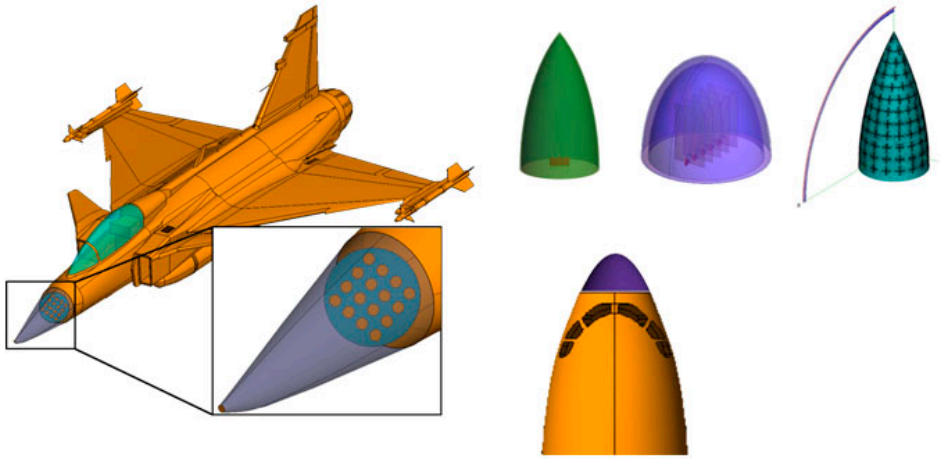


Figure 8: Various radome designs for plane nose cones.

1.2.7 Bio-Electromagnetics

Electromagnetic simulation plays a key role in designing products and investigating safety aspects for healthcare systems, which often include wireless telemetry.

Applications include wireless bio-sensors, implanted devices like pacemakers and neuro-stimulators, and MRI systems. Feko's broad solver offering allows the most efficient method to be used for each task: MoM at early design stages with homogeneous phantoms; finite difference time domain (FDTD) or finite element method (FEM) for final analysis with anatomical phantoms. Use uniform theory of diffraction (UTD) and FEM for investigation of effects on large structures.

Key features:

- Efficient simulation with the most suitable solvers and cross-validation strategies.
- Spatial peak specific absorption rate (SAR) and other relevant post-processing performance parameters.

- Free anatomical models available directly from Feko; other models available from various partners.

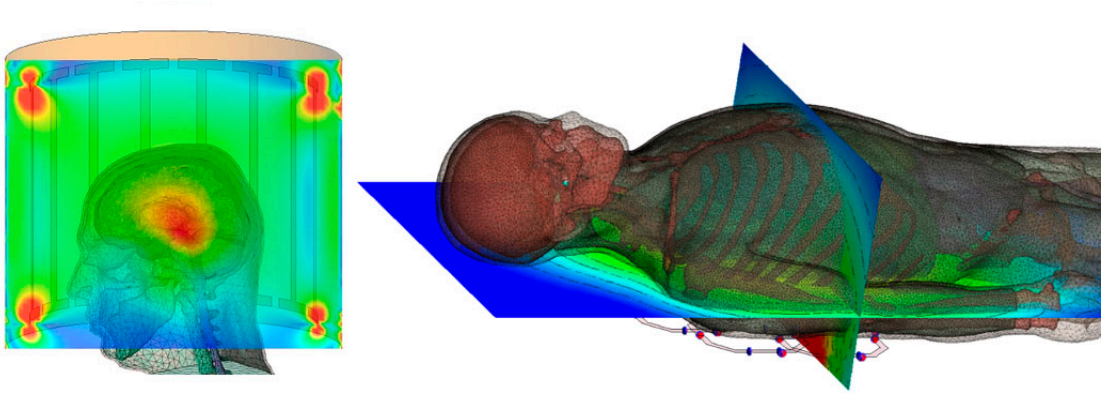


Figure 9: A seven Tesla magnetic resonance (MRI) head coil with anatomical head phantom (on the left) and a three Tesla spinal MRI array (to the right).

1.2.8 Special Materials

Synthetic materials like composites are used increasingly in product design.

In some cases the material is chosen intentionally to influence how the structure interacts with an incident electromagnetic (EM) field. It is therefore important that these materials can be modelled accurately. Feko offers a range of different features that consider these special materials.

Key features:

- Efficient solution methods for periodic structures like frequency selective surfaces (FSS).
- Meta-materials, composites (for example, carbon fiber), metals, dispersive and anisotropic dielectrics.
- Special approximations for coatings and thin dielectric sheets.

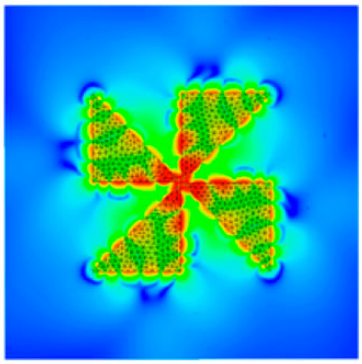


Figure 10: A meta-material resonator.

1.2.9 Microstrip Circuits

Feko offers tools for microstrip circuit design and analysis of filters, resonators, couplers, passive components like spiral inductors, or even complex feed structures for array antennas.

Key features:

- Efficient treatment of multilayer substrates with the planar Green's function.
- Efficient analysis of wideband responses of circuits with the finite difference time domain (FDTD) solver.
- S-parameter analysis, import / export of Touchstone files and general network blocks for circuit representation.
- Parametric model creation using canonical shapes through graphical user interface (GUI) or scripting, and extensive computer-aided design (CAD) import support.

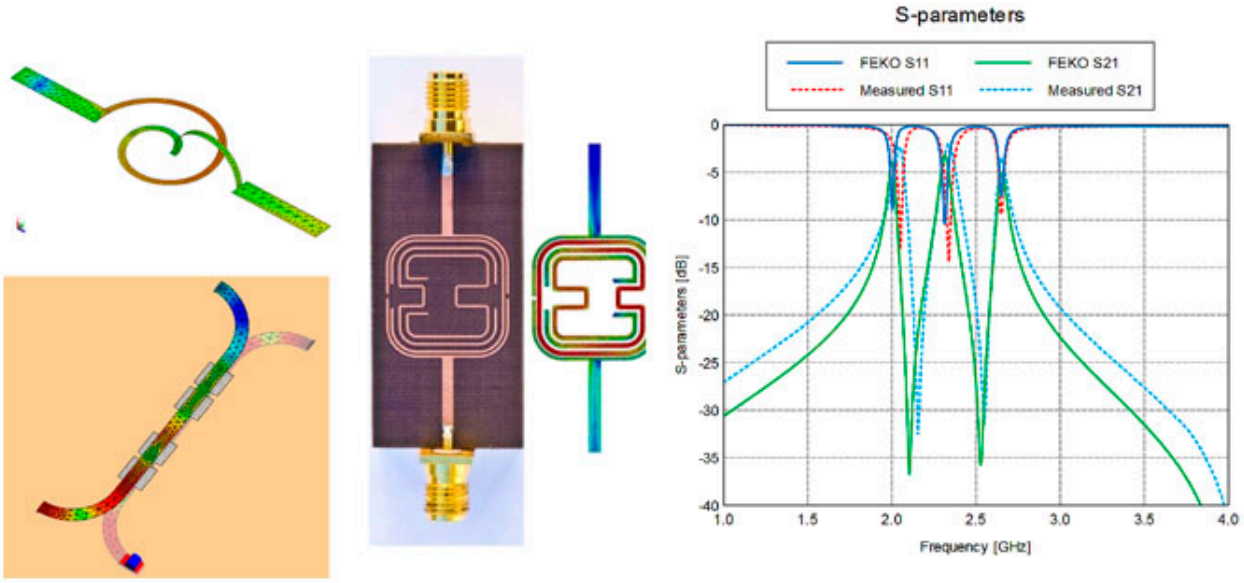


Figure 11: Various microstrip circuit elements and filters.

1.3 How to Get Started

If you are new to Feko, take the following steps to learn about Feko.

1. Watch the **videos** in the Feko installation directory:
   - Tour and demo
   - CADFEKO introduction
   - POSTFEKO introduction
2. The **Quick tips** highlights the essential information regarding the CADFEKO and POSTFEKO environments.
   - Quick tips for CADFEKO
   - Quick tips for POSTFEKO
3. The **Feko Getting Started Guide** contains step-by-step instructions on how to create CADFEKO geometry, request calculations, mesh the geometry, run the Solver and view the results in POSTFEKO.
4. The **Feko Example Guide** contains examples that show the application of features as discussed in the Feko User Guide. The Feko Example Guide assumes you are familiar with interface and focusses on solving more realistic problems. Find an example close to a problem of interest and follow the steps to solve the problem.
5. The **Feko User Guide** contains information regarding Feko and its features.
6. The **Feko Scripting and API Reference Guide** contains information regarding scripting, macro recording, and CADFEKO and POSTFEKO application programming interface (API).
7. The **Feko Errors, Warnings and Notes Reference Guide** is a reference for messages that may be encountered in Feko.
8. The **Feko Practical Guide to Ports and Sources** highlights essential information regarding ports and sources in Feko. It also shows practical examples of feeding typical structures such as waveguides, coaxial cables and microstrip structures.
9. The **Altair web site**[2], provides additional resources as well as online training (self-paced training).
10. The **Altair Community**[3] allows you to post a question or view answers from previous posts. Join the active forum community to get email notifications of new content.

2. https://www.altair.com/feko
3. https://community.altair.com

1.4 About This Manual

The Feko User Guide is part of the Feko documentation and is an extensive reference guide to using Feko.

If you are a beginner user, you are recommended to view the Feko Getting Started Guide.

1.4.1 Purpose of This User Guide

The Feko User Guide provides guidance, best practices and comprehensive technical information regarding the key concepts in Feko.

1.4.2 Document Conventions

The Feko User Guide, makes use of a number of conventions to help you quickly learn about Feko.

- Hyperlinks are indicated in blue.
- Text cited from the GUI interface, are written in bold text, for example, the **Add** button.
- A combination of keystrokes are joined with the "+" sign, for example, Alt+0.
- To draw your attention to important information, the information is marked as a note, tip or warning, for example:

> **Note:** This is a note to draw your attention to critical information.

1.4.3 Feedback

We value your feedback regarding the Feko components and the documentation.

If you have comments or suggestions regarding the Feko component and the documentation, please send an email to support@altair.co.za or contact your local Altair representative.

CADFEKO

2

CADFEKO is used to create and mesh the geometry or model mesh, specify the solution settings and calculation requests in a graphical environment.

This chapter covers the following:

2.1 Introduction to CADFEKO

Use CADFEKO to configure a solver-ready input file for Solver simulations.

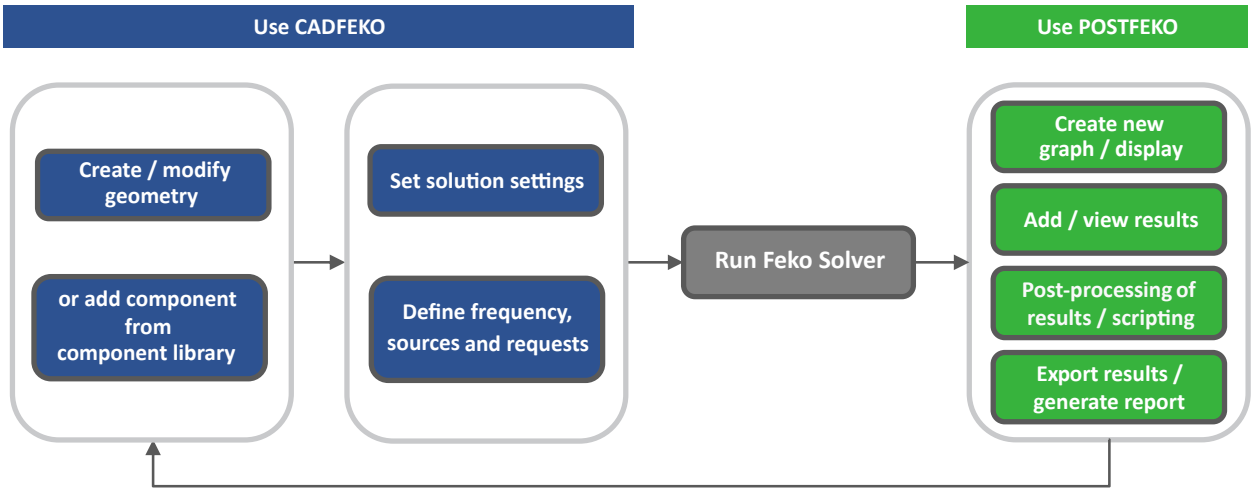
CADFEKO is the Feko component that allows you to create complex CAD geometry using primitive structures (for example, cuboids and polygons) and to perform Boolean operations (for example, union and subtract) on the geometry. Complex geometry models and mesh models can be imported or exported in a wide range of industry standard formats. Reduce development time by using a component from the list of antennas and platforms in the component library.

In CADFEKO, you can request multiple solution configurations, specify calculation requests as well as specify the solution settings for the model. If an optimisation search is required, you can specify the optimisation parameters and goals.

You can generate triangular surface meshes or volume meshes (tetrahedra or voxels) from CAD or mesh parts. The type of mesh generated depends on the solution methods being used.

2.1.1 Feko Components and Workflow

View the typical workflow when working with the Feko components.



CADFEKO

Create or modify the geometry (or model mesh) in CADFEKO, import geometry or mesh, or use a component from the component library. Apply solution settings, define the frequency, specify the required sources and request calculations.

When the frequency is specified or local mesh settings are applied, the automatic mesh algorithm calculates and creates the mesh to obtain a discretised representation of the geometry or model mesh.

View the status of the model in the Notification centre[4]. If any warnings or errors are given, correct the model before running the Solver.

4. Notification centre is the panel to the right of the 3D view under **Model Status**.

Solver

Run the Solver to calculate the specified output requests.

POSTFEKO

Create a new graph or 3D view and add results of the requested calculations on a graph or 3D view. Results from graphs can be exported to data files or images for reporting or external post-processing. Reports can be created that export all the images to a single document or a custom report can be created by configuring a report template.

After viewing the results, it is often required to modify the model again in CADFEKO and then repeat the process until the design is complete.

2.1.2 Launching CADFEKO (Windows)

There are several options available to launch CADFEKO in Microsoft Windows.

Launch CADFEKO using one of the following workflows:

- Open CADFEKO using the Launcher utility.

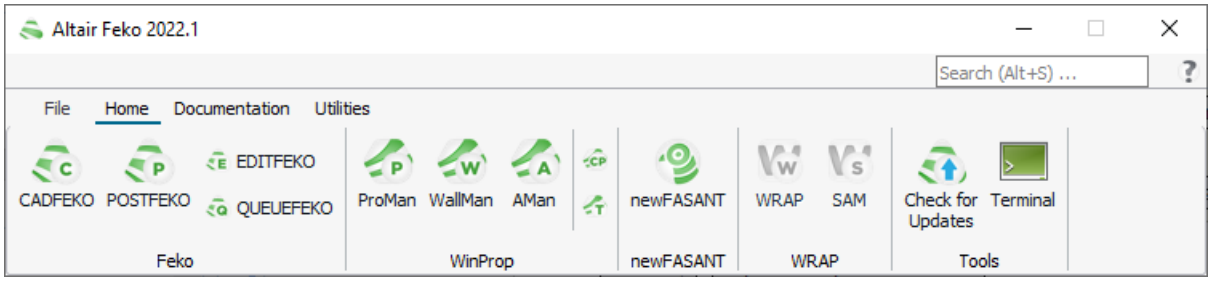


Figure 12: The Launcher utility.

- Open CADFEKO by double-clicking on a `.cfx`[5] file.
- Open CADFEKO from other components, for example, from inside POSTFEKO or EDITFEKO.

> **Note:** If the application icon is used to launch CADFEKO, no model is loaded and the start page is shown. Launching CADFEKO from other Feko components automatically loads the model.

Related tasks
Opening the Launcher Utility (Windows)

2.1.3 Launching CADFEKO (Linux)

There are several options available to launch CADFEKO in Linux.

Launch CADFEKO using one of the following workflows:

- Open CADFEKO using the Launcher utility.
- Open a command terminal. Use the absolute path to the location where the CADFEKO executable resides, for example:

```
/home/user/2024.1/altair/feko/bin/cadfeko
```

- Open a command terminal. Source the "initfeko" script using the absolute path to it, for example:

```
. /home/user/2024.1/altair/feko/bin/initfeko
```

Sourcing `initfeko` ensures that the correct Feko environment is configured. Type `cadfeko` and press Enter.

5. A `.cfx` file is created by CADFEKO and contains the meshed and/or unmeshed CADFEKO model as well as the calculation requests.

> **Note:** Take note that sourcing a script requires a dot (".") followed by a space (" ") and then the path to `initfeko` for the changes to be applied to the current shell and not a sub-shell.

Related tasks
Opening the Launcher Utility (Linux)

2.1.4 Command Line Arguments for Launching CADFEKO

CADFEKO can be called via the command line. Use command line arguments to pass configuration information to CADFEKO.

"If CADFEKO is launched and a file is specified, the model or `.lua` script is opened. Without any models specified, CADFEKO will start and display the start page."

Command-line options:

```
cadfeko [FILENAME] [OPTIONS]
```

FILENAME
Name of the `.cfx` or `.lua` file to load. If the model does not exist, a new empty model is created with this name.

OPTIONS

-h, --help
Displays the help message.

--version
Print the version information and then exit.

--non-interactive
Special execution mode for running automation scripts without user interaction.

--run-script SCRIPTFILE
Specifies an automation script to load and run.

> **Note:** When running a script, the working directory is set to the script to be run before running the script.

--configure-script CONFIGSTRING
Executes the string `CONFIGSTRING` before running the script specified in `SCRIPTFILE`. This option is only used with the "non-interactive" option.

--file-info [=OUTPUTFORMAT] FILENAME.CFX

Display the CADFEKO versions used to create and modify the file.

```
cadfeko startup.cfx --file-info[6]
```

```
cadfeko startup.cfx --non-interactive --file-info |more[7]
```

```
cadfeko startup.cfx --non-interactive --file-info > versions.txt[8]
```

=OUTPUTFORMAT

Optional argument that is used to specify the output format. If the argument is set to `xml`, version information is written out in XML format. XML will only be output to stdout, and only if `--non-interactive` was also specified.

```
cadfeko startup.cfx --file-info=xml --non-interactive | more[9]
```

2.1.5 Start Page

The Feko start page is displayed when starting a new instance (no models are loaded) of CADFEKO, EDITFEKO or POSTFEKO.

The start page provides quick access to **Create a New Project**, **Open an Existing Project**, and a list of **Recent models**.

Links to the documentation (in PDF format), introduction videos and website resources are available on the start page. Click the  icon to launch the Feko help.

6. Opens a dialog and displays the version information.
7. Writes the version information out to standard output stream (stdout).
8. Redirects the version information to the specified file.
9. Writes the version information in XML format in non-interactive mode, displaying the content one screen at a time.

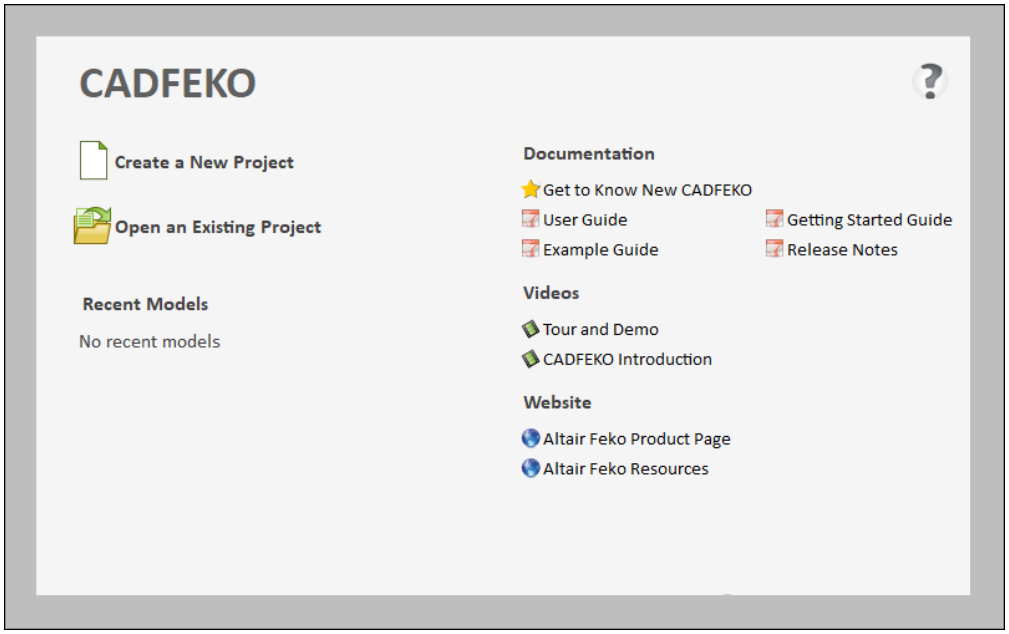


Figure 13: The CADFEKO start page.

2.2 Quick Tour of the CADFEKO Interface

View the main elements and terminology in the CADFEKO application window.

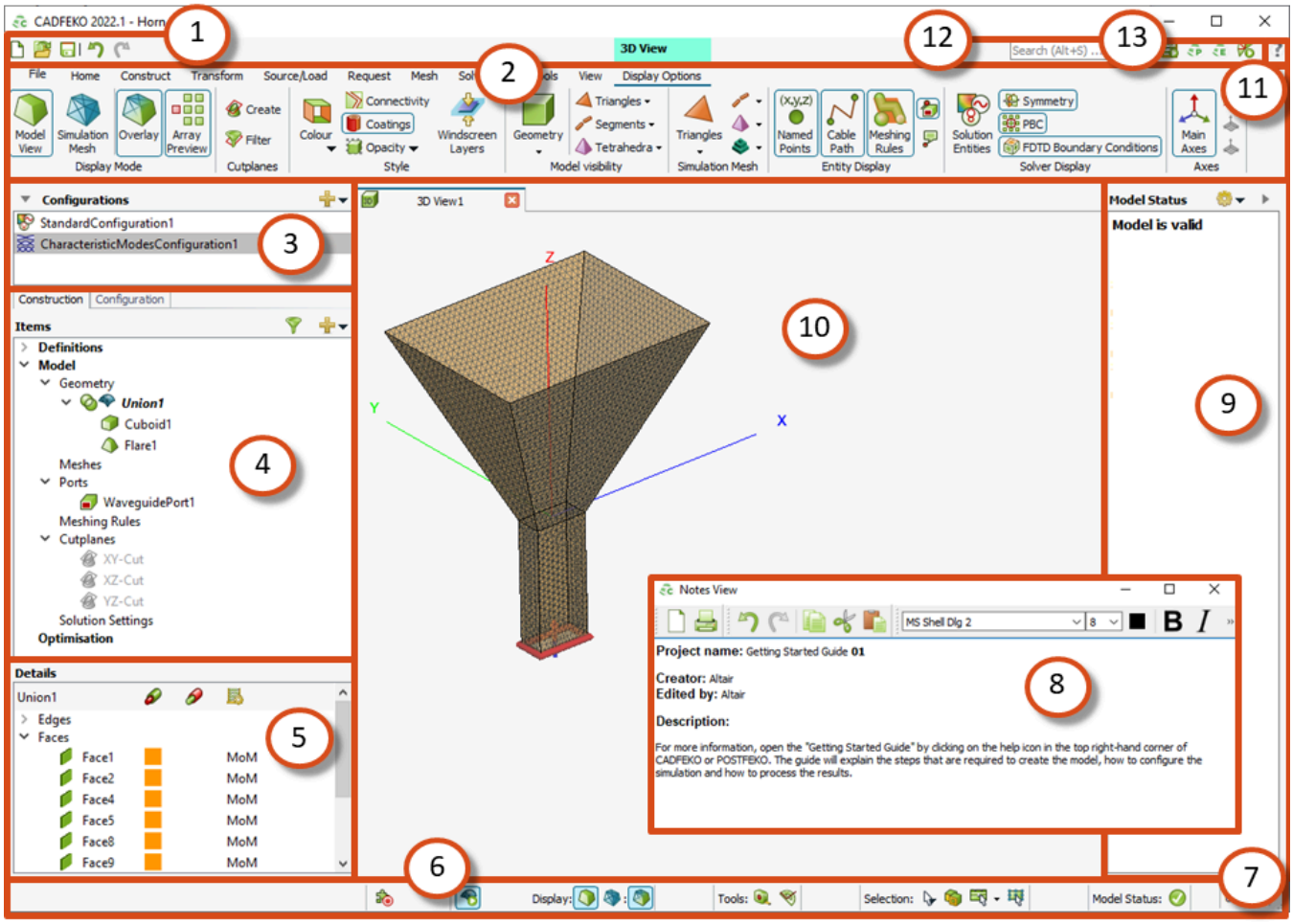


Figure 14: The CADFEKO window.

1. Quick access toolbar
2. Ribbon
3. Configuration list
4. Model tree
5. Details tree
6. Status bar
7. Model Status
8. Notes view
9. Notification Centre
10. 3D view
11. Help
12. Search bar

13. Application launcher

2.2.1 Quick Access Toolbar

The quick access toolbar is a small toolbar that gives quick access to actions that are often performed.

The toolbar is located at the top-left corner of the application window, just below the title bar. It allows you to create a new model, open a model, save a model, undo a model operation or redo a model operation using fewer mouse clicks for a faster workflow. The actions available on the quick access toolbar are also available via the ribbon.

2.2.2 Ribbon

The ribbon is a command bar that groups similar actions in a series of tabs.

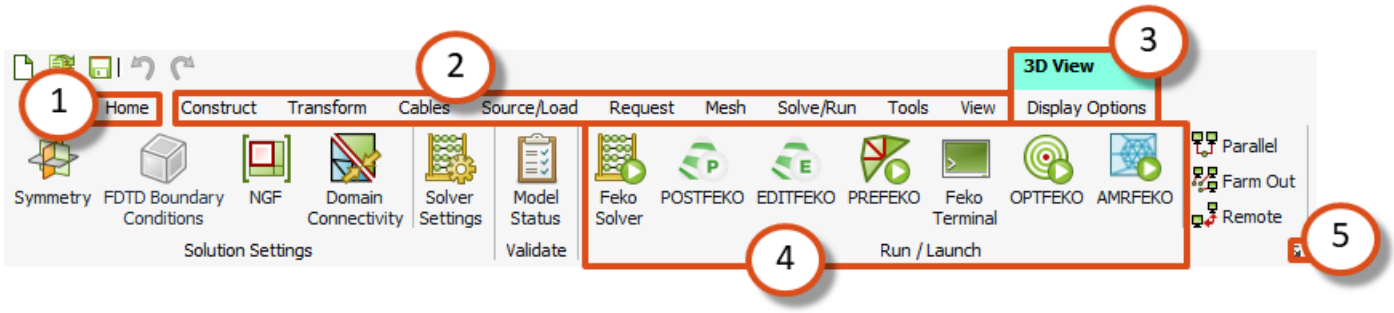


Figure 15: The ribbon in CADFEKO.

1. File menu

The **File** menu is the first item on the ribbon. The menu allows saving and loading of models, import and export options as well as giving access to application-wide settings and a recent file list.

2. Core tabs

A tab that is always displayed on the ribbon, for example, the **Home** tab and **Construct** tab.

The **Home** tab is the first tab on the ribbon and contains the most frequently used commands for quick access.

3. Contextual tab sets

A tab that is only displayed in a specific context.

For example, the **Schematic** contextual tab set contains the **Network Schematic** contextual tab. Contextual tabs appear and disappear as the selected items such as a view or item on a view, change.

4. Ribbon group

A ribbon tab consists of groups that contain similar actions or commands.

5. Dialog launcher

Click the dialog launcher to launch a dialog with additional and advanced settings that relate to that group. Most groups don't have dialog launcher buttons.

Keytips

A keytip is the keyboard shortcut for a button or tab that allows navigating the ribbon using a keyboard (without using a mouse). Press F10 to display the keytips. Type the indicated keytip to open the tab or perform the selected action.

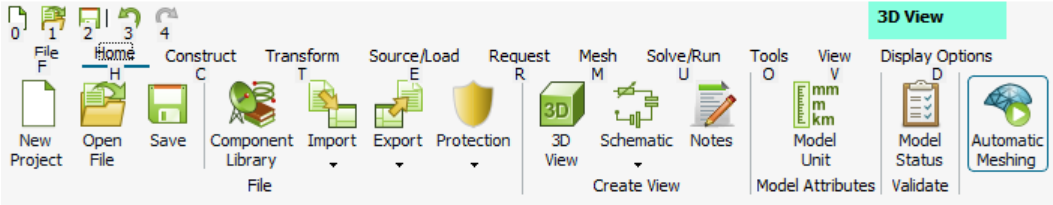


Figure 16: An example of keytips.

Application Menu

The application menu is similar to a standard file menu of an application. It allows saving and loading of models, print functionality and gives access to application-wide settings.

When you click on the application menu drop-down button, the application menu, consisting of two panels, is displayed.

The first panel gives you access to application-wide settings, for example:

- Creating a new model.
- Opening a model, saving a model and closing a model.
- Component library
- Import
- Export
- Model protection
- Print
- Check for updates
- Settings
  - Preferences
  - Keyboard shortcut settings
  - Mouse binding settings
  - Snap settings
  - Model unit
  - Solver settings
  - Component launch options
- Feko help
- About

- Version information about CADFEKO
- Information about Altair HyperWorks Products
- Information about third-party libraries

- Exit

The second panel consists of a recent file list and is replaced by a sub-menu when a menu item is selected.

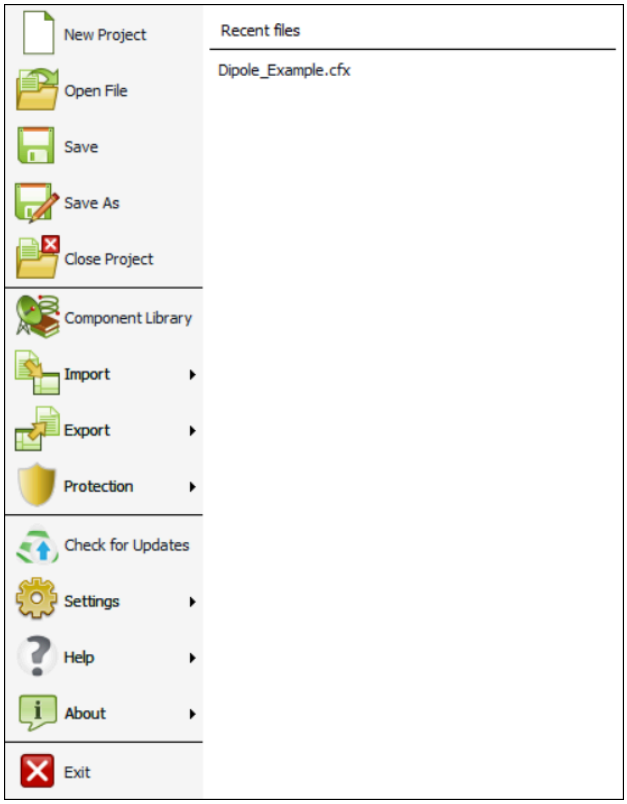


Figure 17: The application menu in CADFEKO.

Home Tab

The **Home** tab is the first tab on the ribbon and contains the most frequently used operations.

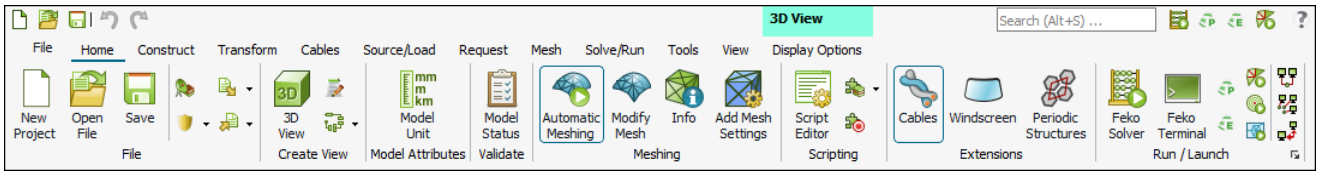


*Figure 18: The **Home** tab in CADFEKO.*

2.2.3 Configuration List

The configuration list displays all configurations in the model.

The panel is located to the left of the application window, just below the ribbon. A new model starts by default with a single standard configuration. The following configuration types are supported:

- Standard configuration
- Multiport S-parameter configuration
- Characteristic modes configuration.

Tip: Multiple configurations allow you to perform efficient simulations using different configurations (different loads or sources) in a single model.

2.2.4 Model Tree

The model tree contains the variables, named points, the model-creation hierarchy, ports and configuration-specific items of the model. The model tree is split between construction and configuration items.

The panel is located below the configuration list and contains a **Construct** tab and **Configuration** tab.

Variables, media and named points are listed in both the **Construct** tab and the **Configuration** tab to provide quick access. A right-click context menu is available for all items in the model tree. Double-click on an item to open its properties.

Construction Tab

The Construction tab contains the geometry and mesh representation of the current model in a tree structure. It also lists ports and the optimisation configuration.

The tree contains a **Definitions** branch, **Model** branch and **Optimisation** branch.

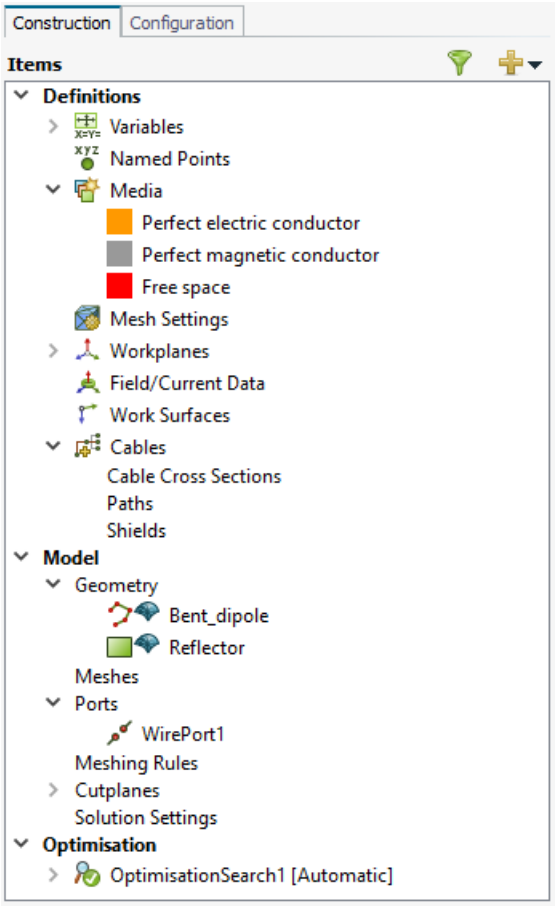


*Figure 19: The **Construction** tab in the model tree.*

Definitions Branch

The **Definitions** branch contains by default the predefined variables, named points, media, mesh settings, workplanes, field/current data, worksurfaces and cables.

Model Branch

The **Model** branch is mainly a visualisation of the geometry and mesh creation hierarchy. Where geometry or mesh objects are derived from existing ones, the original (parent) objects are removed from the top level of the model and listed as sub-levels (children) under the new object.

> **Note:** The highest-level items in the model are referred to as "parts".

For example, **Cone1** and **Cuboid1** (parent objects) were unioned and the result is that they have become children of the new object **Union1**. **Union1** is the highest-level item and referred to as a part.

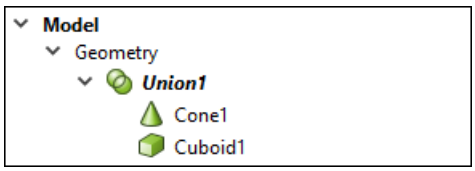


*Figure 20: The **Construction** tab in the model tree showing the part, **Union1**.*

The **Model** branch also contain the ports, meshing rules, cutplanes and solution settings.

Optimisation Branch

The **Optimisation** branch contains the optimisation searches, associated masks, parameters and goal functions defined for the model.

> **Note:** The **Optimisation** branch is only displayed if the model contains an optimisation search or mask.

Configuration Tab

The Configuration tab contains the global and configuration-specific model settings and requests of the current model in tree form.

The tree contains a **Definitions** branch, **Global** branch and **Configuration specific** branch.

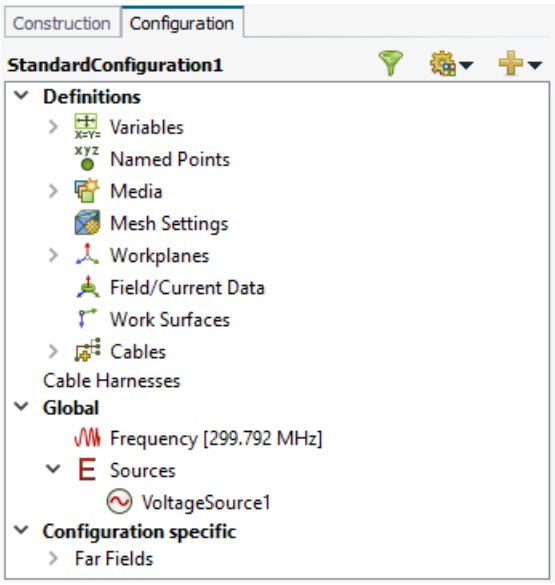


*Figure 21: The **Configuration** tab in the model tree.*

Definitions Branch

The **Definitions** branch contains by default the predefined variables, named points, media, mesh settings, workplanes, field/current data, work surfaces and cables.

Global Branch

The **Global** branch contains the global specific model settings. From the right-click context menu define solver settings, specify the global frequency, sources, loads, networks and power.

Configuration specific Branch

The **Configuration specific** branch contains configuration specific settings. From the right-click context menu define requests per configuration, frequency per configuration, sources per configuration, loads per configuration and power per configuration.

Project Filter Tool

Filter items in the model tree and details tree according to the specified criteria. The filtering can also be applied to 3D views.

In the model tree, at the top right, click the **Project Filter** icon to open the **Project Filter** dialog.

In the **Label filter** group, specify the **Filter text** and the **Filter matching** options:

Partial
: Filter the tree by matching partial text in the model and details tree.

Exact
: Filter the tree by matching the exact text string in the model and details tree.

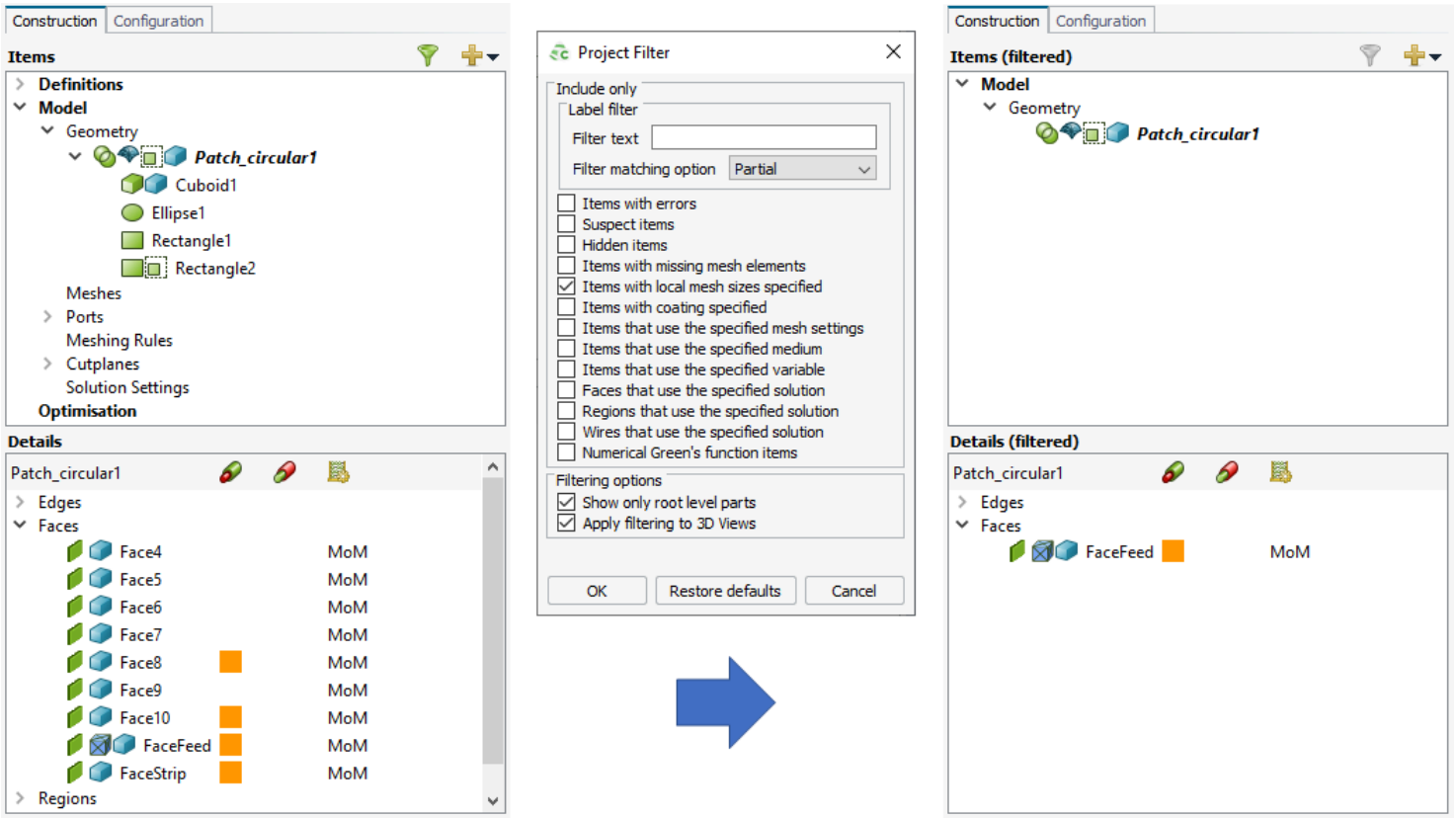


Figure 22: An example showing the unfiltered model tree and details tree (left). An example of the model tree and details tree filtered (right) shows only items with local mesh settings applied.

The following filter criteria are supported:

- Items with errors
- Suspect items
- Hidden items
- Items with missing mesh elements
- Items with local mesh sizes specified
- Items with coating specified

- Items that use the specified mesh settings
- Items that use the specified medium
- Items that use the specified variable
- Faces that use a specified solution
- Regions that use a specified solution
- Wires that use a specified solution
- Numerical Green's function items

Filtering options include the following:

Show only root level parts
When this option is selected, only root level parts are shown in the model tree.

Apply filtering to 3D views
When this option is selected, the filtering is also applied to the 3D view.

When the **Project Filter** tool is active, the text **(filtered)** indicate that the model or details tree are only showing filtered entities.

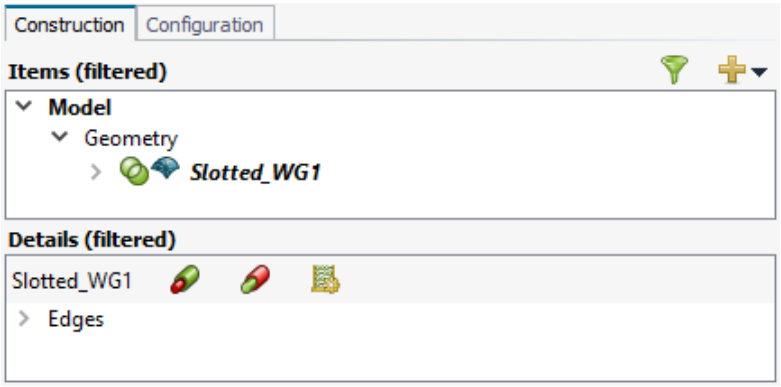


*Figure 23: The text **(filtered)** indicates that **Project Filter** tool is active.*

To disable filtering, re-open the **Project Filter** tool and unselect the filter items.

2.2.5 Details Tree

The details tree panel displays the relevant wires, edges, faces and regions for the geometry or mesh part selected in the **Construct** tab.

The details tree is located below the model tree. From the right-click context menu specify the properties for a wire, edge, face or region properties (which also include solution settings and custom mesh settings) in the details tree.

Edges and Wires (Geometry)

Edges are the boundaries of faces. Wires are not associated with faces and are often referred to as “free edges”.

Selecting an edge or a wire in the 3D view selects the corresponding edge or wire in the details tree. Conversely, selecting an edge or wire in the details tree selects the corresponding edge or wire in the 3D view.

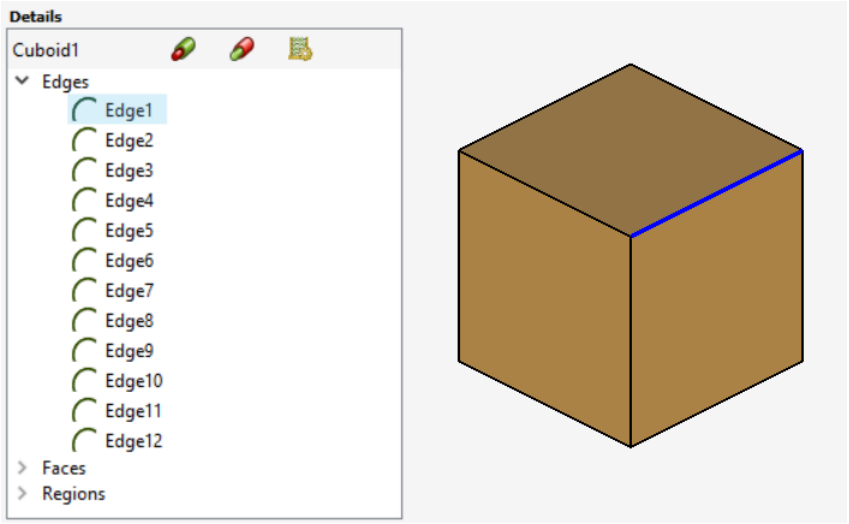


Figure 24: Select an edge in the 3D view to highlight the corresponding entry in the details tree. The converse is also true.

The following can be applied to an edge:

- Local mesh size

The following can be applied to a wire:

- Local wire radius
- Wire core medium (metallic, layered dielectric, impedance sheet)
- Coating (layered dielectric)
- Local mesh size
- Solution method

Faces (Geometry)

Faces are individual surfaces of a part. By default, a face is set to perfect electric conductor (PEC).

> **Note:** The term “face” is used to differentiate from “surface”. A surface refers to a 2D primitive (for example, a polygon).

Selecting a face in the 3D view selects the corresponding face in the details tree. Conversely, selecting a face in the details tree selects the corresponding face in the 3D view.

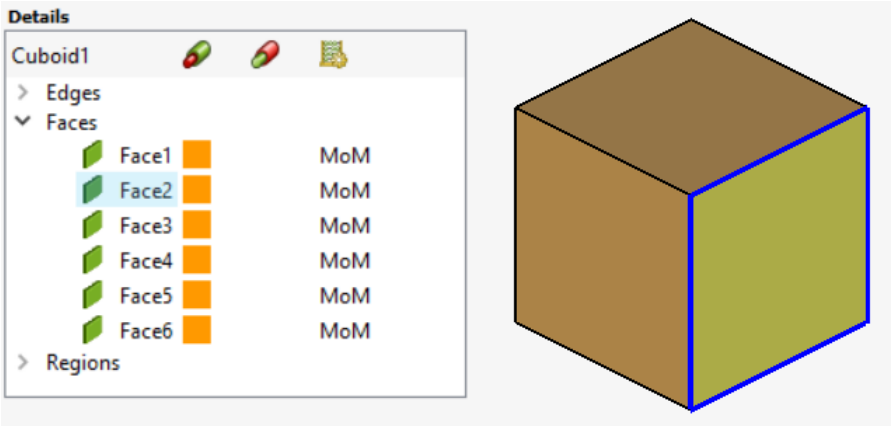


Figure 25: Select a face in the 3D view to highlight the corresponding entry in the details tree. The converse is also true.

The following can be applied to faces:

- Face medium
  - Metallic (to model skin effect).
  - Layered dielectric
  - Impedance sheets (to represent metal surfaces in cases where only the surface impedance per unit area is known).
  - Characterised surface
- Coating (layered dielectric)
- Dielectric sheet
- Local mesh size
- Solution methods
- Basis functions (local setting)

When an operation results in a face being split into multiple faces, both the resulting faces inherit the properties of the parent.

For operations where multiple faces need to be merged but have conflicting properties, an assumption will be made and the face will be marked suspect to indicate that the settings need to be reviewed.

Regions (Geometry)

A region is an enclosed volume. By default, a region is set to perfect electric conductor (PEC).

Selecting a region in the 3D view selects the corresponding region in the details tree. Conversely, selecting a region in the details tree selects the corresponding region in the 3D view.

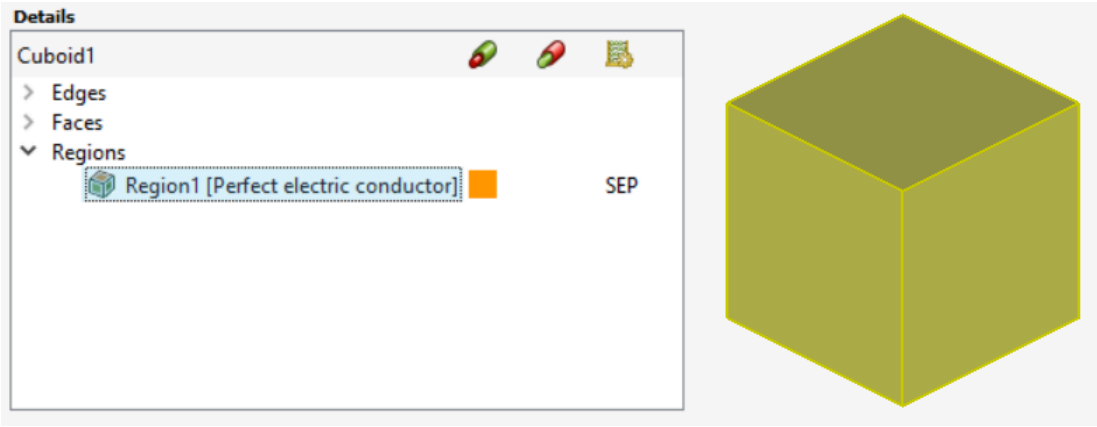


Figure 26: Select a region in the 3D view to highlight the corresponding entry in the details tree. The converse is also true.

The following can be applied to regions:

- Media
  - Dielectrics
  - Anisotropic media (3D)
- Local mesh size
- Solution methods
- Basis functions (local setting)

Boolean operations can be applied to the parents of regions. Where geometry operations introduce intersections of existing regions, and the parent regions have conflicting settings, the resulting regions are marked suspect to indicate that the settings need to be reviewed.

Note: Deleting a face that forms part of the region boundary effectively removes the region or merges the region with the surrounding region.

Any setting applied to a region is also used for faces bounding the region.

Attention: If the face has a conflicting setting, the face setting takes precedence over the region setting.

Changing Settings on Sub-Parts of a Model

It is recommended to change settings of an edge, wire, face or region on a root-level part. Settings changed on a sub-part might not propagate up the tree to the root-level part which could result in settings not being used during the simulation.

> **Note:** The highest-level items in the model are referred to as “parts”.

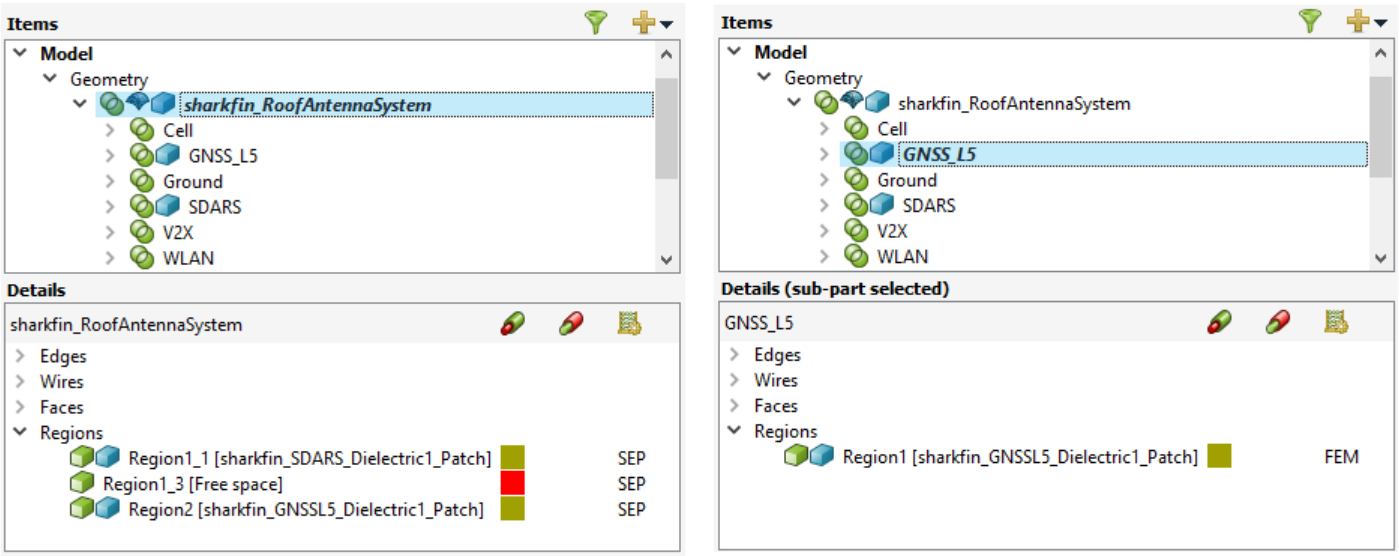


Figure 27: An example showing the root-level part of a model selected (left) and an example showing one of the sub-parts of a model selected (right). Note that setting the region to FEM for a sub-part did not propagate up the tree to the root-level part.

> **Note:** The text **(sub-part selected)** in the details tree indicates that the selected part is not a root level part in the model, see Figure 27 (on the right).

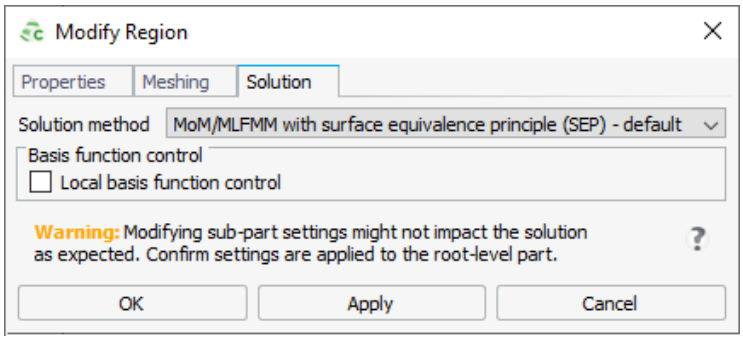


*Figure 28: A warning will be displayed on the dialog when modifying a **sub-part** face/edge/region/wire.*

Related tasks

Applying Media Settings

Applying a Local Mesh Size to a Wire, Edge, Face or Region

2.2.6 Status Bar

The status bar is the small toolbar that provides access to macro recording, general display settings, tools, selection method and type, snap settings and the model unit.

The status bar is located at the bottom-right of the application window. Options on the status bar are also available on the ribbon, but since the status bar is always visible, they are easily accessible no matter which ribbon tab is selected.

2.2.7 Notes View

The notes view is a rich-text editor tool that allows you to add comments to your model.

On the **Home** tab, in the **Create view** group, click the **Notes** icon.

The notes view opens with a basic template in a new window allowing you to use multiple computer screens (model on the one and notes view on another), but only a single notes view is supported for each model.

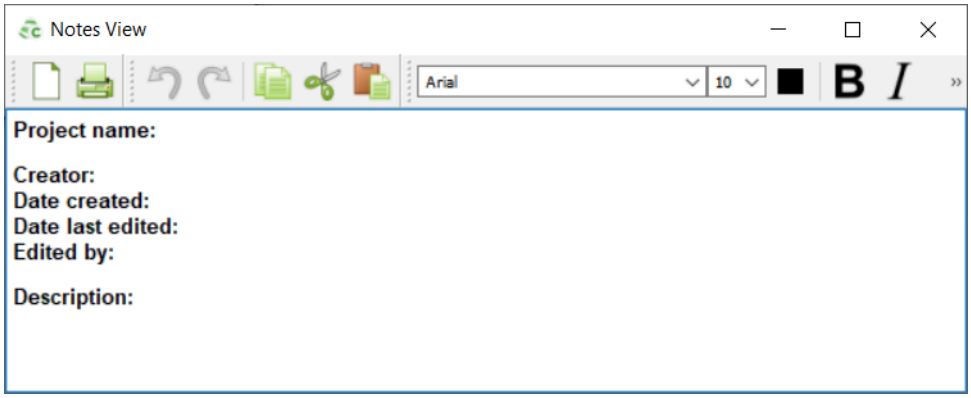


Figure 29: The notes view in CADFEKO. It is by default disabled.

The contents of the notes view are written to the top of the `.pre` file as a series of comments.

2.2.8 Notification Centre

The Notification centre performs computational electromagnetic model (CEM) validation and shows the status of the model and notifications.

The Notification centre lets you stay informed of the model status at all times. When problems in the model are detected, it is highlighted in the Notification centre with hyperlinks to the problematic entities.

The Notification centre can be hidden but the **Model Status** icon in the status bar will still indicate the current status of the model.

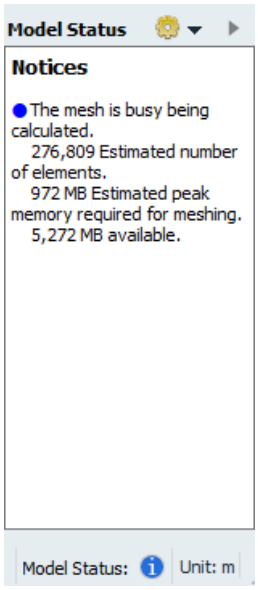


Figure 30: The Notification centre in CADFEKO. Note the Model Status icon at the bottom that shows the current status of the model.

Show or hide the Notification centre using one of the following workflows:

- Click the **Model Status** icon in the status bar.
- Drag the splitter from the right edge of the application to open the pane. To close, drag the splitter all the way to the right.
- On the **Home**, in the Validate group, click the **Model Status** icon.
- Click the icon in the Notification centre to hide the panel. Click the icon to show the panel again.

2.2.9 3D View

3D views are used to display and interact with the model. You can zoom, rotate and pan around a 3D model using the keyboard, mouse or a combination of both. You can use a 3D mouse, specify a view or select specific parts of a model. Multiple 3D views are supported.

2.2.10 Navigate the 3D View Using Keyboard and Mouse

Navigate the 3D view using a mouse, a keyboard or a combination of both.

Related concepts
Custom Keyboard Shortcut Settings
Custom Mouse Bindings

Panning the 3D view

Shift the location of the model (without any magnification) inside the 3D view.

Use one of the following methods to pan the 3D view:

- Press Ctrl and hold down the left mouse button. Drag the view.
- Hold down the middle mouse button. Drag the view.

Related reference
Pan the 3D view using the ribbon

Rotating the 3D view Angle

Rotate the model in the 3D view.

Press the left mouse button and drag the view.

Zooming to Extents

Zoom the model to the full extent of the 3D view.

Press F5 to use the keyboard shortcut.

Zooming In and Out

Zoom the 3D view to display the model at the desired scale.

Use one of the following methods to zoom the 3D view:

- Scroll the mouse wheel. Press Shift to slow down the zooming.
- Press Shift and hold down the left mouse button. Drag the view up or down.

2.2.11 Search Bar

The search bar is a single-line text field that allows you to enter search terms and find relevant information in the GUI or the documentation.

The search bar is located at the top-right of the application window.

Tip:

- Enter a search term in the search bar to populate a drop-down list of actions as well as the location of the action on the ribbon or context menu.
- Click an item in the list to execute the action.
- Partial searches are supported.
- Search the documentation.

2.2.12 Application Launcher

The application launcher toolbar is a small toolbar that provides quick access to other Feko components.

2.2.13 Application Menu

The application menu is similar to a standard file menu of an application. It allows saving and loading of models, print functionality and gives access to application-wide settings.

When you click on the application menu drop-down button, the application menu, consisting of two panels, is displayed.

The first panel gives you access to application-wide settings, for example:

- Creating a new model.
- Opening a model, saving a model and closing a model.
- Component library
- Import
- Export
- Model protection
- Print
- Check for updates
- Settings
  - Preferences
  - Keyboard shortcut settings
  - Mouse binding settings
  - Snap settings
  - Model unit
  - Solver settings

    - Component launch options
- Feko help
- About
    - Version information about CADFEKO
    - Information about Altair HyperWorks Products
    - Information about third-party libraries
- Exit

The second panel consists of a recent file list and is replaced by a sub-menu when a menu item is selected.

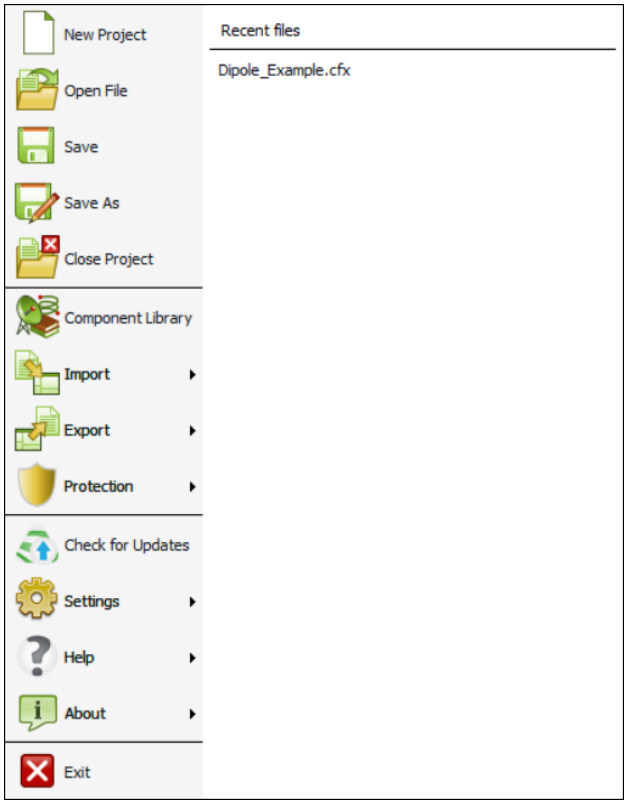


Figure 31: The application menu in CADFEKO.

Custom Keyboard Shortcut Settings

CADFEKO provides default keyboard shortcuts. To better fit your workflow and work style, you can reassign keyboard shortcuts to different commands.

To reassign keyboard shortcuts, click **File** > **Settings** > **Keyboard Shortcut Settings**.

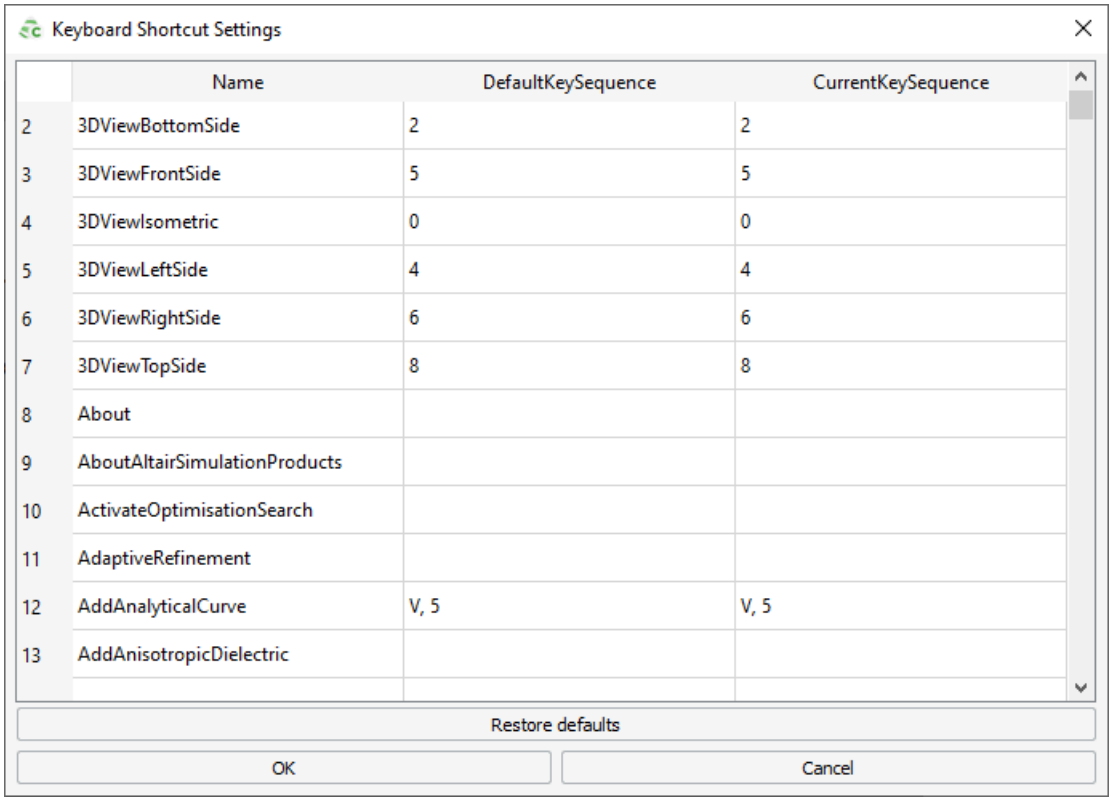


*Figure 32: The **Keyboard Shortcut Settings** dialog.*

For example, to change the shortcut key for the undo command, on the **Keyboard Shortcut Settings** dialog, click in the **CurrentKeySequence** column and enter the shortcut key that suits your work style.

Custom Mouse Bindings

CADFEKO provides default commands for all the mouse buttons. To better fit your workflow and work style, you can reassign mouse buttons to different commands.

To reassign mouse buttons, click **File** > **Settings** > **Mouse Binding Settings**.

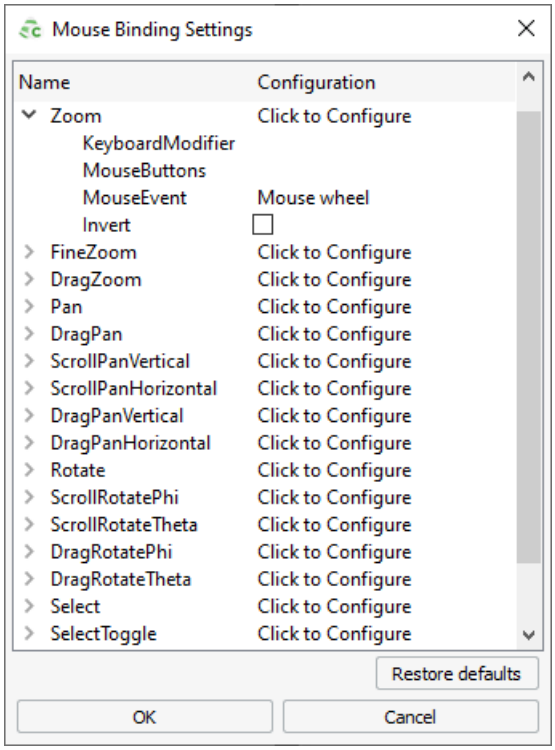


*Figure 33: The **Mouse Binding Settings** dialog.*

For example, to reverse the mouse wheel direction to better suit your workflow, on the **Mouse Bindings** dialog, click **Click to Configure**. On the **Zoom** dialog, select the **Invert** check box.

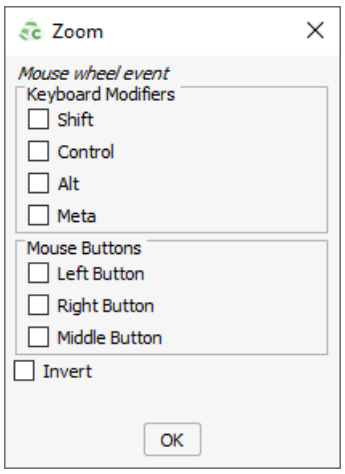


*Figure 34: The **Zoom** dialog.*

2.2.14 Feko Source Data Viewer

The Feko Source Data Viewer is a tool that allows you to view the currents per frequency for a PCB current data, defined using a `.rei` file.

In the model tree, under **Field/Current Data**, select a PCB current data. From the right-click context menu, click **Visualise PCB Current Data**.

The tool allows you to view multiple PCB current data definitions by using the **Import Currents Source data** tool to import additional current source data. For each currents source data (`.rei` file), you can specify the frequency, layers and nets that you want to view, as well as scale the layer height in the 3D view.

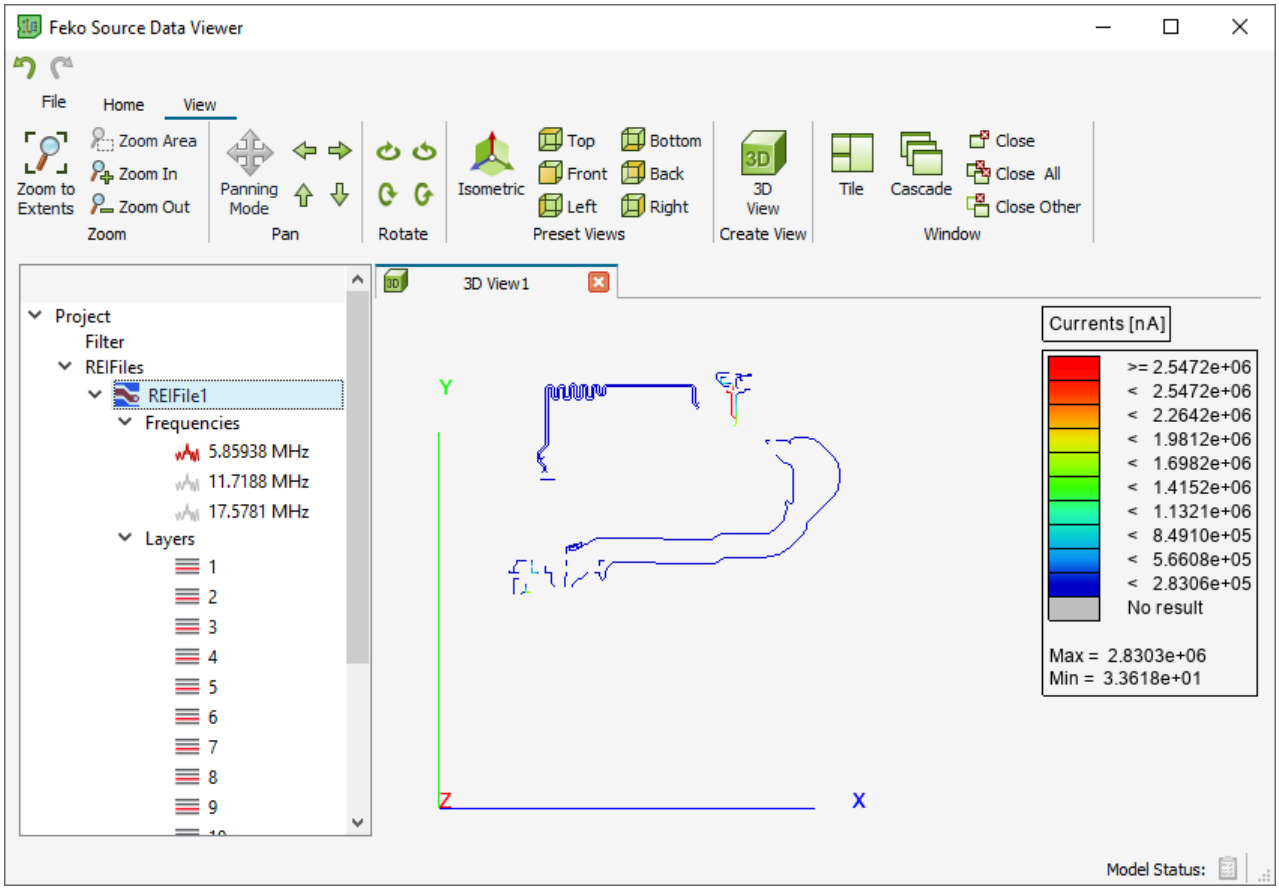


Figure 35: The Feko Source Data Viewer tool where you can view currents per frequency for a PCB current data definition.

Related tasks

Defining PCB Current Data from File

Visualising PCB Current Data

2.2.15 Help

The **Help** icon provides access to the Feko documentation.

Press F1 to access context-sensitive help. The context-sensitive help opens the help on a page that is relevant to the selected dialog, panel or view.

> **Tip:** When no help context is associated with the current dialog or panel, the help opens on the main help page that allows you to navigate the documentation or search in the documentation for relevant information.

2.2.16 Dialog Error Feedback

CADFEKO provides error feedback for dialogs by showing a soft message bubble when validation fails on a dialog.

Click the icon to show or hide the message bubble or click elsewhere in CADFEKO to hide the message bubble. The error feedback is also shown per tab when the validation fails on a multi-tab dialog.

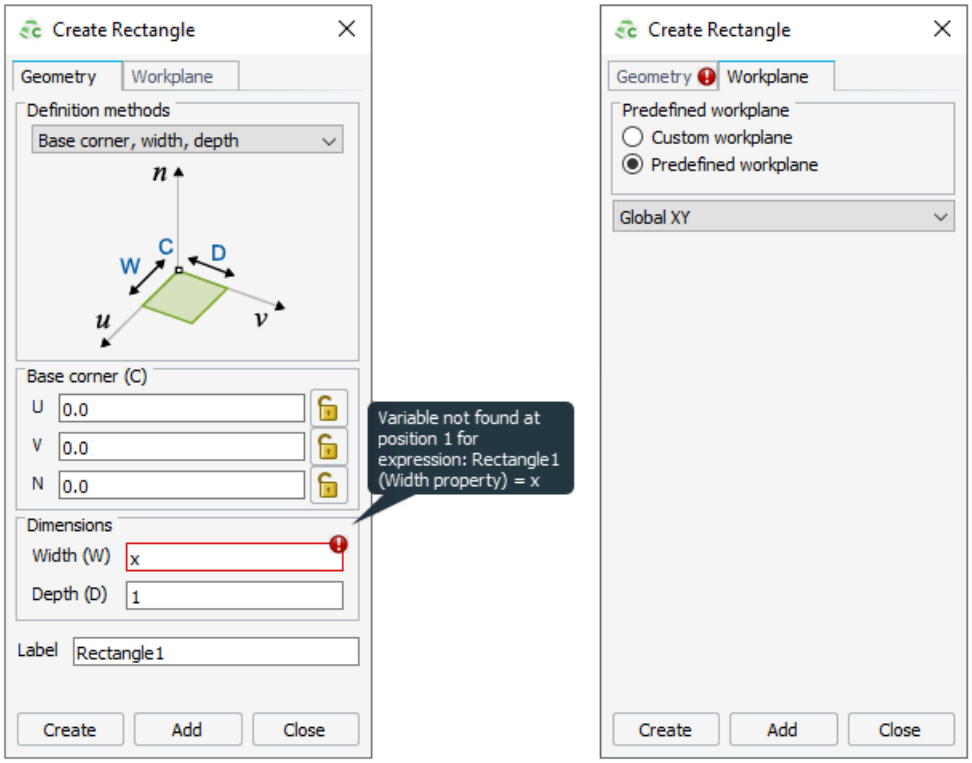


*Figure 36: The soft message bubble indicating that an undefined variable was used on the **Geometry** tab of the **Create Rectangle** dialog.*

2.2.17 Scripting

Use the application programming interface (API) to control CADFEKO from an external script.

Scripting allows repetitive or complex tasks to be performed in a script that would have taken a long time to perform manually. Scripts are created and edited in the script editor or scripts can be recorded (macro recording) by enabling the recording and then performing the actions in the graphical interface. The recorded script can be modified to perform a more complex task. Scripts that are used regularly can be added to the ribbon providing easy access and hiding the complexity of the script. Forms (dialogs) can be created in the scripting environment that obtain input from the script user without having to edit the script.

Script Editor

The script editor allows you to create scripts based on the Lua language to control CADFEKO, POSTFEKO and other applications as well as manipulation of data to be viewed and analysed further in POSTFEKO.

On the **Home** tab, in the **Scripting** group, click the  **Script editor** icon.

The script editor includes the following IDE (integrated development environment) features:

1. Syntax highlighting.
2. Intelligent code completion.
3. Indentation for blocks to convey program structure, for example, loops and decision blocks in scripts.
4. Use of breakpoints and stepping in scripts to debug code or control its execution.
5. An active console to query variables or execute simple commands.

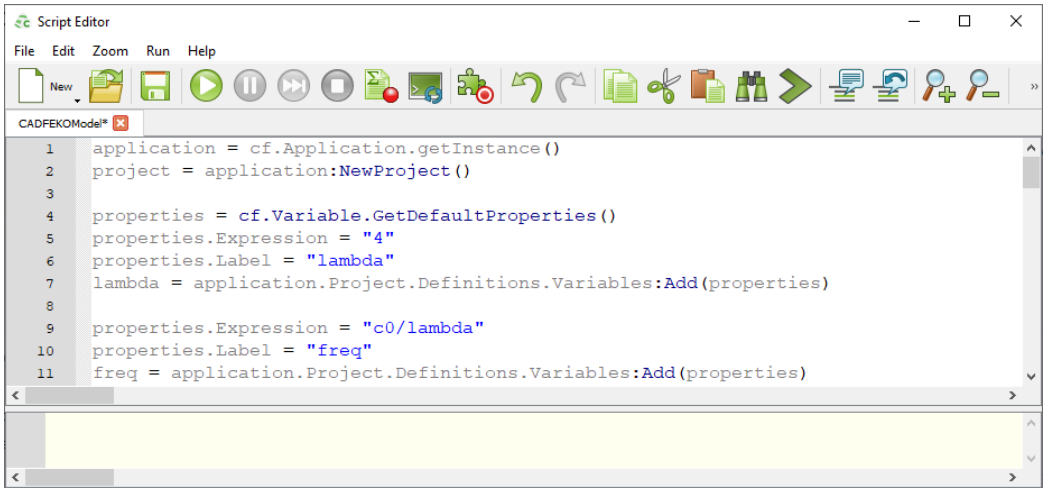


Figure 37: The script editor in CADFEKO.

Macro Recording

Use macro recording to record actions in a script. Play the script back to automate the process or view the script to learn the Lua-based scripting language by example. Macro recording allows you to perform repetitive actions faster and with less effort.

On the **Home** tab, in the **Scripting** group, click the **Record Macro** icon.

Application Macros

An application macro is a reference to an automation script, an icon file and associated metadata.

Application macros are available directly or can be added, removed, modified or executed from the application macro library.

> **Tip:** A large collection of application macros are available in CADFEKO and POSTFEKO.

On the **Home** tab, in the **Scripting** group, click the **Application macro** icon.

Related concepts
CADFEKO Application Macros
POSTFEKO Application Macros

2.3 Preferences

CADFEKO has various default settings that you can configure to customise it to your preference.

On the application menu, click **Settings** > **Preferences**. The settings can be reset to the default settings at any time, restoring the settings to the state of a new installation.

Many of the settings are applied immediately, but some of the settings such as 3D view font changes require the application to be restarted before the changes take effect.

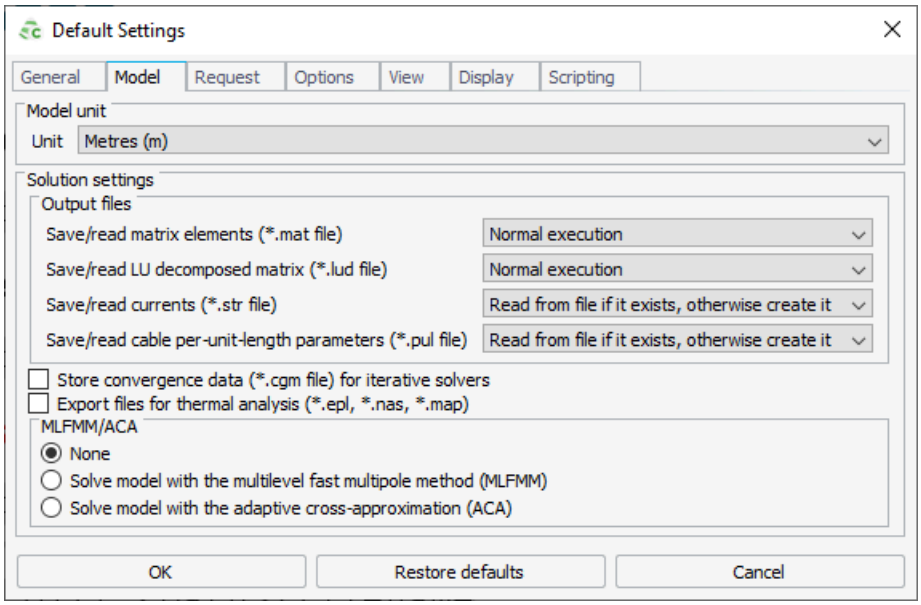


*Figure 38: The **Default settings** dialog.*

2.4 Saving a Model

Store a CADFEKO model and calculation requests to a `.cfx` file to reopen later.

On the **Home** tab, in the **File** group, click the  **Save** icon.

The model is saved to a `.cfx` file. When saving a model, the following files are also created:

- `.cfm` (if the model has a mesh)
- `.pre`

2.5 3D View

3D views are used to display and interact with the model. You can zoom, rotate and pan around a 3D model using the keyboard, mouse or a combination of both. You can use a 3D mouse, specify a view or select specific parts of a model. Multiple 3D views are supported.

2.5.1 Point Entry

Use point entry (Ctrl+Shift+left click) to add values from the 3D view (for example, coordinates, faces and edges) or values from the model tree or details tree (for example, named points, variables, workplanes, faces and edges) to point-entry supported fields on a dialog.

Point entry is the mechanism of entering values in a field that has focus (on a dialog) based on the Ctrl+Shift+left click in the 3D view or model tree. It allows the spatial definition or editing of geometry or solution requests based on a series of clicks in the 3D view (or tree).

> **Note:** A field on a dialog that has focus and a yellow outline, indicates that point entry is active and allowed. Often multiple fields will be active for point entry at the same time.

For a one-dimensional input field (for example, the radius of a sphere), the value is calculated based on the distance between the specified point and the coordinates or values already defined in the previous fields on the dialog (for example, the centre of the sphere).

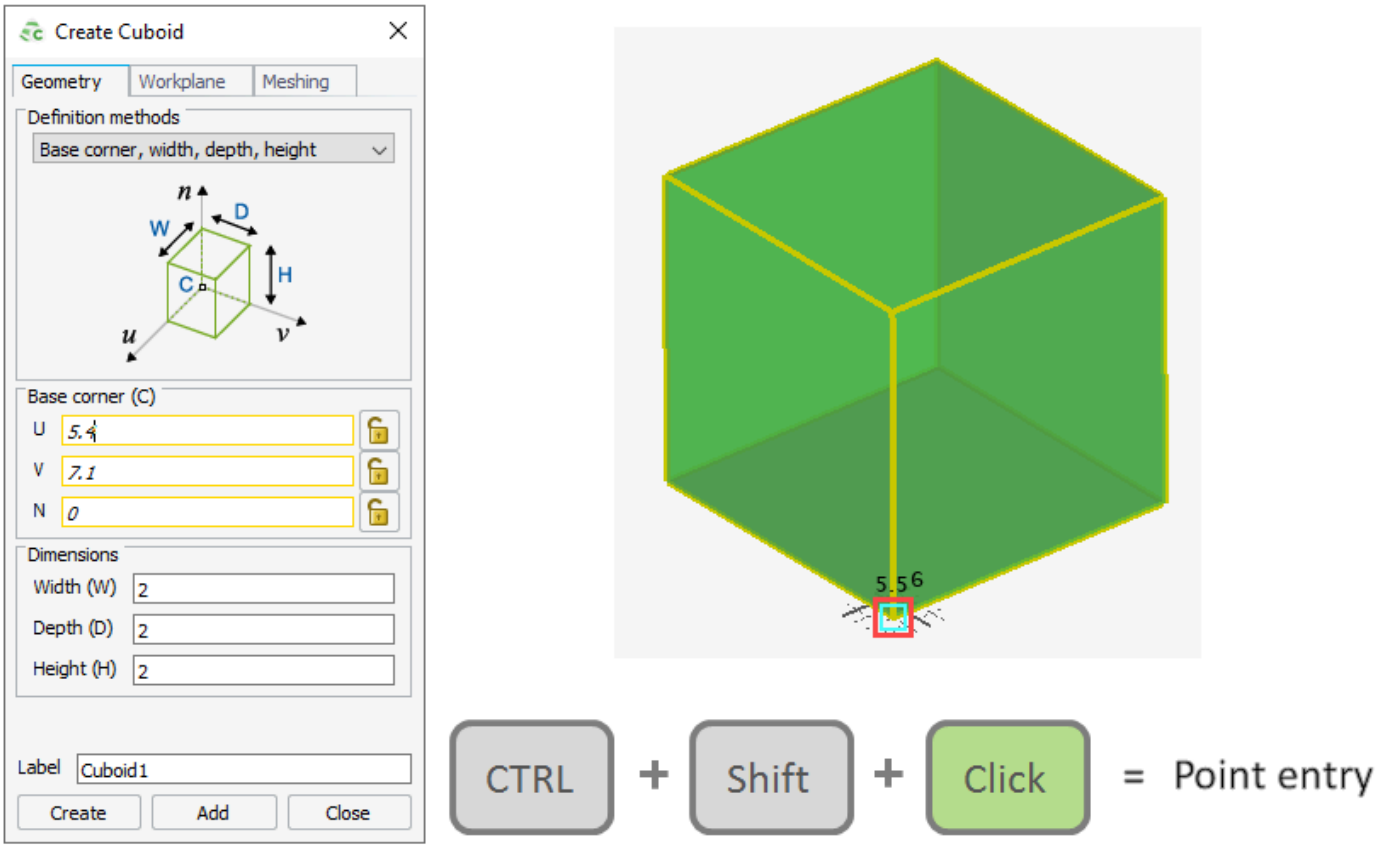


*Figure 39: The **Base corner** field has focus. The yellow outline indicates that point entry is active for that field. The values in italic are a preview of the values.*

Using Point Entry to Add Coordinates to a Dialog

Use point entry to add coordinates from the 3D view to fields on a dialog.

As an example, specify the base corner of a cuboid by snapping to a point in the 3D view and use point entry to add the coordinates to the dialog.

1. Open the **Create cuboid** dialog.
2. Verify that the **Base corner** field is outlined in yellow.
3. Press and hold down the Ctrl+Shift keys while hovering with the mouse cursor over a specific point in the 3D view.
4. Left click on the red rectangle to snap to that point. You can now release the Ctrl+Shift keys. The coordinate of the snapped point is added to the **Base corner** fields and the focus has moved to the next field.

Using Point Entry to Add Named Points to a Dialog

Use point entry to add the coordinates of a named point in the model tree to fields on a dialog.

As an example, specify the base corner of a cuboid using point entry to add the coordinates of the named point (in the model tree) to the dialog.

1. Open the **Create cuboid** dialog.
2. Verify that the **Base corner** field is outlined in yellow.
3. Press and hold down the Ctrl+Shift keys while hovering with the mouse cursor over a named point in the model tree.
4. Left click on a named point in the model tree to perform point entry on the named point. You can now release the Ctrl+Shift keys.
   The named point is added to the **Base corner** fields and the focus has moved to the next field.

Lock Point Entry Fields

Lock an individual field in a collection of fields that accepts multiple components from a single point entry (for example, the U coordinate, V coordinate and N coordinate for a point).

A field which accepts multiple values from point entry has a button next to it. If the button is clicked, the button indicates that the field maintains its current value and will not be updated during point entry.

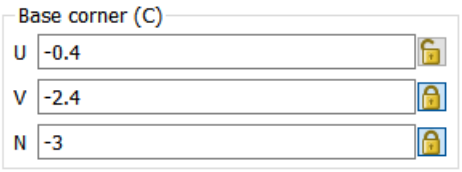


Figure 40: The button indicates that the point entry field is unlocked.

> As an example, use point entry to enter the coordinates of a point in the 3D view as the base corner of a cuboid. Lock the X coordinate and Y coordinate and use point entry again to enter a different Z coordinate.

2.5.2 Snapping to Points in the 3D View

Snap to points (for example, named points, geometry points, geometry face centre, geometry edge centre, mesh vertices and grid) in the 3D view.

1. Press and hold down Ctrl+Shift while hovering with the mouse cursor over the model.

Note:

- An active snapping point is indicated by a dot
- Special snapping points near the mouse cursor are indicated by a dot with a cyan outline.
- A preview of the geometry is displayed in green (transparent).

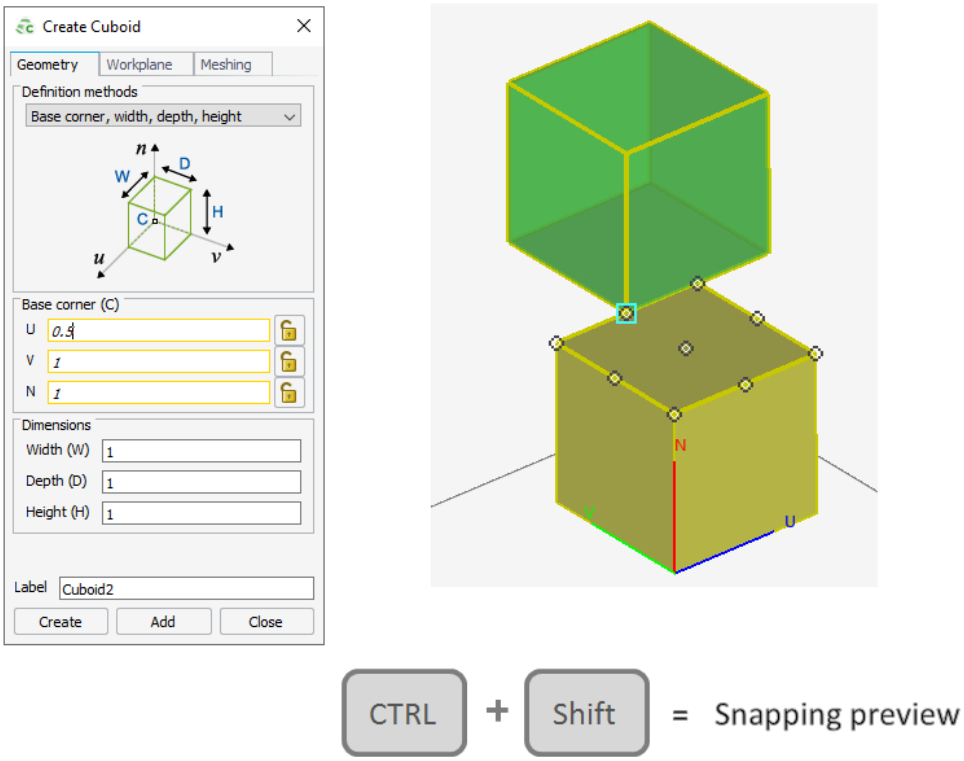


Figure 41: Press Ctrl+Shift to view the snapping preview.

2. Left click on a dot and release Ctrl+Shift to snap to that point.

   When snapping to align a new workplane, the history of the starting point and the route followed to the destination points, affects the orientation of the workplane (for example the orientation of an edge).

Snapping Settings

Specify the snapping targets and workplane grid that apply when pressing Ctrl+Shift.

On the **Tools** tab, in the **Snapping** group, click the **Snap Settings** icon.

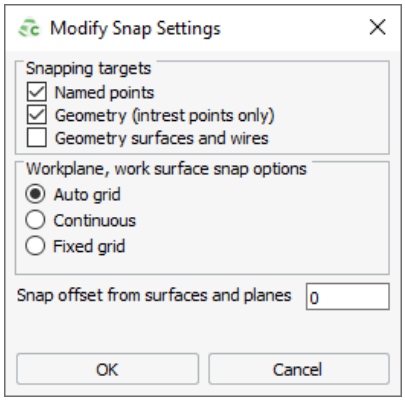


*Figure 42: The **Snapping settings** dialog.*

Snapping targets

You can specify the type of snapping targets that apply when pressing Ctrl+Shift. The following snapping targets are available:

- Named points
- Geometry (interest points only)
- Geometry surfaces and wires
- Mesh

Workplane / Work surface snap options

You can specify how to snap to points on the workplane.

Auto grid
: The workplane grid size is determined automatically. You can snap to any point on the grid lattice.

Continuous
: You can snap to any point on the workplane.

Fixed grid
: The workplane grid is specified by **Size**. You can snap to any point on the grid lattice.

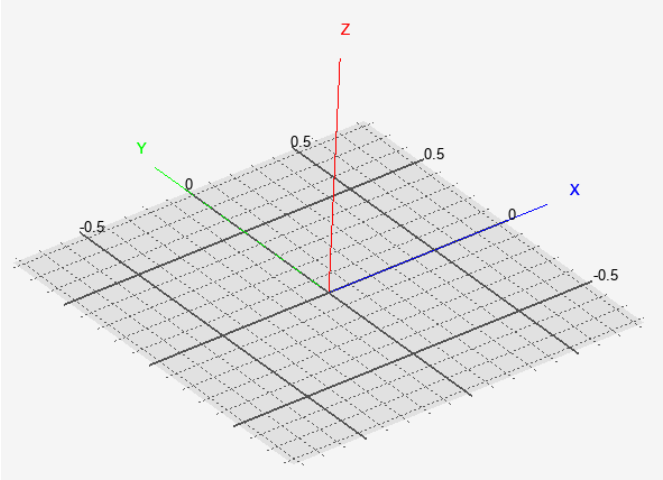


Figure 43: Enable the workplane grid display and tick marks to view the grid lattice.

Snapping Offset

You can specify the snap offset from a surface or plane. In the **Snap offset from surfaces and planes** field, enter a value for the offset.

> **Tip:** Specify the snap offset to define a cable path at an offset from complex geometry.

2.5.3 Navigate the 3D View Using the Ribbon

Navigate by means of panning, rotating and zooming the 3D view using the ribbon.

All 3D view interactions are available on the ribbon. It is not practical for advanced users to use the ribbon for 3D interactions, but the ribbon provides a list of all the interactions. Hovering over a ribbon button will show the tooltip that also shows the keyboard shortcut for that action (if a shortcut exists for that action).

Pan

View the list of the available panning methods using the ribbon.

The zoom settings are found on the **View** tab, in the **Panning** group.

Icon	Icon text	Description
	**Panning Mode**	Places the mouse cursor in panning mode.
	**Pan Left**	Pan to the left.
	**Pan Right**	Pan to the right.
	**Pan Up**	Pan up.
	**Pan Down**	Pan down.

Rotate

View the list of the available rotation types using the ribbon.

The zoom settings are found on the **View** tab, in the **Rotate** group.

Icon	Icon text	Description
	**Phi (-)**	Rotate the model in the negative ϕ direction.
	**Phi (+)**	Rotate the model in the positive ϕ direction.
	**Theta (-)**	Rotate the model in the negative θ direction.
	**Theta (+)**	Rotate the model in the positive θ direction.

Zoom

View the list of the available zoom methods using the ribbon.

The zoom settings are found on the **View** tab, in the **Zoom** group.

Icon	Icon text	Description	Shortcut
	Zoom to Extents	Zoom the content of the window to its extents.	F5
	Zoom Area	Zoom to display an area specified by a rectangular window.	
	Zoom In	Zoom in on the contents of the window.	+
	Zoom Out	Zoom out from the contents of the window.	-

2.5.4 View Settings

Specify the origin, view direction and zoom distance for a 3D view to allow you to consistently reproduce a view for reporting or set the 3D view to a predefined view.

On the **View** tab, in the **View Manipulation** group, click the **Transform View** icon.

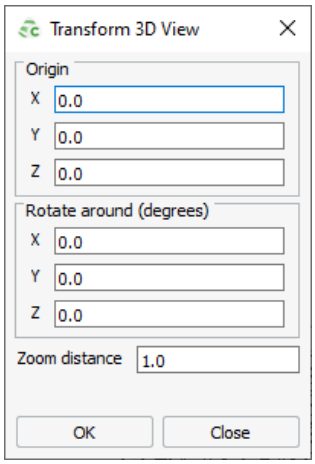


*Figure 44: The **Transform view** dialog.*

Predefined Views

View the list of available predefined views.

The predefined view settings are found on the **View** tab, in the **View Manipulation** group.

Icon	Icon text	Description	Shortcut
	Isometric	Displays an isometric view of the model.	0
	Top	Displays a top view of the model.	8
	Front	Displays front view of the model.	5
	Left	Displays a left view of the model.	4
	Bottom	Displays a bottom view of the model.	2
	Back	Displays a back view of the model.	Ctrl+5
	Right	Displays a right view of the model.	6

Depth Lighting

Depth lighting adds depth perception to a model visualised in the 3D view.

Depth lighting is enabled by default, but you can disable this setting for specific views.

On the **View** tab, in the **View Manipulation** group, click the **Depth Lighting** icon.

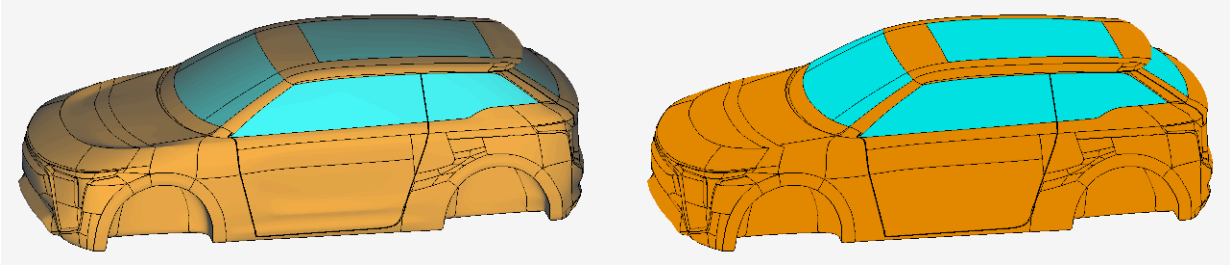


Figure 45: A model with depth lighting enabled (on the left) and a model with depth lighting disabled (to the right).

2.5.5 Selection in the 3D View

Left-click on a part of a model in the 3D view to select it.

Selection in the 3D view is set to **Auto** selection by default. Auto selection cycles through the available selection type each time you left-click on a model in the 3D view.

The selection type is specified at the following locations:

- **Tools** tab, in the **Selection** group

- Status bar

> **Tip:**
> - Press Ctrl+A to select all entities (edge, wire, face or region) of the same type in the collection[10].
> - Press Ctrl+Shift+A to select all entities (edge, wire, face or region) of the same type in the model.

Selection Method

View the list of available selection methods.

Icon	Icon text	Description
	**Single Select**	Select the item under the mouse cursor.
	**Rectangle Select**	Select all items in the rectangle. Click once to place the first corner of the rectangle. Move the mouse cursor and click again to indicate the second point in the rectangle.
	**Polygon Select**	A polygon selection area is defined using successive clicks. The elements inside the polygonal area are selected.

Selection Type

View the list of available selection type settings.

The select by type settings are found on the **Tools** tab (**Selection** group).

Icon	Icon text	Description
	**Auto**	When this option is selected, the selection will cycle through the available selection types.
	**Geometry parts**	Select geometry parts in the 3D view.
	**Faces**	Select faces in the 3D view.

10. For example, in the model tree, a collection can be geometry, meshes, ports, meshing rules, cutplanes and solution settings. In the details tree, a collection can be wires, edges, faces and regions.

Icon	Icon text	Description
	Edges / wires	Select edges / wires in the 3D view.
	Regions	Select regions in the 3D view.
	Mesh Parts	Select mesh parts in the 3D view.
	Mesh Label	Select mesh label in the 3D view.
	Mesh Element	Select mesh element in the 3D view.
	Mesh Vertex	Select mesh vertex in the 3D view.

Edge Selection Tool

Use the edge selection tool when selecting edges for the hole filling tool to minimise the number of edges that need to be selected manually.

On the **Tools** tab, in the **Selection** group, click the **Selection Tools** icon. From the drop-down list select the **Select Edge Loop** icon.

When this tool is activated, the smallest loop containing the already selected laminar and / or free edges (wires) edges is selected. A laminar edge is an edge that is associated with a single face.

Note: An edge is laminar when the edge is only on the boundary of a single face.

Tip: Press Q+C to select the smallest loop containing the edges.

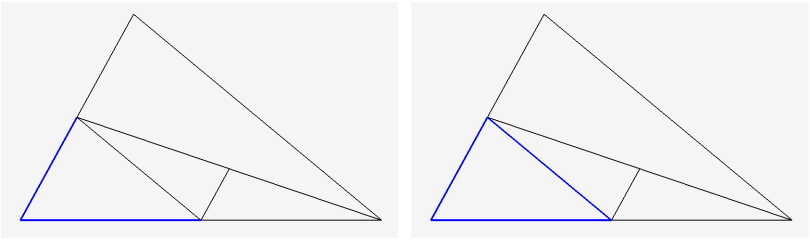


*Figure 46: Example 1 showing two selected wires. Using the **Select edge loop** tool results in the smallest loop containing these edges being selected.*

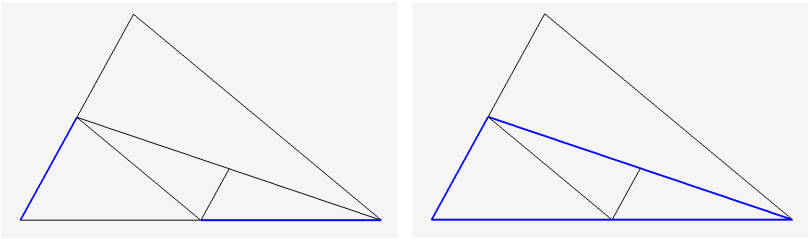


*Figure 47: Example 2 showing two selected wires. Using the **Select edge loop** tool results in the smallest loop containing these edges being selected.*

Selection History

Selection operations can be undone / redone independently of any geometry modifications.

The selection actions are found on the **Tools** tab (**Selection** group).

Icon	Icon text	Description
	**Undo Selection**	Undo the last selection.
	**Redo Selection**	Redo the last selection.

2.5.6 Show / Hide

Items in the 3D view can be shown or hidden.

For geometry or mesh in the model tree or edges, wires, faces or regions in the details tree, from the right-click context menu, click **Visibility** and select one of the following tools:

- *Show Only (Isolate)*
  The tool shows the selected item while hiding all other items.
- *Show / Hide (Toggle)*
  The tool toggles the visibility of selected parts.
- *Show*
  The tool shows the selected items in the model (makes hidden items visible).
- *Hide*
  The tool hides the selected items in the model (makes visible items hidden).
- *Show All in Model*
  The tool shows all items in the model.

2.5.7 Changing the Rendering Speed for a Model

Improve the rendering speed of a large model in the 3D view at the cost of visual quality.

On the **View** tab, in the **Show** group, click the **Speed** icon. From the drop-down list, select one of the following options:

- *Default*

  Finer tessellation results in high quality rendering, but rendering speed is slower.

- *Fast*

  Coarser tessellation results in a medium quality rendering, but rendering speed is faster.

- *Faster*

  Coarsest tessellation results in low quality rendering, but rendering speed is the best.

2.6 Model Protection

Protect a model with a password.

The primary usage of model protection is to allow a prepared simulation model to be shared as a "component" that may be included in the construction of another model, while maintaining limited visibility of the internal details of the model as well as the simulation quantities for anyone who does not know the password for the protected model. When a protected model is imported, only its representation geometry, if defined, and bounding box are visible. The actual model structure (geometry and/or mesh parts) is neither visible nor editable.

Note: Limitations are imposed on the requests that may be calculated for a model that includes a protected model.

Model protection may also be used to ensure that those who do not have the password are unable to open the model. When using protection in this way, please keep in mind that, though a protected model can be simulated by running the solver, some simulation results may not be calculated and no mesh will be available in POSTFEKO for post-processing after simulation.

Tip: Unprotect your model before simulation to avoid any limitations.

2.6.1 Setting Up a Protected Model

Add a password to a model and configure the parts that can be accessed by those who do not have the password.

To add protection, on the **Home** tab, click the **Protection** icon and in the drop-down list click **Protect Model**. Enter a new password in the dialog. To remove protection, click **Unprotect Model** and enter the password that was used to protect the model. When a model is protected, a protection indicator is shown in the bottom right hand corner of the CADFEKO window and the **Protection** contextual tab becomes available on the ribbon.

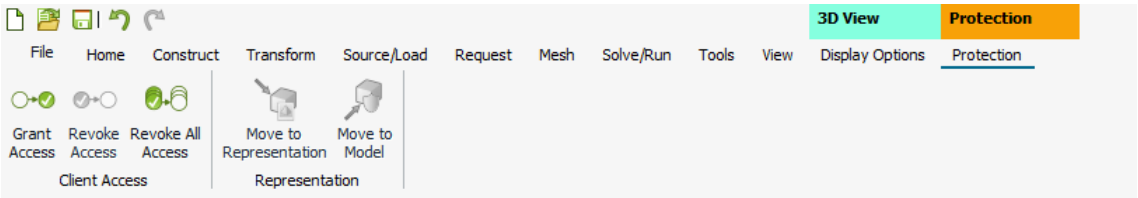


*Figure 48: The ribbon in **CADFEKO** with the **Protection** contextual tab active.*

Access can be granted to any workplanes and ports in the model that the creator wishes to expose to those who do not have the password. These workplanes and ports are indicated by the icon next to the items in the tree. The subsequent user of the protected model will be able to add souces and loads to the accessible ports and use the accessible workplanes to accurately align or position the protected model structure relative to other geometry, assumably with additional information provided by the

creator of the model to guide the user thereof. Access can be revoked to return any port or workplane to the protected model and obscure it from subsequent users who do not possess the password.

A protected model has an additional **Unprotected Information** branch in the model tree (**Construction** tab) that shows items such as its representation geometry that will be visible to those who do not have the password.

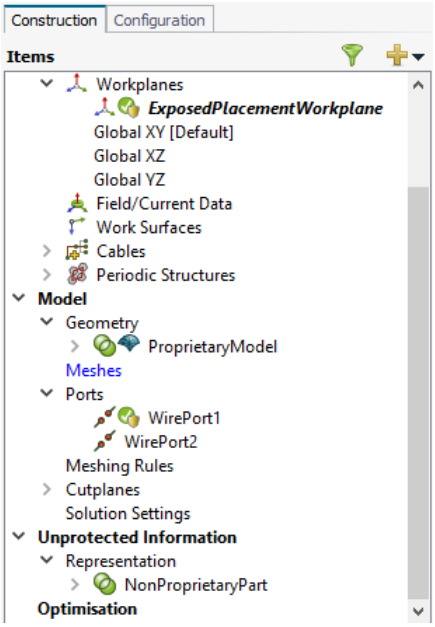


Figure 49: The model tree of a protected model. Note the Unprotected Information section and the indicator icons next to the workplane and the port to which access have been granted.

The protected model representation is created by moving geometry from the model to the representation. The representation is intended solely for visualisation purposes. It does not get meshed and does not form part of the simulation.

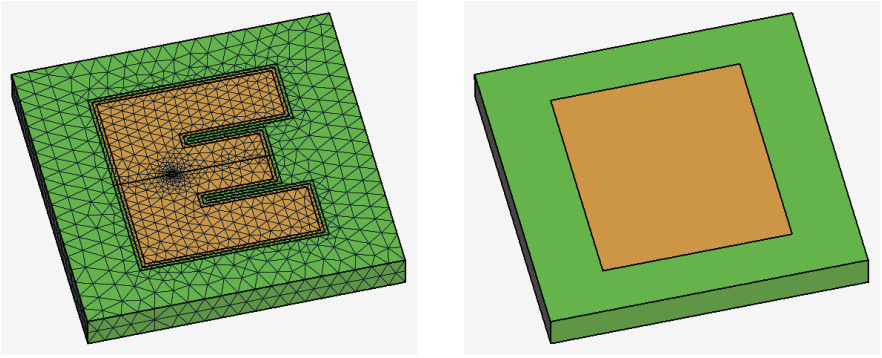


Figure 50: On the left, a model that a user wishes to protect. All the details of the model is visible when it is opened with the password. On the right is the representation geometry that the user defines to be visible to the client. Note that representation geometry is for visualisation only and is not meshed.

When protection is added to a model, CADFEKO will check if the solver selected in the protected model is supported. A subset of solvers are currently supported for protected models to avoid solver combinations and settings that are not supported in the same model. If the solver of the model to

be protected is not currently supported, a warning will be given and protecting the model will not be allowed.

> **Note:** Supported solver combinations will be extended in future versions.

2.6.2 Importing and Using Protected Models

A password-protected model can be incorporated in simulation with other geometry by users who do not have the password and cannot view the contents of the model or make changes to it.

To import a protected model as a component to be used as part of a simulation with other geometry, mesh or requests, click the **Protection** icon (on the **Home** tab) and click **Import Protected Model**. No password is necessary to import the protected model in this way. Once a protected model has been imported, the bounding box of the protected model as well as any representation geometry it contains are shown in the 3D view. The imported model will be listed in the model tree (**Construction** tab) under the **Protected Models** branch.

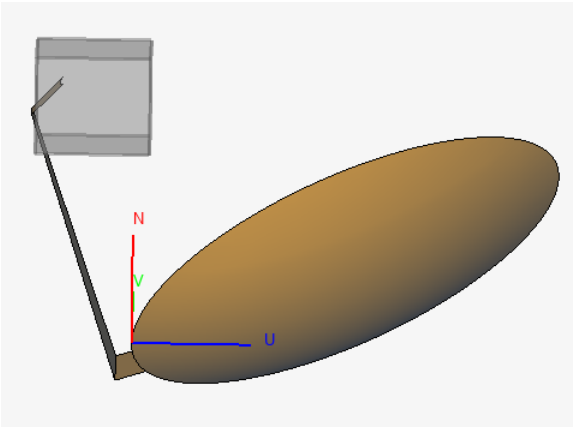


Figure 51: An offset reflector with an imported protected model of a horn (feed) shown in grey in CADFEKO. Note that this protected model does not have a representation defined, therefore only its bounding box gets displayed.

Using an Imported Protected Model

Transforms (such as Translate, Rotate and Scale) can be applied to imported protected models in order to prepare and position them correctly relative to other geometry and requests before simulation. Accurate alignment of a protected model can be achieved using the **Align tool** and by referencing the accessible workplanes for that model. Accessible workplanes from all imported protected models are included in the list of predefined workplanes in the relevant dialogs and may be used just as any other workplane.

Sources and loads may be applied to ports inside the protected model if the creator of the model set up any workplanes or ports in such a way that they can be accessed by the subsequent user.

Rename, **Reload**, **Expose** or **Conceal** a protected model from the right-click context menu in the model tree (**Construction** tab).

Exposing the model requires that the password for the protected model be entered. When exposed, the geometry, mesh and definitions inside the protected model will be shown. This is useful when debugging or visualising the usage of protected models for which a user knows the password. Concealing the model will again hide the protected contents inside the protected model.

Reloading a protected model will re-import the model from the location that it was initially imported from. A successful reload is only possible if a model with the same filename as the model that was initially imported is located in the same path (absolute or relative) when the reload is triggered. Transforms applied to the protected model will be maintained during reload. Some changes made to simulation configurations linked to that model (such as renaming, deleting or disabling a configuration) will not be maintained. The mapping of entities during a reload is performed based on entity labels. If the protected model has an accessible port "Port1", and then that label is changed to "ConnectionPort" in the protected model and the protected model is reloaded, the original port will not be associated with "Port1" even if there are no other ports in the model.

Simulation configurations from the imported protected model (with some details hidden) are loaded into new configurations denoted **ProtectedModelName.ConfigurationName** where ProtectedModelName is the name of the protected model in the model tree (**Construction** tab) and ConfigurationName is the name of the configuration in the imported protected model. Both the protected model and its configurations may be renamed after import.

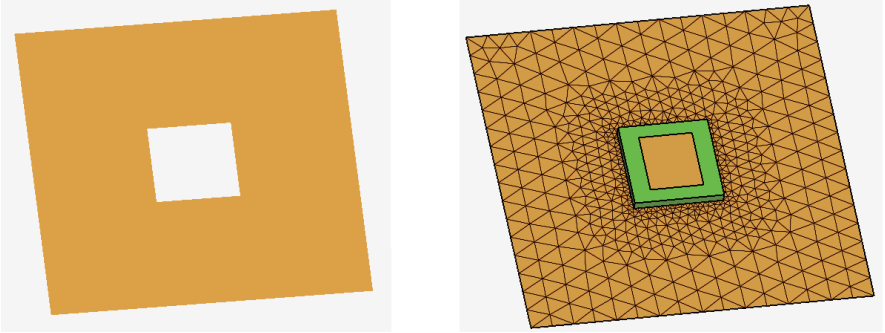


Figure 52: On the left, a model into which a user wishes to import the protected model from the previous section. Note that there are no parts that would overlap with the imported protected model. On the right, the protected model was imported into the model and the model was meshed. The parts touch and were meshed to be connected. Note that the user, who is not in possession of the password, can only see the representation geometry that was defined as unprotected information inside the protected model. They cannot see the other contents and the mesh of the protected model. If the user were to obtain the password, they would be able to Expose the protected model which would show the contents of the model and its mesh.

2.6.3 Model Protection Limitations

There are various restrictions on simulations that include password-protected models. Some restrictions are imposed to protect the information contained inside the protected model, while other restrictions will be lifted in future releases,

The process of running a simulation with a model that includes imported protected models is the same as for a model that does not contain imported protected models. Various limitations on output file formats and solution output will be imposed when parts of the simulation are protected. Some parts

of text files (such as the `.pre` file) will be encrypted and will not be editable or readable. The solver will limit details written to the screen as well as output files and certain output files, such as export of files for thermal analysis, are not supported and if requested, the setting to write out these files will be ignored. Some requests, most notably currents, will be excluded from the simulation even if the requests are defined. Near fields can be requested and will be calculated, but only the results that fall outside the bounding box of the protected part will be available. Warning and error messages in the CADFEKO notification centre will indicate where these limitations are applied.

Note: Limitations on calculation requests will be relaxed in future releases.

`Fek` files which include protected models cannot currently be loaded into POSTFEKO. This means that models that contain protected parts cannot be visualised in the 3D view. Results can be plotted on 2D and surface graphs and post-processed as normal. Results that can be viewed in the 3D view (such as far fields) need to be copied to a custom dataset before adding them to a 3D view. This restriction will be relaxed in future releases.

Table 1: Supported solver combinations of the protected model and the rest of the model.

Protected Model	Rest of Model	Info
SEP	Windscreen	Protected model contains dielectric(s) using SEP. Rest of model contains mesh elements solved with the Windscreen method.
SEP	SEP with MLFMM	Protected model and rest of model contain dielectric(s) using SEP. The MLFMM solver is activated.
SEP	FEM	Protected model contains dielectric(s) using SEP. Rest of model contains dielectric(s) using FEM.
MoM	UTD	Protected model contains metallic-only faces using MoM. Rest of model contains faces set to UTD.
MoM	RL-GO	Protected model contains metallic-only faces using MoM. Rest of model contains faces set to RL-GO.
MoM	PO	Protected model contains metallic-only faces using MoM. Rest of model contains metallic-only faces set to PO.
PO	MoM	Protected model contains metallic-only faces using PO. Rest of model contains metallic faces set to MoM.
PO	MLFMM	Protected model contains metallic-only faces using PO. The MLFMM solver is activated.

2.7 Model Definitions

Define the model unit, variables, named points, workplanes and the model extents for the model.

2.7.1 Model Unit

The model unit specifies the unit that is used for all dimensions in the model.

When you modify the model unit, the unit does not modify any numbers specified in CADFEKO, but rather the internal interpretation of all numbers created before and after the unit change.

> **Note:** You may change the model unit at any stage during and after the construction of the model.

The model unit is specified at the following locations:

- **Home** tab, in the **Model Attributes** group
- **Construct** tab, in the **Define** group
- Status bar

Changing the Model Unit

Change the model unit that is used for all dimensions in the model.

1. On the **Home** tab, in the **Model Attributes** group, click the **Model Unit** icon.

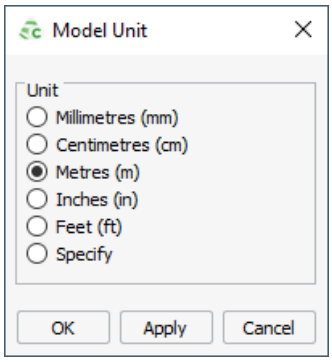


*Figure 53: The **Model unit** dialog.*

2. Specify the unit you want to use in the model.
   - To use one of the standard units, click the unit you want to use for the model.
   - To specify an arbitrary unit conversion factor with respect to metres, click **Specify**. In the **m** field, enter a value.

     > **Note:** For example, if you want to change the unit to micrometres, enter 1e-6 to specify a conversion factor of $1x10^{-6}$.

3. Click **OK** to close the dialog.

2.7.2 Variables

Create a fully parametric geometry in CADFEKO by using variables and mathematical expressions.

Most input fields in CADFEKO allow variables and expressions to be entered. The variables and expressions are stored as part of the model. When a variable is modified, any items referencing that variable are re-evaluated and updated.

Variable names

- The first character must be either:
  - Alphabetic (for example, a - z, A - Z)
  - Underscore (for example, "_")
- The remaining characters may be alphanumeric or an underscore (for example, a - z, A - Z, 0 - 9 and "_").
- Variable names are case-insensitive.

Variable expressions

- A variable expression may be a single value.
- A variable expression may be a mathematical expression using round brackets and the operators +, -, *, \ and ^ (exponential notation).
- A variable expression may reference other variables.
- A variable expression may use trigonometric and other built-in functions.
- A variable expression may use any of the predefined variables in CADFEKO.

Related reference
Math Functions in CADFEKO

Defining a Variable to Create Parametric Geometry

Create a variable to create parametric geometry.

1. On the **Construct** tab, in the **Define** group, click the **Add Variable** icon.
2. In the **Label** field, enter a name for the variable.
3. In the **Expression** field, enter a value, expression or an already defined variable.
4. [Optional] In the **Comment** field, add a comment or description for the variable.
5. [Optional] Select the **Limit** check box to define a range for the value of the variable.
   - In the **Minimum** field, enter a value for the smallest variable value.
   - In the **Maximum** field, enter a value for the largest variable value.
6. Select one of the following workflows to close the dialog.
   - To create the variable and close the dialog, click **Create**.

- To create the variable, but keep the dialog open to create another variable, click **Add**.

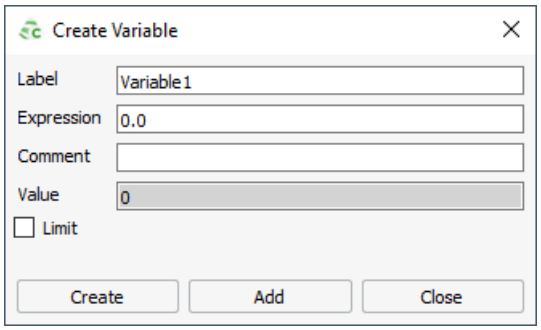


*Figure 54: The **Create Variable** dialog.*

Related reference
Math Functions in CADFEKO

Predefined Variables

A new model contains a list of predefined variables by default.

A predefined variable may be deleted or modified and only have an effect if you explicitly refer to that variable.

Table 2: Predefined variables in CADFEKO.

c0	The speed of light in free space in m/sec.
eps0	The permittivity of free space in F/m.
mu0	The permeability of free space in H/m.
pi	The mathematical constant π (Ludolph's number).
zf0	The characteristic impedance of free space in Ohm.

Modifying Multiple Variables

Modify multiple variables on a single dialog.

1. On the **Construct** tab, in the **Define** group, click the **Edit Variables** icon.

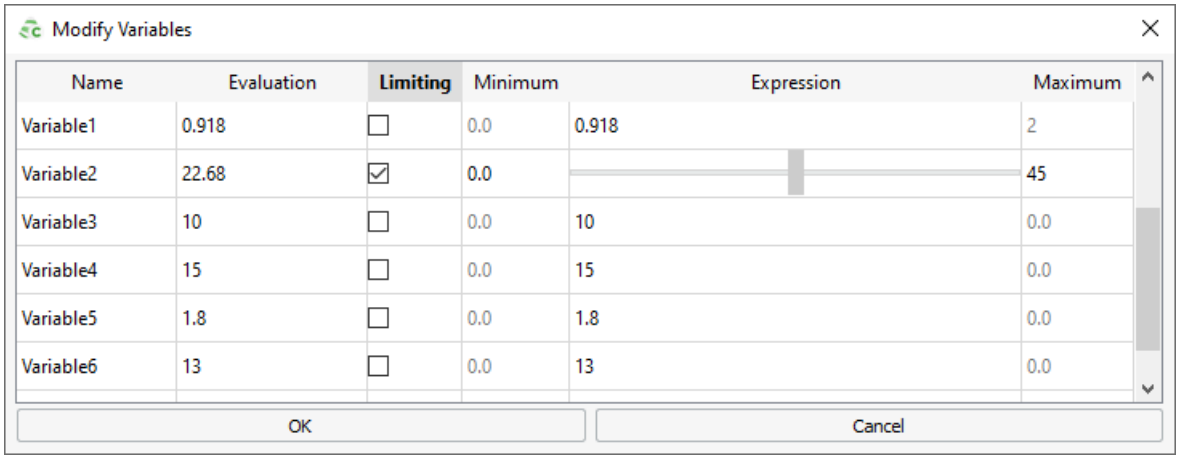


Figure 55: The Modify Variables

2. Click the fields in the **Name** column to modify the name of a variable.
3. Click the fields in the **Expression** column to modify a value of a variable.
4. Select the check box in the **Limiting** column to specify a range for the variable.
   - Enter a value in the **Minimum** and **Maximum** fields.
   - In the **Expression** field, adjust the slider bar to the required value.
   - The result of the **Expression** field is displayed in the **Evaluation** field.
5. Select one of the following workflows to close the dialog.
   - To modify the variables and close the dialog, click **OK**.
   - To modify the variables, but keep the dialog open for further modifications, click **Apply**.

2.7.3 Named Points

Create named points that can be referenced by geometry and requests, similar to variables.

The X coordinate, Y coordinate and Z coordinate of a point can be accessed using a dot followed by the required component.

> For example, *Point1.x* gives access to the X coordinate of named point, *Point1*.

Points can also be constructed using the "pt" command.

> For example, the expression *pt(1,1,1) + pt(2,1,1)* results in a point definition of *pt(3,2,2)*.

The following actions are allowed on points:

- The subtract and add operations are allowed between two points.
- A point may be multiplied or divided by a scalar.
- The distance from the origin to a point is obtained using the *abs* function.

Defining a Named Point to Use in Geometry Creation

Create a named point to create parametric geometry.

1. On the **Construct** tab, in the **Define** group, click the **Add Point** icon.

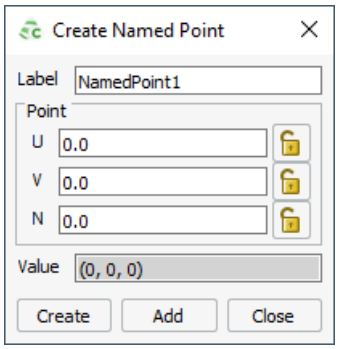


*Figure 56: The **Create named point** dialog.*

2. In the **Label** field, enter a name for the named point.
3. Under **Point**, enter the U coordinate, V coordinate and N coordinate using one of the following workflows:
   - Enter the values manually.
   - Use point-entry to add the coordinates from the 3D view.

   The calculated result is displayed in the **Value** field.

   Note: The result is maintained until the next time the expression is evaluated.
4. Select one of the following workflows to close the dialog.
   - To create the named point and close the dialog, click **Create**.
   - To create the named point, but keep dialog open to create another named point, click **Add**.

2.7.4 Workplanes

Use workplanes to simplify the geometry creation process by creating new geometry on an oblique plane.

When the global coordinates are used to construct primitives in CADFEKO, the orientation of the new entity is fixed. A simple and efficient method is to create a workplane at the intended position and orientation and then create the geometry using the workplane definition.

The following workplanes are predefined by default (and cannot be modified):

- Global XY
- Global XZ
- Global YZ

Anyone of the above workplanes may be set as the default workplane. From its right-click context menu, select **Set as default**.

In addition to the three predefined workplanes, user-defined workplanes may be defined and set as the default workplane.

When a workplane is set as the default workplane, it is indicated by the text **[Default]** in the model tree. The default workplane will be used as the initial workplane for primitive creation, operators and transforms.

Defining a Workplane to Aid in Geometry Creation

Add a new workplane to the model to aid with geometry creation.

1. On the **Construct** tab, in the **Define** group, click the **Add Workplane** icon.

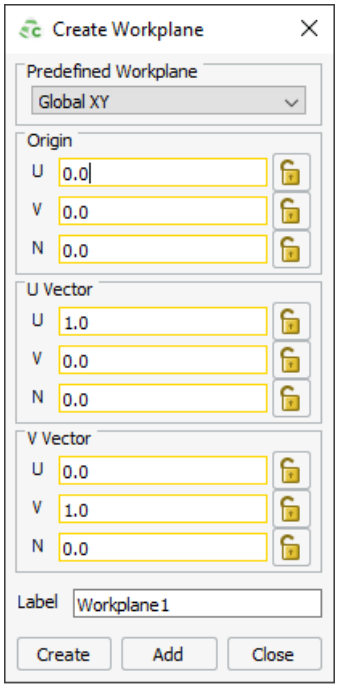


*Figure 57: The **Create Workplane** dialog.*

2. Under **Predefined Workplane** in the drop-down list, select a reference workplane.
3. Under **Origin**, enter the position of the workplane using one of the following methods:
   - Enter the coordinates for the origin manually.
   - Use point entry to enter the coordinates for the origin from the 3D view.
4. Specify the rotation of the workplane by using one of the following methods:
   - Enter values for the U-Vector and V-Vector.
   - Click on any field and from the right-click context menu, click one of the following:
     - Around U
     - Around V

- Around N

and specify the angle of rotation.

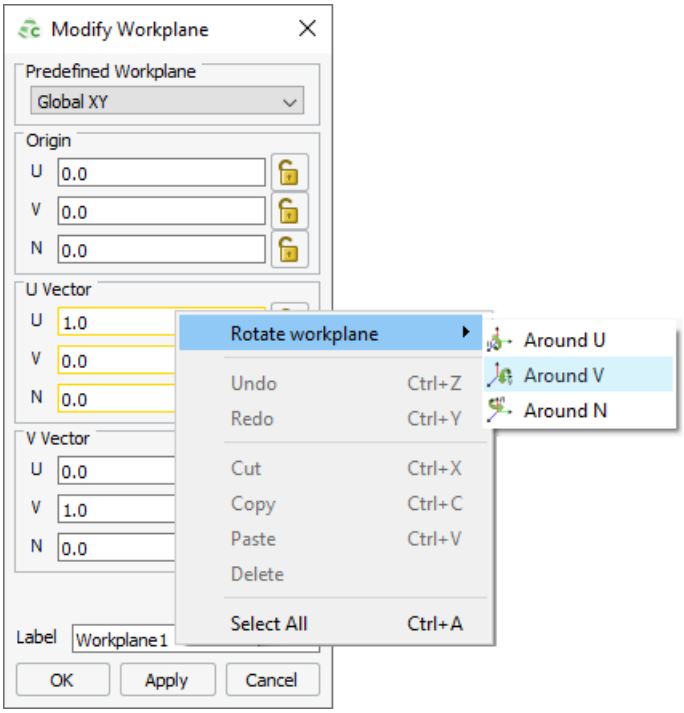


*Figure 58: Rotate the workplane using the **Rotate workplane** right-click context menu.*

5. In the **Label** field, add a unique label for the workplane.
6. Click **OK** to create the workplane and to close the dialog.

Note: Apply transforms on the workplane by selecting the workplane in the model tree. From the right-click context menu, click **Transforms** and select the transform.

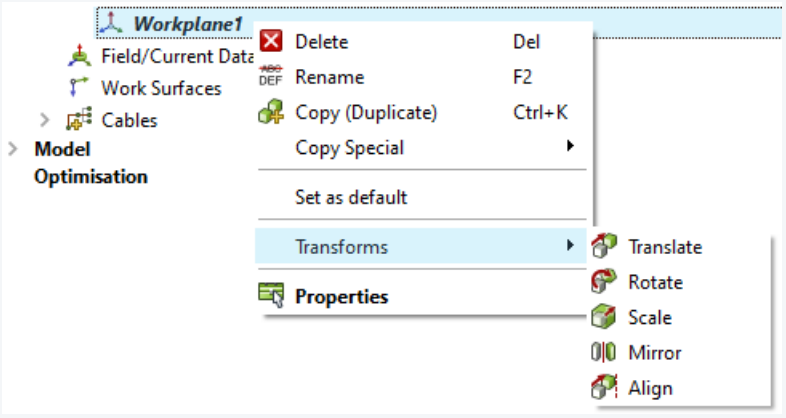


Figure 59: Transforms can be applied to any defined workplanes.

2.8 Constructing Geometry

Create fully parametric and complex CAD geometry using canonical structures and perform Boolean operations on these.

Basic geometry includes solids (cuboid, flare, sphere, cylinder and cone), surfaces (rectangle, polygon, ellipse, paraboloid and NURBS) and arcs (line, polyline, fitted spline, Bézier curve, analytical curve, elliptic arc, parabolic arc, hyperbolic arc and helix). Use Boolean operations such as union, separate, subtract, intersect, split and stitch to create complex geometry. Extend the geometry using spin, loft, sweep and path sweep. Transform the geometry using translate, mirror, rotate, scale, align and project.

2.8.1 Creating Basic Geometry

Create basic geometry using solids, surfaces and arcs and using transforms and Boolean operations.

Creating Arcs

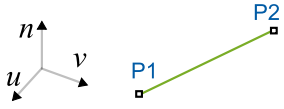
An arc is any smooth curve between two points.

Line

Create a line to be used either as a building block for constructing or modifying geometry or as a wire.

On the **Construct** tab, in the **Create Curve** group, click the **Line** icon.

> **Tip:** Press V,1 to use the shortcut key.

	Start point (P_1)	The starting point of the line.
	End point (P_2)	The end point of the line.

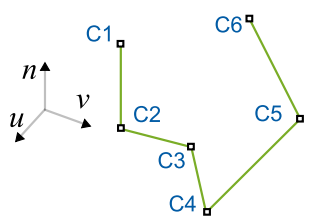
Polyline

Create a polyline to be used either as a building block for constructing geometry or as a wire.

Polylines consist of consecutive straight lines and result in mesh vertices being created at each corner. The lines of a polyline should not cross itself, but if this is required, the polygon can be sub-divided.

On the **Construct** tab, in the **Create Curve** group, click the **Polyline** icon.

Tip: Press V,2 to use the shortcut key.



Corner 1 (C_1)	The first point of the polyline.
Corner 2 (C_2)	The second point of the polyline.
...Corner n (C_n)	Additional points in the polyline. There may be an arbitrary number of points.

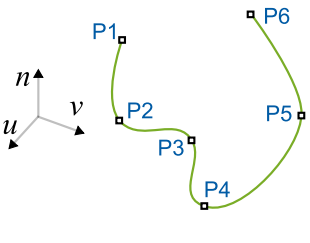
Fitted Spline

Create a fitted spline to be used either as a building block for constructing geometry or as a wire. The fitted spline fits a smooth curve through all the node points in the definition.

Fitted splines are smooth over the entire path (no sharp corners). Fitted splines are preferred over polylines when reconstructing geometry from points (for example, exported from another source) since they do not cause mesh vertices to be created at the node points.

On the **Construct** tab, in the **Create Curve** group, click the **Fitted Spline** icon.

Tip: Press V,3 to use the shortcut key.



Point 1 (P_1)	The starting point of the curve.
Point 2 (P_2)	The second point through which the spline curve will pass.
...Point n (P_n)	The additional points through which the spline must pass. There may be an arbitrary number of points.

Analytical Curve

Create an analytical curve to be used either as a building block for constructing geometry or as a wire. Analytical curves are parametric definitions (in the parameter "t") that define the path in three coordinate systems.

The derivatives of the expressions are required and need to exist over the entire path. If dividing the derivative by zero, the definition is not accepted. An alternative would be to calculate the points in the scripting environment and create a fitted spline.

On the **Construct** tab, in the **Create Curve** group, click the **Analytical Curve** icon.

Tip: Press V,5 to use the shortcut key.

Table 3: Method 1: Cartesian

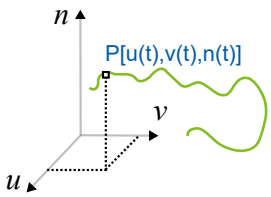
	Parametric interval	Interval over which the analytical curve is parametrically defined.
	Cartesian description	The description of the curve using the Cartesian coordinate system. The U, V and N dimensions as a function of variable t.

Table 4: Method 2: Cylindrical

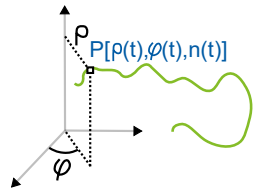
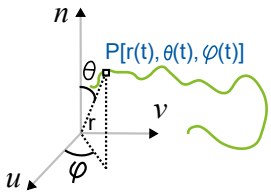
	Parametric interval	Interval over which the analytical curve is parametrically defined.
	Cylindrical description	The cylindrical description of the curve in the ρ, θ and ϕ dimensions as a function of variable t.

Table 5: Method 3: Spherical

	Parametric interval	Interval over which the analytical curve is parametrically defined.
	Spherical description	The spherical description of the curve in the r, θ and ϕ dimensions as a function of variable t.

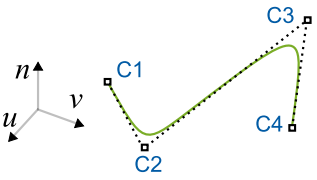
Bézier Curve

Create a Bézier curve to be used either as a building block for constructing geometry or as a wire.

Bézier curves are defined by four points. The curve will start and stop at the first and last point, while the other two points "pull" the curve in their direction, but do not usually pass through them.

On the **Construct** tab, in the **Create Curve** group, click the **Bézier Curve** icon.

Tip: Press V,4 to use the shortcut key.



Corner 1 (C_1)	The starting point of the curve.
Corner 2 (C_2)	The first control point of the Bézier curve (the curve does not necessarily pass through this point).
Corner 3 (C_3)	The second control point of the Bézier curve (the curve does not necessarily pass through this point).
Corner 4 (C_4)	The end point of the curve.

Parabolic Arc

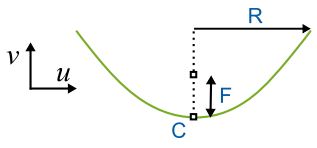
Create a parabolic arc to be used either a building block for constructing geometry or as free-standing wires.

Parabolic arcs are often used in conjunction with the spin operator to create parabolic dishes for reflector antennas.

On the **Construct** tab, in the **Create Arc** group, click the **Parabolic Arc** icon.

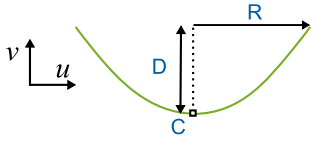
Tip: Press A,2 to use the shortcut key.

Method 1: Base centre, focal depth, radius



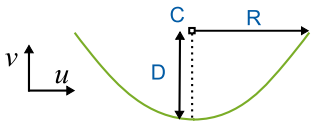
Base centre (C)	The centre of the parabola on which the arc lies.
Focal depth (F)	The focal depth of the parabola.
Radius (R)	The radius of the aperture of the parabolic arc.

Method 2: Base centre, radius, depth



Base centre (C)	The centre of the parabola on which the arc lies.
Radius (R)	The radius of the aperture of the parabolic arc.
Depth (D)	The distance from the apex of the parabola to the centre of the aperture.

Method 3: Aperture centre, radius, depth



Aperture centre (C)	The aperture centre of the parabolic arc section.
Radius (R)	The radius of the aperture of the parabolic arc.
Depth (D)	The distance from the apex of the parabola to the centre of the aperture.

Hyperbolic Arc

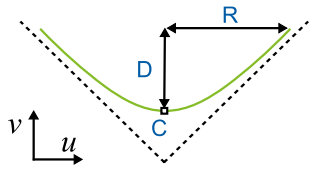
Create a hyperbolic arc to be used either as a building block for constructing geometry or as a wire.

Hyperbolic arcs are often used in conjunction with the spin operator to create hyperbolic dishes for reflector antennas.

On the **Construct** tab, in the **Create Arc** group, click the **Hyperbolic Arc** icon.

> **Tip:** Press A,3 to use the shortcut key.

Method 1: Base centre, depth, radius, eccentricity



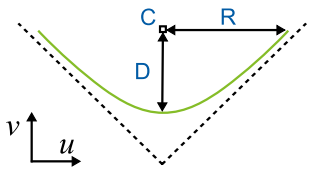
Base centre (C)	The centre of the hyperbola on which the arc lies.
Depth (D)	The distance from the apex of the hyperbola to the centre of the arc aperture.
Radius (R)	The radius of the aperture of the hyperbolic arc.
Eccentricity	The eccentricity of the hyperbola on which the hyperbolic arc section lies.

Conditions to create a valid hyperbolic arc:

$$1 < \varepsilon \leq \sqrt{1+\frac{R^2}{D^2}} \tag{1}$$

where D is denoted by the depth, R by the aperture radius R and ε the eccentricity.

Method 2: Aperture centre, depth, radius, eccentricity



Aperture centre (C)	The centre of the aperture formed by the hyperbolic arc.
Depth (D)	The distance from the apex of the hyperbola to the centre of the arc aperture.
Radius (R)	The radius of the aperture of the hyperbolic arc.
Eccentricity	The eccentricity of the hyperbola on which the hyperbolic arc section lies. The eccentricity must be greater than one to specify a valid hyperbola. **Note:** Not all values greater than one specifies a valid hyperbola.

Conditions to create a valid hyperbolic arc:

$$1<\varepsilon\leq\sqrt{1+\frac{R^2}{D^2}} \tag{2}$$

where D is denoted by the depth, R by the aperture radius R and ε the eccentricity.

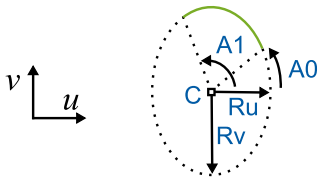
Elliptic Arc

Create an elliptic arc to be used either as a building block for constructing geometry or as a wire.

On the **Construct** tab, in the **Create Arc** group, click the **Elliptic Arc** icon.

Tip: Press A,1 to use the shortcut key.

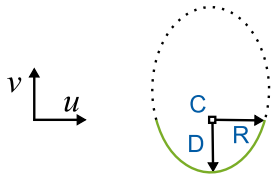
Method 1: Centre point, radii, start angle, end angle



Centre point (C)	The centre of the ellipse on which the arc lies.
Radius (RU)	The radius (half of the axis length) in the U axis direction of the ellipse on which the arc lies.
Radius (RV)	The radius (half of the axis length) in the V axis direction of the ellipse on which the arc lies.
Start angle (A0)	The angle, from the positive U axis direction where the arc begins.

	End angle (A1)	The angle, from the positive U axis direction where the arc ends

Method 2: V major axis direction - Aperture centre, depth, aperture radius, eccentricity



Aperture centre (C)	The centre of the aperture formed by the elliptical arc section.
Depth (D)	The distance from the aperture centre point to the apex of the elliptical arc section.
Aperture radius (R)	The radius of the aperture of the elliptic arc.
Eccentricity	The eccentricity of the ellipse on which the elliptical arc section lies. **Note:** The eccentricity must be less than 1 to specify a valid ellipse.

Conditions for creating a valid elliptic arc:

$$\text{if} \quad R \le D : \sqrt{1-\frac{R^2}{D^2}} \le \varepsilon < 1 \tag{3}$$

where D is denoted by the depth, R by the aperture radius and ε the eccentricity.

Method 3: U major axis direction - Aperture centre, depth, aperture radius, eccentricity

Figure 60: Method 3

Aperture centre (C)	The centre of the aperture formed by the elliptical arc section.
Depth (D)	The distance from the aperture centre point to the apex of the elliptical arc section.
Aperture radius (R)	The radius of the aperture of the elliptic arc.
Eccentricity	The eccentricity of the ellipse on which the elliptical arc section lies.

		Note: The eccentricity must be less than 1 to specify a valid ellipse.

Conditions for creating a valid elliptic arc:

$$\text{if} \quad R \geq D : 0 \leq \varepsilon \leq \sqrt{1 - \frac{D^2}{R^2}} \tag{4}$$

where D is denoted by the depth, R by the aperture radius and ε the eccentricity.

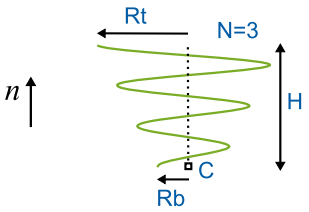
Helix

Create a helix to be used either as a building block for constructing geometry or as a wire.

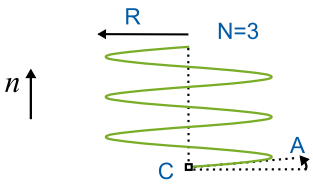
On the **Construct** tab, in the **Create Arc** group, click the **Helix** icon.

Tip: Press A,4 to use the shortcut key.

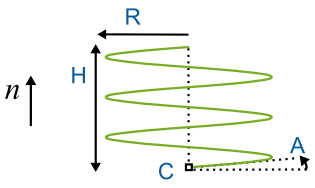
Method 1: Base centre, base radius, end radius, height, turns

	Origin (C)	The centre point of the helix base.
	Base radius (R_b)	The radius of the helix base (parallel to the UV plane).
	End radius (R_t)	The height of the helix, in the N axis direction.
	Turns (N)	The number of turns of the helix (the rotation direction is selected based on the **Left handed** check box).

Method 2: Base centre, radius, pitch angle, turns

	Origin (C)	The centre point of the helix base.
	Radius (R)	The radius of the helix (parallel to the UV plane).
	Pitch angle (A)	The angle formed between the tangent of the curve and the UV plane - constant along the length of the helix.
	Turns (N)	The number of turns of the helix (the rotation direction is selected based on the **Left handed** check box).

Method 3: Base centre, radius, height, pitch angle



Origin (C)	The centre point of the helix base.
Radius (R)	The radius of the helix (parallel to the UV plane).
Height (H)	The height of the helix, in the N axis direction.
Pitch angle (A)	The angle formed between the tangent of the curve and the UV plane - constant along the length of the helix.

Creating Surfaces

A surface can be defined and used to create more complex structures.

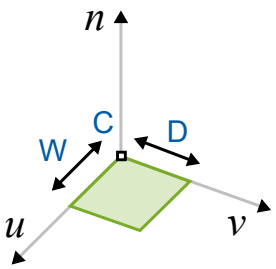
Rectangle

Create a rectangle or a square.

On the **Construct** tab, in the **Create Surface** group, click the **Rectangle** icon.

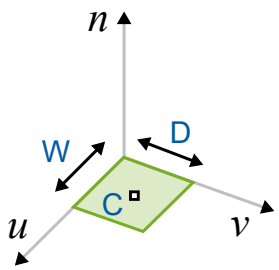
Tip: Press S,1 to use the shortcut key.

Method 1: Base corner, width, depth



Base corner (C)	A corner of the rectangle.
Width (W)	The width of the rectangle.
Depth (D)	The depth of the rectangle.

Method 2: Base centre, width, depth

	Base centre (C)	The centre of the rectangle.
	Width (W)	The width of the rectangle.
	Depth (D)	The depth of the rectangle.

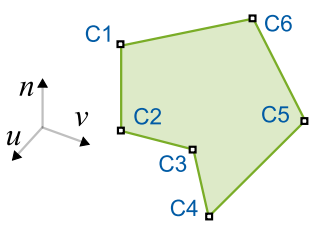
Polygon

Create a two-dimensional polygonal surface.

The corner points of the polygon are required to be on the same plane. If this is not the case, the surface must be constructed using smaller polygons that do lie in a common plane.

On the **Construct** tab, in the **Create Surface** group, click the **Polygon** icon.

> **Tip:** Press S,2 to use the shortcut key.

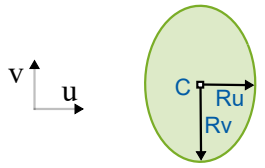
	Corner 1 (C)	The first corner of the polygon.
	Corner 2 (C_2)	The second corner of the polygon.
	...Corner n (C_n)	Additional corners of the polygon (an arbitrary number). All must be in the same plane. The polygon is closed by connecting the last point to C_1.

Ellipse

Create an ellipse or a circle.

On the **Construct** tab, in the **Create Surface** group, click the **Ellipse** icon.

> **Tip:** Press S,3 to use the shortcut key.

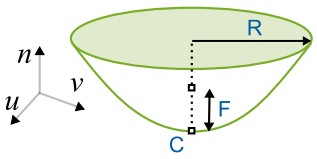
	Centre point (C)	The centre of the ellipse.
	Radius (R_u)	The radius (half of the axis length) in the U axis direction.
	Radius (R_v)	The radius (half of the axis length) in the V axis direction.

Paraboloid

Create a paraboloid.

On the **Construct** tab, in the **Create Surface** group, click the **Paraboloid** icon.

> **Tip:** Press S,4 to use the shortcut key.

	Centre point (C)	The apex of the paraboloid.
	Radius (R)	The radius of the paraboloid aperture, parallel to the UV plane.
	Focal depth (F)	The focal depth (F) of the paraboloid is the distance from the centre point (C) to the focal point. If this is negative, the paraboloid is oriented towards the negative N axis.

The focal depth is related to the dimensions of paraboloid by

$$f = \frac{R^2}{4h} \tag{5}$$

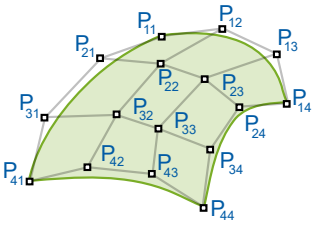
where (H) denotes the distance from the centre point (C) to the aperture centre of the paraboloid.

NURBS

Create a non-uniform rational basis spline (NURBS) surface.

On the **Construct** tab, in the **Create Surface** group, click the **NURBS Surface** icon.

> **Tip:** Press S,5 to use the shortcut key.

	Specify the order of the Bézier curves	The degree of the Bézier curve in the U′ direction and V′ direction.
	Point position	The position of each surface control point is specified.
	Weight	The weight of each surface control points is specified.

Creating Solids

Solid geometries are closed surfaces and have an enclosed region.

A solid primitive is by default a perfect electric conductor (PEC). The solid can be changed to a dielectric or a shell structure by setting the region properties.

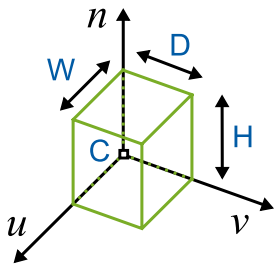
Cuboid

Create a cuboid.

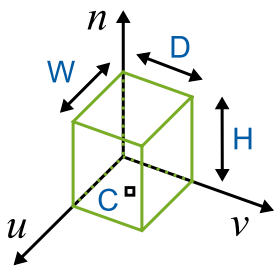
On the **Construct** tab, in the **Create Solid** group, click the **Cuboid** icon.

> **Tip:** Press C,1 to use the shortcut key.

Method 1: Base corner, width, depth, height

	Base corner (C)	One corner of the cuboid.
	Width (W)	The cuboid dimension in the U axis direction.
	Depth (D)	The cuboid dimension in the V axis direction.
	Height (H)	The cuboid dimension in the N axis direction.

Method 2: Base centre, width, depth, height



Base centre (C)	The base centre of the cuboid.
Width (W)	The cuboid dimension in the U axis direction.
Depth (D)	The cuboid dimension in the V axis direction.
Height (H)	The cuboid dimension in the N axis direction.

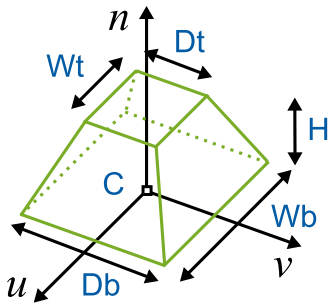
Flare

Create a flare or a pyramid. Flares are often used in the construction of horn antennas and waveguide transitions.

On the **Construct** tab, in the **Create Solid** group, click the **Flare** icon.

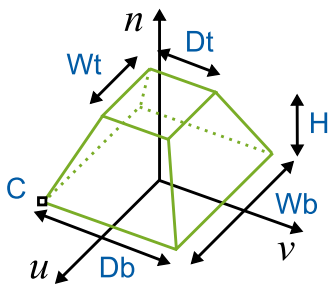
> **Tip:** Press C,2 to use the shortcut key.

Method 1: Base centre, bottom width, bottom depth, height, top width, top depth



Base centre (C)	The centre of the flare base
Bottom width (W_b)	The width of the base in the U axis direction.
Bottom depth (D_b)	The depth of the base in the V axis direction.
Height (H)	The height of the flare, in the N axis direction.
Top width (W_t)	The width of the top in the U axis direction.
Top depth (D_t)	The depth of the top in the V axis direction.

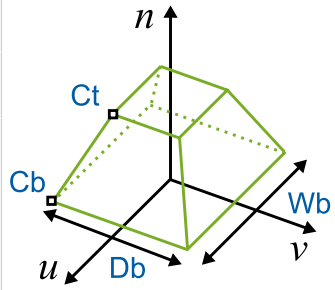
Method 2: Base corner, bottom width, bottom depth, height, top width, top depth



Base corner (C)	The corner of the flare base.
Bottom width (W_b)	The width of the base in the U axis direction.
Bottom depth (D_b)	The depth of the base in the V axis direction.
Height (H)	The height of the flare, in the N axis direction.
Top width (W_t)	The width of the top in the U axis direction.

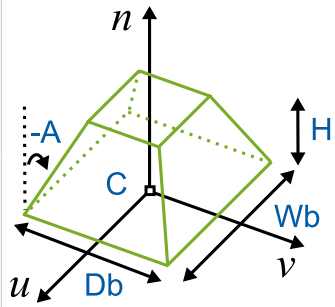
	Top depth (D_t)	The depth of the top in the V axis direction.

Method 3: Base corner, top corner, bottom width, bottom depth



Base corner (C)	The corner of the flare base.
Top corner (C_t)	The corner of the flare top.
Bottom width (W_b)	The width of the base in the U axis direction.
Bottom depth (D_b)	The depth of the base in the V axis direction.

Method 4: Base centre, width, depth, height, flare angle 1, flare angle 2



Base centre (C_b)	The centre of the flare base.
Bottom width (W_b)	The width of the base in the U axis direction.
Bottom depth (D_b)	The depth of the base in the V axis direction.
Height (H)	The height of the flare in the N axis direction.
Flare angle (AU)	The angle of the flare from the UN plane.
Flare angle (AV)	The angle of the flare from the VN plane.

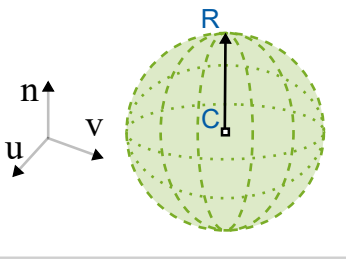
Sphere

Create a sphere or a spheroid (radius varies).

On the **Construct** tab, in the **Create Solid** group, click the **Sphere** icon.

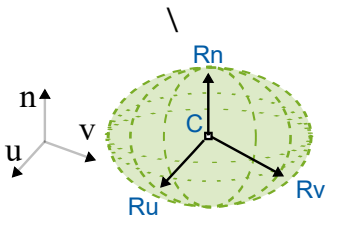
> **Tip:** Press C,3 to use the shortcut key.

Method 1: Centre, radius



Centre (C)	The centre point of the sphere.
Radius (R):	The radius of the sphere.

Method 2: Centre, radius U, radius V, radius N



Centre (C)	The centre point of the sphere.
Radius (R_u)	The radius of the ellipsoid in the U dimension.
Radius (R_v)	The radius of the ellipsoid in the V dimension.
Radius (R_n)	The radius of the ellipsoid in the N dimension.

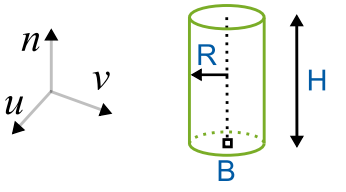
Cylinder

Create a cylinder.

On the **Construct** tab, in the **Create Solid** group, click the **Cylinder** icon.

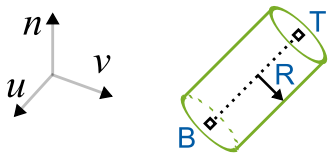
> **Tip:** Press C,4 to use the shortcut key.

Method 1: Base centre, radius, height



Base centre (B)	The centre of the cylinder base.
Radius (R)	Cylinder radius (parallel to the UV plane).
Height (H)	Cylinder height in the N direction measured from *B*.

Method 2: Base centre, top centre, radius



Base centre (B)	The centre of the cylinder base.
Top centre (T)	The centre of the cylinder top.
Radius (R)	The cylinder radius (perpendicular to the line from *B* to *T*).

Related tasks
Creating a UTD Cylinder

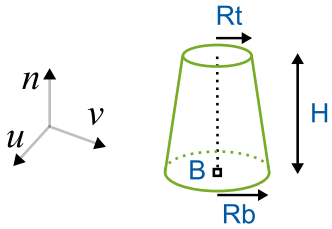
Cone

Create a cone.

On the **Construct** tab, in the **Create Solid** group, click the **Cone** icon.

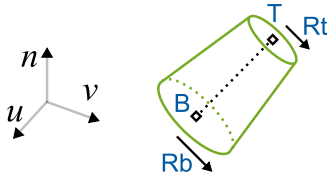
> **Tip:** Press C,5 to use the shortcut key.

Method 1: Base centre, base radius, height, top radius



Base centre (B)	The centre of the base of the cone.
Base radius (R_b)	The radius of the cone base (parallel to the UV plane).
Height (H)	The cone height in the N direction measured from *B*.
Top radius (R_t)	The radius of the cone top (parallel to the UV plane).

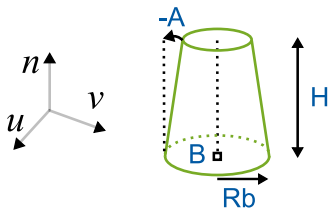
Method 2: Base centre, top centre, base radius, top radius



Base centre (B)	The centre of the base of the cone.
Top centre (T)	The centre of the top of the cone.
Base radius (R_b)	The radius of the cone base (perpendicular to the line from *B* to *T*).

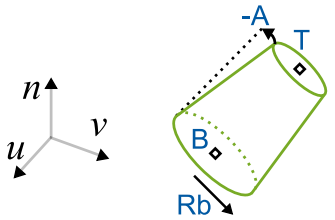
	Top radius (R_t)	The radius of the cone top (parallel to the UV plane).

Method 3: Base centre, base radius, height, cone angle



Base centre (B)	The centre of the base of the cone.
Base radius (R_b)	The radius of the cone base (parallel to the UV plane).
Height (H)	The cone height in the N direction.
Flare angle (A)	The growth angle measured from the N axis.

Method 4: Base centre, top centre, base radius, cone angle



Base centre (B)	The centre of the base of the cone.
Top centre (T)	The centre of the top of the cone.
Base radius (R_b)	The radius of the cone base (perpendicular to the line from *B* to *T*).
Flare angle (A)	The growth angle measured from the line defined between *B* to *T*.

2.8.2 Constructing Complex Surfaces

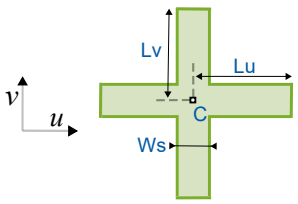
Create fully parametric and complex surfaces.

Complex surfaces include crosses (cross, strip cross, spiral cross and T-cross), rings (ring, open ring and split ring), hexagons (hexagon and strip hexagon) and a trifilar. As with basic geometry, use Boolean, extend and transform operations to create even more complex geometry.

Cross

Create a cross to be used either as a primitive or as a building block for constructing complex geometry.

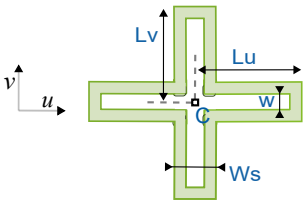
On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. From the **Crosses** drop-down list, select **Cross**.

	Centre point (C)	The centre point of the cross.
	Arm length (Lu)	The arm length in the U axis direction.
	Arm length (Lv)	The arm length in the V axis direction.
	Strip width (Ws)	The width of the strip.

Strip Cross

Create a strip cross to be used either as a primitive or as a building block for constructing complex geometry.

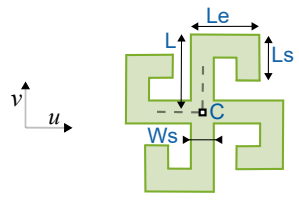
On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. From the **Crosses** drop-down list, select **Strip Cross**.

	Centre point (C)	The centre point of the cross.
	Arm length (Lu)	The arm length in the U axis direction.
	Arm length (Lv)	The arm length in the V axis direction.
	Strip width (Ws)	The width of the strip.
	Slot width (w)	The width of the slot.

Spiral Cross

Create a spiral cross to be used either as a primitive or as a building block for constructing complex geometry.

On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. From the **Crosses** drop-down list, select **Spiral Cross**.

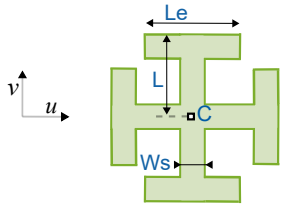
	Centre point (C)	The centre point of the cross.
	Arm length (L)	The arm length (all directions equal)
	Edge length (Le)	The edge length (all equal).
	Spiral length (Ls)	The length of the spiral at the end of each arm.

	Strip width (Ws)	The width of the strip.

T-Cross

Create a T-cross to be used either as a primitive or as a building block for constructing complex geometry.

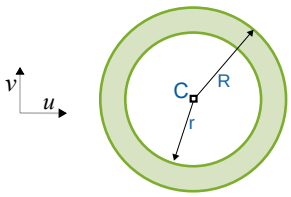
On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. From the **Crosses** drop-down list, select **T-Cross**.

	Centre point (C)	The centre point of the cross.
	Arm length (L)	The arm length (all directions)
	Edge length (Le)	The edge length (all arms).
	Strip width (Ws)	The width of the strip.

Ring

Create a ring to be used either as a primitive or as a building block for constructing complex geometry.

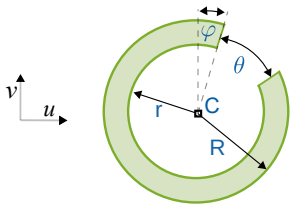
On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. From the **Rings** drop-down list, select **Ring**.

	Centre point (C)	The centre point of the ring.
	Outer radius (R)	The outer radius of the ring.
	Inner radius (r)	The inner radius of the ring.

Open Ring

Create an open ring to be used either as a primitive or as a building block for constructing complex geometry.

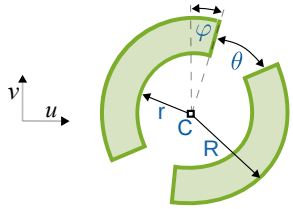
On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. From the **Rings** drop-down list, select **Open Ring**.

	Centre point (C)	The centre point of the ring.
	Outer radius (R)	The outer radius of the ring.
	Inner radius (r)	The inner radius of the ring.
	Start angle (φ)	The start angle of the ring.
	Gap angle (θ)	The gap angle of the ring.

Split Ring

Create a split ring to be used either as a primitive or as a building block for constructing complex geometry.

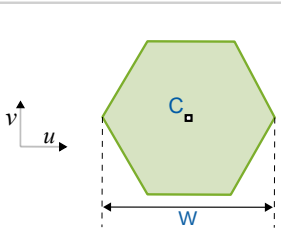
On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. From the **Rings** drop-down list, select **Split Ring**.

	Centre point (C)	The centre point of the ring.
	Outer radius (R)	The outer radius of the ring.
	Inner radius (r)	The inner radius of the ring.
	Start angle (φ)	The start angle of the ring.
	Gap angle (θ)	The gap angle of the ring.

Hexagon

Create a hexagon to be used either as a primitive or as a building block for constructing complex geometry.

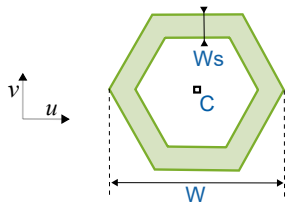
On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. From the **Hexagons** drop-down list, select **Hexagon**.

	Centre point (C)	The centre point of the hexagon.
	Width (W)	The width of the hexagon.

Strip Hexagon

Create a strip hexagon to be used either as a primitive or as a building block for constructing complex geometry.

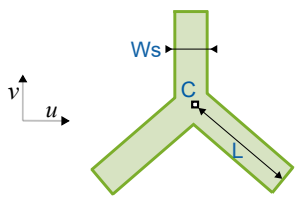
On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. From the **Hexagons** drop-down list, select **Strip Hexagon**.

	Centre point (C)	The centre point of the strip hexagon.
	Width (W)	The width of the hexagon.
	Srip width (Ws)	The width of the strip.

Trifilar

Create a trifilar to be used either as a primitive or as a building block for constructing complex geometry.

On the **Construct** tab, in the **Create Surface** group, click the **Complex Surfaces** icon. Select **Trifilar**.

	Centre point (C)	The centre point of the trifilar.
	Length (L)	The arm length.
	Srip width (Ws)	The width of the strip.

2.8.3 Constructing Periodic Structures

Create periodic structures that include shapes in the form of crosses, rings, hexagons, a trifilar, a plane and an ellipse. Use the shapes to construct a unit cell for solving with periodic boundary conditions.

Workflow for Constructing Periodic Structures

View the workflow for constructing periodic structures using a unit cell.

1. Define the shape(s) that will be used in the unit cell construction.
2. Define the unit cell and its layers.
3. Auto-generate the geometry from the unit cell definition using the **Build Geometry** tool. The equivalent geometry is then available in the model tree.
4. If periodic boundary conditions was not selected on the **Build Geometry** dialog, define the periodic boundary conditions for the unit cell.
5. Add a transmission/reflection request.

Note: See Example C5 in the Feko Example Guide.

Related tasks

Creating Shapes

Defining a Unit Cell

Auto-Generating Geometry from Unit Cell Definition

Defining a Periodic Boundary Condition (PBC)

Requesting Transmission / Reflection Coefficients

Accessing the Periodic Structures Tab on the Ribbon

Open the **Periodic Structures** tab on the ribbon to access advanced tools related to defining periodic structures.

By default, the **Periodic Structures** tab is not displayed on the ribbon. To access the **Periodic Structures** tab, you must configure the ribbon to show the tab.

On the **Home** tab, in the **Extensions** group, click the **Periodic Structures** icon.

When the **Periodic Structures** tab is enabled, it is located on the ribbon between the **Transform** tab and **Source/Load** tab.

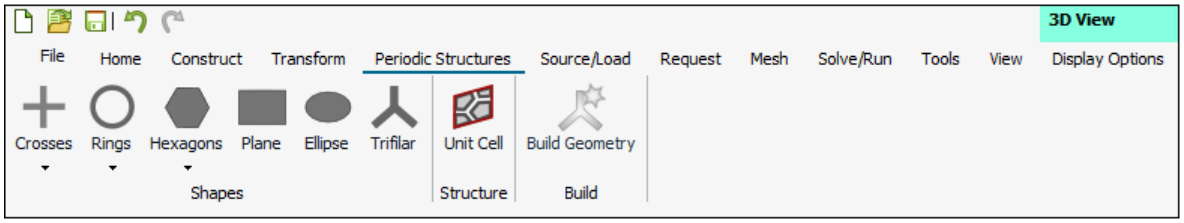


*Figure 61: The ribbon in CADFEKO (**Periodic Structures** tab)*

Creating Shapes

Create shapes to construct a unit cell for solving with periodic boundary conditions and to obtain transmission and reflection coefficients.

Shapes include crosses (cross, strip cross, spiral cross and T-cross), rings (ring, open ring and split ring), hexagons (hexagon and strip hexagon), a trifilar, a plane and an ellipse.

After a shape is defined, it available in the model tree (**Construction** tab) under **Definitions** > **Periodic Structures** > **Shapes**.

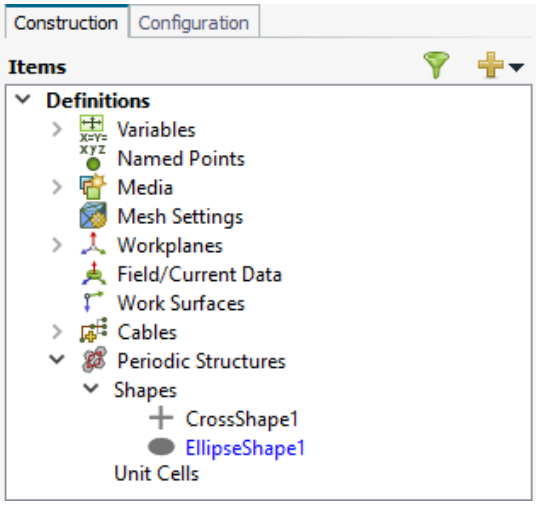


Figure 62: Defined shapes in the model tree.

Related concepts

Workflow for Constructing Periodic Structures

Related tasks

Accessing the Periodic Structures Tab on the Ribbon

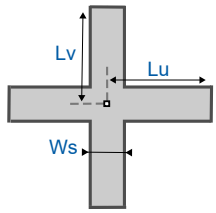
Defining a Unit Cell

Auto-Generating Geometry from Unit Cell Definition

Cross

Create a cross shape to be used in the construction of a unit cell.

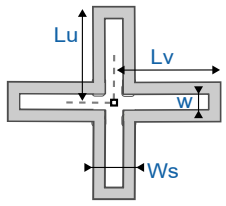
On the **Periodic Structures** tab, in the **Shapes** group, from the **Crosses** drop-down list, select **Cross**.

	Arm length (Lu)	The arm length in the U axis direction.
	Arm length (Lv)	The arm length in the V axis direction.
	Strip width (Ws)	The width of the strip.

Strip Cross

Create a strip cross shape to be used in the construction of a unit cell.

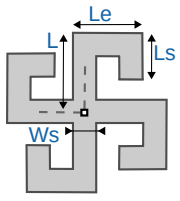
On the **Periodic Structures** tab, in the **Shapes** group, from the **Crosses** drop-down list, select **Strip Cross**.

	Arm length (Lu)	The arm length in the U axis direction.
	Arm length (Lv)	The arm length in the V axis direction.
	Strip width (Ws)	The width of the strip.
	Slot wdith (W)	The width of the slot.

Spiral Cross

Create a spiral cross shape to be used in the construction of a unit cell.

On the **Periodic Structures** tab, in the **Shapes** group, from the **Crosses** drop-down list, select **Spiral Cross**.

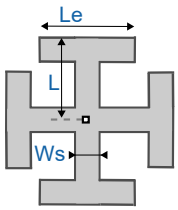
	Arm length (L)	The arm length from the origin (all directions).
	Edge length (Le)	The arm length of the first extension.
	Edge length (Ls)	The arm length of the last extension.

	Strip width (Ws)	The width of the strip.

T-Cross

Create a T-cross shape to be used in the construction of a unit cell.

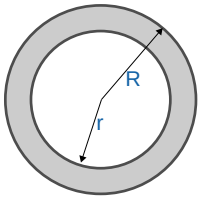
On the **Periodic Structures** tab, in the **Shapes** group, from the **Crosses** drop-down list, select **T-Cross**.

	Arm length (L)	The arm length (all directions).
	Edge length (Le)	The length of the T-section.
	Strip width (Ws)	The width of the strip.

Ring

Create a ring shape to be used in the construction of a unit cell.

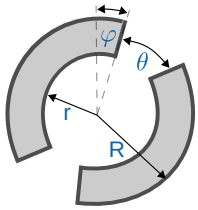
On the **Periodic Structures** tab, in the **Shapes** group, from the **Rings** drop-down list, select **Ring**.

	Outer radius (R)	The outer radius of the ring.
	Inner radius (r)	The inner radius of the ring.

Split Ring

Create a split ring shape to be used in the construction of a unit cell.

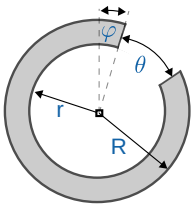
On the **Periodic Structures** tab, in the **Shapes** group, from the **Rings** drop-down list, select **Split Ring**.

	Outer radius (R)	The outer radius of the ring.
	Inner radius (r)	The inner radius of the ring.
	Start angle (φ)	The start angle of the ring.
	Gap angle (θ)	The gap angle of the ring.

Open Ring

Create an open ring shape to be used in the construction of a unit cell.

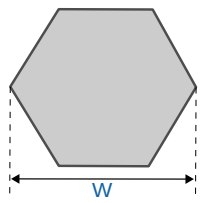
On the **Periodic Structures** tab, in the **Shapes** group, from the **Rings** drop-down list, select **Open Ring**.

	Outer radius (R)	The outer radius of the ring.
	Inner radius (r)	The inner radius of the ring.
	Start angle (φ)	The start angle of the ring.
	Gap angle (θ)	The gap angle of the ring.

Hexagon

Create a hexagon shape to be used in the construction of a unit cell.

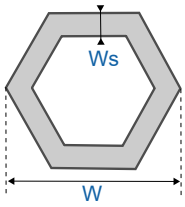
On the **Periodic Structures** tab, in the **Shapes** group, from the **Hexagons** drop-down list, select **Hexagon**.

	Width (W)	The width of the hexagon.

Strip Hexagon

Create a strip hexagon shape to be used in the construction of a unit cell.

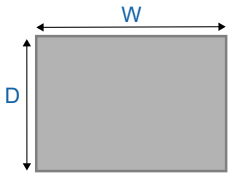
On the **Periodic Structures** tab, in the **Shapes** group, from the **Hexagons** drop-down list, select **Strip Hexagon**.

	Width (W)	The width of the hexagon.
	Srip width (Ws)	The width of the strip.

Plane

Create a plane shape to be used in the construction of a unit cell.

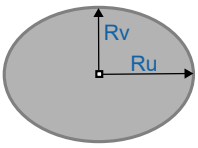
On the **Periodic Structures** tab, in the **Shapes** group, select **Plane**.

	Width (W)	The width of the plane.
	Depth (D)	The depth of the plane

Ellipse

Create an ellipse shape to be used in the construction of a unit cell.

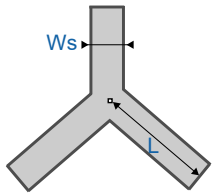
On the **Periodic Structures** tab, in the **Shapes** group, select **Ellipse**.

	Radius (Ru)	The radius in the U axis direction.
	Radius (Rv)	The radius in the V axis direction.

Trifilar

Create a trifilar shape to be used in the construction of a unit cell.

On the **Periodic Structures** tab, in the **Shapes** group, select **Trifilar**.

	Length (L)	The arm length.
	Srip width (Ws)	The width of the strip.

Defining a Unit Cell

Define a finite layered structure comprising layers of substrate and metal to solve with periodic boundary conditions. Alternately, imprint the structure onto a surface to construct frequency selective surfaces (FSS).

Some applications for the unit cell are as follows:

- Use as periodic structure for solving transmission and reflection coefficients.
- Auto-generate the geometry from a unit cell definition and attach the geometry to a flat or curved surface an use as a frequency selective surface (FSS).

1. On the **Periodic Structures** tab, in the **Structure** group, select **Unit Cell**.

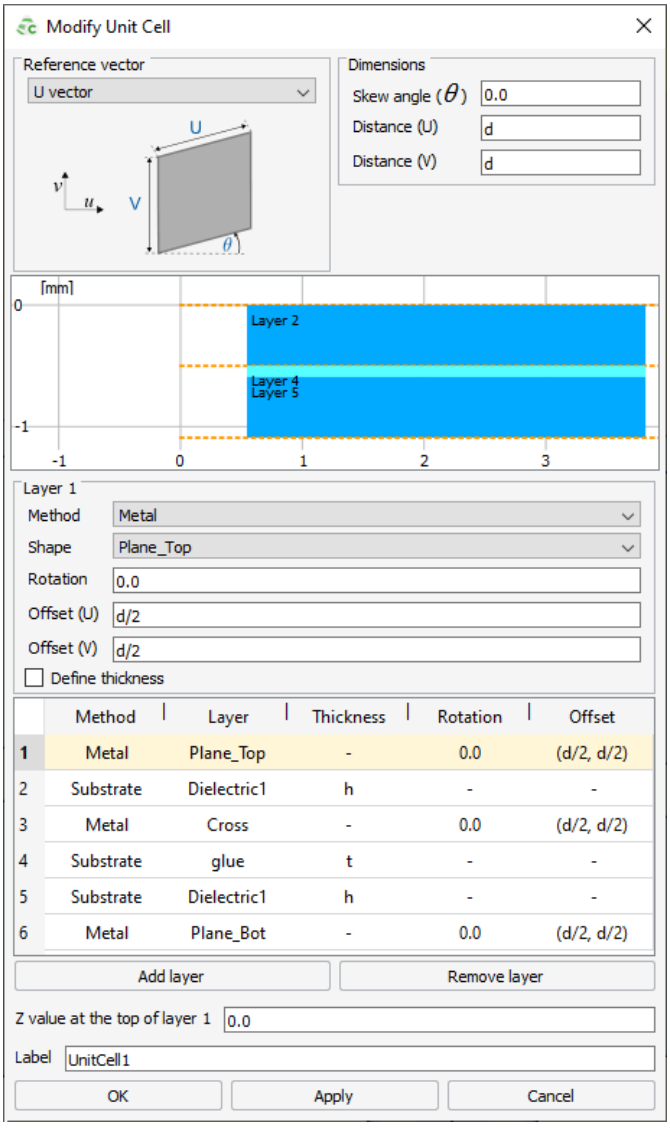


Figure 63: An example of defining a unit cell.

2. From the **Reference Vector** drop-down list, select one of the following:
   - To specify the skew angle relative to the U axis, select **U vector**.

- To specify the skew angle relative to the V axis, select **V vector**.

3. Under **Dimensions**, specify dimensions and orientation of the unit cell:
   a) In the **Skew angle** (θ) field, specify the unit cell skew angle in degrees.

      Tip: A skew angle of 0° yields a rectangular unit cell.

   b) In the **Distance (U)** field, specify the length of the unit cell in the U axis direction.
   c) In the **Distance (V)** field, specify the length of the unit cell in the V axis direction.
4. For each layer, from the **Method** drop-down list select one of the following:
   - To specify a dielectric substrate or free space layer, select **Substrate**.
   - To specify a metal layer, select **Metal**.
   - To specify a metallic layer minus the shape, select **Aperture**.

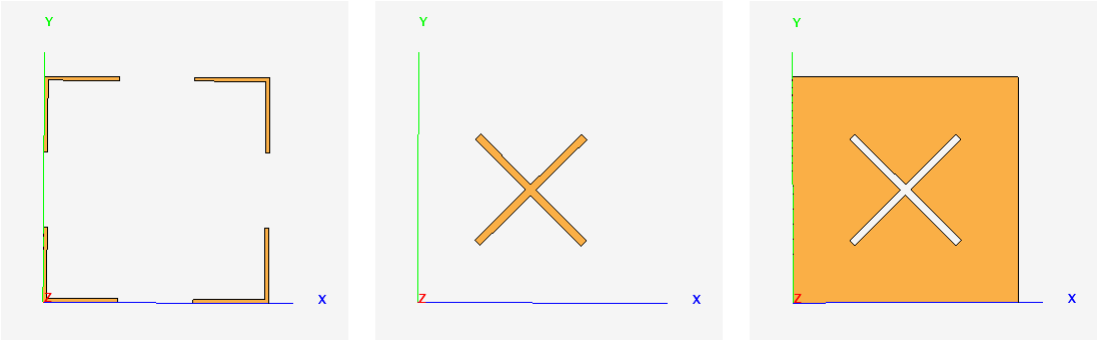


Figure 64: An example of a single layer with a Cross shape. Left: Metal cross with no rotation or U and V ofsets; Centre: Metal cross with 45° rotation and U and V offsets of 1.5; Right: Aperture cross with 45° rotation and U and V offsets of 1.5.

   a) Specify the relevant options for each layer:
      - For **Substrate**:
        - In the **Medium** field, specify the dielectric medium or free space for vacuum.
        - In the **Thickness** field, specify the thickness of the layer.
      - For **Metal** and **Aperture**:
        - In the **Shape** field, specify the shape to be used, for example: cross, ellipse, trifilar.
        - In the **Rotation** field, specify the rotation of the shape in degrees.
        - In the **Offset (U)** field, specify the offset in the U direction from the corners of the unit cell.
        - In the **Offset (V)** field, specify the offset in the V direction from the corners of the unit cell.
        - [Optional] To specify a thickness, select the **Define thickness** check box and specify the metal thickness.
5. In the **Z-value at the top of layer 1** field, specify where the top of layer 1 is located.
6. Click **OK** to define the unit cell and to close the dialog.

After a unit cell is defined, it available in the model tree (**Construction** tab) under **Definitions** > **Periodic Structures** > **Unit Cells**.

> **Note:** Next step is to auto-generate the geometry from the unit cell definition.

Related concepts
Workflow for Constructing Periodic Structures
Related tasks
Accessing the Periodic Structures Tab on the Ribbon
Creating Shapes
Auto-Generating Geometry from Unit Cell Definition

Auto-Generating Geometry from Unit Cell Definition

After a unit cell was defined, the equivalent geometry can be auto-generated and the periodic boundary condition applied based on the unit cell dimensions (optional).

1. On the **Periodic Structures** tab, in the **Build** group, select **Build Geometry**.
2. [Optional] On the **Build Geometry** dialog, select the **Set Periodic Boundary Condition (PBC)** check box to apply a periodic boundary condition based on the unit cell dimensions.

After the **Build Geometry** tool was run, the equivalent geometry is created and available in the model tree.

Related concepts
Workflow for Constructing Periodic Structures
Related tasks
Accessing the Periodic Structures Tab on the Ribbon

2.8.4 Creating Complex Geometry Using Boolean Operations

Boolean operations include union, separate, subtract from, intersection, split and stitch. These operators allow parts to be combined.

Union

Combine multiple geometry parts into a single part and to ensure mesh connectivity once the geometry is meshed.

> **Note:** Geometry parts that touch or intersect, but are not unioned, will not be physically connected in the simulation mesh (except for FDTD where unioning is not required.)

Unconnected geometry results in invalid meshes (overlapping or misaligned) that could generate errors during the Feko solution. Unioning geometry ensures that faces occupying the same space are merged into a single face where solution settings can be applied.

In some cases it could happen that a union of geometry parts fails. First use the *Simplify* (repair operation) tool on the primitive parts before attempting the union again.

> **Note:** In some cases it could appear that a union of faces from imported geometry form closed regions, but do not result in new regions (details tree).
>
> The stitch tool would be better suited in this case.

Related concepts
Stitch
Simplify

Combining Geometry Using Union

Apply the union operation to obtain a single, physically connected part.

1. Select the geometry parts that you want to union.
2. On the **Construct** tab, in the **Modify** group, click the **Union** icon.

The geometry parts are physically connected and will create an electrically connected mesh once the geometry is meshed.

Editing a Part in a Union

Modify an item in a union. The union may contain multiple multi-level unions. Parts inside unions can be edited directly, for example, changing the height of a cuboid. When more complex editing is required (setting properties on faces), the part should be copied out, edited and placed back into the union.

1. In the model tree, find the relevant union that you want to edit.
2. Under the union, select the part that you want to edit.
3. From the right-click context menu, select **Copy (duplicate)**.
4. Edit the duplicated part.
5. In the model tree, drag the modified part back into the union of Step 2.
6. From the right-click context menu, select **Replace**.
   You can replace a part with a completely different part. For example, replace a cuboid with a flare or sphere.

Subtract From

Create complex geometry by subtracting geometry from an overlapping (target) part.

After a subtract operation is performed, the target part is indicated by the icon in the model tree.

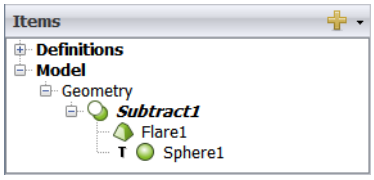


Figure 65: The T *indicating the target part in the subtract operation.*

Note: Regions are taken into account during the subtract operation:

- Subtracting a shell structure (a free space region that implies an empty / hollow closed structure) generates new regions, faces or edges on intersecting geometry. No geometry is removed.
- Subtracting a solid structure (a region that is not set to free space) results in geometry being removed from any intersecting parts.

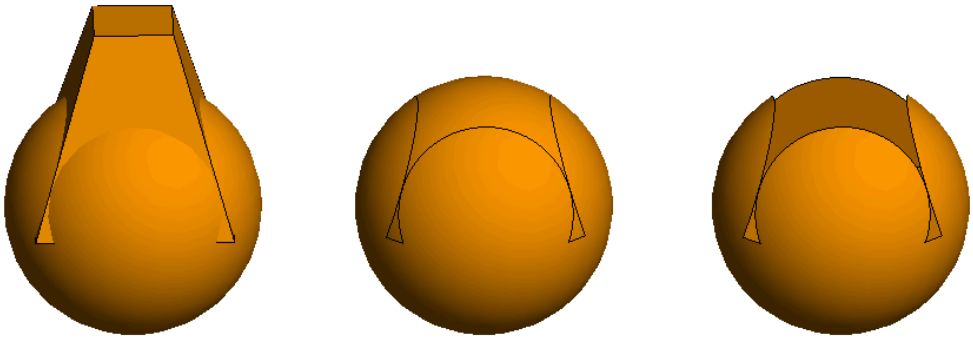


Figure 66: An intersecting flare and sphere (on the left), flare subtracted from sphere (shell) (middle) and flare subtracted from sphere (solid) (to the right).

Subtracting Geometry

Apply the subtract operation to remove the overlapping part of the geometry.

1. Select the geometry part to subtract.
2. On the **Construct** tab, in the **Modify** group, click the **Subtract From** icon.
3. Select the geometry part to be subtracted from (target).

The overlapping part of the geometry is removed (for solids) or new faces, edges, regions are created on the overlapping part (for shells).

Intersection

Create complex geometry by removing non-overlapping parts and keeping the common part.

> **Note:** If an intersection operation intersects two overlapping faces, the resulting faces have the properties common to both parents.

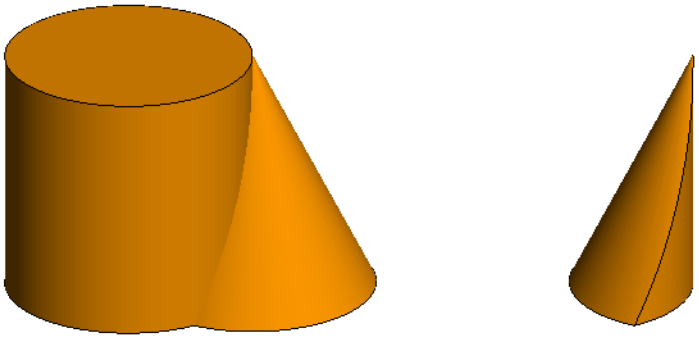


Figure 67: A cylinder and cone (on the left) and the intersection of the cylinder and cone (on the right).

Intersecting Geometry

Apply the intersect operation to remove non-overlapping parts.

1. Select the relevant geometry parts.
2. On the **Construct** tab, in the **Modify** group, click the **Intersection** icon.

The overlapping parts are removed.

Split

Divide the selected geometry parts at a specified plane.

> **Note:** If a subtract operation splits a face in two, both the resulting faces inherit the properties of the parent (original) face.

Splitting a Geometry Part at a Specified Plane

Apply the split operation at a specified plane to divide a geometry part.

1. Select the geometry part that you want to split.
2. On the **Construct** tab, in the **Modify** group, click the **Split** icon.

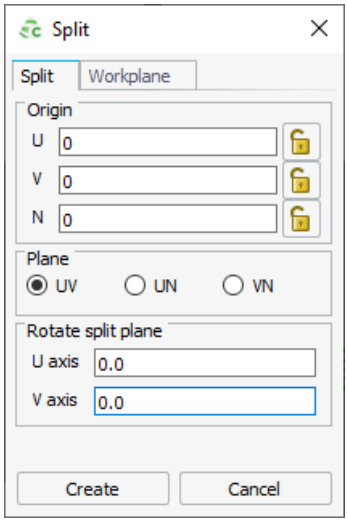


*Figure 68: The **Split** dialog.*

3. Under **Origin**, specify the position of the plane to split the geometry.
4. Under **Plane**, select one of the following:
   - **UV**
   - **UN**
   - **VN**
5. Under **Rotate split plane**, specify the angle of rotation around the plane selected in Step 4.
6. Click **Create** to split the geometry and to close the dialog.

Separate

Obtain the individual parts that were used to create a union.

A union could consist of several parts of which some could also be unions themselves. The separate tool, in a sense, copies out all the parts that were used to make the last union. If the separated (copied out) parts are unions themselves, then the tool can be run again to separate those parts as well.

> **Note:**
> When a union that contains a transform is separated the transform is deleted and the entities revert back to the state prior to being unioned.

Separating a Union

Apply the separate operation to separate (disband) a union.

1. Select the geometry part that you want to separate.
2. On the **Construct** tab, in the **Modify** group, click the **Separate** icon.
   All the parts contained in the union are listed in the model tree.

Convert to Group

Collect all the sub-parts (child parts) of a parent part into a group.

A part could consist of several sub or child parts of which some could contain further child parts. This tool converts the child parts into a group with a single label.

> **Note:** Only the highest level child parts are converted to a group.

Converting a Part to a Group

Apply the convert to group tool to obtain the child parts of a part and group them together.

1. Select the geometry part that you want to convert to a group.
2. On the **Construct** tab, in the **Modify** group, click the **Convert to Group** icon.

   The child parts contained in the parent part are grouped together under a single label in the model tree.

Stitch

When imported geometry (sheet parts) are unconnected or have small sections that overlap, use the stitch operation to ensure mesh connectivity.

Sheet parts that are within the specified tolerance are considered to be connected and meshed correctly.

The stitch tool can lead to strange geometry display in CADFEKO due to the tolerance of edges, faces and nodes being large and displayed anywhere within the tolerance area. The mesh does not suffer the same display issues since the mesh elements have a very small tolerance.

> **Note:**
>
> Use the stitch operation as a replacement for the union operation for sheet parts that have small misalignments or imperfections (usually introduced through CAD translation).
>
> The stitch operation is generally faster and more efficient than the union operation, but is limited to sheet bodies and introduces a tolerance (small uncertainty in the exact geometrical location).

Stitching Sheet Parts

Apply the stitch operation to ensure electrical connectivity for unconnected sheet parts.

1. Select the sheet parts that you want to stitch.
2. On the **Construct** tab, in the **Modify** group, click the **Stitch** icon.

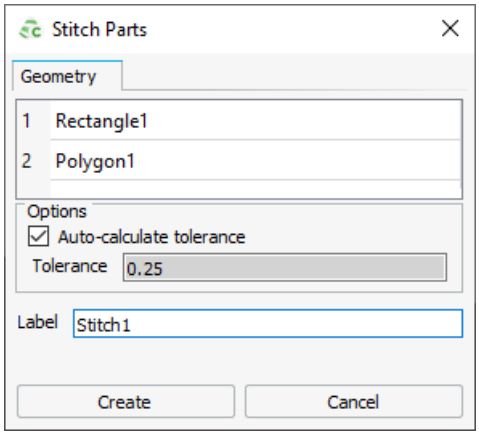


*Figure 69: The **Stitch Parts** dialog.*

3. [Optional] Clear the **Auto-calculate tolerance** check box and in the **Tolerance** field specify the range over which adjoining faces are stitched.
4. Click the **Create** button to stitch the faces and to close the dialog.

Electrical Connectivity When Combining Geometry

Geometry parts need to physically connect to ensure electrical connectivity once the model is meshed (except when using FDTD). Use the union, stitch or imprint operation to ensure electrical connectivity.

The simulated model has electrical connectivity as long as the mesh elements align and do not intersect. This results in the correct basis functions being created for the simulation. The following operations allow for geometry parts to be physically connected by ensuring that the resulting mesh elements align correctly:

Union
: This operation is used to physically connect geometry parts. This is the default and most common operator used in CADFEKO.

Stitch
: This operation is used to physically connect imported sheet parts where geometry is unconnected or have small sections that overlap. The stitch operation is generally faster than the union operation.

Imprint points
: This option allows you to specify points and projecting the points onto the closest point of the selected geometry part, either on a face or on an edge. The imprinted points are considered when meshing the model. Snap to the imprinted points to create physically connected geometry. Imprinting points is especially useful when connecting wires to faces when you do not want to union the wire onto the face.

Related concepts

Union

Stitch

Display Setting for Mesh Connectivity
Related tasks
Imprinting Points onto a Face or Edge

2.8.5 Extending Geometry to Create Complex Geometry

Use the spin, loft, sweep and path sweep on basic geometry to create complex geometry.

Spin

Rotationally sweep a geometry part, containing only edges and faces (not solids or closed regions), around an axis through a specified angle.

The spin operation applied to lines produces faces and to faces produces volumes. For surface bodies, the body must have a single boundary which does not close on itself and no edge may be attached to more than two faces.

Note: The spin operation is applied separately to each of the selected geometry parts.

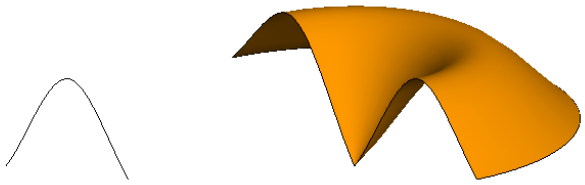


Figure 70: Spinning a curve results in a surface.

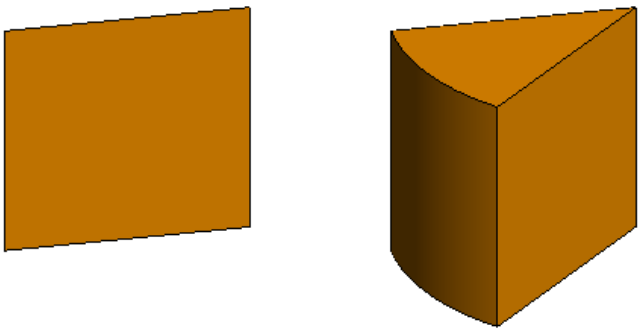


Figure 71: Spinning a surface results in a solid.

Spinning Geometry to Create Surface or Solids

Apply the spin operation to rotate the selected geometry around an axis to create surfaces or solids.

1. Select the geometry part that you want to spin.

2. On the **Construct** tab, in the **Extend** group, click the **Spin** icon.

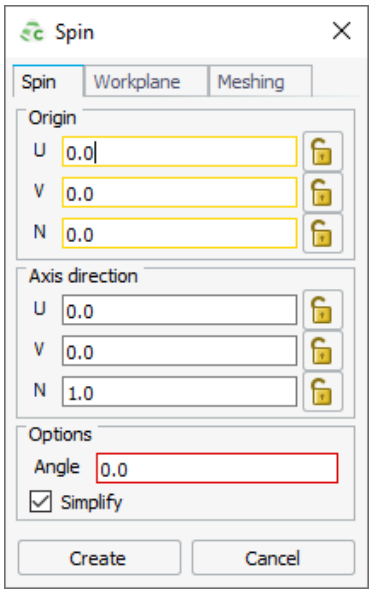


*Figure 72: The **Spin** dialog.*

3. Under **Origin**, specify the origin around which the geometry is spun.
4. Under **Axis direction**, specify the orientation of the spin axis.
5. Under **Rotation angle**, specify the angle through which the geometry is spun.
6. [Optional] Select the **Simply** check box to remove redundant vertices and edges from the spun geometry.
7. Click **Create** to spin the selected geometry and to close the dialog.

Loft

Create a smooth surface by connecting two curves, two surfaces or an edge and a surface; or connect two surfaces to create a solid shape.

Loft any geometry parts that contain edges / faces. Solid or closed regions cannot be lofted.

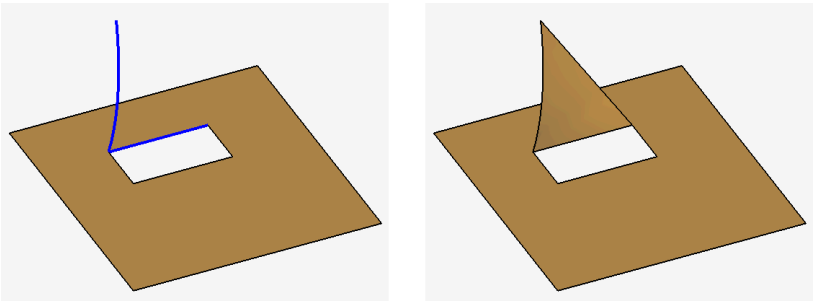


Figure 73: The lofting of edges within parts.

For surface bodies, the body must have a single boundary which does not close on itself and no edge may be attached to more than two faces.

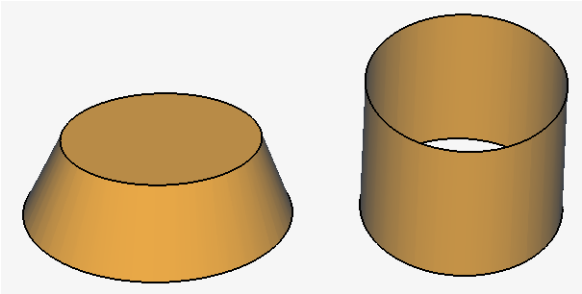


Figure 74: The loft of two ellipses to create a solid (on the left) and the loft of two elliptic arcs to create a cylindrical surface (on the right).

For curved bodies (bodies without faces), the body must be continuous. Open profiles (arcs) and closed profiles (circles) may be lofted, but cannot be used together in a single loft. A valid surface loft can be created, for example, between two lines, between two circles (closed elliptical arcs), between a line and an open polyline, or between a circle and a closed polyline, but a line cannot be lofted to a circle or to a closed polyline.

Different surface primitives, such as ellipses and rectangles, can be specified as the loft cross-section profiles to create a solid body.

If the two loft profiles have an equal number of edges or vertices, the loft operation connects each pair of edges. For an unequal number of wires / edges, some vertices on one profile are matched to a single vertex on the other profile. Points can be imprinted on one or both of the profiles to improve matching, or to influence the shape of the loft.

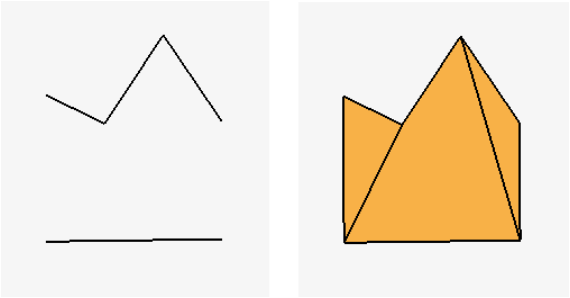


Figure 75: A line and polyline (on the left) and the loft result (on the right).

When lofting closed edges or faces, use the **Alignment index** to change the relative alignment of the two profiles in the loft, thereby introducing or removing twists.

Loft operations can be performed on edges / wires and faces within parts if the selected entities are of the same type. Entities within parts are copied out during the loft operation.

Note: Copied entities are snapshots of the model when the copy was made. The resulting loft is not linked to the parent object in any way and will not update with changes made to the original geometry.

Related tasks
Alignment Index

Lofting to Create a Smooth Surface

Apply the loft operation between curves, between surfaces or between an edge and face to create a smooth surface.

1. Select the two geometry entities that you want to loft.
2. On the **Construct** tab, in the **Extend** group, click the **Loft** icon.

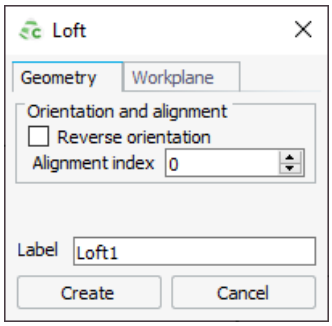


*Figure 76: The **Loft** dialog.*

To achieve a valid loft, it may be necessary to reverse the orientation of one of the profiles in the loft, so that the bounding edge matching is done in the opposite direction along the profile.

3. [Optional] Select the **Reverse orientation** check box to reverse the orientation of one of the profiles in the loft.

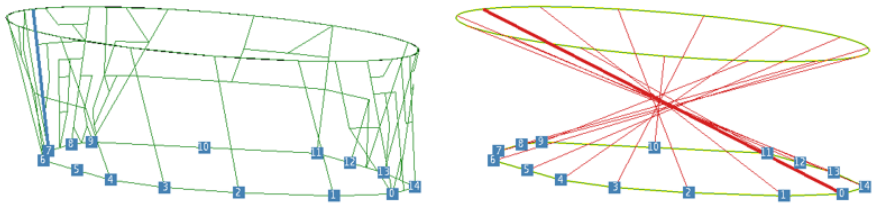


Figure 77: Loft preview showing a valid loft (on the left) and an invalid loft (to the right).

When lofting closed edges or faces, you can introduce twists or remove twists to change the relative alignment of the two profiles in the loft.

4. In the **Alignment index** field, increase the number to change the alignment index.
5. In the **Label** field, specify a unique label for the loft operation.
6. [Optional] Use the **Workplane** tab to enter a resulting origin (location) and orientation for the loft.
7. Click the **Create** button to apply the loft operation and to close the dialog.

Sweep

Extrude the selected geometry part from a start point to an end point.

The sweep path is taken as the straight line between the start point and the end point.

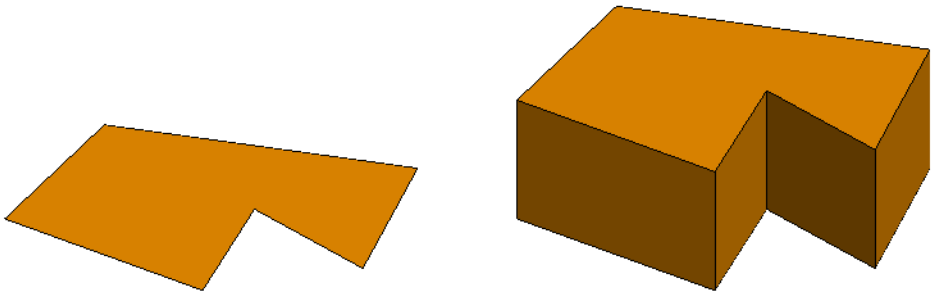


Figure 78: A polygon (on the left) and the polygon swept along the Z axis (on the right).

Sweeping Geometry to Create a Surface or Solid

Apply the sweep operation to curves to create surfaces, and to surfaces to create solids.

1. Select the geometry part that you want to sweep.
2. On the **Construct** tab, in the **Extend** group, click the **Sweep** icon.

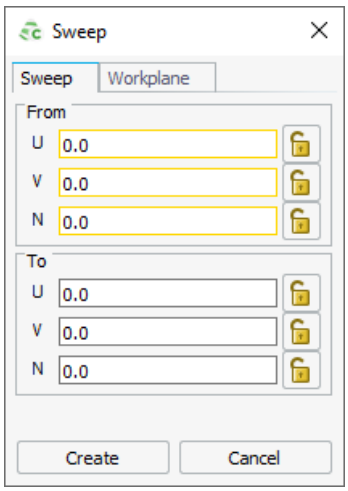


*Figure 79: The **Sweep** dialog.*

3. Under **From**, specify the start point for the sweep operation.
4. Under **To**, specify the end point for the sweep operation.
5. Click **Create** to sweep the geometry and to close the dialog.

Path Sweep

Sweep (or extrude) geometry along a path.

The sweep path may be any part that consists of only free edges and curves that form a joined, non-overlapping path.

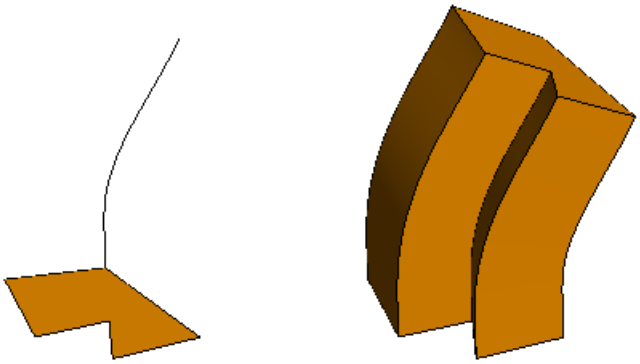


Figure 80: A polygon and fitted spline (on the left) and the polygon swept along the fitted spline (on the right).

Sweeping Geometry Along a Path

Sweep curves along a path to create surfaces, and surfaces to create solids.

1. Select the geometry part that you want to sweep.
2. On the **Construct** tab, in the **Extend** group, click the **Path Sweep** icon.

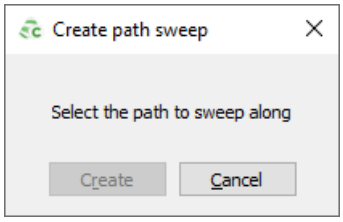


*Figure 81: The **Create path sweep** dialog.*

3. Select the path to sweep along.

The selected geometry is swept along the specified path.

2.8.6 Transforming Geometry

Transform geometry (or meshes) using the translate, mirror, rotate, scale and align operations.

Translating Geometry from Start Point to End Point

Move the selected geometry from a specified start point to an end point.

1. On the **Transform** tab, in the **Transform** group, click the **Translate** icon.

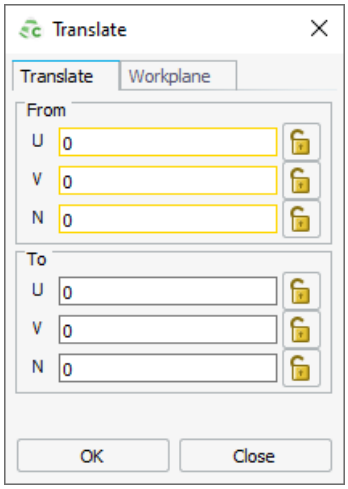


*Figure 82: The **Translate** dialog.*

2. Under **From**, specify the start point of the translation.
3. Under **To**, specify the end point of the translation.
4. Click **OK** to translate the selected geometry parts and to close the dialog.

Mirroring Geometry about an Axis

Apply the mirror operation to move the selected geometry part about an axis.

1. Select the geometry part that you want to mirror.
2. On the **Transform** tab, in the **Transform** group, click the **Mirror** icon.

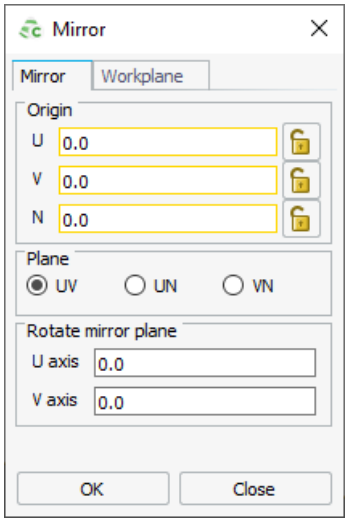


*Figure 83: The **Mirror** dialog.*

3. Under **Origin**, specify the origin of the mirror operation.
4. Under **Plane**, specify the mirror plane.
5. Under **Rotate mirror plane**, specify the rotation of the mirror plane.
6. Click the **OK** button to mirror the selected geometry part and to close the dialog.

Rotating Geometry

Apply the rotate operation on the selected geometry part.

1. Select the geometry part that you want to rotate.
2. On the **Transform** tab, in the **Transform** group, click the **Rotate** icon.

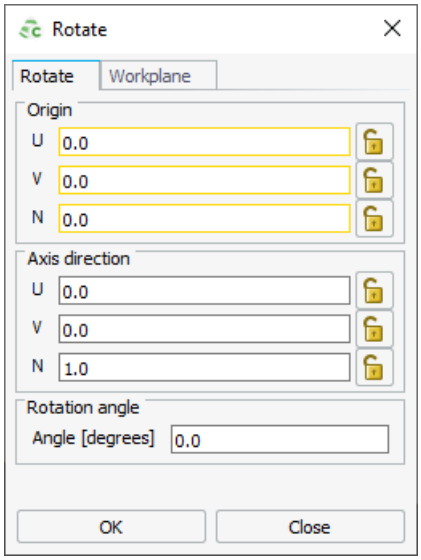


*Figure 84: The **Rotate** dialog.*

3. Under **Origin**, specify the origin for the rotate operation.
4. Under **Axis direction**, specify the direction of the axis around which the rotation will take place.
5. In the **Angle [degrees]** field, specify the rotation angle in degrees.
6. Click the **OK** button to rotate the selected geometry part and to close the dialog.

Scaling Geometry

Apply a scale operation on the selected geometry part.

1. Select the part that you want to scale.
2. On the **Transform** tab, in the **Transform** group, click the **Scale** icon.

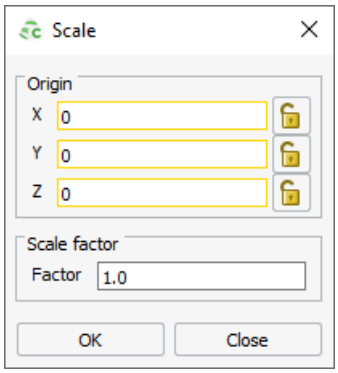


*Figure 85: The **Scale** dialog.*

3. Under **Origin**, specify the origin around which the scaling is applied.
4. Under **Scale factor**, specify the scaling factor.
5. Click **OK** to scale the selected geometry part and to close the dialog.

Placing Geometry on Objects (Align)

Align an object onto another object, for example, placing an antenna onto a ship.

1. Select the geometry part that you want to align relative to another part.
2. On the **Transform** tab, in the **Transform** group, click the **Align** icon.

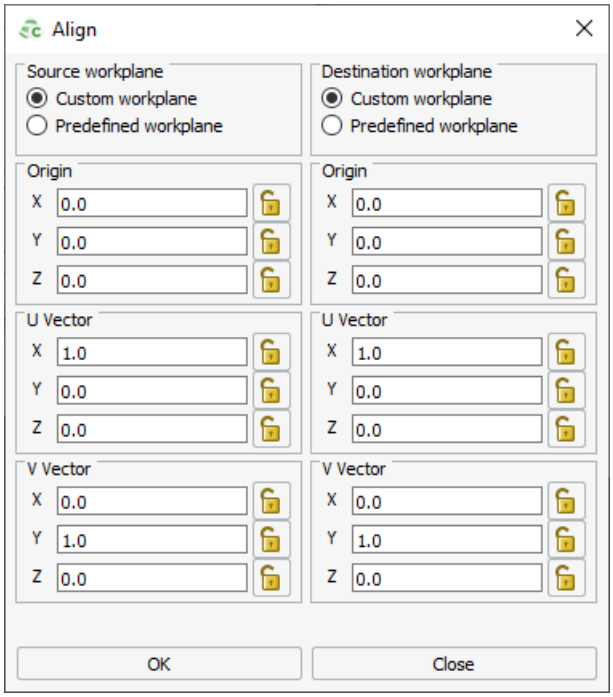


*Figure 86: The **Align** dialog.*

3. Under **Source workplane**, specify the following:
   a) Under **Origin**, specify the origin for the source workplane.
   b) Under **U vector** and **V vector**, specify the workplane orientation.
   c) Under **Rotate workplane**, specify the angle to rotate the workplane by.
4. Repeat Step 3 for the **Destination workplane**.
5. Click **OK** to create align the selected geometry part and to close the dialog.

> **Tip:** Press Ctrl+Shift at the location of the desired origin in the 3D view.

Projecting Geometry onto Other Geometry

Project the edges of the selected part onto a target part. Where projected faces form a closed path, a new face is created.

1. Select the part whose edges you want to project.
2. On the **Transform** tab, in the **Imprint** group, click the **Project** icon.

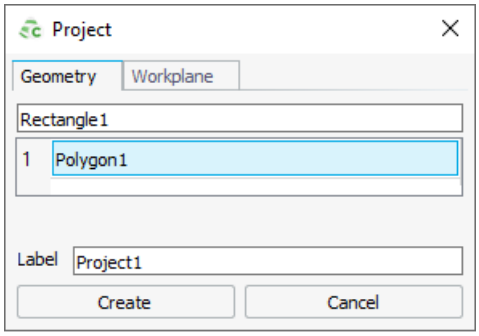


*Figure 87: The **Project** dialog.*

3. Select the target part to project onto.

The edges of the selected parts are projected onto the faces of the target part.

Imprinting Points onto a Face or Edge

Project a list of specified points onto the selected geometry (either on a face or an edge) to ensure vertices at these points after meshing.

The specified points are projected onto the closest point on the selected part.

Note:

- Points may not be imprinted on top of existing points.
- Points can only be imprinted on a single geometry part at a time.

1. Select the geometry part where you want to imprint the points.
2. On the **Transform** tab, in the **Imprint** group, click the **Imprint Points** icon.

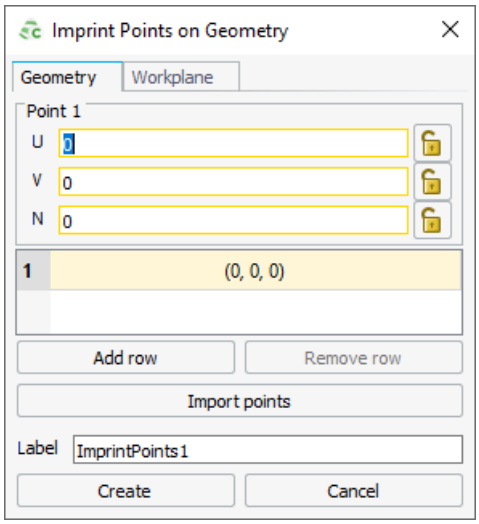


*Figure 88: The **Imprint Points on Geometry** dialog.*

3. Specify the points to imprint using one of the following workflows:
   - Specify the points manually.
   - Import the points from a file.
   - Use point-entry to specify the points.
4. Click **Create** to imprint the points and to close the dialog.

The imprint operation creates a new entry in the model tree to still allow access to the part without the imprinted points.

Modifying Face Normal / Orientation

The face normal of a triangle is determined in a mathematically positive sense from the direction of the edges.

A face normal is of importance when it is required to specify settings for a specific side of a face since the setting is applied on either the face normal side or the opposite side. The face normal can be modified so that a group of faces have the same orientation and settings can be applied to the group.

For example:

- PO using the option to only illuminate from the front.
- RL-GO when setting the face absorbing properties for the normal side and opposite to normal side.
- The combined field integral equation (CFIE) requires closed structures and the face normal to point away from the zero-field region.
- Windscreen and specifying the order of the layers for the windscreen.
- Thin dielectric sheets for RL-GO, PO and LE-PO, the normal side is important when the specifying the order of the layers.

Related concepts
Thin Dielectric Sheets
Related tasks
Creating a Windscreen Layer
Using the CFIE For Closed PEC Regions
Solving Faces with Physical Optics (PO)
Solving a Model with RL-GO

Reversing Face Normals

Invert the normal of a face.

1. Select the part for which you want to reverse the face normal.
2. On the **Transform** tab, in the **Alter** group, click the **Reverse Normals** icon.

Use a display setting to colour the geometry according to the normal direction of the faces.

3. [Optional] On the **3D View** context tab, on the **Display Options** tab, in the **Style** group, click the **Colour** icon. From the drop-down list select **Element Normal**.

Note:

- Geometry and mesh
  - Normal side: Green
  - Reverse side: Red

Simplify

The simplify tool removes redundant regions, faces and edges.

Dielectric boundary faces are redundant if they have the same medium (for example, free space, the same metal or dielectric) on both sides. When a face separating an internal free space region from the outside free space is deleted, the internal region is merged with the outside one. Since the outside medium is free space, faces can only be removed from closed regions if the internal medium is set to free space.

Edges are not redundant if the face normal on either side of the edge are in opposite directions.

The simplify operation results in the model being electromagnetically the same as the original, but may not have the same meshing constraints. For example, if an imprinted point is removed, a mesh vertex cannot be guaranteed at this location. Faces cannot be deleted unless the regions they separate can be merged. The same applies to edges on face boundaries and geometry points at the ends of edges.

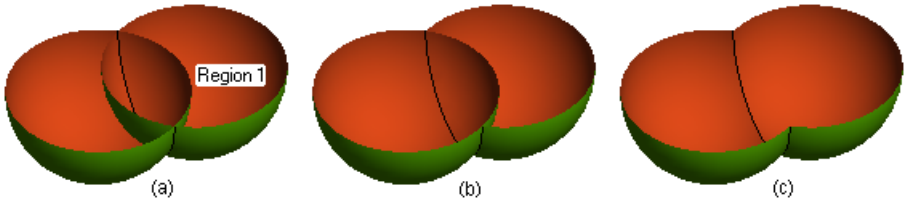


Figure 89: Simplify operations.

Simplifying Geometry

Use the simplify tool to remove redundant regions, faces and edges.

1. On the **Transform** tab, in the **Simplify** group, click the **Simplify** icon.

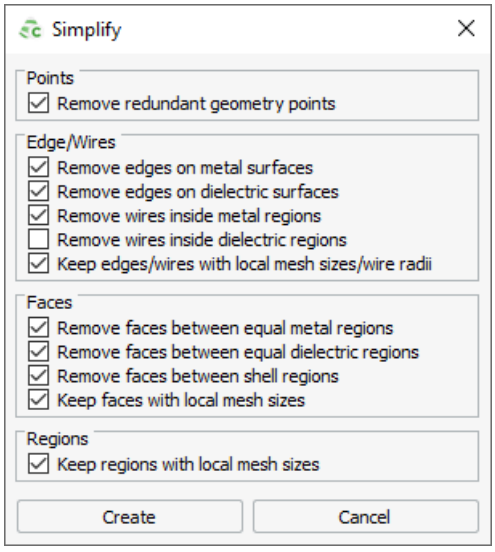


*Figure 90: The **Simplify** dialog.*

2. Select the relevant options.
3. Click **Create** to simplify the model and to close the dialog.

Exploding Geometry Parts

Break up a geometry part into separate faces and wires.

> **Note:** The new parts represent a snapshot of the geometry at the time it was exploded. The faces and wires are not parametric.

1. Select the geometry part that you want to explode.
2. On the **Transform** tab, in the **Alter** group, click the **Explode** icon.

The geometry part is exploded into separate parts and listed in the model tree. The icon indicates that it is an exploded face.

2.8.7 Re-Evaluating Geometry

Re-evaluate rebuilds the full model and performs all operations again, replacing any cached geometry in the operator tree. It is mostly used when there have been improvements or corrections in Parasolid (3D modelling engine).

CADFEKO uses advanced mapping algorithms to keep track of individual items when the geometry is modified. A model re-evaluates automatically when a model is loaded that was created in earlier versions. Resolve any suspect items before making any changes to the model or setting additional properties.

Not all the mapping information is available in models created in earlier CADFEKO versions. As a result, it may be impossible to map all items during the re-evaluation and the items are marked suspect. All properties (such as local mesh sizes) set on suspect items, will be lost during re-evaluation. Faces or edges that were deleted, may re-appear if they cannot be successfully mapped.

It is possible that during geometry re-evaluation, new faults are identified in the model. This could happen with models built in previous versions of CADFEKO, or models containing imported geometry. These faults may provide additional information that is useful in repairing a model.

Re-Evaluating Geometry

For models created in previous CADFEKO versions, re-evaluate to rebuild and perform all operations on the model again.

1. Select the geometry part or full model that you want to re-evaluate.
2. On the **Transform** tab, in the **Alter** group, click the **Re-evaluate** icon.

The selected geometry part (or full model) is re-evaluated.

2.8.8 Excluding a Part from the Model and Solution

A geometry part (or mesh part) that does not contain any ports, sources or loads can be temporarily excluded from the model without having to delete the part.

1. In the model tree select the geometry part (or mesh part) that you want to exclude from the model.
2. From the right-click context menu, select **Include/Exclude**.

The excluded part is hidden in the 3D view. The part is not (re)meshed and is excluded from the **Mesh Info** dialog.

An excluded part is indicated by the icon in the model tree.

> **Note:** To include a part, from the right-click context menu select **Include/Exclude**.

Related tasks
Viewing the Mesh Information

2.8.9 Hiding a Part in the 3D view

Hide a part temporarily in the 3D view.

1. In the model tree select the geometry part (or mesh part) that you want to hide in the 3D view.
2. Select one of the following workflows to hide the selected part:
   - From the right-click context menu, select **Show / Hide**.
   - Press Ctrl+H.

The selected part is hidden in the 3D view. A hidden part is indicated by a grey icon in the model tree.

2.8.10 Opening Legacy Models in CADFEKO

CADFEKO models created before version 2022.1 are converted to a new format when opening in CADFEKO 2024.1.

When opening a legacy model in CADFEKO, the original model is stored in a backup folder at:

```
%FEKO_USER_HOME%/converted_models
```

When geometry fails to convert during legacy conversion, the geometry part is converted to a primitive. If a part consisting of boolean operators fails to convert to the new format, a group is created to show the creation hierarchy.

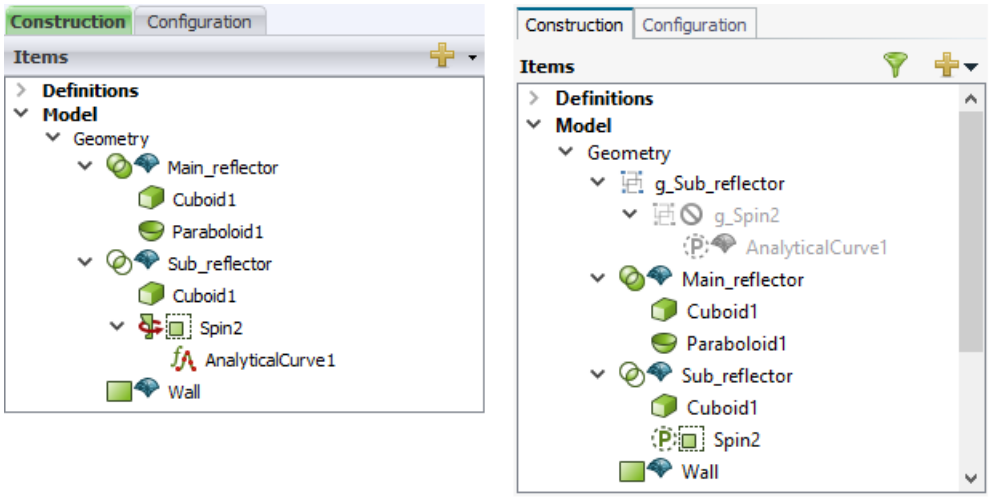


Figure 91: An example of where a geometry part failed to convert to the new format. Left: legacy CADFEKO; Right: An excluded group is created that contains the failed boolean operator in CADFEKO.

Tip: When a model fails to convert during legacy conversion, save the model in a newer legacy version[11].

Related information
CADFEKO [LEGACY] Model Fails to Open in CADFEKO

11. See How to run CADFEKO [LEGACY] for more information.

2.9 Component Library

The component library contains an extensive list of common components, such as antennas and platforms.

This tool allows you to add components to new or existing models. Change the centre frequency, solver method and other settings to create an antenna that suits your requirements, thereby reducing the development time.

The component library tool is also a great way for students to learn about the different types of antennas and its typical characteristics.

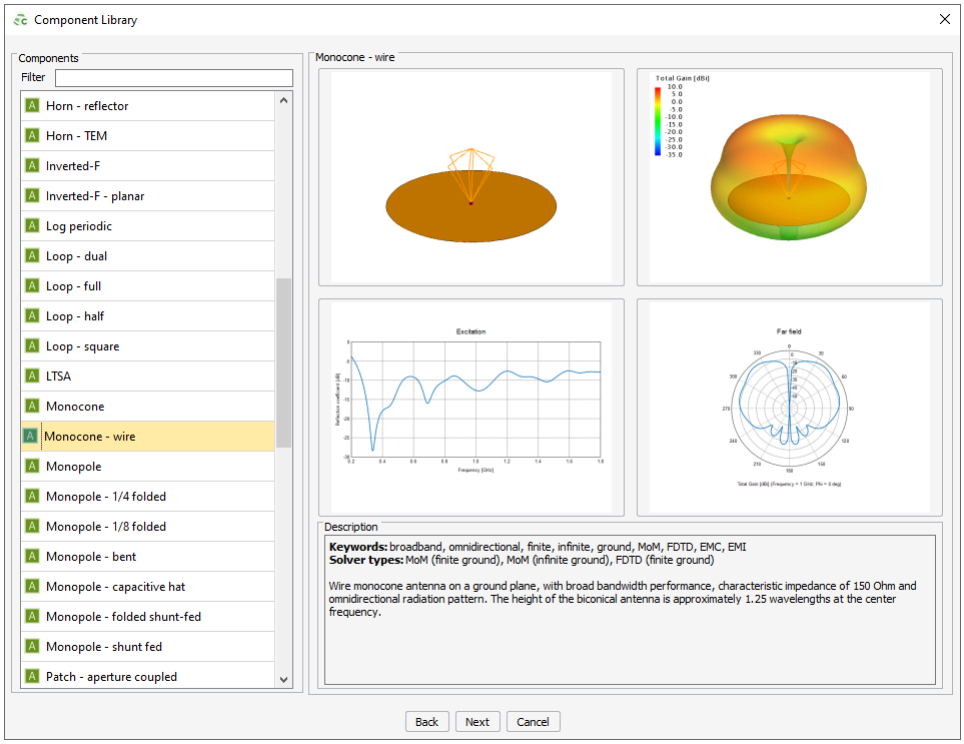


Figure 92: The component library.

2.9.1 Introduction to the Component Library

The component library contains a list of components that can be used as a new model or be added to an existing model.

A component can be one of the following:

- *Antenna*

  Each antenna component is defined by fully parametric geometry, created using variables and mathematical expressions. When a variable is modified, any item in the antenna that references the variable is re-evaluated and updated.

  Antennas are indicated by the A icon in the library.

- *Platform*

  A platform is geometry that is used as a mechanical structure for mounting antennas, for example, a tower truss.

  Platforms are indicated by the P icon in the library.

2.9.2 Component Library Start Page

The component library start page is displayed when opening the **Component library** dialog, and a component has yet to be selected from the component list.

The start page shows the steps for adding a component.

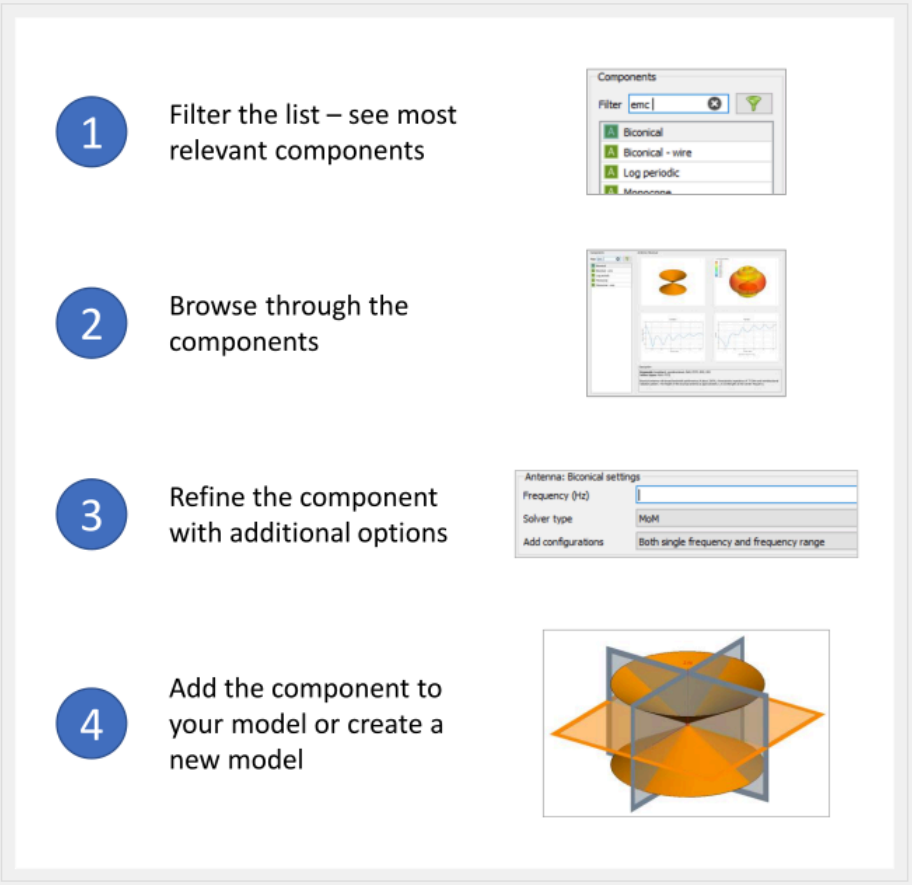


Figure 93: The component library start page.

Tip: To view the component library start page once a component was selected, close and reopen the **Component library** dialog.

2.9.3 Quick Tour of the Component Library

The component library consists of two main panels.

The first panel allows you to browse through the available components. The second panel (only applicable to antennas) allows you to refine the selected antenna.

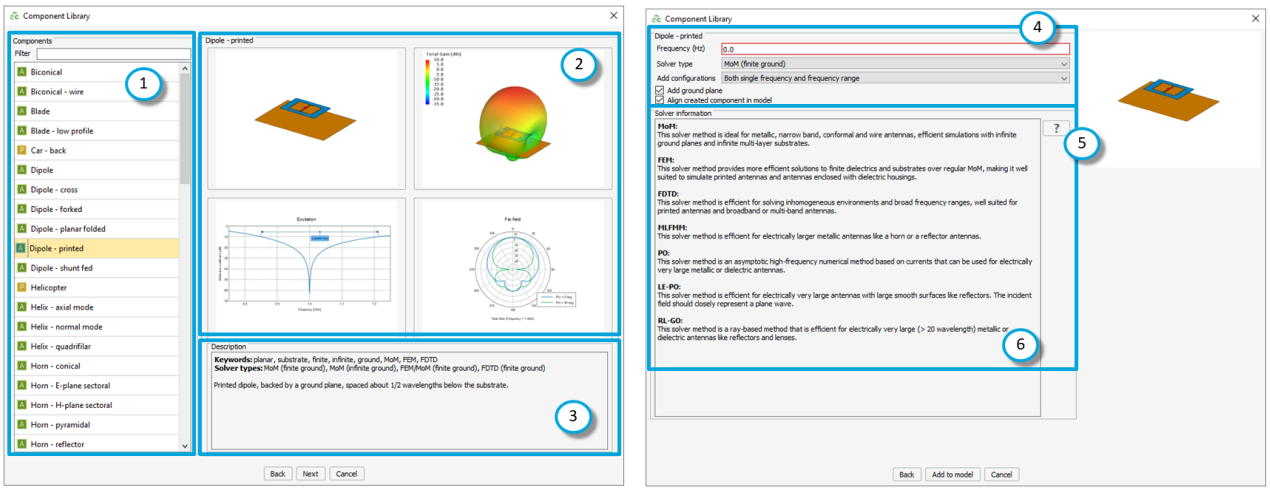


Figure 94: The Component library dialog. On the left, the first panel for browsing components and to the right, the second panel for refining the component.

1. *Components*

   View the list of components (antennas and platforms) in the component library. Find a component quickly by either entering filter text or filter the list according to the component type. To select a component, click on an item in the list.

2. *Geometry preview / results*

   View images of the selected component in 1. Click on an image to maximise and click again to minimise.

3. *Description*

   View a summary of the selected component in 1, as well as keywords relevant to the component.

   > **Tip:** Use the keywords as filter text in 1 to find similar components.

4. *Antenna settings*

   Refine the selected antenna with additional options and settings.

5. 

   Click the  icon to view the solution methods in the Feko documentation.

6. *Solver information*
   View a summary of the solution methods that can be selected to solve the antenna. Not all solution methods are supported for each component due to differences in how the component model is set up.

2.9.4 Workflows for Using the Component Library

There are two workflows available when using the component library.

1. Browse the component library, select a component, and create a new model[12].
2. Open an existing `.cfx` file and add a component to the existing model.

2.9.5 Adding a Component to a Model

Browse the components in the component library, select a component and either create a new model from the component or add the component to an existing model.

1. Open the **Component library** dialog using one of the following workflows:
   - On the **Home** tab, in the **File** group, click the **Component Library** icon.
   - Press Ctrl+L to use the keyboard shortcut.

   The **Component library** dialog is displayed
2. Select the component using one of the following workflows:
   - Select a component from the library by clicking on a component in the list.
   - Enter the filter text[13] criteria in the **Filter** field.
3. Proceed to the next step in adding the component:
   - if you selected an antenna, click **Next**[14] and proceed to Step 4.
   - if you selected a platform, proceed to Step 9.
4. In the **Frequency (Hz)** field, specify the centre frequency of the antenna.
   Frequency scaling will be applied and the geometry of the antenna will be added according to the specified centre frequency,
5. From the **Solver type** drop-down list, select the desired solver type for the antenna.
6. In the **Add configurations** field, select one of the following:

12. This workflow assumes that there is no model open (the CADFEKO start page is displayed).
13. The filter text criteria is not case-sensitive.
14. An alternative method is to double-click the antenna in the list.

- *Both single frequency and frequency range*
  Select this option to add two standard configurations to the model. The first configuration is at a single frequency and the second configuration uses a continuous (interpolated) range.

- *Single frequency*
  Select this option to add a standard configuration at a single frequency to the model.

- *Frequency range*
  Select this option to add a standard configuration using a frequency range to the model.

  For FDTD a discrete frequency configuration is added.

- *None*
  Select this option if no configurations are to be added to the model.

7. Click the **Add ground plane** check box to add a ground plane. This option is only available for antenna components that have a ground plane (either finite or infinite ground plane).
8. Click the **Align created component in model** check box if you want to use the **Align** tool to place the component.
9. Click **Create model/Add to model** to create the model or to add the component to the existing model.
   If the **Align created component in model** check box was selected in Step 8, the **Align** dialog is displayed. Use the **Align** tool to change the placement and orientation of the component.

   Note: Click **Cancel** on the **Align** dialog to use the default placement at the origin.

The component is added as a new model or added to an existing model using the default workplane.

Related concepts
Workplanes
Related tasks
Placing Geometry on Objects (Align)

Component Library Conventions

A set of naming conventions are used in the definition of antenna-specific variables, configurations, and units in components.

Variables

All antenna-specific variables start with the antenna name (for example, *Dipole1*). When a new antenna is added and one of its variable names is not unique, the antenna name is incremented (for example, *Dipole1*, *Dipole2*).

An underscore ("_") is used as a delimiter.

Note: To change the centre frequency of an antenna after it was added to a model, modify the *antenna_f_ctr* variable (for example, *Dipole1_f_ctr*).

Configurations

All antenna-specific configurations start with the antenna name (for example, *Dipole1*).

- A configuration specified at a single (centre) frequency is indicated by the *_f_ctr* suffix (for example, *Dipole1_f_ctr*).
- A configuration specified over a frequency range is indicated by the *_f_range* suffix (for example, *Dipole1_f_range*). For a FDTD a discrete frequency configuration is added.

Units

When creating a component in a new model, the model unit is set to millimetres if the centre frequency is greater than 1 GHz.

Related concepts

Variables

Multiple Configurations

Model Unit

2.10 Groups

Organise geometry and mesh parts by grouping geometry and mesh in the model tree.

2.10.1 Creating a Group

Move the selected geometry parts or mesh parts into a group in the model tree.

> **Note:** Geometry parts and mesh parts cannot be added to the same group.

1. Select the geometry part or mesh parts that you want to place in a group.
2. On the **Transform** tab in the **Groups** group, click the **Create** icon.

A group is indicated by the icon in the model tree.

2.10.2 Moving a Part into an Existing Group

Move a selected geometry part or mesh part into an existing group.

1. Select the geometry part or mesh parts that you want to move into in a group.
2. From the right-click context menu, select **Group** > **Move to**.
3. From the drop-down list select one of the following:
   - To create a new group, click **(new)**.
   - To move the selected part into an existing group, select a group.

2.10.3 Disassembling a Group

Separate a group and move its content back to the same level as the group in the model tree.

1. Select the group in the model tree that you want to disassemble.
2. On the **Transform** tab in the **Groups** group, click the **Disassemble** icon

The group is disassembled and its parts placed at the same level in the model tree as the original group.

2.10.4 Removing a Part from a Group

Remove a part from a group and place at the same level in the model tree as the group.

1. Select the geometry part or mesh parts that you want to remove from the group.
2. On the **Transform** tab in the **Groups** group, click the **Move Out** icon.

The selected part is moved out from the group and placed at the same level in the model tree as its former group.

2.11 Repairing Geometry

2.11.1 Creation History of a Geometry Part

Remove the creation history of a geometry part to reduce memory and processing time for complex models by converting it to a primitive part.

CADFEKO stores the creation history of every part, allowing you to modify the part at any point in the creation history, although the history may require a significant amount of memory and processing time for complex models.

For example, if a small component of a union operation is modified, CADFEKO needs to recreate the parent parts to re-execute the union. Since these are not stored at every level, it means constructing them again from the lowest level up. However, quite often a large part of the model remains unchanged.

For example, often the model of a specific motor vehicle will remain the same, but it is common to place different antennas on such a vehicle. As a result, it is not required to re-evaluate the vehicle each time a small part of the geometry is changed.

> **Note:** Depending on the complexity of the model, this operation may require a significant amount of time.

Converting a Geometry Part to a Primitive

Create a primitive of the geometry part.

1. Select the geometry part that you want to convert to a primitive.
2. On the **Transform** tab, in the **Alter** group, click the **Convert to Primitive** icon.

The primitive part is indicated by the icon in the model tree.

> **Note:**
> - The history of the CAD model is removed.
> - Removing the history can reduce the size of large `.cfx` files.

2.11.2 CAD Fixing Tools Overview

Use the CAD fixing tools to repair a range of CAD geometry issues and faults. The tools repair the fault-containing CAD model for analysis by reducing the complexity and removing faults in the model.

Tip:

- Apply the CAD fixing tools directly after importing a model.
- First, attempt the **Repair part** tool.
- In a number of cases, it may be necessary to apply several of the CAD fixing tools in succession to obtain a usable model.

For most cases, the default settings of the CAD fixing tools will suffice. When modifying the advanced settings of the CAD fixing tools, a general rule of thumb is to use the smallest tolerance possible, or no tolerance at all, if the option allows.

Note: Specified tolerances are given in the model unit.

2.11.3 Repair Part

The *Repair part* tool heals a body in an attempt to create a valid geometry part.

The tool attempts to fix the following:

- topology with an invalid sense
- invalid edge and vertex tolerances
- invalid geometry
- self-intersecting geometry
- non-G1[15] geometry
- missing edge or vertex geometry
- missing vertices
- vertices not on curve of edge
- edges and vertices not on surface of face

15. A surface can be composed of several NURBS surfaces known as patches. These patches should be fitted together in a way that the boundaries are invisible. This is mathematically expressed by the concept of geometric continuity. One of the options to establish geometric continuity is by means of Tangential continuity (G1). It requires the end vectors of two curves or surfaces to be parallel which eliminates sharp edges.

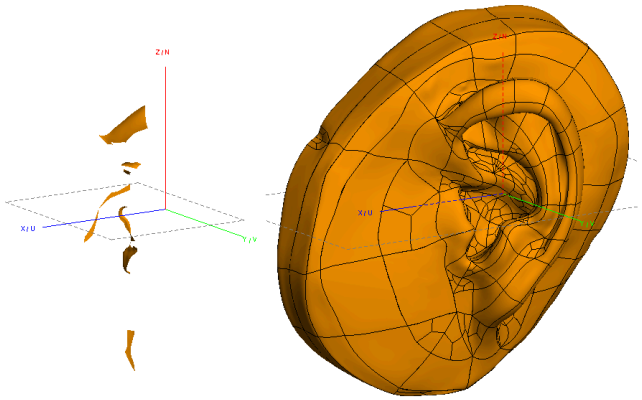


Figure 95: The imported model containing faults (left) and the result of the Repair part tool on the model (right).

On the **Transform** tab in the **Repair** group, click the **Repair Part** icon.

After repairing a part, the part is renamed to the default **RepairPart1** and the repair part icon is displayed next to the part in the model tree.

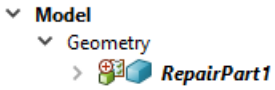


Figure 96: A repaired part displayed in the model tree.

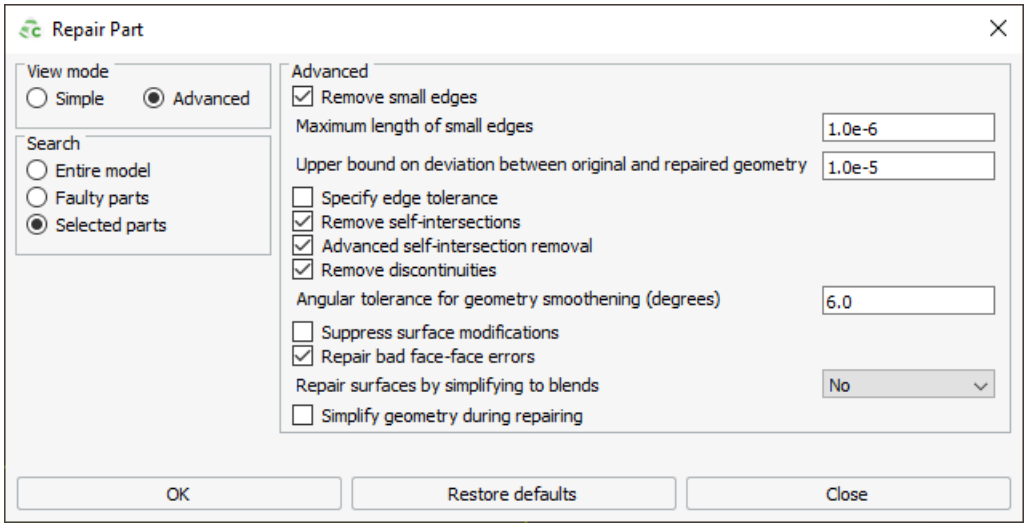


*Figure 97: The **Repair Part** dialog, **Advanced** tab.*

Advanced settings

Remove small edges
: Edges are removed whose arc length is less than the **Maximum length of small edges**.

Maximum length of small edges
: Edges are removed with arc lengths of edges less than the **Maximum length of small edges**.

Upper bound on deviation between original and repaired geometry
The tolerance for repairing the part.

Specify edge tolerance
To be more consistent with the surfaces in the model, it may be the case that the distance changes in the surface geometry are more important than the exact locations of the edges. If this option is selected, a more lenient tolerance for edge geometry can be specified for **Edge repair tolerance**.

Edge repair tolerance
An optional tolerance that is specified to permit greater latitude in repairing edges.

Remove self-intersections
When a surface contains self-intersections located outside its face boundaries, then this portion of the surface will be removed by splitting the surface. This may result in the face being split into multiple faces.

Advanced self-intersection removal
A more in-depth algorithm is used to fix self-intersecting surfaces.

Remove discontinuities
Surface discontinuities are removed. If the discontinuity has a change in the tangent of less than the angular tolerance, the discontinuity will be smoothed. If the change in tangent is greater than **Angular tolerance for geometry smoothening (degrees)**, the face or edge will be split at the surface's discontinuity. The same applies to curve G1 discontinuities.

Angular tolerance for geometry smoothening (degrees)
The tangent change angle in degrees above which G1 discontinuities are removed by splitting topology rather than smoothening the geometry.

Suppress surface modifications
Surface geometry is preserved and repairs are confined to repairing face boundaries as far as possible.

Repair bad face-face errors
Attempt to repair face-face collisions in the body.

Repair surfaces by simplifying to blends
Surfaces are cleaned by simplifying to blends.

Simplify B-surfaces to analytic / swept / spun surfaces
Any B-surfaces are simplified where possible to planes, cylinders, cones, spheres or tori where possible.

Simplify swept / spun surfaces to analytic surfaces
Any swept or spun surfaces are simplified to planes, cylinders, cones, spheres or tori.

Simplify B-curves to analytic curves
Any B-curves are simplified to lines, circles or ellipses.

Simplify rational B-geometry to non-rational geometry
Any rational B-surfaces are simplified to non-rational B-surfaces. Non-rational B-surfaces have fewer degrees of freedom than rational B-surfaces.

Reduce high-degree and trim large B-geometry
Any high-degree B-surfaces are trimmed or simplified to cubic B-surfaces.

Simplify to constant U or V curves
The tool will attempt to simplify SP-curves[16] to be constant in one parameter (U or V).

Merge multiple segments
The tool will attempt to merge multiple curve segments into a single segment.

Operating precision (tolerance)
The tolerance for replacement geometry.

Specify edge tolerance
To be more consistent with the surfaces in the model, it may be the case that the distance changes in the surface geometry are more important than the exact locations of the edges. If this option is selected, a more lenient tolerance for edge geometry can be specified for Edge repair tolerance.

Edge repair tolerance
This is an optional tolerance to permit greater latitude in repairing edges.

Convert surfaces to blend surfaces
The surfaces are cleaned by attempting to simplify to blends.

Constrain surface normals along smooth edges
The tool will attempt to ensure that smooth edges will remain smooth[17] since the maximum deviation between the normals for these faces will be equal to the surface normal tolerance.

Surface normal tolerance (degrees)
The angular tolerance for constraining surface normals in degrees.

2.11.4 Simplify Part Representation

The *Simplify part representation* tool simplifies a curve or a surface.

The tool will attempt to fix the following:

- simplification of rational and non-rational B-spline surfaces to analytic surfaces (plane, cylinder, sphere, cone, torus) where possible
- simplification of rational and non-rational B-spline curve to analytic curves (line, circle, ellipse)

16. They are surface parameter curves and are defined only in terms of the U and V parameters of the surface they belong to.
17. Edges between faces where there is a smooth transition from face to face.

- simplification of swept and spun surfaces to analytic surfaces (plane, cylinder, sphere, cone, torus) where possible
- simplification of surface parameter curves controlled by given options

On the **Transform** tab in the **Repair** group, click the **Simplify Part Representation** icon.

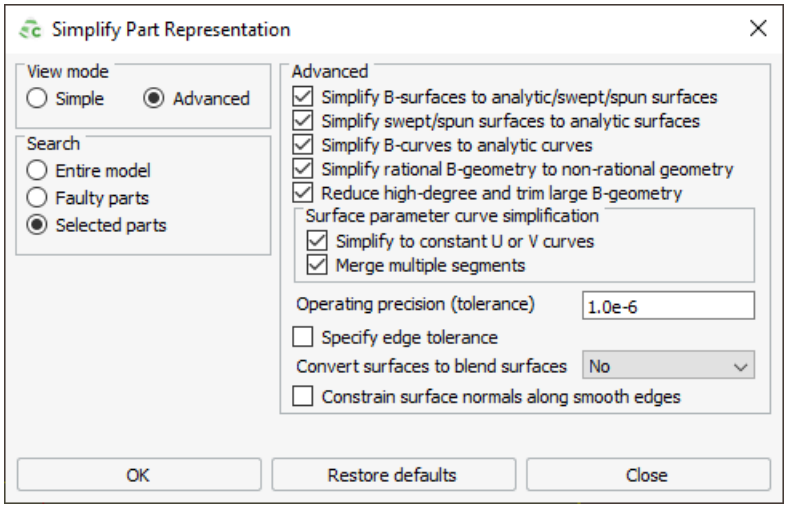


*Figure 98: The **Simplify Part Representation** dialog.*

Advanced settings

Simplify B-surfaces to analytic / swept / spun surfaces
: Any B-surfaces are simplified where possible to planes, cylinders, cones, spheres or tori where possible.

Simplify swept / spun surfaces to analytic surfaces
: Any swept or spun surfaces are simplified to planes, cylinders, cones, spheres or tori.

Simplify B-curves to analytic curves
: Any B-curves are simplified to lines, circles or ellipses.

Simplify rational B-geometry to non-rational geometry
: Any rational B-surfaces are simplified to non-rational B-surfaces. Non-rational B-surfaces have fewer degrees of freedom than rational B-surfaces.

Reduce high-degree and trim large B-geometry
: Any high-degree B-surfaces are trimmed or simplified to cubic B-surfaces.

Simplify to constant U or V curves
: The tool will attempt to simplify SP-curves[18] to be constant in one parameter (U or V).

18. They are surface parameter curves and are defined only in terms of the U and V parameters of the surface they belong to.

Merge multiple segments
The tool will attempt to merge multiple curve segments into a single segment.

Operating precision (tolerance)
The tolerance for replacement geometry.

Specify edge tolerance
To be more consistent with the surfaces in the model, it may be the case that the distance changes in the surface geometry are more important than the exact locations of the edges. If this option is selected, a more lenient tolerance for edge geometry can be specified for Edge repair tolerance.

Edge repair tolerance
This is an optional tolerance to permit greater latitude in repairing edges.

Convert surfaces to blend surfaces
The surfaces are cleaned by attempting to simplify to blends.

Constrain surface normals along smooth edges
The tool will attempt to ensure that smooth edges will remain smooth[19] since the maximum deviation between the normals for these faces will be equal to the surface normal tolerance.

Surface normal tolerance (degrees)
The angular tolerance for constraining surface normals in degrees.

2.11.5 Repair Edges

The *Repair edges* tool attempts to repair inaccuracies in the edges of a sheet or a solid body.

The tool pairs tolerant geometry by recalculating edge and vertex geometry to a specified tolerance wherever possible. It also ensures that edges, designed to be tangential, are tangential within a specified tolerance. The tool will attempt to remove any mergeable edges or vertices in the geometry part (this option is disabled by clearing the **Merge edges** check box).

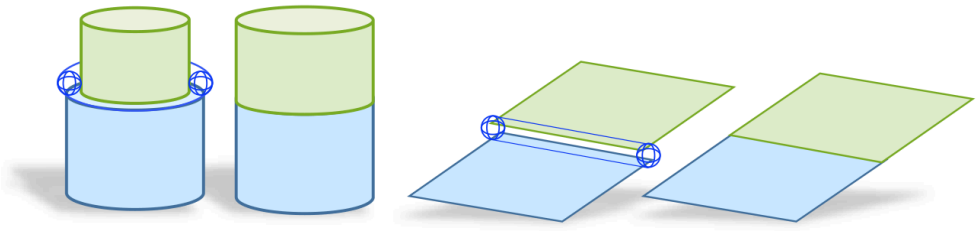


Figure 99: Examples of situations where edges can be repaired. The blue spheres represent the tolerance, within which the differing edges are meant to be considered as the same edge. The first and third image show the input geometry, while the second and last images indicate the edges after they were repaired.

19. Edges between faces where there is a smooth transition from face to face.

On the **Transform** tab in the **Repair** group, click the **Repair Edges** icon.

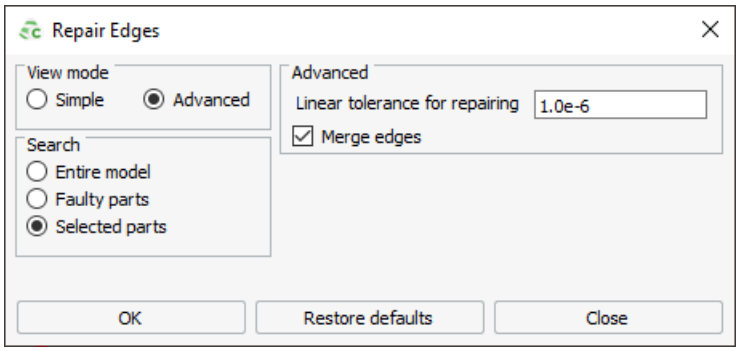


*Figure 100: The **Repair Edges** dialog, **Advanced** tab.*

Advanced settings

Linear tolerance for repairing
: The linear tolerance used for repairing.

Merge edges
: Any redundant edges or vertices are removed.

2.11.6 Repair and Sew Faces

The *Repair and Sew Faces* tool attempts to repair any problems in the faces of the part and then tries to sew them into a solid or sheet part.

The tool will perform the following actions:

- heal the input faces
- pre-process the faces for sewing by identifying and removing those that are invalid due to bad trimming curves
- identify and remove sliver faces from the part
- sew the faces
- post-process the resulting part by sewing up remaining thin gashes
- construct new faces to fill any holes caused by missing geometry that were removed during the pre-processing stage.

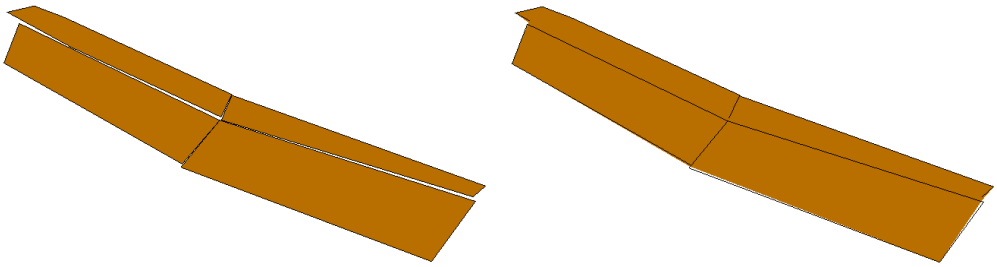


Figure 101: A part with several faults where the edges and vertices do not align (left) and after the Repair and sew faces tool was used on the model (right).

On the **Transform** tab in the **Repair** group, click the **Repair and Sew Faces** icon.

After applying the repair and sew faces tool to a part, the part is renamed to the default **RepairAndSewFaces1** and the repair and sew faces icon is displayed next to the part in the model tree.

Figure 102: A part displayed in the model tree for which the repair and sew faces tool was used.

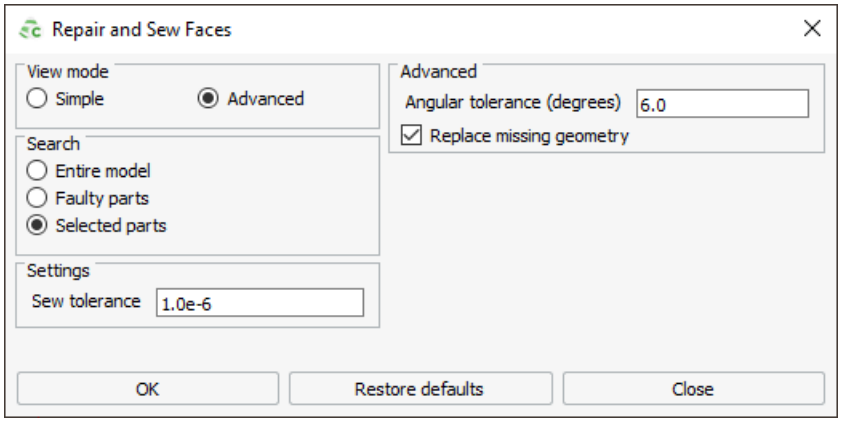


*Figure 103: The **Repair and Sew Faces** dialog, **Advanced** tab.*

Sew tolerance
: The supplied tolerance is used as the tolerance when sewing the sheets.

Advanced settings

Angular tolerance (degrees)
: The tangent change angle in degrees above which G1discontinuities will be removed by splitting rather than smoothing. (A surface can be composed of several NURBS surfaces or "patches". These patches should be fitted together in a way that the boundaries are invisible. This is mathematically expressed by the concept of geometric continuity. One of the options to establish

geometric continuity is by means of Tangential continuity (G1).It requires the end vectors of two curves or surfaces to be parallel which eliminates sharp edges).

Replace missing geometry
The tool attempts to generate surface geometry for faces that will cap holes in the resulting body. If the resultant body has closed circuits of laminar edges that appear to bound a missing face, the tool attempts to generate a surface to span the gap (bounded by the edges) and make a capping face from it.

2.11.7 Remove Small Features

The *Remove small features* tool attempts to remove small features, such as edges, faces, spikes and gashes.

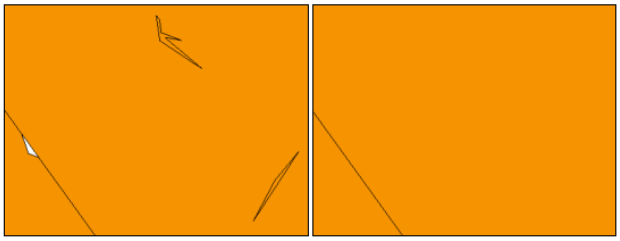


Figure 104: A model containing small entities (left) and the model after the small entities were removed (right).

On the **Transform** tab in the **Repair** group, click the **Remove Small Features** icon.

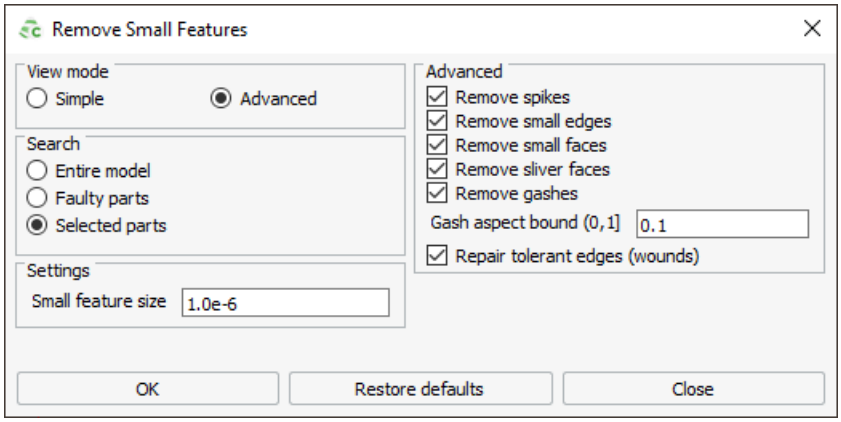


*Figure 105: The **Remove Small Features** dialog, **Advanced** tab.*

Small feature size
This field specifies the radius of a sphere drawn around a small face. If the face falls within the radius of the sphere, the face is removed. A gash will be removed if its width is less than the **Small feature size.**

Advanced settings

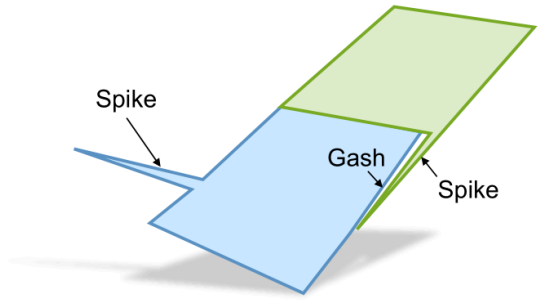
Remove spikes
A spike is a section of a face that has a high aspect ratio and small area. Spikes can lead to modelling failures. If this option is selected, spikes are removed from the geometry part.

Remove small edges
Small edges have a length less than specified by **Small feature size**. If this option is selected, small edges are removed.

Remove small faces
A small face is any face that fits within a sphere of a radius specified by Small feature size. If this option is selected, small faces are removed.

Remove sliver faces
Sliver faces have a high aspect ratio and small area. Removing unwanted sliver faces can simplify a body and lead to more reliable downstream operations. If this option is selected, sliver faces are removed. **Small feature size** for sliver faces is defined as the tolerance which is the width of the sliver face.

Remove gashes
Gashes are similar to spikes. They also have a high aspect ratio and small area. Gashes are always located between at least two faces. If this option is selected, gashes are removed. The **Small feature size** for gashes is the maximum width of any gash to be removed.



Gash aspect bound (0,1]
The maximum width to length ratio of any gash that is to be removed. Any gashes with an aspect ratio larger than this value are not removed from the body.

Repair tolerant edges (wounds)
If this option is selected, the tool attempts to heal tolerant edges. These edges are created during the removal of narrow features such as sliver faces, spikes and gashes.

2.11.8 Fill Hole Tool

The *Fill hole* tool attempts to fill a hole based on the currently selected laminar or free edges (wires).

Note: An edge is laminar when the edge represents (part of) the boundary of a single face.

On the **Transform** tab in the **Rebuild** group, click the **Fill Hole** icon.

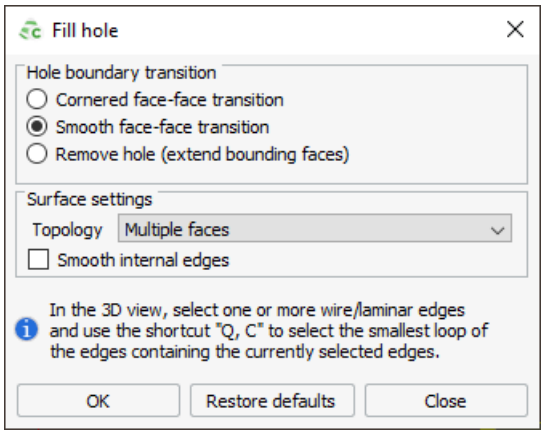


*Figure 106: The **Fill Hole** dialog.*

Tip:

- Select one or more wire/laminar edges in the 3D view.
- Press Q+C (Q followed by C) to automatically select the smallest loop of edges containing the currently selected edges.
- Click **OK** to activate the tool.

The following hole filling (hole boundary transition) options are available:

Cornered face-face transition
: The hole is filled while ignoring all smoothness requirements at the boundary. The sheet is analytic if possible.

Smooth face-face transition
: The hole is filled with a sheet that is smooth at the boundary. The sheet is analytic if possible.

Remove hole (extend bounding faces)
: The tool attempts to grow neighbouring faces to fill the hole, without creating additional faces.

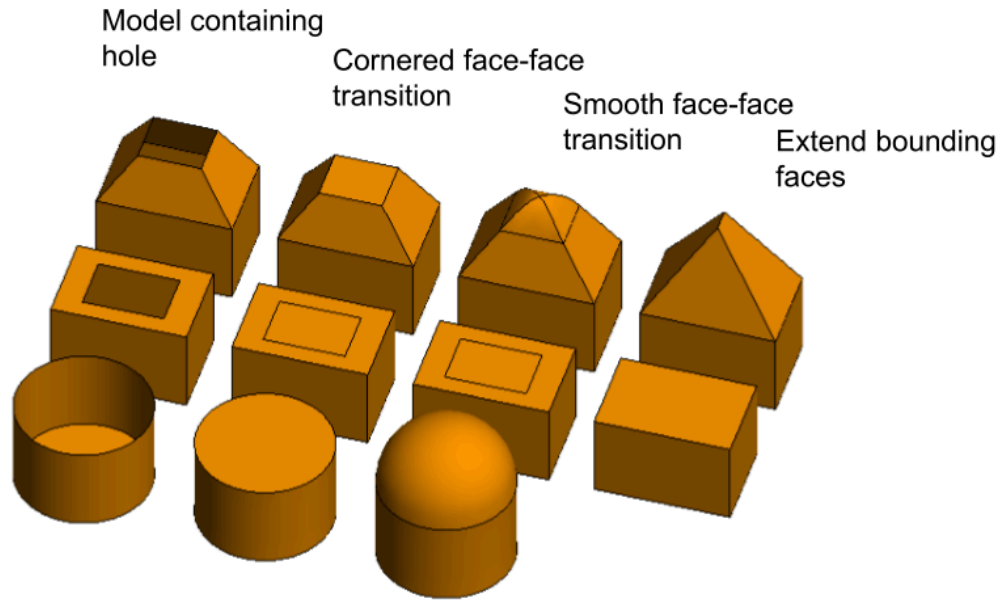


Figure 107: Example showing the result of hole filling.

The following topology surface settings are available for the patch filling the hole:

Minimum number of faces
The tool attempts to minimise the number of faces in the patch.

Multiple faces
The tool creates a patch if it is possible. The patch may contain multiple faces.

Single face
The tool attempts to fill the hole with a single face patch. This option may result in performance improvements if a single face solution is required, but will not work in all cases.

Smooth internal edges
If this option is selected and the **Topology** is set to **Multiple faces**, the internal edges of the faces used to fill the hole will be smooth without discontinuities.

2.12 Repairing Mesh Parts

Mesh parts can be manipulated in CADFEKO. Basic mesh editing and fixing capabilities allow triangles to be added, removed and mesh parts to be merged.

2.12.1 Creating a Mesh Triangle

When the mesh contains holes or faulty triangles were deleted, you can add a mesh triangle manually to the mesh.

1. Select the model mesh using one of the following workflows:
   - In the 3D view, select the relevant mesh part.
   - In the model tree, select the relevant mesh part.

   Note: The selection must be a single mesh part.

2. On the **Mesh** tab, in the **Repair** group, click the **Create Triangle** icon.

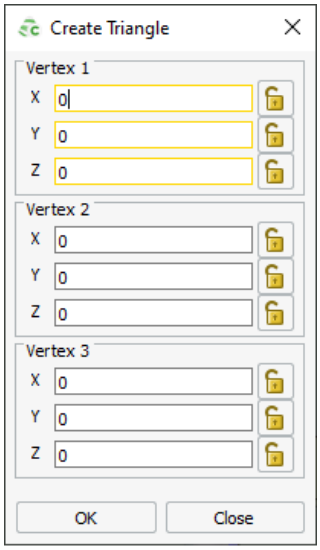


*Figure 108: The **Create Triangle** dialog.*

3. Specify the triangle vertices using point-entry.
4. Click **OK** to create the mesh triangle and to close the dialog.
   The mesh triangle is added to the mesh part.

2.12.2 Merging Meshes (Union for Meshes)

When mesh parts are imported as separate mesh parts, merge the separate mesh parts to ensure electrical connectivity or to add an edge port between the mesh parts.

1. Select the multiple model meshes using one of the following workflows:
   - In the 3D view, select the relevant mesh parts.
   - In the model tree, select the relevant mesh parts.
2. On the **Mesh** tab, in the **Repair** group, click the **Merge Meshes** icon.

2.12.3 Regrouping Mesh Part Labels

Split or regroup a selected mesh part label based on boundary edges.

1. Select the relevant mesh part in the details tree.
2. On the **Mesh** tab, in the **Repair** group, click the **Regroup** icon.

2.12.4 Removing Collapsed Mesh Elements

A collapsed mesh element is a degenerate triangle where two or mode vertices coincide.

1. Select the model mesh using one of the following workflows:
   - In the 3D view, select the relevant mesh part.
   - In the model tree, select the relevant mesh part.
2. On the **Mesh** tab, in the **Repair** group, click the **Remove Collapsed** icon.

2.12.5 Removing Mesh Duplicates

When using imported meshed or you have edited the mesh manually, it can often occur that the mesh contains duplicate triangles.

1. Select the model mesh using one of the following workflows:
   - In the 3D view, select the relevant mesh part.
   - In the model tree, select the relevant mesh part.
2. On the **Mesh** tab, in the **Repair** group, click the **Remove Duplicates** icon.

   If duplicate elements exist in the model, CADFEKO deletes all but one.

   When deleting duplicate elements from separate faces, the face with the non-PEC face medium is regarded as the original. The duplicate elements from the face with a default/PEC face medium is removed.

   View the number of duplicate mesh faces removed in the Notification centre.

2.12.6 Merging Vertices

Mesh connectivity relies on vertices of adjacent mesh elements being within a small tolerance of each other.

1. Select the model mesh using one of the following workflows:
   - In the 3D view, select the relevant mesh part.
   - In the model tree, select the relevant mesh part.
2. On the **Mesh** tab, in the **Repair** group, click the **Merge Vertices** icon.

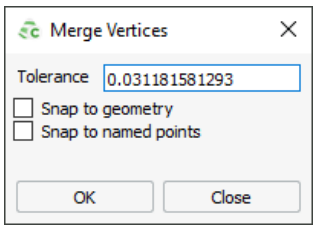


*Figure 109: The **Merge mesh vertices** dialog.*

3. In the **Tolerance** field, specify a value for the tolerance. Any two points separated by less than this distance are merged to the coordinates of one of the original vertices.
4. [Optional] Select the **Snap to geometry** check box to allow mesh vertices to snap to geometry points lying within the specified tolerance.
5. [Optional] Select the **Snap to named points** check box to allow mesh vertices to snap to named points lying within the specified tolerance.

   For example, if a named point lies between two mesh vertices that are less than the specified tolerance away from one another, the mesh vertices are merged to the named point.
6. Click the **OK** button to merge the vertices and to close the dialog.

2.13 Importing Models into CADFEKO

Import a CAD (geometry) model or mesh model from a wide range of industry formats into CADFEKO to save time and development costs.

2.13.1 Importing CADFEKO Model Files (.CFX)

Import an existing CADFEKO model (`.cfx` file) into a CADFEKO model.

1. On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list select the **CADFEKO Model (*.cfx)** icon.
2. Select the `.cfx` file you want to import.
3. Under **Import**, select the entities to import (for example, geometry, meshes, meshing rules, cable definitions, solution entities and optimisation searches).

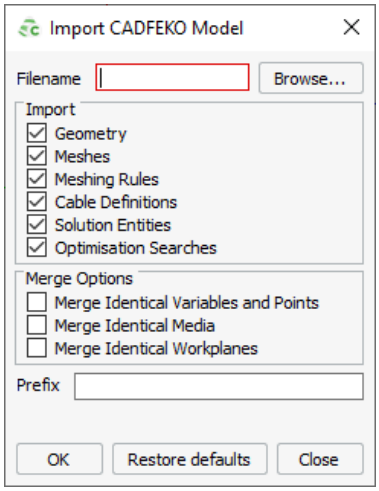


*Figure 110: The **Import CADFEKO Model** dialog.*

> **Note:** Frequency, infinite planes and mesh settings are not considered during the import process and will not affect the destination model in any way.

4. [Optional] Merge identical variables and media in the imported model and target model.
   a) To merge identical variables and points, under **Merge options**, select the **Merge identical variables and points** check box.
   b) To merge identical media, under **Merge options**, select the **Merge identical media** check box.
   c) To merge identical media, under **Merge options**, select the **Merge identical workplanes** check box.
5. [Optional] If there are naming conflicts between the names of the imported entities and the existing entities in the model, in the **Prefix** field, enter a prefix that will be pre-pended to all imported entity names.

6. Click **OK** to import the file and to close the dialog.

2.13.2 PCB Formats for Import

View the supported ECAD file formats for import into CADFEKO.

> **Note:** The ECAD file formats are only supported on Microsoft Windows in ASCII format.

The following file formats are supported:

- Altium
  - Designer (`.pcbdoc`)
  - PCAD (`.pcb`)
- AutoDesk - Eagle (`.brd`)
- Cadence
  - Allegro (folder)
  - Specctra/OrCAD Layout (`.dsn`)
- CADVANCE (folder)
- CAM350 (`.cam`)
- IPC2581 (`.xml`)
- Mentor Graphics
  - Board Station (folder)
  - Neutral (folder)
  - Xpedition (folder)
  - PADs (`.asc`)
- ODB++ (`.tgz`, `.tar.gz`)
- Zuken
  - CADSTAR (`.cpa`)
  - CR5000/CR8000 (`.pcf`, `.pnf`, `.ftf`)
  - CR5000 PWS (`.bsf`, `.ccf`, `.mdf`, `.udf`, `.wdf`)
- PEMA (Altair) (`.pema`)

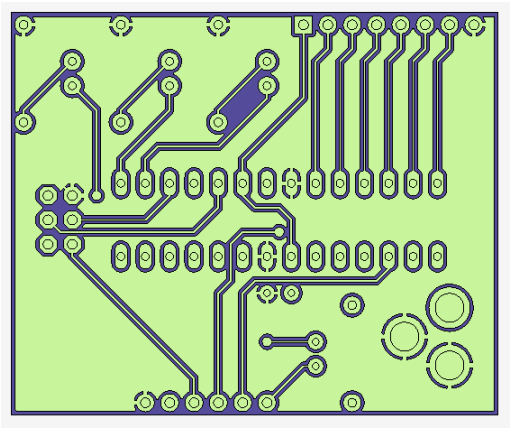


Figure 111: An example of a PCB import.

Importing a PCB

Use the PCB import tool to import all major ECAD file formats.

1. On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list select the **Import PCB File** icon.

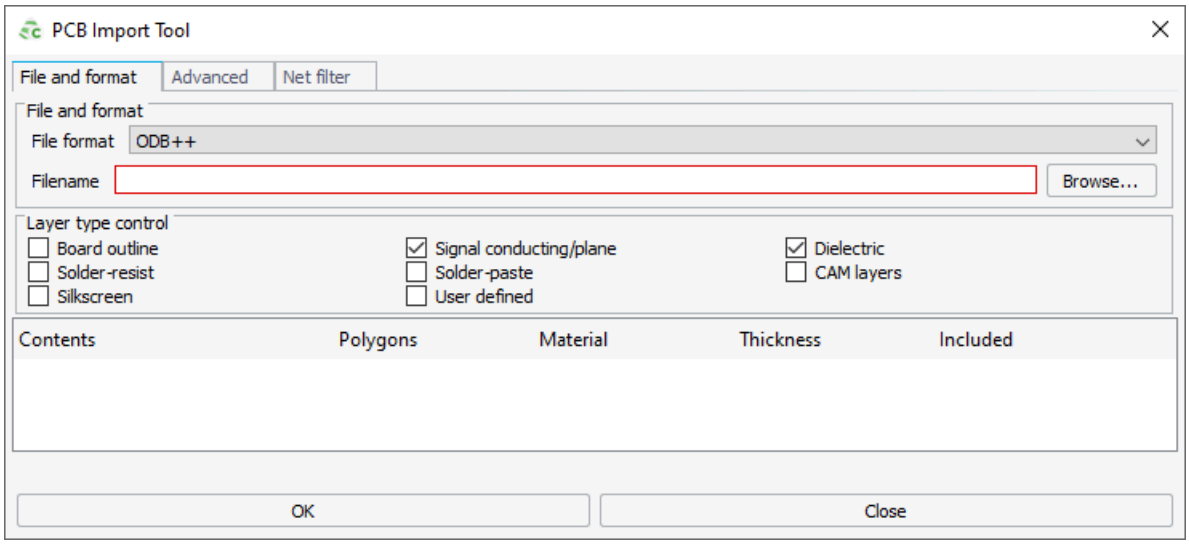


*Figure 112: The **PCB Import Tool** dialog.*

2. In the **File format** drop-down list, select the type of file to import.
3. Browse to the location of the file to import.
4. [Optional] Under **Layer type control**, select the applicable options:
   - *Board outline*
     Include the layer(s) that contains the PCB outline including cutouts/holes that are cut out during PCB fabrication using a routing bit.

- *Signal conducting/plane*
  Include all metal layer(s) such as signal traces, ground and power planes and metal fills.

- *Dielectric*
  Include the dielectric layer(s).

- *Solder-resist*
  Include the layer(s) that contains the insulating coating which covers the circuit pattern.

- *Solder-paste*
  Include the layer(s) that define the solder paste mask.

- *Silkscreen*
  Include the layer(s) that contains annotations (such as letters numbers and symbols as well as component footprints).

- *User defined*
  Include all user defined layer(s).

- *CAM layers*
  Include the CAM layers.

  Note: This check box will be automatically checked and hidden for **Gerber (RS-274D/274X)** import.

Note: Any media defined in the PCB file and referenced in the layers are available in the model tree under **Media**. Media without labels are given the label of the layer. For example, if the solder paste layer contains a dielectric without a label, it is given the label **SolderPaste**.

5. Click **OK** to import the PCB model and to close the dialog.

Related tasks
Changing the Rendering Speed for a Model

Advanced PCB Import Options

View the advanced PCB options available for import (depending on the type of file to be imported).

Use infinitely thin layers
This option reduces the layers of a PCB with a finite thickness to infinitely thin layers.

Import vias
This option imports the vias defined in the PCB file and adds as wires between the PCB layers.

Union resultant geometry
This option unions the imported PCB model.

Simplify resultant geometry
This option simplifies the imported PCB model.

Heal and simplify internal representation of PEMA data
When using this option, simplification (removal of redundant faces and edges) as well as tolerance-related corrections are applied during the conversion from the PCB data. In some cases this may result in desired information (such as shapes defined within a metal fill) being removed from the layout. Re-do the import without this option to retain this information.

Scale by
Specify a value by which the PCB model must be scaled during import. CADFEKO will suggest a default value based on the CADFEKO model unit and the unit of the PCB file that is to be imported. This suggested value can be changed as needed.

Importing Subsets of Electrical Connections

Use the **Net filter** tab to selectively import collections of PCB electrical connections.

> **Note:** The net filter only applies to the ODB++ import.

1. On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list select the **Import PCB File** icon.
2. On the **PCB Import Tool** dialog, on the **File and format** tab, in the **File format** drop-down list, select **ODB++**.
3. Select the **Net filter** tab.

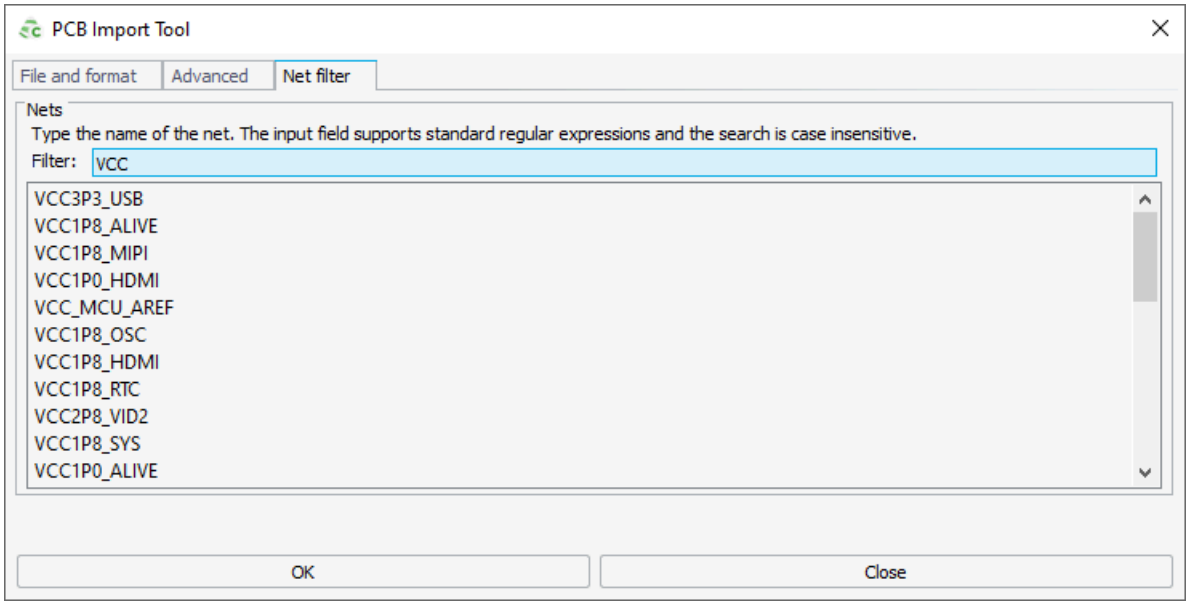


*Figure 113: The **PCB Import Tool** dialog, **Net filter** tab.*

4. In the **Filter** field, type the label of a net or a part thereof.

> **Note:** Populating the filtered list of available nets could take some time.

5. Click **OK** to import the PCB model and to close the dialog.

2.13.3 Geometry (CAD) Formats for Import

View the supported geometry formats that can be imported into CADFEKO and the supported versions.

CADFEKO is based on the Parasolid solid modelling kernel that allows models to be imported and exported from and to the native Parasolid format without any translation.

Since all imported CAD models are converted to a Parasolid format during the import process, importing from other CAD formats may cause unexpected results. Differences in the internal representation used by various CAD formats may cause adjoining surfaces not to line up correctly. This discrepancy is due to tolerance differences. Models that use a numerical representation can cause faults during scaling.

The following geometry (CAD) formats are supported for import:

Formats	File Format	Supported Versions
ACIS	`.sat`	R1 – 2022 1.0
AutoCAD	`.dxf`	2.5 – 2023
CATIA V4	`.model`, `.session`, `.exp`	4.1.9 to 4.2.4
CATIA V5	`.CATPart`, `.CATProduct`, `.CATShape`	V5 R8 to V5–6 R2022
IGES	`.iges`, `.igs`	Up to 5.3
Parasolid	`.x_t`, `.x_b`, `.xmt_txt`	9.0 - 36.1.212
Pro / Engineer	`.prt`, `.asm`	16 to Creo 9.0
STEP	`.step`, `.stp`	AP203, AP214, AP242
Unigraphics and NX	`.prt`	11 - NX 2206

AutoCAD

Supported entities are:

- 3D face
- Arc
- Circle
- Ellipse

- Line
- Polyline
- Polyface mesh (3D)

Unsupported entities are:

- Point
- Spline
- 3D Solid
- Trace
- Dimensional annotations

Parasolid

Isolated vertices (acorns) are not imported.[20] The coordinates are written to the message window and can be created manually should they be required.

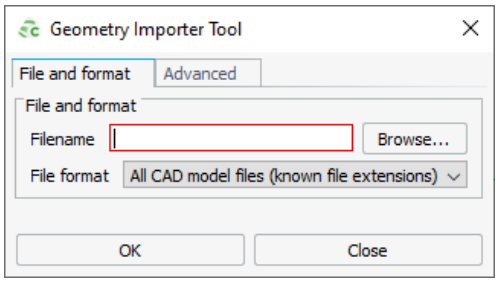
Importing CAD (Geometry)

Import a CAD (geometry) model into CADFEKO.

Note:

- Best results are obtained during importing if the CADFEKO model unit is in "m".
- If a large model is imported and the source file unit is different to the CADFEKO model unit, the import process may be slow.

1. On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list select the **Geometry** icon.
2. Click the **File and Format** tab.
   a) In the **Filename** field, browse for the file you want to import.



Specify the advanced settings for the geometry import.

3. [Optional] Select the **Advanced** tab. Specify the relevant advanced import settings.
4. Click **Import** to import the geometry and close the dialog.

20. These are not the same as named points.

Advanced CAD Import Options

View the advanced geometry options available for import (depending on the type of file to be imported).

Healing options

This option controls the healing of data containing performance-expensive errors.

- No healing
  - No healing is applied to the imported file.
- Standard healing / Advanced healing
  - Geometrical and topological irregularities are repaired and healed. The translated file is scanned for corrupted data and invalid data is fixed.

    If the imported model contains face-face inconsistencies, it will cause multiple separate parts to be created during the healing process. After importing, use the union or stitch operation to combine the parts into a single part.

    Note: The advanced healing option adds a few more time consuming and extensive healing operations to the conversion process.

Simplify model

This option controls the process of cleaning and removing redundant topologies and geometries from the model during translation. If a vertex is redundant, the vertex is deleted and the associated edges are merged. If an edge is redundant, the edge is removed and the associated faces merged.

Stitch trimmed faces

This option controls the stitching of trimmed[21] faces during the translation process.

Use two step import process

Some models may not import correctly with the current import process. Use the two-step import process that makes use of the older, legacy import process to attempt to import problematic models.

Extrude

Enables the extrusion option (only for `.dxf` file imports).

Auto-stitch faces

Faces that touch are automatically stitched (only for `.dxf` file imports).

Auto-merge wires

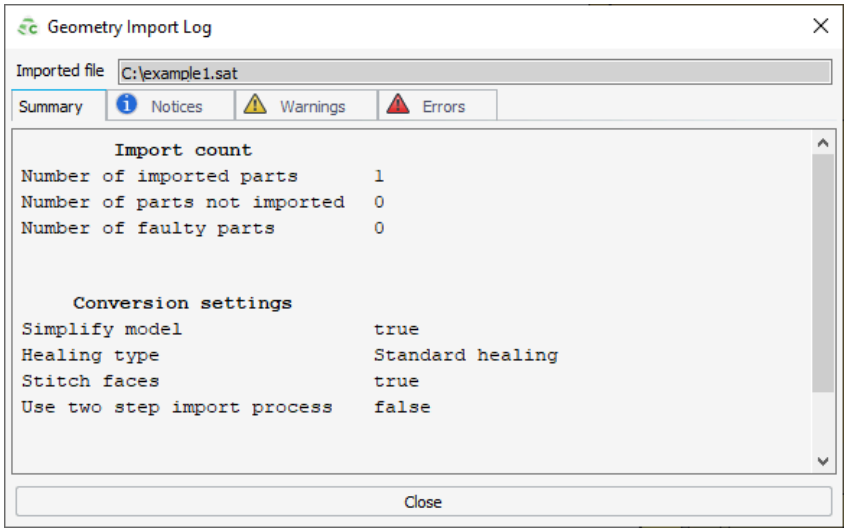
Wires that touch are automatically stitched (only for `.dxf` file imports).

21. A trimmed surface is a surface which was divided into multiple pieces as a result of a modelling operation. A portion of the surface may no longer be required to support the model topology. The redundant pieces are then discarded.

Viewing the Geometry Import Log

View the log file for a summary of the last geometry import. This information is useful in cases where the import conversion fails.

1. On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list select the **Geometry Import Log** icon.
2. Click the **Warning** tab or **Errors** tab to view any errors in the import process (when applicable).



> **Troubleshooting:** View the model format conversions performed during a geometry import in the log file located at `%FEKO_USER_HOME%/logs/CADimport.*.log`[22].

3. Click **Close** to close the dialog.

2.13.4 Mesh Formats for Import

View the supported mesh formats for import.

CADFEKO imports most of the mesh formats[23] by running PREFEKO and importing the resulting `.fek` file. Since these formats do not support specifying dielectric media, all segments, triangles and polygonal plates are imported as PEC structures in free space. Tetrahedra obtain the medium *Unknown*.

22. This directory is used to write user specific initialisation files. It is provided to allow different users to save unique configurations, and for situations where the user does not have write access to the Feko directory. For Microsoft Windows systems this is normally %APPDATA%\feko\xx.yy and on UNIX systems it is usually set to $HOME/.feko/xx.yy during the installation. Here xx.yy represent the major and minor version numbers.
23. Except for `.fek` file, .raw file and .txt file imports

When importing `.fek` files, only the mesh parts (wire segments, triangles, polygonal plates and tetrahedra) are imported. Information regarding the solution configuration is completely ignored. Medium information and segment radii are retained during import.

The following mesh formats are supported for import:

Formats	File Format
Feko model	`.fek`
CADFEKO mesh	`.cfm`
Feko HyperMesh	`.fhm`
Femap neutral	`.neu`
NASTRAN	`.nas`
AutoCAD	`.dxf`
STL	`.stl`
PATRAN	`.pat`
ANSYS	`.cdb`
CONCEPT	`.dat`
ABAQUS	`.inp`
ASCII	
GiD	`.msh`
NEC data	`.nec`
I-DEAS universal format	`.unv`
Voxel	`.raw`, `.txt`

Femap Neutral Mesh
Boundary surfaces, bordered with line curves, are imported as polygonal plates.

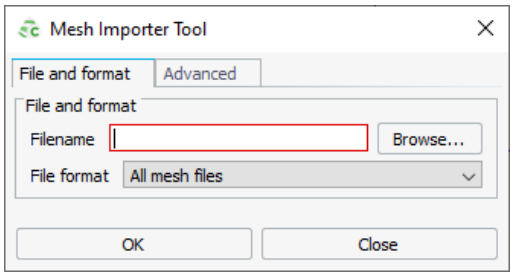
AutoCAD
Only *LINE* and *POLYLINE* structures, which define segments and triangles, are supported.

GiD
Hexahedral elements are ignored.

Importing a Mesh

Import a mesh model into CADFEKO.

1. On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list select the **Mesh** icon.
2. Select the **File and Format** tab.
   a) In the **Filename** field, browse for the file you want to import.

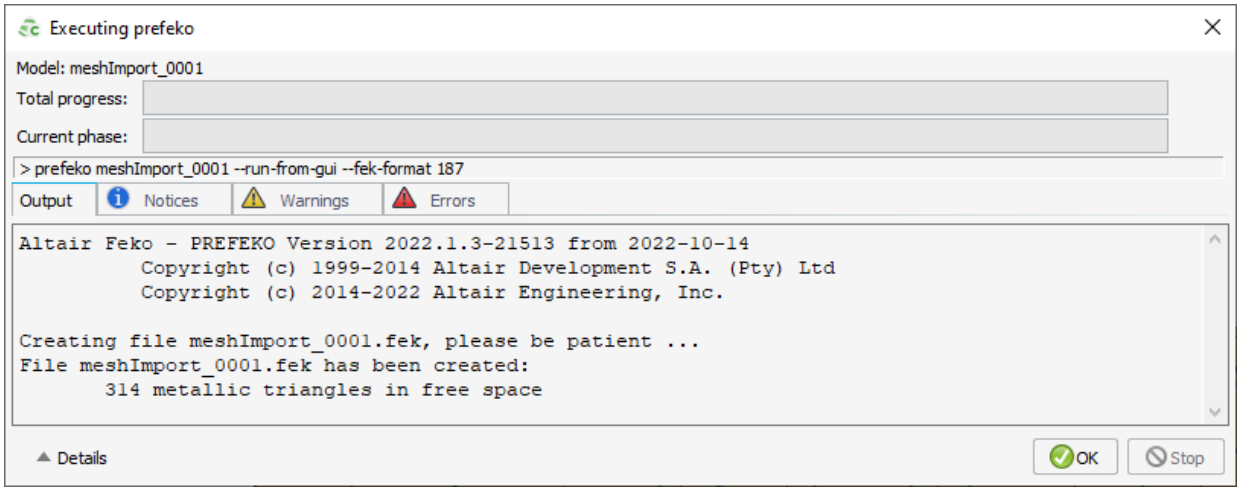


Specify the advanced settings for the mesh import.

3. [Optional] Select the **Advanced** tab. Specify the relevant advanced import settings.
4. Click **Import** to import the mesh model and close the dialog.

> **Note:** When importing a `.fhm` file and an `.inc` file exists in the same folder with the exact file name, then the media definitions are imported from the `.inc` file.

PREFEKO imports the resulting `.fek` file[24].



5. The dialog is closed if there are no errors or warnings.

   If there are errors or warnings given, click the **Warnings** tab or **Errors** tab to view the relevant information.

24. PREFEKO is not run for `.fek` file, .raw file and .txt file imports

Advanced Mesh Import Options

View the supported mesh import options (depending on the type of file to be imported).

Import segments, triangles, tetrahedra, polygons, cylinders or quadrangles
Select the elements to import.

> **Note:** Quadrangles are divided into triangles.

Mesh conversion
In cases where the resulting mesh is in a different format than the source mesh, the mesh conversion option can be specified to determine the output mesh format. This option is available for voxel mesh imports.

Merge identical media
For mesh formats where materials are specified, this option provides functionality to merge media with identical properties. An optional prefix can be provided to append the imported material label.

Group into separate parts (using labels)

Import meshes into separate parts. During the import process, faces are grouped into parts as they were at the time of export. Imported meshes without labels are grouped as *UnknownMeshParts* parts in the model tree.

To import the mesh as a single mesh part with label *MeshImport*, unselect the **Group into separate parts (using labels)** check box.

Default wire radius
Only ANSYS files support segment radius information. For all other formats and ANSYS files where the segment radius is not specified, a default radius must be specified.

Scale factor to metres
A scale factor can be specified if the unit of the imported mesh is not in metres.

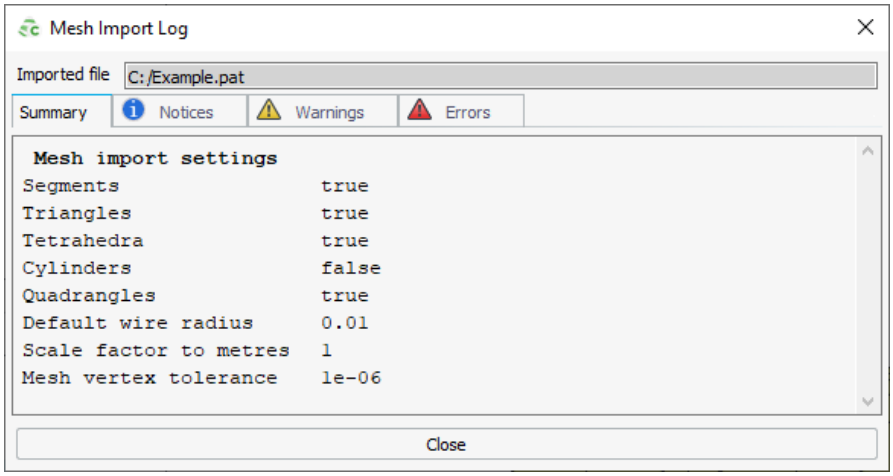
Segment length
For meshed AutoCAD DXF files, the *LINE* elements are divided into segments according to the value of the **Segment length**. If the *LINE* elements may not be sub-divided, this value must be larger than the longest line. This option is only available for `.dxf` mesh imports.

Mesh vertex tolerance
The mesh vertex tolerance is specified. If the tolerance is small, Feko will interpret the vertices as connected. Usually, the default setting should suffice.

Viewing the Mesh Import Log

View the log file for a summary of the last mesh import. This information is useful in cases where the import conversion fails.

1. On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list select the **Mesh** icon.
2. Click the **Warning** tab or **Errors** tab to view any errors in the import process (when applicable).



3. Click **Close** to close the dialog.

2.14 Exporting Models from CADFEKO

A CADFEKO model can be exported to a variety of industry standard geometry and mesh formats to be used in other applications.

2.14.1 CAD (Geometry) Formats for Export

The geometry can be exported to a number of industry-standard CAD formats.

The following CAD formats are supported for export:

Formats	File Format
ACIS	`.sat`
CATIA V4	`.model`, `.session`, `.exp`
CATIA V5	`.CATPart`, `.CATProduct`, `.CATShape`
IGES	`.iges`, `.igs`
Parasolid	`.x_t`, `.x_b`
STEP	`.STEP`, `.stp`

Scaling is often the source of many importing errors when translating between CAD file formats (ACIS, CATIA, IGES, Pro Engineer, STEP, Unigraphics / NX, Parasolid).

CADFEKO does not perform any scaling during the export (scaling could cause tolerance errors during the subsequent import process). Change the scaling by modifying the CADFEKO model unit or model extents. If a model does not import as expected, change the scaling to import the geometry correctly.

Parasolid Models

Parasolid models are inherently limited to a 1000x1000x1000 unit box centred at the origin. CADFEKO introduces a scaling factor to make this more flexible. The **Scale factor** is the factor by which the CADFEKO model must be scaled during export to convert it to correct units required in the Parasolid model. A scale factor of 0.1 implies that the dimensions of the saved Parasolid model are one-tenth of the native dimensions as set in CADFEKO.

Typically, programs that import Parasolid models allow specifying a factor by which the Parasolid model must be scaled during the import. To maintain the correct units and scale, this factor should then be the inverse of the scale factor used in the export of the model from CADFEKO.

For large models (larger than 500 of the current CADFEKO units), the extents must be increased.

For smaller models (less than 50 CADFEKO units), the extents should be decreased.

In general, changing the model extents is not recommended (unless the model is very small and precision or geometric accuracy problems are encountered). Using the default extents results in an unscaled Parasolid model, and it is not necessary to keep track of the scale factor during model import / export.

Exporting CAD to Parasolid Format

Export the geometry in Parasolid CAD format.

> **Note:** Only the final geometry is exported. The full creation history is lost, similar to creating a primitive.

1. On the **Home** tab, in the **File** group, click the **Export** icon. From the drop-down list select the **Geometry** icon. From the drop-down list select **Parasolid (*.x_t / *.x_b)**.

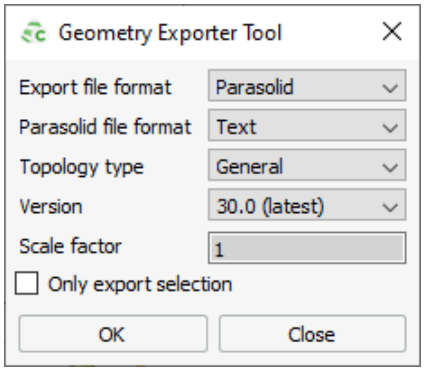


*Figure 114: The **Export dialog model** dialog.*

2. In the **Parasolid file format** drop-down list, select one of the following:
   - To export the Parasolid model in text format, select **Text**.
   - To export the Parasolid mode in binary format, select **Binary**.
3. In the **Topology type** drop-down list, select one of the following:
   - *Manifold*
     A manifold body is any body that can exist in the real world or could be manufactured. Wire bodies must be one-dimensional open (linear sections with two endpoints) or closed (loops with no endpoints); they may not contain junctions. Sheet bodies must be two-dimensional open or closed and may not contain junctions.
   - *General*
     General bodies differ from manifold bodies in that they usually cannot exist in the real world. They are often idealized representations of bodies, for example, infinitely thin sheets joined in a T-junction. Bodies of mixed dimensions are also general, for example, a body with wires, sheets and solids.
4. In the **Version** field, from the drop-down list select a version between 16 and 30 (latest).
5. [Optional] To export only the selected geometry, click the **Only export selection** check box.

6. Click the **OK** to export the geometry and to close the dialog.

2.14.2 Mesh Formats for Export

The model mesh or simulation mesh can be exported to a variety of industry-standard mesh formats.

The following mesh formats are supported for export:

Formats	File Format
CADFEKO mesh	`.cfm`
Feko HyperMesh	`.fhm`
NASTRAN	`.nas`
STL	`.stl`
Gerber	`.gbr`
I-DEAS mesh	`.unv`
DXF	`.dxf`

Exporting a Mesh

Export the model mesh or simulation mesh.

1. [Optional] Select the geometry parts or mesh parts to export. If no part is selected, all included meshes of the specified type are exported.
2. On the **Home** tab, in the **File** group, click the **Export** icon. From the drop-down list select the **Mesh** icon.

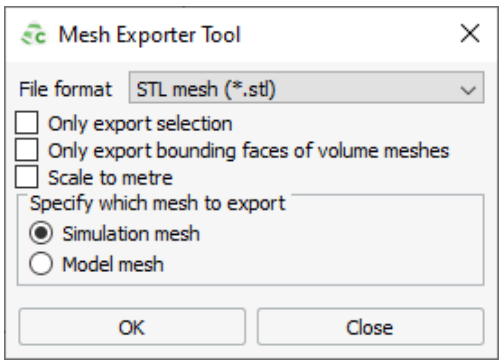


*Figure 115: The **Mesh Exporter Tool** dialog.*

If a geometry or mesh part was selected in Step 1, the **Only export selection** check box is selected.

3. [Optional] To export only the meshes associated with the selected parts, select the **Only export selection** check box.
4. [Optional] To export only the bounding faces of volume meshes (NASTRAN mesh, STL mesh, I-DEAS mesh), select the **Only export bounding faces of volume meshes** check box.
5. [Optional] To export dimensions in metre for CADFEKO mesh, Feko HyperMesh, NASTRAN mesh or STL mesh, select the **Scale to metre** check box.

If the model contains mesh parts (imported meshes) and simulation meshes (meshed geometry or remeshed mesh parts), the options under **Specify which mesh to export** are enabled.

6. Under **Specify which mesh to export**, select one of the following:
   - To export the meshed model of the geometry or the remeshed version of an imported mesh, click **Simulation mesh**.
   - To export an imported mesh, click **Model mesh**.
7. Click **OK** to export the mesh and to close the dialog.

> **CAUTION:** Exporting a `.fhm` file replaces the `.inc` file of the same name (if it exists).

Exporting Mesh to Gerber Format

A Gerber mesh is typically used in printed circuit board industry software, for example, to describe the printed circuit board images such as copper layers, solder mask and legends.

When exporting planar structures to Gerber format, all entities are projected onto the XY plane.

Each model entity is written to its own Gerber layer with its layer name equal to the entity label.

> **Note:** This is an information layer and not a layer in the PCB sense.

If a model outline is required, use CADFEKO wires as the wires are exported as zero width wires.

> **Note:** The export of drilling information is not supported.

1. [Optional] Select the geometry parts or mesh parts to export. If no part is selected, all included parts are exported.
2. On the **Home** tab, in the **File** group, click the **Export** icon. From the drop-down list select the **Mesh** icon.

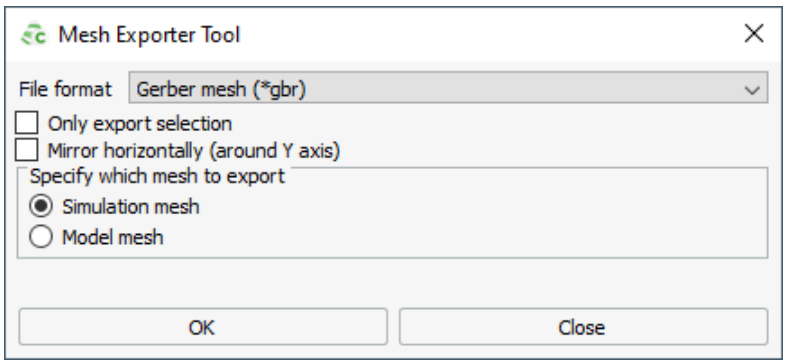


*Figure 116: The **Gerber export** dialog.*

3. From the **File format** drop-down list select **Gerber mesh (*.gbr)**.

If a geometry part or mesh part is selected, the **Only export selection** check box is shown.

4. [Optional] To export only the selected part(s), select the **Only export selection** check box.
5. [Optional] To export a mirrored version of the geometry, select the **Mirror horizontally (around Y axis)** check box.
6. Under **Specify which mesh to export**, select one of the following:
   - To export the meshed model of the geometry or the remeshed version of an imported mesh, click **Simulation mesh**.
   - To export an imported mesh, click **Model mesh**.
7. Click **OK** to export the mesh to Gerber format and to close the dialog.

Exporting Mesh Outline to a DXF File

Export the mesh outline to `.dxf` file as two-dimensional data by projecting onto the XY plane. Use the outline, for example, to create the printed circuit board (PCB) layout of your design.

1. [Optional] Select the geometry parts or mesh parts to export. If no part is selected, all meshes of the specified type are exported.
2. On the **Home** tab, in the **File** group, click the **Export** icon. From the drop-down list select the **Mesh** icon.

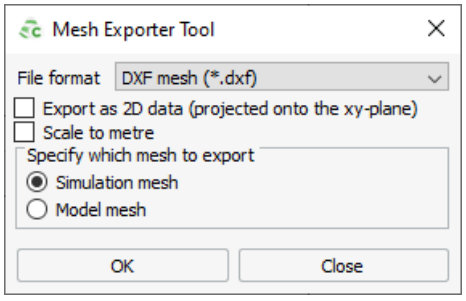


*Figure 117: The **Mesh Exporter Tool** dialog.*

If a geometry or mesh part was selected in Step 1, the **Only export selection** check box is selected.

3. [Optional] To export all included meshes, clear the **Only export selection** check box.
4. Select the **Export as 2D data (projected onto the xy-plane)** check box to export the model as two-dimensional data.
5. [Optional] To export dimensions in metre, select the **Scale to metre** check box.

If the model contains mesh parts (imported meshes) and simulation meshes (meshed geometry or remeshed mesh parts), the options under **Specify which mesh to export** are enabled.

6. Under **Specify which mesh to export**, select one of the following:
   - To export the meshed model of the geometry or the remeshed version of an imported mesh, click **Simulation mesh**.
   - To export an imported mesh, click **Model mesh**.
7. Click **OK** to export the mesh outline and to close the dialog.

2.15 Field/Current Data

Define field or current data using either far field data, near field data, spherical mode data or PCB current data. Use the field/current definition when defining an equivalent source or a receiving antenna.

The workflow for creating field/current data:

1. *Create a field/current definition*
   Create a field/current data definition by defining the field data manually or by importing the field/current data from a file. A range of file formats is supported.

2. *Define an equivalent source or receiving antenna*
   Use the field/current data definition to define an equivalent source or a receiving antenna.

Related concepts
Equivalent Sources
Ideal Receiving Antennas

2.15.1 Defining Far Field Data from File

Import far field data from a Feko (`.ffe`), external ASCII or a CST far field scan (`.ffs`) file to create a far field data definition. Use the far field data when defining an equivalent source or receiving antenna.

The far field must be defined in spherical coordinates.

1. On the **Construct** tab, in the **Define** group, click the **Field/Current Data** icon. From the drop-down list select **Define Far Field Data**.

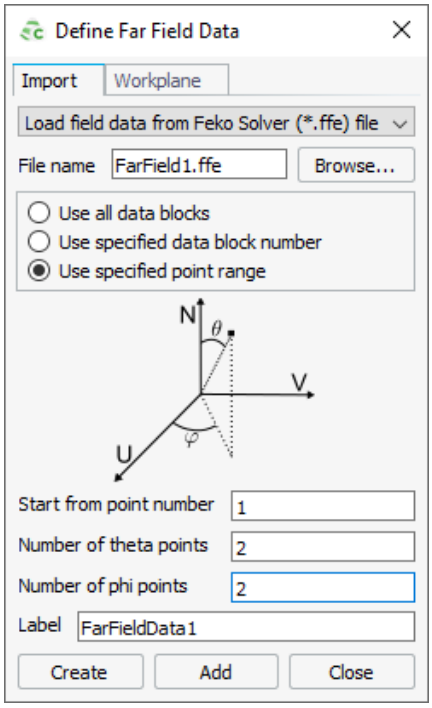


*Figure 118: The **Define Far Field Data** dialog.*

2. Select one of the following file types to import:
   - **Load field data from Feko Solver (*.ffe) file**
   - **Load field data from an external data file**
   - **Loaf field data from a CST far field scan (*.ffs)**
3. In the **File name** field, browse to the file location.
4. Select one of the following:
   - To select far field data from a multi-frequency `.ffe` or `.ffs` file, select **Use all data blocks**. The data is interpolated for use at the operating frequency.
   - To select far field data at a specific frequency in a `.ffe` or `.ffs` file, select **Use specified data block number** and enter the number of the relevant data block.
   - To select a specific far field pattern in a `.ffe` or `.dat` file, select **Use specified point range**[25].
     1. In the **Start from point number** field, specify the line number of the first line to read.
     2. In the **Number of theta points** field, specify the number of theta points used in the imported far field.
     3. In the **Number of phi points** field, specify the number of phi points used in the imported far field.
5. In the **Label** field, specify a unique label for the far field data.

25. Far field patterns are typically frequency-dependent and models with radiation pattern sources usually have only a single solution frequency. If the radiation pattern is calculated using a frequency sweep in Feko, the `.ffe` file contains multiple patterns.

6. Click **Create** to define the far field data and to close the dialog.

Related tasks
Adding a Far Field Source
Requesting Ideal Receiving Antenna (Far Field Pattern)

2.15.2 Defining Near Field Aperture from File

Import near field data from a `.efe` file and / or `.hfe` file to define a near field data aperture. Use the near field data aperture when defining an equivalent source or receiving antenna.

The `.efe` and `.hfe` files do not contain information regarding the coordinates system, frequency or number of points. As a result, you need to supply the above information to define the near field data aperture.

Important: Only a single surface definition is supported.

1. On the **Construct** tab, in the **Define** group, click the **Field/Current Data** icon. From the drop-down list select **Define Near Field Data**.

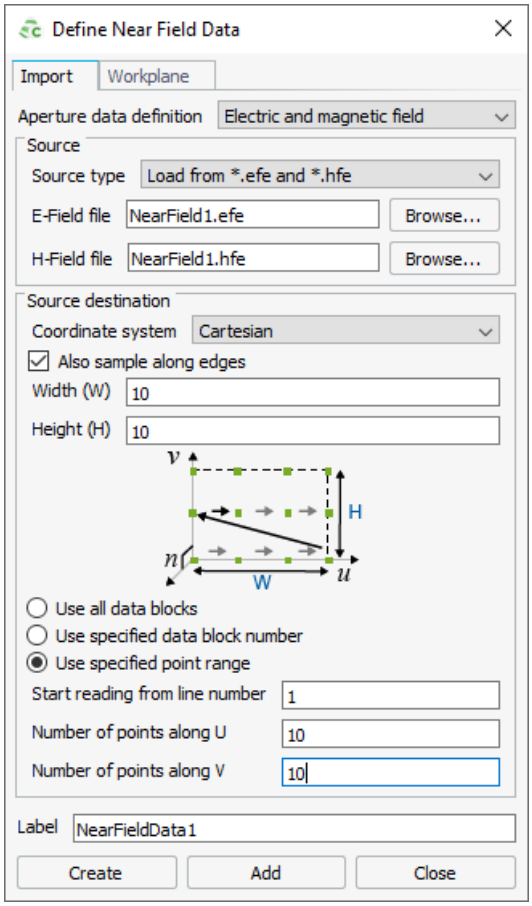


*Figure 119: The **Define Near Field Data** dialog.*

2. In the **Aperture data definition** drop-down list, select one of the following:
   - **Electric and Magnetic field**
   - **Electric field**
   - **Magnetic field**
3. In the **Source type** field, select one of the following:
   - **Load an ASCII text file**

     Note: The units are V/m for the E-field and A/m for the H-field.
   - **Load from *.hfe file**
4. In the **E-field file** field, browse to the E-field file location.
5. In the **H-field file** field, browse to the H-field file location
6. In the **Coordinate system** field, select one of the following:
   - **Cartesian**
   - **Cylindrical** (option only available when selecting **Electric and Magnetic field**)
   - **Spherical** (option only available when selecting **Electric and Magnetic field**)

The physical location of the sample points and how they relate to the defined aperture can be specified.

7. [Optional] Select the **Also sample along edges** check box to assume the outer sample points lie on the edges of the defined aperture.

   CAUTION: For multiple near field sources in a single model, sample points may not lie on any two aperture edges that share a common side. This results in two elementary dipoles with the same location and polarisation to be included, leading to incorrect results.
8. For options Cylindrical or Spherical, select the **Swap source and field validity regions** check box if the fields on the inside of the region are equivalent to the calculated field values.
9. In the **Width (W)** field, specify the aperture width.
10. In the **Height (H)** field, specify the aperture height.
11. Select one of the following:
    - To select near field data from multi-frequency `.efe` and `.hfe` files, select **Use all data blocks**. The data is interpolated for use at the operating frequency.
    - To select near field data at a specific frequency in `.efe` and `.hfe` files, select **Use specified data block number** and enter the number of the relevant data block.
    - To select a specific near field pattern in `.efe` and `.hfe` files, select **Use specified point range**[26].
      1. In the **Start reading from line number** field, specify the first line number to be read in the file.

26. Near field patterns are typically frequency-dependent and models with radiation pattern sources usually have only a single solution frequency. If the radiation pattern is calculated using a frequency sweep in Feko, the `.efe` and `.hfe` files contain multiple patterns.

> **Note:** Comment lines and empty lines are not counted.

For example, a file with 100 points per near field, the second block starts reading from line 101, regardless of any comment lines.

2. In the **Number of points along U** field, specify the number of points along the U axis.
3. In the **Number of points along V** field, specify the number of points along the V axis.

12. In the **Label** field, specify a unique label for the near field data.
13. Click **Create** to define the near field data and to close the dialog.

Related tasks

Adding a Near Field Source

Requesting Ideal Receiving Antenna (Near Field Pattern)

2.15.3 Defining Near Field Data from File

Import near field data from Feko field on a Cartesian boundary (`.efe` and / or `.hfe`), Sigrity input file (`.nfd`), MVG measurement file (`.mfxml`) or a CST near field scan (`.nfs`) to create a near field data definition. Use the near field data definition when defining an equivalent source or receiving antenna.

1. On the **Construct** tab, in the **Define** group, click the **Field/Current Data** icon. From the drop-down list select **Import Near Field Data From File**.

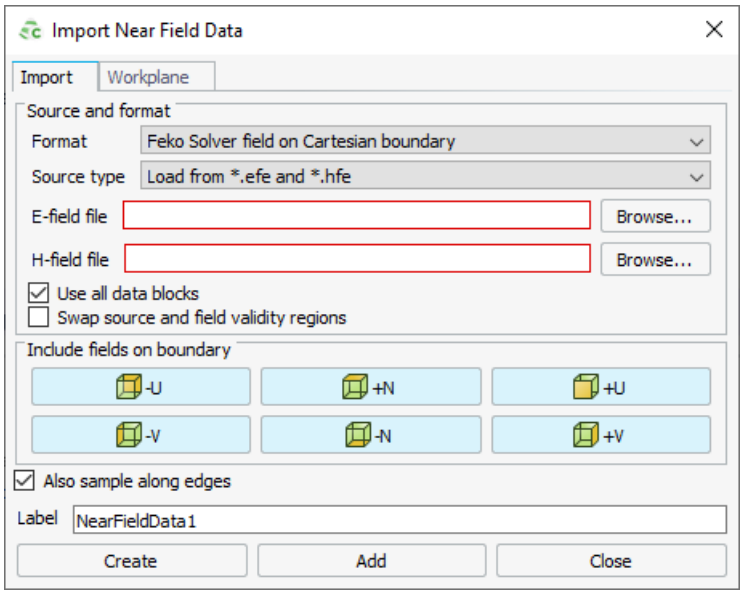


*Figure 120: The **Import Near Field Data** dialog.*

2. In the **Format** field, select one of the following:
   - **Feko Solver field on Cartesian boundary**
   - **Sigrity (*.nfd) input file**

- **MVG (*.mfxml) measurement file**
- **CST near field scan (CST NFS)**

3. For option **Feko Solver field on Cartesian boundary**, specify the following:
   a) In the **Source type** drop-down list, select one of the following:
      - **Load from *.efe and *.hfe file** and browse to the file locations.
      - **Load from *.efe file** and browse to the file location for the e-field.
      - **Load from *.hfe file** and browse to the file location for the h-field.
4. For options **Sigrity (*.nfd) input file** and **MVG (*.mfxml) measurement file**, in the **File name** field, specify the file location.
5. For option **CST near field scan (CST NFS)**, in the **Directory** field, specify the folder location.
6. [Optional] Unselect the **Use all data blocks** check box to specify a specific data block to use.
   a) In the **Use data block number** field, specify the data block number that corresponds to the near field data at a specific frequency.
7. Select the **Swap source and field validity regions** check box if the fields on the opposite side of the aperture (inside the boundary) are equivalent to the measured or calculated field values.
8. To exclude a face (or faces) when importing Cartesian boundary near field data, under **Include fields on boundary**, click the applicable button(s) to disable the button and exclude the face.
   - **-U**: Exclude the surface in the negative U direction.
   - **+N**: Exclude the surface in the positive N direction.
   - **+U**: Exclude the surface in the positive U direction.
   - **-V**: Exclude the surface in the negative V direction.
   - **-N**: Exclude the surface in the negative N direction.
   - **+V**: Exclude the surface in the positive V direction.
9. [Optional] Select the **Also sample along edges** check box to assume the outer sample points lie on the edges of the defined aperture.
10. In the **Label** field, specify a unique label for the near field data.
11. Click **Create** to define the near field data and to close the dialog.

Related tasks

Adding a Near Field Source

Requesting Ideal Receiving Antenna (Near Field Pattern)

2.15.4 Defining Spherical Modes Data from File

Import spherical modes data from a TICRA `.sph` file or import from a `.sph` file exported by CADFEKO, to create a spherical modes data definition. Use the spherical modes data definition when defining an equivalent source or receiving antenna.

1. On the **Construct** tab, in the **Define** group, click the **Field/Current Data** icon. From the drop-down list select **Import Spherical Modes Data From File**.

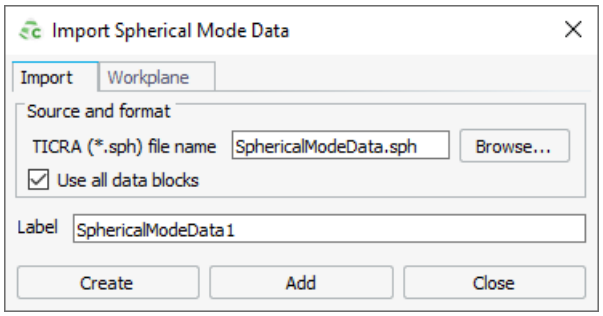


*Figure 121: The **Import Spherical Mode Data** dialog.*

2. In the **TICRA (*.sph) file** field, browse to the file location.
3. Specify the data block to be used:
   - To select data from a multi-frequency `.sph` file, select the **Use all data blocks**. The data is interpolated for use at the operating frequency.
   - To select data from a specific frequency in the `.sph` file, clear the **Use all data blocks** and enter the number of the relevant data block.
4. In the **Label** field, specify a unique label for the spherical modes data.
5. Click **Create** to define the spherical modes data and to close the dialog.

Related tasks
Adding a Spherical Mode Source
Requesting Ideal Receiving Antenna (Spherical Modes)

2.15.5 Defining Spherical Modes Data Manually

Define the propagation direction, index scheme and modes to create spherical modes data. Use the spherical modes data definition when defining an equivalent source or receiving antenna.

1. On the **Construct** tab, in the **Define** group, click the **Field/Current Data** icon. From the drop-down list select **Manually Define Spherical Modes Data**.

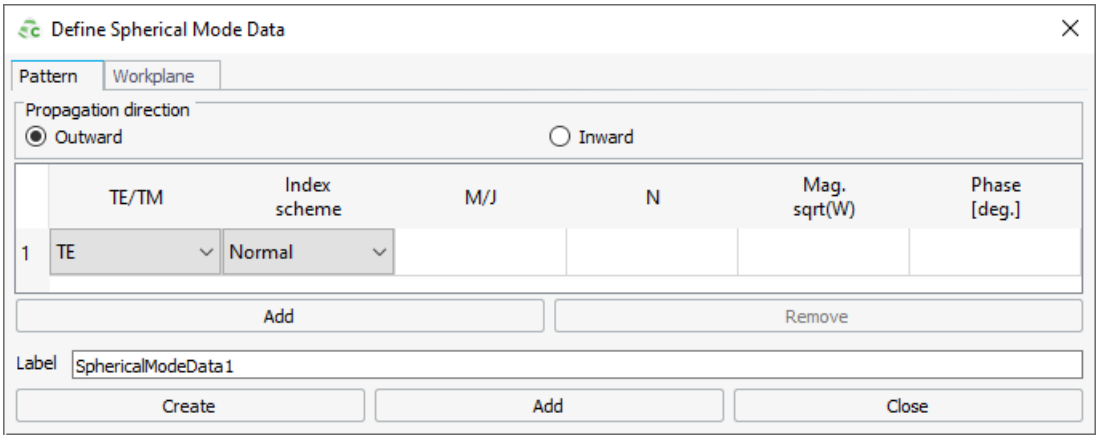


*Figure 122: The **Define spherical modes** dialog.*

2. Under **Propagation direction**, select one of the following:
   - To illuminate the model with modes propagating towards $r=0$, spherical Hankel function of the first kind, $z_N^{(3)}=h_N^{(1)}$, select **Inward**.
   - To illuminate the model with modes propagating towards $r=\infty$, spherical Hankel function of the second kind, $z_N^{(4)}=h_N^{(2)}$, select **Outward**.
3. In the **TE/TM** cell, select one of the following:
   - *TE*

     The transverse electric mode of propagation (no E-field in the direction of propagation).

     TM

     The transverse magnetic mode of propagation (no H-field in the direction of propagation).
4. In the **Index scheme** cell, select one of the following:
   - *Normal*

     This scheme uses the traditional *smn* index. You can specify TE-mode (s = 1) or TM-mode (s = 2) and the indices *M* and *N*.

     M is the mode index in the azimuth direction ϕ and *N* is the mode index in the radial direction and must be in the range 1, 2...∞.

     Feko does not distinguish between even and odd modes (with $\cos(M\varphi)$ and $\sin(M\varphi)$ angular dependencies), but rather use the angular dependency $e^{jM\varphi}$. The index *M* can also be negative, but it must be in the range $-N..N$.

     Compressed

     This scheme uses compressed one-dimensional mode numbering scheme. The *J* mode index is then specified in the index column. Here

     $$J=2\left[N(N+1)+M-1\right]+s \qquad (6)$$

     where s = 1 for TE-modes and s = 2 for TM-modes.

This unified mode numbering scheme allows the computation of an extended scattering matrix (with network and radiation ports). The index *J* then represents a unique port number in the scattering matrix.

5. In the **Mag. sqrt(W)** cell, specify the absolute value of the complex amplitude for the spherical mode. Due to the spherical modes normalisation, the amplitude unit is $\sqrt{W} = \sqrt{VA}$.
6. In the **Phase [deg.]** field, specify the phase of the complex amplitude for the spherical mode.
7. In the **Label** field, specify a unique label for the spherical modes data.
8. Click **Create** to define the spherical modes data and to close the dialog.

Related concepts
Exporting a `.sph` file in CADFEKOExporting a `.sph` file in CADFEKO
Related tasks
Adding a Spherical Mode Source
Requesting Ideal Receiving Antenna (Spherical Modes)

2.15.6 Defining PCB Current Data from File

Import printed circuit board (PCB) current data from a PollEx radiated emission interface (`.rei`) file to create a PCB current data definition. Use the current data definition when defining an equivalent source.

1. On the **Construct** tab, in the **Define** group, click the **Field/Current Data** icon. From the drop-down list select **Import PCB Current Data From File**.

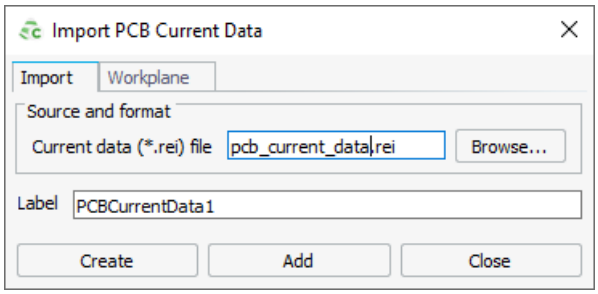


*Figure 123: The **Import PCB Current Data** dialog.*

2. In the **Current data (*.rei) file** field, browse to the file location of the `.rei` PCB current data file that was written out by PollEx.
3. In the **Label** field, specify a unique label for the PCB current data.
4. Click **Create** to define the PCB current data and to close the dialog.

Related concepts
Feko Source Data Viewer
Related tasks
Adding a PCB Source
Visualising PCB Current Data

Visualising PCB Current Data

View the PCB board outline with currents per frequency for a PCB current data definition.

1. In the model tree, under **Field/Current Data**, select a PCB current data. From the right-click context menu, click **Visualise PCB Current Data**.

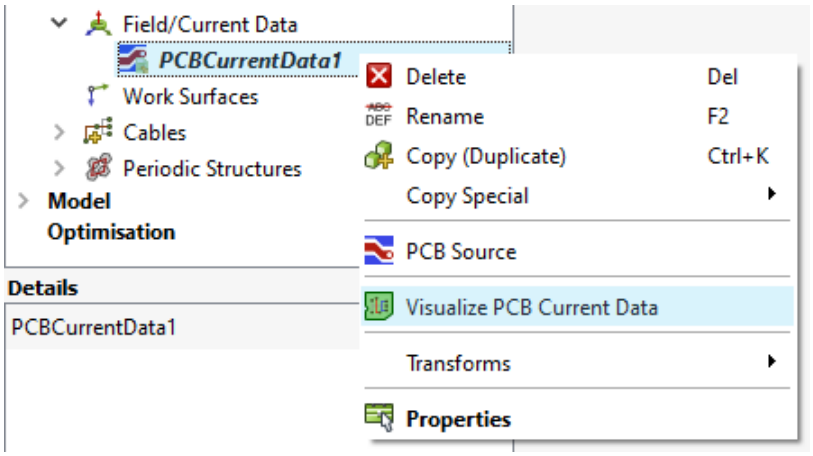


Figure 124: The Visualise PCB Current Data right-click context menu option.

The **Visualise PCB Current Data** dialog is displayed.

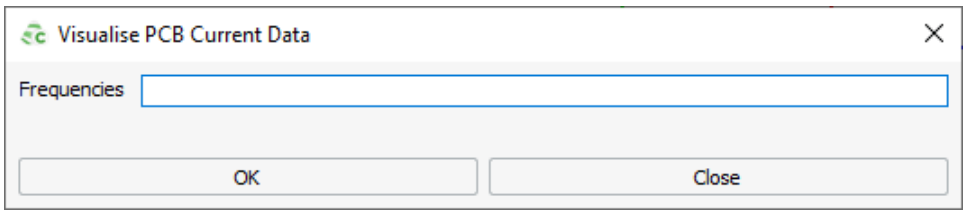


*Figure 125: The **Visualise PCB Current Data** dialog.*

2. On the **Visualise PCB Current Data** dialog, in the **Frequencies** field, specify the frequencies (values are in Hz):

Note:

- Leave the field empty to visualise all frequencies.
- Specify a single value:

```
1e9
```

- Specify a comma-separated list:

```
500 MHz, 2GHz
```

- Specify a frequency range:

```
1GHz - 2GHz
```

- Use a combination of a comma-separated list and range:

```
500 MHz, 2GHz, 1GHz - 2GHz
```

If the requested frequency is not included in the PCB current data, the closest frequencies to the requested value are included.

3. Click **OK** to view the PCB current data in the **Feko Source Data Viewer**.

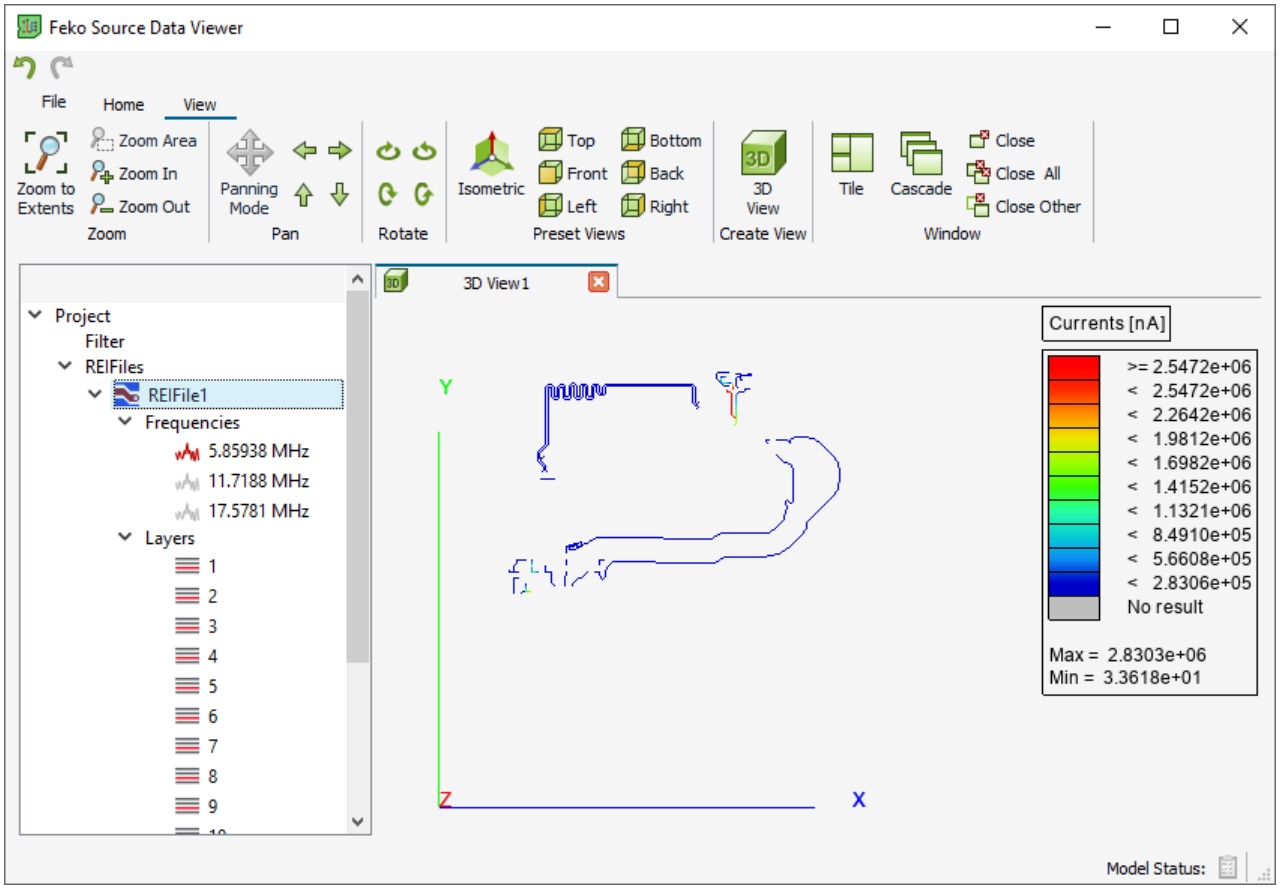


Figure 126: The Feko Source Data Viewer where you can view currents per frequency for a PCB current data definition.

Related concepts
Feko Source Data Viewer
Related tasks
Defining PCB Current Data from File

2.15.7 Defining Solution Coefficient Data from File

Import solution coefficient data (multi-frequency or single frequency) from a `.sol` file to create a solution coefficient data definition. Use the solution coefficient data definition when defining a solution coefficient source.

> **Note:** Export surface currents on specified structures to a `.sol` file using a domain decomposition request.

1. On the **Construct** tab, in the **Define** group, click the **Field/Current Data** icon. From the drop-down list select **Import Solution Coefficient Data From File**.

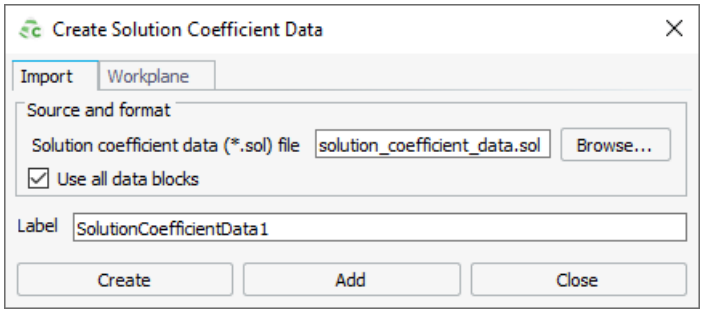


*Figure 127: The **Import Solution Coefficient Data** dialog.*

2. In the **Solution coefficient data (*.sol) file** field, browse to the file location.
3. Specify the data block to be used:
   - To select data from a multi-frequency `.sol` file, select the **Use all data blocks** check box. The data is interpolated for use at the operating frequency.
   - To select data at a specific frequency in the `.sol` file, clear the **Use all data blocks** check box and enter the number of the relevant data block.
4. In the **Label** field, specify a unique label for the solution coefficient data.
5. Click **Create** to define the spherical modes data and to close the dialog.

Related tasks

Exporting a `.sol` file in CADFEKOExporting a `.sol` file in CADFEKO

Adding a Solution Coefficient Source

Requesting Model Decomposition

2.16 Defining Media

Define a medium with specific material properties, import a predefined medium from the media library or add a medium from your model to the media library.

Note: Only passive media are supported. Passive media can be either lossless or lossy.[27]

The following media types are supported:

1. Dielectric
2. Metal
3. Layered dielectric (isotropic and anisotropic)
4. Impedance sheet
5. Characterised surface
6. Windscreen layer
7. Anisotropic medium (3D)

Media are displayed in the model tree. This includes user-defined media and media added from the media library.

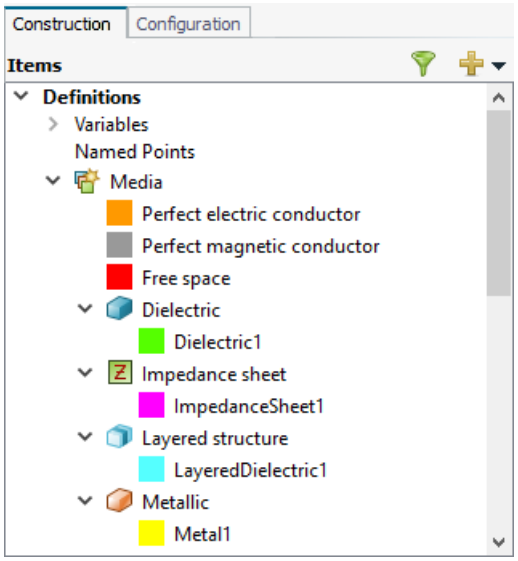


Figure 128: The media definitions in the model tree

The colour square next to each medium entry indicates the colour that is used to display the medium in the 3D view as well as in POSTFEKO. To change the display colour, click the dielectric in the model tree and from the right-click context menu, select **Change display colour**.

27. A lossless passive medium allows fields to pass through the medium without attenuation. In a lossy passive medium, a fraction of the power is transformed to heat, as an example.

Predefined Media

Predefined media are available by default in CADFEKO and includes **Perfect electric conductor**, **Perfect magnetic conductor** and **Free space**.

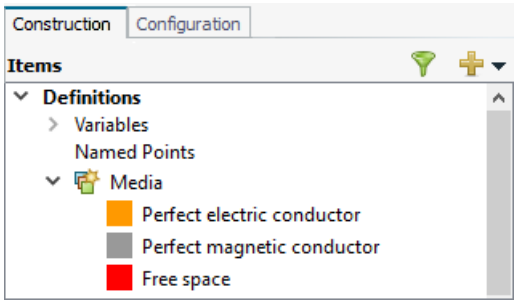


Figure 129: The predefined media in the model tree

Tip: Edit the properties of free space if the model is inside an infinite medium.

Related reference

Media and Supported Solution Methods

2.16.1 Altair Material Data Centre

The Altair Material Data Centre (AMDC) enables designers, engineers, and scientists to browse, search, and compare materials.

To solve electromagnetic problems, you require accurate material properties. The AMDC gives you direct access to a wide range of materials through a common user experience and enable you to make faster, more informed material decisions.

Importing Material Data from Altair Material Data Centre

Import material data from Altair Material Data Centre (AMDC).

Note:

- You require an Altair One account to use AMDC.
- The AMDC is only supported on Microsoft Windows.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Altair Material Data Center** icon.
2. Log in to your Altair One account.
3. Enter a search term in the search box.
4. In the **Filters** list, under **Provider**, select **Feko**.
5. Double-click on a material.

6. In the information window, select the **CADFEKO** tab.
7. Click **Download**.
   The material is imported and available in the model tree under **Media**.

Note: For more information, view the Altair Material Data Centre documentation available at https://community.altair.com/community.

2.16.2 Media Library

The media library contains a list of predefined media that can be added to the model.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Media Library** icon.

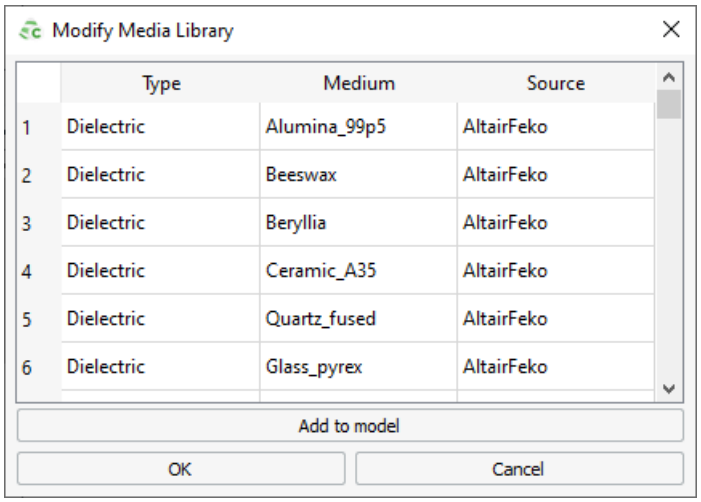


*Figure 130: The **Modify Media Library** dialog.*

2. Click the medium in the list you want to add to your model.
3. To add the selected medium to your model, click **Add to model**.
4. Click **Close** to close the dialog.

2.16.3 Creating a Dielectric Medium

Create a frequency-independent dielectric.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, click the **Dielectric** icon.
2. Click **Manually define medium**.
3. [Optional] In the **Mass density (kg/m^3)** field, enter a value for ρ.

4. In the **Label** field, enter a unique label for the dielectric.
5. Click **Create** to create the dielectric and to close the dialog.

Related tasks
Applying a Dielectric to a Region
Creating an Isotropic Layered Dielectric (2D)

Dielectric Properties

Specify the dielectric properties of the dielectric medium.

1. Click the **Dielectric modelling** tab.
2. In the **Definition method** field, from the drop-down list select **Frequency independent**.
3. In the **Relative permittivity** field, enter a value for ε_r.

Specify the dielectric losses in the dielectric by either specifying the dielectric loss tangent or the conductivity. The two loss terms are related by $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$.

> **Note:** Low loss dielectric substrates are typically specified in terms of loss tangent, while human tissue (used in SAR[28] studies) are specified in terms of conductivity.

4. Select one of the following:
   - To specify the dielectric loss tangent, click **Dielectric loss tangent**.
     - In the **Dielectric loss tangent** field, enter a value for tanδ.
   - To specify the conductivity directly, click **Conductivity (S/m)**.
     - In the **Conductivity (S/m)** field, enter a value for σ.

Magnetic Properties

Specify the magnetic properties of the dielectric medium.

1. Click the **Magnetic modelling** tab.
2. Specify the magnetic properties of the dielectric.
   - To create a non-magnetic dielectric, from the **Definition method** drop-down list select **Non magnetic**.
   - To create a dielectric with frequency-independent-magnetic-properties, from **Definition method** drop-down list select **Frequency independent**.
     1. In the **Relative permeability** field, enter the value for μ_r.
     2. In the **Magnetic loss tangent field**, enter the value for $\tan\delta_\mu$.
   - To create a dielectric with frequency-dependent-magnetic-properties, from the **Definition method** drop-down list select **Frequency list (linear interpolation)**.

28. specific absorption rate

- To enter each frequency points manually, enter the magnetic properties for each frequency point.
- To import the frequency points from a file, click **Import points**.
  1. In the **File name** field, browse for the file you want to import.
  2. [Optional] In the **Scale by** field, enter a value to scale the points.

     For example, if you import a value of "2", scale it by "10e9" to change the value to 2 GHz.
  3. Under **Delimiter**, click the delimiter type you use in your file.
  4. Click **OK** to close the **Import Points** dialog.

Frequency Dependent Dielectrics

Define a frequency-dependent medium to use in your model or to add to the media library.

The following frequency-dependent definitions are supported:

Debye relaxation	Use this method to describe the relaxation characteristics of gasses and fluids at microwave frequencies. It is derived for freely rotating spherical polar molecules in a predominantly non-polar background.
Cole-Cole	This method is similar to the Debye relaxation but makes use of an additional parameter to describe the model.
Havriliak-Negami	Use this method to model liquids, solids and semi-solids.
Djordjevic-Sarkar	Use this method for composite dielectrics.
Frequency List	Use this method to define a frequency-dependent dielectric by specifying data points at a range of frequencies. The values for the dielectric properties are linearly interpolated to obtain the dielectric properties at frequency points other than specified.

Related concepts
Dielectric Media Formulations

Creating a Dielectric Medium (Debye Relaxation)

Create a frequency-dependent dielectric using the Debye relaxation method. Use the Debye model to describe the relaxation characteristics of gasses and fluids at microwave frequencies. It is derived for freely rotating spherical polar molecules in a predominantly non-polar background.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, click the **Dielectric** icon.

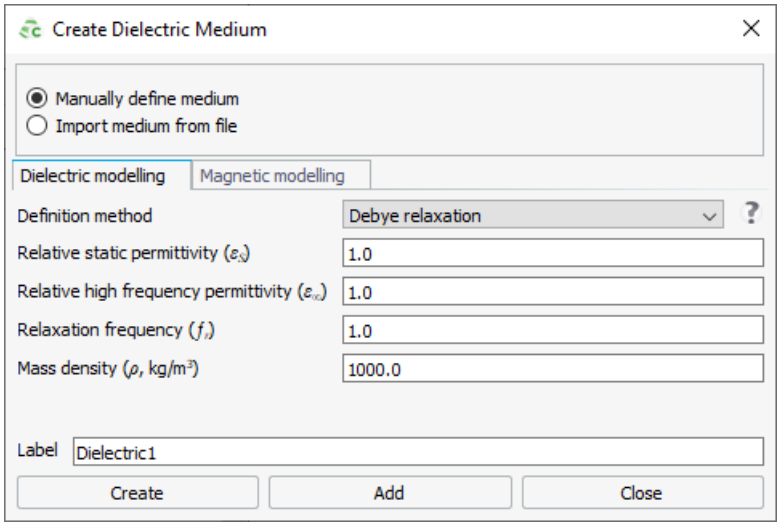


*Figure 131: The **Create Dielectric Medium** dialog (Debye relaxation).*

2. In the **Definition method** field, from the drop-down list select **Debye relaxation**.
3. In the **Relative static permittivity** field, enter a value for ε_s.
4. In the **Relative high frequency permittivity** field, enter a value for ε_∞.
5. In the **Relaxation frequency** field, enter a value for f_r.
6. [Optional] Specify the magnetic properties of the dielectric.
7. [Optional] In the **Mass density (kg/m^3)** field, enter a value for ρ.
8. In the **Label** field, enter a unique label for the dielectric.
9. Click **Create** to create the dielectric and to close the dialog.

Related tasks

Applying a Dielectric to a Region

Creating an Isotropic Layered Dielectric (2D)

Creating a Dielectric Medium (Cole-Cole)

Create a frequency-dependent dielectric using the Cole-Cole method. The method is similar to the Debye relaxation but makes use of an additional parameter to describe the model.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, click the **Dielectric** icon.

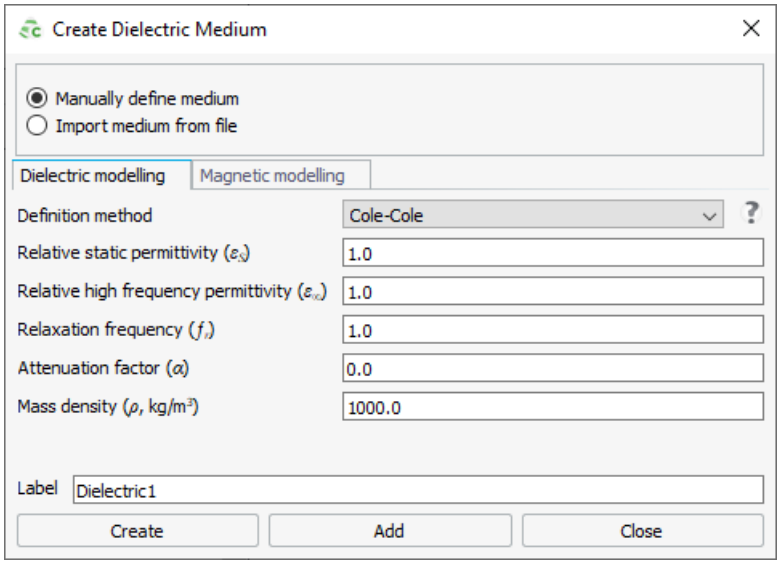


*Figure 132: The **Create Dielectric Medium** dialog (Cole-Cole).*

2. In the **Definition method** field, from the drop-down list select **Cole-Cole**.
3. In the **Relative static permittivity** field, enter a value for ε_s.
4. In the **Relative high frequency permittivity** field, enter a value for ε_∞.
5. In the **Relaxation frequency** field, enter a value for f_r.
6. In the **Attenuation factor** field, enter a value for α.
7. [Optional] Specify the magnetic properties of the dielectric.
8. [Optional] In the **Mass density (kg/m^3)** field, enter a value for ρ.
9. In the **Label** field, enter a unique label for the dielectric.
10. Click **Create** to create the dielectric and to close the dialog.

Related tasks
Applying a Dielectric to a Region
Creating an Isotropic Layered Dielectric (2D)

Creating a Dielectric Medium (Havriliak-Negami)

Create a frequency-dependent dielectric using the Havriliak-Negami method. Use this method to model liquids, solids and semi-solids.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, click the **Dielectric** icon.

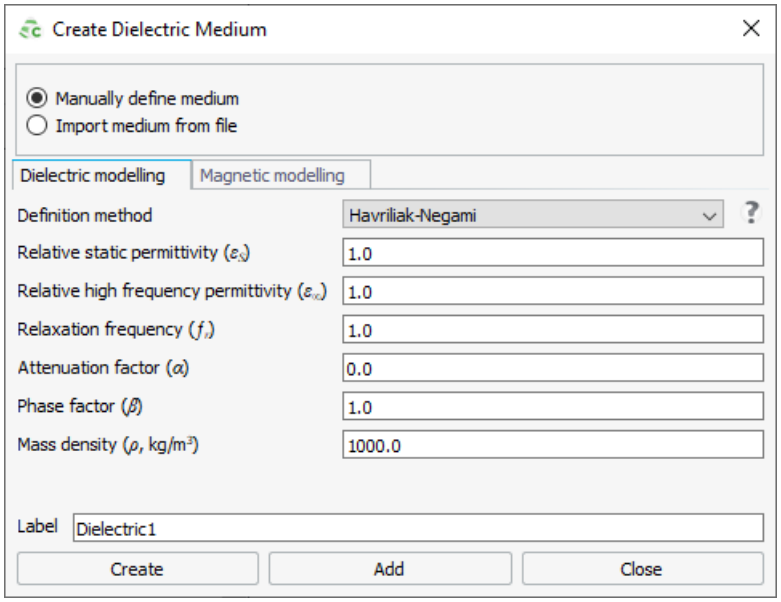


*Figure 133: The **Create Dielectric Medium** dialog (Havriliak-Negami).*

2. In the **Definition method** field, from the drop-down list select **Havriliak-Negami**.
3. In the **Relative static permittivity** field, enter a value for ε_s.
4. In the **Relative high frequency permittivity** field, enter a value for ε_∞.
5. In the **Relaxation frequency** field, enter a value for f_r.
6. In the **Attenuation factor** field, enter a value for α.
7. In the **Phase factor** field, enter a value for β.
8. [Optional] Specify the magnetic properties of the dielectric.
9. In the **Label** field, enter a unique label for the dielectric.
10. Click **Create** to create the dielectric and to close the dialog.

Related tasks

Applying a Dielectric to a Region

Creating an Isotropic Layered Dielectric (2D)

Creating a Dielectric Medium (Djordjevic-Sarkar)

Create a frequency-dependent dielectric using the Djordjevic-Sarkar method. Use this method for composite dielectrics.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, click the **Dielectric** icon.

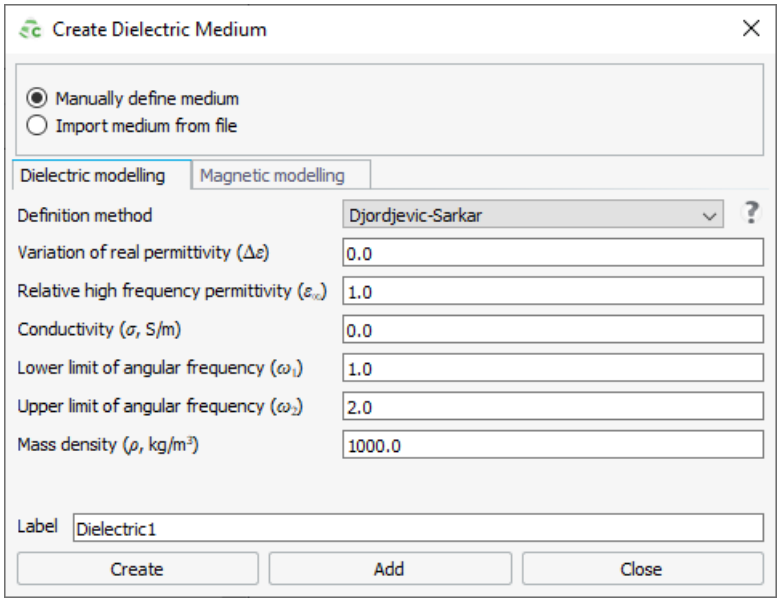


*Figure 134: The **Create Dielectric Medium** dialog (Djordjevic-Sarkar).*

2. In the **Definition method** field, from the drop-down list select **Djordjevic-Sarkar**.
3. In the **Variation of real permittivity** field, enter a value for Δε.
4. In the **Relative high frequency permittivity** field, enter a value for ε_∞.
5. In the **Conductivity (S/m)** field, enter a value for σ.
6. In the **Lower limit of angular frequency** field, enter a value for ω_1.
7. In the **Upper limit of angular frequency**, enter a value for ω_2.
8. [Optional] Specify the magnetic properties of the dielectric.
9. [Optional] In the **Mass density (kg/m^3)** field, enter a value for ρ.
10. In the **Label** field, enter a unique label for the dielectric.
11. Click **Create** to create the dielectric and to close the dialog.

Related tasks
Applying a Dielectric to a Region
Creating an Isotropic Layered Dielectric (2D)

Creating a Dielectric Medium from a Frequency List

Define a frequency-dependent dielectric by specifying data points at a range of frequencies. The values for the dielectric properties are linearly interpolated to obtain the dielectric properties at frequency points other than specified.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, click the **Dielectric** icon.

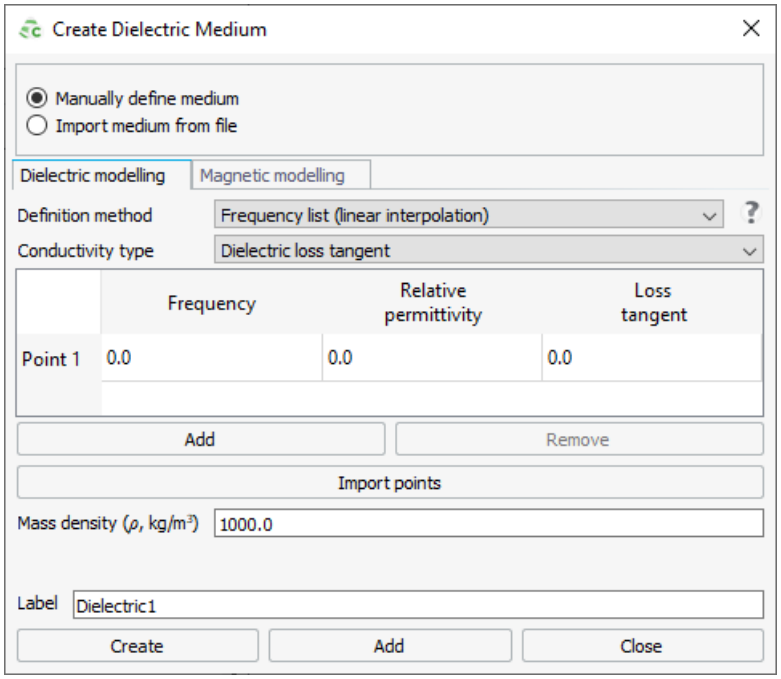


*Figure 135: The **Create Dielectric Medium** dialog (Frequency list).*

2. In the **Definition method** field, from the drop-down list select **Frequency list (linear interpolation)**.

Specify if the frequency points are manually entered or imported from a file.

3. Select one of the following:
   - To enter each frequency point, click **Dielectric loss tangent** or **Conductivity** and enter the dielectric properties for each frequency point.
   - To import the frequency points from a file, click **Import points**.
     1. In the **File name** field, browse for the file you want to import.
     2. [Optional] In the **Scale by** field, enter a value to scale the points.

        For example, if you import a value of "2", scale it by "10e9" to change the value to 2 GHz.
     3. Under **Delimiter**, click the delimiter type you use in your file.
     4. Click **OK** to close the **Import Points** dialog.
4. [Optional] Specify the magnetic properties of the dielectric.
5. [Optional] In the **Mass density (kg/m^3)** field, enter a value for ρ.
6. In the **Label** field, enter a unique label for the dielectric.
7. Click **Create** to create the dielectric and to close the dialog.

Related tasks

Applying a Dielectric to a Region

Creating an Isotropic Layered Dielectric (2D)

2.16.4 Creating an Isotropic Layered Dielectric (2D)

Create a layered dielectric medium.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Layered Dielectric (2D)** icon.

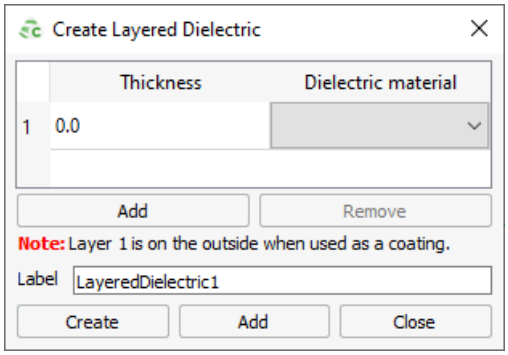


*Figure 136: The **Create Layered Dielectric** dialog.*

2. In the **Label** field, enter a unique label for the dielectric.
3. In the **Thickness** field, enter a value for the layer thickness.
4. In the **Dielectric material** field, select one of the following options:
   - To add a dielectric layer consisting of a predefined dielectric, select the dielectric.
   - To add a dielectric layer consisting of a dielectric, which is not yet defined in the model, click the icon to define the dielectric or add a dielectric from the media library.
5. [Optional] To add an additional layer, click **Add**.
6. [Optional] To remove a layer, click **Remove**.
7. Click **Create** to create the dielectric and to close the dialog.

Related tasks

Applying a Coating to a Wire or Face

Applying a Thin Dielectric Sheet to a Face

2.16.5 Creating an Anisotropic Layered Dielectric (2D)

Create an anisotropic layered dielectric.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select **Anisotropic Layered Dielectric (2D)** icon.
2. In the **Label** field, enter a unique label for the dielectric.

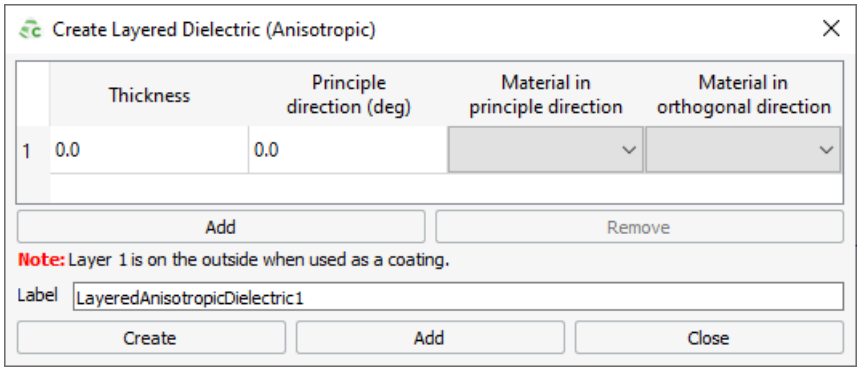


*Figure 137: The **Create Layered Dielectric (Anisotropic)** dialog.*

3. In the **Thickness** field, enter a value for the layer thickness.
4. In the **Principal direction (deg)** field, enter the angle of the principal direction.
5. In the **Material in principal direction** field, select one of the following:
   - To add a dielectric layer in the principal direction, consisting of a predefined dielectric, select the dielectric.
   - To add a dielectric layer in the principal direction consisting of a dielectric, which is not yet defined in the model, click the icon to define the dielectric or add a dielectric from the media library.
6. In the **Material in orthogonal direction** field, select one of the following:
   - To add a dielectric layer in the orthogonal direction consisting of a predefined dielectric, select the dielectric.
   - To add a dielectric layer in the orthogonal direction consisting of a field, which is not yet defined in the model, click the icon to define the dielectric or add a dielectric from the media library.
7. [Optional] To add an additional layer, click **Add**.
8. [Optional] To remove a layer, click **Remove**.

Related tasks
Applying a Layered Anisotropic to a Face

2.16.6 Creating a Metallic Medium

Create a metal.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Metallic** icon.
2. Click **Manually define medium**.
3. Specify the magnetic properties of the metal.

- To create a frequency-independent metal, from the **Definition method** drop-down list select **Frequency independent**.
- To create a frequency-dependent metal, from the **Definition method** drop-down list select **Frequency list (linear interpolation)**.
  - To enter each frequency point manually, enter the magnetic properties for each frequency point.
  - To import the frequency points from a file, click **Import points**.
    1. In the **File name** field, browse for the file you want to import.
    2. [Optional] In the **Scale by** field, enter a value to scale the points.

       For example, if you import a value of "2", scale it by "10e9" to change the value to 2 GHz.
    3. Under **Delimiter**, click the delimiter type you use in your file.
    4. Click **OK** to close the **Import Points** dialog.

4. [Optional] To include the effects of a rough surface, select the **Surface roughness (RMS value in m)** check box and enter a value (root mean square value in metre).[29]

Related tasks
Applying a Metal to a Face

2.16.7 Importing a Medium from File

Import a dielectric, metal or impedance sheet from a `.XML` file that describes the medium properties.

1. Select one of the following:
   - Create a dielectric. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, click the **Dielectric** icon.
   - Create a metal. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Metallic** icon.
   - Create an impedance sheet. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Impedance Sheet** icon.
2. Click **Import medium from file**.
3. In the **File name** field, browse for the file you want to import.
4. Click **Create** to import the medium and to close the dialog.

29. Specifying the surface roughness will not impact the total runtime or memory used.

XML File Format for Importing Media

View the `.xml` file format that describes the medium properties of a dielectric, metal or impedance sheet.

Overview

Define the medium using the following workflow:

1. Define the frequency independent (static) properties for the medium using keywords.
2. Define the frequency dependent properties for the medium using keywords.

When the file is read, the internal XML parser populates the missing values for the frequency dependent data points using the static data points.

Keywords for the `.xml` file

Use the following keywords to define the medium:

Dielectric
: *freq*, *permittivity*, *diel_loss_tangent*, *mag_loss_tangent*, *conductivity*, *permeability*.

Metal
: *freq*, *conductivity*, *permeability*, *mag_loss_tangent*.

Impedance sheet
: *freq*, *surf_imp_re*, *sur_imp_im*

Example of an `.xml` file

The following is an example of an `.xml` file that describes a medium with frequency independent (static) values as well as measured frequency independent values.

> **Note:** For demonstrative purposes, the keywords *val_A*, *val_B* and *val_C* are used as the same format is applicable for defining a dielectric, metallic or impedance sheet.

```
<?xml version="1.0" encoding="UTF-8"?>
<materialDB creator="Altair Feko" date="2017-06-29" version="1.0">
<material name="mediumA" val_B="7.0" val_C="9.0" >
<dataPoint freq="2.0" val_A="2.0" val_B="6.0" />
<dataPoint freq="3.0" val_A="3.0" />
<dataPoint freq="4.0" val_A="1.0" />
<dataPoint freq="5.0" val_B="5.0" />
<dataPoint freq="6.0" val_A="1.0" />
<dataPoint freq="8.0" val_B="6.0" />
<dataPoint freq="9.0" val_A="4.0" />
</material>
</materialDB
```

In line 3: Define the frequency independent (static) properties for *mediumA*.

```
<material name="mediumA" val_B="7.0" val_C="9.0" >
```

In line 4 to 10: Define the frequency dependent properties for *mediumA*.

```
<dataPoint freq="2.0" val_A="2.0" val_B="6.0" />
```

The internal XML parser then populates the missing values. The above example is parsed internally as if you specified the following file:

```
<?xml version="1.0" encoding="UTF-8"?>
<materialDB creator="Altair Feko" date="2017-06-29" version="1.0">
<material name="mediumA" >
<dataPoint freq="2.0" val_A="2.0" val_B="6.0" val_C="9.0" />
<dataPoint freq="3.0" val_A="3.0" val_B="7.0" val_C="9.0" />
<dataPoint freq="4.0" val_A="1.0" val_B="7.0" val_C="9.0" />
<dataPoint freq="5.0" val_B="5.0" val_C="9.0" />
<dataPoint freq="6.0" val_A="1.0" val_B="7.0" val_C="9.0" />
<dataPoint freq="8.0" val_B="6.0" val_C="9.0" />
<dataPoint freq="9.0" val_A="4.0" val_B="7.0" val_C="9.0" />
</material>
</materialDB>
```

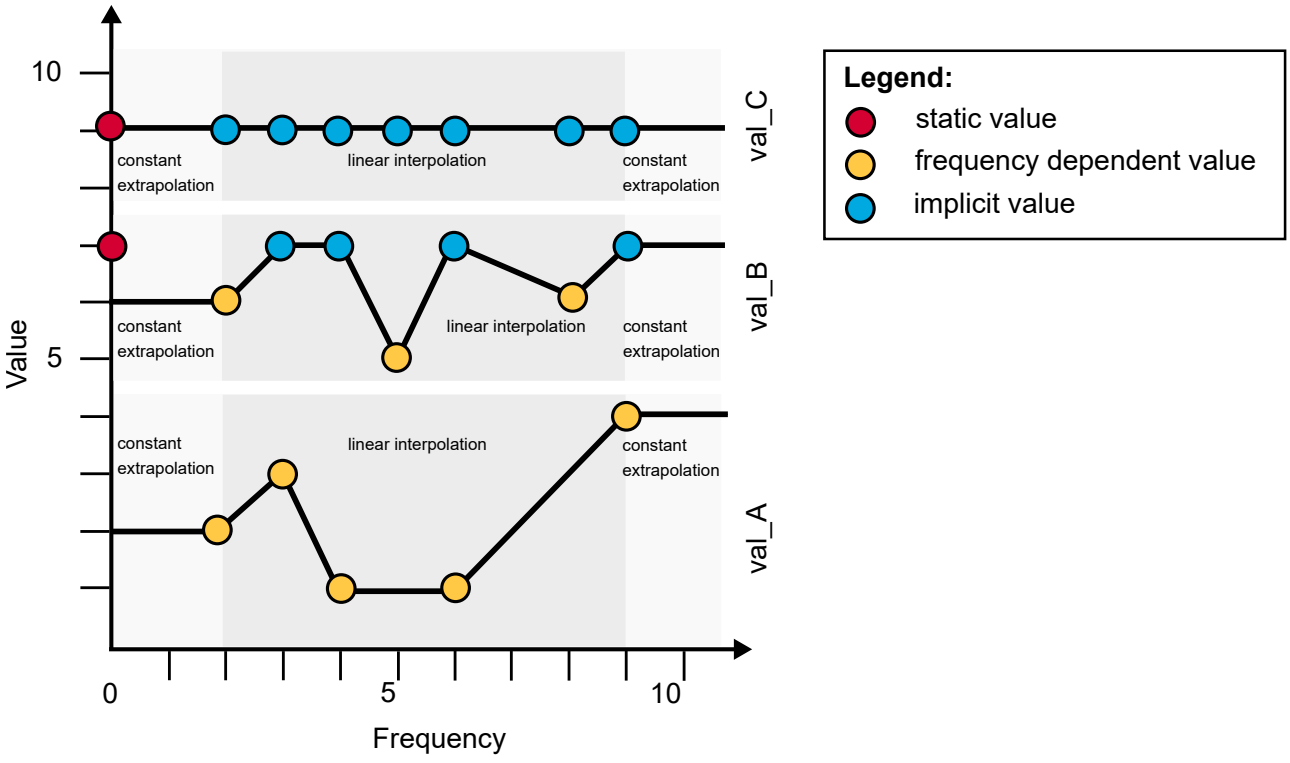


Figure 138: An illustration showing the result of the parsed XML file.

2.16.8 Anisotropic Media (3D)

Create an anisotropic medium with specified material properties.

The following tensor types descriptions are supported:

- Polder tensor (for ferrites)

- Diagonalised tensor
- Full tensor
- Complex tensor

Related concepts
Anisotropic Media Formulations

Creating an Anisotropic Medium (Ferrite)

Create a ferrimagnetic medium. The medium is described by the permittivity and permeability tensors where the static magnetic field is orientated along the U axis, V axis and N axis respectively.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Anisotropic (3D)** icon.

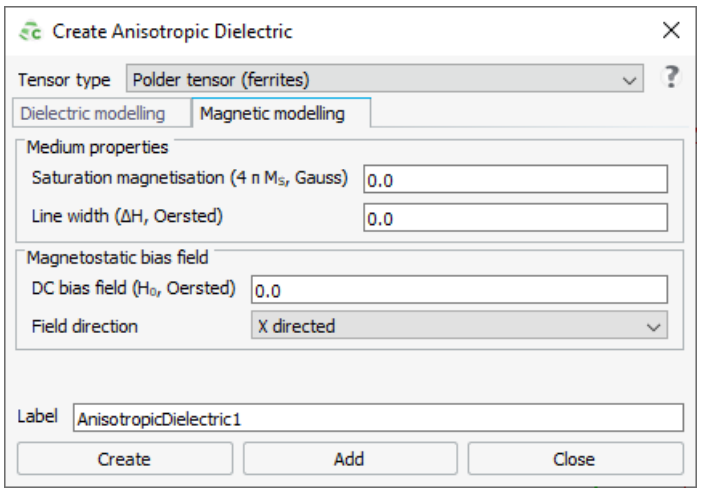


*Figure 139: The **Create Anisotropic Dielectric** dialog (Polder tensor).*

2. In the **Tensor type** field, from the drop-down list select **Polder tensor (ferrites)**.
3. Specify the magnetic properties.
   a) Under **Medium properties**, in the **Saturation magnetisation (Gauss)** field, enter a value for $4\pi M_s$.
   b) Under **Medium properties**, in the **Line width (Oersted)** field, enter a value for ΔH.
   c) Under **Magnetostatic bias field**, in the **DC bias field (Oersted)** field, enter a value for H_0.
   d) Under **Magnetostatic bias field**, from the **Field direction** drop-down list, select one of the following:
      - **X directed**
      - **Y directed**
      - **Z directed**
4. Specify the dielectric properties.
   a) In the **Relative permittivity** field, enter a value for ε_r.

b) In the **Dielectric loss tangent** field, enter a value for tanδ.

5. In the **Label** field, enter a unique label for the ferrite medium.
6. Click **Create** to create the anisotropic medium and to close the dialog.

Related tasks
Applying an Anisotropic Dielectric to a Region

Creating an Anisotropic Medium (Diagonalised Tensor)

Create an anisotropic medium by defining the diagonal permittivity tensor and diagonal permeability tensor.

Tip: Create up to three dielectrics constituting the medium properties along the UU axis, VV axis and NN axis.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Anisotropic (3D)** icon.

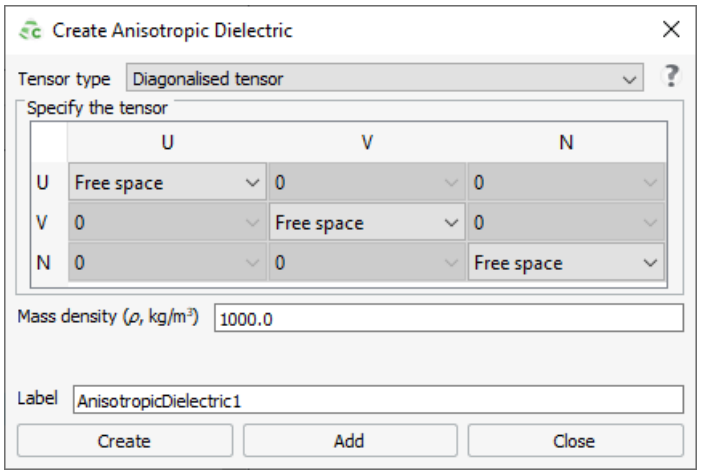


*Figure 140: The **Create Anisotropic Dielectric** dialog (Diagonalised tensor).*

2. In the **Tensor type** field, from the drop-down list select **Diagonalised tensor**.
3. For each diagonal entry, select one of the following options:
   - To use the medium properties of free space, from the drop-down list select **Free space**.
   - To indicate that no linear dependencies exist between the two axes, from the drop-down list select **0**.
   - To use the medium properties of a predefined dielectric, from the drop-down list, select the dielectric.
   - To use the medium properties of a dielectric, which is not yet defined in the model, click the icon to define the dielectric or add a dielectric from the media library.
4. [Optional] In the **Mass density (kg/m^3)** field, enter a value for ρ.

5. Click **Create** to create the anisotropic medium and to close the dialog.

Related tasks
Applying an Anisotropic Dielectric to a Region

Creating an Anisotropic Medium (Full Tensor)

Create an anisotropic medium by defining the diagonal-permittivity tensor and diagonal-permeability tensor along the UU axis, UV axis, UN axis, VU axis, VV axis, VN axis, NU axis, NV axis and NN axis.

> **Tip:** Create up to nine dielectrics constituting the medium properties along the UU axis, UV axis, UN axis, VU axis, VV axis, VN axis, NU axis, NV axis and NN axis.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Anisotropic (3D)** icon.

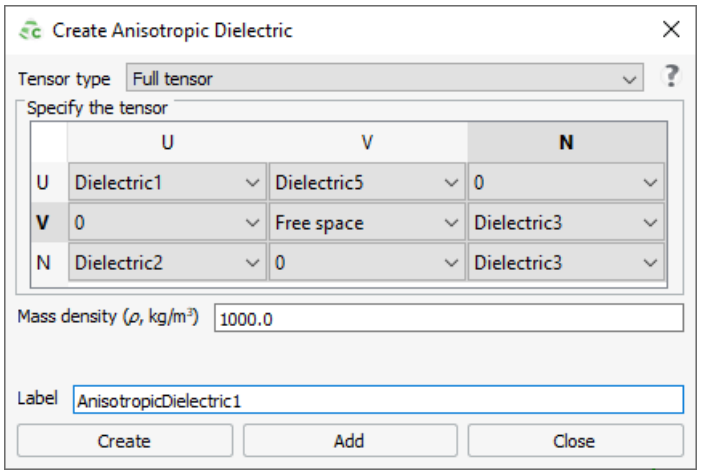


*Figure 141: The **Create Anisotropic Dielectric** dialog (Full tensor).*

2. In the **Tensor type** field, from the drop-down list select **Full tensor**.
3. For each entry, select one of the following options:
   - To use the medium properties of free space, from the drop-down list select **Free space**.
   - To indicate that no linear dependencies exist between the two axes, from the drop-down list select **0**.
   - To use the medium properties of a predefined dielectric, from the drop-down list, select the dielectric.
   - To use the medium properties of a dielectric, which is not yet defined in the model, click the icon to define the dielectric or add a dielectric from the media library.
4. [Optional] In the **Mass density (kg/m^3)** field, enter a value for ρ.
5. Click **Create** to create the anisotropic medium and to close the dialog.

Related tasks
Applying an Anisotropic Dielectric to a Region

Creating an Anisotropic Medium (Complex Tensor)

Create an anisotropic medium by defining the permittivity tensor and permeability tensor using complex values.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Anisotropic (3D)** icon.

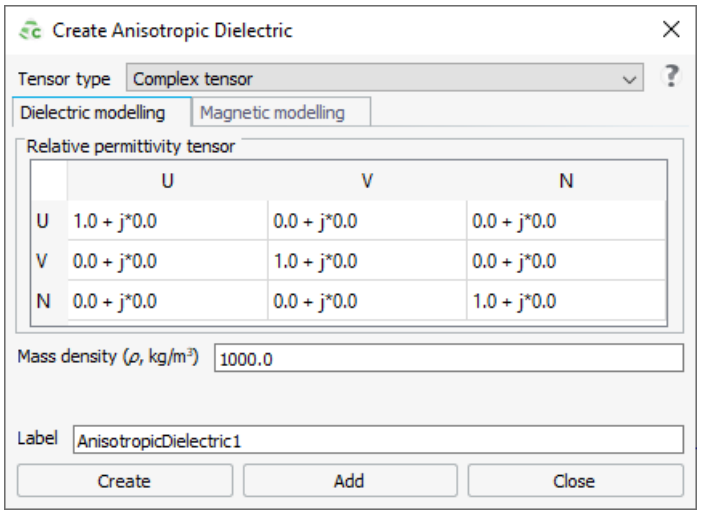


*Figure 142: The **Create Anisotropic Dielectric** dialog (Complex tensor).*

2. In the **Tensor type** field, from the drop-down list select **Complex tensor**.
3. Click the **Dielectric modelling** tab.
   a) Enter a complex value in the relevant entries.
4. Click the **Magnetic modelling** tab.
   a) Enter a complex value in the relevant entries.

> **Important:**
> - An entry in the tensor must be a complex number, pure real number or a pure imaginary number.
> - An entry may not be 0.

5. [Optional] In the **Mass density (kg/m^3)** field, enter a value for ρ.
6. Click **Create** to create the anisotropic medium and to close the dialog.

Related tasks
Applying an Anisotropic Dielectric to a Region

2.16.9 Creating a Windscreen Layer

Create a windscreen layer consisting of dielectric layers.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Windscreen** icon.

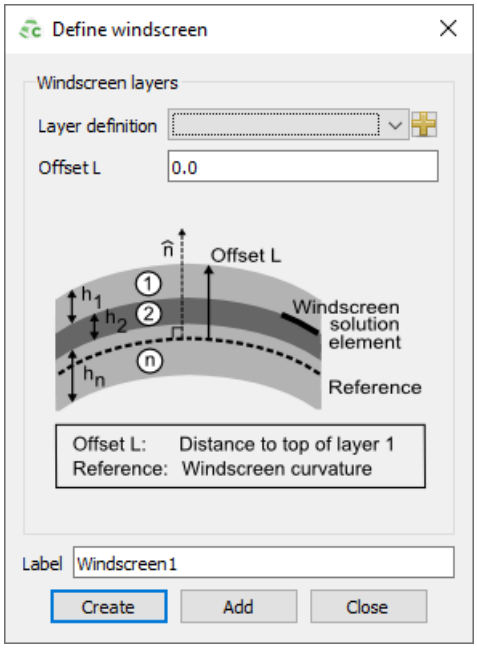


*Figure 143: The **Define windscreen** dialog.*

2. In the **Layer definition** field, select one of the following:
   - To create a windscreen layer consisting of a predefined layered dielectric, select the layered dielectric.
   - To create a windscreen layer consisting of a layered dielectric, which is not yet defined in the model, click the icon to define the layered dielectric.

> **Note:** The enumeration of the windscreen layers increases in the opposite direction as the reference direction.

3. In the **Offset L** field, enter a value for the distance from the windscreen curvature reference to the top of layer 1.
4. In the **Label** field, enter a unique label for the windscreen medium.
5. Click **Create** to create a windscreen layer and to close the dialog.

Related tasks

Applying a Windscreen Layer to a Face

Displaying Windscreen Thickness

2.16.10 Creating an Impedance Sheet

Define a frequency-dependent impedance sheet. Apply the impedance sheet to wires or faces bordering free space or a dielectric region.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Impedance Sheet** icon.

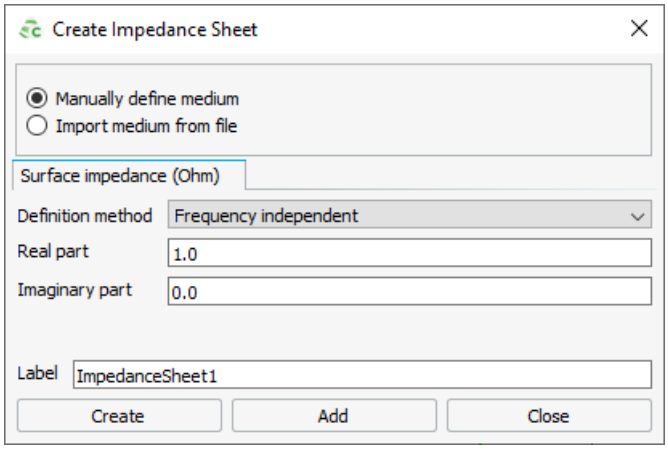


*Figure 144: The **Create Impedance Sheet** dialog.*

2. In the **Definition method** field, from the drop-down list select **Frequency independent**.
3. In the **Real part** field, enter a value for the real part of the impedance sheet.
4. In the **Imaginary** field, enter a value for the imaginary part of the impedance sheet.

Specify the frequency points manually or import them from a file.

5. Select one of the following:
   - To import the frequency points from a file, click **Import points**.
     1. In the **File name** field, browse for the file you want to import.
     2. [Optional] In the **Scale by** field, enter a value to scale the points.
     3. Under **Delimiter**, click the delimiter type you use in your file.
     4. Click **OK** to close the **Import Points** dialog.
6. In the **Label** field, enter a unique label for the impedance sheet.
7. Click **Create** to create the impedance sheet and to close the dialog.

2.16.11 Creating a Characterised Surface

Define a surface that is characterised by previously obtained data in a `.tr` file.

Note: Only supported in conjunction with the RL-GO solution method.

1. On the **Construct** tab, in the **Define** group, click the **Media** icon. From the drop-down list, select the **Characterised Surface** icon.

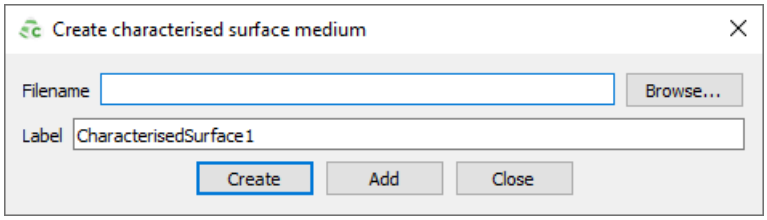


*Figure 145: The **Create characterised surface medium** dialog.*

2. In the **Filename** field, browse to the location of the `.tr` file.
3. In the **Label** field, enter a unique label for the characterised surface.
4. Click **Create** to define the characterised surface and to close the dialog.

Related concepts
Ray Launching Geometrical Optics (RL-GO)
Related tasks
Applying a Characterised Surface to a Face

2.16.12 Media and Supported Solution Methods

View the restrictions when using media in a model since not all media types are supported for all solution methods.

Table 6: Restrictions when using dielectrics.

Solution Method	Dielectric	Layered Dielectric (2D)	Anisotropic Layered Dielectric (2D)	Anisotropic Dielectric (3D)
MoM	Yes	Yes – properties must differ from the surrounding medium	Yes	No
FEM	Yes	Yes - properties must differ from the surrounding medium	Outside FEM region	Yes
FDTD	Yes – non-dispersive	No	No	Yes, biaxial
PO	Limited - reflection only	Yes	No	No

Solution Method	Dielectric	Layered Dielectric (2D)	Anisotropic Layered Dielectric (2D)	Anisotropic Dielectric (3D)
	No metal			
LE-PO	No	No	No	No
RL-GO	Yes	Yes	No	No
UTD	No	Yes (reflection only)	No	No
Faceted UTD	No	Yes (reflection and transmission)	No	No
CMA	Yes	Yes	Yes	No
CBFM	No	Yes	Yes	No
Dielectric Surface Impedance Approximation	Yes	Yes	No	No
VEP	Yes	Yes	No	No

Table 7: Restrictions when using metallic, impedance sheet, characterised surface and windscreen.

Solution Method	Metallic	Impedance Sheet	Characterised Surface	Windscreen
MoM	Yes	Yes	Yes	Yes
FEM	Yes - inside and outside FEM region	Yes – inside and outside FEM region	No	Yes
FDTD	PEC only	No	No	No
PO	PEC only	No	No	No
LE-PO	PEC only	No	No	No
RL-GO	PEC only	No	Yes	No
UTD	Yes	No	No	No

Solution Method	Metallic	Impedance Sheet	Characterised Surface	Windscreen
Faceted UTD	Yes	Yes (reflection and transmission)	Yes (reflection and transmission)	No
CMA	Yes	Yes	No	Yes
CBFM	Yes	Yes	No	No
Dielectric Surface Impedance Approximation	Yes	Yes	No	Yes
VEP	Yes	Yes	No	No

2.17 Applying Media Settings

Defined media can be applied to the model in various ways. Some media settings are applied to regions, others on faces and wires. The rules for defining media varies between the different solution methods.

2.17.1 Applying a Metal to a Face

Apply a metal to a face bordering a free space or dielectric region.

1. Select the face where you want to apply a metal.
2. From the right-click context menu, select **Properties**.
3. On the **Modify Face** dialog, click the **Properties** tab.

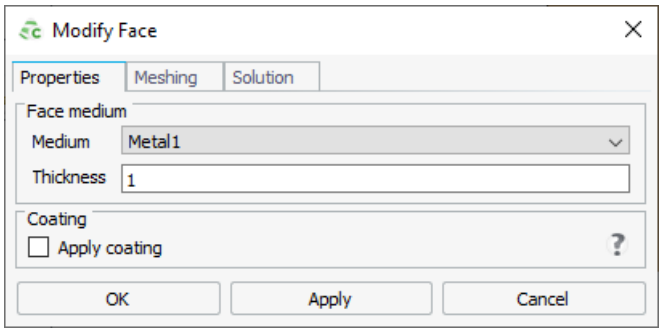


*Figure 146: The **Modify Face** dialog (**Properties** tab).*

4. In the **Medium** drop-down list, select the metal that you want to apply to the face.
5. In the **Thickness** field, enter the metal thickness.
6. Click **OK** to apply the metallic medium and to close the dialog.

> **Note:** Closed regions containing metallic faces cannot be set to (solid) PEC.

2.17.2 Applying a Dielectric to a Region

Apply a dielectric to a region.

1. Select the region where you want to apply a dielectric.
2. From the right-click context menu, select **Properties**.
3. On the **Region properties** dialog, click the **Properties** tab.

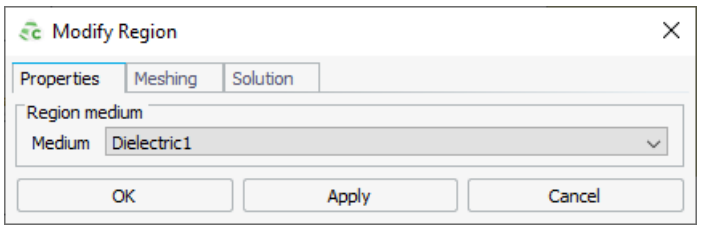


*Figure 147: The **Modify Region** dialog (**Properties** tab).*

4. In the **Medium** drop-down list, select the dielectric that you want to apply to the region.
5. Click **OK** to apply the dielectric and to close the dialog.

2.17.3 Coatings

A coating can be applied to wires or to both sides of a conducting face.

A coating can be applied to a face under the following conditions:

- One side must have free space.
- The other side must have free space or PEC.

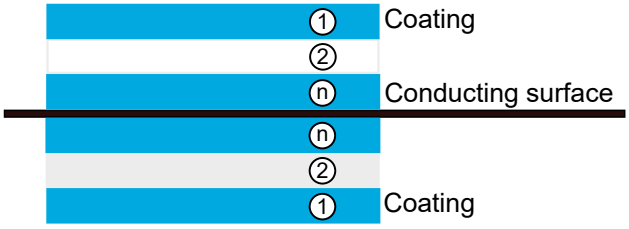


Figure 148: Coatings are applied to both sides of conducting surfaces.

The following coating thickness requirements apply when using the MoM / MLFMM, PO or RL-GO solution methods:

Solution Method	Coating Thickness Requirement	CO Card Equivalent
MoM / MLFMM	Both electrically thin and geometrically thin	Electrically thin surface coating
MoM / MLFMM	Electrically thick, but geometrically thin (single layer).	Dielectric / magnetic surface coating (single layer)

Solution Method	Coating Thickness Requirement	CO Card Equivalent
	Note: • Only for closed structures with a PEC surface and the normal vector pointing towards the source(s). • Coating is applied to both sides of the PEC surface, since fields will be zero where there is no sources.	
PO	Electrically thick, but geometrically thin	Dielectric / magnetic surface coating
RL-GO	Both electrically thick and geometrically thick	Dielectric / magnetic surface coating

Note: A geometrically thin coating must be thin relative to the triangle size (and as a result also to the free space wavelength) as well as the curvature radius of the surface.

Related reference
Coatings and Supported Solution Methods

Applying a Coating to a Wire or Face

Add a surface coating to a wire or to both sides of a conducting face.

1. Create the layered dielectric to use as a coating:
   a) Create the dielectric(s).
   b) Create a layered dielectric to be used as the coating.
2. In the details tree select the wire or face where you want to apply a coating.
3. From the right-click context menu, select **Properties**.

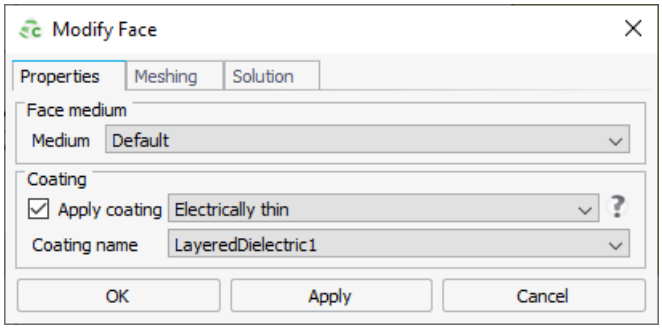


*Figure 149: The **Modify Face** dialog.*

4. On the **Modify Edge** / **Modify Face** dialog, click the **Properties** tab.
5. Under **Coating**, select the **Apply coating** check box and select the coating type.
   - *Electrically thin*
     Use this option to add an electrically thin, multilayer dielectric / magnetic coating.
   - *Electrically thick (single layer only)*
     Use this option to add a single layer, electrically thick, dielectric / magnetic coating to a closed structure with a PEC surface. This is typically used to model radar-absorbing materials (RAM)[30].
   - *Characterised surface*
     Use this option to add a characterised surface as coating.
6. In the **Coating name** field, from the drop-down list, select the layered dielectric or characterised surface to be used as the coating.
   The next step only applies for characterised surfaces.
7. Under , specify the start point and end point for the . The vector is not required to be exactly in the plane of the face, since it is projected onto the face, but it should be approximately parallel to the face.
8. In the **Thickness** field, specify the thickness of the coating.
9. Click **OK** to apply the coating to the wire or face and to close the dialog.

30. A high-shielding coating.

Coatings and Supported Solution Methods

View the restrictions when using a coating in a model, since coatings are not supported for all solution methods.

Table 8: Restrictions when using dielectrics.

Solution Method	Coatings	
	Electrically Thin	Electrically Thick (Single Layer Only)
MoM	Yes	Yes – on closed PEC region
FEM	Yes – inside and outside FEM region	No
FDTD	No	No
PO	Yes	Yes
LE-PO	No	No
RL-GO	Yes	Yes
UTD	Yes	Yes
Faceted UTD	Yes	No
CMA	Yes	No
CBFM	Yes	No
Dielectric Surface Impedance Approximation	No	No
VEP	Yes	No

2.17.4 Thin Dielectric Sheets

A thin dielectric sheet can be applied to one side of a face to model flat multilayer dielectric structures. Typical applications include radome enclosed antennas and automobile windscreens.

Note: A thin dielectric sheet can only be set on faces bordering free space regions.

Important: The order of the dielectric sheet layers is important when using RL-GO or UTD.

For example,consider a model with two layers. One layer is a good absorber and the second is a good conducting layer. When a ray is incident on the side of the absorber, the reflection is zero. When a ray is incident on the side of the conducting layer, the reflection is near perfect. The transmission coefficient in both cases is zero.

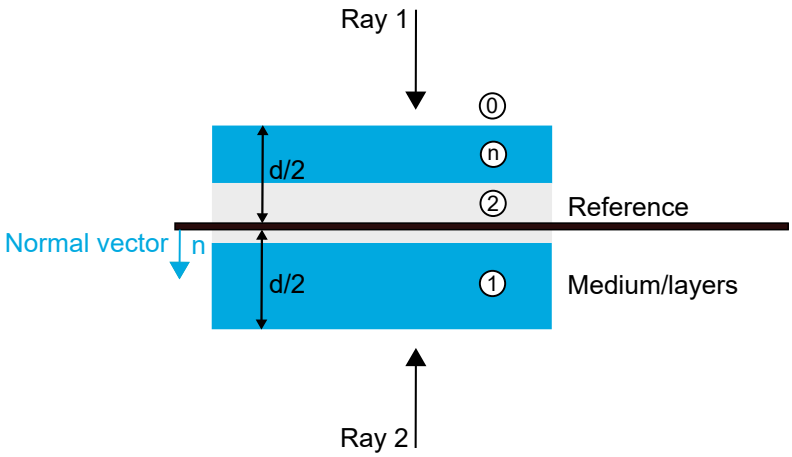


Figure 150: The order of the layers for thin dielectric sheets is important when used in conjunction with RL-GO and UTD.

The definition of which side is the front / rear is determined by the normal vector *n* of the triangles. If one ray is incident in the direction of the normal vector (ray 1) and as a result hits the first layer (index number *n* - layer with the highest index number). An incident ray in the opposite direction of the normal vector (ray 2) will first hit the layer with the lowest index number.

Applying a Thin Dielectric Sheet to a Face

Add a thin dielectric sheet to a face bordering a free space region.

The thin dielectric sheet approximation changes the surface impedance of triangular elements. Only the boundary condition is affected.

1. Create the layered dielectric to use as a thin dielectric sheet:
   a) Define the dielectric(s).
   b) Define the layered dielectric.
2. In the details tree select the face where you want to apply the thin dielectric sheet.
3. From the right-click context menu, select **Properties**.
4. On the **Modify Face** dialog, click the **Properties** tab.

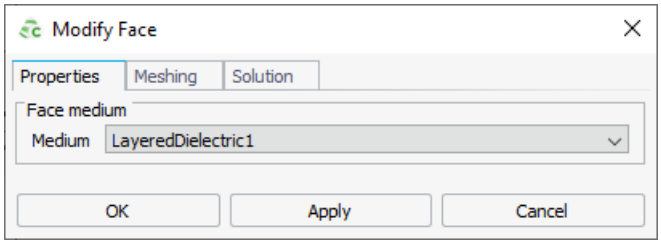


*Figure 151: The **Modify Face** dialog.*

5. Under **Face medium**, in the **Medium** drop-down list, select the layered dielectric medium to be used as the thin dielectric sheet.
6. Click **OK** to apply the thin dielectric sheet to the face and to close the dialog.

2.17.5 Applying a Layered Anisotropic to a Face

Apply a layered anisotropic dielectric to a face bordering a free space or dielectric region.

1. Define a layered anisotropic dielectric.
2. Select the face where you want to apply a layered anisotropic dielectric.
3. From the right-click context menu, select **Properties**.
4. On the **Modify Face** dialog, click the **Properties** tab.

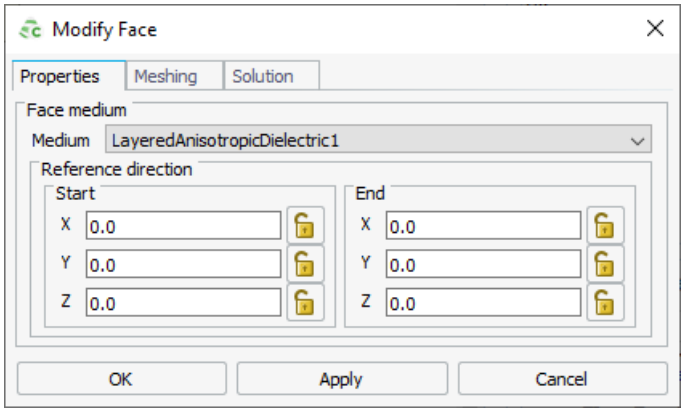


*Figure 152: The **Modify Face** dialog (**Properties** tab).*

5. In the **Medium** drop-down list, select the layered anisotropic that you want to apply to the face.
6. Under **Reference direction**, specify the **Start point** and **End point** to define the principal direction of the layered anisotropic medium.
7. Click the **OK** to apply the layered anisotropic to the face and to close the dialog.

2.17.6 Applying an Anisotropic Dielectric to a Region

Apply an anisotropic dielectric to a region.

> **Note:** When using the FDTD solution method, the anisotropic medium orientation is defined in the global coordinate system. For FEM, the orientation can be defined in a local coordinate system.

1. Define an anisotropic (3D) medium.
2. Select the region where you want to apply the anisotropic dielectric.
3. From the right-click context menu, select **Properties**.
4. On the **Modify Region** dialog, click the **Properties** tab.

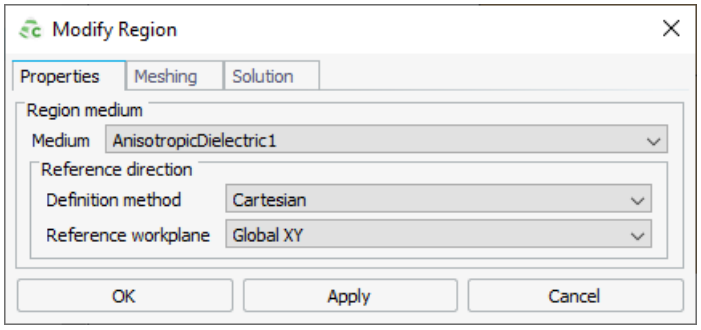


*Figure 153: The **Modify Region** dialog (**Properties** tab).*

5. In the **Medium** drop-down list, select the layered anisotropic that you want to apply to the face.
6. In the **Definition method** drop-down list, select the coordinate system in which the medium orientation is defined (only for FEM solution method).
   - **Cartesian**
   - **Cylindrical**
   - **Spherical**
7. In the **Reference workplane** drop-down list, select the workplane in which the medium orientation is defined.
8. Click **OK** to apply the anisotropic medium to the region and to close the dialog.

2.17.7 Applying an Impedance Sheet to a Wire or a Face

Apply a surface impedance to a wire or a face bordering a free space or dielectric region.

As an example, an impedance sheet is applied to a face. The steps are similar for applying an impedance sheet to a wire.

1. Define an impedance sheet.
2. Select the face where you want to apply a surface impedance.
3. From the right-click context menu, select **Properties**.
4. On the **Modify Face** dialog, click the **Properties** tab.

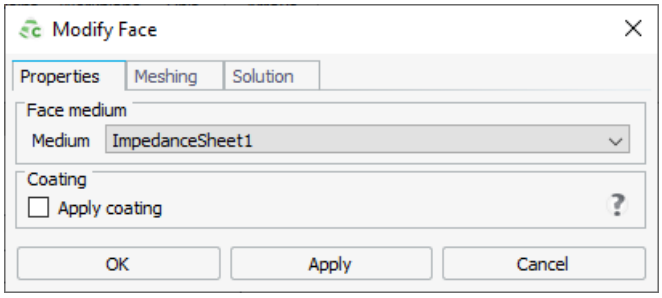


*Figure 154: The **Modify Face** dialog (**Properties** tab).*

5. In the **Medium** drop-down list, select the impedance sheet that you want to apply to the face.
6. Click the **OK** to apply the impedance sheet and to close the dialog.

2.17.8 Applying a Windscreen Layer to a Face

Apply a windscreen layer to a face bordering a free space or dielectric region.

1. Define a windscreen layer.
2. Select the face where you want to apply a windscreen layer.
3. From the right-click context menu, select **Properties**.
4. On the **Modify Face** dialog, click the **Solution** tab.

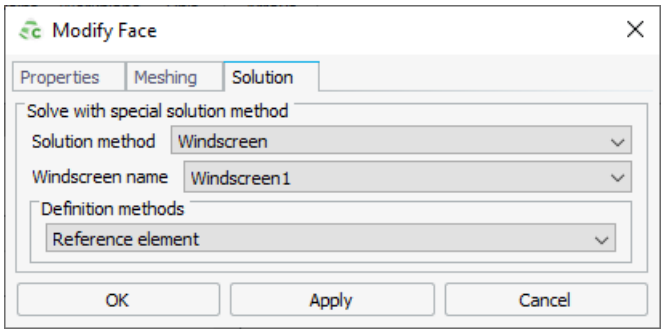


*Figure 155: The **Modify Face** dialog (**Solution** tab).*

5. In the **Solve with special solution method** group, in the **Solution method** drop-down list, select **Windscreen**.
6. In the **Windscreen name** drop-down list, select the windscreen layer that you want to apply to the face.
7. In the **Definition methods** drop-down list, select one of the following:
   - To define the curvature reference for the windscreen, select **Reference element**.
   - To define the metallic antenna elements for the windscreen, select **Windscreen solution element**.
8. Click **OK** to apply the windscreen layer to the face and to close the dialog.

2.17.9 Applying a Characterised Surface to a Face

Apply a characterised surface to a face bordering free space or dielectric regions.

> **Note:** Characterised surfaces are only supported in conjunction with the RL-GO or MoM/MLFMM solution methods.

When a characterised surface definition is applied to a face, you must specify a vector to ensure the correct surface orientation. The U-Vector should be set to point into the direction of the U-Vector (or **X vector** in global coordinates) of the original characterised surface. This characterisation is performed either through solving with periodic boundary conditions, an infinite ground plane or measurements. The projection of the U-Vector onto the face correspond to the U-Vector (or principal direction) of the original characterised surface.

The orientation of the U-Vector is only important when the characterised surface is anisotropic (properties dependent on the plane wave angle of incidence). Isotropic surfaces do not depend on the orientation of the U-Vector. Consequently the only requirement for the U-Vector is that there should be a valid projection of the vector onto the face. In essence this means that the U-Vector is not allowed to point into the direction of the face normal.

> **Note:** The face normal vector is the vector that is perpendicular to the face.
>
> - For flat faces, the normal is the same everywhere on the face.
> - For curved faces, the normal changes as a function of the position on the face.

Curved surfaces such as radomes have to be split into smaller faces so that a valid U-Vector can be defined for each surface. As an illustration, consider a sphere. There is no single vector that has a valid projection onto the surface of a sphere, since at two points, the vector points in the direction of the face normal.

1. Select a face to apply a characterised surface.
2. From the right-click context menu, select **Properties**.
3. On the **Modify Face** dialog, click the **Properties** tab.

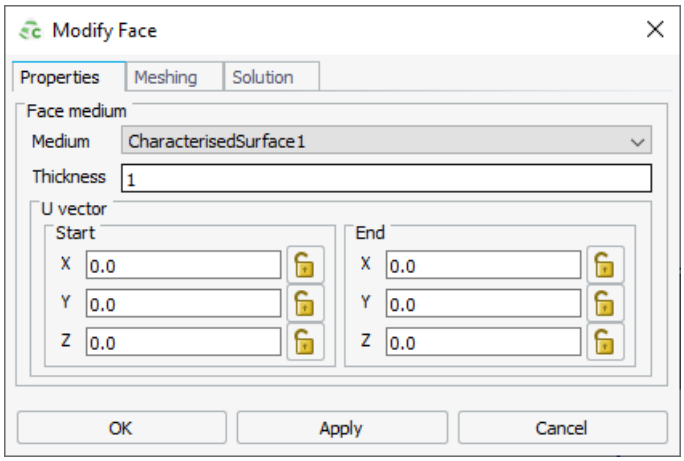


*Figure 156: The **Modify Face** dialog (**Properties** tab).*

4. In the **Medium** drop-down list, select the characterised surface to apply to the face.
5. In the **Thickness** field, specify the thickness of the characterised surface (only supported for MoM/MLFMM).

The U-Vector is defined as the reference direction projected onto the face.

6. Under U-Vector, specify the start point and end point for the U-Vector. The vector is not required to be exactly in the plane of the face, since it is projected onto the face, but it should be approximately parallel to the face.

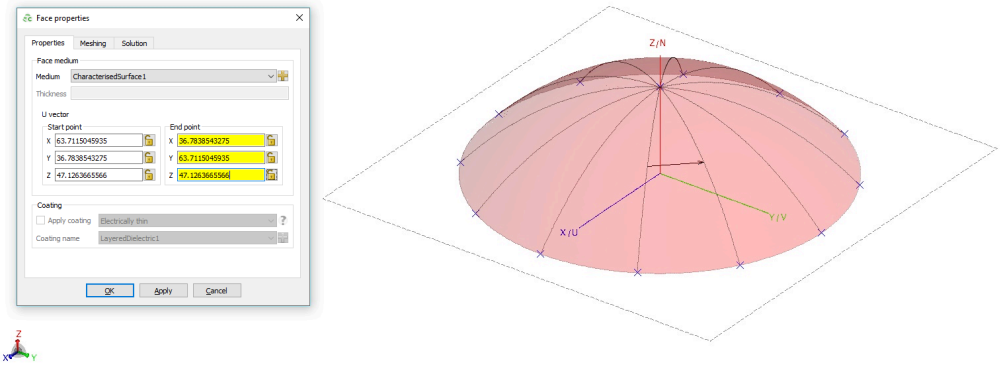


Figure 157: The display in CADFEKO when setting the U-Vector. Opacity settings were modified in order to see the U-Vector preview.

7. Click **OK** to apply the characterised surface and to close the dialog.
The U-Vector can be displayed in POSTFEKO to verify that all faces have the correct settings and U-Vector orientations applied.

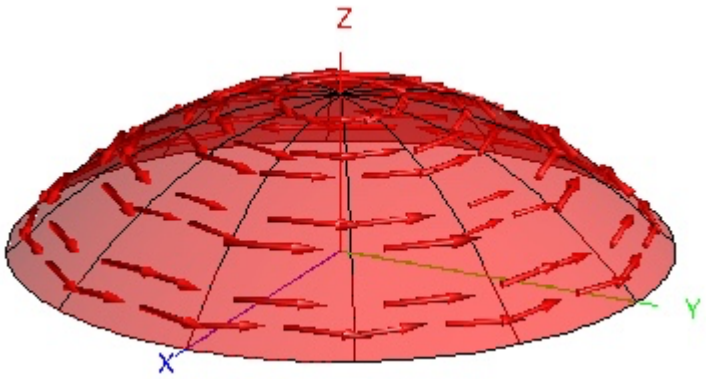


Figure 158: Characterised surface orientation displayed in POSTFEKO where each face has a different U-Vector applied.

Related concepts

Ray Launching Geometrical Optics (RL-GO)

2.17.10 Creating a Slot in a Face Using Aperture Triangles

Define an aperture or slot in an infinite ground plane using aperture triangles.

1. Select the face where you want to apply the aperture triangles.
2. From the right-click context menu, select **Properties**.
3. On the **Modify Face** dialog, click the **Solution** tab.

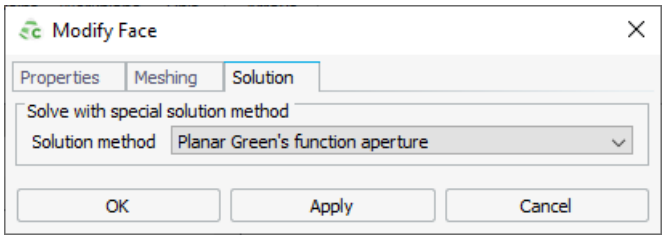


*Figure 159: The **Modify Face** dialog (**Solution** tab).*

4. Under **Solve with special solution method**, in the drop-down list select **Planar Green's function aperture**.
5. Click **OK** to apply the aperture triangles and to close the dialog.

2.18 Periodic Boundary Condition (PBC)

Use a periodic boundary condition (PBC) to analyse infinite periodic structures. A typical application of PBC is to analyse frequency selective surface (FSS) structures.

The unit-cell definition for the periodic boundary condition solution is based on vectors. For the one-dimensional case, the start point and end point of a single vector are required. Periodicity is defined based on two planes perpendicular to the vector formed between them. The vector used to define one-dimensional periodicity can have any orientation but must have a non-zero length.

For the two-dimensional case, two vectors are required. These vectors form the two boundaries of the unit cell which is infinite in the direction normal to the plane in which both vectors lie. The vectors that define the unit-cell for two-dimensional periodicity must have non-zero length, and cannot be oriented in the same direction.

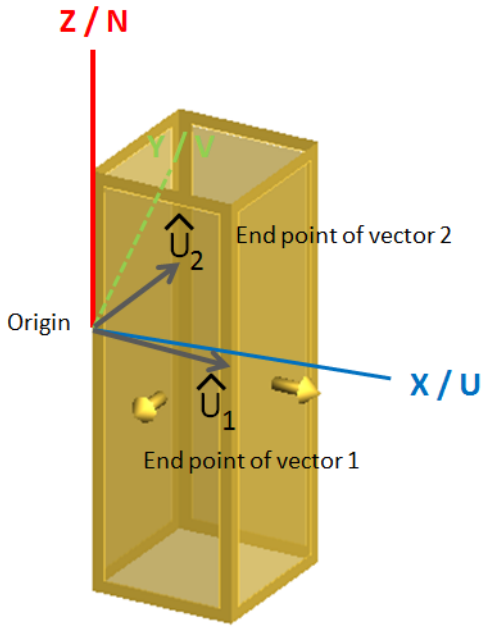


Figure 160: The periodic boundary condition for two-dimensional periodicity.

A phase shift can be applied in the direction of the vectors defining the unit-cell. The specified values for the phase-shift are only used if a plane wave source is not present.

> **Note:** If a plane wave source is present and a phase is specified, the Solver will return an error during the solution.

For array modelling using periodic boundary conditions, the beam (squint) angle is specified by defining the theta and phi angle. The phase along the periodic lattice vectors is computed automatically to ensure the specified beam direction.

Related reference
Supported Solution Method and Technique Combinations

2.18.1 Defining a Periodic Boundary Condition (PBC)

Specify a periodic boundary condition (PBC) to analyse infinite periodic structures.

1. On the **Construct** tab, in the **Structures** group, click the **Planes/Arrays** icon. From the drop-down list, select **Periodic Boundary Conditions**.

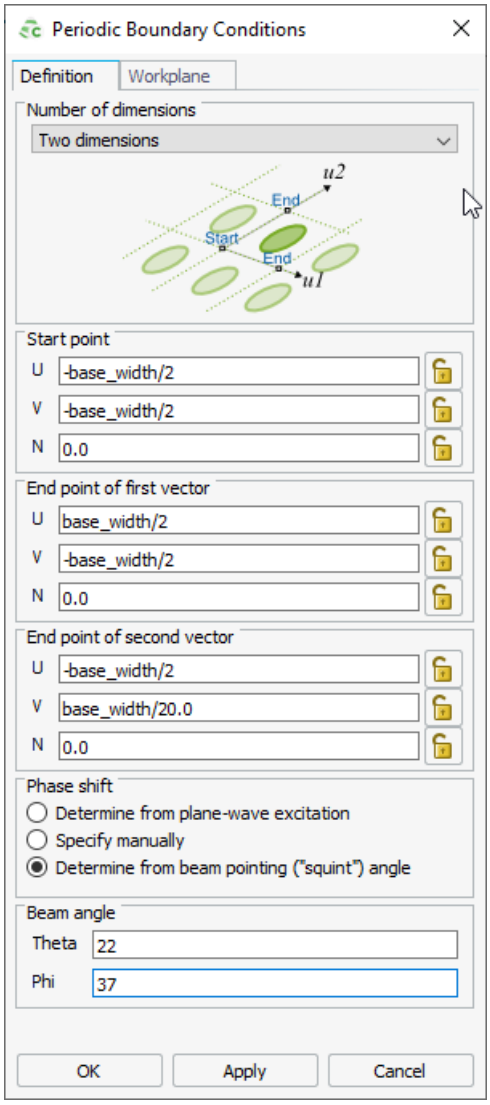


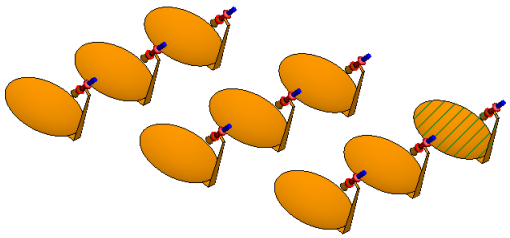
*Figure 161: The **Periodic Boundary Conditions** dialog.*

2. From the **Number of dimensions** list, select one of the following:
   - To create a one-dimensional PBC where the unit cell is repeated along a line, select **One dimension**.
   - To create a two-dimensional PBC where the unit cell is repeated to form a surface, select **Two dimensions**.
   - To remove the PBC from the model, select **No periodic boundary**.
3. Under **Start point**, specify the start point of the vector.

4. Under **End point of first vector**, specify the end point of first vector.
5. Under **End point of second vector**, specify the end point of the second vector.
6. Under **Phase shift**, select one of the following:
   - When a plane wave is used as excitation, the phase difference between the cells cannot be specified. To determine the phase shift of the excitation, select **Determine from plane-wave excitation**.
   - To specify the phase shift, select **Specify manually**.
     - In the **u1** field, specify the phase shift in the first direction, u1.
     - In the **u2** field, specify the phase shift in the second direction, u2.
   - To specify the theta and phi angle of the "squint" angle, select **Determine from beam pointing (squint) angle**.
     - In the **Theta** field, specify the theta angle of the "squint" angle.
     - In the **Phi** field, specify the phi angle of the "squint" angle.
7. Click **OK** to define the PBC and to close the dialog.

2.19 Finite Antenna Arrays

Create an arbitrary finite antenna array that consists of an array of contributing elements, either with direct feeds for each element or via indirect coupling, and solve with the efficient domain Green's function method (DGFM).



Create the base element (antenna) and create a linear, planar, cylindrical or circular finite antenna array with ease. Add custom antenna array elements to create complex and arbitrary finite antenna array structures.

The DGFM solver considers the self-coupling, mutual coupling and the edge effects of the finite array, but only uses the computational resources equivalent to solving the base element.

Note: The base element encompasses all structures in the model when creating a finite antenna array (except for infinite planes).

Related reference
Supported Solution Method and Technique Combinations

2.19.1 Creating a Linear / Planar Antenna Array

Create a linear or planar finite antenna array model.

1. On the **Construct** tab, in the **Structures** group, click the **Planes/Arrays** icon. From the drop-down list, select the **Linear/Planar Array** icon.
2. Specify the elements in the U dimension.
   a) Under **U dimension**, in the **Number of elements** field, specify the number of array elements.
   b) Under **U dimension**, in the **Offset along U axis** field, specify the offset between the array elements.
3. Specify the elements in the V dimension.
   a) Under **V dimension**, in the **Number of elements** field, specify the number of array elements.
   b) Under **V dimension**, in the **Offset along V axis** field, specify the offset between the array elements.
4. In the **Label** field, add a unique label for the antenna array.

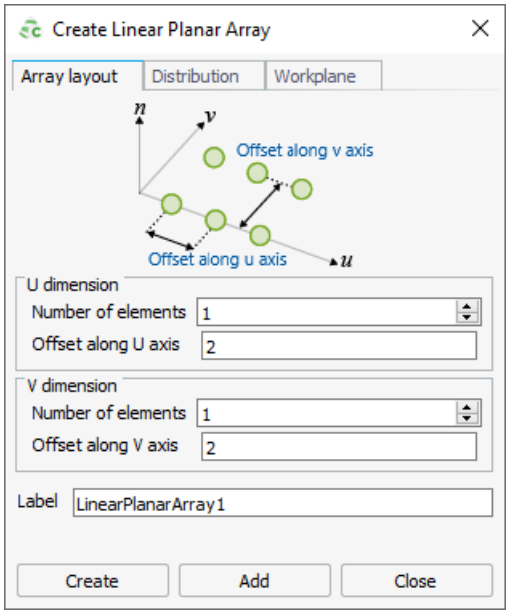


*Figure 162: The **Create Linear Planar Array** dialog.*

5. Click the **Distribution** tab to specify the array distribution.
   - To create an antenna array where the distribution is calculated from the plane wave (if a plane wave is present in the model), click the **Uniform distribution or calculated from plane wave** check box.
   - To create an antenna array with a specified excitation for each element, clear the **Uniform distribution or calculated from plane wave** check box.

     Note: When specifying each element, take note of the array element indexing.

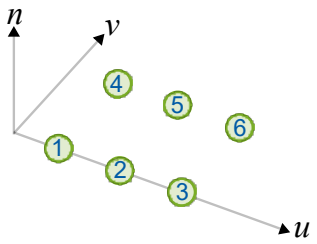
     

     Figure 163: Image depicting the array element indexing.

     - To specify the magnitude scaling and phase offset manually:
       1. In the **Magnitude scaling** field, specify the excitation magnitude for the individual element relative to the base element.
       2. In the **Phase offset (degrees)** field, specify the phase offset (in degrees) for the individual element relative to the base element.
     - To specify the magnitude scaling and phase offset by importing the points from file, click **Import**.

- In the **File name** field, browse for the file you want to import.
- [Optional] In the **Scale by** field, enter a value to scale the points.
- Under **Delimiter**, click the delimiter type you use in your file.
- Click **OK** to close the **Import Points** dialog.

2.19.2 Creating a Cylindrical / Circular Antenna Array

Create a cylindrical or circular finite antenna array model.

1. On the **Construct** tab, in the **Structures** group, click the **Planes/Arrays** icon. From the drop-down list, select the **Cylindrical/Circular Array** icon.
2. In the **Radius** field, enter the radius of the cylindrical / circular antenna array.
3. Specify the elements in the phi dimension.
   a) Under **Phi dimension**, in the **Number of elements** field, specify the number of array elements.
   b) Under **Phi dimension**, in the **Specify increment** field, specify the offset between the array elements.
4. Specify the elements in the N dimension.
   a) Under **N dimension**, in the **Number of elements** field, specify the number of array elements.
   b) Under **N dimension**, in the **Offset along N axis** field, specify the offset between the array elements.
5. Specify the element rotation.
   - To place the array elements with the same orientation as the base element, clear the **Element orientation** check box.
   - To rotate the array elements sequentially, select the **Element orientation** check box.
6. In the **Label** field, add a unique label for the antenna array.

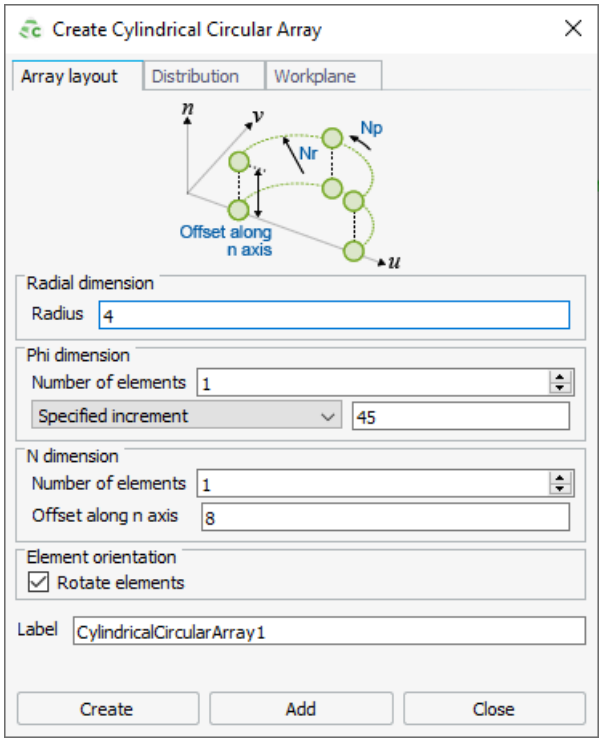


*Figure 164: The **Create Cylindrical Circular Array** dialog.*

7. Click the **Distribution** tab to specify the array distribution.

- To create an antenna array where the distribution is calculated from the plane wave (if a plane wave is present in the model), click the **Uniform distribution or calculated from plane wave** check box.
- To create an antenna array with a specified excitation for each element, clear the **Uniform distribution or calculated from plane wave** check box.

> **Note:** When specifying each element, take note of the array element indexing.

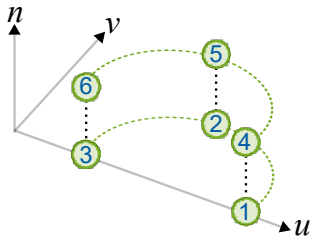


Figure 165: Image depicting the array element indexing.

- To specify the magnitude scaling and phase offset manually:
  1. In the **Magnitude scaling** field, specify the excitation magnitude for the individual element relative to the base element.

2. In the **Phase offset (degrees)** field, specify the phase offset (in degrees) for the individual element relative to the base element.

- To specify the magnitude scaling and phase offset by importing the points from file, click **Import**.
  - In the **File name** field, browse for the file you want to import.
  - [Optional] In the **Scale by** field, enter a value to scale the points.
  - Under **Delimiter**, click the delimiter type you use in your file.
  - Click **OK** to close the **Import Points** dialog.

2.19.3 Creating a Custom Array Element

Create a custom array element. Use a custom array element to create an irregular-spaced array.

1. On the **Construct** tab, in the **Structures** group, click the **Planes/Arrays** icon. From the drop-down list, select the **Custom Array Element** icon.
2. Under **Origin**, enter the position of the workplane using one of the following methods:
   - Enter the coordinates for the origin manually.
   - Use point entry to enter the coordinates for the origin from the 3D view.
3. Under **Excitation**, in the **Magnitude scaling** field, enter the excitation magnitude for the element.
4. Under **Excitation**, in the **Phase offset (degrees)** field, enter the phase offset for the element in degrees.
5. In the **Label** field, add a unique label for the antenna array.

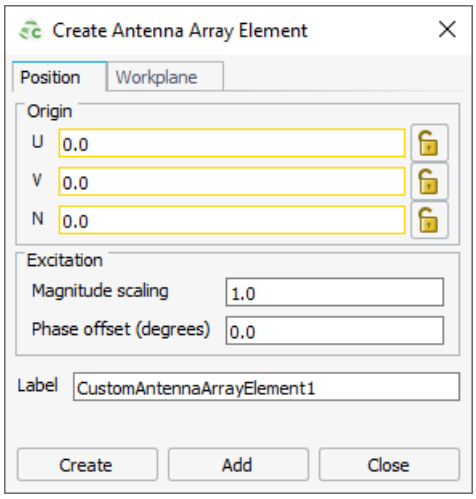


*Figure 166: The **Create Antenna Array Element** dialog.*

2.19.4 Converting an Array to a Custom Array

Convert a linear, planar, cylindrical or circular antenna array to a custom array as a starting point to create complex and regular-spaced or irregular-spaced array elements.

1. In the model tree, click the linear, planar, cylindrical or circular antenna array that you want to convert to custom antenna array elements.
2. From the right-click context menu, select **Convert to Custom Array**.
   The individual array elements from the original antenna array are converted to custom array elements.

2.19.5 Finite Antenna Array Solver Settings

View the solver settings applicable to finite antenna arrays.

On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon. Click the **Domain decomposition** tab.

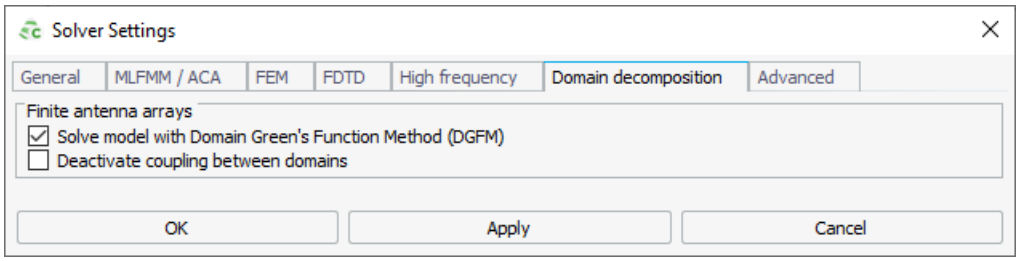


*Figure 167: The **Solver Settings (Domain decomposition)** dialog.*

Solve model with Domain Green's Function Method (DGFM)
Select the **Solve model with Domain Green's Function Method (DGFM)** check box to solve the model with the faster finite antenna array solution method. Clear the check box to solve the model using a full wave solution method (for example MoM or MLFMM)

> **Tip:** Clear the check box to do comparisons at specific frequencies over a frequency range.

Deactivate coupling between domains
Select the **Deactivate coupling between domains** check box to ignore the mutual coupling between the antenna array elements.

> **Tip:** Use this option when the coupling between array elements is negligible.

Related reference
Finite Antenna Array Limitations

2.19.6 Base Element Display

View the original base element and the full finite antenna array in the 3D view.

After a finite antenna array has been created, the base (original) element is indicated by green hatching in the 3D view.

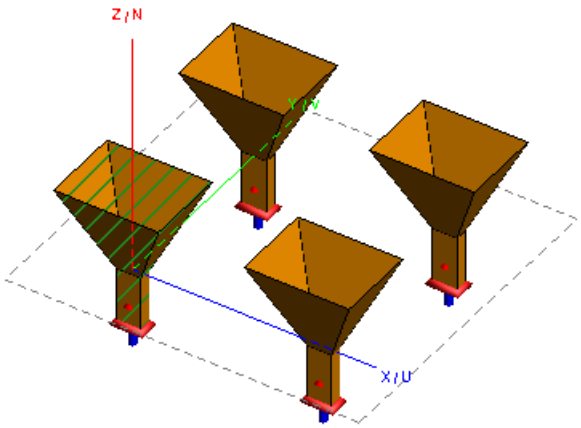


Figure 168: The base (original) element is indicated by green hatching in the 3D view.

Note: A large finite array with non-uniform distribution will affect the 3D rendering and performance in POSTFEKO and in CADFEKOwhen modifying large arrays.

2.20 Windscreen Tools

Use the windscreen tools to define a curved reference surface constrained by a cloud of points, normals and optional *U'V'* parameters. The constrained surface is then used as a reference to create a work surface where windscreen layers and curved parameterised windscreen antenna elements can be created.

Related reference
Supported Solution Method and Technique Combinations

2.20.1 Parametrised Windscreen Antenna Elements

View the workflow for creating curved parameterised windscreen antenna elements.

1. Import windscreen glass and antenna boundary.
   - The antenna boundary will be used as the outline for the constrained surface.
2. Project antenna boundary onto windscreen surface.
   - To ease snapping, the antenna boundary and windscreen surface should be a single part. Union the windscreen and antenna boundary. If the antenna boundary is not coincident with the windscreen surface, project its outline onto the windscreen surface.
3. Create a constrained surface.
   - The windscreen surface is approximated by a constrained surface specified by a cloud of points, normals and optional *U'V'* parameters.
4. Create a work surface.
   - A work surface is created from the constrained surface. The work surface is used to define surface curves in the *U'V'* parameter space of this surface.
5. Create windscreen antenna elements.
   - Windscreen antenna elements are defined in the specified work surface using *Surface lines*, *Surface Bézier curves* and *Surface regular lines* curves.
6. Mesh the model.
   - To prevent the constrained surface from being meshed, exclude the constrained surface in the model.

2.20.2 Accessing the Windscreen Tab on the Ribbon

The **Windscreen** tab is not displayed on the ribbon by default. To access the **Windscreen** tab, you must configure the ribbon to show the tab.

On the **Home** tab, in the **Extensions** group, click the **Windscreen** icon.

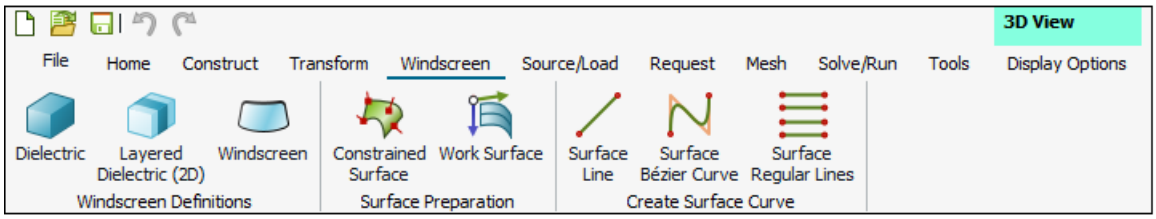


Figure 169: The ribbon with Windscreen tab selected.

When the **Windscreen** tab is enabled, it is displayed between the **Transform** tab and **Source/Load** tab.

2.20.3 Constrained Surface

A constrained surface is a reference surface constrained by a cloud of points, normals and optional *U'V'* parameters. Use a constrained surface as a reference to create windscreen layers and curved parameterised windscreen antenna elements.

Surface parameters (*U'* and *V'*) define the outline and internal grid lines of a constrained surface.

Define the same *U'* value at a set of points to force a single *U'* grid line to flow through the points. Similarly, define the same *V'* value at a set of points to force a single *V'* grid line.

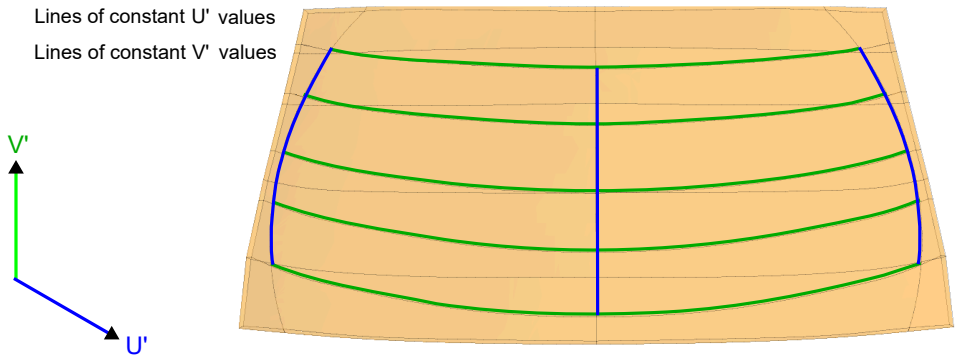


Figure 170: Constrained surface with lines of constant U' and V' values highlighted.

The range of values assigned to the *U'* or *V'* grid lines only determine relative distances in the *U'V'* space.

> For example, setting the range of the *U'* grid lines as -1...1, 0...1 or 0...100 will only influence the relative *U'* distance between the adjacent *U'* grid lines.

Creating a Constrained Surface

Define a constrained surface that will be used to create windscreen layers and curved parametrised windscreen antenna elements.

Before creating a constrained surface:

1. Enable the **Windscreen** tab on the ribbon.
2. Import the windscreen glass and antenna boundary.
3. Union the windscreen glass and antenna boundary.
4. If the antenna boundary does not lie on the windscreen surface, project its outline onto the windscreen surface.

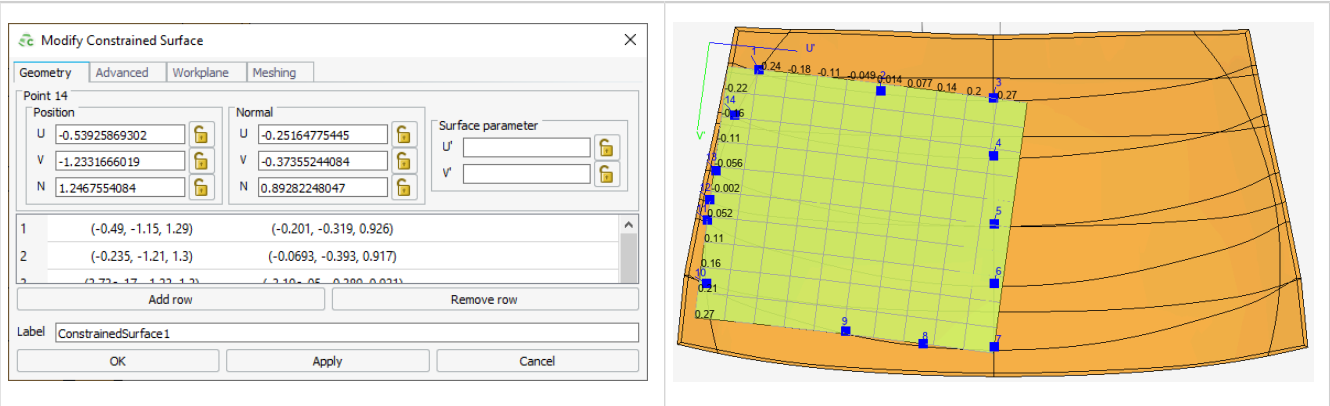
If the prerequisite steps have been executed, then proceed as follows:

1. On the **Windscreen** tab, in the **Surface Preparation** group, click the **Constrained Surface** icon.

Specify the outline of the constrained surface.

2. Use point-entry to add points to the table by snapping to points on the windscreen outline.

Table 9: Creating a constrained surface.



> **Tip:** For the moment ignore the **Surface parameter** column. The surface parameters (*U'* and *V'*) control the flow of the grid lines and are specified in Step 5 to Step 8.

A blue square indicates a point added to the table. A red square indicates the current selected point in the table. Its number corresponds to its location in the table.

The preview of the constrained surface is indicated in green.

> **Note:** The windscreen was offset slightly away from the camera to improve rendering.

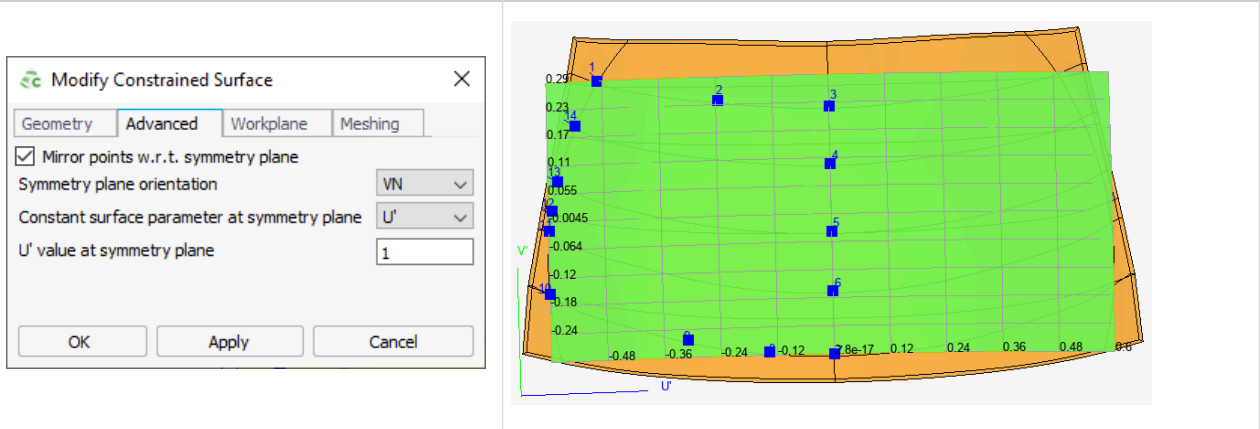
If the windscreen is symmetric, you only need to specify points on half of the windscreen.

3. [Optional] Click the **Advanced** tab.
   a) Check the **Mirror points w.r.t symmetry plane** check box.

b) Specify the symmetry plane for the windscreen by either clicking **UV**, **UN** or **VN**.

c) Define the surface parameter (*U'* or *V'*) which is constant at the symmetry plane and its value at the symmetry plane.

Table 10: Mirroring the constrained surface.



4. Click the **Geometry** tab.

[Optional] Specify the *U'* (or *V'*) value at the symmetry plane.

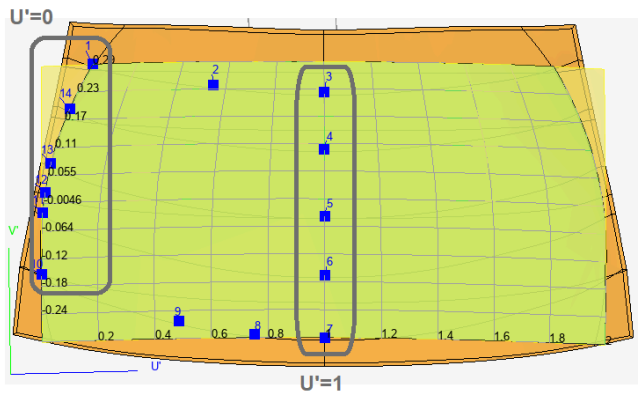
5. If the *U'* (or *V'*) value at the symmetry plane was defined in Step 3.c, specify this value at the points on the symmetry grid line.
For this example, the **Constant surface parameter at plane** is specified as *U'*=1.

Points 3, 4, 5, 6 and 7 on the symmetry plane are set to *U'*=1.

Define the *U'* values for the points to control the left and right grid lines.

6. Start at a corner point on and enter a *U'* value. Repeat for remaining points on the left grid line and right grid line.

Table 11: Setting the constant U' parameters for the constrained surface.



Number	*U'*	*V'*
3	**1**	
4	**1**	
5	**1**	
6	**1**	
7	**1**	
1	**0**	
14	**0**	

Number	*U′*	*V′*
13	**0**	
12	**0**	
11	**0**	
10	**0**	

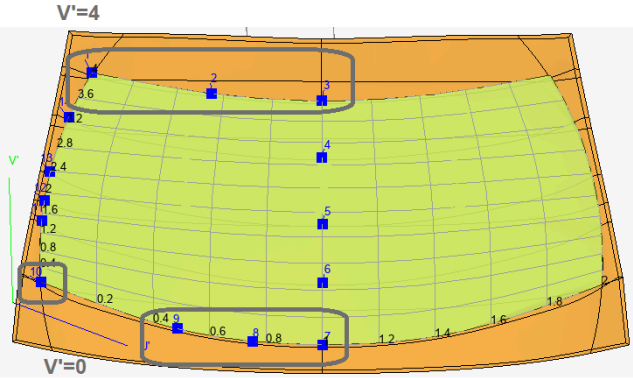
For this example, corner point 1 and points 14, 13, 12, 11 and 10 are set to $U'=0$.

Points 3, 4, 5, 6 and 7 were defined in Step 5 as it is located on the symmetry plane.

Define the *V′* values for the points to control the top and bottom grid lines.

7. Start at a corner point and enter a *V′* value. Repeat for the remaining points on the top grid line and bottom grid line.

Table 12: Setting the constant V' parameters for the constrained surface.



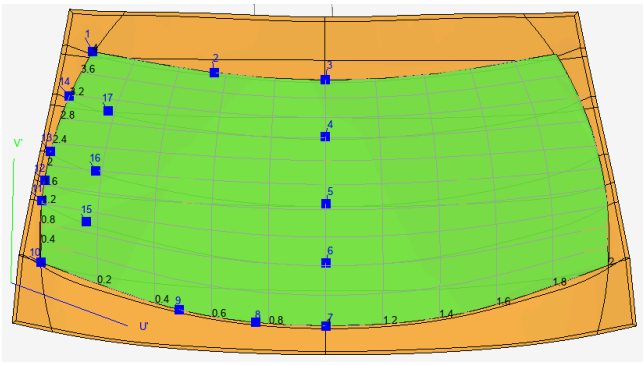
Number	*U′*	*V′*
10		**0**
9		**0**
8		**0**
7	1	**0**
1	0	**4**
2		**4**
3		**4**

For this example, corner point points 10, 9, 8 and 7 are set to $V'=0$.

Points 1, 2 and 3 are set to $V'=4$.

8. [Optional] Add additional internal points to ensure the internal grid lines follow the imported guidelines.
For this example, points, 15, 16 and 17 were added.

Table 13: Setting optional additional internal points for the constrained surface.



Number	*U'*	*V'*
11	0	1
15		1
13	0	2
16		2
17		3
14	0	3

9. Click **Create** to create the constrained surface and to close the dialog.

Related tasks
Accessing the Windscreen Tab on the Ribbon

2.20.4 Creating a Work Surface

Create a work surface from a constrained surface. The work surface is used to define surface curves in the *U'V'* parameter space of this surface.

1. On the **Windscreen** tab, in the **Surface Preparation** group, click the **Work Surface** icon.
2. With the **Reference face** field active, click on the constrained surface in the 3D view.

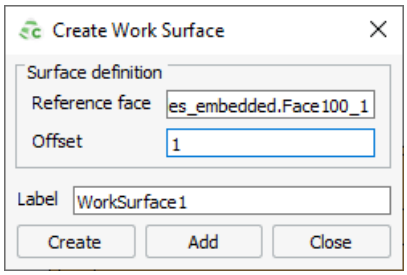


Figure 171: The Create Work Surface dialog.

3. In the **Offset** field, specify the work surface offset from the constrained surface.
4. Click **Create** to create the work surface and to close the dialog.
The defined work surface is displayed in the model tree.

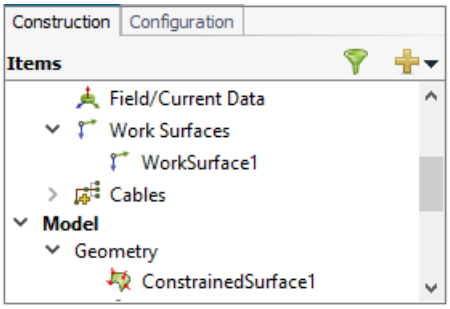


Figure 172: Snippet of the model tree after creating the work surface.

Related tasks
Accessing the Windscreen Tab on the Ribbon

2.20.5 Defining Windscreen Elements on a Curved Work Surface

Define curved parameterised windscreen antenna elements on a specified work surface by using surface lines, surface Bézier curves or an array of linearly spaced lines.

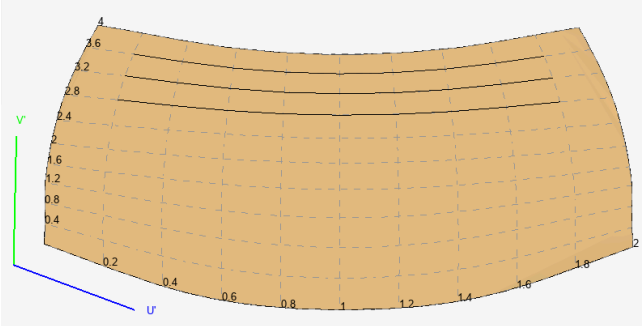


Figure 173: A preview of regular spaced surface lines on a curved work surface.

1. Create a line in the curved work space.
   a) On the **Windscreen** tab, in the **Create Surface Curve** group, click the **Surface Line** icon.

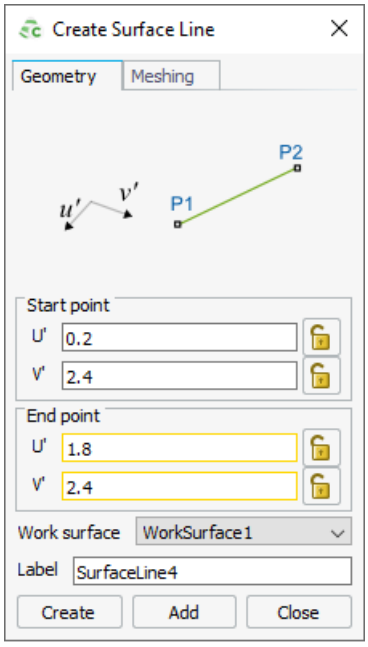


Figure 174: The Create Surface Line dialog.

b) Specify the **Start point**, **End point** and **Work surface**.

c) Click **Create** to create the surface line and to close the dialog.

2. Create a surface Bézier curve in the curved workspace.

a) On the **Windscreen** tab, in the **Create Surface Curve** group, click the **Surface Bézier Curve** icon.

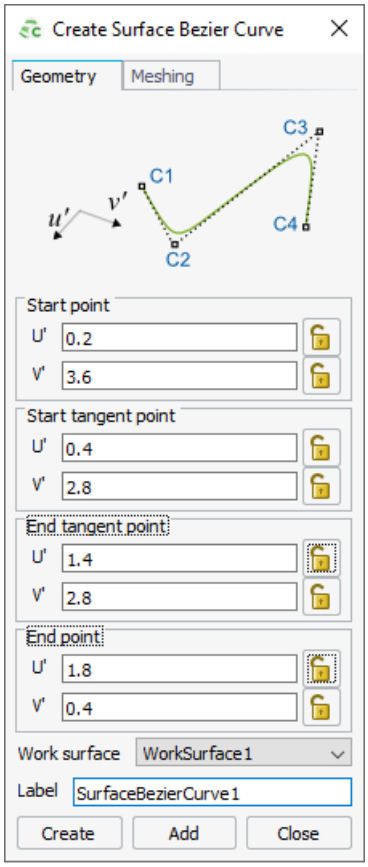


Figure 175: The Create Surface Bezier Curve dialog.

b) Specify the **Start point**, **Start tangent point**, **End tangent point**, **End point** and **Work surface**.

c) Click **Create** to create the surface Bézier curve and to close the dialog.

3. Create an array of linearly-spaced lines in the curved workspace.

a) On the **Windscreen** tab, in the **Create Surface Curve** group, click the **Surface Regular Lines** icon.

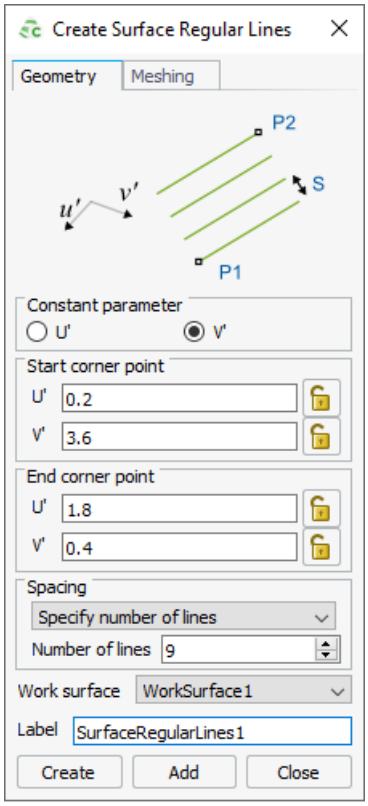


Figure 176: The Create Surface Regular Lines dialog.

b) Specify the **Constant parameter**, **Start corner point**, **End corner point**, **Number of lines** and the **Work surface**.

c) Click **Create** to create the regular spaced surface lines and close the dialog.

Related tasks

Accessing the Windscreen Tab on the Ribbon

2.21 Cables

Many electromagnetic compatibility and interference problems involve cables that either radiate, are irradiated or cause coupling into other cables, devices or antennas. Use the cable modelling tool and solver to analyse the coupling and radiation.

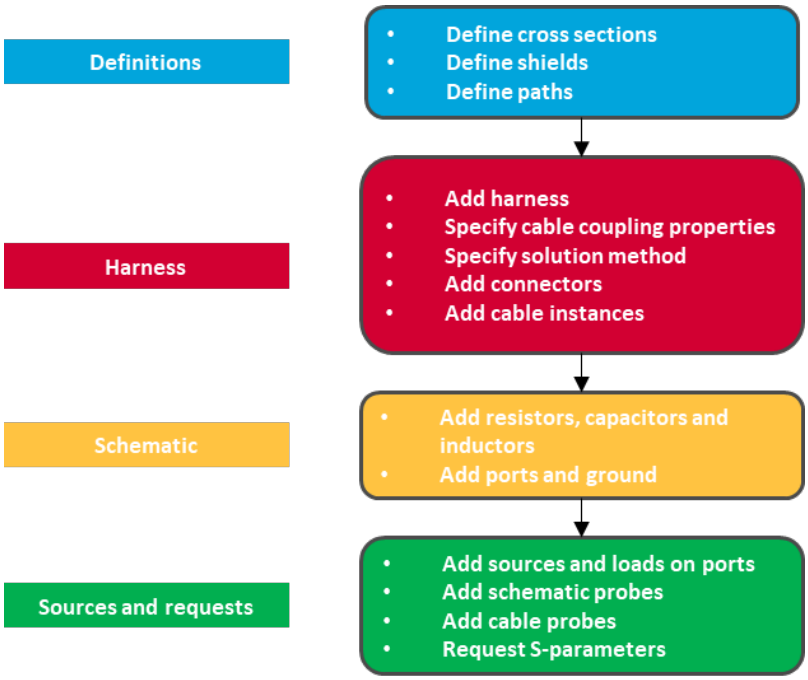
The following terminology is used:

Cable instance
: A cable instance is a single cable (for example, coaxial cable or cable bundle) with a start connector and an end connector, routed along a cable path.

Cable harness
: A cable harness is a collection of cable instances along a specific cable path with a specified solution method and cable coupling parameters.

2.21.1 Workflow for Analysing Cables

The general process is explained for setting up a complete cable analysis.



1. Define a cable instance.
   a. Define the cable type or the cable cross section (for example, a cable bundle).
   b. [Optional] Define the cable shield.
   c. Define the cable path.
   d. Define the start connector and end connector.
   e. Define the cable instance.

2. Define the cable harness.
   a. Specify the relevant cable path and view the cable instances routed along this path.
   b. Specify the solution method for the outer cable problem.
   c. Specify the cable coupling parameters.
3. Open the cable schematic view for each harness.
   a. Add the circuit elements (for example, resistors, capacitors, inductors, SPICE circuits, ports, ground) and connect the connector pins to the circuit elements.
4. Define the sources, loads and requests.
   a. Add sources and loads to the cable ports.
   b. Add schematic probes, cable probes or S-parameter requests to request results.

Related tasks
Defining a Cable Instance
Defining a Cable Harness

2.21.2 Accessing the Cables Tab on the Ribbon

Open the **Cables** tab on the ribbon to access advanced tools related to defining cable harnesses.

By default, the **Cables** tab is not displayed on the ribbon. To access the **Cables** tab, you must configure the ribbon to show the tab.

On the **Home** tab, in the **Extensions** group, click the **Cables** icon.

When the **Cables** tab is enabled, it is located on the ribbon between the **Transform** tab and **Source/Load** tab.

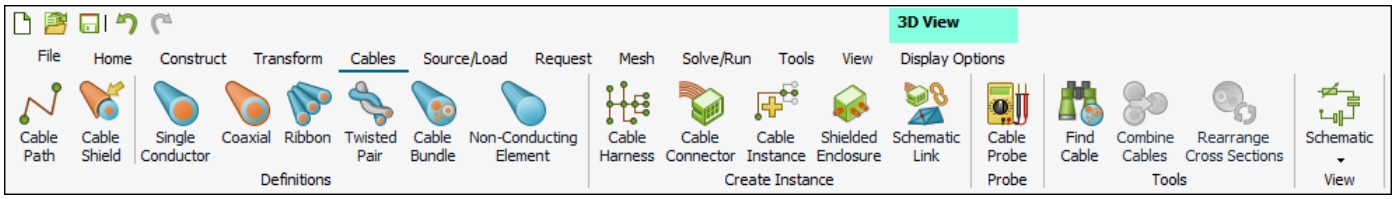


*Figure 177: The ribbon in **CADFEKO** (**Cables** tab).*

2.21.3 Harness Description List (KBL) Specification

Cable harnesses are becoming increasingly complex with innovations in the automotive industry. Import a complex cable harness from a `.kbl` file using the "harness description list" (KBL) specification.

The following KBL[31] entities are supported:

- Coordinates

31. Kabelbaumliste, the German translation for "harness description list".

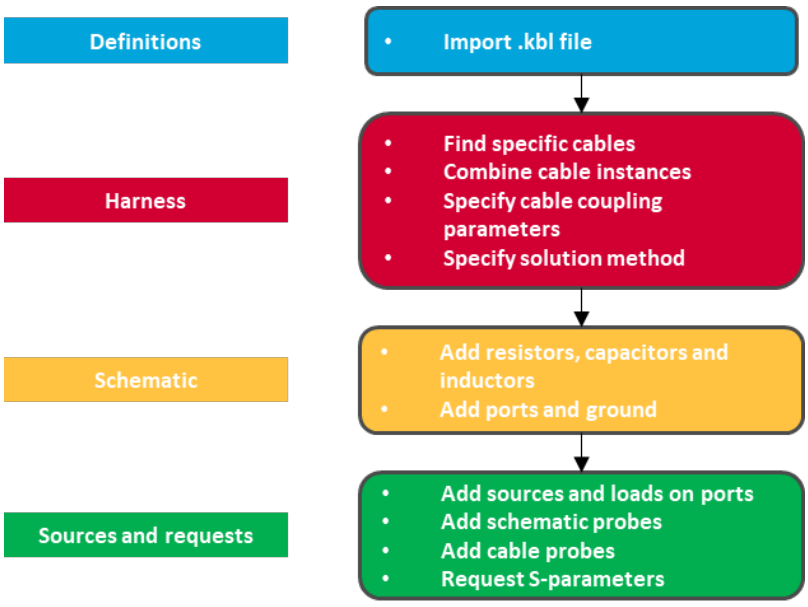
- Cartesian_point
- Node
- Segment
- Routing
- Cable paths
- Cross sections
- Cable harnesses
- Contact_points
- Connector_occurrance
- Special_terminal_occurrence
- Component_box_occurrence
- Reference_Components

Note:

- Cross sections are read as single conductors and media properties are ignored.
- CADFEKO supports only a subset of the v2.3 KBL format.
- No manufacturing information, material information or proprietary information is parsed from the `.kbl` file.

Workflow for Analysing Cables by Importing From a .KBL File

The general process is explained for setting up a complete cable analysis by importing a complex cable harness from a `.kbl` file.



1. Import the `.kbl` file.

2. Find specific cable instances and combine the multiple single conductors into a single cable.
3. Specify the solution method for the outer cable problem.
4. Specify the cable coupling parameters.
5. Open the cable schematic view of each harness.
   a. Add the circuit elements (for example, resistors, capacitors, inductors, SPICE circuits, ports, ground) and connect the connector pins to the circuit elements.
6. Define the sources, loads and requests.
   a. Add sources and loads to the cable ports.
   b. Add schematic probes, cable probes or S-parameter requests to request results.

Importing a Harness Description List (.KBL) File

Add a complex cable harness (defined in a `.kbl`) to the cable assembly for analysis.

1. On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list select the **KBL File (*.kbl)** icon.

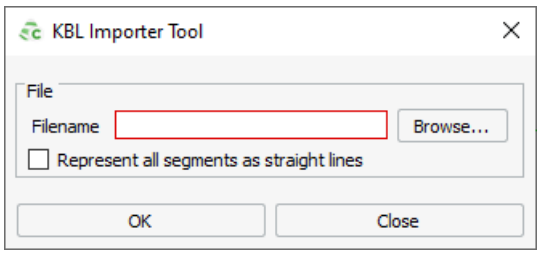


*Figure 178: The **KBL Importer Tool** dialog.*

2. In the **Filename** field, browse for the file you want to import.
3. [Optional] Select the **Represent all segments as straight lines** check box if it is required to represent curved segments as straight segments.
4. Click the **OK** to import the `.kbl` file and to close the dialog.

2.21.4 Cable Types

A comprehensive range of cable types is supported for cable analysis.

The following cable types are supported:

- Single conductor
- Coaxial cable
  - Add a predefined coaxial cable from industry
  - Define using cable characteristics
  - Define using cable dimensions
- Ribbon cable

- Twisted pair
- Cable bundle
- Non-conducting element

Defining a Single Conductor Cable

Define a single conductor consisting of a core with an optional outer coating.

1. On the **Cables** tab, in the **Definitions** group, click the **Single Conductor** icon.

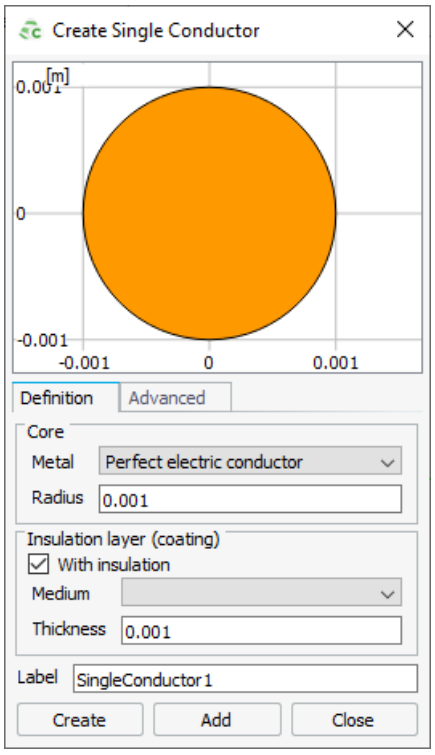


*Figure 179: The **Create Single Conductor** dialog.*

2. Under **Core**, from the **Metal** drop-down list, select one of the following:
   - To create a PEC core, select **Perfect electric conductor**.
   - To create a core consisting of a predefined metal, select a metal.
   - To create a core consisting of a metal, which is not yet defined in the model, click the icon to define a metal or add a metal from the media library.
3. Under **Core**, in the **Radius** field, enter the cable radius.
4. Under **Insulation layer (coating)**, specify the following:
   - To remove the coating, clear the **With insulation** check box.
   - To add a coating, select the **With insulation** check box.
     - To add a coating consisting of a predefined dielectric, select a dielectric.

- To add an insulation layer consisting of a dielectric, which is not yet defined in the model, click the icon to define a dielectric or add a dielectric from the media library.

5. Under **Insulation layer (coating)**, in the **Thickness** field, enter the layer (coating) thickness.
6. In the **Label** field, add a unique label for the single conductor.
7. Click **Create** to create the single conductor and to close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

Adding a Predefined Coaxial Cable from Industry

Feko has an internal coaxial cable database that contains more than 20 popular coaxial cable types from industry.

1. On the **Cables** tab, in the **Definitions** group, click the **Coaxial** icon.

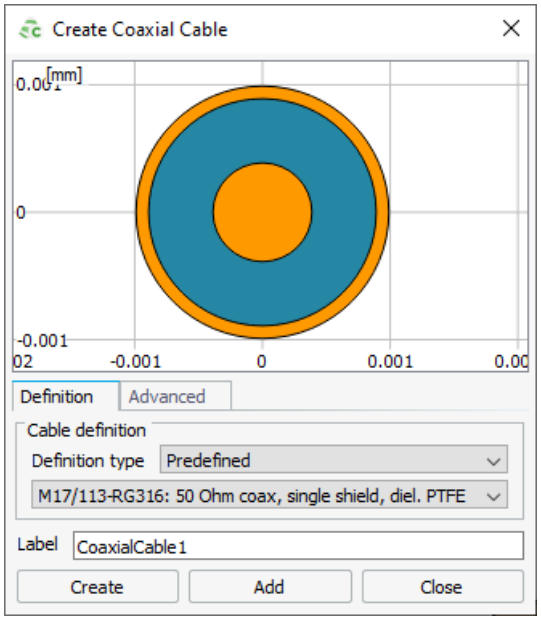


*Figure 180: The **Create Coaxial Cable** dialog.*

2. From the drop-down list, select a predefined coaxial cable from industry.
3. In the **Label** field, add a unique label for the coaxial cable.
4. Click **Create** to create the coaxial cable and close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

Defining a Coaxial Cable Using Cable Characteristics

Define a coaxial cable by its characteristic impedance and propagation constant.

1. On the **Cables** tab, in the **Definitions** group, click the **Coaxial** icon.
2. Under **Cable definition**, from the **Definition type** drop-down list, select **Specify cable characteristics**.

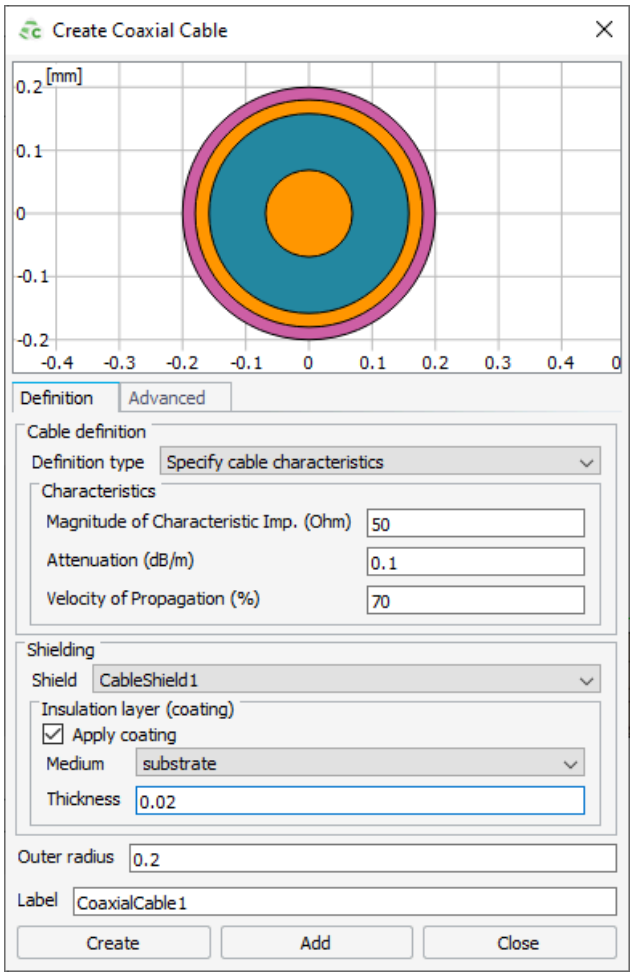


*Figure 181: The **Create Coaxial Cable** dialog (Specify coaxial cable characteristics).*

3. Under **Characteristics**, specify the following:
   - In the **Magnitude of characteristic imp. (Ohm)** field, enter the characteristic impedance (magnitude) for the coaxial cable.
   - In the **Attenuation (dB/m)** field, enter the attenuation of the coaxial cable in dB/m.
   - In the **Velocity of propagation (%)** field, enter a percentage for the velocity of propagation through the coaxial cable.
4. Under **Cable definition**, in the **Outer radius** field, enter the outer radius of the coaxial cable.

> **Note:** If a braided shield is applied to the coaxial cable the outer radius should be inside the stretching limits for a braided shield.

5. Under **Shielding**, from the **Shield** drop-down list, select one of the following:
   - To add an outer cable shield consisting of a predefined shield, select a cable shield.
   - To add an outer cable shield consisting of a shield, which is not yet defined in the model, click the icon to define a new cable shield.
6. Under **Insulation layer (coating)**, specify the following:
   - To add a coating, select the **Apply coating** check box.
     1. From the **Medium** drop-down list, specify the coating medium.
     2. In the **Thickness** field, specify the coating thickness.
   - To remove the coating, clear the **Apply coating** check box.
7. In the **Label** field, add a unique label for the coaxial cable.
8. Click **Create** to create the coaxial cable and to close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

Defining a Coaxial Cable Using Cable Dimensions

Define a coaxial cable consisting of a core with outer insulating layers and a shield.

1. On the **Cables** tab, in the **Definitions** group, click the **Coaxial** icon.
2. Under **Cable definition**, from the **Definition type** drop-down list, select **Specify coaxial cable dimensions**.

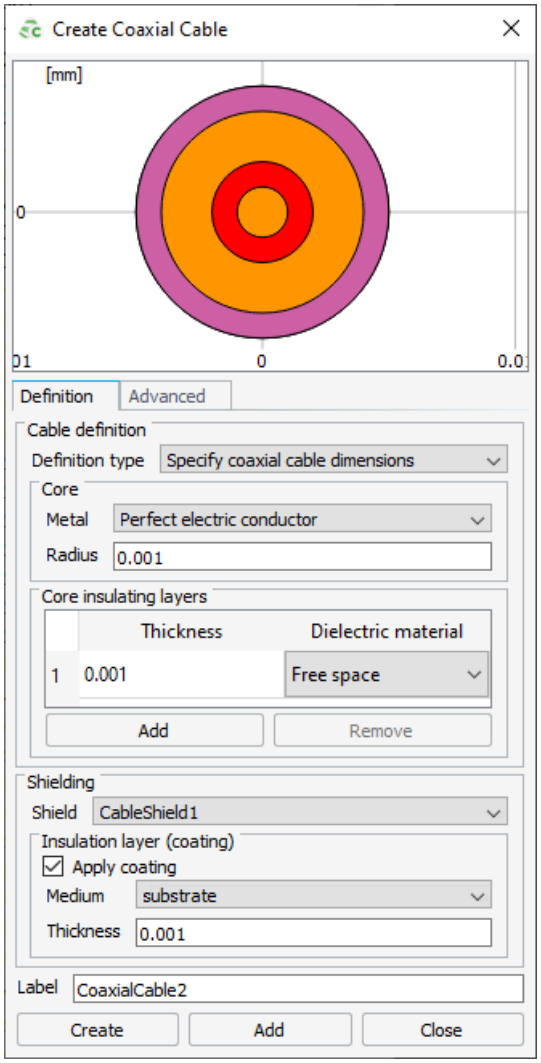


*Figure 182: The **Create Coaxial Cable** dialog (Specify coaxial cable dimensions).*

3. Under **Core**, from the **Metal** drop-down list, select one of the following:
   - To create a PEC core, select **Perfect electric conductor**.
   - To create a core consisting of a predefined metal, select a metal.
   - To create a core consisting of a metal, which is not yet defined in the model, click the icon to define a metal or add a metal from the media library.
4. Under **Core**, in the **Radius** field, enter the cable radius.
5. Under **Core insulating layers**, for each layer:
   - To add a layer consisting of free space, select **Free space**.
   - To add a coating consisting of a predefined dielectric, select a dielectric.
   - To add an insulation layer consisting of a dielectric, which is not yet defined in the model, click the icon to define a dielectric or add a dielectric from the media library.
6. Under **Core insulating layers**, in the **Thickness** field, enter the layer thickness for each layer.

> **Note:** If a braided shield is applied to the coaxial cable the outer radius (core + core insulation layer (s) thickness + total shield thickness) should be inside the stretching limits defined for the braided shield.

7. Under **Shielding**, from the **Shield** drop-down list, select one of the following:
   - To add an outer cable shield consisting of a predefined shield, select a cable shield.
   - To add an outer cable shield consisting of a shield, which is not yet defined in the model, click the icon to define a new cable shield.
8. Under **Insulation layer (coating)**, specify the following:
   - To add a coating, select the **Apply coating** check box.
     1. From the **Medium** drop-down list, specify the coating medium.
     2. In the **Thickness** field, specify the coating thickness.
   - To remove the coating, clear the **Apply coating** check box.
9. In the **Label** field, add a unique label for the coaxial cable.
10. Click **Create** to create the coaxial cable and to close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

Defining a Ribbon Cable

Define a ribbon cable consisting of multiple cores with an optional coating for each core.

1. On the **Cables** tab, in the **Definitions** group, click the **Ribbon** icon.

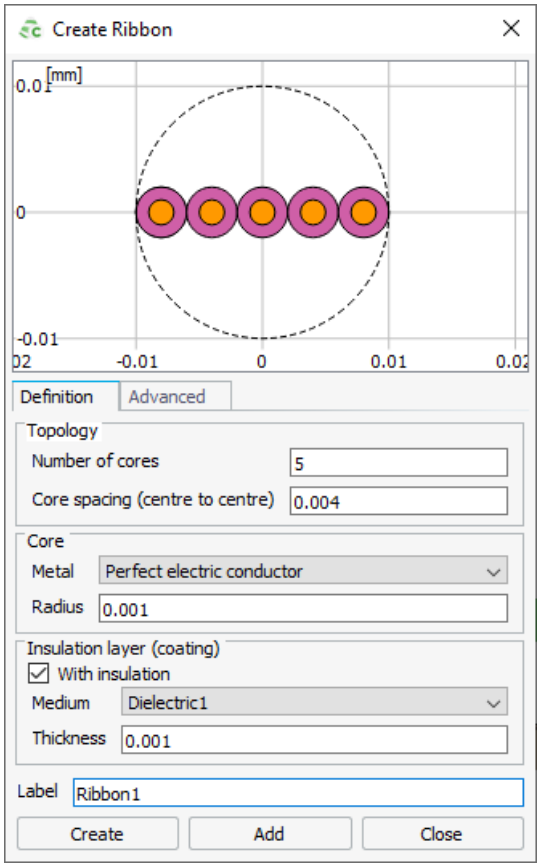


*Figure 183: The **Create Ribbon** dialog.*

2. Under **Topology**, in the **Number of cores** field, enter the number of cores in the ribbon cable.
3. Under **Topology**, in the **Core spacing (centre to centre)** field, enter the distance between the adjacent cables.
4. Under **Core**, from the **Metal** drop-down list, select one of the following:
   - To create a PEC core, select **Perfect electric conductor**.
   - To create a core consisting of a predefined metal, select a metal.
   - To create a core consisting of a metal, which is not yet defined in the model, click the icon to define a metal or add a metal from the media library.
5. Under **Core**, in the **Radius** field, enter the radius of the core.
6. Under **Insulation layer (coating)**, specify the following:
   - To remove the coating, clear the **With insulation** check box.
   - To add a coating, select the **With insulation** check box.
     - To add a coating consisting of a predefined dielectric, select a dielectric.
     - To add an insulation layer consisting of a dielectric, which is not yet defined in the model, click the icon to define a dielectric or add a dielectric from the media library.
7. In the **Label** field, add a unique label for the ribbon cable.
8. Click **Create** to create the ribbon cable and to close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

Defining a Twisted Pair

Define a twisted pair consisting of two cores that are twisted together for the purposes of cancelling electromagnetic interference. Each core can have an optional coating.

1. On the **Cables** tab, in the **Definitions** group, click the **Twisted Pair** icon.

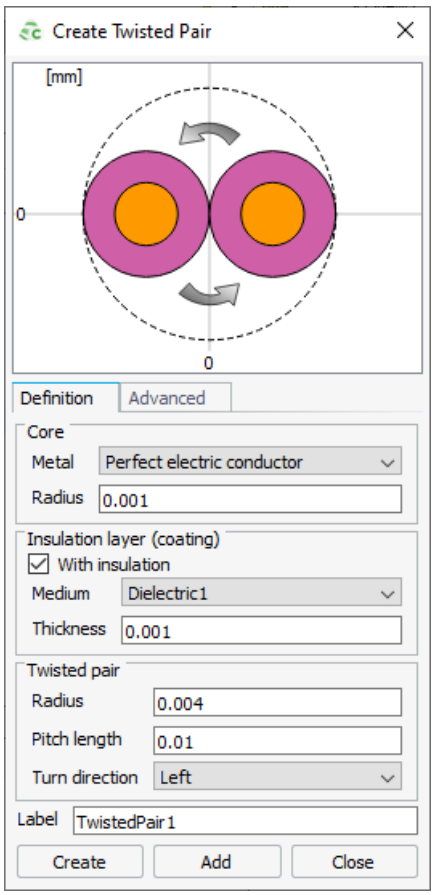


*Figure 184: The **Create Twisted Pair** dialog.*

2. Under **Core**, from the **Metal** drop-down list, select one of the following:
   - To create a PEC core, select **Perfect electric conductor**.
   - To create a core consisting of a predefined metal, select a metal.
   - To create a core consisting of a metal, which is not yet defined in the model, click the icon to define a metal or add a metal from the media library.
3. Under **Insulation layer (coating)**, specify the following:
   - To remove the coating, clear the **With insulation** check box.
   - To add a coating, select the **With insulation** check box.

- To add a coating consisting of a predefined dielectric, select a dielectric.
- To add an insulation layer consisting of a dielectric, which is not yet defined in the model, click the icon to define a dielectric or add a dielectric from the media library.

4. Under **Twisted pair**, in the **Radius** field, enter the outer radius of the twisted pair cable.
5. Under **Twisted pair**, in the **Pitch length** field, enter the axial length required to complete one revolution of the strand around the diameter of the conductor.
6. Under **Twisted pair**, in the **Turn direction** drop-down list, select one of the following:
   - To define a twisted pair with its strands turning right, leading away from you, select **Right**.
   - To define a twisted pair with its strands turning left, leading away from you, select **Left**.
7. In the **Label** field, add a unique label for the twisted pair.
8. Click **Create** to create the twisted pair and to close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

Defining a Cable Bundle

Define a cable bundle that may consist of multiple defined cables (for example, single conductors, coaxial cables, ribbon cables, twisted pairs, other cable bundles and non-conducting elements) and that are embedded in a medium with an optional shield.

Note: The following shield types are supported for cable bundles:

1. Insulated, embedded in background medium (sheath/jacket)
2. Not shielded, embedded in a dielectric
3. Not shielded, embedded in background medium
4. Shielded, dielectric filled

1. On the **Cables** tab, in the **Definitions** group, click the **Cable Bundle** icon.

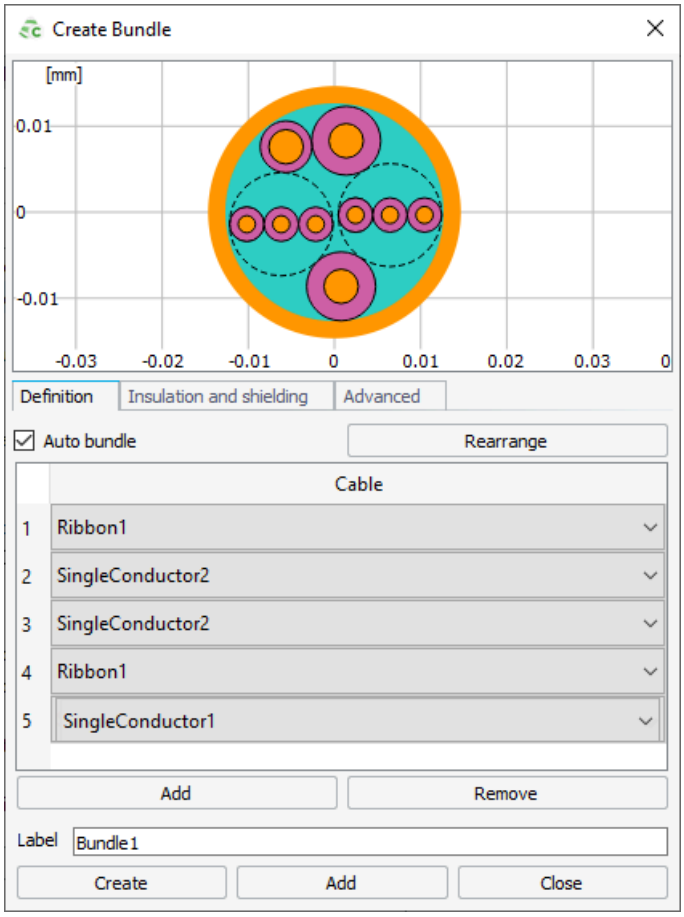


*Figure 185: The **Create Bundle** dialog.*

2. On the **Bundle** tab, bundle the cables using one of the following methods:
   - To create a cable bundle where the exact orientation of the cable in the bundle is unknown or not relevant, select the **Auto bundle** check box.
     - Click the **Rearrange** button to place the cables in a new random location inside the bundle.
   - To specify the location and orientation of the cables inside the cable bundle, clear the **Auto bundle** check box.
     - Specify the **Offset X**, **Offset Y** and **Rotation** of each cable contained in the bundle.
3. From the **Cable** drop-down list, specify the cables contained in the bundle using one of the following methods:
   - To specify a predefined cable, select the cable you want to add.
   - To specify a cable, not yet defined in the model, click the icon to define a cable type.

On the **Insulation and Shielding** tab, for shield types 1, 2 and 4, specify the **Insulation medium**.

4. On the **Insulation and shielding** tab, from the **Insulation medium** drop-down list, select one of the following:
   - To specify the insulation medium consisting of a predefined dielectric, select the dielectric.

- To specify the insulation medium consisting of dielectric, which is not yet defined in the model, click the  icon to define a dielectric or add a dielectric from the media library.

For shield types 1, 2 and 4, specify the **Outer radius** for the cable bundle.

5. On the **Insulation and shielding** tab, to specify the **Outer radius**, select one of the following:
   - To allow CADFEKO to calculate the outer radius of the cable bundle, select the **Compute automatically** check box.
   - To manually specify the outer radius of the cable bundle in the **Outer radius** field, clear the **Compute automatically** check box.

For shield type 1, specify the **Shield** for the cable bundle.

> **Note:** If a braided shield is applied to the cable bundle the outer radius should be inside the stretching limits defined for the braided shield.

6. On the **Insulation and shielding** tab, under **Shielding**, from the **Shield** drop-down list, select one of the following:
   - To add an outer cable shield consisting of a predefined shield, select a cable shield.
   - To add an outer cable shield consisting of a shield, which is not yet defined in the model, click the  icon to define a new cable shield.
7. Under **Insulation layer (coating)**, specify the following:
   - To add a coating, select the **Apply coating** check box.
     1. From the **Medium** drop-down list, specify the coating medium.
     2. In the **Thickness** field, specify the coating thickness.
   - To remove the coating, clear the **Apply coating** check box.
8. On the **Advanced** tab, under **Twist**, from the **Turn direction** drop-down list select one of the following:
   - To define a bundle with no twist, select **No twist**.
   - To define a bundle turning right, leading away from you, select **Right handed**.
   - To define a bundle turning left, leading away from you, select **Left handed**.
9. On the **Advanced** tab, under **Twist**, in the **Pitch length** field, enter the axial length required to complete one revolution of a cable in the bundle around the diameter of the bundle.
10. In the **Label** field, add a unique label for the cable bundle.
11. Click **Create** to create the cable bundle and to close the dialog.

Related concepts

Insulation and Shielding For Cable Bundles

Rearrange Cable Bundle Using CADFEKO_BATCH.

Related tasks

Accessing the Cables Tab on the Ribbon

Insulation and Shielding For Cable Bundles

When defining a cable bundle, you can specify the outer insulation and shielding for the cables contained in the bundle.

The following shield types are supported for the cable bundle:

1. Insulated, embedded in background medium (sheath/jacket)
2. Not shielded, embedded in a dielectric
3. Not shielded, embedded in background medium
4. Shielded, dielectric filled

Insulated, Embedded in Background Medium (Sheath / Jacket)

Add an outer sheath / jacket to cables contained in a bundle. The cables are embedded in the background medium, which is by default free space.

> **Note:** A sheath or jacket is a close-fitting cover that protects the internal conductors of the cable against moisture, chemicals, and mechanical damage and insulates the cable electrically.

The sheath/jacket is specified using **Insulation medium** and **Sheath thickness**.

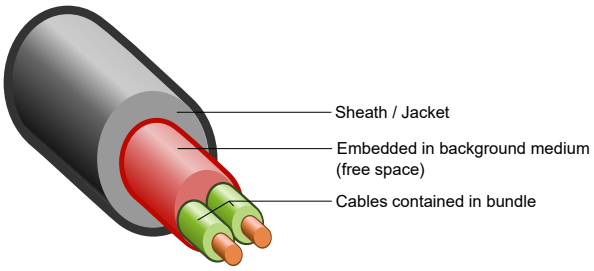


Figure 186: A 3D representation of a cable bundle (two single conductors, each with a coating) embedded in the background medium (in red), covered by a sheath / jacket (in black).

Not Shielded, Embedded in a Dielectric

Embed a cable bundle inside a dielectric. The bundle is unshielded (no shield). The dielectric is specified using **Insulation medium**.

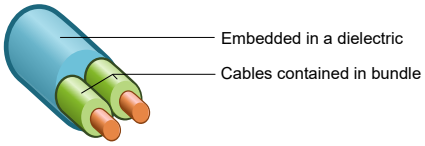


Figure 187: A 3D representation of a cable bundle (two single conductors, each with a coating) embedded in a dielectric (in blue). Note that the outer cable bundle does not contain a shield.

Not Shielded, Embedded in Background Medium

Embed a cable bundle inside the background medium. By default the background medium is free space.

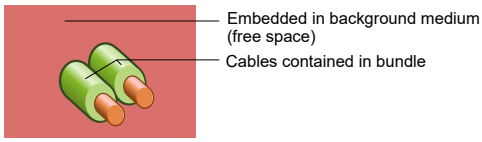


Figure 188: A 3D representation of a cable bundle routed in the background medium (by default, free space).

Shielded, Dielectric Filled

Add an outer shield to cables contained in a bundle. The inner cable bundle is embedded in a dielectric. You can also choose to add an insulation layer / coating over the shield.

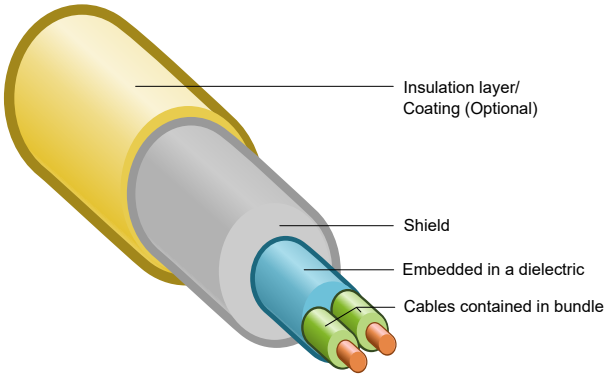


Figure 189: A 3D representation of a cable bundle (two single conductors, each with a coating) embedded in a dielectric (in blue), covered by a shield (in grey) and coated with an insulation medium (in yellow).

Defining a Non-Conducting Element

Define a non-conducting element consisting of a fibre core.

1. On the **Cables** tab, in the **Definitions** group, click the **Non-Conducting Element** icon.

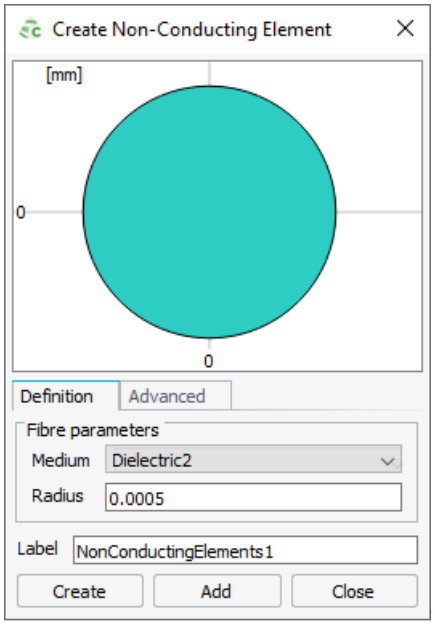


*Figure 190: The **Create Non-Conducting Element** dialog.*

2. Under **Fibre parameters**, from the **Medium** drop-down list, select one of the following:
   - To specify the medium consisting of a predefined dielectric, select the dielectric.
   - To specify the medium consisting of a dielectric, which is not yet defined in the model, click the icon to define a dielectric or add a dielectric from the media library.
3. Under **Fibre parameters**, in the **Radius** field, enter a value for the cable radius.
4. In the **Label** field, add a unique label for the non-conducting element.
5. Click **Create** to create the coaxial cable and to close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

Advanced Settings for Cable Types

Advanced settings are used to specify the accuracy of the cable per-unit-length parameters.

On the **Cables** tab, in the **Definitions** group, click any of the cable types.

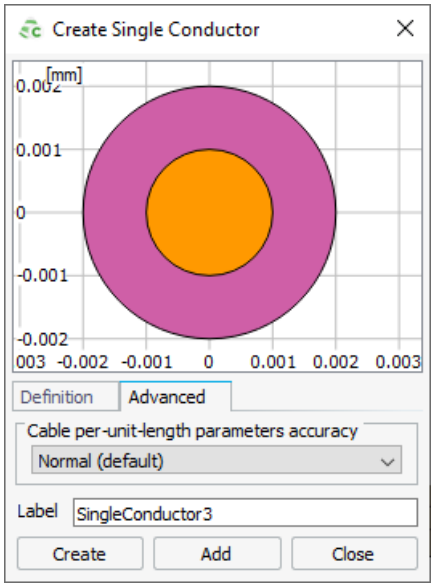


*Figure 191: An example of a cable type dialog - the **Create Single Conductor** dialog, **Advanced** tab.*

For each cable type definition, the **Cable per-unit-length parameters accuracy** can be increased from **Normal (default)** to **High** or **Very high** to allow for increasingly finer meshing of the cable cross sections.

2.21.5 Cable Shields

A cable shield is a conductive layer that encloses a cable to reduce electromagnetic interference (EMI) and crosstalk to other cables.

CADFEKO enables you to specify several types of shielding:

- solid shields with a specified material and thickness
  - Schelkunoff
- braided shields
  - Kley
  - Vance
  - Tyni
  - Demoulin
- defining the frequency-dependent shield properties
  - Transfer impedance (Zt) and surface impedance (Zs)
  - Transfer admittance (Yt)
  - Transfer capacitance

Cable Shield Layer Combinations

When creating a cable shield, you need to specify the impedance and admittance for each layer. The following combinations are supported when defining the impedance and admittance for a shield layer.

Table 14: Supported shield layer combinations when specifying the impedance and admittance for a shield layer.

Impedance Definition (Zt + Zs)	Admittance Definition (Yt)
Solid (Schelkunoff)	Not applicable
Braided (Kley)	Same as impedance definition Transfer capacitance Define properties
Braided (Tyni)	Same as impedance definition Transfer capacitance Define properties
Braided (Vance)	Same as impedance definition Transfer capacitance Define properties
Braided (Demoulin)	Same as impedance definition Transfer capacitance Define properties
Define properties	Same as impedance definition Define properties Transfer capacitance

For example, when selecting a Braided (Kley) impedance definition it can be combined with a braided (Kley), transfer capacitance or the frequency-dependent (define properties) admittance definition.

Related tasks

Creating a Solid Cable Shield Layer (Schelkunoff)
Creating a Braided Cable Shield Layer (Kley)
Creating a Braided Cable Shield Layer (Vance, Tyni or Demoulin)
Creating a Cable Shield Layer (Shield Properties)
Creating a Cable Shield Layer (Transfer Capacitance)

Creating a Solid Cable Shield Layer (Schelkunoff)

Create a single-layered solid shield with a specified material and thickness.

1. On the **Cables** tab, in the **Definitions** group, click the **Cable Shield** icon.
2. Under **Shield layer(s)**, click **Single** to create a single-layered shield.
3. On the **Inner layer** tab, on the **Impedance definition** tab, from the **Definition method** drop-down list, select **Solid (Schelkunoff)**.

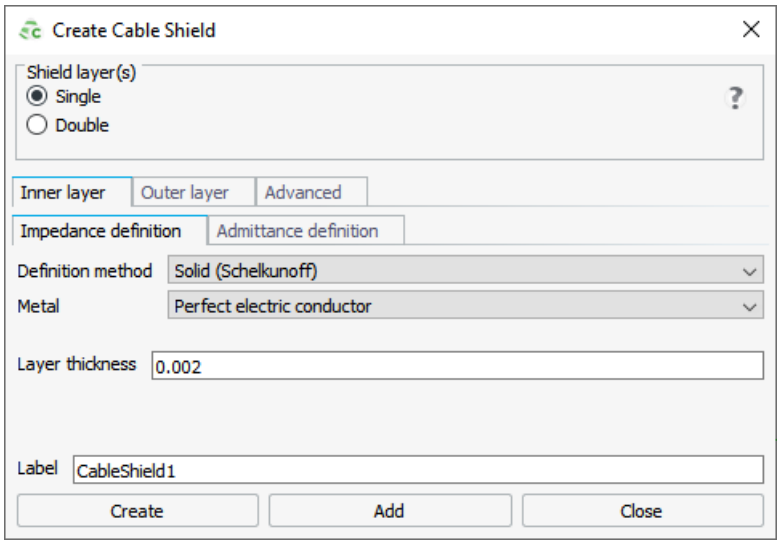


*Figure 192: The **Create Cable Shield** dialog.*

4. From the **Metal** drop-down list, select one of the following:
   - To create a PEC shield, select **Perfect electric conductor**.
   - To create a shield consisting of a predefined metal, select the metal.
   - To create a shield consisting of a metal, which is not yet defined in the model, click the icon to define a metal or add a metal from the media library.
5. On the **Inner layer** tab, in the **Layer thickness** field, specify the inner layer thickness.
6. In the **Label** field, add a unique label for the cable shield.
7. Click **Create** to create the cable shield and to close the dialog.

Related tasks

Accessing the Cables Tab on the Ribbon

Weave Definitions for a Braided Cable Shield

CADFEKO supports two methods to specify the size of the apertures for a braided shield layer.

The optical coverage for a braid indicates how visible the apertures are, where 0% is completely open (no shielding) and 100% is a completely filled (approximating a solid shield). The optical coverage and weave angle are coupled by the fill factor (*F*) for a braid, which is a quantity between 0 and 1.

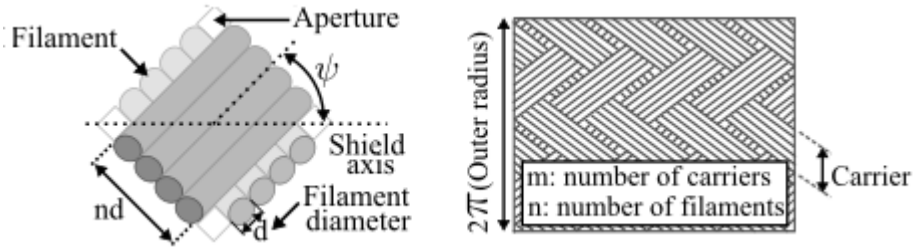


Figure 193: Illustration of the braid parameters for a braided shield layer.

The fill factor (*F*) is calculated from the optical coverage using the following equation:

$$F = 1 - \sqrt{\frac{100 - OpticalCoverage}{100}} \tag{7}$$

where the optical coverage is between 0% and 100%.

The coupled equation for the fill factor from the braid parameters is:

$$F = \frac{nmd}{2\pi D\cos(\psi)} \tag{8}$$

where D is the mean braid diameter (outer radius of the shield minus half the thickness of the braided layer).

Optical Coverage

For the optical coverage definition, a minimum optical coverage is specified. The minimum optical coverage relates to the largest aperture size or minimum shielding that CADFEKO tries to achieve when optimising the braid for maximum coverage, by varying the weave angle.

Weave Angle

A nominal weave angle and deviation is specified for the weave angle definition. The optical coverage is calculated for the range of weave angles to represent the size of the apertures (using the braid parameters defined for a braided layer).

> **Note:** As the weave angle changes, the shield diameter also changes depending on the weave angle and other braid parameters.

Related concepts

Advanced Settings for a Braided Cable Shield Layer

Related tasks

Creating a Braided Cable Shield Layer (Kley)

Creating a Braided Cable Shield Layer (Vance, Tyni or Demoulin)

Creating a Braided Cable Shield Layer (Kley)

Create a single layer, braided (Kley) cable shield. The relevant braid parameters, weave metal and braid-fixing materials (optional) are specified and the Solver determines the frequency-dependent impedance (Zs + Zt) and admittance (Yt) matrix using the Kley formulation.

The Kley formulation models the coupling mechanism accurately due to the field penetration through the shield apertures.

1. On the **Cables** tab, in the **Definitions** group, click the **Cable Shield** icon.
2. Under **Shield layer(s)**, click **Single** to create a single-layered shield.
3. On the **Inner layer** tab, on the **Impedance definition** tab, from the **Definition method** drop-down list, select **Braided (Kley)**.

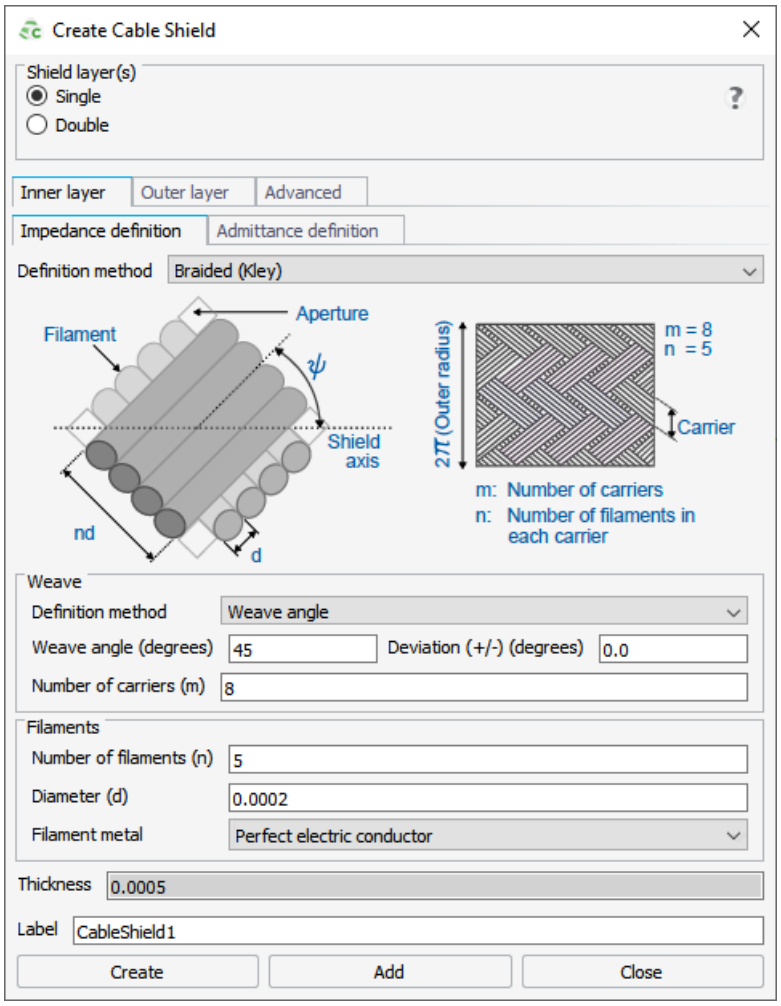


*Figure 194: The **Create Cable Shield** dialog.*

4. Under **Weave**, specify the following:
   a) From the **Definition method** , select **Weave angle** to create a braided layer using the weave angle definition:

- In the **Weave angle (degrees)** field, enter a value in degrees for the nominal weave angle.
- In the **Deviation (+/-) (degrees)** field, enter a value for the deviation of the weave angle from the nominal weave angle in degrees.

b) From the **Definition method** , select **Optical coverage** to create a braided layer using the optical coverage definition:

- In the **Minimum optical coverage (%)** field, enter a percentage for the minimum optical coverage for the braided layer.

c) In the **Number of carriers (m)** field, specify the number of carriers in the braided layer.

5. Under **Filaments**, specify the following:

a) In the **Number of filaments (n)** field, enter a value for the number of filaments in a single carrier.

b) In the **Diameter (d)** field, enter a value for the filament diameter.

c) In the **Filament metal** , select one of the following:

- To create a filament consisting of PEC, select **Perfect electric conductor**.
- To create a filament consisting of a predefined metal, select the metal.
- To create a filament consisting of a metal, which is not yet defined in the model, select the  icon to define a metal or add a metal from the media library.

Note: The **Thickness** of a Kley shield layer is 2.5 times the filament diameter (d).

6. On the **Inner layer** tab, on the **Admittance definition** tab, from the **Definition method** drop-down list, select **Same as impedance definition** to use the Kley formulation for the admittance matrix.

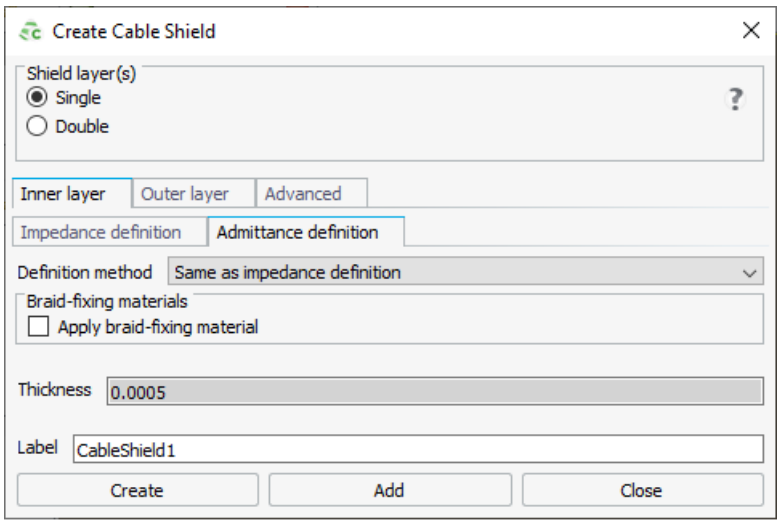


*Figure 195: The **Create Cable Shield** dialog.*

7. Under **Braid-fixing materials**, select one of the following:

- To apply an inside and outside braid-fixing material, select the **Apply braid-fixing material** check box.
  - To add an inside and outside braid-fixing material consisting of a predefined dielectric, select the dielectric.
  - To add an inside and outside braid-fixing material consisting of a dielectric, which is not yet defined in the model, click the icon to define a dielectric or add a dielectric from the media library.
- To remove the braid-fixing material, clear the **Apply braid-fixing material** check box.

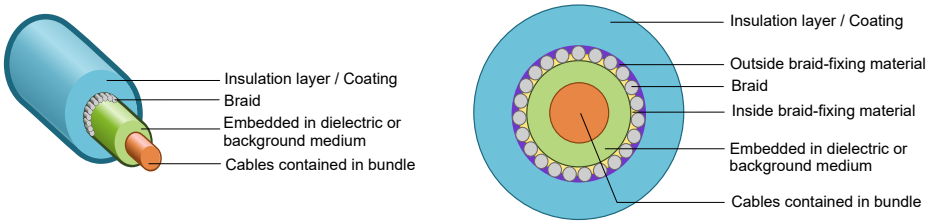


Figure 196: A 3D representation of a cable containing a braid (on the left) and a cross-section of the cable showing the inside braid-fixing material and the outside braid-fixing material (on the right).

8. In the **Label** field, add a unique label for the cable shield.
9. Click **Create** to create the cable shield and close the dialog.

Related concepts
Weave Definitions for a Braided Cable Shield
Related tasks
Accessing the Cables Tab on the Ribbon

Creating a Braided Cable Shield Layer (Vance, Tyni or Demoulin)

Create a single layer, braided (Vance, Tyni, Demoulin) cable shield. For a braided shield layer, the relevant braid parameters and weave metal are specified and the Solver determines the frequency-dependent impedance (Zs + Zt) and admittance (Yt) matrix using the Vance, Tyni or Demoulin formulation.

The Vance, Tyni and Demoulin formulation models the coupling mechanism accurately due to the field penetration through the shield apertures.

1. On the **Cables** tab, in the **Definitions** group, click the **Cable Shield** icon.
2. Under **Shield layer(s)**, click **Single** to create a single-layered shield.
3. On the **Inner layer** tab, on the **Impedance definition** tab, from the **Definition method** drop-down list, select one of the following:
   - **Braided (Vance)**
   - **Braided (Tyni)**
   - **Braided (Demoulin)**

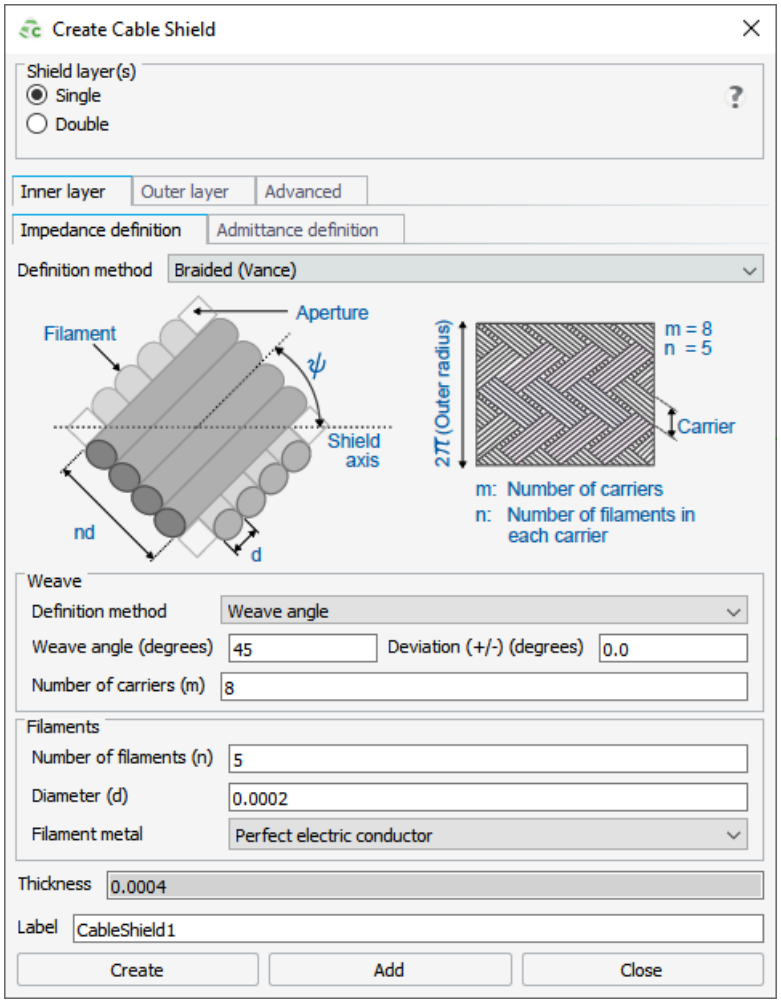


*Figure 197: The **Create Cable Shield** dialog.*

4. Under **Weave**, specify the following:
   a) From the **Definition method** drop-down list, select **Weave angle** to create a braided layer using the weave angle definition:
      - In the **Weave angle (degrees)** field, enter a value in degrees for the nominal weave angle.
      - In the **Deviation (+/-) (degrees)** field, enter a value for the deviation of the weave angle from the nominal weave angle in degrees.
   b) From the **Definition method** drop-down list, select **Optical coverage** to create a braided layer using the optical coverage definition:
      - In the **Minimum optical coverage (%)** field, enter a percentage for the minimum optical coverage for the braided layer.
   c) In the **Number of carriers (m)** field, specify the number of carriers in the braided layer.
5. Under **Filaments**, specify the following:
   a) In the **Number of filaments (n)** field, enter a value for the number of filaments in a single carrier.

b) In the **Diameter (d)** field, enter a value for the filament diameter.

c) In the **Filament metal** drop-down list, select one of the following:

- To create a filament consisting of PEC, select **Perfect electric conductor**.
- To create a filament consisting of a predefined metal, select the metal.
- To create a filament consisting of a metal, which is not yet defined in the model, select the  icon to define a metal or add a metal from the media library.

Note: The **Thickness** of a Vance, Tyni and Demoulin shield layer is two times the filament diameter (d).

6. On the **Inner layer** tab, on the **Admittance definition** tab, select **Same as impedance definition**, from the **Definition method** drop-down list to use the Vance, Tynior Demoulin formulation for the admittance matrix.

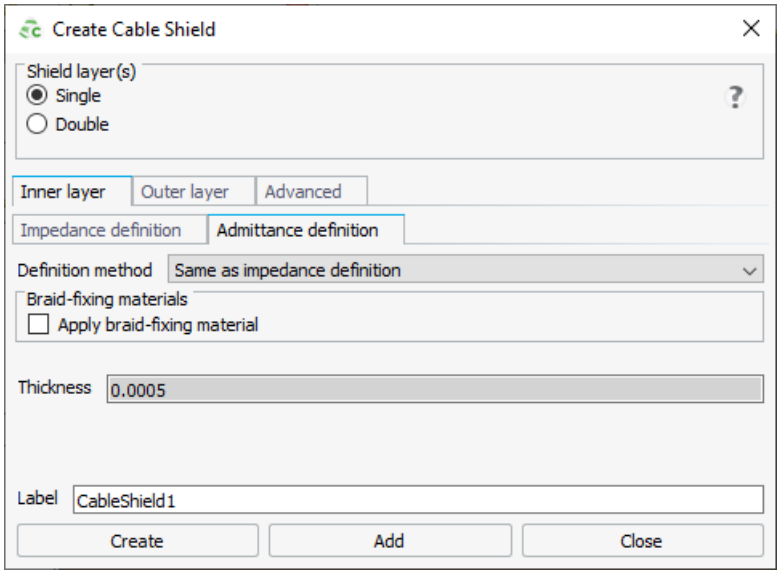


*Figure 198: The **Create Cable Shield** dialog.*

Note:
The weave and filaments values are used from the impedance definition to calculate the admittance matrix.

7. In the **Label** field, add a unique label for the cable shield.
8. Click **Create** to create the cable shield and close the dialog.

Related concepts
Weave Definitions for a Braided Cable Shield
Related tasks
Accessing the Cables Tab on the Ribbon

Advanced Settings for a Braided Cable Shield Layer

Use advanced settings to specify how the weave angle and optical coverage changes when a braided shield is applied to a cable bundle or coaxial cable.

On the **Cables** tab, in the **Definitions** group, click the **Cable Shield** icon. The advanced settings are available on the **Advanced** tab.

Optimisation Method: Maximise the Optical Coverage

The total shield radius is the outermost radius of the shield and is used as a common reference when creating double shields. The total shield radius is used to look up the weave angle and optical coverage for a specific shield size in the stretching table.

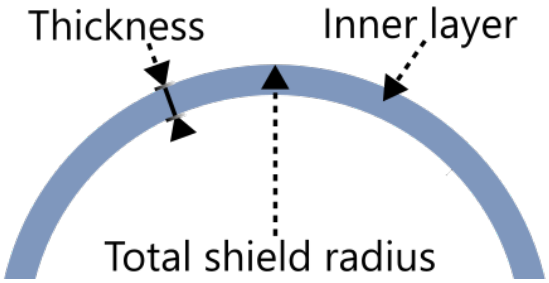


Figure 199: Illustration of a single layered shield.

The maximum and minimum stretching radius of the shield is indicated by the top and bottom entry in the **Total shield radius** column (outer radius of the shield).

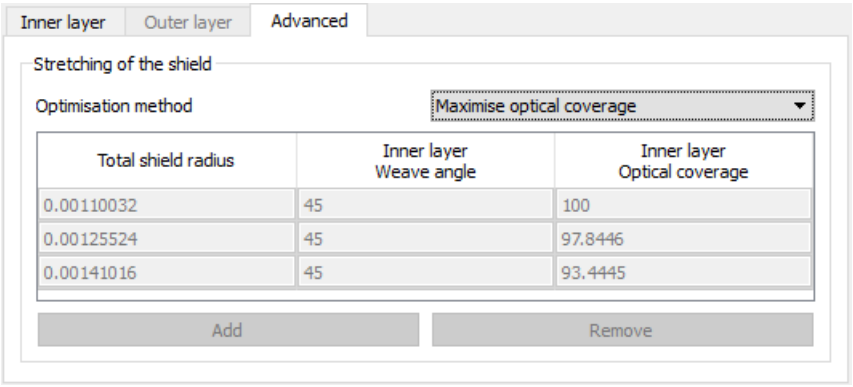


*Figure 200: The **Maximise optical coverage** method for the stretching of a braided shield layer.*

Note:

The weave angle and optical coverage range displayed in the stretching table is dependent on the following weave definition limits.

For the optical coverage definition method:

- The optimal weave angle is in the range of 20° to 70°.
- The optical coverage is limited from the minimum optical coverage defined to 100%.

For the weave angle definition method:

- The optimal weave angle is between the deviation (+/-) limits from the nominal weave angle defined for the inner layer.
- The optical coverage is limited to a range of 60% to 100%.

Optimisation Method: Specify Manually

Specify the **Total shield radius** and **Inner layer Weave angle** manually to define the stretching of the shield.

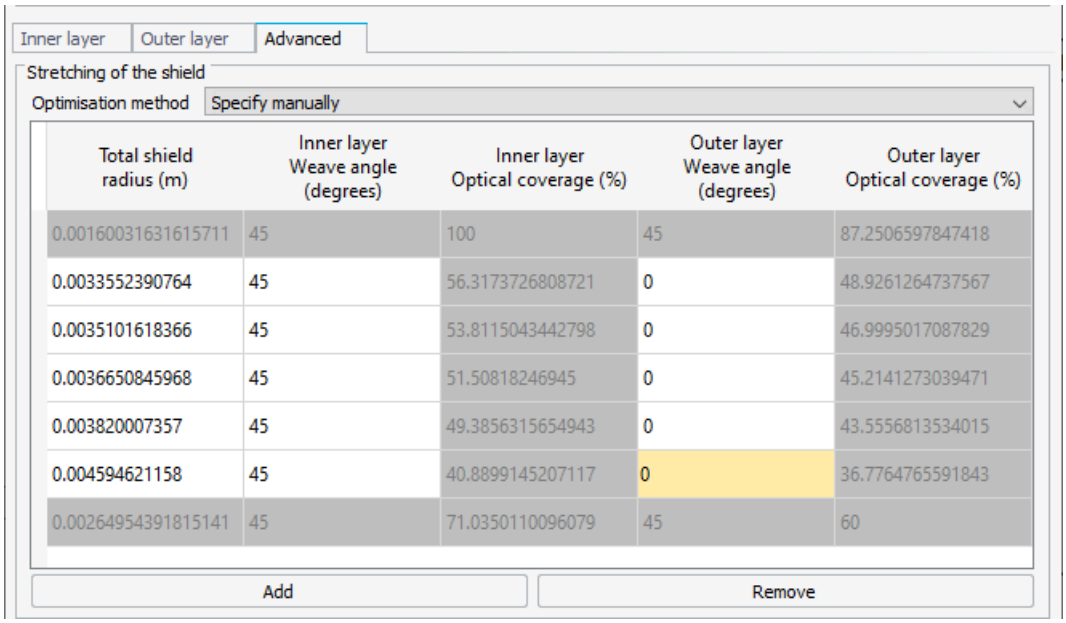


*Figure 201: The **Specify manually** method for the stretching of a braided shield layer.*

Note:

- **Total shield radius** must be between the minimum radius (first row, first cell) and maximum radius (last row, first cell).
- The **Inner layer Weave angle** values must be between the minimum angle (first row, second cell) and maximum angle (last row, second cell).
- The **Inner layer Optical coverage** is calculated automatically from the **Inner layer Weave angle** and **Total shield radius**.
- Table rows can only be added between the first row and last row.

Related concepts
Weave Definitions for a Braided Cable Shield

Single-Layered Braided Shield Models

A summary is given of the single-layered braided shield models available in CADFEKO and EDITFEKO.

Many cables used today in the industry and at home, has a flexible woven braid to shield the cable from EMI as well as to provide mechanical strength. The braid consists of conducting filaments or wires, with the weave providing a series of periodic apertures along the cable length and around its circumference.

At high frequencies, the shielding properties are dominated by the braid coverage and at lower frequencies, it is the number of wires, weave angle and the shield material conductivity.

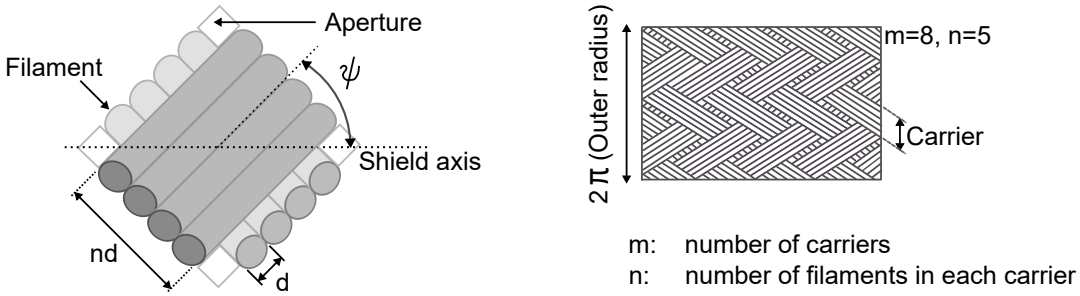


Figure 202: The weave parameters of a braided cable where d is the filament diameter and Ψ the weave angle.

Common to all the models are the diffusion impedance (Z_d) due to the current induced in the shield:

$$Z_d = \frac{R_0 \gamma d}{\sinh(\gamma d)} \tag{9}$$

where:

- R_0 is the resistance per unit length of the shield
- $\gamma = \frac{1+j}{\delta}$ is the skin depth in the wire
- d is the filament diameter

Vance

This model takes into account hole inductance caused by fields penetrating the apertures in the braided cable.

The transfer impedance expression consists of two terms:

$$Z_t \approx Z_d + j\omega L_h \quad (10)$$

where:

L_h represents penetration of magnetic fields through diamond shaped holes.

Tyni

This model takes into account braid inductance caused by magnetic coupling between the inner and outer braid layers.

The transfer impedance expression consists of three terms:

$$Z_t \approx Z_d + j\omega L_h \pm j\omega L_b \quad (11)$$

where:

L_b represents a porpoising term given by the mutual inductance between inner and outer carrier layers at the crossovers

Demoulin

This model takes into account that Z_t is not linearly increasing with frequency and that the phase variance is not between $\pm\frac{\pi}{2}$ as theory predicts for the diffraction model.

The transfer impedance expression is given by:

$$Z_t \approx Z_d + j\omega L_h + k'\sqrt{\omega}\, e^{\frac{j\pi}{4}} \pm j\omega L_b$$

Kley

This model relies on "tuning" models using measurement data. The transfer impedance expression is given by:

$$Z_t \approx Z_d + j\omega L_t + (1+j)\omega L_s$$

where:

- $L_t = M_L + M_G$ is the penetration inductance
- M_L is a hole inductance, also correcting for shield curvature
- M_G is a mutual braid inductance between the carriers in the braid
- $(1+j)\omega L_s$ is due to the magnetic field inducing two types of eddy currents

References

- *E.F. Vance, "Shielding Effectiveness of Braided-Wire Shields," IEEE Transactions on Electromagnetic Compatibility, vol. EMC-17, no. 2, pp. 71-77, May 1975.*
- *F.M. Tesche, M.V. Ianoz, and T. Karlsson, "EMC Analysis Methods and Computational Models," Wiley Interscience, Chapter 9, 1997.*

- *T. Kley, "Optimized Single-Braided Cable Shields," IEEE Transactions on Electromagnetic Compatibility, vol. EMC-35, no. 1, Feb. 1993.*
- *M. Schoeman, E.A.Attardo, J.S Castany, "Recent Advances to the Feko Integrated Cable Harness Modeling Tool", 2019 International Symposium on Electromagnetic Compatibility - EMC EUROPE, September 2019.*

Creating a Cable Shield Layer (Shield Properties)

Create a cable shield by defining the frequency-dependent surface impedance, transfer impedance and transfer admittance matrix.

1. On the **Cables** tab, in the **Definitions** group, click the **Cable Shield** icon.
2. Under **Shield layer(s)**, click **Single** to create a single-layered shield.
3. On the **Inner layer** tab, on the **Impedance definition** tab, in the **Definition method** drop-down list, select **Define properties**.

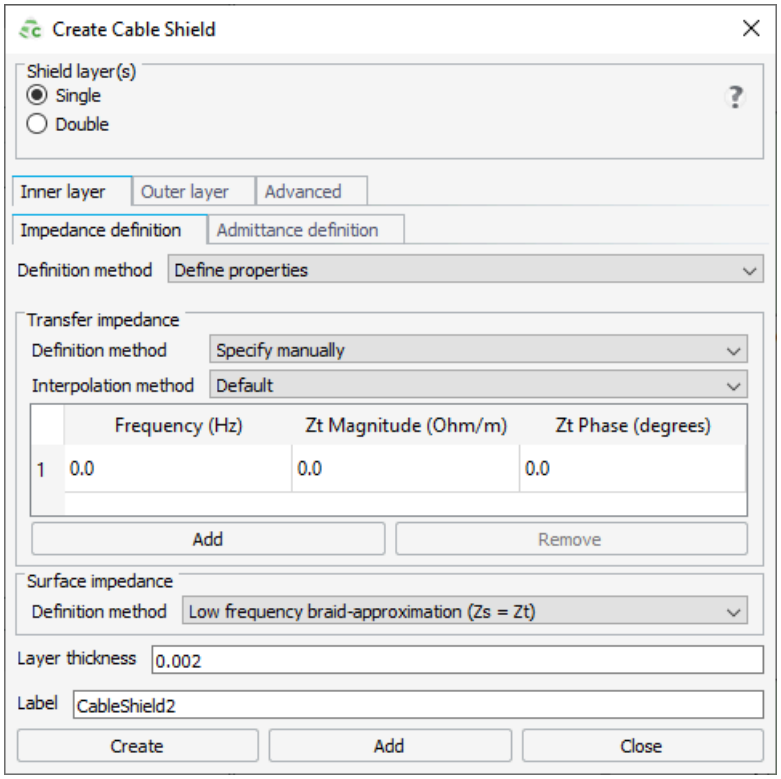


*Figure 203: The **Create Cable Shield** dialog.*

4. Under **Transfer impedance**, from the **Definition method** drop-down list, select one of the following:
   - To define the properties manually, select **Specify manually**.
     - In the **Frequency (Hz)** column, specify the frequency at which the transfer impedance and admittance are specified.

- In the **Zt Magnitude (Ohm/m)** column, specify the magnitude of the transfer impedance.
- In the **Zt phase (degrees)** column, specify the phase of the transfer impedance in degrees.

- To define the properties from an `.xml` file, select **Load from file**.
  - In the **Filename** field, browse to the file location.

5. Under **Transfer impedance**, in the **Interpolation method** drop-down list, select one of the following:
   - To use the default interpolation method between the data points, select **Default**.
   - To use a linear interpolation method between the data points, select **Linear**.
   - To use a cubic spline interpolation method between the data points, select **Cubic spline**.
   - To use a rational (Thiele) interpolation method between the data points, select **Rational**.
   - To use a constant interpolation method between the data points, select **Constant**.
6. Under **Surface impedance**, from the **Definition method** drop-down list, select one of the following:
   - To define the surface impedance (Zs) equal to the transfer impedance (Zt), select **Low frequency braid-approximation (Zs = Zt)**.
   - To define the properties manually, select **Specify manually**.
     - In the **Frequency (Hz)** column, specify the frequencies at which the surface impedance are specified.
     - In the **Zs Magnitude (Ohm/m)** column, specify the magnitude of the surface impedance for each frequency.
     - In the **Zs Phase (degrees)** column, specify the phase of the surface impedance for each frequency in degrees.
   - To define the properties from an `.xml` file, select **Load from file**.
     - In the **Filename** field, browse to the file location.
   - To define the properties from a metallic material, select **Solid (metallic material)**
     - In the **Shield metal** drop-down list, select one of the following:
       - To create a PEC shield, select **Perfect electric conductor**.
       - To create a shield consisting of a predefined metal, select the metal.
       - To create a shield consisting of a metal, which is not yet defined in the model, click the  icon to define a metal or add a metal from the media library.
7. On the **Inner layer** tab, on the **Admittance definition** tab, in the **Definition method** drop-down list, select **Define properties**.

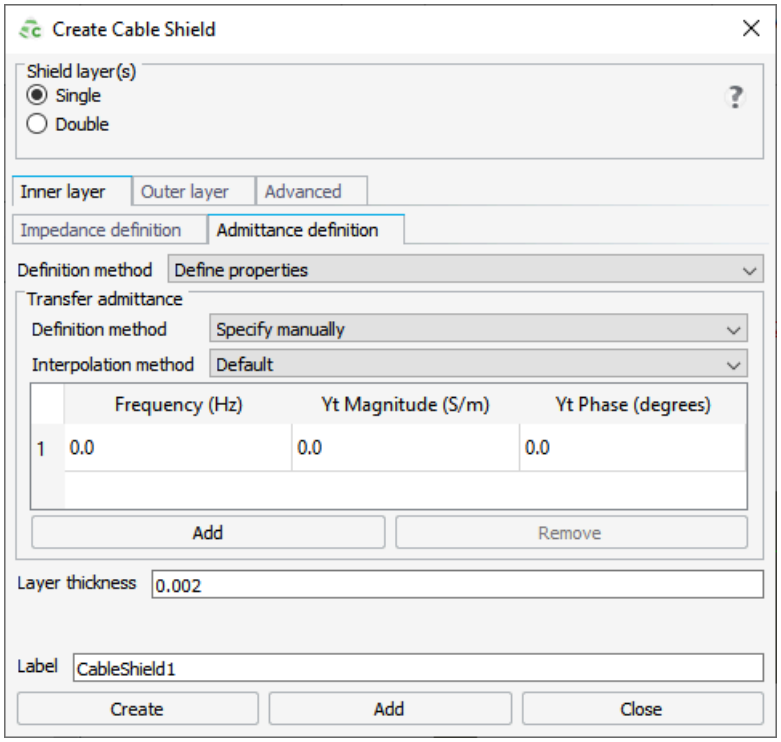


*Figure 204: The **Create Cable Shield** dialog.*

8. Under **Transfer admittance**, from the **Definition method** drop-down list, select one of the following:
   - To define the properties manually, select **Specify manually**.
     - In the **Frequency (Hz)** column, specify the frequencies at which the transfer impedance and admittance are specified.
     - In the **Yt Magnitude (S/m)** column, specify the magnitude of the transfer admittance for each frequency.
     - In the **Yt Phase (degrees)** column, specify the phase of the transfer admittance for each frequency in degrees.
   - To define the properties from an `.xml` file, select **Load from file**.
     - In the **Filename** field, browse to the file location.
9. Under **Transfer impedance**, from the **Interpolation method** drop-down list, select one of the following:
   - To use the default interpolation method between the data points, select **Default**.
   - To use a linear interpolation method between the data points, select **Linear**.
   - To use a cubic spline interpolation method between the data points, select **Cubic spline**.
   - To use a rational (Thiele) interpolation method between the data points, select **Rational**.
   - To use a constant interpolation method between the data points, select **Constant**.
10. On the **Inner layer** tab, from the **Thickness** field, enter a value for the thickness of the shield.
11. In the **Label** field, add a unique label for the cable shield.

12. Click **Create** to create the cable shield and to close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

Load Shield Properties from a .XML File

Defining `.xml` files for the different impedance and admittance combinations for the define properties shield layer.

> **Note:** The data containing phase is in degrees.

Example: Cable Shield Data (Version 1) - No Surface Impedance

An XML example containing fictitious measured data to show the file format for importing measured cable data when no surface impedance is specified.

```
<?xml version="1.0" encoding="UTF-8"?>
<cableDB creator="name" date="2011-07-30" version="1.0">
<shielding name="shield definition label">
<dataPoint freq="100e6" trans_imp_abs="5" trans_imp_phase="0" trans_adm_abs="0"
trans_adm_phase="2"/>
<dataPoint freq="300e6" trans_imp_abs="6" trans_imp_phase="2" trans_adm_abs="4"
trans_adm_phase="1"/>
<dataPoint freq="500e6" trans_imp_abs="4" trans_imp_phase="3" trans_adm_abs="3"
trans_adm_phase="2"/>
<dataPoint freq="700e6" trans_imp_abs="1" trans_imp_phase="5" trans_adm_abs="2"
trans_adm_phase="5"/>
</shielding>
</cableDB>
```

Example: Cable Shield Data (Version 2) - Same Frequency Range

An XML example containing fictitious measured data to show the file format for importing measured cable data with surface impedance measured at the same frequencies as the transfer impedance and admittance.

```
<?xml version="1.0" encoding="UTF-8"?>
<cableDB creator="name" date="2018-05-30" version="2.0">
<shielding name="shield definition label">
<dataPoint freq="1" trans_imp_abs="1" trans_imp_phase="-1"
surface_imp_abs="1" surface_imp_phase="-1" trans_adm_abs="0" trans_adm_phase="0"/>
<dataPoint freq="2" trans_imp_abs="1" trans_imp_phase="-1"
surface_imp_abs="1" surface_imp_phase="-1" trans_adm_abs="0" trans_adm_phase="0"/>
<dataPoint freq="3" trans_imp_abs="1" trans_imp_phase="-1"
surface_imp_abs="1" surface_imp_phase="-1" trans_adm_abs="0" trans_adm_phase="0"/>
<dataPoint freq="4" trans_imp_abs="1" trans_imp_phase="-1"
surface_imp_abs="1" surface_imp_phase="-1" trans_adm_abs="0" trans_adm_phase="0"/>
</shielding>
</cableDB>
```

Example: Cable Shield Data (Version 2) - Different Frequency Ranges

An XML example with different frequencies for surface impedance. The transfer impedance and admittance can also be specified separately using a divider line if required.

```
<?xml version="1.0" encoding="UTF-8"?>
<cableDB creator="name" date="2018-05-30" version="2.0">
<shielding name="shield definition label">
<dataPoint freq="1" trans_imp_abs="1" trans_imp_phase="-1"
  trans_adm_abs="0" trans_adm_phase="0"/>
<dataPoint freq="2" trans_imp_abs="1" trans_imp_phase="-1"
  trans_adm_abs="0" trans_adm_phase="0"/>
<!-- optional divider -->
<dataPoint freq="3" surface_imp_abs="1" surface_imp_phase="-1"/>
<dataPoint freq="4" surface_imp_abs="1" surface_imp_phase="-1"/>
</shielding>
</cableDB>
```

Creating a Cable Shield Layer (Transfer Capacitance)

The transfer capacitance shield layer is used with a braided and frequency-dependent impedance definition to represent the transfer admittance matrix of the shield layer.

Specify the transfer admittance (Yt) for a shield layer in Farad per meter.

1. On the **Cables** tab, in the **Definitions** group, click the **Cable Shield** icon.
2. Under **Shield layer(s)**, click **Single** to create a single-layered shield.
3. On the **Inner layer** tab, on the **Admittance definition** tab, from the **Definition method** drop-down list, select **Transfer capacitance**.

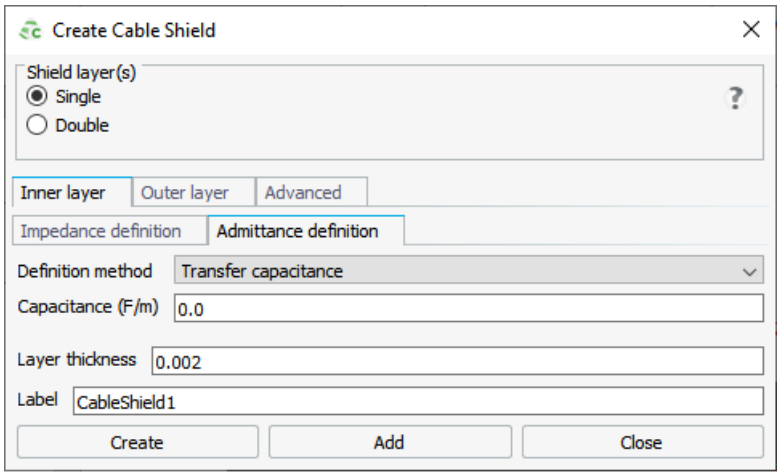


*Figure 205: The **Create Cable Shield** dialog.*

4. In the **Capacitance (F/m)** field, enter a value for the transfer capacitance in Farad per meter.

Related concepts

Cable Shield Layer Combinations

Creating a Double Layered Cable Shield

A double cable shield consists of two shield layers. CADFEKO supports any combination of the shield types, solid, braided and frequency-dependent for the inner and outer layer of the shield.

When creating a double cable shield, you need to specify the shield layer impedance and admittance for the inner layer and outer layer of the shield. A gap size also needs to be specified between the inner and outer layer of the double shield.

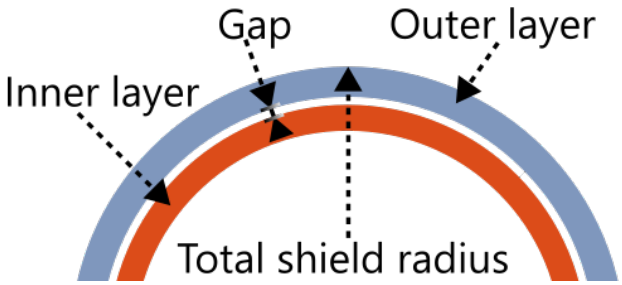


Figure 206: Illustration of double cable shield.

Note:

The total shield radius is the outer radius of the shield. The radius is used as a common reference point between the inner and outer layer when stretching the shield.

The stretching range is applicable when a braided (Kley, Vance, Tyni and Demoulin) layer definition is used on the inner layer or outer layer.

1. On the **Cables** tab, in the **Definitions** group, click the **Cable Shield** icon.
2. Under **Shield layer(s)**, click **Double** to create a double layered shield.

*Figure 207: The **Create Cable Shield** dialog - Shield layer(s)*

3. Under **Shield layer(s)**, in the **Gap between layers** field, enter a value for the gap between layers. The gap should be greater than 0.
4. On the **Inner layer** tab, create a solid, braided or frequency dependent shield layer.
5. On the **Outer layer** tab, create a solid, braided or frequency dependent shield layer.
6. On the **Advanced** tab, under **Stretching of the shield**, for the **Optimisation method** drop-down list, select **Maximise optical coverage**, to automatically calculate the optimal weave angle and optical coverage for maximum shielding for the braided inner or braided outer layer.

   The minimum and maximum radius of the double shield is indicated by the top and bottom entry in the **Total shield radius** column.

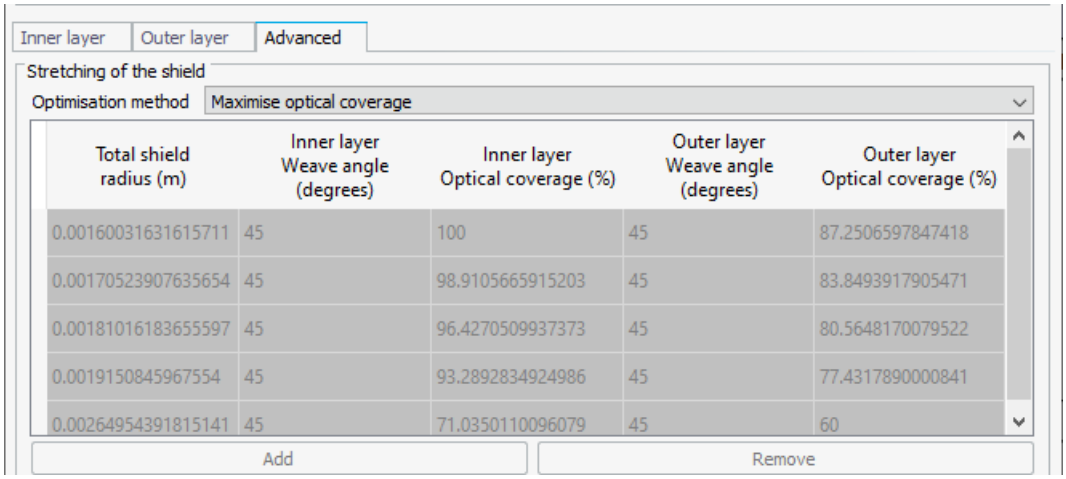


*Figure 208: The **Create Cable Shield** dialog, setting the optimisation method for the stretching of a braided shield layer.*

> **Note:** For a double braided shield, the stretching of the shield is limited to the stretching capability of both braided layers. The stretching range (**Total shield radius** range) could be smaller for a double braided shield than for a single braided shield, depending on the braid parameters selected for each shield.

7. On the **Advanced** tab, under **Stretching of the shield**, from the **Optimisation method** drop-down list, select **Specify manually** to manually define values in the stretching table:
   - In the **Total shield radius** field, edit an existing radius.
   - In the **Inner layer** or **Outer layer Weave angle** column, enter a value for the weave angle in degrees.

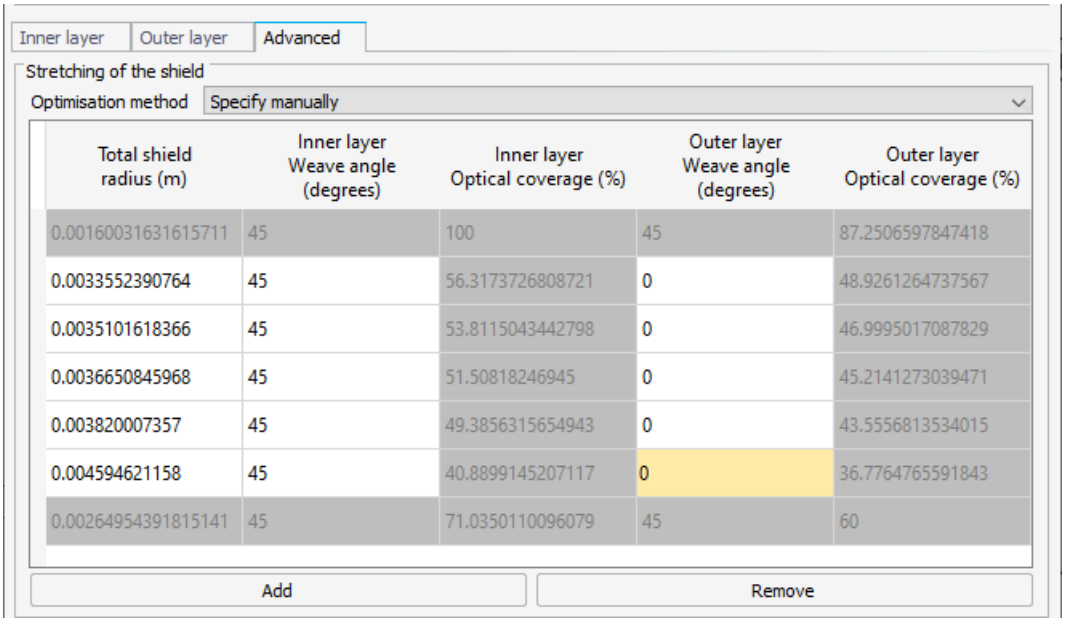


*Figure 209: The **Create Cable Shield** dialog, setting the optimisation method for the stretching of a braided shield layer.*

Note:

- **Total shield radius** must be between the minimum radius (first row, first cell) and maximum radius (last row, first cell).
- The values in the **Inner layer Optical coverage** and **Outer layer Optical coverage** columns, are calculated automatically from the **Total shield radius**, **Inner layer Weave angle** and **Outer layer Weave angle**.
- The **Inner layer Weave angle** or **Outer layer Weave angle** values must be between the minimum angle (first row, third cell) and maximum angle (last row, fifth cell).

8. In the **Label** field, add a unique label for the double cable shield.
9. Click **Create** to create the double cable shield and to close the dialog.

Related concepts
Cable Shield Layer Combinations

2.21.6 Defining a Cable Path

Create a cable path (route) along which cables are installed and specified as a series of straight lines.

Note: A cable path may not consist of overlapping sections.

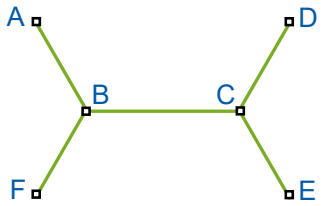


Figure 210: To create this cable path, five separate cable paths need to be defined namely: AB, BC, CD, EC and FB.

1. On the **Cables** tab, in the **Definitions** group, click the **Cable Path** icon.
2. Define the cable path using one of the following methods:
   - To specify the corner points, use point entry or add the U, V or N values directly for each point.
   - To import the points from a `ASCII` text file, click the **Import points** button.
     1. Under **Source file**, click **ASCII text file**.
     2. Under **Source file**, in the **Filename** field, browse to the file.
     3. Under **Delimiter**, click the relevant delimiter for your ASCII file.
   - To import the points from a **NASTRAN file**, click the **Import points** button.
     1. Under **Source file**, in the **Filename** field, browse to the file.
     2. Under **Settings**, in the **Scale factor to metres** field, modify the value to scale the cable path.
     3. Under **Settings**, in the **NASTRAN segment ID** field, enter the id of the segment to import.

   Note: Points imported from a NASTRAN file are assumed to be in metres.
3. View the cable path in the 3D view.
   a) Select the **Construct** tab in the model tree.

   Cable paths are displayed as dotted-blue lines in the 3D view.

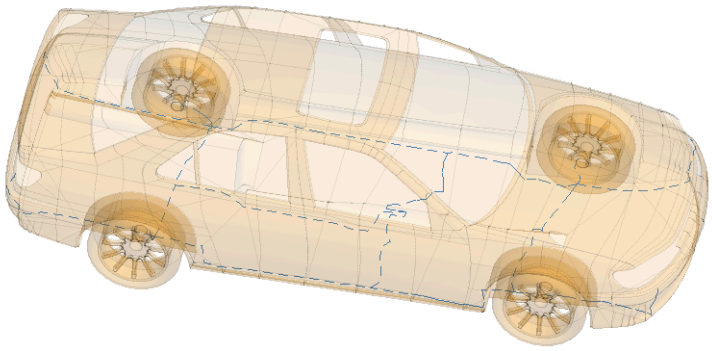


*Figure 211: Cable paths are visible in the 3D view when the **Construct** tab is selected. The cable paths are indicated by dotted-blue lines.*

4. In the **Label** field, add a unique label for the cable path.

5. Click the **Create** button to create the cable path and to close the dialog.

Related concepts
Routing a Cable Path at an Offset from the Geometry
Related tasks
Accessing the Cables Tab on the Ribbon

Advanced Settings for Cable Paths

Use advanced settings to specify the sampling point density, mesh refinement close to cable terminals and the cable cross section orientation along the cable path.

On the **Cables** tab, in the **Definitions** group, click the **Cable Path** icon. The advanced settings are available on the **Advanced** tab.

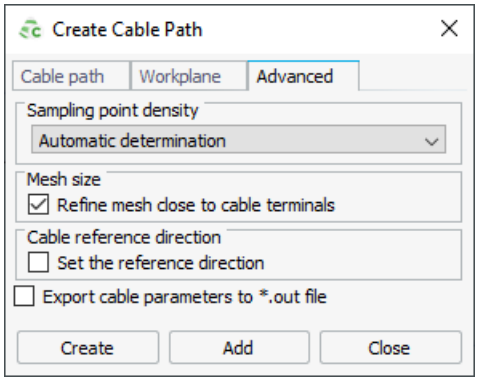


*Figure 212: The **Create Cable Path** dialog (**Advanced** tab).*

Sampling Point Density

Each cable path is subdivided into segments to compute the induced currents and voltages. At the centroid of each segment, the electric field strength and magnetic field strength are evaluated. You can specify the segment length to influence the accuracy of the computed results.

Automatic determination
This option allows CADFEKO to determine the segment lengths.

Specify maximum separation distance
This option allows you to specify the segment lengths.

Mesh Size

Refine mesh close to cable terminals
This option enables automatic mesh refinement near cable terminals.

Cable Reference Direction

Select the **Cable reference direction** check box to manually orientate the cable cross section along a cable path. Enter the following fields to change the orientation.

U, V and N
Specify the U coordinate, V coordinate and N coordinate at the start of the cable path for the cable reference direction.

Twist angle
Enter the angle at the cable path end to twist the cable cross section along the path (in degrees).

Note:
The **Cable reference direction** may not be parallel to the first cable path segment.

Export Cable Parameters

*Export cable parameters to *.out file*
This option exports the cable parameters such as inductance/capacitance matrices and transfer impedance/admittance to the `.out` file.

Related concepts
Example: Cable Reference Direction - Connected to an Installation
Example: Cable Reference Direction - Disconnected from an Installation
Example: Cable Reference Direction - No Installation
Related tasks
Accessing the Cables Tab on the Ribbon

Example: Cable Reference Direction - Connected to an Installation

Consider an example where a cable path is defined within a distance of $\frac{\lambda}{5}$ from a ground plane.

The cable cross section orientation is indicated by:

- dotted green line: **v** vector
- blue line: **u** vector
- solid dark green line: cable reference direction

Note: CADFEKO tries to orientate the **v** vector at the start of the cable path to align with the cable reference direction automatically using the constraint that the **u** and **v** vector must be perpendicular to the cable path.

The cable reference direction is defined normal to the installation or ground (in the "up" direction) and is not pointing towards it. In the cable schematic the cable is connected to the installation ⏚ and acts as a return path for the signal. No twist angle is defined along the cable path.

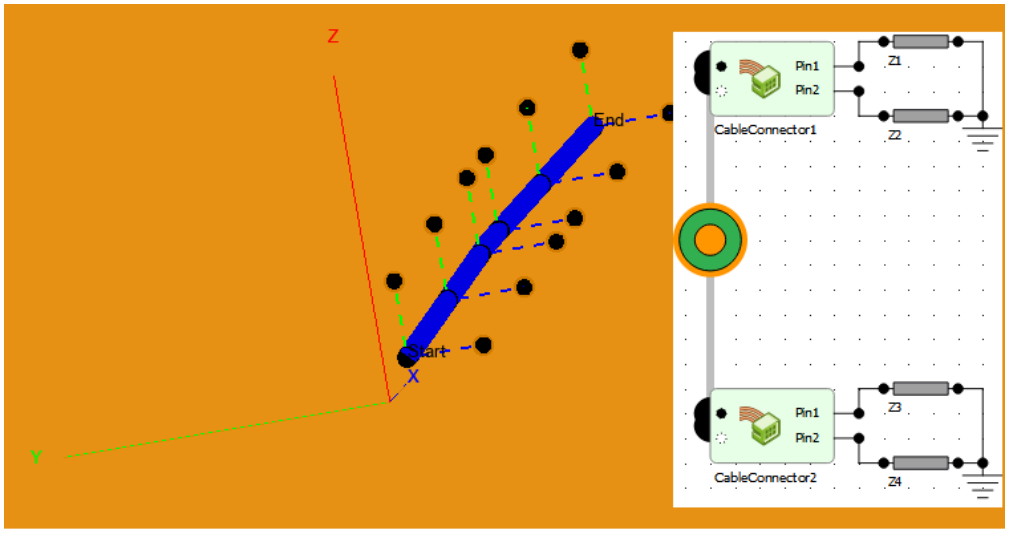


Figure 213: Specify the cable reference direction above a ground plane.

Note: This option can be used with only one signal in the cable harness as the installation acts as the return path.

Example: Cable Reference Direction - Disconnected from an Installation

Consider an example where a cable is disconnected from the installation / geometry.

The cable cross section orientation is indicated by:

- dotted green line: **v** vector
- dotted blue line: **u** vector
- solid dark blue line: cable reference direction

Note: CADFEKO tries to orientate the **v** vector at the start of the cable path to align with the cable reference direction automatically using the constraint that the **u** and **v** vector must be perpendicular to the cable path.

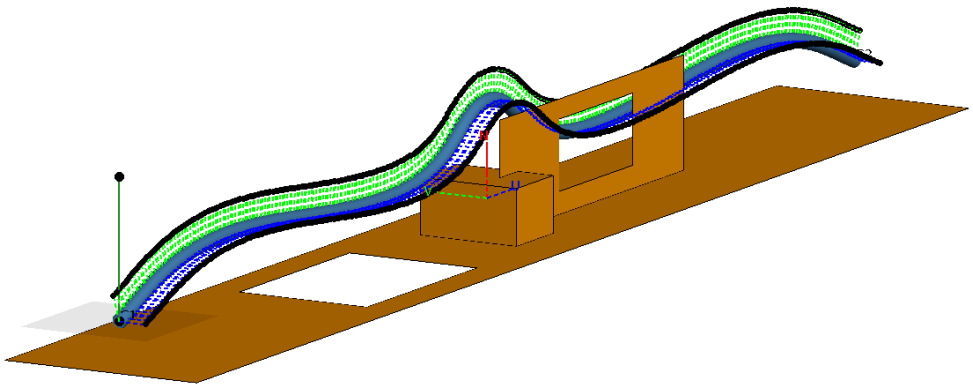


Figure 214: Cable reference direction with an installation present.

When using the cable reference direction where the cable is disconnected from the geometry, the following is required when connecting circuit elements in the schematic view:

- No bonding impedances are allowed in the harness, for example, no termination / interconnect ⏚ circuit connections to the installation.
- All cable paths should at least have two signals in the outermost problem.
- All cable paths in the harness should have an orientation vector defined (if that path does not have any nearby geometry / installation).

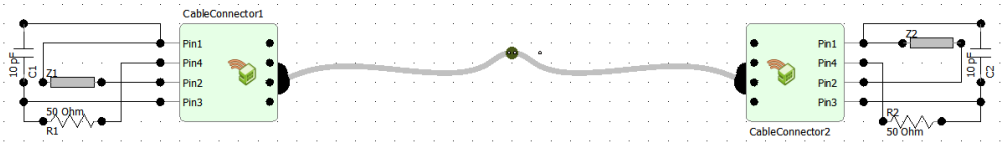


Figure 215: Cable schematic view with cable reference direction set.

Example: Cable Reference Direction - No Installation

Consider an example where a cable path is created, but with no geometry present in the model.

The **Cable reference direction** is defined normal to a fictitious ground (in the “up” direction). No twist angle is defined along the cable path.

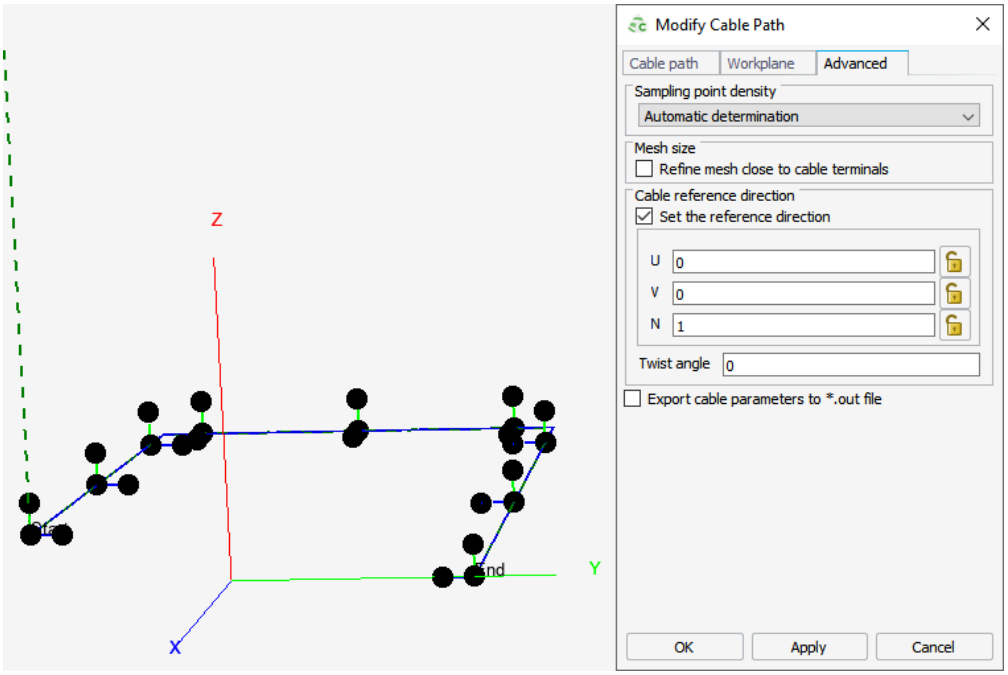


*Figure 216: Specify **Cable reference direction** with no geometry in the model.*

The cable cross section orientation is indicated by:

- dotted green line: **v** vector
- dotted blue line: **u** vector
- solid dark blue line: cable reference direction

> **Note:** CADFEKO tries to orientate the **v** vector at the start of the cable path to align with the cable reference direction automatically using the constraint that the **u** and **v** vector must be perpendicular to the cable path.

When using the cable reference direction with no installation present, the following is required in the cable harness and schematic view:

- No bonding impedances are allowed in the harness, for example, no termination / interconnect ⏚ circuit connections to the installation.
- All paths should at least have two signals in the outermost problem to create a return path.
- All paths in the harness should have an orientation vector defined (if that path does not have any nearby geometry / installation).

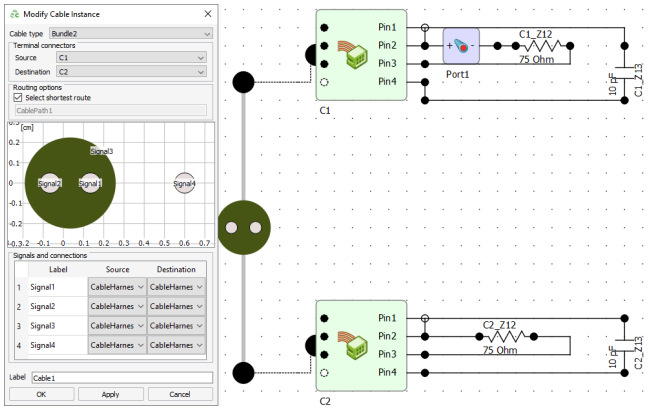


Figure 217: Cable harness and schematic view with cable reference direction set.

2.21.7 Defining Cable Connectors

Create a cable connector at the end terminal of a cable.

1. On the **Cables** tab, in the **Create Instance** group, click the **Cable Connector** icon.

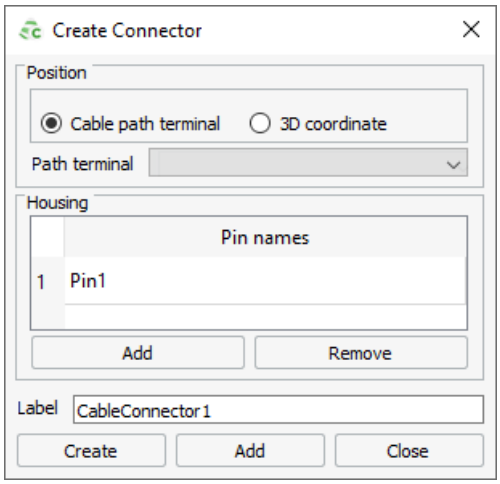


*Figure 218: The **Create Connector** dialog.*

Specify the position of the cable connector.

2. Under **Position**, select one of the following:
   - To select the start or end point of a cable path, click **Cable path terminal**.
     - To select the start or end point of a predefined cable path, from the **Path terminal** drop-down list, select the cable path you want to use.
     - To create a cable path, which is not yet defined in the model, click the icon to define a cable path.
   - To specify the X, Y and Z coordinates, click **3D coordinate**.

Add pins to the cable connector. Pins represent connection points between cables and cable components (for example, capacitors and inductors). Connections to the pins are made in the cable schematic view.

3. Under **Housing**, add pins to by clicking the **Add** button. Remove a pin by clicking the **Remove** button.
4. In the **Label** field, add a unique label for the cable connector.
5. Click the **Create** button to create the cable connector and close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

2.21.8 Defining a Cable Instance

Create a cable instance consisting of a single cable (for example, ribbon, cable bundle, coaxial cable) with its cable connectors that is routed along a cable path.

1. On the **Cables** tab, in the **Create Instance** group, click the **Cable Instance** icon.

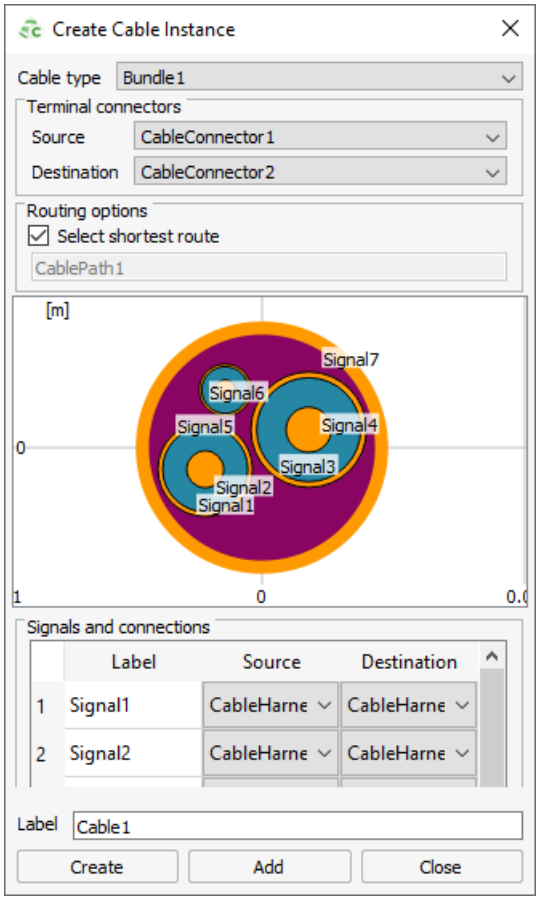


*Figure 219: The **Create Cable Instance** dialog.*

Specify the cable type.

2. From the **Cable type** drop-down list, select one of the following options:

- To specify a predefined cable, select the cable you want to use.
- To specify a cable, which is not yet defined in the model, click the icon to define a cable path.

3. Specify the start connector and end connector for the cable.
   a) Under **Terminal connectors**, from the **Source** drop-down list, select the start connector for the cable.
   b) Under **Terminal connectors**, from the **Destination** drop-down list, select the end connector the cable.

Specify the cable path along which the cable instance is routed.

4. Under **Routing options**, select one of the following:
   - To use a specific cable path, clear the **Select shortest route** check box. From the drop-down list, select the cable path you want to use.
   - To use the cable path with the shortest route between the specified start connector and end connector, select the **Select shortest route** check box.

For each conductor (signal) in the cable instance, specify the pins of the cable connector to which the conductor is connected.

> **Note:**
> - Each conductor in the cable instance is a signal.
> - A signal is connected to the pins of a cable connector.

5. Under **Signals and connections**, select the **Signal name**, **Source** (start pin) and **Destination** (end pin) of the connector (specified in Step 3) to which the signal is connected.
6. In the **Label** field, add a unique label for the cable instance.
7. Click **Create** to create the cable instance and to close the dialog.
8. In the **Label** field, add a unique label for the shielded enclosure.
9. Click **Create** to create the shielded enclosure and to close the dialog.

Related tasks
Accessing the Cables Tab on the Ribbon

2.21.9 Defining a Shielded Enclosure

Define a shielded enclosure that allows the outermost shield signals of different cable paths in a combined MoM/MTL cable harness to be connected to a shielded conductive enclosure defined in the full wave (3D) model.

1. On the **Cables** tab, in the **Create Instance** group, click the **Shielded Enclosure** icon.

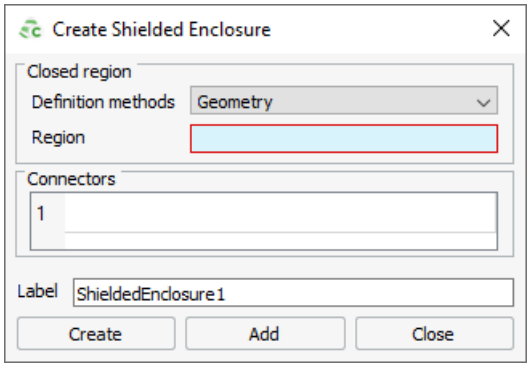


*Figure 220: The **Create Shielded Enclosure** dialog.*

2. From the **Definition methods** drop-down list, select one of the following options:
   - To connect the shielded enclosure to a geometry model, select **Geometry**. In the details tree, click on the region of the shielded enclosure.
   - To connect the shielded enclosure to the mesh face(s) of a model mesh, select **Mesh**. In the details tree, under **Triangle Labels**, click on a face or faces.
3. Under **Connectors**, specify the connectors related to the cable paths that terminate on the shielded enclosure by clicking on the connector(s) in the model tree.
4. In the **Label** field, add a unique label for the shielded enclosure.
5. Click **Create** to create the shielded enclosure and to close the dialog.

A shielded enclosure component is added to the cable schematic view with an orange background.

Related concepts
Solution Methods for Cables

2.21.10 Defining a Schematic Link

Define a connection(s) between circuit elements (defined in a cable schematic view) and the full wave model.

> **Note:** Only applicable to shielded cables solved with the MoM/MTL solution method.

1. On the **Cables** tab, in the **Create Instance** group, click the **Schematic link** icon.

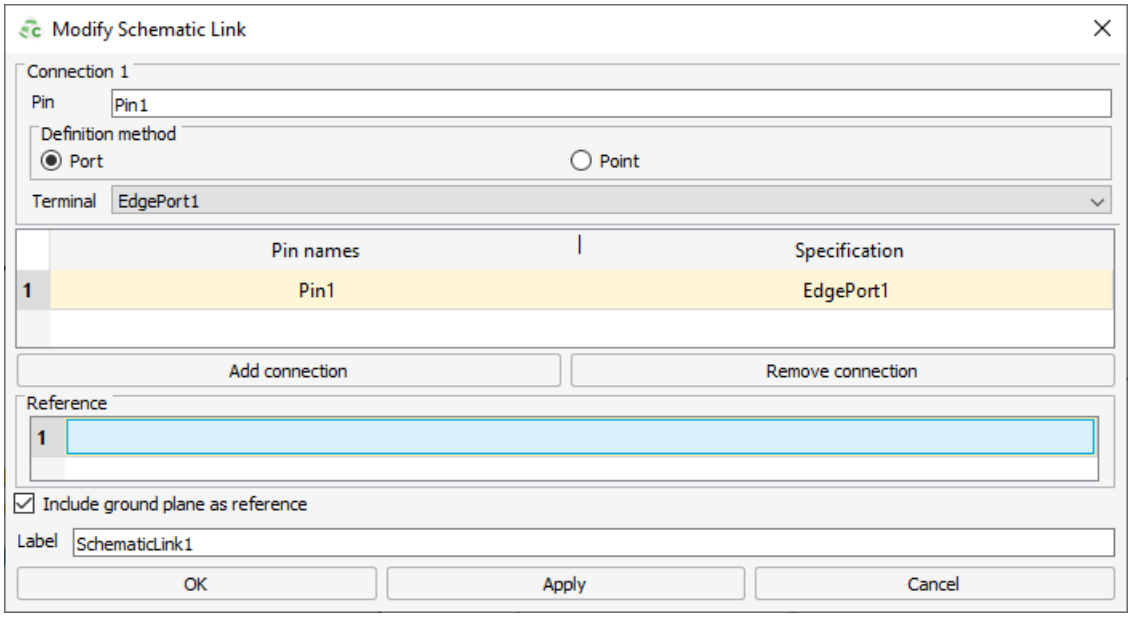


*Figure 221: The **Create Schematic Link** dialog.*

2. In the **Pin name** field, specify the name of the pin. This pin is the transitioning point from circuit elements (defined in the cable schematic view) to the full wave model.
3. Under **Definition method**, select one of the following options for each connection:
   - *Port*

     In the **Terminal** field, select a defined port from the drop-down list or create a new port. The positive port terminal is exposed on the schematic as a circuit node. The negative port terminal should coincide with the reference surface.

     > **Note:** The defined port is limited to wire/edge ports.

   - *Point*

     Under **Point**, specify the coordinates of the point that is exposed on the schematic as a circuit node.

   > **Note:** The physical connection is not indicated in the 3D view.

4. [Optional] Under **Reference**, specify the face(s) defining the common reference surface that translates to a single node of reference on the cable schematic view.

   > **Note:**
   > - If one of the connections references a port, a reference face is required.
   > - If none of the connections references a port, a reference face is optional.

5. Select the **Include ground plane as reference** check box to include a PEC infinite ground plane as part of the reference surface.

6. In the **Label** field, add a unique label for the schematic link.
7. Click **Create** to create the schematic link and to close the dialog.
8. Circuit connections between the cable path connector pins and the schematic link connector pins can be added to the schematic view.

Related concepts
Cable Schematic View
Ports

2.21.11 Defining a Cable Harness

Create a cable harness consisting of a collection of cable instances routed along a cable path with a solution method specified for the outer conductor.

1. On the **Cables** tab, in the **Create Instance** group, click the **Cable Harness** icon.

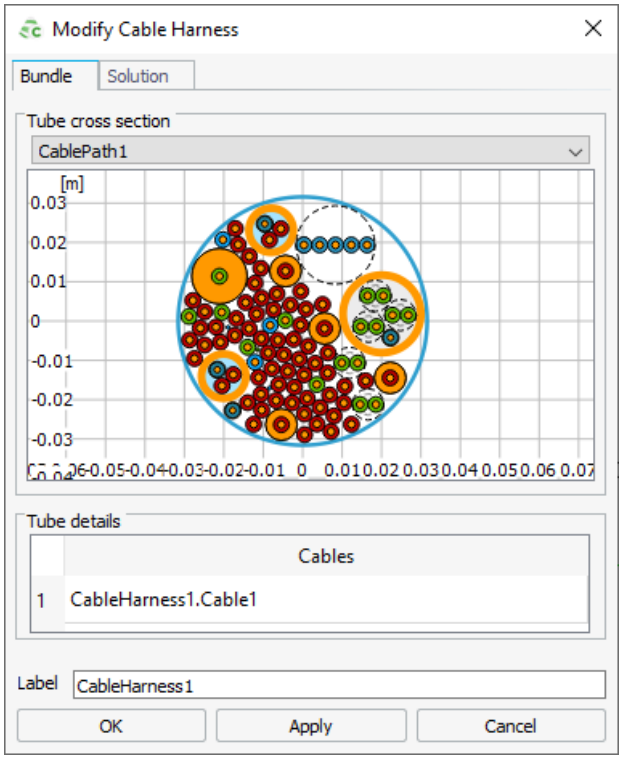


*Figure 222: The **Modify Cable Harness** dialog.*

> **Note:** An empty cable harness instance is created in the **Configuration** tab without launching a dialog.

View the cable instances routed along a specific cable path.

2. Open the right-click context menu for **CableHarness1**.
3. On the **Bundle** tab, under **Tube cross section**, select the cable path you want to view.

View the cable instances routed along the cable path in the preview and under **Tube details**.

Specify the cable coupling properties for the cable harness.

4. On the **Solution** tab, under **Cable coupling properties**, select one of the following:
   - To only consider the effect of external fields coupling into the cable harness, click **Irradiating**.
   - To only consider the effect of currents radiating from the cable harness, click **Radiating**.
   - To consider the combined effect of external fields coupling into the cable harness and currents radiating from the cable harness, click **Radiating (taking irradiation into account)**.
   - To consider the effect of intra coupling between cables in a harness (no external field coupling into the harness), click **Circuit crosstalk**.

   > **Note:** The wideband **Circuit crosstalk** solution is active when:
   > - All cable harnesses have **Circuit crosstalk** cable coupling properties enabled.
   > - No requests (except cable probe requests) are defined or the requests for each configuration is excluded.
   > - No sources (except cable sources) are defined in the configuration.

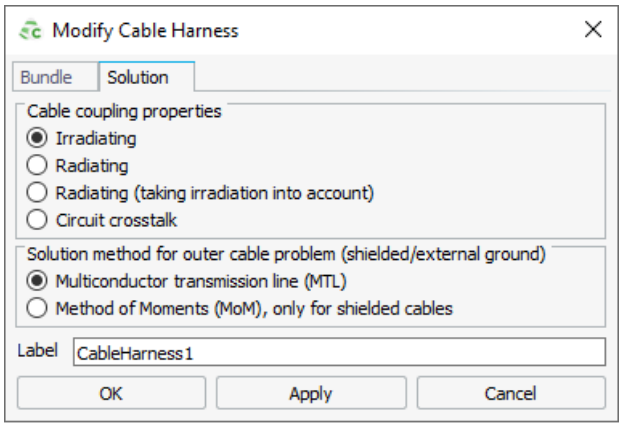


*Figure 223: The **Modify Cable Harness** dialog (Solution tab).*

Specify the solution method for the harness (outer cable).

5. On the **Solution** tab, under **Solution method for outer cable problem (shielded/external ground)**, select one of the following:
   - To solve the harness with the MTL, click **multiconductor transmission line (MTL)**.
   - To solve a harness containing only shielded cables with the MoM, click **Method of moments (MoM), only for shielded cables**.
6. In the **Label** field, add a unique label for the cable harness.
7. Click **OK** to apply the changes and to close the dialog.

Related tasks

Accessing the Cables Tab on the Ribbon

Viewing a Cable Harness in the Cable Schematic View

Excluding a Configuration from the Model

Solution Methods for Cables

Solve a cable harness with either the **Multiconductor transmission line (MTL)** method or the **Method of moments (MoM), only for shielded cables** method.

Multiconductor Transmission Line (MTL)

- Solve the model with the multiconductor transmission line theory, hybridised with the MoM or MLFMM.
- The cable path should ideally be within $\frac{\lambda}{10}$ of the conducting surface, although distances of up to $\frac{\lambda}{5}$ are allowed. Alternatively, a cable reference direction can be set on the cable path.
- Connections between the cable and MoM geometry are not allowed.

Method of Moments (MoM), Only for Shielded Cables

- Solve the model with the combined MoM/MTL solver.
- Any arbitrary cable path is allowed - the height restriction of the MTL does not apply.
- Cables must be shielded.
- Connections between the cable and MoM geometry are allowed.
- Cables can be connected to one another using a shielded enclosure that has a physical size and include radiation and grounding effects between the cable and the antenna or circuit board.

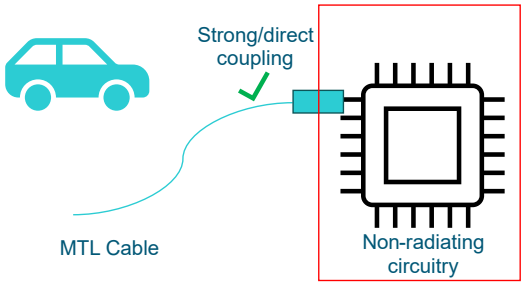


Figure 224: Practical example of a supported connection solution.

Related tasks
Accessing the Cables Tab on the Ribbon
Related reference
Supported Solution Method and Technique Combinations

2.21.12 Searching for a Cable Instance in the Model

Use the "Find cable" tool to locate a cable instance contained in a complex cable harness in the 3D view.

1. On the **Cables** tab, in the **Tools** group, click the **Find Cable** icon.

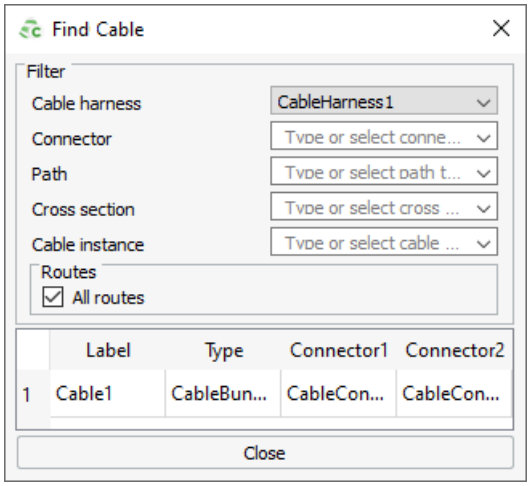


*Figure 225: The **Find Cable** dialog.*

2. Under **Filter**, from the **Cable Harness** drop-down list, select the cable harness you want to find.
3. [Optional] Under **Filter**, from the **Connector** drop-down list, narrow down the search by specifying a connector in the cable harness.
4. [Optional] Under **Filter**, from the **Path** drop-down list, narrow down the search by specifying a connector in the cable harness.
5. [Optional] Under **Filter**, from the **Cross section** drop-down list, narrow down the search by specifying a defined cable cross section in the cable harness.
6. [Optional] Under **Filter**, from the **Cable instance** drop-down list, narrow down the search by specifying a cable instance in the cable harness.

Specify the route along which to search for a cable instance.

7. Under **Routes**, select one of the following options:
   - To specify a cable instance in a specific route, clear the **All routes** check box.
   - To view all routes that satisfy the above criteria, select the **All routes** check box.

   Cable instances which satisfy the search criteria are listed in the table.
8. Click the cable instance in the table to highlight the cable instance in the 3D view and model tree.

Related tasks

Accessing the Cables Tab on the Ribbon

2.21.13 Combining Cable Instances

Convert multiple single conductor cables into a new type of cable after a harness description list is imported.

Note:

- The target cable cross-section is the new cross section of the cable instance which replaces or combines the selected cable instances.
- The target cable cross-section must be defined before it can be used to replace the single conductor.
- The cable instances to be combined must share the same cable path.

1. In the model tree, multi-select the cable instances to be combined.
2. On the **Cables** tab, in the **Tools** group, click the **Combine Cables** icon.

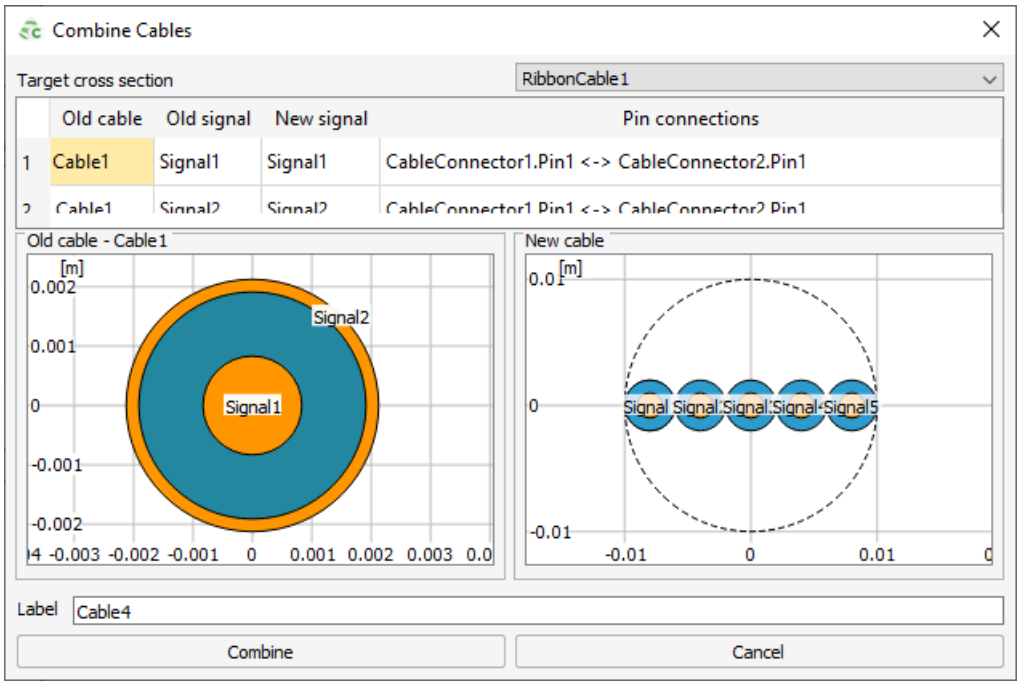


*Figure 226: The **Combine Cables** dialog.*

Specify the target cross-section (the cable cross section which replaces the selected cable instances).

3. From the **Target cross section** drop-down list, select the cable cross-section to replace the selected cable types.

   Note: The total number of signals for the cables to be combined must equal the total number of signals for the new target cable.

   For example, a single conductor (one signal) and twisted pair (two signals) may be combined to create a bundle cable with three signals.

   In the **Old cable** column, the selected cable instances to be combined are displayed.

In the **Old signal** column, the signals of the selected cable instances are displayed.

4. Click **Combine** to combine the selected cable instances and to close the dialog.
   The selected cable instances in the model tree are replaced with the new target cable instance.

Related tasks
Accessing the Cables Tab on the Ribbon

2.21.14 Rearranging Cables in a Cable Harness

Randomly place all tubes in a cable harness using the "Rearrange cross sections" tool.

1. In the model tree, select the cable harness for which you want to rearrange the cable instances.
2. On the **Cables** tab, in the **Tools** group, click the **Rearrange Cross Section** icon.
   In the model tree, double-click the cable harness to view the rearranged cross-section.

> **Note:** To rearrange the cables inside the cable bundle, from the **Modify bundle** dialog, click **Rearrange**.

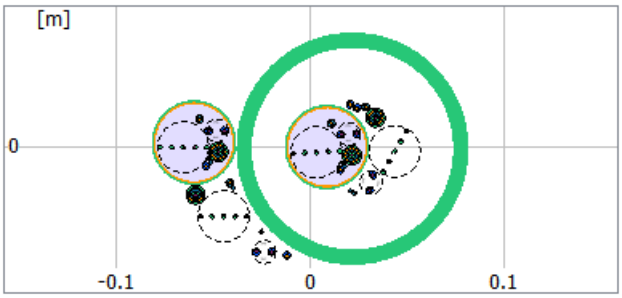


Figure 227: The cross-section of the cable harness before the rearrange.

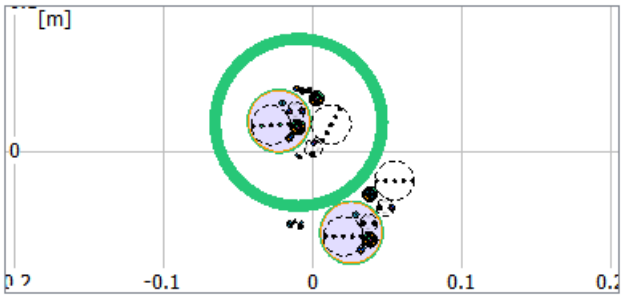


Figure 228: The cross-section of the cable harness after the rearrange.

Related tasks
Accessing the Cables Tab on the Ribbon

2.21.15 Cable Schematic View

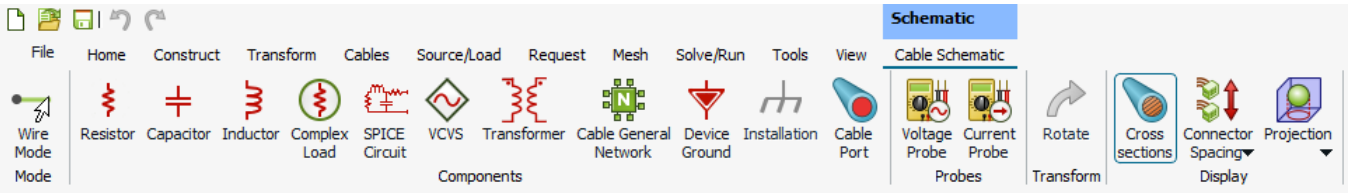
The cable schematic view allows you to add cable ports, complex loads, resistors, capacitors, inductors, external SPICE circuits, general networks (defined using N-port Touchstone files) and probes to a specific cable harness as well as connecting cables to one another.

Related tasks
Defining a Schematic Link

Viewing a Cable Harness in the Cable Schematic View

Open the cable schematic view for a specific cable harness to add circuit elements, probes or general networks to the connector pins or connect cables to one another.

1. On the **Cables** tab, in the **View** group, click the **Schematic** icon. From the drop-down list, select the relevant cable harness.
   The **Schematic** contextual tab set containing the **Cable schematic** contextual tab is displayed on the ribbon.



A new tab is opened in the 3D view that contains the three-dimensional cable harness projected onto a two-dimensional plane. The tab label indicates the specific cable harness that is viewed.

2. Click on the tab to view the cable harness in the cable schematic view.

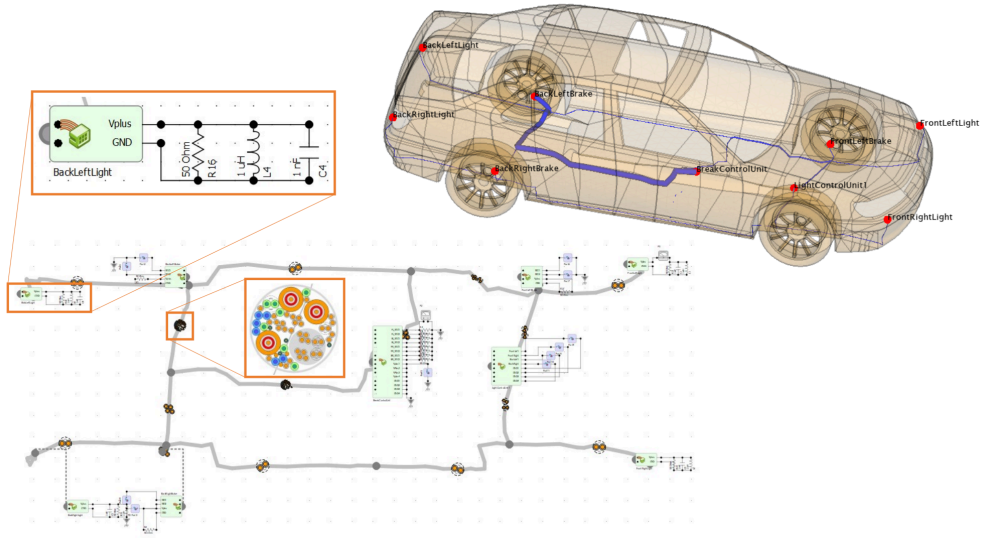


Figure 229: The cable schematic where the 3D cable harness is projected onto a 2D plane.

Note: A cable instance is displayed as a grey line (indicating the cable path) between its two connectors. A cross-section of the cable instances running along this cable path is also displayed on the cable path.

Connecting Circuit Elements and Pins in the Schematic View

Connect circuit elements to the cable connector pins in the cable schematic view.

1. On the **Schematic** contextual tabs set, on the **Cable schematic** tab, in the **Mode** group, click the **Wire Mode** icon.

A " + " at the cursor position indicates that the wire mode is enabled.

2. Click and drag to create a wire connection.
3. Release the mouse button at the position where the end of the wire connection is required.

> **Note:** Connection points for cable connectors and circuit elements are indicated by a dot.
>
> - A white dot indicates no connection between pins and/or circuit elements.
> - A black dot indicates a connection between pins and/or circuit elements.

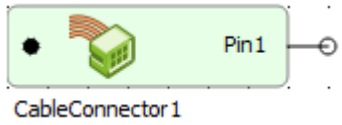


Figure 230: The white dot (to the right of the connector) indicates that no connection is made to the connector pin.

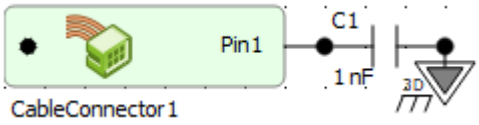


Figure 231: The black dot (to the right of the connector) indicates that a connection is made to the connector pin.

Cable Schematic Components

View the circuit components that can be added to the cable schematic view.

Icon	Name	Description
	Resistor	Add a resistor to the cable schematic view.
	Capacitor	Add a capacitor to the cable schematic view.
	Inductor	Add an inductor to the cable schematic view.
	Complex Load	Add a complex load to the cable schematic view.
	SPICE Circuit	Import a SPICE circuit from a file to define the circuit.
	VCVS	Add a voltage controlled voltage source (VCVS) to the cable schematic view.

Icon	Name	Description
	Transformer	Add a transformer to the cable schematic view.
	Cable General Network	Import an N-port Touchstone from file to define a general network.
	Device Ground	Add a device ground to the schematic view with the option to short to the installation. The following symbols are used: •  Open circuit •  Short circuit •  For combined MoM/MTL harnesses, the device ground indicates the transitioning point from circuit to full wave solver. Connections beyond the device ground should be included in the full wave model.
	Installation	For MTL harnesses, add a connection to the installation (3D model geometry).
	Cable Port	Add a cable port to the cable schematic view. Voltage sources and loads can be applied to cable ports.
	Voltage Probe	Define a voltage probe to measure the voltage between two points and add to the cable schematic view.
	Current Probe	Define a current probe to measure the current at a point and add to the cable schematic view.

Note: Schematic circuit connections take preference over mesh connections when working with the combined MoM/MTL solver.

Related concepts
Device Ground and Installation Ground

Device Ground and Installation Ground

For cable harnesses, there is a distinction between a local circuit device ground and the global installation ground.

The multiconductor transmission line (MTL) method supports a distinction in the concepts local circuit device ground and global installation ground.

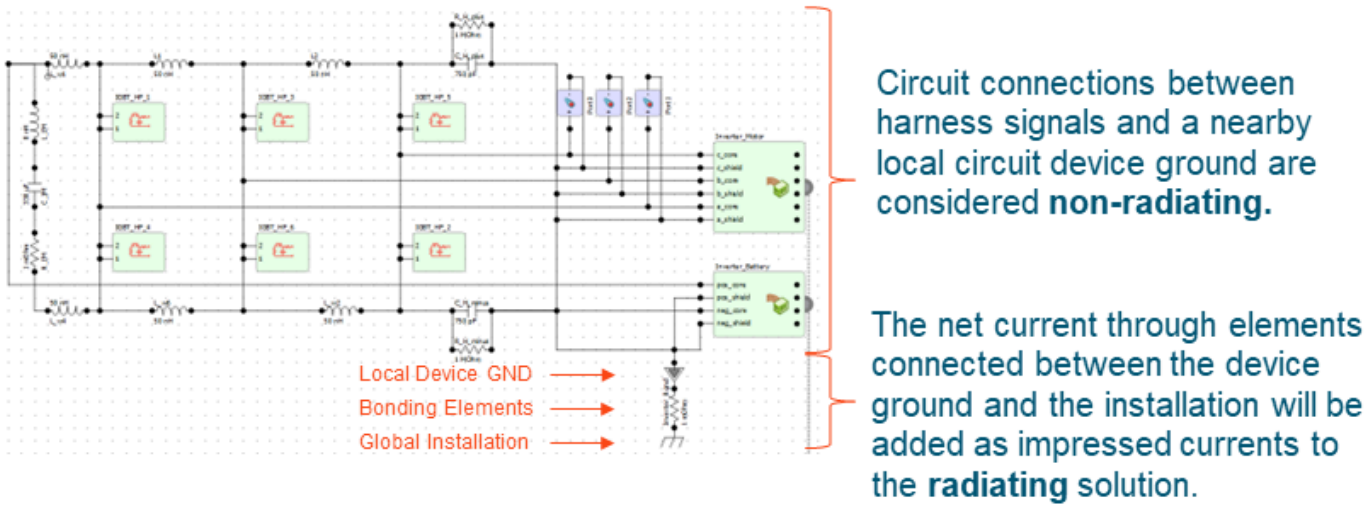


Figure 232: An example showing the distinction between a local circuit device ground and the global installation.

For the combined MoM/MTL harnesses, the device ground indicates the transitioning point from circuit to full wave solver. Connections beyond the device ground should be included in the full wave model.

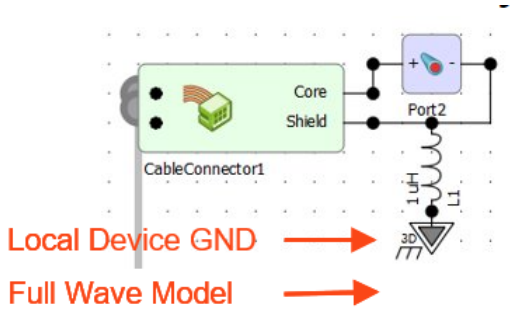


Figure 233: An example showing the transitioning point from circuit to full wave solver.

Related concepts
Solution Methods for Cables
Related reference
Cable Schematic Components

Cable Schematic Display Options

View the display options for the cable schematic view.

Icon	Name	Description
	Rotate	Rotate the selected item.
	Cross Sections	Show / Hide the cable-cross section display.
	Connector Spacing	Set the spacing factor between connectors on the cable schematic view.
	Projection	Projection on the XY plane.
	Projection	Projection on the XZ plane.
	Projection	Projection on the YZ plane.

2.22 Solution Frequency

For a frequency domain result, the electromagnetic fields and currents are calculated at a single frequency or frequency range. When the finite difference time domain (FDTD) solver is used, the frequency must be specified to convert the native time domain results to the frequency domain.

2.22.1 Frequency Options

The supported frequency options are single frequency, continuous range, linearly spaced discrete points, logarithmically spaced discrete points and list of discrete points. Select the frequency option that is best suited to the model and the specified requests.

> **Note:** Frequencies can be specified globally or per configuration.

On the **Source/Load** tab, in the **Settings** group, click the **Frequency** icon.

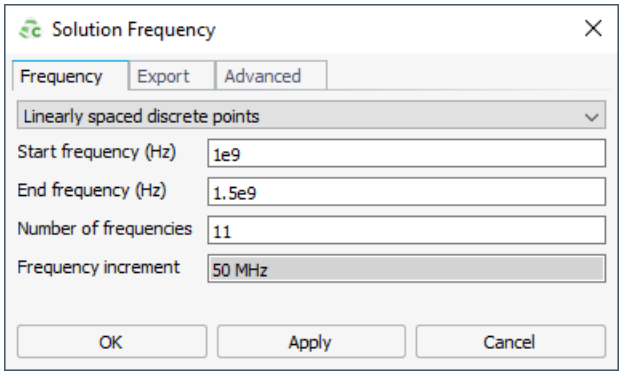


*Figure 234: The **Solution frequency** dialog (**Frequency** tab).*

Single frequency
: The requested results are calculated at a single frequency.

Continuous (interpolated) range
: The requested results are calculated using adaptive sampling in the range **Start frequency** to **End frequency**. The sampling algorithm uses finer sampling in areas where the results change rapidly to ensure that all resonance effects are calculated accurately.

> **Note:** Use this option with as little result requests as possible, since the requested results are interpolated and increases the run time.

Linearly spaced discrete points
: The requested results are calculated at a fixed number of linearly spaced points between the **Start frequency** and the **End frequency**. This option is typically used when the solution is required at exact frequencies.

Logarithmically spaced discrete points
The requested results are calculated at a fixed number of logarithmically spaced points between the **Start frequency** and the **End frequency**. This is typically used over a wide bandwidth.

List of discrete points
The requested results are calculated at a list of discrete points. This is typically used when the exact frequencies are known where the solution is required.

Tip: Use point entry (Ctrl+Shift+left click) to set the frequency to a defined variable in the model tree.

Related concepts
Multiple Configurations

2.22.2 Continuous Frequency (Advanced Settings)

Choose from a number of advanced settings for a continuous (interpolated) simulation frequency to ensure a computationally efficient solution.

On the **Source/Load** tab, in the **Settings** group, click the **Frequency** icon. On the **Solution frequency** dialog, click the **Advanced** tab.

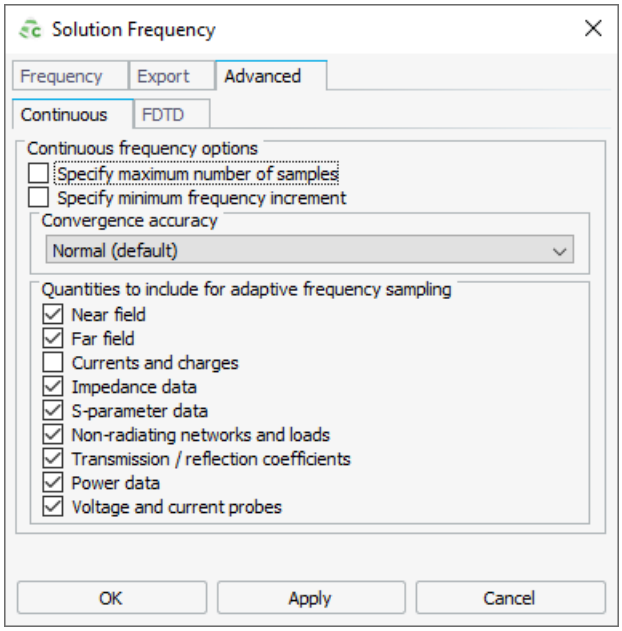


*Figure 235: The **Solution frequency** dialog (**Advanced** tab).*

Maximum number of samples
This option limits the number of frequencies solved and as a result, the runtime.

Warning: If the solution is not fully converged, the results may be inaccurate.

Minimum frequency increment
This option limits the minimum frequency increment when refining the frequency. It is useful if there are small discontinuities in the results.

Convergence accuracy

- **High**: More samples, highly resonant structure
- **Normal**: Default
- **Low**: Fewer samples, smooth frequency response

Quantities to include for adaptive frequency sampling
This option allows you to select the quantities to include for the adaptive frequency sampling. Quantities that are not selected, are calculated at the discrete solution frequency points.

> **Tip:** The defaults are recommended. For example, including **Currents and charges** in a model with many triangles increases the run-time due to interpolation.

2.22.3 Continuous Frequency (Export Settings)

Choose the frequency stepping and number of samples exported to a `.isd` or `.snp` file in a solution with continuous (interpolated) frequency.

On the **Source/Load** tab, in the **Settings** group, click the **Frequency** icon. On the **Solution frequency** dialog, click the **Export** tab.

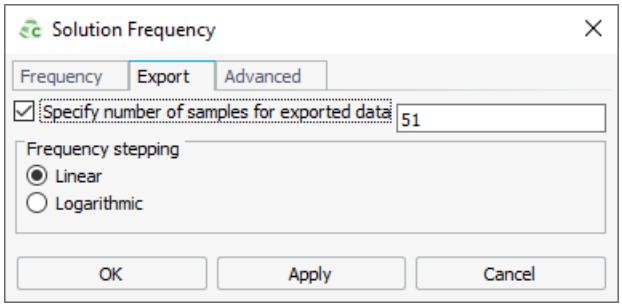


*Figure 236: The **Solution frequency** dialog (**Export** tab).*

Specify number of samples for exported data
This option allows you to specify the number of discrete frequency samples to be extracted from the continuous data when exporting to a `.isd` file or a `.snp` file.

Frequency stepping
This option allows you to select either a **Linear** increment or a **Logarithmic** increment for the extracted discrete frequency samples for export to a `.isd` file or `.snp` file.

2.22.4 FDTD Frequency Settings

A number of settings related to the time interval are available when using the FDTD solver.

On the **Source/Load** tab, in the **Settings** group, click the **Frequency** icon. On the **Solution frequency** dialog, click the **Advanced** tab. Click the **FDTD** tab to show the finite difference time domainsettings.

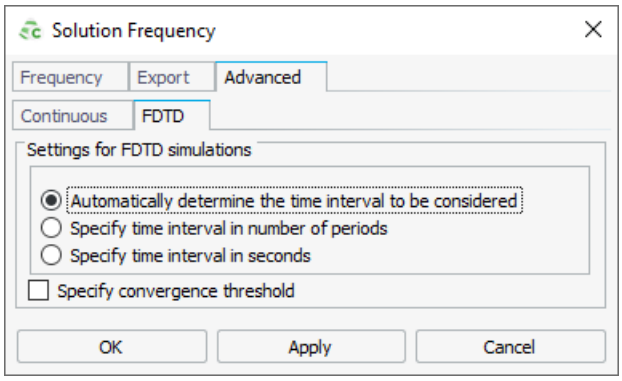


*Figure 237: The **Solution frequency** dialog (**Advanced** tab).*

Automatically determine the time interval to be considered

Select this option to automatically determine the time interval[32] based on the time signals used by configuration sources, the size of the computational domain and the material properties. An estimate is made for the propagation time required for the time signal to pass through the domain.

Specify the time interval in number of periods

Select this option to specify the maximum time interval and / or minimum time interval in sinusoidal periods. A period is defined as $\frac{1}{f_{centre}}$, where f_{centre} is the average between the upper and lower frequencies in the requested band.

Specify time interval in seconds

Select this option to specify the absolute maximum time interval and / or minimum time interval in seconds.

Specify convergence threshold

Select the **Specify convergence threshold** check box to specify the convergence threshold for the FDTD simulation. For example, to specify a threshold of -100 dB, enter a value of 1e-5. The simulation terminates if the threshold is reached and the simulation time is larger or equal to the minimum simulation time.

32. A time interval is the time duration for which the model is simulated.

2.23 Power

The excitation of an antenna is normally specified as a complex voltage, but it may be useful to specify the total radiated or source power instead. The result is then scaled to yield the desired source power level.

Note: Power can be specified globally or per configuration.

Note:

- Feko uses peak magnitude for all complex values. Voltage and current sources must be specified with peak magnitude (as opposed to root mean square values) if no power scaling is performed.
- Power settings are specified as time-averaged values.

On the **Source/Load** tab, in the **Settings** group, click the **Power** icon.

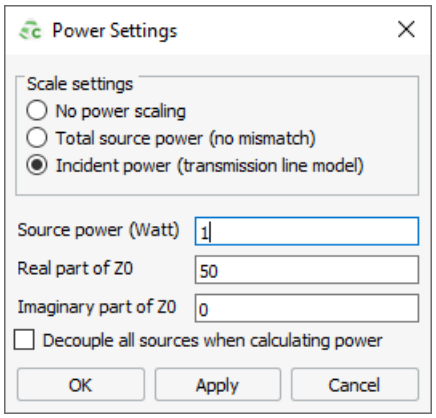


*Figure 238: The **Power Settings** dialog.*

No power scaling

Select this option to calculate the results using the specified source magnitudes.

Tip: A plane wave source has an infinite extent and therefore infinite power. If a model contains plane wave sources, select **No power scaling**.

Total source power (no mismatch)

Select this option to scale the results such that the total source power (the sum of the power delivered by all the individual sources in a model with multiple sources) is equal to the amount specified in the **Source power (Watt)** field. No mismatch is taken into account.

Note: This option can be used with any source, except plane waves.

Incident power (transmission line model)

Select this option to assume that all structures are fed using transmission lines with a complex characteristic impedance Z_0. The **Source power** field specifies the sum of the incident power from all these transmission lines. If there is a mismatch between the transmission line impedance and the structure input impedance at the excitation point, a fraction of the incident power will be reflected to the source. This is the mismatch loss.

Feko always calculates the total source power for all solutions. For large models or models with many sources, the calculation of mutual coupling (which is required for accurate source power calculations), can be time-consuming.

Select the **Decouple all sources when calculating power** check box to ignore the mutual coupling for Hertzian electric / magnetic dipoles or impressed line current elements when calculating the source power.

This is acceptable in the following cases:

- when sources, which in terms of the wavelength, are relatively far from each other and from other structures in the model
- when accurate power values are not required

Gain and directivity extraction are based on source power and are in general likely to be inaccurate if the **Decouple all sources when calculating power** option is selected.

Related concepts
Multiple Configurations

2.24 Ports

A port is a mathematical representation of where energy can enter (source) or leave a model (sink). Use a port to add sources and discrete loads to a model.

The following types of ports are supported:

1. wire port
2. edge port
3. microstrip port
4. waveguide port
5. FEM line port
6. FEM modal port
7. cable port

Use an appropriate port for a model to obtain more accurate results.

Generally, ports are created on geometry items and such ports contain only a geometry instance. When the geometry part (containing a port) is meshed, a mesh port instance is created automatically. If the mesh is unlinked, the mesh instance of the port is displayed in the model tree.

Ports can be created directly on unlinked meshes, but this option should only be used for imported meshes or in cases where the geometry is no longer available.

Note: Conventional current is defined as the current that flows through the port from the negative side to the positive side.

View the geometry and mesh instances of the ports in the model tree (Construct tab).

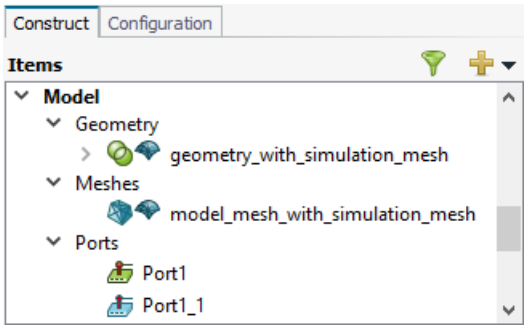


Figure 239: Example of (1) a geometry part that was meshed and its geometry port instance (port icon in green) and (2) a mesh part that has a simulation mesh with a mesh port instance (port icon in blue).

Note:

- The icon indicates the geometry instance of the port.
- The icon indicates the mesh instance of the port.

2.24.1 Wire Ports

Wire ports can be applied to wires (geometry), mesh segments or on a vertex between segments.

Apply a wire port to a vertex when:

- a wire or mesh segment is connected to a structure and the phase difference from the end point to the first segment centre results in a significance effect on the input impedance.
- a wire or mesh segment is connected between an infinite ground plane and a UTD plate.

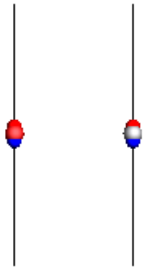


Figure 240: A wire port on a segment (on the left) and a wire port on a vertex (on the right) in the 3D view.

Creating a Wire Port

Apply a wire port to wires (free edges that do not form a face boundary).

1. On the **Source/Load** tab, in the **Ports** group, click the **Wire Port** icon.

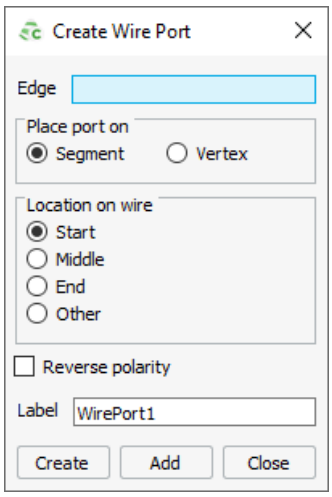


*Figure 241: The **Create Wire Port** dialog.*

Specify the wire where the port is to be placed.

2. In the **Edge** field, use point entry to specify the wire using one of the following workflows:
   - In the 3D view, click on the relevant wire.
   - In the details tree, click on the relevant wire.
3. Specify whether the port is to be placed on a segment or a vertex (after the wire is meshed).

- To add the wire port to a segment, select **Segment**.
- To add the wire port to a vertex between two segments, select **Vertex**.

  Vertex ports are mainly used where wires are connected to other structures and the phase difference from the end point to the centre of the first segment would have a significant effect on the input impedance.

  > **Note:** Vertex ports can be set on the end of wires that are connected to infinite ground planes and UTD plates.

4. Specify where the port is located on the wire. Under **Location on wire**, select one of the following:
   - To specify one of the predefined geometric points on a line, select **Start**, **Middle** or **End**.
   - To specify an arbitrary position along the wire in terms of the position as a percentage of the total wire length, select **Other**, where 0% is interpreted as the start point and 100% as the end point.

     If the wire is modified after the port was specified, the port maintains the same relative position along the wire. For example, if the port was one third from the end of a wire and the wire is shortened for a higher frequency, the port remains one third from the end of the shortened wire.

     > **Tip:** Enter a named point or a "pt" expression in the **%** field to fix the absolute position of the port. The port is then located at the projection of the point onto the wire. If the wire is modified, the point will remain as close as possible to the absolute position.

5. [Optional] If the polarity of the port is to be reversed, select the **Reverse polarity** check box.
6. In the **Label** field, add a unique label for the wire port.
7. Click **Create** to create the wire port and to close the dialog.

Creating a Wire Port (Mesh)

Apply a wire port directly on a mesh segment or vertex (imported mesh or an unlinked mesh).

1. On the **Source/Load** tab, in the **Ports** group, click the **Wire Port** icon.

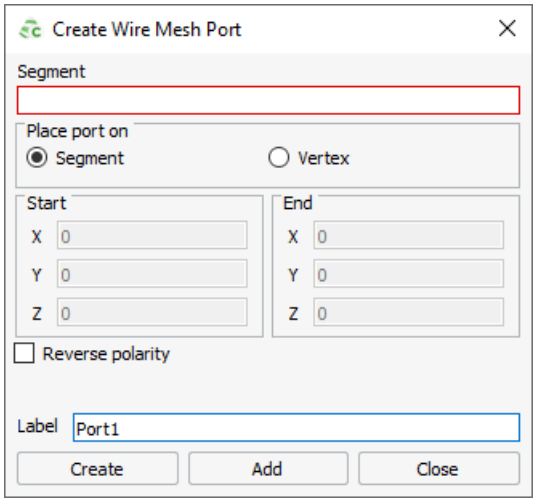


*Figure 242: The **Create Wire Mesh Port** dialog.*

2. Specify the mesh segment where the port is to be placed.
3. In the **Segment** field, use point entry to specify the mesh segment using one of the following workflows:
   - In the 3D view, click on the relevant mesh segment.
   - In the details tree, click on the relevant mesh segment.
4. Specify whether the port is to be placed on a mesh segment or a vertex.
   - To add the wire port to a segment, select **Segment**.
   - To add the wire port to a vertex between two segments, select **Vertex**.
5. [Optional] If the polarity of the port is to be reversed, select the **Reverse polarity** check box.
6. In the **Label** field, add a unique label for the wire port.
7. Click the **Create** button to create the wire port and close the dialog.

2.24.2 Edge Ports

Apply an edge port to an edge between two sets of faces.

Faces referenced in the port definition must belong to the same part for a valid port definition.

> **Note:** An edge port can be applied to a single UTD face, but the faces on the other side of the defined port must be standard MoM faces.

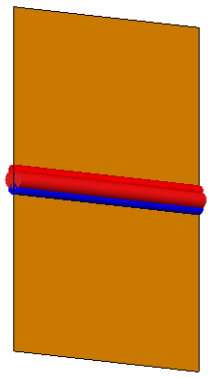


Figure 243: The edge port in the 3D view. The side of the positive faces is indicated with a red cylinder and the negative faces with a blue cylinder.

Related concepts
Create Edge Port on Finite Substrate Macro

Creating an Edge Port

Apply an edge port to an edge defining the boundary between two sets of faces.

1. On the **Source/Load** tab, in the **Ports** group, click the **Edge Port** icon.

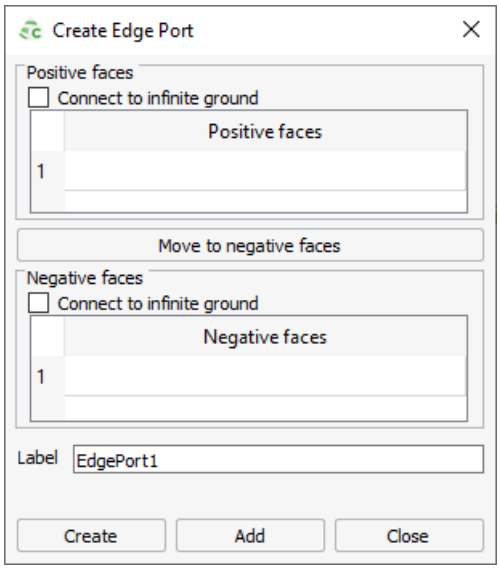


*Figure 244: The **Create Edge Port** dialog.*

When an infinite ground plane is present in the model, any edges that lie in the plane can be excited with respect to that ground plane.

2. Connect a side of the port to the infinite ground:
   - To connect the positive side of the port, under **Positive faces**, click the **Connect to infinite ground** check box.

- To connect the negative side of the port, under **Negative faces**, click the **Connect to infinite ground** check box.

Specify the positive faces of the edge port.

3. In the **Positive faces** table, use point entry to specify the positive faces using one of the following workflows:
   - In the 3D view, click on the relevant face.
   - In the details tree, click on the relevant face.

Specify the negative faces of the edge port.

4. In the **Negative faces** table, use point entry to specify the negative faces using one of the following workflows:
   - In the 3D view, click on the relevant face.
   - In the details tree, click on the relevant face.
5. [Optional] To switch a face between the lists, select one of the following workflows:
   - Double-click the face entry.
   - Click **Move to ... faces**.
6. In the **Label** field, add a unique label for the edge port.
7. Click the **Create** button to create the edge port and to close the dialog.

Creating an Edge Port (Mesh)

Apply an edge port to an edge between two sets of mesh faces (imported mesh or an unlinked mesh).

1. On the **Source/Load** tab, in the **Ports** group, click the **Edge Port** icon.

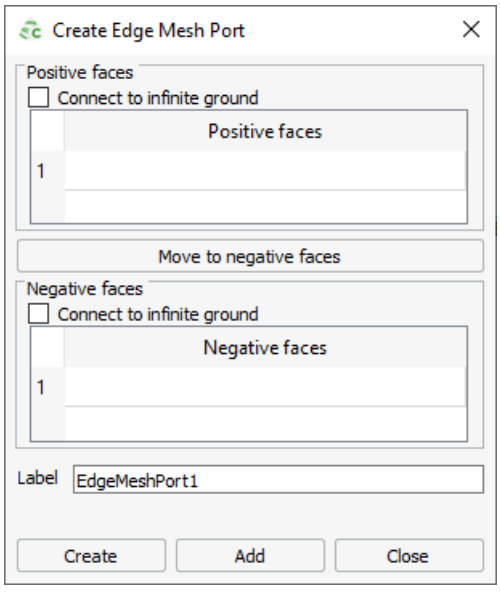


*Figure 245: The **Create Edge Mesh Port** dialog.*

When an infinite ground plane is present in the model, any edges that lie in the plane can be excited with respect to that ground plane.

2. Connect a side of the port to the infinite ground:
   - To connect the positive side of the port, under **Positive faces**, click the **Connect to infinite ground** check box.
   - To connect the negative side of the port, under **Negative faces**, click the **Connect to infinite ground** check box.

Specify the positive faces of the mesh edge port.

3. In the **Positive faces** table, use point entry to specify the positive faces using one of the following workflows:
   - In the 3D view, click on the relevant face.
   - In the details tree, click on the relevant face.

Specify the negative faces of the mesh edge port.

4. In the **Negative faces** table, use point entry to specify the negative faces using one of the following workflows:
   - In the 3D view, click on the relevant face.
   - In the details tree, click on the relevant face.
5. [Optional] To switch a face between the lists, select one of the following workflows:
   - Double-click the face entry.
   - Click **Move to ... faces**.
6. In the **Label** field, add a unique label for the mesh edge port.
7. Click the **Create** button to create the edge port and to close the dialog.

Edge Port on a Thick Dipole

Excite a dipole made from a cylinder and add an edge port.

1. Create a cylinder.
2. Split the cylinder to create two sections.

   CADFEKO automatically adds a face on the split plane to keep the two split parts as solid parts.

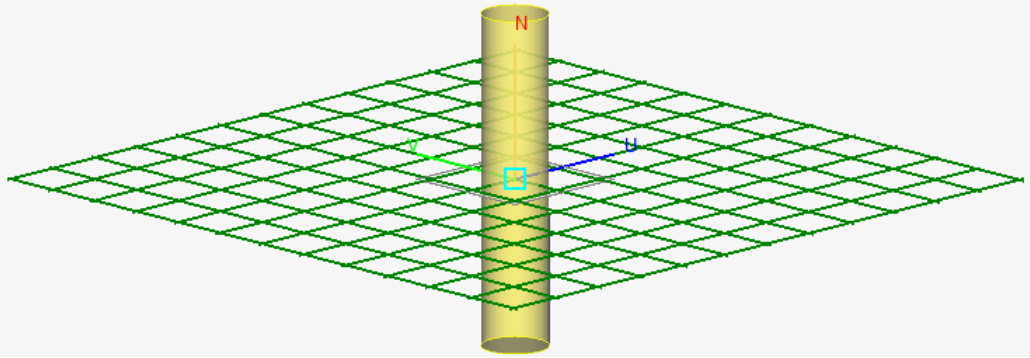


Figure 246: The preview of the split operation on the thick dipole.

3. Union the two sections to ensure the faces in the edge port belong to the same part.
4. Delete the face on the split plane.

   When specifying an edge port, all faces bordering the edge must be specified in the edge port definition. Delete the extra face created when the two sections were unioned. Deleting the face results in a single region with only two faces bordering the edge.

   > **Tip:** Model the cylinder as Free Space (shell object) to avoid the face.

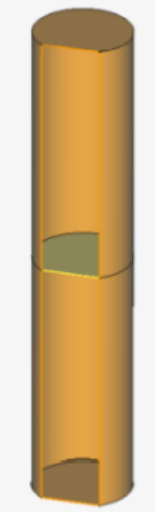


Figure 247: Cutplane view of the thick dipole with face in the middle to be deleted after Union.

5. Add the edge port.

   Specify the outer face of the one section as the positive face. Specify the outer face of the second section as the negative face. The result is an edge port which is not straight but closes on itself.

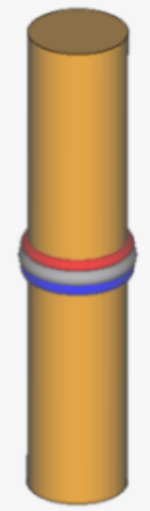


Figure 248: An edge port is added to the thick dipole. Note the edge closes on itself.

Using an Edge Port with the FDTD solution method

The following requirements must be met when using an edge port with the FDTD solution method:

- All meshed port faces must lie in the same plane. Port faces which do not lie in the same plane results in conflicting potentials at a point.
- All meshed port faces must point in the same direction.

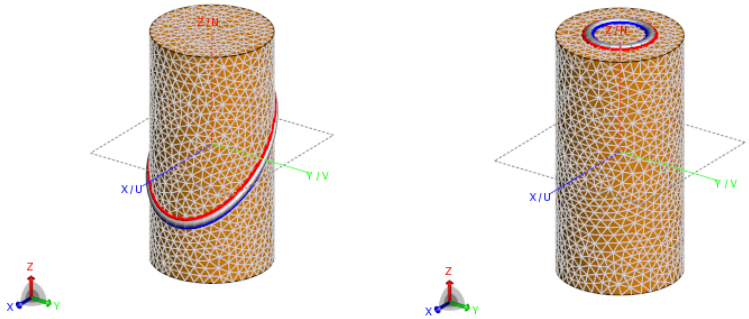


Figure 249: Examples of valid edge ports on a triangular mesh.

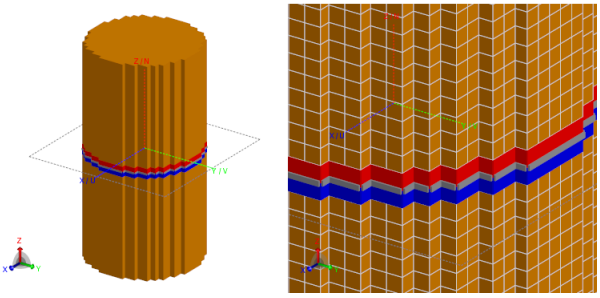


Figure 250: A valid edge port on a voxel mesh. Note the meshed port faces all lie in the same plane.

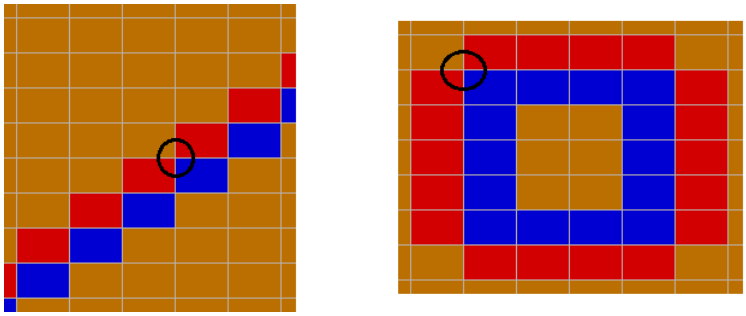


Figure 251: If a voxel mesh is applied to the same model as the top example and a similar edge port is specified, it results in an invalid edge port as displayed in the section views. The black circles indicate an example of a point with conflicting potentials.

2.24.3 Microstrip Ports

Apply a microstrip port to represent a feed line in a microstrip structure. A microstrip port is specified on an edge or a set of edges that form a continuous, straight, horizontal (lie in a constant Z plane in the global coordinates) edge that borders only a single face.

> **Note:** To apply a microstrip port, the model must contain a planar dielectric substrate with a conducting ground plane at the bottom.

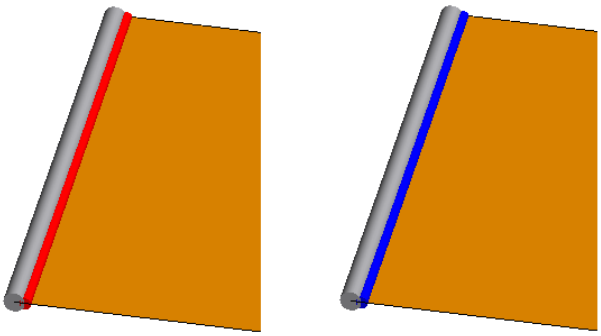


Figure 252: Examples of a microstrip port connected to an edge. The positive side of the microstrip port is indicated in red (on the left) and the negative side of the port indicated in blue (on the right).

Creating a Microstrip Port

Apply a microstrip port to a geometry face.

1. On the **Source/Load** tab, in the **Ports** group, click the **Microstrip Port** icon.

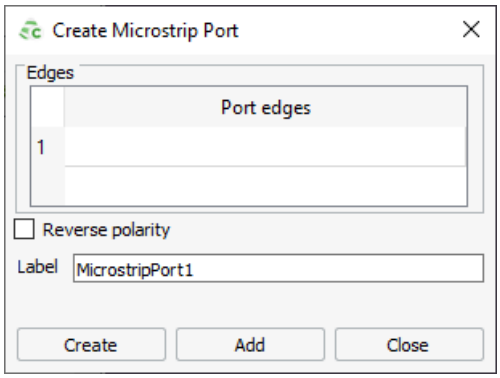


*Figure 253: The **Create Microstrip Port** dialog.*

2. In the **Port edges** table, specify the edges for the microstrip port.
3. [Optional] If the polarity of the port is to be reversed, select the **Reverse polarity** check box.
4. In the **Label** field, add a unique label for the microstrip port.
5. Click **Create** to create the microstrip port and to close the dialog.

Creating a Microstrip Port (Mesh)

Apply a microstrip port between vertices of an imported mesh or an unlinked mesh.

1. On the **Source/Load** tab, in the **Ports** group, click the **Microstrip Port** icon.

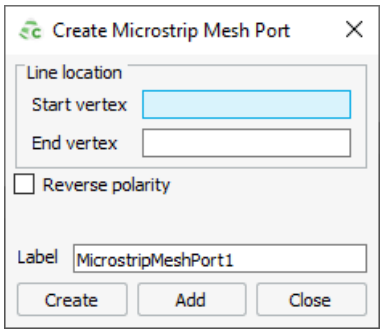


*Figure 254: The **Create Microstrip Mesh Port** dialog.*

2. In the **Start vertex** field, add the start vertex point by using point entry and clicking on the relevant vertex in the 3D view.
3. In the **End vertex** field, add the end vertex point by using point entry and clicking on the relevant vertex in the 3D view.
4. [Optional] If the polarity of the port is to be reversed, select the **Reverse polarity** check box.
5. In the **Label** field, add a unique label for the microstrip port.
6. Click **Create** to create the microstrip port and to close the dialog.

2.24.4 Waveguide Ports

A waveguide port is used to define the planes of excitation for waveguide structures.

> **Note:** Waveguide ports can be applied to:
> - a single flat face solved using SEP.
> - a single flat face on the boundary of a FEM region.

Three basic waveguide cross-sections are supported:

- Rectangular
- Circular
- Coaxial

Waveguide ports are specified on a single face with the correct shape. To apply a port to a face, the following requirements must be met:

- the face must be flat or a flat face on the boundary of a FEM region
- the face cannot contain any internal edges
- the face must form the boundary of a PEC or dielectric region
- the face cannot have any special material properties (for example, dielectric coating)

- the face cannot be solved with special solution methods (for example, UTD)

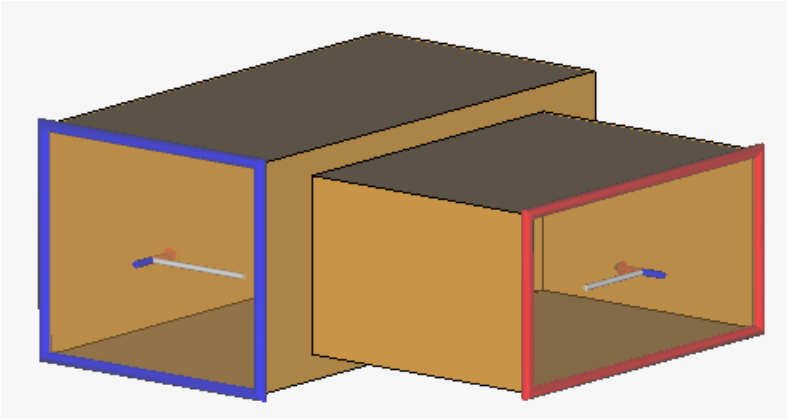


Figure 255: The waveguide port is indicated by a blue border. A waveguide port containing a waveguide source, is indicated by a red border.

Reference Vector

The reference vector specifies the reference direction for a waveguide port.

The reference vector is indicated by a white line connecting the edge of a waveguide with the centre of the waveguide port and shows the direction of *m*, where *m* corresponds to:

- the number of half-wavelengths across the width of the waveguide (rectangular waveguides).
- the number of radial variations (circular waveguides).

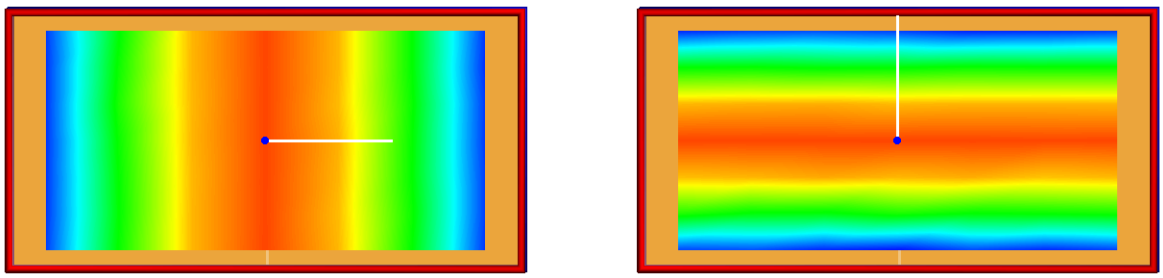


Figure 256: On the left, the reference vector is defined in the direction of the waveguide width. To the right, the reference vector is defined in the direction of the waveguide height.

For a rectangular waveguide, defining the reference vector in one direction, the dominant mode at a frequency might be TE_{10}. Rotating the reference vector with 90° and solving the same problem, the dominant mode will be indicated as TE_{01}.

For a circular waveguide there is no ambiguity with regards to which direction is for *m* or *n*.

> **Note:** Information given for the modes in the `.out` file corresponds to the specified direction of the reference vector.

If the reference vectors differ between ports, it results in a phase mismatch between the S_{21} and S_{11}.

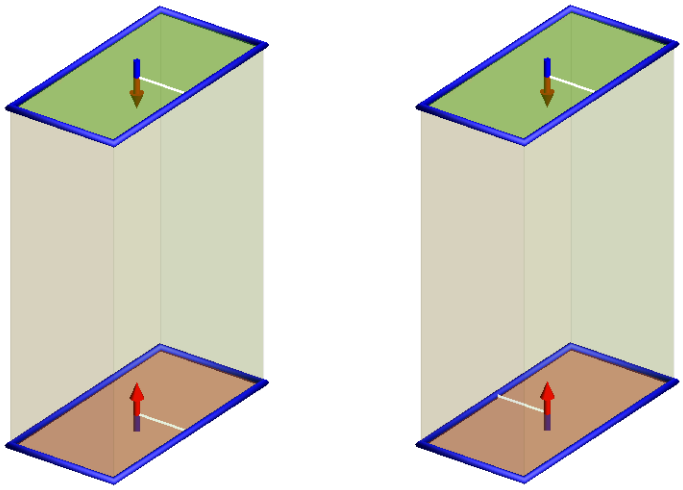


Figure 257: Two waveguide ports with equal reference directions (on the left) and two waveguide ports with opposite reference directions (to the right). For both these cases, the magnitude for S_{11} and S_{21} are identical. For the case where the reference directions differs, the phase for S21 differ by 180° from S_{11}.

Creating a Waveguide Port

Apply a waveguide port to a face.

1. On the **Source/Load** tab, in the **Ports** group, click the **Waveguide Port** icon.

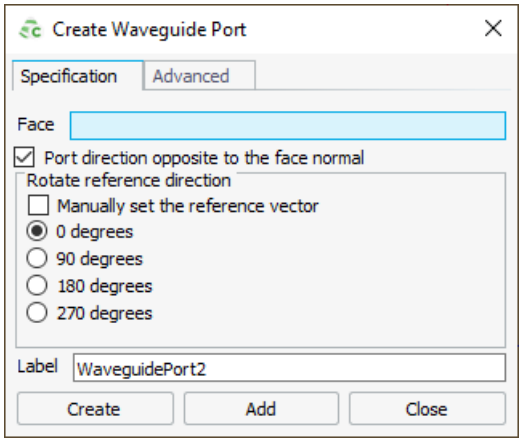


*Figure 258: The **Create Waveguide Port** dialog (**Specification** tab).*

2. Click the **Specification** tab.
3. In the **Face** field, use point entry to specify the face using one of the following workflows:
   - In the 3D view, click on the relevant face.
   - In the details tree, click on the relevant face.

   A port direction and reference direction are automatically defined.

If the port direction is not correct, you can change the direction.

The port direction (indicated by a blue and red arrow in the centre of the port face) indicates the orientation of the port. When a waveguide excitation is applied to the port, impressed/excited modes will be launched in the direction of the red arrowhead. Waveguide modes which propagate toward the port in the opposite direction to the port direction will be absorbed/received at the port face. For a typical application, the port direction should therefore be set such that the red arrow points into the waveguide irrespective of whether the port is used as a source or a sink/load.

4. [Optional] Clear the **Port direction opposite to the face normal** check box to change the port direction to be in the same direction as the face normal.

If the reference vector is not correct, you can specify the reference vector.

5. [Optional] Under **Reference vector**, specify the reference vector.
6. In the **Label** field, add a unique label for the waveguide port.
7. Click the **Advanced** tab.

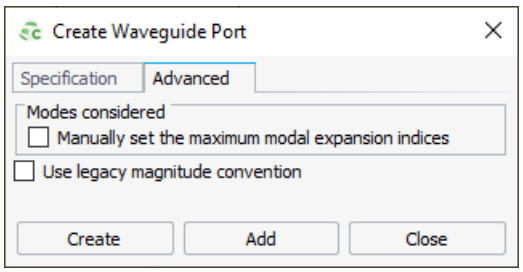


*Figure 259: The **Create Waveguide Port** dialog (**Advanced** tab).*

When the number of modes to be considered is not specified, Feko calculates the number automatically.

8. [Optional] To specify the number of modes, select the **Manually set the maximum modal expansion indices** check box and specify *m* and *n*.
9. Select the **Use legacy magnitude convention** check box to use the legacy definition of mode-dependent units (for example, for TE-mode it is A/m; for TM-mode it is V/m).

> **Note:** The default is to use the power-based definition of magnitude which is common to all mode types.

10. Click **Create** to create the waveguide port and to close the dialog.

Creating a Waveguide Port (Mesh)

Apply a waveguide port to a mesh face.

1. On the **Source/Load** tab, in the **Ports** group, click the **Waveguide Port** icon.

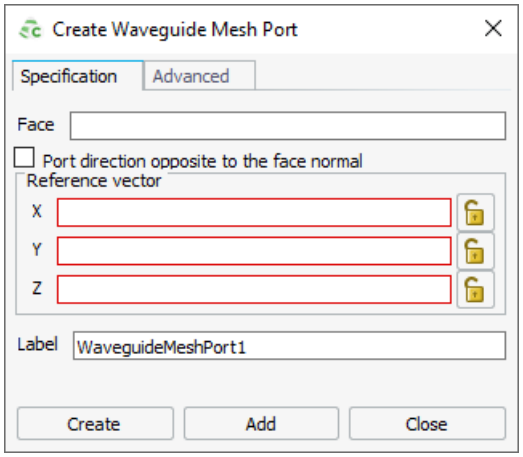


*Figure 260: The **Create Waveguide Mesh Port** dialog.*

2. Click the **Specification** tab.
3. In the **Face** field, use point entry to specify the face using one of the following workflows:
   - In the 3D view, click on the relevant face.
   - In the details tree, click on the relevant face.

   A port direction and reference direction are automatically defined.

If the port direction is not correct, you can change the direction.

The port direction (indicated by a blue and red arrow in the centre of the port face) indicates the orientation of the port. When a waveguide excitation is applied to the port, impressed/excited modes will be launched in the direction of the red arrowhead. Waveguide modes which propagate toward the port in the opposite direction to the port direction will be absorbed/received at the port face. For a typical application, the port direction should therefore be set such that the red arrow points into the waveguide irrespective of whether the port is used as a source or a sink/load.

4. [Optional] Clear the **Port direction opposite to the face normal** check box to change the port direction to be in the same direction as the face normal.

If the reference vector is not correct, you can specify the reference vector.

5. [Optional] Under **Reference vector**, specify the reference vector.
6. In the **Label** field, add a unique label for the waveguide port.
7. Click the **Advanced** tab.

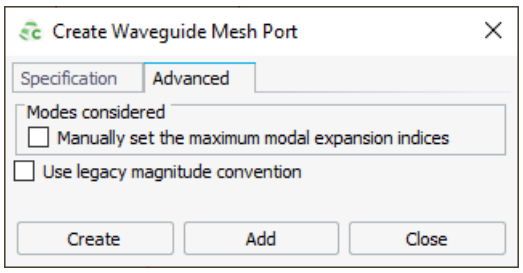


*Figure 261: The **Create Waveguide Mesh Port** dialog (**Advanced** tab).*

When the number of modes to be considered is not specified, Feko calculates the number automatically.

8. [Optional] To specify the number of modes, click the **Manually set the maximum modal expansion indices** check box.
9. Select the **Use legacy magnitude convention** check box to use the legacy definition of mode-dependent units (for example, for TE-mode it is A/m; for TM-mode it is V/m).

   Note: The default is to use the power-based definition of magnitude which is common to all mode types.

10. Click **Create** to create the waveguide port and to close the dialog.

2.24.5 FEM Modal Ports

A finite element method (FEM) modal port is used to apply a port to a flat face on the boundary of a FEM region. A FEM modal port essentially represents an infinitely long guided wave structure (transmission line), connected to a dielectric volume modelled with FEM.

The FEM modal port can be excited with the fundamental mode of the associated guided wave structure, or it can act as a passive port. S-parameters can be computed between the fundamental mode of the FEM modal port and other sources in the model.

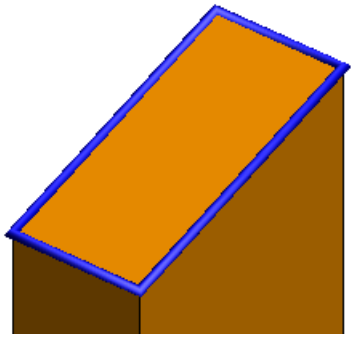


Figure 262: The display of the FEM modal port in the 3D view.

Creating a FEM Modal Port

Apply a FEM modal port to a flat face on the boundary of a FEM region.

1. On the **Source/Load** tab, in the **Ports** group, click the **FEM Modal Port** icon.

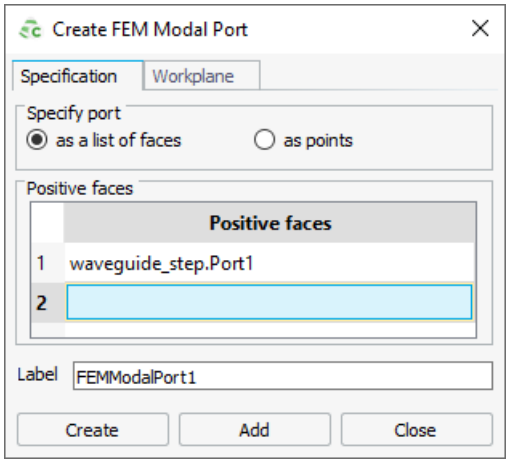


*Figure 263: The **Create FEM Modal Port** dialog.*

2. Specify the port position using one of the following workflows:
   - Specify the faces. Under **Specify port**, click **as a list of faces** and use point-entry to add the faces.
   - Specify the face using three points. Under **Specify port**, click **as points** and specify the three corner points of the rectangular-shaped port.
3. In the **Label** field, add a unique label for the FEM modal port.
4. Click **Create** to create the FEM modal port and to close the dialog.

Creating a FEM Modal Port (Mesh)

Apply a FEM modal port to a flat mesh face on the boundary of a FEM region.

1. On the **Source/Load** tab, in the **Ports** group, click the **FEM Modal Port** icon.

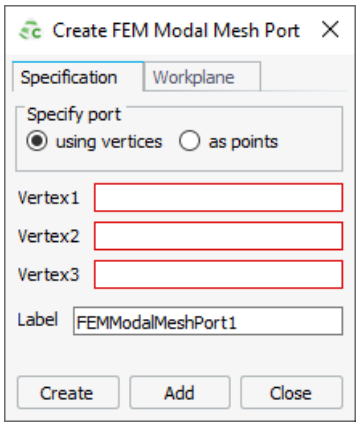


*Figure 264: The **Create FEM Modal Mesh Port** dialog.*

2. Specify the port position using one of the following workflows:

- Specify the vertices. Under **Specify port**, click **using vertices** and use point-entry to add the vertices in the 3D view.
- Specify the points. Under **Specify port**, click **as points** and specify the three corner points of the rectangular-shaped port.

3. In the **Label** field, add a unique label for the FEM modal port.
4. Click **Create** to create the FEM modal port and to close the dialog.

2.24.6 FEM Line Ports

Finite element method (FEM) line ports are used to define the location of impressed current sources and loads in a FEM region.

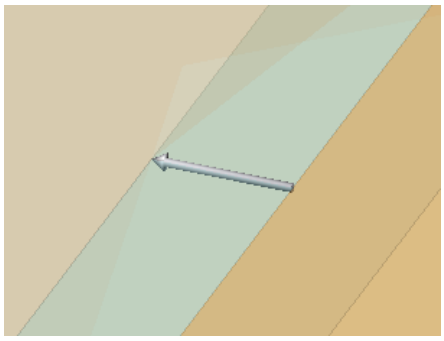


Figure 265: The display of the FEM line port in the 3D view.

Creating a FEM Line Port

Apply a FEM line port to a FEM region when using the finite element method (FEM) solution method.

1. On the **Source/Load** tab, in the **Ports** group, click the **FEM Line Port** icon.

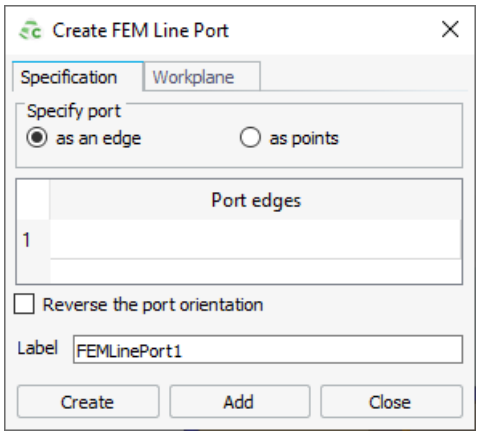


*Figure 266: The **Create FEM Line Port** dialog.*

2. Specify the port position using one of the following workflows:

- The edges (or a connected set of free edges that form a continuous straight line) of the port. Under **Specify port**, click **as an edge**.
- The start point and end point of the FEM line port (in global coordinates). Under **Specify port**, click **as points**.

3. [Optional] If the polarity of the port is to be reversed, select the **Reverse polarity** check box.
4. In the **Label** field, add a unique label for the FEM line port.
5. Click **Create** to create the port and to close the dialog.

Creating a FEM Line Port (Mesh)

Apply a FEM line port between two vertices in a tetrahedral mesh using the finite element method (FEM) solution method.

1. On the **Source/Load** tab, in the **Ports** group, click the **FEM Line Port** icon.

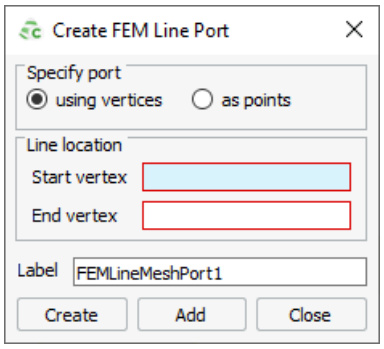


*Figure 267: The **Create FEM Line Port** dialog.*

2. Specify the port position using one of the following workflows:
   - Specify the vertices. Under **Specify port**, click **using vertices**.
   - The start point and end point of the FEM line port (in global coordinates). Under **Specify port**, click **as points**.
3. [Optional] If the polarity of the port is to be reversed, select the **Reverse polarity** check box.
4. In the **Label** field, add a unique label for the FEM line port.
5. Click **Create** to create the line port and to close the dialog.

2.24.7 Cable Ports

Cable ports allow adding sources and loads to cable harnesses.

Apply and define connections to a cable port on the cable harness schematic view.

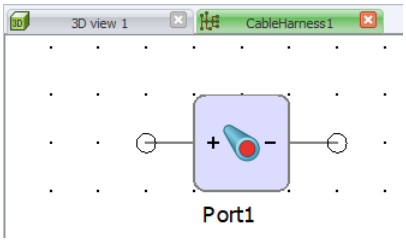


Figure 268: A cable port in the cable schematic view.

Related tasks
Viewing a Cable Harness in the Cable Schematic View

Creating a Cable Port

Apply a cable port to a cable harness.

1. Open the cable schematic view for the cable harness of interest.
2. On the **Source/Load** tab, in the **Ports** group, click the **Cable Port** icon.
   The cable port symbol is added to the active cable schematic view.
3. Connect the port to other schematic elements.
4. From the right-click context menu, select **Properties** to change the label for the cable port.
   The **Modify Cable Port** dialog is displayed.

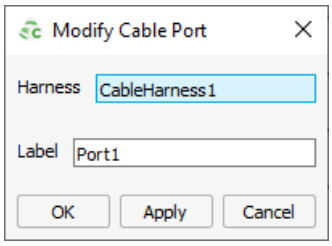


*Figure 269: The **Modify Cable Port** dialog.*

5. In the **Label** field, add a unique label for the cable port.
6. Click **OK** to apply the new label and to close the dialog.

Related tasks
Viewing a Cable Harness in the Cable Schematic View

2.25 Sources

A source is used to excite or illuminate the model and cause current to flow.

> **Note:** Sources can be specified globally or per configuration.

The following sources are supported:

- sources on ports
  - voltage source
  - current source
  - waveguide source
  - FEM modal source
- ideal sources
  - plane wave
  - electric dipole
  - magnetic dipole
  - impressed current
- equivalent sources
  - near field source
  - spherical modes source
  - far field source
  - printed circuit board (PCB) source
  - solution coefficient source

Related concepts
Multiple Configurations

2.25.1 Sources on Ports

Apply a source to a port using either a voltage source, current source, waveguide source or a FEM modal source.

Adding a Voltage Source

Apply a voltage source to any wire, edge, line, network, transmission-line, or cable port.

1. On the **Source/Load** tab, in the **Sources on Ports** group, click the **Voltage Source** icon.

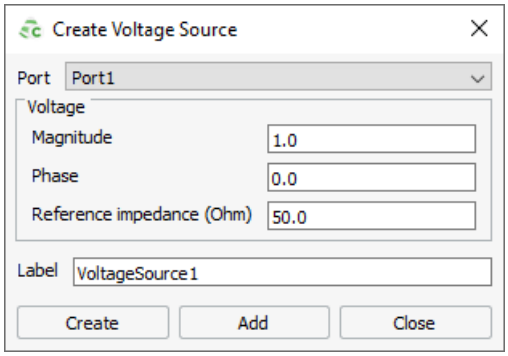


*Figure 270: The **Create Voltage Source** dialog.*

2. In the **Port** field, from the drop-down list, select a port.
3. In the **Magnitude** field, specify the magnitude of the voltage in Volt. The voltage gives the potential difference between the positive side of the port relative to the negative side. A positive voltage results in a positive current flowing out of the positive side and into the negative side of the port.
4. In the **Phase (degrees)** field, specify the phase of the voltage source.
5. In the **Reference impedance** field, specify the impedance of the voltage source.

   Note: The reference impedance is only used when plotting the input reflection coefficient and realised gain in POSTFEKO. If this field is empty, the default value is taken as 50 Ohm.
6. In the **Label** field, add a unique label for the voltage source.
7. Click the **Create** button to create the voltage source and to close the dialog.

Function of Reference Impedance for Voltage and Current Sources

Specify the reference impedance to allow for the plotting of realised gain in POSTFEKO.

When the gain is calculated in Feko, it is calculated according to the IEEE definition where all losses are included, except for mismatch losses.

To view the realised gain in POSTFEKO with mismatch losses included, you need to specify the reference impedance.

Note: Specifying the reference impedance does not affect the gain, source power or radiated power.

Adding a Current Source

Apply a current source to a line port in a dielectric region solved with the finite element method (FEM) to realise an impressed current source.

> **Note:** An intrinsic limitation of the impressed current source is that no radius is considered. The field is singular in the vicinity of the filament affecting the accuracy of the computed input impedance of the source.

1. On the **Source/Load** tab, in the **Sources on Ports** group, click the **Current Source** icon.

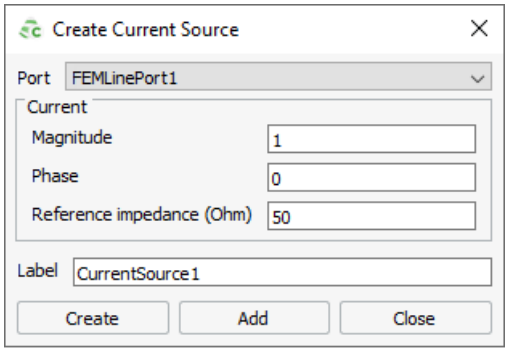


*Figure 271: The **Create Current Source** dialog.*

2. In the **Port** field, from the drop-down list, select a port.
3. In the **Magnitude (A)** field, specify the magnitude of the current in Ampere.
4. In the **Phase (degrees)** field, specify the phase of the current source.
5. In the **Reference impedance** field, specify the impedance of the current source.

> **Note:** The reference impedance is used when calculating the input reflection coefficient and realised gain. If this field is empty, default value is taken as 50 Ohm.

6. In the **Label** field, add a unique label for the current source.
7. Click the **Create** button to create the current source and to close the dialog.

Adding a Waveguide Source

Apply a waveguide source to a waveguide port.

1. On the **Source/Load** tab, in the **Sources on Ports** group, click the **Waveguide Source** icon.

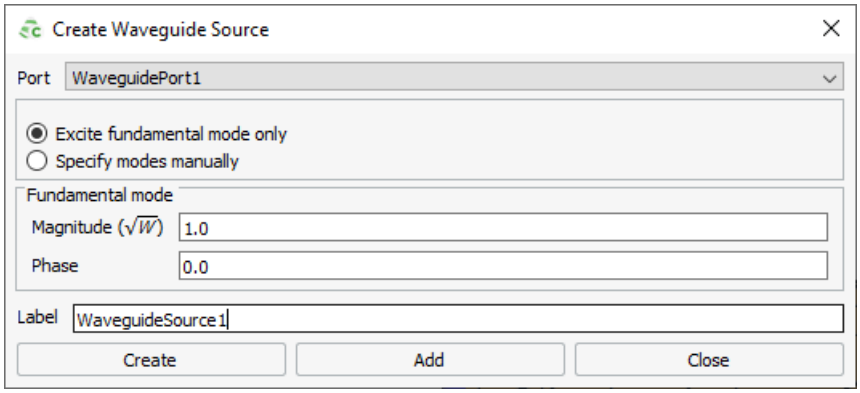


*Figure 272: The **Create Waveguide Source** dialog.*

2. In the **Label** field, add a unique label for the waveguide source.
3. In the **Port** field, from the drop-down list, select any waveguide port.
4. Select one of the following regarding the mode(s) to excite:
   - To excite only the fundamental waveguide mode, select **Excite fundamental mode only**. When this option is selected, the mode type and its indices cannot be specified since they are determined automatically.
   - To manually specify the modes using their mode indices, select **Specify modes manually**.
5. In the **Magnitude** field, specify the magnitude of the mode.
6. In the **Phase** field, specify the phase of the mode.
7. Click the **Create** button to create the waveguide source and to close the dialog.

Adding a FEM Modal Source

Apply a FEM modal source to a finite element method (FEM) modal port.

Note: A FEM modal source excites the associated long-guided wave structure of the FEM modal port with the fundamental mode.

Important: When no source is defined, the modal port acts as a passive port (sink) for fields incident on the port.

1. On the **Source/Load** tab, in the **Sources on Ports** group, click the **FEM Modal Source** icon.

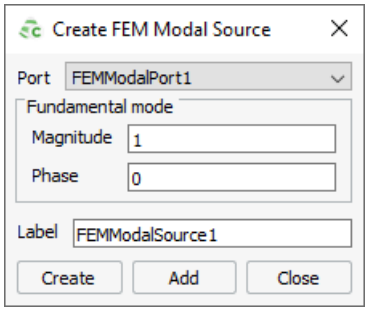


*Figure 273: The **Create FEM Modal Source** dialog.*

2. In the **Port** field, from the drop-down list, select any FEM modal port.
3. In the **Magnitude** field, specify the magnitude of the fundamental mode.
4. In the **Phase** field, specify the phase of the fundamental mode.
5. In the **Label** field, add a unique label for the FEM modal source.
6. Click the **Create** button to create the FEM modal source and to close the dialog.

2.25.2 Ideal Sources

An "ideal" source is a source that applies a field, voltage or current and has no internal impedance.

Adding a Plane Wave Source

Add a plane wave source to illuminate a model with a uniform electric field.

1. On the **Source/Load** tab, in the **Ideal Source** group, click the **Plane Wave Source** icon.

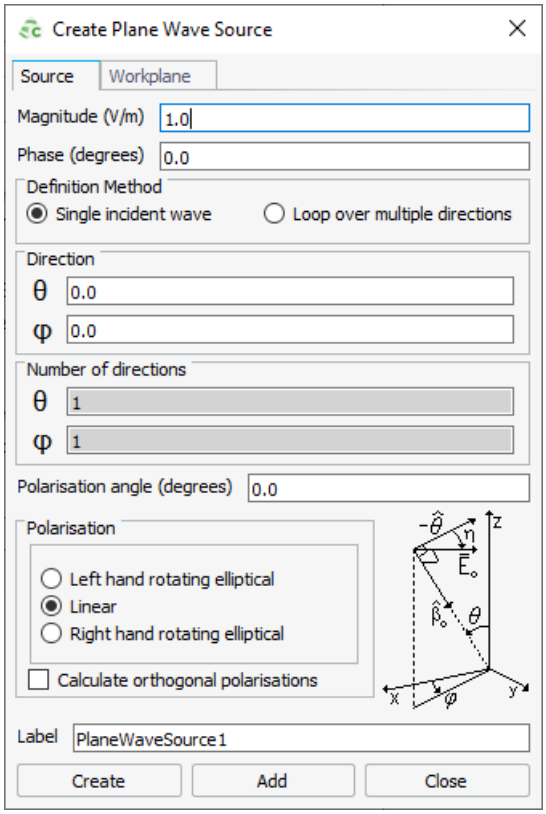


*Figure 274: The **Create Plane Wave Source** dialog.*

2. In the **Magnitude (V/m)** field, specify the magnitude of the plane wave.
3. In the **Phase (degrees)** field, specify the phase of the plane wave.
4. Specify the operation mode using one of the following:
   - To create a single plane wave, click **Single incident wave**.

     Tip: Use multiple single incident plane wave sources to create a specific field distribution.

   - To create a single plane wave that loops over multiple directions, select **Loop over multiple directions**.
5. In the **Polarisation angle** field, specify the angle, η in degrees, measured in a right-handed sense around the direction of propagation, from ϑ to $\mathbf{E}_0$.
6. Under **Polarisation**, specify the polarisation type:
   - Left hand rotating elliptical
   - Linear
   - Right hand rotating elliptical
7. [Optional] Select the **Calculate orthogonal polarisations** check box to create an additional orthogonal plane wave (although still a single plane wave source).

> **Note:** Select this option when exporting transmission / reflection coefficients to a `.tr` file.

8. In the **Ellipticity (0 to 1)** field, specify the polarisation.

> **Note:**
> - Ellipticity = 0: linear polarisation
> - Ellipticity ≤ 1: elliptical / circular polarisation

9. In the **Label** field, add a unique label for the plane wave source.
10. Click the **Create** button to create the plane wave source and to close the dialog.

Related tasks
Exporting Transmission / Reflection Coefficients to a `.TR` File

Adding an Electric Dipole Source

Apply an electric dipole source that represents an elementary dipole element with a specified orientation, magnitude and phase.

1. On the **Source/Load** tab, in the **Ideal Source** group, click the **Electric Dipole Source** icon.

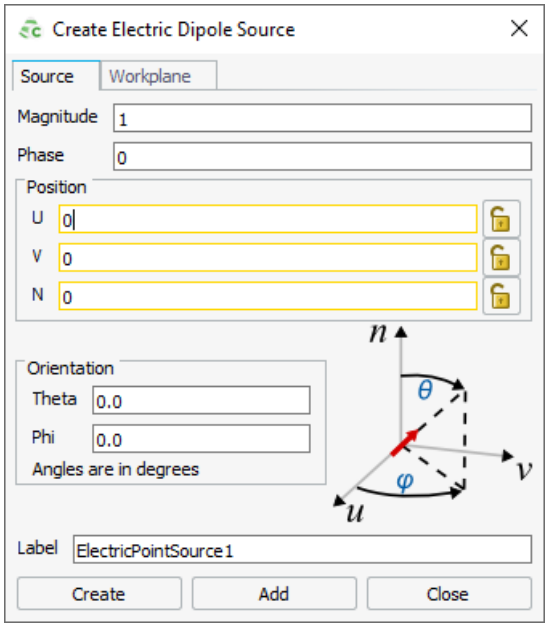


*Figure 275: The **Create Electric Dipole Source** dialog.*

2. In the **Magnitude of Idl (Am)** field, specify the magnitude of the current.
3. In the **Phase (degrees)** field, specify the phase of the current.
4. In the **Position** field, specify where the source is to be placed.
5. In the **Orientation** field, specify the orientation of the source.

6. In the **Label** field, add a unique label for the electric dipole source.
7. Click the **Create** button to create the electric dipole source and to close the dialog.

Adding a Magnetic Dipole Source

Apply a magnetic dipole that can be either an electric ring current or a magnetic line current that represents an elementary dipole element with a specified orientation, magnitude and phase.

1. On the **Source/Load** tab, in the **Ideal Source** group, click the **Magnetic Dipole Source** icon.

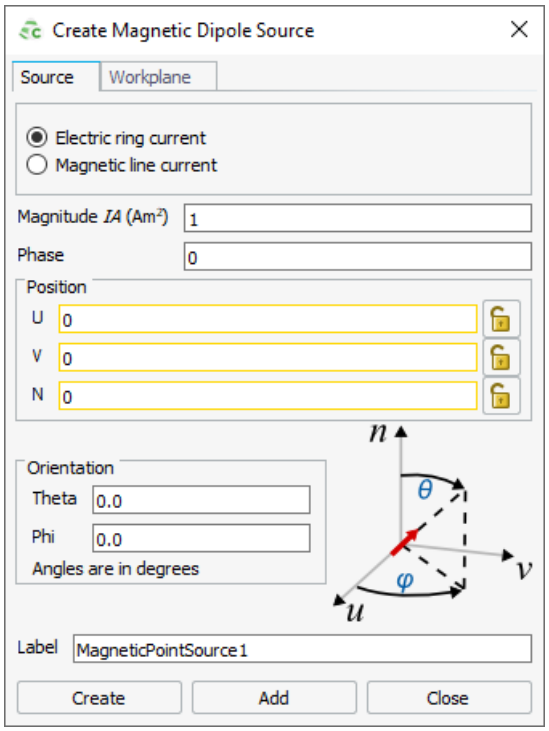


*Figure 276: The **Create Magnetic Dipole Source** dialog.*

2. Specify the magnitude of the source using one of the following:
   - To specify the magnitude as the product of the loop current and loop area, click **Electric ring current**.
     - In the **Magnitude of IA (Am^2)** field, specify the current in Am^2.
   - To specify the magnitude as the product of the dipole length and magnetic current, click **Magnetic line current**.
     - In the **Magnitude of Iml (Vm)** field, specify the voltage in Vm.
3. In the **Phase (degrees)** field, specify the phase of the current.
4. In the **Position** field, specify where the source is to be placed.
5. In the **Orientation** field, specify the orientation of the source.
6. In the **Label** field, add a unique label for the magnetic dipole source.

7. Click the **Create** button to create the magnetic dipole source and to close the dialog.

Adding an Impressed Current Source

Apply an impressed current source to represent a lightning strike and exit points.

> **Tip:** For lightning strikes, use two impressed current sources to model the strike point and exit point where the current flows off the structure. The source should either have a current magnitude of -1 or a phase of 180°.

1. On the **Source/Load** tab, in the **Ideal Source** group, click the **Impressed Current Source** icon.

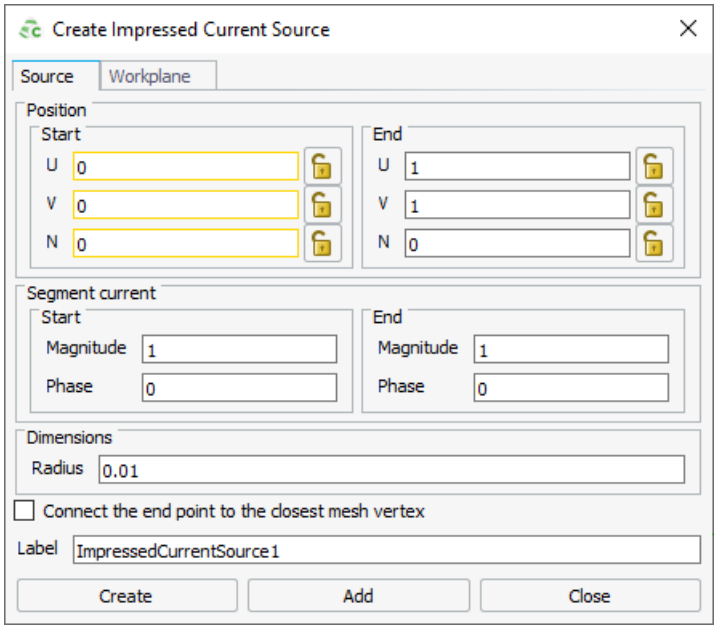


*Figure 277: The **Create Impressed Current Source** dialog.*

2. Under **Position**, specify the start point and end point of the current segment.
3. Under **Segment current**, specify the magnitude and phase of the segment current at the start point and end point.
4. In the **Radius** field, specify the radius of the current segment.
5. [Optional] Select the **Connect the endpoint to the closest mesh vertex** check box to terminate the current at the closest mesh vertex during the solution.
   - To connect to the closest triangle vertex, select **On triangle**.
   - To connect to the closest segment vertex, select **On segment**.

   > **CAUTION:** Visually confirm in POSTFEKO that the source connects at the required vertex.

6. In the **Label** field, add a unique label for the impressed current source.

7. Click the **Create** button to create the impressed current source and to close the dialog.

2.25.3 Equivalent Sources

An "equivalent" source is a numerically equivalent (simulated or measured) of a complex source. Significant reductions in computational requirements is achieved when solving a complex problem through model decomposition and using an equivalent source.

Adding a Near Field Source

Apply an array of electric and magnetic dipoles in the model (in the form of a planar, cylindrical or spherical aperture) that is equivalent to measured or calculated field values.

1. On the **Source/Load** tab, in the **Equivalent Sources** group, click the **Near Field Source** icon.

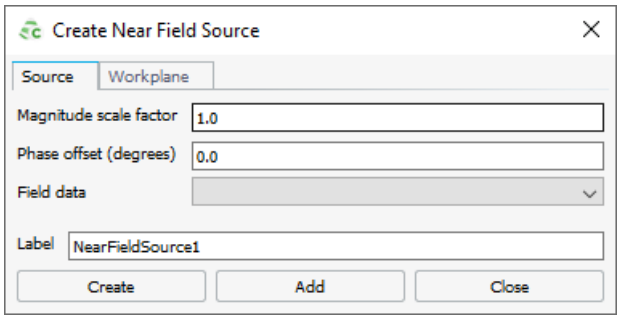


*Figure 278: The **Create Near Field Source** dialog.*

2. In the **Magnitude scale factor** field, specify the scaling factor.

> **Tip:** Use the scaling factor when data files have different units (for example, µV/m).

3. In the **Phase offset (degrees)** field, specify the phase (in degrees) to be added to the phase of the fields.
4. In the **Field data** field, specify the field data to be used to define the near field source.
5. In the **Label** field, add a unique label for the near field source.
6. Click the **Create** button to create the near field source and to close the dialog.

Adding a Spherical Mode Source

Apply an impressed spherical mode source based on pre-calculated spherical modes. The spherical modes are either radiating to infinity or incident onto a structure (converging on the coordinate system origin).

This source can be used for the synthesis of an arbitrary electromagnetic field as well as determining the response of a receiving antenna due to the incident modes.

1. On the **Source/Load** tab, in the **Equivalent Sources** group, click the **Spherical Mode Source** icon.

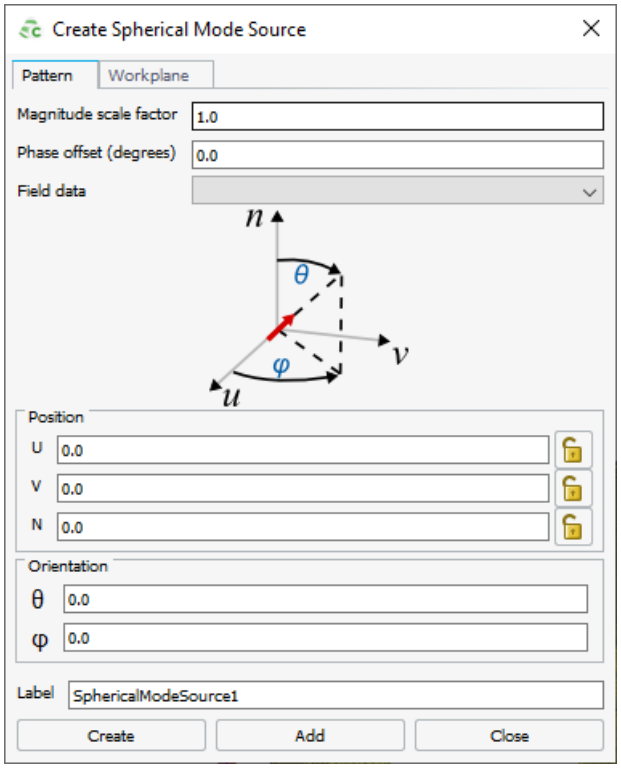


*Figure 279: The **Create Spherical Mode Source** dialog.*

2. In the **Magnitude scale factor** field, specify the scaling factor.
3. In the **Phase offset (degrees)** field, specify the phase (in degrees) to be added to the phase of the fields.
4. In the **Field data** field, specify the field data to be used to define the spherical modes source.
5. In the **Position** field, specify where the source is to be placed.
6. In the **Orientation** field, specify the orientation of the source.
7. In the **Label** field, add a unique label for the spherical modes source.
8. Click the **Create** button to create the spherical modes source and to close the dialog.

Adding a Far Field Source

Apply a radiation pattern of an antenna and use as an impressed source at a specified point in space.

1. On the **Source/Load** tab, in the **Equivalent Sources** group, click the **Far Field Source** icon.

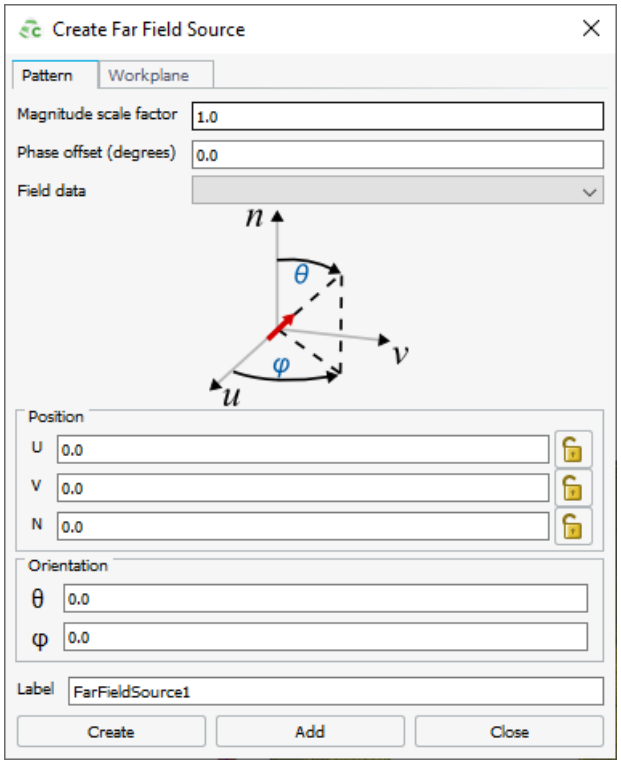


*Figure 280: The **Create Far Field Source** dialog.*

2. In the **Magnitude scale factor** field, specify the scaling factor.
3. In the **Phase offset (degrees)** field, specify the phase (in degrees) to be added to the phase of the fields.
4. In the **Field data** field, specify the field data to be used to define the far field source. The field data must be a far field specified using the spherical coordinate system.
5. In the **Position** field, specify where the source is to be placed.
6. In the **Orientation** field, specify the orientation of the source.
7. In the **Label** field, add a unique label for the far field point source.
8. Click the **Create** button to create the far field point source and to close the dialog.

Adding a PCB Source

Apply impressed line currents in the model to represent a printed circuit board (PCB). The impressed line currents are equivalent to the current values calculated for the traces and vias of a PCB.

1. On the **Source/Load** tab, in the **Equivalent Sources** group, click the **PCB Source** icon.

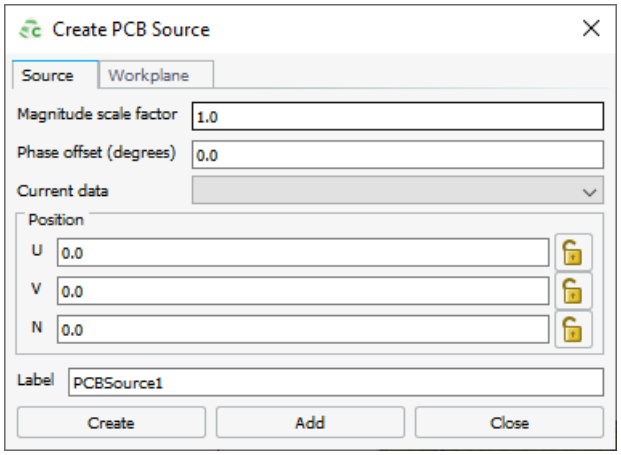


*Figure 281: The **Create PCB source** dialog.*

A preview of the PCB outline is displayed in green in the 3D view.

2. In the **Magnitude scale factor** field, specify the scaling factor.
3. In the **Phase offset (degrees)** field, specify the phase (in degrees) to be added to the phase of the currents.
4. In the **Current data** field, specify the PCB current data to be used to define the PCB source.
5. In the **Position** field, specify where the source is to be placed.
6. In the **Label** field, add a unique label for the PCB source.
7. Click the **Create** button to create the PCB source and to close the dialog.
   The PCB outline is displayed in the 3D view when selecting the **Configuration** tab in the model tree.

Adding a Solution Coefficient Source

Apply a solution coefficient data definition and use as an impressed current source at a specified point in space.

1. On the **Source/Load** tab, in the **Equivalent Sources** group, click the **Solution Coefficient** icon.

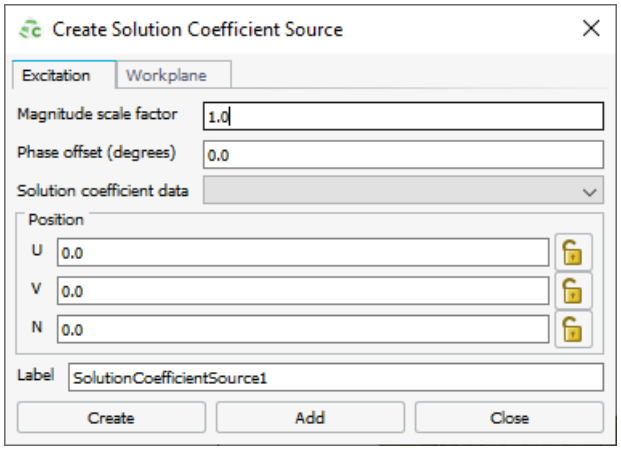


*Figure 282: The **Create Solution Coefficient Source** dialog.*

2. In the **Magnitude scale factor** field, specify the scaling factor.
3. In the **Phase offset (degrees)** field, specify the phase (in degrees) to be added to the phase of the currents.
4. In the **Solution coefficient data** field, specify the solution coefficient data to be used to define the solution coefficient source.
5. In the **Position** field, specify where the source is to be placed.
6. In the **Label** field, add a unique label for the solution coefficient data source.
7. Click the **Create** button to create the solution coefficient source and to close the dialog.

Related tasks

Defining Solution Coefficient Data from File

Requesting Model Decomposition

2.26 Loads and Non-Radiating Networks

Complex feed networks can be simplified by including them as a circuit representation using general network blocks.

Note: Loads can be specified globally or per configuration.

Non-radiating networks operate on the principle of connecting networks or "black boxes" using a connection diagram. The benefit of this scheme is that only a single connection is drawn between black boxes. However, it may seem less intuitive when working with, for example, S-parameters or Z-parameters, as graphical representations of these always show a signal pin and a ground pin.

Non-radiating networks are connected to, for example, wire ports in the geometry and these networks are implicitly seen as being in series with the wire segment.

Related concepts
Multiple Configurations

2.26.1 Adding a Load

Apply an impedance load to a wire port, microstrip port, FEM line port, a general network or an ideal transmission line.

1. On the **Source/Load** tab, in the **Loads/Networks** group, click the **Add Load** icon.

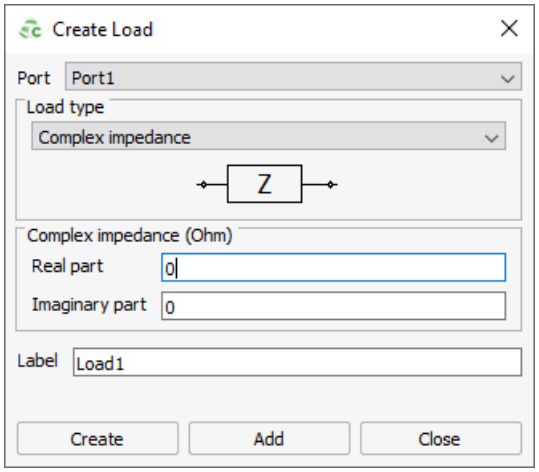


*Figure 283: The **Create Load** dialog.*

2. From the **Port** drop-down list, select a port.
3. Under the **Load type** drop-down list, specify one of the following:
   - **Complex impedance**
     - In the **Real part** field, specify the real part of the complex impedance in Ohm.

- In the **Imaginary part** field, specify the imaginary part of the complex impedance in Ohm.

- **Series circuit**
  - To specify a resistor, select the **Resistor** check box and in the **Resistor (Ohm)** field, enter a value in Ohm.
  - To specify an inductor, select the **Inductor** check box and in the **Inductor (H)** field, enter a value in Henry.
  - To specify a capacitor, select the **Capacitor** check box and in the **Capacitor (F)** field, enter a value in Farad.

  If no option is selected, the result is a short circuit.
- **Parallel circuit**
  - To specify a resistor, select the **Resistor** check box and in the **Resistor (Ohm)** field, enter a value in Ohm.
  - To specify an inductor, select the **Inductor** check box and in the **Inductor (H)** field, enter a value in Henry.
  - To specify a capacitor, select the **Capacitor** check box and in the **Capacitor (F)** field, enter a value in Farad.

  If no option is selected, the result is an open circuit.
- **SPICE circuit**
  - In the **Filename** field, browse for a one-port SPICE circuit file (`.cir`) to define a load between two pins.

    Note:
    - supports only a subset of Berkeley SPICE3f5 syntax.
    - Only linear circuits are supported.

  - Specify a subcircuit defined in the `.cir` file.
    - Clear the **Auto** check box.
    - In the **Circuit name** field, specify the subcircuit name.

    Note: **Circuit name** must correspond to the subcircuit name in the file.
- **Touchstone file**
  - In the **Filename** field, browse for a one-port Touchstone file (`.s1p`, `.z1p`, `.y1p`).

    Note: If the load is added to a port that has a voltage source, the load is placed in series with the voltage source.

4. In the **Label** field, add a unique label for the impedance load.
5. Click the **Create** button to create the load and to close the dialog.

Load Configuration for Networks

When a network port is connected to a wire (segment/vertex) or edge port, a load can be defined in series or across the terminals of a network.

Note: A network port refers to a port connected to a general network or transmission line.

Load Placed in Series with Network

To place a load in series with the network, define the load on the wire (segment/vertex) or edge port.

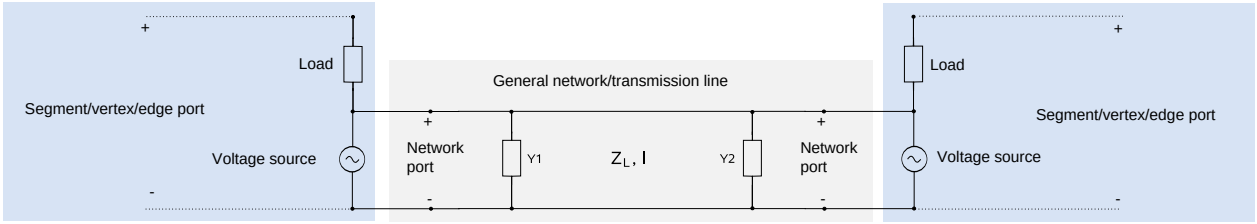


Figure 284: The load is placed in series with the network.

Load Placed Across Network

To place the load across the network, define the load on the network port.

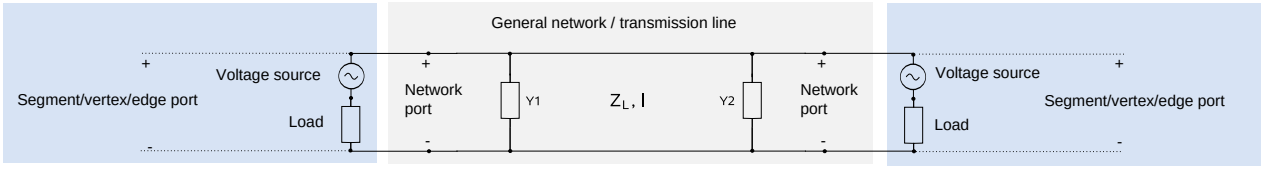


Figure 285: The load is placed across the network.

Discrete Loads

Load a wire port, microstrip port, FEM line port, general networks and a transmission line with a discrete load such as a complex impedance, series circuit or parallel circuit.

Note: If multiple loads or sources are applied to the same port, they are placed in series.

Complex Impedance

A frequency-independent load consisting of a constant real and imaginary part. This load type can be applied to wire, edge, network or transmission-line ports.

Series Circuit

A frequency-dependent load consisting of a series-connected resistor (R), capacitor (C) and inductor (L). This load can only be applied to a wire port and an edge port. The load impedance is given by

$$Z_s = R + j\omega L + \frac{1}{j\omega C} \quad (12)$$

Parallel Circuit

A frequency-dependent load consisting of a resistor (R), a capacitor (C) and inductor (L) connected in parallel. This load can only be applied to a wire port and an edge port. The load impedance is given by

$$Z_p = \frac{1}{\frac{1}{R} + \frac{1}{j\omega L} + j\omega C} \quad (13)$$

where the resistance or inductance is taken as infinite when set to 0 (it does not contribute to the impedance).

Note: For the parallel circuit the circuit elements are connected in parallel **inside the circuit**, but the circuit itself is connected **in series** with the source.

Waveguide and Modal Port Sinks

When no source is defined for a waveguide port or a FEM modal port, the port acts as a passive port (sink) for fields incident on the port.

2.26.2 Network Schematic View

The network schematic view is a panel that shows all general networks, transmission lines and ports (wire and edge ports) in the model. Use this view to connect general networks, transmission lines, ports and loads.

On the **Home** tab, in the **Create View** group, click the **Schematic** icon. From the drop-down list select **Network Schematic** icon.

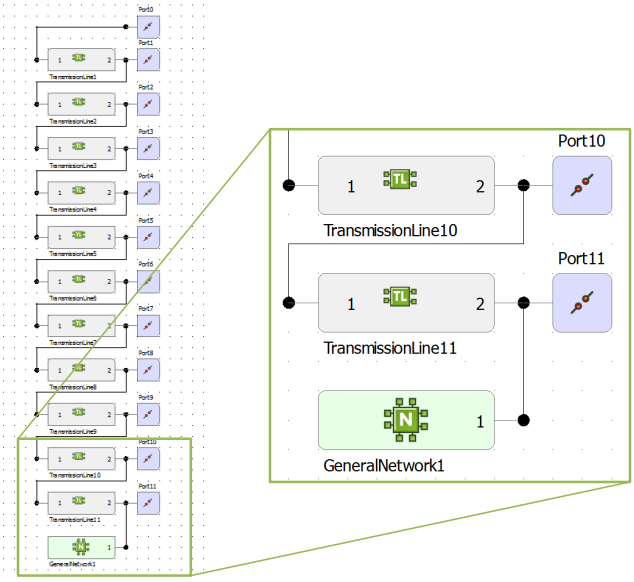


Figure 286: An example showing the network schematic view with connections between transmission lines, general networks and ports.

Connect two elements by clicking on the connector point (indicated by a white dot) and dragging the connection until the mouse cursor is over the desired second connection point. When clicking on the second point, a connection between wires is indicated by a black dot.

Selected networks, transmission lines, ports and connections are indicated by a dotted outline.

Delete an element by selecting the respective element and pressing Delete.

2.26.3 Adding a General Network (Data Matrix)

Define a general non-radiating network using network parameter matrices. In the network schematic view, interconnect the networks (cascading) and excite or load the network ports.

1. On the **Source/Load** tab, in the **Loads/Networks** group, click the **Network** icon.

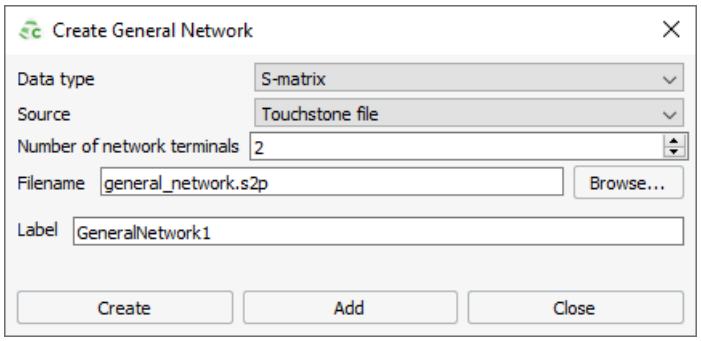


*Figure 287: The **Create General Network** dialog.*

2. From the **Data type** drop-down list, select one of the following network parameters:

- **S-matrix**
- **Z-matrix**
- **Y-matrix**

3. From the **Source** drop-down list, select one of the following:
   - To import the network parameters from file, select **Touchstone file**.

     Note: Only Touchstone format v1.1 is supported.

   - To specify the coupling parameters, select **Specify network manually**.
     - Specify the coupling parameters and the reference impedance.
4. In the **Number of network terminals** field, specify the number of network terminals.
5. In the **Label** field, add a unique label for the general network.
6. Click the **Create** button to create the general network and to close the dialog.

2.26.4 Adding a General Network (SPICE)

Define a general non-radiating network by importing a direct component-based network from a SPICE `.cir` file. In the network schematic view, interconnect the networks (cascading) and excite or load the network ports.

Note:
- Feko supports only a subset of Berkeley SPICE3f5 syntax.
- Only linear circuits are supported.

1. On the **Source/Load** tab, in the **Loads/Networks** group, click the **Network** icon.

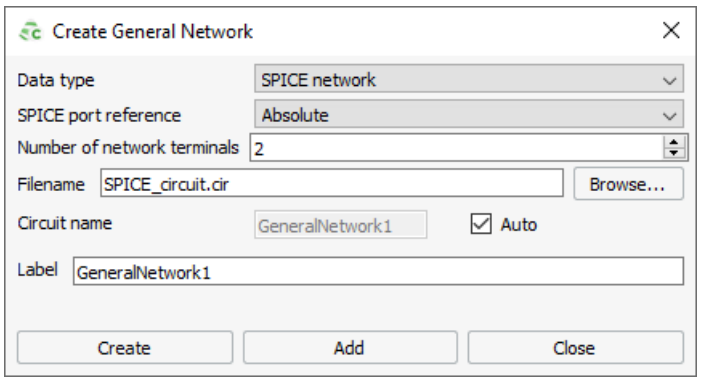


*Figure 288: The **Create General Network** dialog.*

2. From the **Data type** drop-down list, select **SPICE network**.
3. From the **SPICE port reference** drop-down list, select one of the following:
   - To use a SPICE file with an absolute port reference, select **Absolute**.
   - To use a SPICE file with a relative port reference, select **Relative**.

4. In the **Number of network terminals** field, specify the number of network terminals.

   Note:
   - The number of network terminals must correspond to the number of ports in the `.cir` file for an absolute port reference.
   - The number of network terminals must be half the number of ports in the `.cir` file for a relative port reference.

5. In the **Filename** field, browse to the location of the `.cir` file.
6. In the **Circuit name** field, specify the subcircuit network name in the `.cir` file.

   Note: **Circuit name** must correspond to the subcircuit name in the `.cir` file.

7. In the **Label** field, add a unique label for the general network.
8. Click the **Create** button to create the general network and to close the dialog.

Related concepts
SPICE3f5

Setting Up Different Port References for a SPICE Network

Define SPICE files with an absolute or relative port reference for a SPICE network.

Example of a SPICE file with an absolute port reference

A SPICE example of pi-network using an absolute port reference (each port is referenced to the SPICE global ground n0). The total number of output nodes is equal to the amount of network terminals when creating a general network.

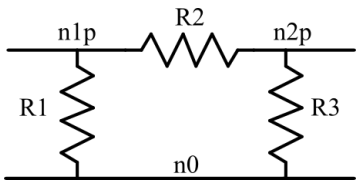


Figure 289: Example of a circuit using an absolute reference (node 0).

```
* Subcircuit PI Network
.SUBCKT PINETWORK n1 n2
R1 n1 0 290
R2 n1 n2 18
R3 n2 0 290
.ENDS   PINETWORK

.SUBCKT NWN1 n1 n2 (absolute port reference)
X1 n1 n2 PINETWORK
.ENDS NWN1
.END
```

Example of a SPICE file using an relative port reference

A SPICE example of a pi-network using a relative port reference, exposing the negative and positive pins of each terminal in the network. The total number of output nodes is double the amount of network terminals.

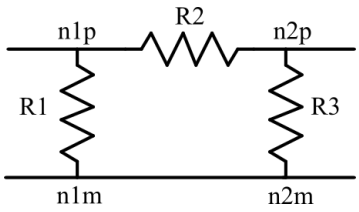


Figure 290: Example of a circuit using an relative reference.

```
* Subcircuit PI Network
.SUBCKT PINETWORK n1 n2 n3
R1 n1 n3 290
R2 n1 n2 18
R3 n2 n3 290
.ENDS   PINETWORK

.SUBCKT NWN1 n1p n1m n2p n2m (relative port reference)
X1 n1p n2p n1m PINETWORK
R1 n1m n2m 0
.ENDS NWN1
.END
```

2.26.5 Adding a Transmission Line

Define an ideal, non-radiating transmission line. In the network schematic view, connect the transmission line to a port, other transmission lines or general networks.

1. On the **Source/Load** tab, in the **Loads/Networks** group, click the **TX Line** icon.

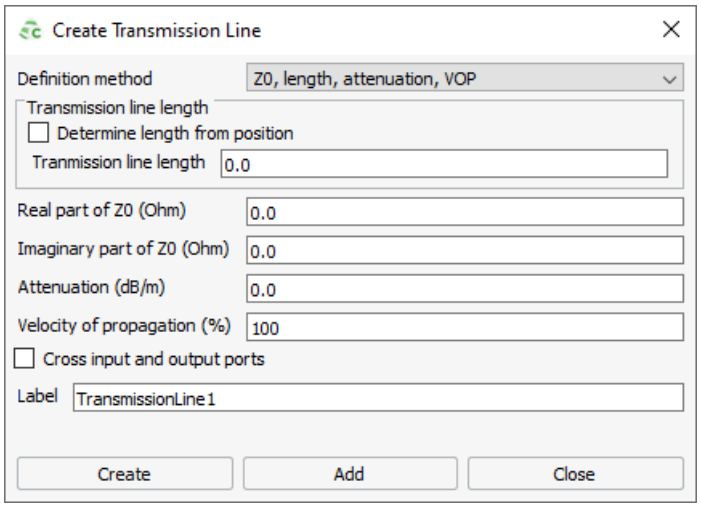


*Figure 291: The **Create Transmission Line** dialog.*

2. From the **Definition method** drop-down list, specify one of the following:
   - **Z0, length, attenuation, VOP**
   - **Z0, length, attenuation**
   - **Z0, length, medium**
3. Specify the transmission line length using one of the following:
   - To determine the distance between the start point and end point of the transmission line automatically, select the **Determine length from position** check box.
   - To specify the transmission line length, in the **Transmission line length** field, enter the length.
4. In the **Real part of Z0 (Ohm)** field, specify the real value.
5. In the **Imaginary part of Z0 (Ohm)** field, specify the imaginary value.
6. Specify one of the following, depending on the selection in 2.
   - *Attenuation (dB/m)*
     Losses of the transmission line in dB/m.

     Note: The propagation constant is taken as the propagation constant of the medium in which the start and end ports are located. As a result, the attenuation specified is added to any losses of this medium.

     Velocity of propagation
     The propagation speed through the transmission line relative to the speed of light.

     Medium
     The medium used as the background medium for the transmission line.

The positive port voltage is in the direction of the connected segment (from the start to the end point of the segment). As a result, the input and output ports of the transmission line have unique orientations.

7. [Optional] Select the **Cross input and output ports** check box to cross the input and output ports.
8. In the **Label** field, add a unique label for the transmission line.
9. Click the **Create** button to create the transmission line and to close the dialog.

- View the transmission line definition in the model tree (Configuration tab), under **Non-radiating networks**.
- The transmission line is added to the network schematic view where you can connect the transmission line to a port, other transmission lines or general networks.

2.27 Multiple Configurations

Perform multiple solutions for a single model using multiple configurations. Multiple configurations remove the requirement to create multiple models with different solution requests.

For example:

- Calculate the input impedance of an antenna over a frequency range (*Configuration 1*) and the radiated far field at the centre frequency (*Configuration 2*).
- Calculate the antenna parameters of a dual-band antenna over two frequency bands (*Configuration 1* and *Configuration 2*).
- Calculate the current on a wire above ground in three configurations:
  - Terminated in an open circuit (*Configuration 1*).
  - Terminated in short circuit (*Configuration 2*).
  - Terminated with a matched load or system impedance (*Configuration 3*).
- Calculate the two-port S-parameters of a system (*Configuration 1*) and the radiated currents when both ports are active at the same time (*Configuration 2*).

2.27.1 Configuration Types

Three configuration types are supported: standard configurations, S-parameter configurations and characteristic mode configurations.

Standard Configuration

A standard configuration is the default configuration type.

The following requests are supported in conjunction with a standard configuration:

- far fields
- near fields
- currents
- specific absorption rate (SAR)
- transmission / reflection coefficients
- cable harness
- receiving antennas
- error estimations
- model decomposition

Adding a Standard Configuration

Define a standard configuration and add it to the model.

Add a standard configuration using one of the following workflows:

- On the **Request** tab, in the **Configurations** group, click the **Standard Configuration** icon.
- In the configuration list, click the . From the right-click context menu, select **Standard Configuration**.

S-Parameter Configuration

Add an S-parameter configuration to calculate S-parameters between an arbitrary number of ports.

The following requests are supported in conjunction with a S-parameter configuration:

- far fields
- near fields
- currents
- specific absorption rate (SAR)
- cable probe
- receiving antennas
- error estimations
- model decomposition

Note:

- Only a single S-parameter request is allowed per configuration.
- No configuration-specific (local) sources or power settings are allowed.

Adding an S-Parameter Configuration

Define an S-parameter configuration and add it to the model.

1. Add an S-parameter configuration using one of the following workflows:
   - On the **Request** tab, in the **Configurations** group, click the **S-parameter Configuration** icon.
   - In the configuration list, click the icon. Select **S-parameter Configuration** from the menu .

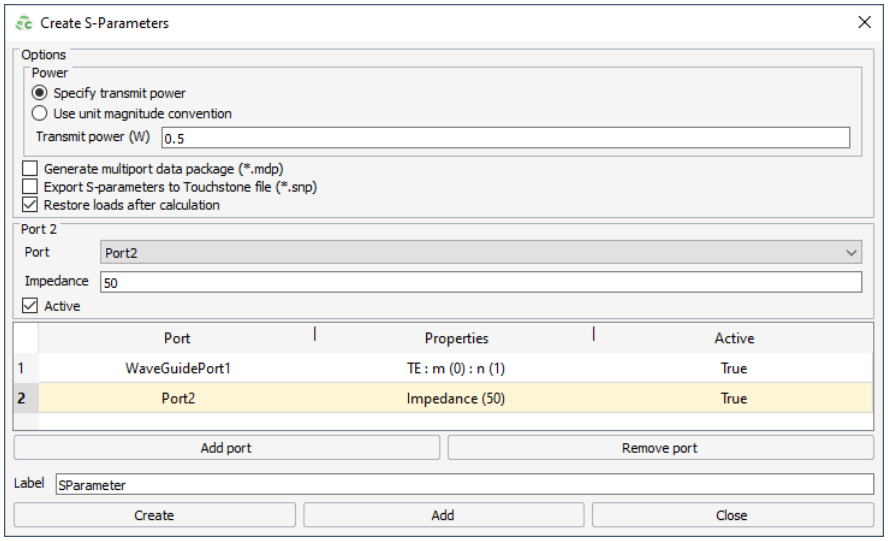


*Figure 292: The **Create S-Parameters** dialog.*

2. In the **Power** group select one of the following:
   - Select **Specify transmit power** to specify the transmit power for all the ports when calculating the requests associated with the S-parameter request.
   - Select **Use unit magnitude convention** to use the unit magnitude convention to calculate the requests associated with the S-parameter request.

   > **Note:** The S-parameter matrix result is not affected when selecting the power option, only the fields or current requests associated with the S-parameter request is altered based on the power information.

3. [Optional] In the **Options** group, select the **Export S-parameters to Touchstone** file check box to export the S-parameters to a `.snp` file.

   For each S-parameter configuration, a separate Touchstone file is created. The file name is in the form `<FEKO_base_filename>_<requestname>(k).snp` where:

   FEKO_base_filename
   : file name of the model

   requestname
   : request name

   n
   : number of ports

   k
   : a counter (integer) to distinguish between the results of multiple requests with the same name and the same number of ports.

> **Note:**
> Feko does not normalise the S-parameter values to a global reference impedance when exporting the S-parameters to a Touchstone file. The values are referenced to the impedance specified on each port.

> **CAUTION:** Some industry tools that use the Touchstone format often assume that all values are referenced to a common impedance. When exporting S-parameters for use in an industry tool that supports only a single reference impedance, specify the reference impedance for each port to ensure the correct interpretation.

4. [Optional] In the **Options** group, select the **Restore loads after calculation** check box to remove the loads once the S-parameter calculation is complete.
5. In the **Port** column, from the drop-down list, select the port.
6. In the **Properties** column, specify the following:
   - For waveguide ports, specify the type (TE[33] / TM[34]/ TEM[35]), indices and rotation of the mode.
   - For ports other than waveguide and FEM modal ports, specify the reference impedance. If no impedance is specified, a default reference impedance of 50 Ohm is used.
7. In the **Active** column, select the check box to use the port as a "source" (else the port is only a "receiving" or "sink" port).

> **Note:** For example, if a *Port1* and *Port2* is defined, but only *Port1* is active, only S_{11} and S_{21} are calculated.

During the calculation of S-parameters, the specified reference impedances are added as loads to the ports. These loads remain in place after the S-parameter calculation. If the loads are removed once the S-parameter calculation is complete and there are subsequent output requests, such as near fields, the full matrix computation and LU decomposition steps will be repeated for the MoM solution method. This is typically the most time-consuming step in the analysis. It must be noted that should near fields be requested, and the loads remain, the near fields will be lower in magnitude due to the losses in the loads.

8. In the **Label** field, add a unique label for the request.
9. Click **Create** to request the S-parameter results and to close the dialog.

The port numbers in an S-parameter solution are indexed based on the order of appearance in the port list on the **Create S-parameters** dialog, and not according to the label of the selected port.

33. transverse electric (TE)
34. transverse magnetic (TM)
35. transverse electric and magnetic (TEM)

Adding an S-parameter Configuration to Generate a Multiport Data Package

Define an S-parameter configuration to export a multiport data package.

1. Add an S-parameter configuration using one of the following workflows:
   - On the **Request** tab, in the **Configurations** group, click the **S-parameter Configuration** icon.
   - In the configuration list, click the icon. Select **S-parameter Configuration** from the menu .
2. In the **Power** group select one of the following:
   - Select **Specify transmit power** to specify the transmit power for all the ports when calculating the requests associated with the S-parameter request.
   - Select **Use unit magnitude convention** to use the unit magnitude convention to calculate the requests associated with the S-parameter request.

   > **Note:** The S-parameter matrix result is not affected when selecting the power option, only the fields or current requests associated with the S-parameter request is altered based on the power information.
3. In the **Options** group, select the **Generate multiport data package (*.mdp)** check box to create the data package archive.

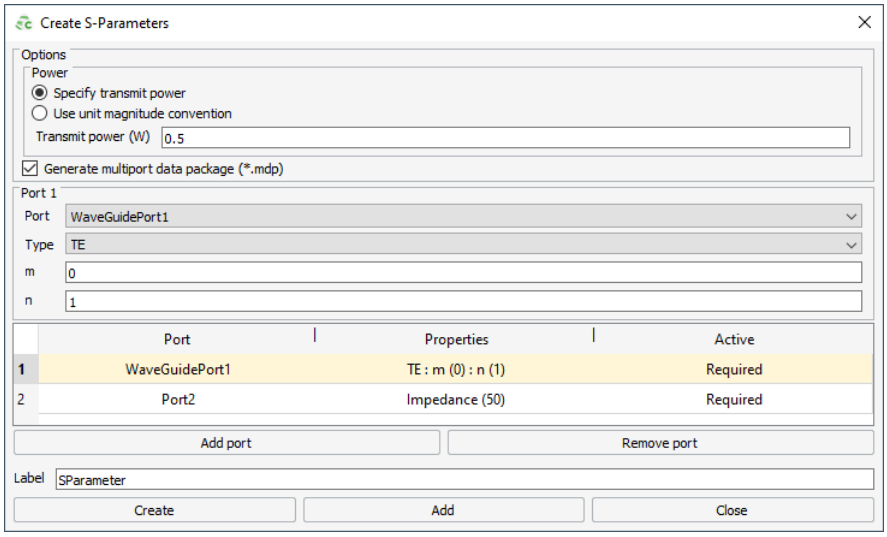


*Figure 293: The **Create S-Parameters** dialog.*

> **Note:** All the ports in the S-parameter configuration are activated.

4. Follow Step 5 and Step 6 to select the ports and set the properties.
5. In the **Label** field, add a unique label for the request.
6. Click **Create** to request the S-parameter results and to close the dialog.

7. [Optional] Add additional requests, for example, far fields, near fields and current requests to the S-parameter configuration to export the data to the multiport data package.

Characteristic Mode Configuration

A characteristic mode configuration results in a characteristic mode analysis (CMA) request. The analysis is based on the numerical calculation of a weighted set of orthogonal mode currents.

CMA gives insight into the fundamental resonant behaviour of a structure allowing you to follow a systematic design approach, making use of the parameters calculated with CMA, the modal current distribution and the modal weighting coefficients.

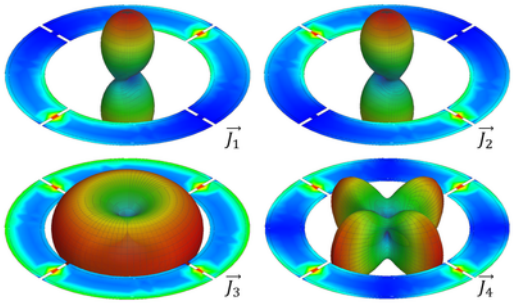


Figure 294: The first four modal currents and associated far field of a ring antenna with four slots bridged with edge ports.

The currents are supported on conducting surfaces as well as dielectric and magnetic materials with MoM / SEP and apertures with the planar Green's function.

Key parameters such as the resonance frequency of these modes and their radiating behaviour can be determined by studying the current distribution of these modes.

Related concepts
Which Solution Methods Support GPU Acceleration?

Adding a Characteristic Mode Configuration

Define and add a characteristic mode configuration to the model.

1. Add a characteristic mode configuration using one of the following workflows:
   - On the **Request** tab, in the **Configurations** group, click the **Characteristic Modes Configuration** icon.
   - In the configuration list click the . From the menu select **Characteristic Modes Configuration**.

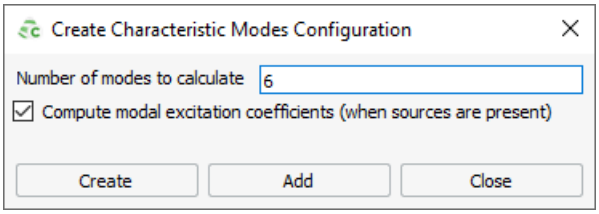


*Figure 295: The **Request Characteristic Modes Configuration** dialog.*

2. In the **Number of modes to calculate** field, enter the maximum number of modes to calculate.
3. Select the **Compute modal excitation coefficients (when sources are present)** check box to (in addition to the characteristic modes) calculate the modal excitation coefficients given a source.
4. Click **Create** to add the request and to close the dialog.

2.27.2 Modifying the Solution Order of Configurations

The Solver solves the configurations in the order in which the configurations are listed in the configuration list, but the order can be changed.

As an example, a configuration is moved up in the configuration list. Similar steps are followed to move the configuration down in the list.

1. In the configuration list, select the configuration.
2. Move the configuration using one of the following workflows:
   - From the right-click context menu, select **Move up**.
   - Press Ctrl++ to use the keyboard shortcut.

2.27.3 Excluding a Configuration from the Model

A configuration can be excluded from the solution without removing it from the model.

Note: Excluding a configuration does not delete the configuration.

1. In the configuration list, select the configuration that you want to exclude.
2. From the right-click context menu, select **Include/Exclude**.

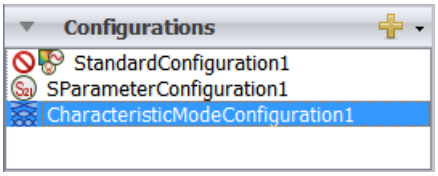


Figure 296: The icon indicates that the configuration is excluded from the solution.

2.27.4 Deleting a Configuration from the Model

Remove a configuration from the model.

1. In the configuration list, select the configuration that you want to remove.
2. From the right-click context menu, select **Delete**.

2.27.5 Global and Configuration Specific Entities

Entities such as frequency, sources, loads and power can be set globally or specified per configuration.

Global
Global refers to entities that are relevant to all configurations.

> **Tip:** Sources defined globally are applicable to standard configurations since sources are not applicable to S-parameter configurations. Sources are not required for characteristic mode configurations, but can be added.

Configuration-specific
Configuration-specific entities are only applicable to a specified configuration. For example, frequency per configuration and sources per configuration. Configurations inherit all global items.

> **Note:** By default, frequency, sources, loads and power are set globally.

Creating Configuration-Specific Entities

Convert a global entity to a configuration specific entity.

As an example, loads are converted to configuration specific loads, but the steps are similar for converting frequency, sources and power.

1. In the model tree, click **Loads**.
2. From the right-click context menu, select **Specify Loads per Configuration**.

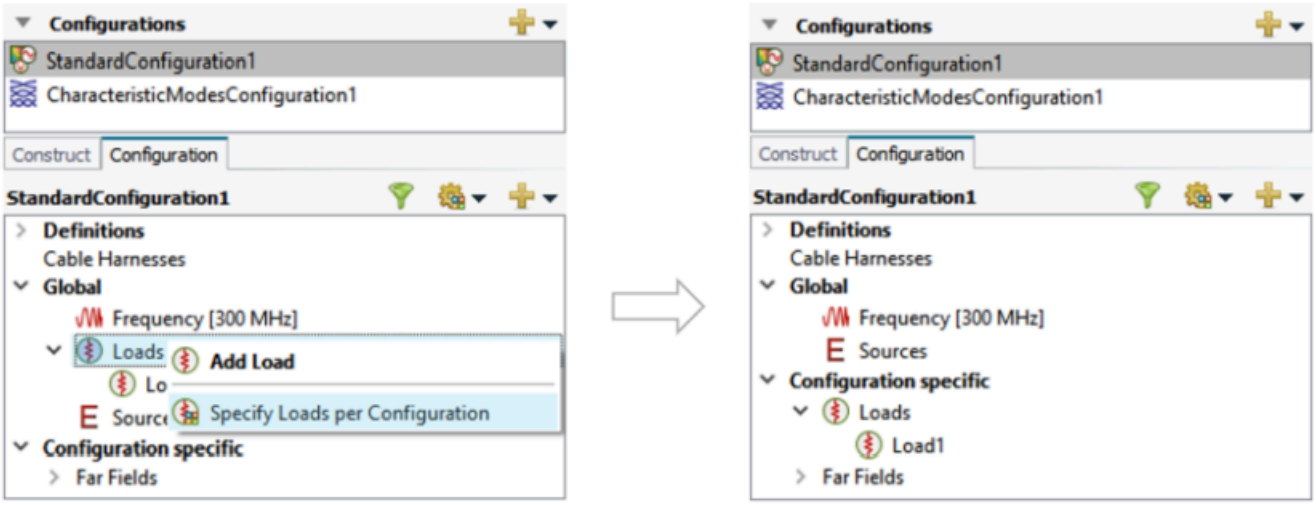


Figure 297: Convert a globally specified load to a configuration specific load.

All loads are converted from a global item to a configuration specific item and copied to all configurations.

Tip: Alternatively, in the model tree click to specify the configuration settings.

Converting Configuration Specific Entities to Global

Convert a configuration specific entity to a global configuration.

As an example, loads from a configuration are converted to global loads, but the steps are similar for converting frequency, sources and power.

1. In the model tree, click **Loads**.
2. From the right-click context menu, select **Specify Loads Globally**.
3. On the **Choose configuration** dialog, from the drop-down list, select the configuration for which the loads are to be converted to global loads (and as a result inherited by all configurations).
4. Click **OK** to create the loads and to close the dialog.

Copying Entities Between Configurations

Duplicate an entity and send to another configuration.

1. In the configuration list, select the configuration.
2. In the model tree, click the entity to be copied.
3. From the right-click context menu select **Send copy to** and select the destination configuration.

2.28 Requesting Calculations

Before running the Solver, define the output requests and view in the model tree (Configuration tab).

2.28.1 Automatically Calculated Results

When a model contains voltage sources or loads, some results are available by default.

The following results are available without requesting it:

- The input impedance for voltage and current sources.
- The voltages and currents for loads.

2.28.2 Requesting a Far Field

Add a far field request to the model.

1. On the **Request** tab, in the **Solution Requests** group, click the **Far Fields** icon.

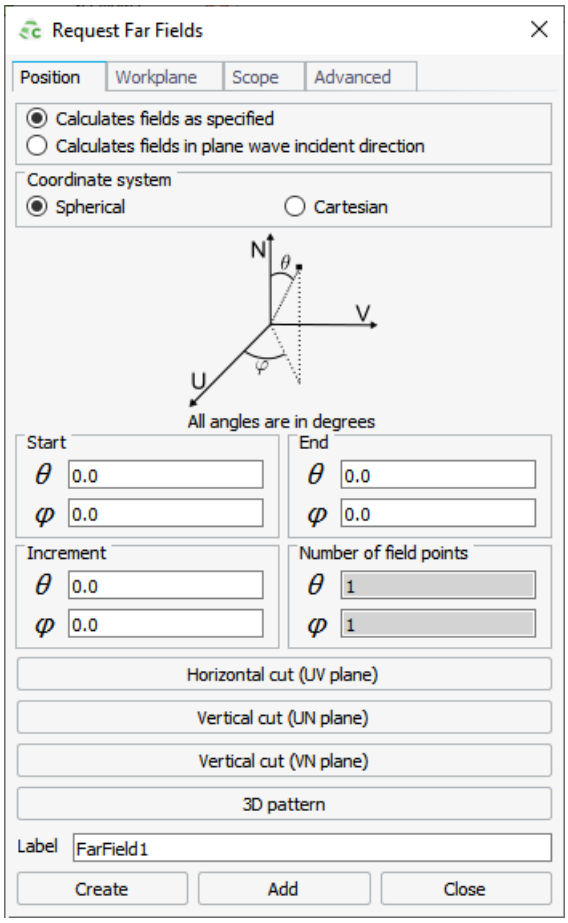


*Figure 298: The **Request Far Fields** dialog.*

2. Select one of the following:
   - To calculate the general far field pattern or bistatic RCS[36], click **Calculate fields as specified**. The **Spherical** coordinate system is generally used to define far fields.
     - To specify a pattern, enter the **Start**, **End** and **Increment**.
     - To use a commonly-defined pattern, click one of the following:
       - **Horizontal cut (UV plane)**
       - **Vertical cut (UN plane)**
       - **Vertical cut (VN plane)**
       - **3D pattern**
   - To calculate monostatic radar cross section (RCS) or if an RCS optimisation search is based on this far field request, click **Calculate fields in plane wave incident direction**.

     No additional parameters are required as the scattered fields are only calculated in the direction that the incident plane wave is coming from. The workplane settings of the incident plane (rotation or translation of the local coordinate system) are used for the far field calculation.

     Note: For an RCS calculation, the model must contain a plane wave source[37].
3. In the **Label** field, add a unique label for the request.
4. Click **Create** to request the far field result and to close the dialog.

Advanced Settings for Far Field Requests

Use advanced settings to specify the currents or subset thereof to be taken into account for the calculation of fields, the export of far field data, ignoring the radiated contribution from impressed sources, calculating spherical modes and calculating the far field for an array of elements.

On the **Request** tab, in the **Solution Requests** group, click the **Far Fields** icon. The advanced settings are available on the **Scope** tab and **Advanced** tab.

Specify the Currents Taken Into Account During Field Calculation

Field calculation using all sources
: With this option, the currents on all structures are taken into account when calculating the far field result (default).

Field calculation using only sources on elements with specified labels
: With this option, only the currents on structures with specified labels are taken into account when calculating the far field result.

36. radar cross section
37. with single or multiple directions of incidence

Export Directivity or Gain

This setting controls what is written to file (either directivity or gain).

Directivity
: This option writes out the directivity to the selected output files (`.out` and `.ffe` file).

Gain
: This option writes out the gain to the selected output files (`.out` and `.ffe` file).

Export Far Field Data

Export fields to ASCII file (.ffe)*
: This option exports the far fields to a `.ffe` file. Use this file for further post-processing or, when using spherical coordinates, as source pattern for a radiation pattern point source or a receiving antenna.

*Export fields to *.out file*
: This option exports the fields to the `.out` file.

Only determine radiated far field power by integration
: This option calculates the far fields and the total radiated power but the field values are not written to the `.bof` or `.out` output files.

> **Tip:** Use this option if the individual field values are not required and the output files would otherwise become too large. Far field values will not be available for viewing in POSTFEKO or for optimisation in OPTFEKO.

Far Field Interpolation

Calculate continuous far field data
: This option uses interpolation to display far field data in POSTFEKO.

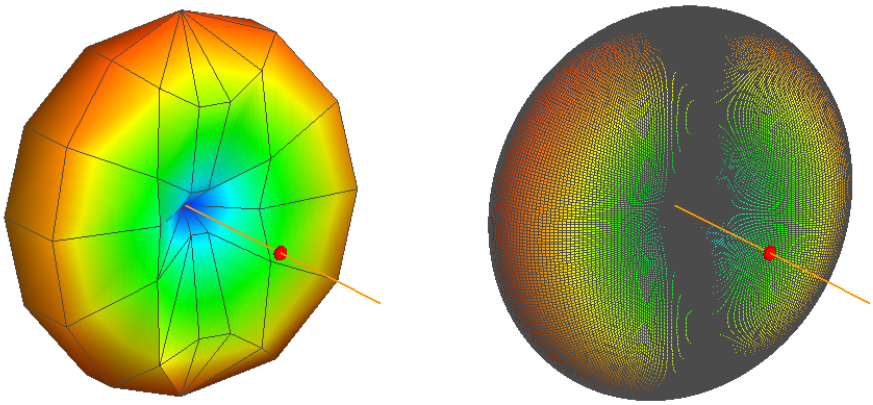


*Figure 299: The result of a 3D pattern far field request with θ and φ = 13 points (on the left) and the result of a 3D pattern far field request with θ = 7 and φ = 13 points with the **Calculate continuous far field data** option selected (to the right).*

Ignore Radiated Contribution from Impressed Sources

Calculate only the scattered part of the field
This option ignores the radiated contribution from impressed sources (for example, electric point source, magnetic point source) as well as contribution from plane wave sources, yielding only the scattered fields.

> **Note:** The default setting is recommended.

Spherical Mode Options

Calculate spherical expansion mode coefficients
This option calculates the coefficients and exports the data to a `.sph` (TICRA) file.

> **Note:** Set the radiated power in Feko to 4π Watts to ensure the gain in GRASP will be correct.

Specify number of modes
This option allows you to specify the maximum mode index. If no value is specified, the maximum mode index is calculated automatically.

Maximum mode index N
Specify the maximum mode index.

Export spherical expansion coefficients to ASCII file
This option exports the spherical expansion mode coefficients to an ASCII file.

Periodic Boundary Condition Options

Calculate far field for an array of elements
This option allows the far field to be calculated for an array of elements.

2.28.3 Requesting a Near Field

Add a near field request to the model.

1. On the **Request** tab, in the **Solution Requests** group, click the **Near Fields** icon.

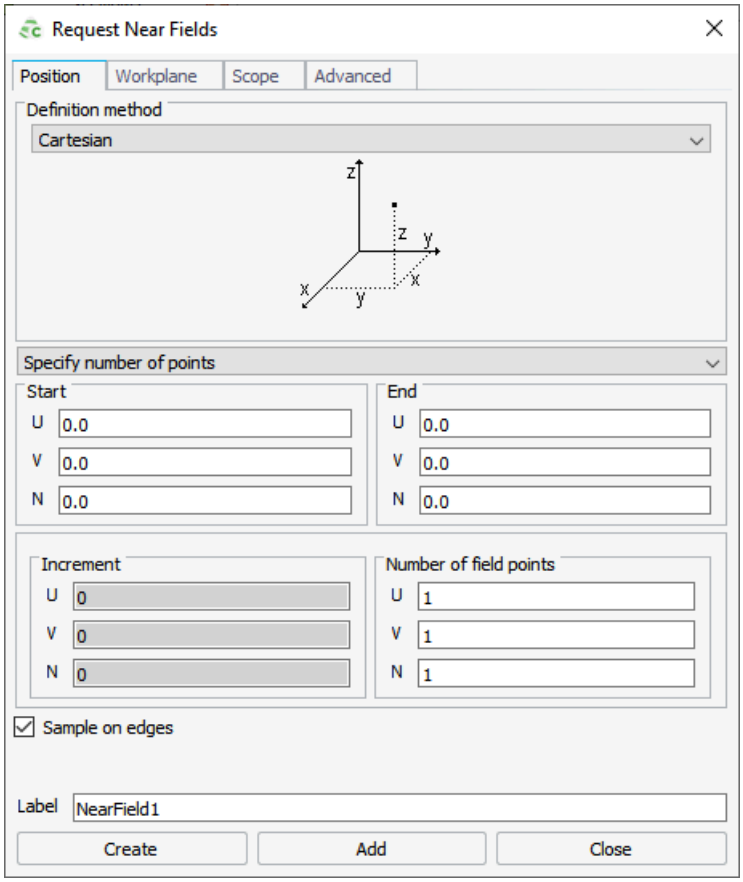


*Figure 300: The **Request Near Fields** dialog.*

2. Under **Definition methods**, select one of the following coordinate systems:
   - **Cartesian**
   - **Cartesian boundary**
   - **Conical**
   - **Cylindrical**
   - **Cylindrical (X axis)**
   - **Cylindrical (Y axis)**
   - **Spherical**
   - **Specified points**
   - **Tetrahedral mesh**

   Tip: To define a near field without using a coordinate system, select:
   - **Specified points** to calculate near fields at a list of defined or imported points.
   - **Tetrahedral mesh** to calculate near fields at the vertices and edge mid-points inside a tetrahedral mesh.

3. In the drop-down list, select one of the following:

- To specify the start and end points and the number of field points, select **Specify number of points**.
- To specify the start and end points and the increment between points, select **Specify increments**.

> **Tip:** The *actual* end point (depends on the start point, the number of field points and increment) may not coincide with the *specified* end point.

4. Select the **Sample on edges** check box to ensure the sample points lie on the edges of the request. If the check box is cleared, the sample points lie half an increment away from the edge of the request.
5. In the **Label** field, add a unique label for the request.
6. Click **Create** to request the near field result and to close the dialog.

Requesting a Near Field Boundary

Add a near field boundary request to the model. This type of request allows you to define a cuboidal near field request where the request points are located on the surface of the cuboid, but you have the option to exclude specific surfaces (faces).

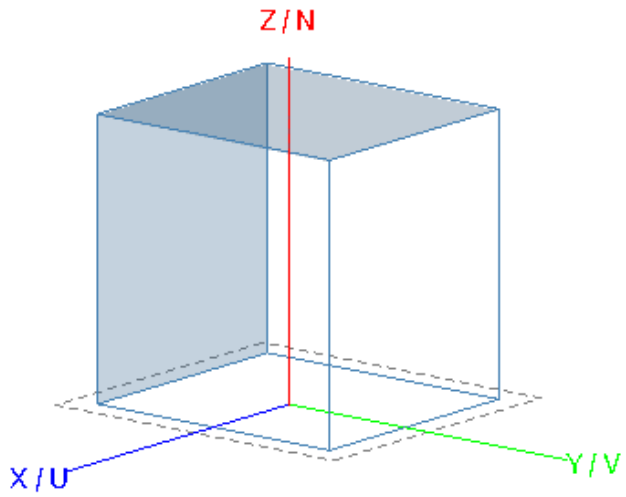


Figure 301: An example of a Cartesian boundary near field request with only the +N surface and -V surface included (faces shown in blue).

1. On the **Request** tab, in the **Solution Requests** group, click the **Near Fields** icon.

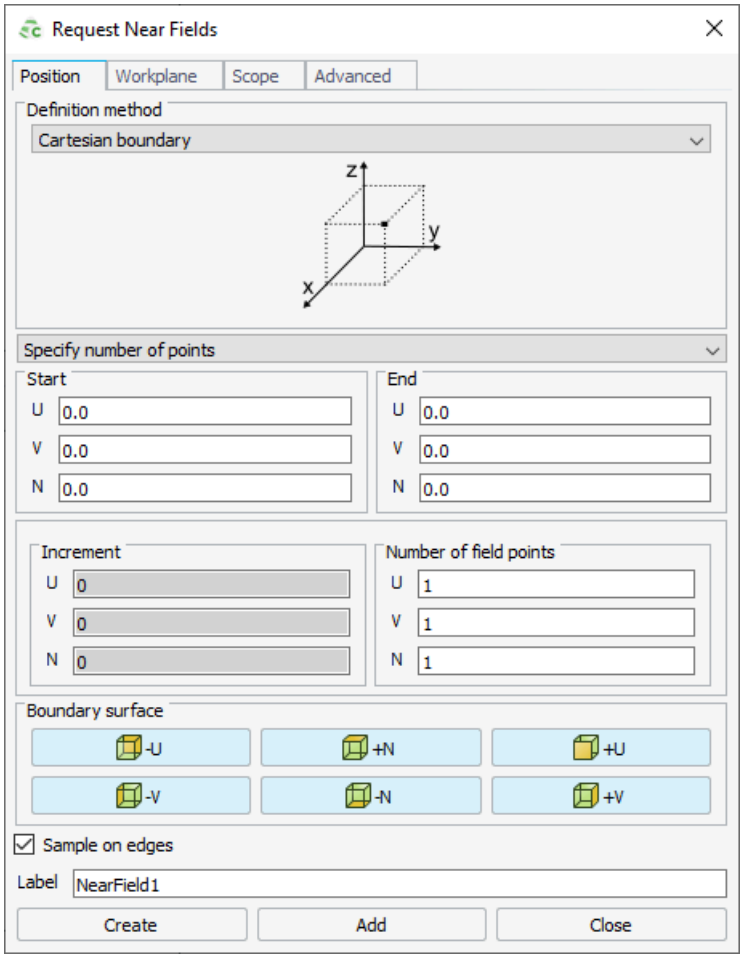


*Figure 302: The **Request Near Fields** dialog.*

2. Under **Definition method**, from the drop-down list, select **Cartesian boundary**.
3. In the , select one of the following:
   - To specify the start and end points and the number of field points, select **Specify number of points**.
   - To specify the start and end points and the increment between points, select **Specify increments**.

> **Tip:** The *actual* end point (depends on the start point, the number of field points and increment) may not coincide with the *specified* end point.

4. Under **Boundary surface**, click the applicable button if you want to exclude a surface from the Cartesian boundary near field request. Click on one or more of the following to exclude:
   - **-U**: Exclude the surface in the negative U direction.
   - **+N**: Exclude the surface in the positive N direction.
   - **+U**: Exclude the surface in the positive U direction.

- **-V**: Exclude the surface in the negative V direction.
- **-N**: Exclude the surface in the negative N direction.
- **+V**: Exclude the surface in the positive V direction.

5. [Optional] Select the **Sample on edges check** box to disable sampling on the edges of the Cartesian boundary.
6. In the **Label** field, add a unique label for the request.
7. Click to request the near field result and to close the dialog.

Advanced Settings for Near Field Requests

Use advanced settings to specify the currents taken into account for the calculation of fields or potentials, the export of near field data and ignoring radiated contributions from impressed sources.

On the **Request** tab, in the **Solution Requests** group, click the **Near Fields** icon. The advanced settings are available on the **Scope** tab and **Advanced** tab.

Specify the Currents Taken Into Account During Field Calculation

Field calculation using all sources
With this option, the currents on all structures are taken into account when calculating the near field result.

Field calculation using only sources on elements with specified labels
With this option, only the currents on structures with specified labels are taken into account when calculating the near field result.

Calculate Fields or Potential

Fields
This option calculates the actual near field components and are stored in the `.out`. The electric component and / or the magnetic field component can be included.

Potentials
This option allows a single potential type to be included in the near field request.

- Electric vector potential
- Electric scalar potential
- Gradient of the scalar electric potential
- Magnetic vector potential
- Magnetic scalar potential
- Gradient of the scalar magnetic potential

Export Near Field Data (Fields and Potential)

Export fields to ASCII file (.efe, *.hfe)*
This option exports the electric fields to a `.efe` file and the magnetic fields to a `.hfe` file.

*Export fields to *.out file*
This option exports the electric fields / potentials and the magnetic fields / potentials to the `.out` file.

*Export fields to SEMCAD *.dat file*
This option export the electric fields / potentials and the magnetic fields / potentials to the `.dat` file.

*Export fields to SPARK3D *.fse file*
This option exports the electric fields / potentials and magnetic fields / potentials calculated at the vertices and edge mid-points of the tetrahedra to a `.fse` file.

> **Tip:** Only valid if the **Tetrahedral mesh** option was selected (**Position** tab).

Ignore Radiated Contribution from Impressed Sources

Calculate only the scattered part of the field
This option ignores the radiated contribution from impressed sources (for example electric point sources and magnetic point sources), yielding only the scattered fields.

> **Note:** The default setting is recommended.

2.28.4 Requesting an Error Estimation

Add an error estimation request. Error estimation is an a-posteriori error indicator which gives feedback on the mesh quality. The mesh quality is determined by testing the solution against an unconstrained physical test.

1. On the **Request** tab, in the **Solution Requests** group, click the **Error Estimation** icon.

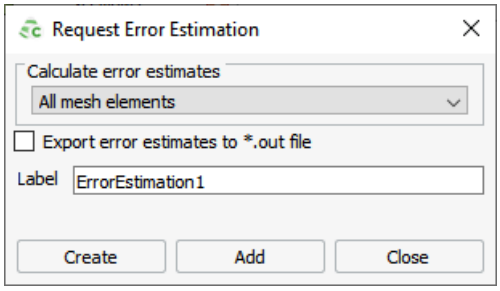


*Figure 303: The **Request Error Estimation** dialog.*

2. From the drop-down list, select one of the following:
   - To request error estimates on all mesh elements in the model, select **All mesh elements**.
   - To request currents only on triangles, select **Only error estimates on triangles**.
   - To request error estimates only on segments, select **Only error estimates on segments**.
   - To request error estimates only on segments, select **Only error estimates on tetrahedra**.

- To request error estimates only on mesh elements with specified labels, select **Only error estimates on specified labels**.

3. [Optional] Select the **Export error estimates to *.out file** check box to export the error estimates to the .out file.
4. In the **Label** field, add a unique label for the request.
5. Click **Create** to request the error estimates results and to close the dialog.

Related tasks
Refining the Mesh Adaptively Using Error Estimates

2.28.5 Requesting Currents

Add a request to calculate the currents in the model to be displayed in POSTFEKO.

Tip: The storage of currents in large models can lead to large output files. When using adaptive (continuous frequency) sampling, the interpolation of currents could increase the number of frequencies required for solution convergence as well as the total runtime. In general it is recommended to calculate the required currents at specific frequencies only.

Tip: If a continuous frequency solution is performed, add a second (standard) configuration to calculate currents at specific frequencies only.

1. On the **Request** tab, in the **Solution Requests** group, click the **Currents** icon.

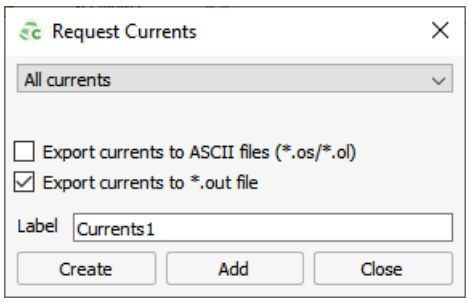


*Figure 304: The **Request Currents** dialog.*

2. From the drop-down list, select one of the following:
   - To request all currents in the model, select **All currents**.
   - To request currents only on segments and triangles with specified labels, select **Only currents on specified labels**.
   - To request currents only on wire segments, select **Only segment currents**.
   - To request currents only on triangles, select **Only triangle currents**.
3. [Optional] Select the **Export currents to ASCII file (*.os/*.ol)** check box to export the currents to a `.os` file and the charges to a `.ol` file.

4. [Optional] Select the **Export currents to *.out file** check box to export the currents to the .out file.
5. In the **Label** field, add a unique label for the request.
6. Click **Create** to request the currents and charges result and to close the dialog.

2.28.6 Requesting Model Decomposition

Add a model decomposition request to the model. This request exports the surface currents on selected faces to a `.sol` file. Use a `.sol` file to define a solution coefficient source in another model.

1. On the **Request** tab, in the **Solution Requests** group, click the **Model Decomposition** icon.

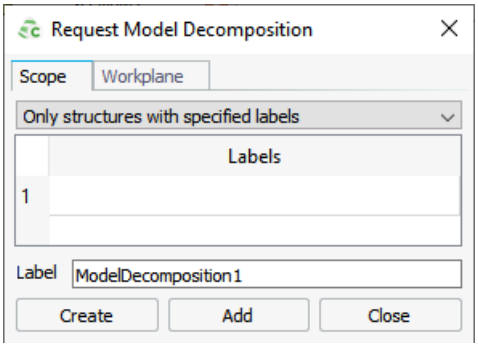


*Figure 305: The **Request Model Decomposition** dialog.*

2. From the drop-down list, select one of the following:
   - To request model decomposition for all structures, select **All structures**.
   - To request model decomposition only on structures with specified labels, select **Only structures with specified labels**.
3. In the **Label** field, add a unique label for the request.
4. Click **Create** to request the model decomposition and to close the dialog.

Related tasks
Defining Solution Coefficient Data from File
Adding a Solution Coefficient Source

2.28.7 Transmission and Reflection Coefficients

Calculate the properties of frequency selective surfaces (FSS) in a multilayer scattering scenario by using transmission and reflection coefficients for plane waves. Use in conjunction with periodic boundary conditions (PBC), multilayer planar Green's functions or infinite planes for a more efficient solution.

Note: Only a single plane wave is supported (no additional sources are allowed) when requesting transmission / reflection coefficients. The plane wave is allowed to have (loop over) multiple incident angles.

The model must either contain:

- planar multilayer substrate without any other geometry / mesh in the model

or

- a 2D periodic boundary condition (PBC).

The transmission coefficient is defined as:

$$\tau = \frac{E_t}{E_i} \tag{14}$$

and the reflection coefficient:

$$\rho = \frac{E_r}{E_i} \tag{15}$$

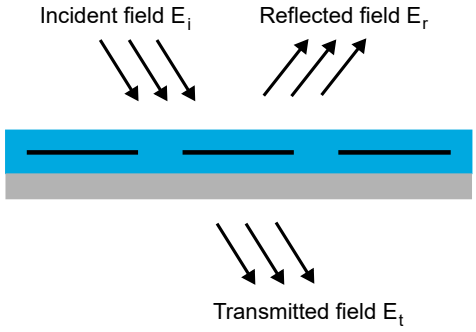


Figure 306: A plane wave interacting with a planar structure.

Requesting Transmission / Reflection Coefficients

Add a request to calculate the transmission / reflection coefficients for a plane wave interacting with a planar structure.

1. On the **Request** tab, in the **Solution Requests** group, click the **Transmission / Reflection** icon.

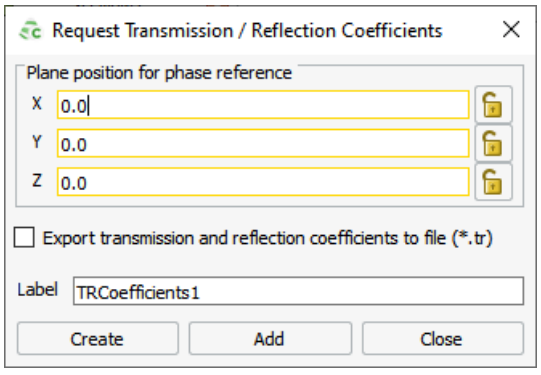


*Figure 307: The **Request Transmission / Reflection coefficients** dialog.*

2. Under **Plane position for phase reference**, specify the location (origin) of the plane wave in Cartesian coordinates.
3. [Optional] Select the **Export transmission and reflection coefficients to file (*.tr)** check box to export the transmission / reflection data from infinite surface structures.

   Note: To export a valid `.tr` file, your model must either contain a periodic boundary condition (PBC) or a planar Green's function.

4. In the **Label** field, add a unique label for the request.
5. Click **Create** to request the transmission / reflection coefficients and to close the dialog.

Related concepts
Periodic Boundary Condition (PBC)
Infinite Planes and Half-Spaces

2.28.8 Ideal Receiving Antennas

An ideal receiving antenna is a tool that calculates the power that would be received by an ideal antenna. Use this type of antenna for a more computationally efficient solution.

Note: Ideal receiving antennas are supported by all solution methods, except FDTD.

The following types of receiving antennas are available:

RX far field antenna
The antenna is located at a point in space with the spatial receiving properties of a far field imported from simulated or measured data.

RX near field antenna
The antenna consists of near field apertures that serve as weights for receiving energy via the specified aperture points.

RX spherical modes antenna
The antenna is located at a point in space with the spatial receiving properties defined by spherical modes. The combination of the spherical modes effectively defines how energy is received, incident upon the antenna from any particular direction.

The following assumptions are made regarding ideal receiving antennas:

- The antenna is considered to be matched (no mismatch loss is taken into account).
- The antenna and model geometry are assumed to have no impact on each other during the solution phase (no coupling is taken into account).

Requesting Ideal Receiving Antenna (Far Field Pattern)

Add an ideal receiving antenna (far field pattern) request to the model.

1. On the **Request** tab, in the **Solution Requests** group, click the **Receiving Antenna** icon. From the drop-down list, select the **RX Far Field Antenna** icon.

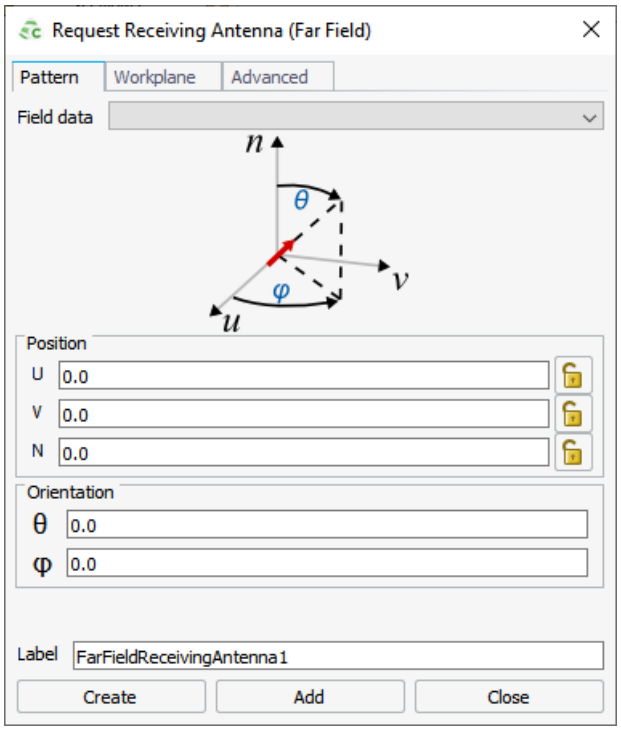


*Figure 308: The **Request Receiving Antenna (Far Field)** dialog.*

2. In the **Field data** field, specify the field data to be used to define the far field receiving antenna. The field data must be a far field specified using the spherical coordinate system.
3. In the **Position** field, specify where the receiving antenna is to be placed.
4. In the **Orientation** field, specify the orientation of the receiving antenna.
5. [Optional] Click the **Advanced** tab. Select the **Include only the scattering part of the field** check box to ignore the radiated contribution from impressed sources as well as the contribution from plane wave sources, yielding only the scattered fields.
6. In the **Label** field, add a unique label for the request.
7. Click **Create** to request the receiving antenna result and to close the dialog.

Requesting Ideal Receiving Antenna (Near Field Pattern)

Add an ideal receiving antenna (near field pattern) request to the model.

1. On the **Request** tab, in the **Solution Requests** group, click the **Receiving Antenna** icon. From the drop-down list, select the **RX Near Field Antenna** icon.

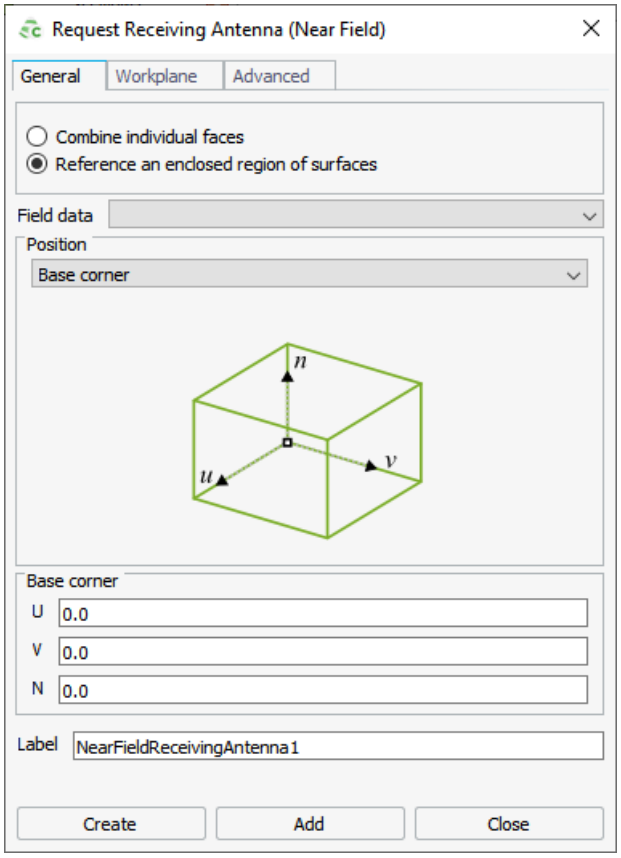


*Figure 309: The **Request Receiving Antenna (Near Field)** dialog.*

2. Define the near field aperture using one of the following:
   - To create a single near field aperture using individual near field definitions, click **Combine individual faces**.
     - In the **Field data** column, specify the field data for each face.
   - To create a single near field enclosed region, select **Reference an enclosed region of surfaces**.
     - Specify the field data and location of the region.
3. In the **Label** field, add a unique label for the request.
4. [Optional] Click the **Advanced** tab. Select the **Include only the scattering part of the field** check box to ignore the radiated contribution from impressed sources as well as the contribution from plane wave sources, yielding only the scattered fields.
5. Click **Create** to request the receiving antenna result and to close the dialog.

Requesting Ideal Receiving Antenna (Spherical Modes)

Add an ideal receiving antenna (spherical modes) request to the model.

1. On the **Request** tab, in the **Solution Requests** group, click the **Receiving Antenna** icon. From the drop-down list, select the **RX Spherical Mode Antenna** icon.

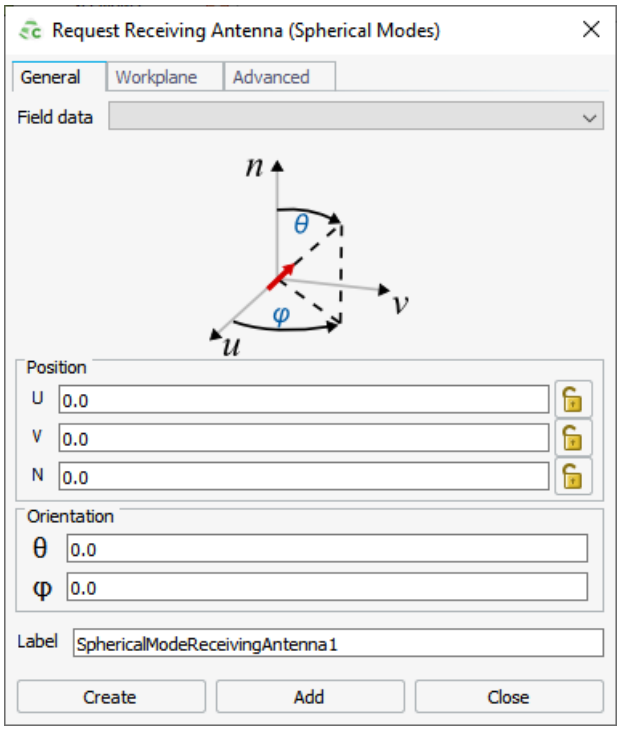


*Figure 310: The **Request Receiving Antenna (Spherical Modes)** dialog.*

2. In the **Field data** field, specify the field data to be used to define the spherical modes receiving antenna.
3. In the **Position** field, specify where the receiving antenna is to be placed.
4. In the **Orientation** field, specify the orientation of the receiving antenna.
5. [Optional] Click the **Advanced** tab. Select the **Include only the scattering part of the field** check box to ignore the radiated contribution from impressed sources as well as the contribution from plane wave sources, yielding only the scattered fields.
6. [Optional] Specify the internal spherical modes approximation method by selecting one of the following from the drop-down list:
   - To describe the receiving antenna by the spherical mode expansions of the radiated and received antenna fields, select **Use spherical modes approximation**.
   - To describe the receiving antenna by an impressed radiation pattern obtained internally from the spherical mode description, select **Use far field approximation**.
   - To let the Solver determine automatically if either the far field or spherical mode approximation should be used for the model, select **Auto**.
7. In the **Label** field, add a unique label for the request.

8. Click **Create** to request the receiving antenna result and to close the dialog.

2.28.9 Requesting Specific Absorption Rate (SAR)

Add a request to calculate the average absorption over a volume (volume-average SAR) or the maximum absorption in a 1 g or 10 g cube in a given volume (spatial-peak SAR).

1. On the **Request** tab, in the **Solution Requests** group, click the SAR icon.

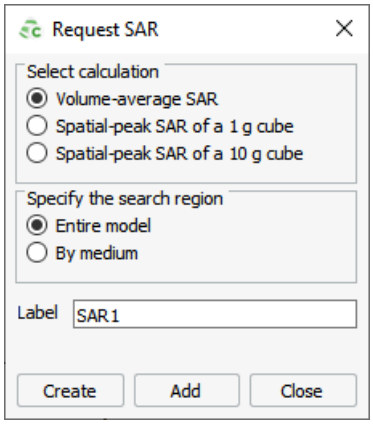


*Figure 311: The **Request SAR** dialog.*

2. Under **Select calculation**, select the type of SAR calculation:
   - To calculate the average absorption over a volume, select **Volume-average SAR**.
   - To calculate the maximum absorption in a 1g cube in the model, select **Spatial-peak SAR of a 1g cube**.
   - To calculate the maximum absorption in a 10g cube in the model, select **Spatial-peak SAR of a 10g cube**.
3. Specify the region where the SAR is calculated. Under **Specify the search region**, select one of the following:
   - To calculate SAR in all the dielectric regions in the model and calculate a single average or peak SAR value, select **Entire model**.
   - To calculate SAR in all media or a specified medium, select **By medium**.
   - To calculate SAR in a specific layer or in all the layers of a planar substrate, select **In a planar substrate**.

     Note:
     - Layer 0 is the upper free space region.
     - Layer 1 is the uppermost dielectric layer.

   - To calculate the 1g or 10g cube SAR at a specified location, select **At a specified position**. This option is not available for **volume average SAR**.
4. In the **Label** field, add a unique label for the request.

5. Click **Create** to request the SAR result and to close the dialog.

Related reference
SAR Standards

2.28.10 Requesting Cable Probe Data

Add a request to calculate the voltage or current along a cable path.

1. On the **Request** tab, in the **Solution Requests** group, click the **Cable Probe** icon.

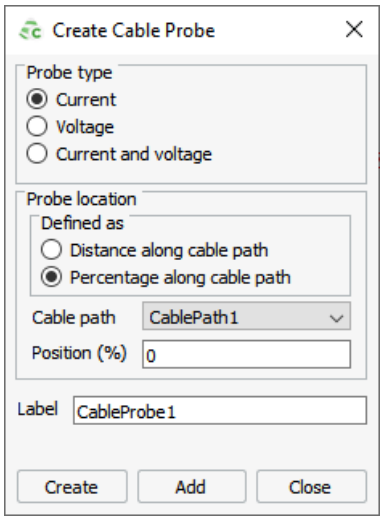


*Figure 312: The **Create Cable Probe** dialog.*

2. Under **Probe type**, select one of the following:
   - To view the current along a cable path, select **Current**.
   - To view the voltage along a cable path, select **Voltage**.
   - To view the current and voltage along a cable path, select **Current and voltage**.
3. Specify the probe location using one of the following workflows:
   - To specify the location as a percentage of the total path length (beginning from the start connector), under **Probe location**, select **Percentage along cable path**.
     - In the **Position (%)** field, specify the percentage of the total path length, where "0%" translates to the start of the cable path.
   - To specify the location as a specified distance from the start connector, under **Probe location**, select **Distance along cable path**.
     - In the **Position (distance)** field, specify the distance along the cable path, where a "0" distance translates to the start of a cable path.
4. In the **Cable path** drop-down list, select the cable path where the cable probe is to be placed.
5. In the **Label** field, add a unique label for the request.
6. Click **Create** to request the cable probe result and to close the dialog.

2.28.11 Requesting S-Parameters

To add an S-parameter request to the model, add an S-parameter configuration.

On the **Request** tab, in the **Configurations** group, click the **S-parameter Configuration** icon.

Related tasks
Adding an S-Parameter Configuration

2.28.12 Requesting Characteristic Mode Analysis (CMA)

To add a characteristic mode analysis (CMA) request to the model, add a characteristic modes configuration

On the **Request** tab, in the **Configurations** group, click the **Characteristic Modes Configuration** icon.

Related tasks
Adding a Characteristic Mode Configuration

2.29 Infinite Planes and Half-Spaces

Use an infinite plane or half-space to model a ground plane efficiently. The number of triangles in the model is reduced as the ground plane is not discretised into triangles.

On the **Construct** tab, in the **Structures** group, click the **Planes/Arrays** icon. From the drop-down list, select **Plane / Ground**.

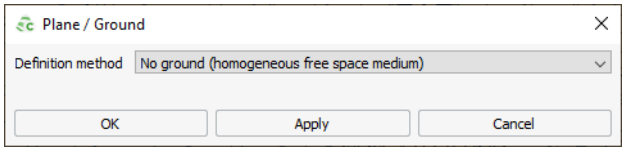


*Figure 313: The **Plane / ground** dialog.*

No Ground (Homogeneous Free Space Medium) [Default]

The model is solved in a homogeneous environment filled with free space medium. Edit the properties of free space if required.

Perfect Electric (PEC) Ground Plane at Z=0

Add an infinite PEC ground plane at Z=0 (in the global coordinate system) using the exact reflection coefficients, where the reflected field is added to get the total field.

Note: For a PEC ground plane, dielectric and metallic faces may connect to the ground plane and may coincide with ground plane, but may not cross or be below the ground plane.

Perfect Magnetic (PMC) Ground Plane at Z=0

Add an infinite PMC ground plane at Z=0 (in the global coordinate system) using the exact reflection coefficients, where the reflected field is added to get the total field.

Note: For a PMC ground plane, only metallic faces may connect to the ground plane, but may not coincide with the ground plane.

Homogeneous Half Space in Region Z<0 (Reflection Coefficient Approximation)

Add an infinite dielectric or a metallic half space for Z<0 with the boundary at Z=0 (in the global coordinate system). The half space uses the reflection coefficient ground plane approximation, where the reflected field is added to get the total field.

Note:

- For the reflection coefficient approximation ground, the structures must be above (Z>0) and at least $\frac{\lambda}{10}$ away from the ground plane, where λ is the free space wavelength.
- This technique is faster and potentially less accurate than the exact Sommerfeld integrals method.

Homogeneous Half Space in Region Z<0 (Exact Sommerfeld Integrals)

Add an infinite dielectric ground plane for Z<0 with the boundary at Z=0 (in the global coordinate system). The half space uses the Sommerfeld integrals to solve the exact boundary condition with the appropriate Green's function.

Note:

- A dielectric face may not coincide with the Z=0 half-space boundary.
- A metallic face may coincide with the Z=0 half-space boundary.
- Structures may cross the Z=0 half-space boundary.
- Structures may be inside (Z<0) the half-space boundary.

Planar Multilayer Substrate

Add a planar multilayer substrate (finite or infinite) orthogonal to the Z axis (in the global coordinate system).

Note:

- Supports arbitrarily shaped structures inside the substrate. Structures may cross multiple layers.
- Enclose the substrate in a MoM / SEP region to create a finite planar multilayer substrate.

Related tasks

Defining an Infinite Planar Multilayer Substrate

Defining a Finite Planar Multilayer Substrate

Related reference

Supported Solution Method and Technique Combinations

2.29.1 Defining an Infinite Planar Multilayer Substrate

Define an infinite planar multilayer substrate.

Some applications for infinite planar multilayer substrates are as follows:

- Add a PEC ground plane at the top and bottom layers to model a stripline.
- Add a PEC ground plane at the bottom layer to model a microstrip.
- Use the substrate (without a PEC ground plane) to model real earth.
- Use a finite thickness substrate without any ground plane to model a printed antenna (for example, a log-periodic antenna).

1. On the **Construct** tab, in the **Structures** group, click the **Planes/Arrays** icon. From the drop-down list, select **Plane / Ground**.

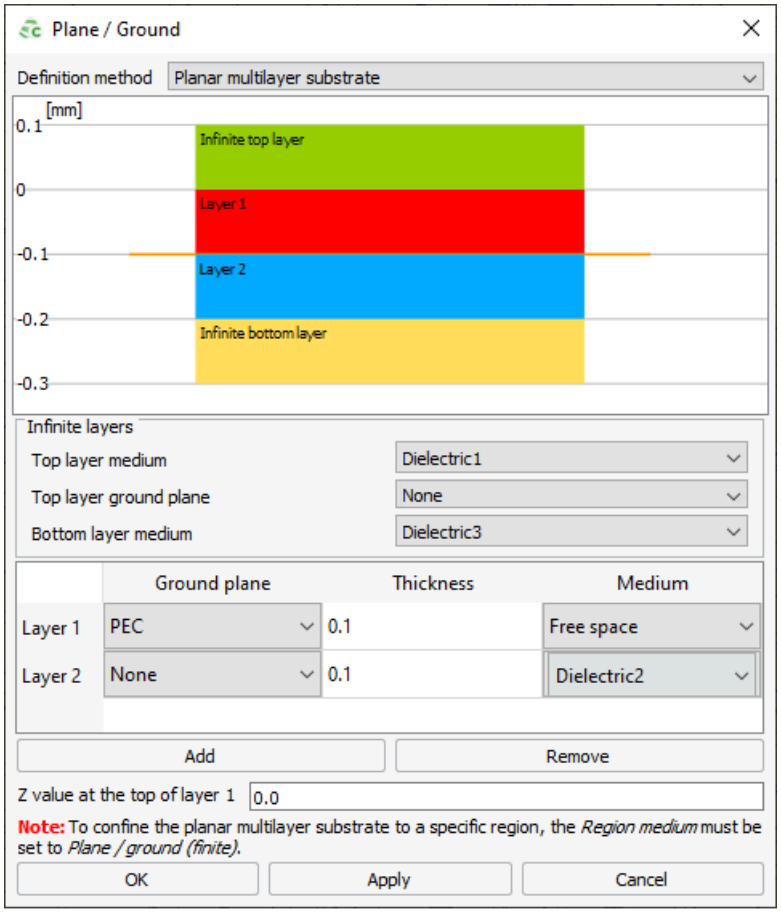


*Figure 314: The **Plane / ground** dialog.*

2. From the **Definition method** drop-down list, select **Planar multilayer substrate**.
3. Under **Infinite layers**, specify the top and bottom infinite layers:
   a) In the **Top layer medium** drop-down list, select the medium for the top layer that extends into infinity.
   b) In the **Top layer ground plane** drop-down list, specify whether the top layer should have a ground plane.
      - If the top layer has no ground plane, select **None**.
      - To add a ground plane for the top layer, select **PEC**.
   c) In the **Bottom layer medium** drop-down list, select the medium for the bottom layer that extends into infinity.
4. Click **Add** to add an additional layer. Click **Remove** to remove the selected layer from the substrate.
5. For each layer:
   a) From the **Ground plane** drop-down list, select one of the following:
      - To remove the PEC ground plane located below the current layer, click **None**.
      - To add a PEC ground plane located below the current layer, click **PEC**.
   b) Specify the layer thickness.

c) Specify the medium.

> **Note:**
> - The top layer and bottom layer extends into infinity.
> - In the CADFEKO GUI, the first layer is indexed as 0.
> - In the CADFEKO API, the first layer is indexed as 1.

6. In the **Z-value at the top of layer 1** field, specify where the top of layer 1 is located.
7. Click **OK** to define the infinite planar multilayer substrate and close the dialog.

Related reference
Planar Multilayer Substrate Limitations

2.29.2 Defining a Finite Planar Multilayer Substrate

Enclose an infinite multilayer substrate inside a MoM / SEP region to model a finite-size planar multilayer substrate.

1. Define an infinite planar multilayer substrate.
2. Create the solid that will contain the substrate (if it does not already exist).
3. In the model tree, select the solid.
4. In the details tree, select the region.
5. From the right-click context menu, select **Properties**.
6. On the **Modify Region** dialog, click the **Properties** tab.
7. From the **Medium** drop-down list, select **Plane / ground (finite)**.

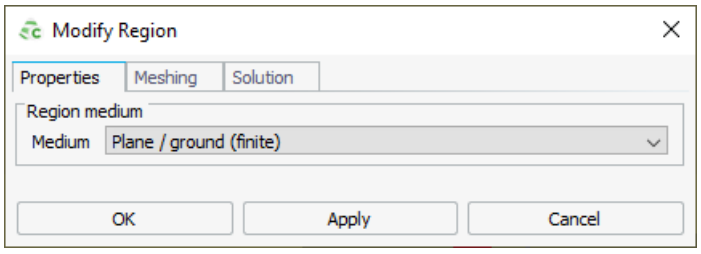


*Figure 315: The **Region properties** dialog.*

8. Click **OK** to enclose the substrate inside the region and close the dialog.

Related reference
Planar Multilayer Substrate Limitations

2.29.3 Planar Multilayer Substrate Limitations

A number of limitations apply when using the planar multilayer substrate (infinite or finite).

Tip: A finite planar multilayer substrate is a multilayer substrate confined to a region.

Table 15: Solution methods and planar multilayer substrates.

Solution Method	Planar Multilayer Substrate	
	Infinite	Finite
MoM (SEP)	Yes	Yes
MoM (VEP)	No	No
MLFMM	No	Yes
FEM	Yes	Yes
FDTD	No	No
PO	No	No
LE-PO	No	No
RL-GO	Yes (if confined to the uppermost layer of the substrate)	No
UTD & faceted UTD	No	No
ACA[38]	Yes	Yes
Low-frequency stabilisation for MoM	No	No
DGFM[39]	No	No
Periodic boundary conditions	No	No
CBFM[40]	No	No
Cable modelling	No	No

38. adaptive cross-approximation method
39. domain Green's function method
40. characteristic basis function method

Solution Method	Planar Multilayer Substrate	
	Infinite	**Finite**
MFIE, CFIE	No	No

Media

The following media types are not supported in conjunction with the planar multilayer substrate:

- Dielectric and magnetic cuboids
- Windscreen and layered dielectrics

Ports and Sources

The following ports and sources are not supported in conjunction with the planar multilayer substrate:

- Waveguide port
- Spherical mode source
- Solution coefficient source

General Restrictions

The following restrictions apply when using a planar multilayer substrate:

- A reflective ground plane in the model is not supported in conjunction with a planar multilayer substrate

  Note: A ground plane may be defined in the layers of the planar multilayer substrate.

- A mesh element must be located within a single layer of the planar multilayer.
- A mesh element may not intersect a ground plane in the planar multilayer.
- A dielectric triangle may not lie on the interface (boundary) between substrate layers.
- An aperture triangle must coincide with a planar multilayer substrate PEC ground plane.

Related tasks

Defining an Infinite Planar Multilayer Substrate

Defining a Finite Planar Multilayer Substrate

2.30 Meshing the Geometry / Model Mesh

2.30.1 Mesh Overview

A mesh is a discretised representation of a geometry model or mesh model. The geometry model (or mesh model) is meshed to obtain a simulation mesh which is given as input to the Solver to calculate the requests. The accuracy of the results depends greatly on generating a mesh that is an accurate representation of the model.

For preliminary simulations of your model, you are recommended to use a coarse mesh to obtain initial and fast results that can be used as a rough verification (although the results may be inaccurate). As a next step, you should do mesh convergence tests[41] before taking the simulated results as truly final and accurate.

The following terminology is used:

Geometry part / Model geometry
: Geometry part or model geometry refers to the computer-aided design (CAD) in the model. The CAD could be created in CADFEKO or imported from a wide range of CAD formats.

Mesh part / Model mesh
: The mesh part or model mesh is similar to CAD parts, but the mesh parts are models created from mesh elements, not CAD. The mesh is created either in CADFEKO (by unlinking from meshed CAD or creating mesh elements directly) or importing a mesh.

Simulation mesh
: The simulation mesh refers to the final mesh used by the Solver. CAD always has to be meshed. Models created from mesh parts can either be remeshed to create a simulation mesh, or they can be used without being remeshed. The simulation mesh is then the same as the model mesh.

View the model mesh and the respective simulation mesh in the model tree (Construct tab).

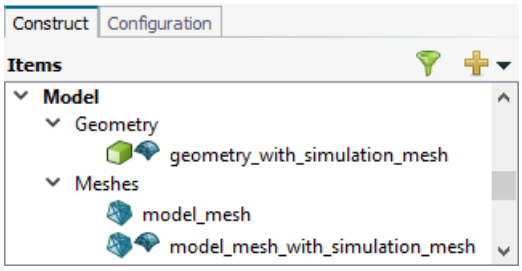


Figure 316: Example of (1) a geometry part that was meshed and has a simulation mesh, (2) a mesh part without a simulation mesh (still needs to be meshed) and (3) a mesh part that has a simulation mesh.

41. Rerun the model with 50% more elements and compare the results with that of the original mesh.

Note:

- The icon indicates a mesh part or model mesh.
- The icon indicates that a mesh instance has already been created or defined. This mesh instance is used when simulating the model.

2.30.2 Auto Meshing

The automatic mesh algorithm calculates and creates the mesh automatically once the frequency is set or local mesh settings are applied.

While the mesh is calculated by the automatic mesh algorithm, you can continue with setting up the model configuration. When geometry is transformed, no re-meshing is required since transforms are applied directly to the mesh. The automatic mesh algorithm also takes into account the proximity of other geometry or mesh to create a finer mesh.

While the mesh is being calculated, its status is indicated by a progress bar in the status bar.

For large models that takes a while to mesh, the automatic meshing may be disabled.

Related concepts
Status Bar
Auto Determine the Mesh Sizes
Related tasks
Disabling/Enabling Auto Meshing

Disabling/Enabling Auto Meshing

For very large models that have millions of mesh elements, auto meshing can be disabled while you make changes to the model.

Disable the auto meshing using one of the following workflows:

- On the **Mesh** tab, in the **Meshing** group, click the **Automatic Meshing** icon.
- On the status bar, click the **Automatic Meshing** icon.

Tip: Click the icon again to enable the auto meshing.

2.30.3 Mesh Element Types

CADFEKO supports segments, triangles, tetrahedra and voxels as mesh elements. The type of mesh element used to create a mesh is directly coupled to the solver method.

CADFEKO supports the following types of mesh elements:

Segments
A line segment consists of two vertices. Segments are the default type of mesh element to represent wires and are used for all solver methods.

Triangles
A triangle consists of three sides with three corner vertices. A curvilinear triangle consists of three sides with six vertices. Triangles are the default type of mesh element and is used for all solver methods, but excluding finite difference time domain (FDTD), finite element method (FEM) and uniform theory of diffraction (UTD).

Tetrahedra
A tetrahedron consists of four triangular faces, six edges and four corner vertices. Tetrahedral elements are used for volume equivalence principle (VEP) and finite element method (FEM) solver methods.

Voxels
A voxel is a cuboid on a grid in three-dimensional space. Voxels are the mesh element type used in conjunction with the FDTD solver method.

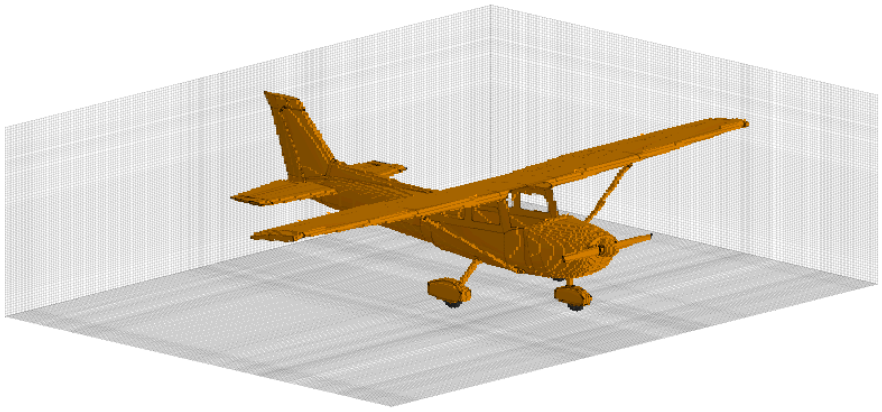


Figure 317: A model with voxel mesh gridlines indicating the size of the voxels.

2.30.4 Auto Determine the Mesh Sizes

A mesh can be created quickly without you having any knowledge of what the ideal mesh size should be for the model. Use the coarse, standard or fine mesh size to determine the correct mesh size for the model that takes into account the frequency, solution method, media properties and curvature of the model.

When the mesh size is determined automatically, the mesh is discretised relative to the wavelength of an electromagnetic wave in the medium of propagation. Each solution method has different requirements that influence the mesh size. In most cases, the automatic mesh size using the coarse, standard or fine option will give a reasonable result, but local mesh refinement may still be necessary.

The following model properties are considered when creating an automatic mesh size using the coarse, standard or fine option[42]:

42. The coarse, standard or fine mesh option is available on the **Modify Mesh** dialog.

Frequency
The simulation frequency of the model impacts the automatic mesh generated. The shortest wavelength corresponds to the highest simulation frequency.

Solver method
The solver method being used to solve the problem impacts the mesh requirements. For example, a finite element method (FEM) model requires settings for tetrahedra, a method of moments (MoM) solution requires settings for triangles and wires, while a hybrid solution needs to take into account the mesh requirements for multiple solution methods.

Dielectric properties
The dielectric properties of the media in the model affects the wavelength. Dielectric media are taken into account in all cases (except in the case where infinite layers are used). In the case where infinite layers are used, a local mesh refinement must be applied.

Geometry curvature
In cases where a finer mesh is not paramount for accurate solution results, it may still be required to accurately model aspects of the geometry. An automatic mesh size will attempt to reasonably conform to the original geometry.

> **Tip:** Modify the mesh settings on the **Create mesh** dialog, on the **Advanced** tab and note the effect the settings have on the resulting mesh.

Automatic mesh settings are only applied to regions, faces, edges and wires that do not have a mesh size set (locally or globally). When a local mesh refinement is applied to an individual component in the model, the local mesh refinement receives higher priority and will never be overwritten by the automatic meshing sizes.

Related reference
Automatic Meshing for Wires
Automatic Meshing for Faces and Edges
Automatic Meshing for Regions
Automatic Meshing for Voxels

2.30.5 Modifying the Auto-Generated Mesh

Adjust the auto-generated mesh that is generated when the frequency is set or local mesh settings are applied to the geometry.

> **Note:**
> - To generate a tetrahedral mesh, activate the FEM solution method.
> - To generate a voxel mesh, activate the FDTD solution method.

1. Open the **Modify Mesh** dialog using one of the following workflows:
   - On the **Mesh** tab, in the **Meshing** group, click the **Modify Mesh** icon.
   - Press Ctrl+M to use the keyboard shortcut.

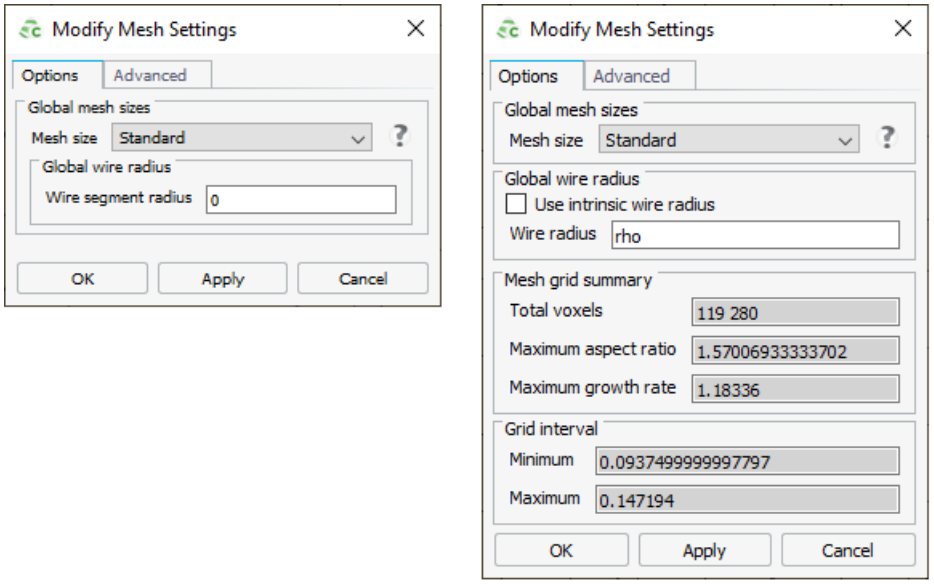


*Figure 318: The **Modify Mesh Settings** dialog (**Options** tab). On the left, the dialog for triangles and tetraheda and to the right, the dialog for voxels.*

2. Specify the mesh size.
   - To create a mesh using automatic mesh sizes, in the **Mesh size** field, from the drop-down list select **Coarse**, **Standard** or **Fine**.
   - To create a mesh with a custom mesh size, in the **Mesh size** field, from the drop-down list select **Custom**. Specify the lengths applicable to the model.
     1. In the **Triangle edge length** field, specify the triangle edge length.
     2. In the **Wire segment length** field, specify the wire edge length.
     3. In the **Tetrahedron segment length** field, specify the tetrahedron edge length.
     4. In the **Voxel size** field, specify the length for the voxel width, depth and height.
3. Specify the global wire radius.
   - To specify a wire radius, clear the **Use intrinsic wire radius** check box and in **Wire radius** field, enter a value for the global wire radius.
   - To allow the Solver to determine the wire representation, select the **Use intrinsic wire radius** check box.

     Tip: This option may improve the FDTD convergence.

   Note: A local mesh refinement takes precedence over global mesh settings.
4. Click **OK** to modify the mesh and to close the dialog.

Related tasks

Solving a Model with FDTD

Solving a Region with FEM

Related reference
Automatic Meshing for Wires
Automatic Meshing for Faces and Edges
Automatic Meshing for Regions
Automatic Meshing for Voxels

Advanced Meshing Options

A number of advanced meshing options are available that allows you the flexibility and advanced mesh control for segments, triangles, tetrahedra and voxels.

On the **Mesh** tab, in the **Meshing** group, click the **Modify Mesh** icon. On the **Create mesh** dialog, click the **Advanced** tab.

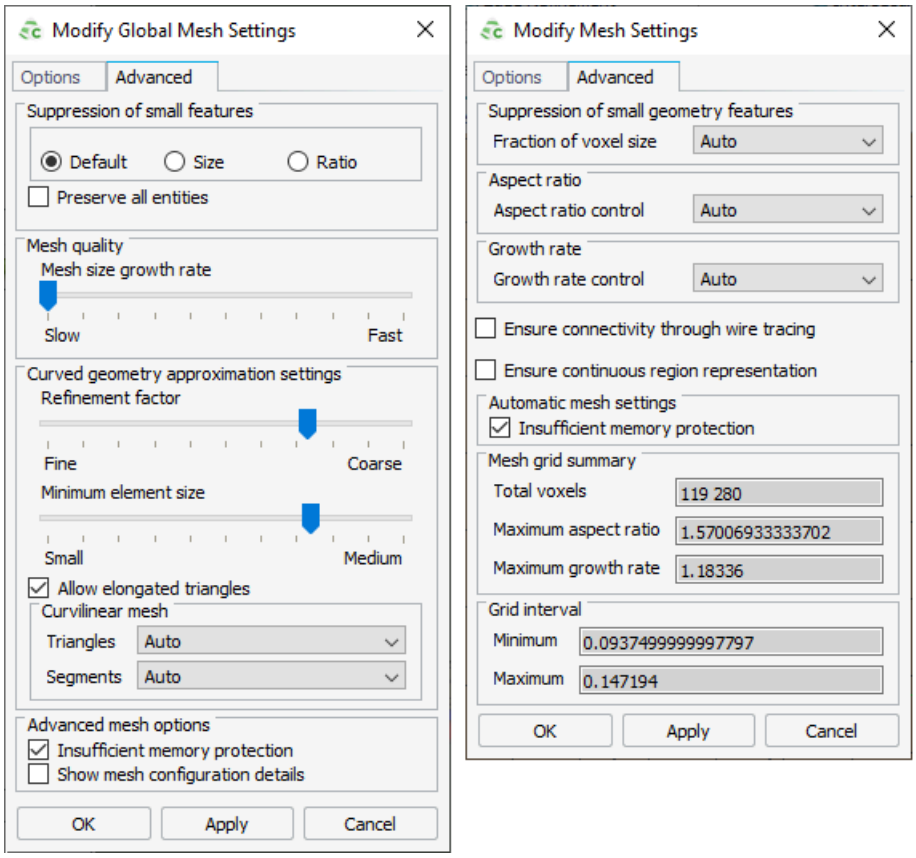


*Figure 319: The **Modify Mesh Settings** dialog (**Advanced** tab). On the left, the dialog for triangles and tetraheda and to the right, the dialog for voxels.*

Suppression of Small Geometry Features

This option controls how small geometry features are meshed into segments, triangles, tetrahedra or voxels (where applicable).

Default

The mesh created with this option will not mesh (mesh over) very fine geometric details. To include very fine geometric details in the mesh, select the **Preserve all entities** check box.

Preserve all entities

This option is used to retain all the edges after meshing. This means that tiny edges will be preserved and will not get collapsed.

Size

This option creates a mesh with a specified minimum element size. If the geometry detail is smaller than this limit, the mesher will ignore this detail.

Ratio

This option creates a mesh where the ratio of the maximum element to the minimum element conforms to the specified ratio.

Fraction of voxel size (0,1)

This option limits how small voxels are allowed to become compared to their ideal size, where ideal size refers to the size determined by electromagnetic properties or the specified value. The position of gridlines is influenced by points of interests on the geometry. Points of interest that are closely spaced will result in unnecessarily small voxels.

Select **Manual setting** to specify a value between 0 and 1, where the value 1 relates to the ideal voxel size.

Mesh Quality

This option controls the quality of the mesh.

Mesh size growth rate

This option controls how quickly the mesh size changes. *Fast* allows an abrupt jump from small to large elements, while for *Slow*, each adjacent triangle will increase in size with less than twice the size of its previous neighbour.

Curved Geometry Approximation

These options control how curved geometry is approximated when creating the mesh.

Refinement factor

This option controls how fast the mesh will refine when it determines that the mesh does not adequately conform to the model. *Fine* allows smaller triangles to be used for small details but will increase the time to mesh the model.

Minimum element size

This option limits the size of the small triangles that are used to conform to the geometry, relative to the requested mesh size on that part.

Allow elongated triangles

This option allows the use of long, thin triangles and can lead to a reduction in the number of mesh elements (depending on the geometry that is meshed).

Curvilinear Mesh

Advanced mesh options are available to create a mesh using curvilinear triangles and curvilinear segments. A curvilinear triangle mesh allows you to use fewer higher order basis functions (HOBF) or RL-GO triangles.

Curvilinear Mesh Triangles

Auto
: This option allows the curvilinear mesh triangles to be used if curvilinear mesh triangles are likely to result in a more efficient solution using less memory.

Disabled
: This option creates flat triangular elements.

Enabled
: This option creates curvilinear mesh triangles (if supported by the solution method).

> **Note:** HOBF must be enabled for curvilinear meshing (except for windscreen reference elements and when using the RL-GO solution method).

Figure 320: A model with flat triangular mesh (on the left) and with curvilinear mesh and higher order basis functions enabled (to the right).

Curvilinear Mesh Segments

A curvilinear mesh segment is created using second order segments with three vertices.

Auto
: This option allows the curvilinear mesh segments to be used if curvilinear mesh segments will result in a more accurate solution with the selected solution method.

Disabled
: This option creates straight mesh segments.

Enabled
: This option creates curvilinear mesh segments (if supported by the solver method).

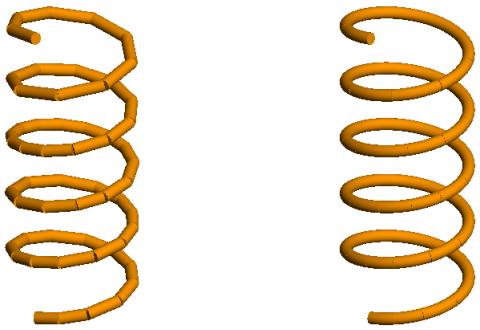


Figure 321: A helical wire meshed with straight segments (on the left) and with curvilinear mesh segments (to the right).

> **Note:** Curvilinear mesh segments not supported for windscreen solution elements.

Aspect Ratio

This option controls the ratio between the longest and shortest side lengths of a voxel. Specifying the aspect ratio adds additional grid lines to decrease the side with the greatest length.

Select **Manual setting** to specify a value between 1 and 100, where:

- the value 1 relates to voxels with equal side lengths.
- the value 100 relates to a 100:1 aspect ratio.

Growth Rate

The option limits the changes in size between adjacent voxels.

Select **Manual setting** to specify a value between 1 and 100, where a growth rate of unity implies no growth and will result in a uniform mesh.

> **Tip:** Use the default **Growth rate control** setting of 1.2.

Ensure Connectivity Through Wire Tracing

This option allows a thin face to be replaced by a wire to ensure connectivity in a voxel mesh.

Select the **Ensure connectivity through wire tracing** check box to replace a thin PEC face with a wire. The intrinsic wire radius is determined by the Solver.

When this option is not selected, sections of the model that should be electrically connected might not be connected in the voxel representation.

Ensure Continuous Region Representation

This option allows a thin dielectric region (with a thickness smaller than the voxel size) be represented in the voxel mesh.

When this option is not selected, thin dielectric regions will not be represented in the voxel mesh.

Insufficient Memory Protection

This option prevents a large model to mesh if there is not sufficient memory available. Clear the **Insufficient memory protection** check box to disable this feature.

Show Mesh Configuration Details

This option when selected shows the calculated meshing parameter values such as **Grade factor**, **Minimum to maximum size ratio**, **Maximum angle per element** and **Curvature minimum element size**.

2.30.6 Preventing Future Mesh Modification

Lock a part to prevent modification to the simulation mesh (and prevent the part from being edited).

Some parts of a model take long to mesh or for some reason, it may be required to ensure that a part is not remeshed (remeshing could result in a different mesh). Locking a part allows the mesh to be locked for modification. If the part does not have a simulation mesh, it is meshed once, but the mesh will not be remeshed again (until it is unlocked).

> **Note:** Locking a part to prevent mesh modification is not supported for a voxel mesh.

1. Select the geometry or mesh part in the model tree (**Construct** tab).
2. Lock the part. From the right-click context menu click **Lock/Unlock**.
   The mesh is now locked and will not be remeshed when the model is meshed.

2.30.7 Viewing the Mesh Information

After the geometry or model mesh was meshed, the quality of the mesh part (or model mesh) or simulation mesh can be examined.

1. Select the model (or part) in the model tree (**Construct** tab).
2. View the mesh information using one of the following workflows:
   - From the right-click context menu, click **Info**.
   - On the **Mesh** tab, in the **Meshing** group, click the **Info** icon.

The **Mesh information** dialog gives a summary of the mesh statistics such as the average edge length and the standard deviation of the edge lengths as well as a visual representation of the mesh edge lengths distribution. The data gives an indication of the quality of the mesh and how many edges are longer than the desired length.

You can also view the number of triangles (flat and curvilinear), tetrahedra, voxels and line segments (straight and curvilinear).

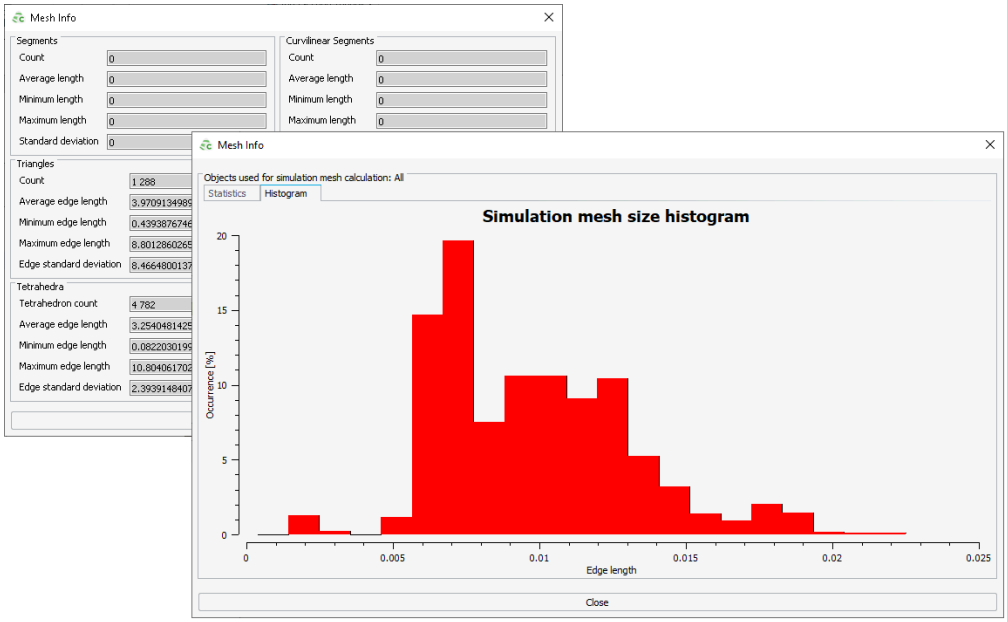


*Figure 322: The **Mesh Information** dialog.*

2.30.8 Mesh Refinement

When an accurate solution of the model requires a fine mesh, the mesh can be refined at specific areas of the model without simply meshing the entire model finer.

There are several mesh refinement options available when it is required to refine specific areas in the mesh. Although the mesh can be refined globally and this approach is attractive due to its simplicity, this will lead to an unnecessarily large number of mesh elements that in turn will increase the simulation time and resource requirements.

A more efficient approach is to only refine the mesh locally where a finer mesh is required. In CADFEKO, you have the following options to refine a mesh locally:

- Define multiple local mesh settings for a model, each one with a unique label. Apply the local mesh settings to root-level geometry and mesh parts in the model tree by referencing its label.
- Apply a local mesh size to a wire / edge, face or region.
- Use point refinement to define the local point and its radius where the mesh should be refined.
- Use polyline refinement to define a line and radius where the mesh should be refined along the line.
- Use adaptive mesh refinement in conjunction with an error estimate request to iterate and refine the mesh at areas with the largest error estimates.
- Use automatic mesh refinement (AMRFEKO).

Related concepts
Local Mesh Refinement
Related tasks
Applying Local Mesh Settings to a Part
Refining the Mesh Around a Point

Applying Local Mesh Settings to a Part

Define multiple local mesh settings, each one with a label. Apply the local mesh settings to a root-level geometry or mesh part by referencing its label.

Note: This feature replaces the functionality where you could select a root-level geometry or mesh part, set the mesh scope to selection and only mesh the selected part using the specified mesh settings.

1. Define a local mesh setting.
   a) On the **Mesh** tab, in the **Meshing** group, click the **Add Mesh Settings** icon.

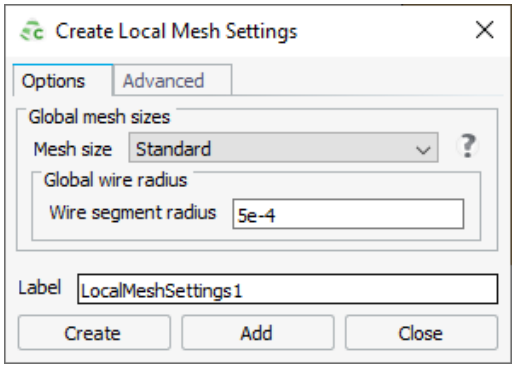


*Figure 323: The **Create Local Mesh Settings** dialog.*

   b) On the **Create Local Mesh Settings** dialog, specify the mesh sizes.
   c) In the **Label** field, enter a unique label for the local mesh settings.
   d) Click **Create** to create the mesh settings and to close the dialog.
2. Apply the local mesh settings to a part.
   a) In the model tree, double-click on a root level geometry or mesh part.
   b) On the **Modify ...** dialog, click the **Meshing** tab.

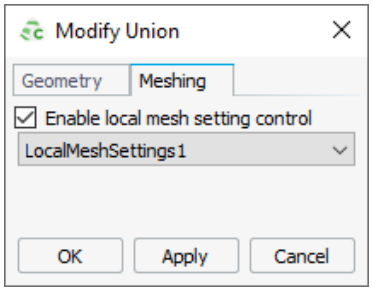


*Figure 324: The **Modify ...** dialog.*

c) Select the **Enable local mesh setting control** check box.
d) From the drop-down list, select the local mesh settings to be applied to the part.
e) Click **OK** to apply the local mesh settings and to close the dialog.

Local Mesh Refinement

A local mesh refinement can be specified on wires / edges, faces and regions to refine the mesh locally.

When a region that has a local mesh size is meshed into tetrahedra, the local mesh size is also applicable to the bounding faces. Likewise, setting a local mesh size on a face also affects the size on the bounding edges. If a finer mesh is specified on an edge of a face, then the triangles of that face will adhere to this length along the specific edge, even though the rest of the face may have much larger mesh sizes.

> **Note:** The local mesh size that takes precedence on an item is always the minimum of all applicable local mesh sizes.

If no local mesh size is specified on an item, the global mesh size[43] applies.

A local mesh refinement specified on an entity is indicated by the icon in the details tree.

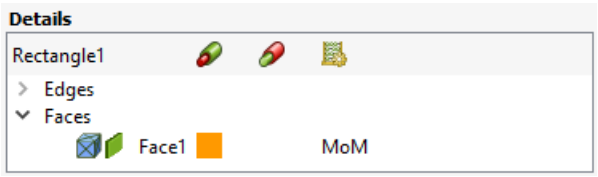


Figure 325: An example of face that has a local mesh size specified in the details tree.

Related concepts
Details Tree

Applying a Local Mesh Size to a Wire, Edge, Face or Region

A local mesh size can be set on a wire, edge, face or a region to influence the mesh size.

The steps described below to define a local mesh size are similar for wires, edges and regions.

1. In the model tree, select the relevant part.
2. In the details tree select the wire, edge, face or region where you want to apply a local mesh setting.

   > **Tip:** Multiple entities can be selected and edited simultaneously.

3. From the right-click context menu, click **Properties**.
4. On the **Modify Face** dialog, click the **Meshing** tab.

43. As specified on the **Create mesh** dialog.

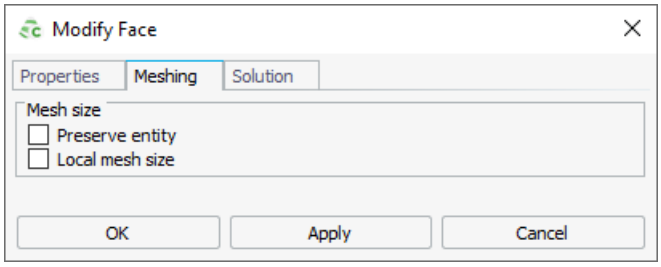


*Figure 326: The **Modify Face** dialog (**Meshing** tab).*

5. Select the **Preserve entity** check box to include entity in the mesh should it be very small.
6. Select the **Local mesh size** check box to enable the local mesh size for the selected item.
7. In the **Mesh size** field, enter a local mesh size.
8. Click **OK** to set the local mesh size and to close the dialog.
9. Remesh the model to view the local mesh refinement.

Tip: Define a local mesh size variable to simplify mesh convergence studies.

Refining the Mesh Adaptively Using AMRFEKO

Refine the mesh adaptively using AMRFEKO.

Note: Specify the AMRFEKO settings on the **Component Launch Options** dialog.

1. Open and save a model whose mesh will be refined adaptively.
2. On the **Solve/Run** tab, in the **Run/Launch** group, click the AMRFEKO icon.

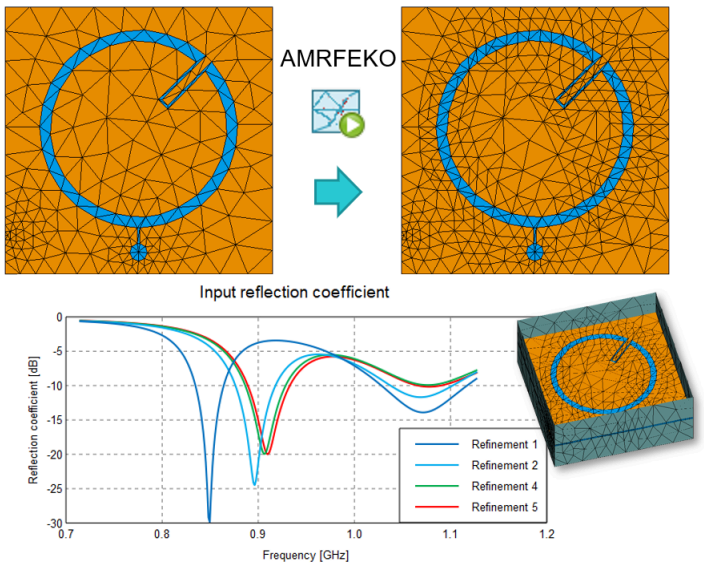


Figure 327: An example of a slot antenna whose mesh was refined using AMRFEKO.

For each adaptive mesh refinement iteration, the Solver is run. After the final adaptive mesh refinement is reached, the refined model is saved. If a `.cfx` file was provided, the refined model is saved as `<model>_refined.cfx`.

AMRFEKO can also be run on an EDITFEKO model (`.pre` file) or a `.fek` file. When there is no `.cfx` file present, the refined model is saved as `<model>_fek_refined.fek`.

Related reference

AMRFEKO Options

Command Line Arguments for Launching AMRFEKO

Refining the Mesh Adaptively Using Error Estimates

Error estimates can be used to automatically place mesh refinement rules (point refinement) in the model where the error is estimated to be large. Refining the model in the areas with the largest errors results in a model where the errors are comparable everywhere in the model and produces a model with increased accuracy without an excessive increase in mesh elements.

When a model has error estimates calculated, the error estimates can be used to define mesh refinement rules. The model is then solved again and new mesh refinement rules are added where the estimated error is large. This process is repeated until the model has sufficiently been refined. It is recommended that you perform mesh convergence tests to confirm that the model is sufficiently refined to produce the required level of accuracy.

The steps required to add additive mesh refinement points using error estimates follows:

1. Add an error estimate calculation request.
2. Specify the frequency or apply local mesh settings to allow the automatic mesh algorithm to calculate and mesh the model.
3. Save the model.

4. Run the Solver to obtain a solution.

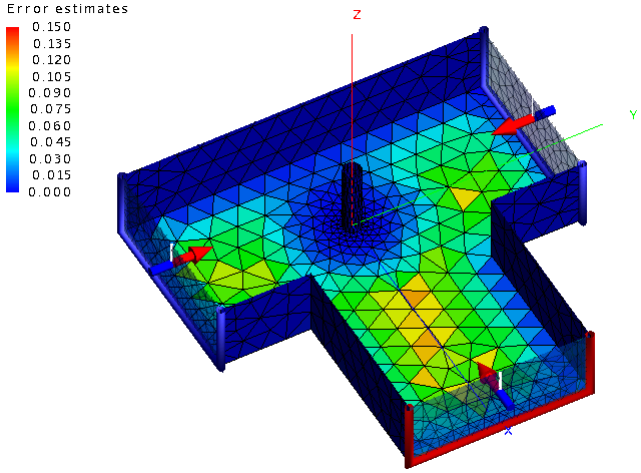


Figure 328: The result of the error estimation request viewed in POSTFEKO (with a cutplane) indicating the areas with the highest errors in red.

Add an adaptive mesh refinement rule.

5. On the **Mesh** tab, in the **Refinement Rules** group, click the **Error Estimate Refinement** icon.

   The error estimate data (in the `.bof` file) is then matched with the mesh elements (for example, triangle, segments) information (from the `.fek` file) to calculate the areas where the errors are estimated to be the highest.

   The actual point refinement rule is added to the model tree (Configuration tab) and is indicated by the icon.

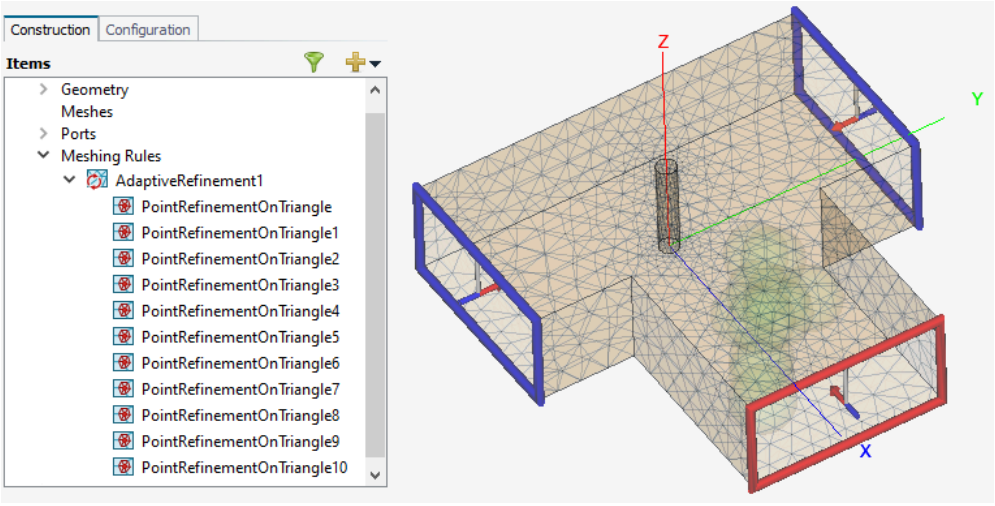


Figure 329: The mesh is refined in the areas where errors are estimated to be the highest. The transparent green spheres are a display setting and indicate areas where the mesh refinement is applied. Opacity was set to 20% to highlight the green spheres.

6. [Optional] Multiple iterations of adaptive mesh refinement can be applied by repeating Step 4 to Step 5 for each iteration.
   For each iteration, an adaptive mesh refinement rule is added to the model tree (**Configuration** tab).

[Optional] Hide the display of mesh rules in the 3D view.

7. On the **3D View** context tab, on the **Display Options** tab, in the **Entity Display** group, click the **Meshing Rules** icon.

Related tasks
Requesting an Error Estimation

Refining the Mesh Around a Point

Define a mesh refinement rule around a specified point. Mesh elements in the vicinity of the point are refined. Mesh refinement around a point is used when only a subset of a face, edge or wire needs to be refined.

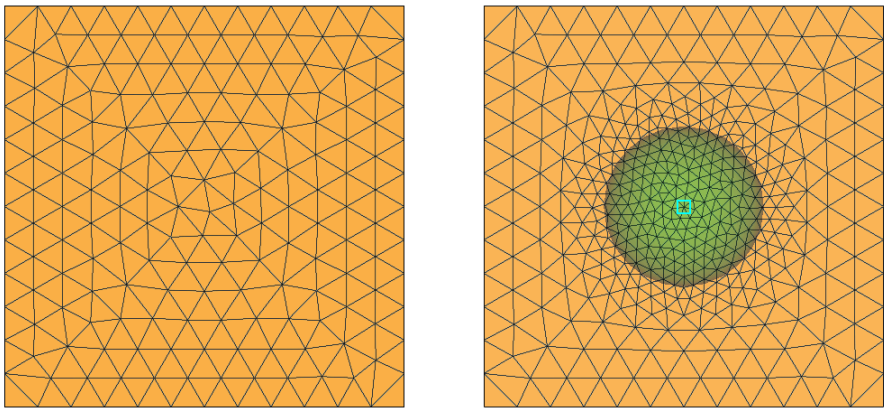


*Figure 330: A plate with no point mesh refinement (on the left) and with point mesh refinement (to the right). The transparent green sphere is the preview when the **Create Point Refinement** dialog is open and indicates the area where the mesh refinement is specified.*

1. On the **Mesh** tab, in the **Refinement Rules** group, click the **Point Refinement** icon.
2. Under **Position**, specify the origin of the point where the mesh is to be refined.
3. In the **Radius** field, enter a value for the radius to specify the mesh area that is to be refined.
4. In the **Mesh size** field, enter a value for the mesh element length.
5. Enter a unique label for the point refinement.
6. Click **Create** to create the point refinement rule and to close the dialog.
   The mesh refinement rule is added to the model tree, on the Configuration tab.

[Optional] Hide the display of mesh rules in the 3D view.

7. On the **3D View** context tab, on the **Display Options** tab, in the **Entity Display** group, click the **Meshing Rules** icon.

Refining the Mesh Along a Polyline

Define a mesh refinement rule along a specified polyline. Mesh elements in the vicinity of this polyline are refined. Polyline mesh refinement is often used to refine the mesh under a cable, wire or a transmission line.

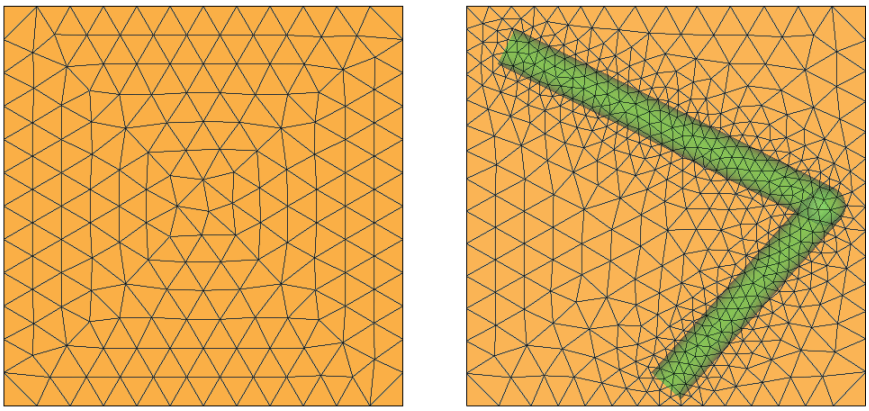


*Figure 331: A plate with no polyline mesh refinement (on the left) and with polyline mesh refinement (to the right). The transparent green sphere is the preview when the **Create Polyline Refinement** dialog is open and indicates the area where the mesh refinement is specified.*

1. On the **Mesh** tab, in the **Refinement Rules** group, click the **Polyline** icon.
2. Specify the polyline.
   - To specify the corner points manually, specify the corner points in the table.
   - To import the corner points from file, click **Import points**.
3. In the **Radius** field, enter a value for the radius to specify the mesh area that is to be refined.
4. In the **Mesh size** field, enter a value for the mesh element length.
5. Enter a unique label for the point refinement.
6. Click **Create** to create the polyline refinement rule and to close the dialog.
   The mesh refinement rule is added to the model tree, on the **Configuration** tab.

[Optional] Hide the display of mesh rules in the 3D view.

7. On the **3D View** context tab, on the **Display Options** tab, in the **Entity Display** group, click the **Meshing Rules** icon.

2.30.9 Mesh Editing

It is not possible to edit a simulation mesh directly, but you can unlink the simulation mesh and edit the mesh part as though it is an imported mesh. A mesh part can also be replaced with a different mesh.

After the simulation mesh was unlinked, you can edit the mesh and remesh to obtain a new simulation mesh or use the model mesh as the simulation mesh directly.

Related concepts
Model Mesh / Simulation Mesh
Related tasks
Unlinking a Mesh
Replacing a Mesh

Unlinking a Mesh

When a mesh is unlinked, the simulation mesh is converted to a separate mesh part that can be edited.

1. In the model tree, select the geometry or mesh part that has a corresponding simulation mesh.
2. Unlink the mesh using one of the following workflows:
   - On the **Mesh** tab, in the **Simulation Mesh** group, click the **Unlink Mesh** icon.
   - From the right-click context menu, select **Unlink Mesh**.
3. On the **Unlink Mesh** dialog, two options are available:

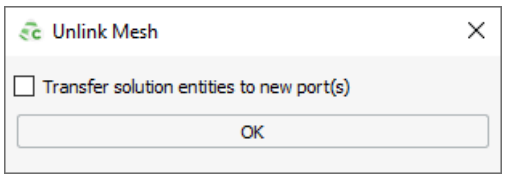


*Figure 332: The **Unlink Mesh** dialog.*

   - ***Transfer solution entities to new port(s)*** *check box cleared*
     Unlink the mesh and create a separate mesh part. Mesh ports are created but the sources, loads and solution entities are not transferred.
   - ***Transfer solution entities to new port(s)*** *check box selected*
     Unlink the mesh and create a separate mesh part. Mesh ports are created but the sources, loads and solution entities are transferred to the new ports on the separate mesh part. S-parameter configurations are updated accordingly.

   When the simulation mesh is unlinked, the geometry ports connected to the selected geometry remain, but equivalent mesh ports are created with new labels. For example, if a geometry port has the label “Port1”, the mesh port is labelled “Port1_1”.
4. Click **OK** to unlink the mesh and to close the dialog.

Replacing a Mesh

Replace mesh elements or update the mesh elements in a mesh part while keeping all the settings that have been applied to the wires, faces and regions.

A typical workflow is to create a model with configurations, ports and mesh elements with applied solution settings. The mesh is then exported and modified using third-party tools and then re-imported

into CADFEKO. This workflow eliminates the need to reapply solution settings and ports to the mesh elements, provided that the mesh labels remain largely unchanged.

1. Ensure you have both the old and new mesh available in your model.
2. In the model tree, select the mesh part to be replaced.
3. Replace the mesh using one of the following workflows:
   - From the right-click context menu, click **Replace With**.
   - On the **Mesh** tab, in the **Replace** group, click the  **Replace With** icon.
4. In the model tree, select the replacement mesh part.

   The old mesh is replaced and removed from the model.

When a mesh is replaced, solution settings and ports applied to the old mesh are transferred to the new mesh. Mesh properties of mesh labels that are new and only present in the new mesh, are set to the default mesh properties. Default mesh properties include faces set to PEC, wires set to PEC and the front and back medium of a face set to free space.

Mesh labels that were in the old mesh but no longer present in the new mesh, will not affect the mesh, but could affect the solution and request items that use labels. For example, ports and requests with scope options (far fields, near fields, error estimates and currents).

2.30.10 Batch Meshing

A stand-alone batch meshing tool can be called from the command line to mesh a model and modify variable values in a CADFEKO model file, without launching the CADFEKO GUI.

During the optimisation process, OPTFEKO calls the batch meshing tool to mesh the model for each optimisation run.

Launch the CADFEKO batch mesher using the command:

```
cadfeko_batch <filename> [options]
```

filename	The file name of an existing CADFEKO model (with or without the `.cfx` file extension). The path may be included in the file name.

The command line options are:

`--version`	Output only the version information to the command line and then terminate. No file name is required to use this option.
`-#var=value`	Allows variables to be assigned new values before re-evaluation and meshing. Multiple variables may be included. For example, to set variables "*a*" and "*b*" to 1, the options should contain ...-#*a*=1 -#*b*=1...).
`--run-from-gui`	This uses a special execution mode for the GUI. In this mode, additional information regarding the progress of each phase of the model re-evaluation and meshing is included in the screen output.

--cable-seed	This option rearranges the cables in a cable bundle to a new random location.

After the model is re-evaluated and meshed, the modified CADFEKO model replaces the existing `.cfx` file as well as the `.cfm`, `.pre`, `.opt` and `.pfg` files.

If new variable values cause an error during re-evaluation or meshing, the batch meshing is aborted and an error reported. If any suspect entities are found in the model after re-evaluation, the meshing are completed, the new model is created, but an error will be reported. The error is reported for all suspect items, even if they were not introduced due to changes made by the batch-mesher.

If the solution configuration is deactivated in the CADFEKO model, or if the `.pre` file has been edited outside of CADFEKO, then the `.pre` file is not overwritten.

Related tasks
Defining a Cable Bundle

Using Batch Meshing to Mesh a Model

A simple example is given to show how to mesh an existing model or modify multiple variables and remeshing the model.

As an example, Example 1 of the Feko Example Guide will be meshed using the batch meshing tool (`Dipole_Example.cfx`).

1. Open the Feko terminal.
2. [Optional] Change the directory to where the file is located.

   Note: If the directory is not changed to where the file is located, the path can be included in the file name.

3. Launch the batch meshing tool and mesh (without modifying the values of any variables).

   ```
   cadfeko_batch Dipole_Example.cfx
   ```

4. Launch the batch meshing tool, modify the variables *h* and *radius*, re-evaluate and remesh the model.

   ```
   cadfeko_batch Dipole_Example.cfx -#h=lambda/4 -#radius=1.8e-3
   ```

2.31 Domain Connectivity

Domain connectivity approach allows meshes of specific parts to be treated as if "connected" during MoM and MoM/MLFMM solutions in places where the borders of the meshes are close together, even if the mesh vertices on those borders are not coincident.

Related concepts
Discontinuous Mesh and Geometry Parts
Workflow for Connecting Discontinuous Mesh and Geometry Parts
Related tasks
Defining Domain Connectivity
Related reference
Supported Solution Method and Technique Combinations

2.31.1 Defining Domain Connectivity

Specify faces that are close together to be treated as "connected".

Note: Only supported for MoM and MoM/MLFMM solutions.

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **Domain Connectivity** icon.

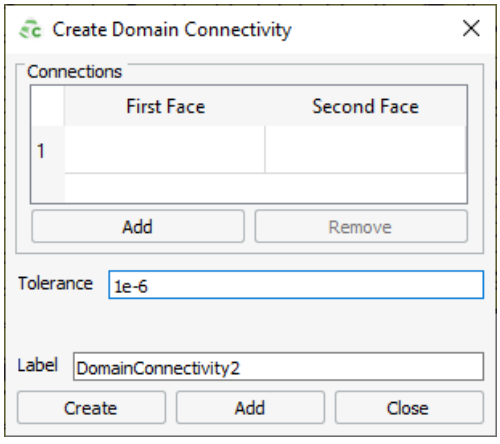


*Figure 333: The **Create Domain Connectivity** dialog.*

2. Click the first row under **First face** to activate the field (indicated in blue) and click on the face either in the 3D view or in the details tree.
3. Click the first row under **Second face** to activate the field and click on the face either in the 3D view or details tree.
4. Click **Add** if more "connections" are required.
5. In the **Tolereance** field, specify for each connection point, a tolerance distance to distinguish between regions, where a gap in the model is desired (or not desired).

6. In the **Label** field, enter a name for the domain connectivity.
7. Click **Create** to define the domain connectivity and to close the dialog.

Related concepts
Domain Connectivity
Related tasks
Viewing Domain Connectivity

2.32 Working with CADFEKO Models in EDITFEKO

A CADFEKO `.cfm` file can be imported into EDITFEKO to make use of more advanced features available in EDITFEKO and to directly edit the `.pre` file for more flexible solution configurations.

2.32.1 Modification of the Model in EDITFEKO

Modification of CADFEKO models in EDITFEKO is considered to be an advanced workflow with important considerations.

CADFEKO writes a `.pre` file when the model is saved. The `.pre` file can be modified manually with EDITFEKO. In EDITFEKO changes can be made such as adding custom frequency loops. When the model is saved again in CADFEKO, the modified `.pre` file will trigger a prompt in CADFEKO asking whether a copy of the `.pre` file (modified in EDITFEKO) should be saved. Clicking **No** will overwrite the changes made in EDITFEKO. Clicking **Yes** will save a copy of the `.pre` file. The copy uses the naming convention, `<model_name>_modified_n.pre`, where the part `<model_name>` is the original filename in CADFEKO and the part `_modified_n` is an extension chosen to ensure that no existing file is overwritten. Repeated modifications are retained by the counter, `n`.

2.32.2 Units

When working with a CADFEKO model in EDITFEKO, the units are considered to be in metres.

Use an SF card in the geometry section of the `.pre` file to specify the units. For example, if the model was constructed in millimetres, an SF card with a 0.001 scale factor should be added to the `.pre` file.

> **Tip:** Place the SF card at the beginning of the `.pre` file.

2.32.3 Reference Elements

In EDITFEKO, properties can be set on specific elements using their full labels.

Segments have the label of the edge (typically called *Wire*), triangles that of the face (typically *Face*) and tetrahedra that of the dielectric region (typically *Region*). These labels can be modified on the geometry or the mesh elements.

Since setting sources or loads on wire segments require unique labels, CADFEKO exports the port segments with unique labels. These labels are created by appending the port name to the wire label.

For example, if Port1 is located on the centre segment of Line1.Wire1, this segment will be written with the label *Line1.Wire1.WirePort1* while the remaining segments will have the label *Line1.Wire1*.

For vertex ports, the associated segment is the shorter segment connected to the vertex. Use POSTFEKO to check which segment was relabelled.

2.32.4 Variables and Named Points in EDITFEKO

When exporting a `.cfm` file, CADFEKO evaluates all named points and variables and writes their numerical values to the file.

If requested in the IN card settings, these variables and named points are then imported by PREFEKO and can be referenced in the `.pre` file at any point after the IN card.

2.32.5 Media

Media can still be defined and applied to regions or mesh elements in CADFEKO, but in the `.pre` file, the DI card must reference the name of the medium specified in CADFEKO.

2.33 Validating the CADFEKO Model

During the design process, the development of a model can introduce a range of issues that can lead to a non-simulation-ready model. Use the validation toolset to verify that the model is simulation-ready or to search, detect and flag discrepancies.

Use the validation toolset to verify the following:

- Fix the settings of an item in the model that is unresolved or invalid (suspect item).
- Verify that the windscreen is defined correctly by viewing the thickness of the individual layers.
- Verify that the mesh is connected.
- Use display options to colour regions and faces according to their media.
- Use display options to highlight the relevant geometry with a specified solution method.
- Search for clashing geometry.
- Search for distorted, intersecting and oversized mesh elements.
- Use cutplanes to cut through a model to view inside the model.

2.33.1 Suspect Items

An item is marked suspect when changes in the model result in the settings of an item becoming unresolved or invalid.

A suspect item is indicated by a ⚠ icon in the model tree or details tree. Move the mouse cursor over the ⚠ icon to view the tooltip and the reason why it is marked suspect.

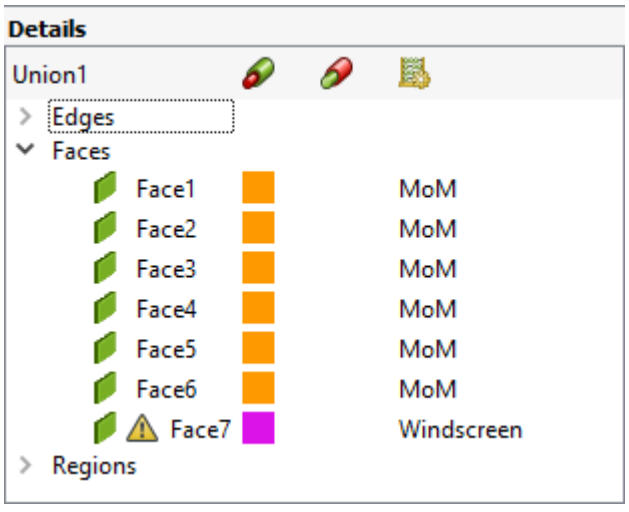


Figure 334: An example of a suspect item and its tooltip in the model tree.

Examples of situations where items can become suspect;

- If a lossy conducting surface is set on a face bordered by free space and one of the bordering regions is set to PEC, the unsupported lossy conducting surface is removed. The face is marked “suspect” and its medium displayed as PEC in the details tree.

- If a port becomes invalid due to a change in the model, the port is marked “suspect”.

Note: Resolve all suspect items before launching the Solver or OPTFEKO. The loss of properties on the model geometry may change the electromagnetic problem description and impact the computed results.

Tip: Set the correct properties on the item and remove the suspect icon. From the right-click context menu select **Set Not Suspect**.

Related concepts
Edges and Wires (Geometry)
Faces (Geometry)
Regions (Geometry)

2.33.2 Displaying Windscreen Thickness

When defining a windscreen, the layer thickness is not displayed by default. Enable the windscreen layer thickness to visually verify that the model is correct.

On the **3D View** context tab, on the **Display Options** tab, in the **Style** group, click the **Windscreen Layers** icon.

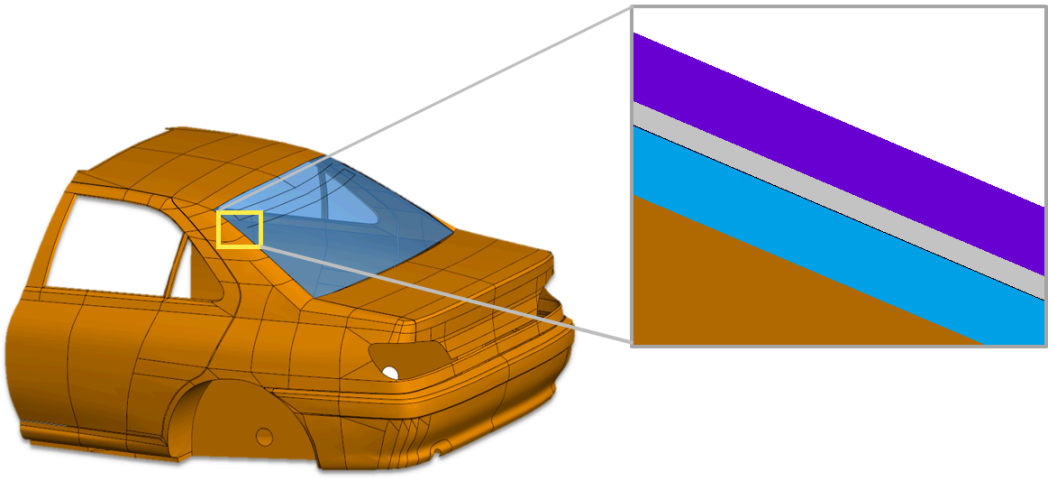


Figure 335: An example of a windscreen showing the individual layer thickness.

2.33.3 Cutplanes

A cutplane is a display option that creates a plane at a designated location that cuts through an object to create a sectional view. Create multiple cutplanes to create a sectional view that exposes inner details that would otherwise not be visible from outside the model.

On the **3D View** context tab, on the **Display Options** tab, in the **Cutplanes** group, click the **Create** icon.

Three (default) cutplanes coincident with the main axes are accessible from the model tree under the Cutplanes group.

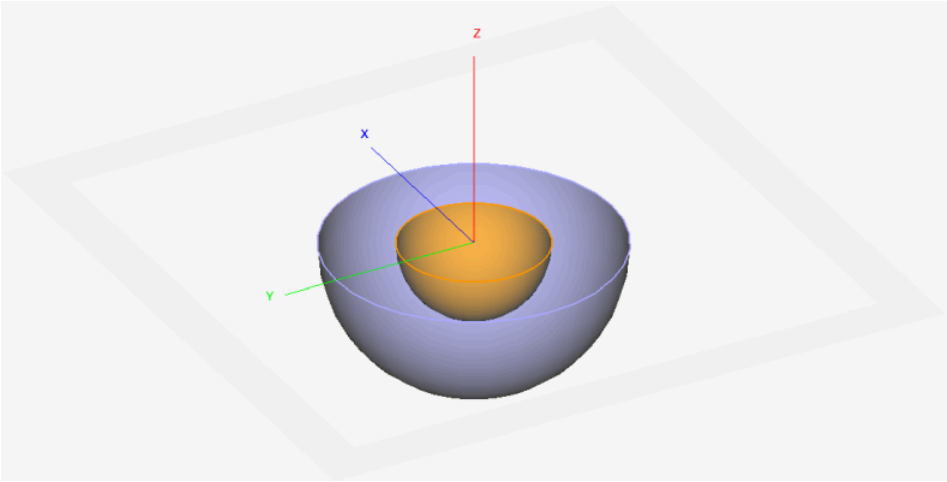


Figure 336: A simple example of a cutplane preview. The cutplane cuts through a larger sphere revealing a smaller sphere inside.

To add more cutplanes, click **Create**. Each cutplane is listed under the Cutplanes group.

The operation of the cutplane can be reversed, hiding the visible region and showing the invisible region. On the **Create/Modify Cutplane** dialog, select **Flipped**, or in the tree, from the right-click context menu for the cutplane, click **Flip cutplane**.

To activate/deactivate a cutplane, click the icon for the cutplane. To modify, select the label.

Filter Cutplanes

When more than one 3D view is used, the already active cutplanes can be modified on a per-view basis using the **Filter** tool.

On the **3D View** context tab, on the **Display Options** tab, in the **Cutplanes** group, click the **Filter** icon.

On the **Filter Cutplanes** dialog, select the checkbox(es) for the cutplane(s) to be disabled.

> **Note:** The filter cutplanes tool is applied to the selected 3D view.

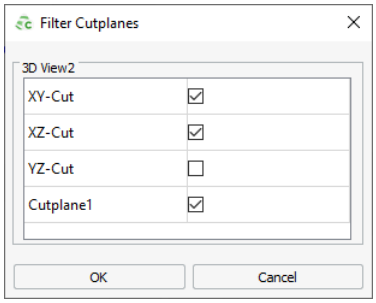


Figure 337: The Filter Cutplanes dialog.

2.33.4 Colour Display Options

A number of display options are available to colour regions and faces according to their media.

On the **3D View** context tab, on the **Display Options** tab, in the **Style** group, click the **Colour** icon.

Table 16: Colour display options.

Icon	Icon text	Description
	Element normal	All parts display with the same colour. The two sides of faces are coloured differently to indicate the normal direction of the faces. • The normal side of each element is coloured green. • The reverse side of each element is coloured red.
	Region medium	Regions are coloured according their assigned media. Surface mesh elements are coloured on each side according to the medium on that side of the face. For example, when viewing the mesh of a dielectric/metallic object, the entire object has the colour of free space when viewed from the outside. When viewed from the inside (utilising a cutplane or after hiding faces) the colour is consistent with the dielectric/metallic medium of the inside region. When viewing the geometry, regions are displayed using the colour of the internal medium (whether viewed from outside or inside the region). If the display of the segment radii and coatings are activated on wire mesh elements, these are coloured according to the core medium or the layered medium (coating) for that wire.
	Face medium	The faces are displayed according to the medium of each face.

Icon	Icon text	Description
		When viewing the mesh, the display of segment radii is automatically activated for wire elements in the mesh and these are coloured according to the core medium. (The segment radii display may be manually deactivated if required, in which case no specific colouring will be shown for wire elements in the mesh.)
	Face normal medium	The faces are displayed according to the material colour on the two sides of the face. For example, an object in free space will have the colour of free space (red by default) on the outside of the object.

2.33.5 Mesh Connectivity

After applying union or stitching operations, you can verify that all the intended edges are connected. A face with unbounded edges could indicate an unconnected mesh.

On the **3D View** context tab, on the **Display Options** tab, in the **Style** group, click the **Connectivity** icon.

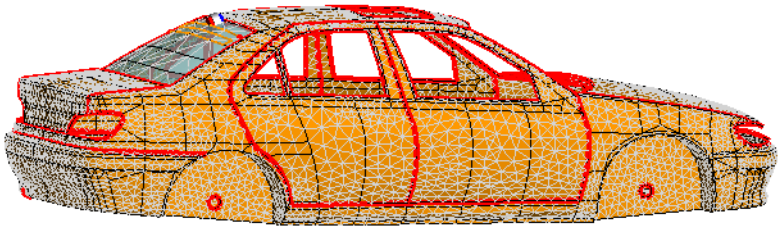


Figure 338: An example showing mesh connectivity. Faces with unbounded edges are shown in red.

2.33.6 Geometry and Mesh Consistency Checks

Searching for Clashing Geometry

Parts clash if there is contact between the parts without a mesh connection or if one is completely inside another. These disconnected mesh elements need to be either connected or removed before running a simulation to obtain an accurate result.

1. Select the model or geometry part either in the model tree or 3D view.
2. On the **Mesh** tab, in the **Find** group, click the **Clashing Geometry** icon.

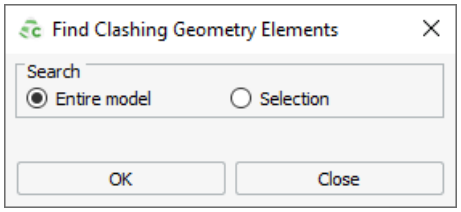


*Figure 339: The **Find Clashing Geometry Elements** dialog.*

3. Specify the parts to be searched for clashing geometry elements.
   - To search the full model, under **Search**, click **Entire model**.
   - To search only the selected part of the model, under **Search**, click **Selection**.
4. Click **OK** to search for clashing geometry elements and to close the dialog.

> **Tip:** Create a union where a part is contained in another.

The result of the search is displayed in the Model Status and on the **Message Details** dialog. Any parts containing clashing geometry are selected in the model tree and in the 3D view. A hyperlink to the part containing the clashing geometry is also given in the Model Status and on the **Message Details** dialog.

Searching for Distorted Mesh Elements

A distorted mesh element is a distorted (high aspect ratio) triangle. Distorted mesh elements can result in decreased accuracy of the results and could lead to poor convergence for iterative solvers.

The Solver does not directly search for distorted mesh elements, but the consequence of distorted mesh elements is that the condition number for the method of moments (MoM) matrix increases.

> **Note:** The Solver will give a warning or error if the condition number becomes too high. The condition number can also be too high for very low frequency problems.

Distorted mesh elements are specified in terms of the minimum internal angle. In an ideal mesh, all internal angles are 60°, but this rarely possible. If any of the three angles in a mesh element are very small, the element is considered a sliver element.

> **Tip:** Remove sliver elements by deleting mesh vertices or redundant geometry points.

1. Select the model mesh or mesh part either in the model tree or 3D view.
2. On the **Mesh** tab, in the **Find** group, click the **Distorted Elements** icon.

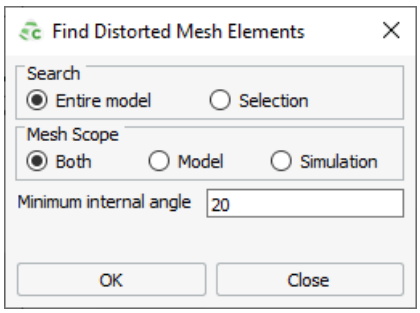


*Figure 340: The **Find Distorted Mesh Elements** dialog.*

3. Specify the parts to be searched for distorted mesh elements.
   - To search the full model, under **Search**, click **Entire model**.
   - To search only the selected part of the model, under **Search**, click **Selection**.
4. Specify the mesh parts to be searched for distorted mesh elements.
   - To search only the simulation mesh, under **Mesh Scope**, click **Simulation**.
   - To search only the model mesh, under **Mesh Scope**, click **Model**.
   - To search both the model and simulation meshes, under **Mesh Scope**, click **Both**.
5. In the **Minimum internal angle** field, enter a value for the minimum internal angle of a triangle. Any internal angles found to be smaller than the minimum angle will be listed.
6. Click **OK** to search for distorted mesh elements and to close the dialog.

The result of the search is displayed in the Model Status and on the **Message Details** dialog. A hyperlink to the mesh part containing the distorted mesh elements is also given in the Model Status and on the **Message Details** dialog.

Searching for Intersecting Mesh Elements

Imported meshes often contain intersecting mesh elements. These intersecting mesh elements need to be either repaired or removed to obtain accurate results.

Intersecting mesh elements can overlap (entirely or partially) or intersect other mesh elements, while not electrically connected at the point of intersection.

1. Select the model or geometry part either in the model tree or 3D view.
2. On the **Mesh** tab, in the **Find** group, click the **Intersecting Triangles** icon.

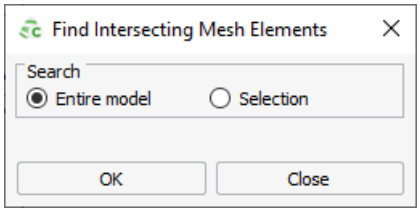


*Figure 341: The **Find Intersecting Mesh Elements** dialog.*

3. Specify the parts to be searched for intersecting mesh elements.
   - To search the full model, under **Search**, click **Entire model**.
   - To search only the selected part of the model, under **Search**, click **Selection**.
4. Click **OK** to search for intersecting mesh elements and to close the dialog.

The result of the search is displayed in the Model Status and on the **Message Details** dialog. Any parts containing intersecting mesh elements are selected in the model tree and in the 3D view. A hyperlink to the part containing the intersecting mesh elements is also given in the Model Status and on the **Message Details** dialog.

Searching for Oversized Mesh Elements

An oversized mesh element is a triangle with an edge length larger that the specified maximum edge length. Oversized mesh elements can lead to reduced accuracy in the results.

1. Select the model or geometry part either in the model tree or 3D view.
2. On the **Mesh** tab, in the **Find** group, click the **Oversized Elements** icon.

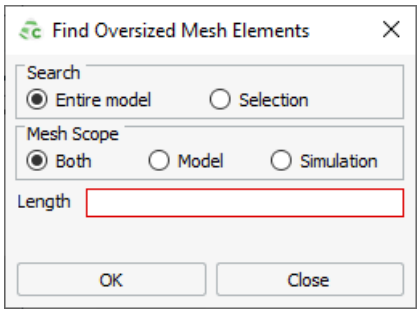


*Figure 342: The **Find Oversized Mesh Elements** dialog.*

3. Specify the parts to be searched for oversized mesh elements.
   - To search the full model, under **Search**, click **Entire model**.
   - To search only the selected part of the model, under **Search**, click **Selection**.
4. Specify the mesh parts to be searched for oversized mesh elements.
   - To search only the simulation mesh, under **Mesh Scope**, click **Simulation**.
   - To search only the model mesh, under **Mesh Scope**, click **Model**.
   - To search both the model and simulation meshes, under **Mesh Scope**, click **Both**.
5. In the **Length** field, enter a value that is taken as the upper limit for the triangle edge length. Any triangle edge length longer than this length will be marked as oversized.
6. Click **OK** to search for oversized mesh elements and to close the dialog.

The result of the search is displayed in the Model Status and on the **Message Details** dialog.. A hyperlink to the part containing the oversized mesh elements is also given in the Model Status and on the **Message Details** dialog.

2.34 Solver Settings

The default solver used in Feko is the method of moments (MoM) - surface equivalence principle (SEP). A solver is specified per model, per face or per region, and depends on the solver in question.

Table 17: The available solvers in Feko and where these solvers are specified.

Solvers		Per Model	Per Face	Per Region
Full-wave	MoM (SEP)	**Default solver**		✓
	MoM (VEP)			✓
	MLFMM	✓		
	FEM			✓
	FDTD	✓		
	ACA	✓		
High frequency	PO & LE-PO		✓	
	RL-GO		✓	
	UTD & faceted UTD		✓	

A number of advanced settings are available for each solver, but it is recommended to use the default settings. The incorrect application of these advanced settings may result in poor result accuracy or inefficient calculations.

Related concepts
Which Solution Methods Support GPU Acceleration?
Related reference
Media and Supported Solution Methods
Coatings and Supported Solution Methods
Supported Solution Method and Technique Combinations

2.34.1 Defining Symmetry in the Model

Define and exploit the symmetry in the model (where applicable).

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **Symmetry** icon.

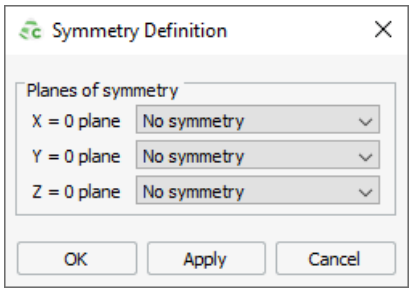


*Figure 343: The **Symmetry Definition** dialog.*

2. Under **Planes of symmetry**, select one of the following for the relevant planes:
   - No symmetry
   - Geometric symmetry
   - Electric symmetry
   - Magnetic symmetry
3. Click **OK** to set the symmetry and to close the dialog.

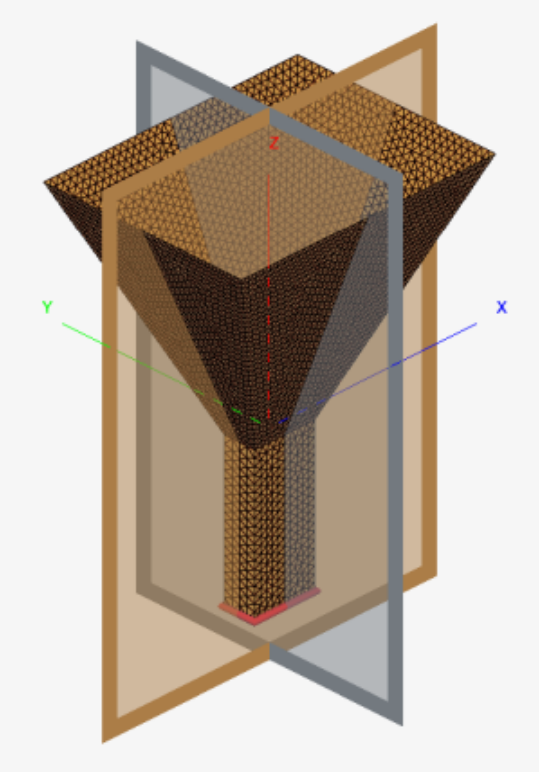


Figure 344: Example of a horn antenna with magnetic symmetry (in grey) defined at the X=0 plane and electric symmetry (in brown) defined at the Y=0 plane.

4. [Optional] Hide the display of the symmetry planes in the 3D view. On the **3D View** context tab, on the **Display Options** tab, in the **Solver Display** group, click the **Symmetry** icon.

2.34.2 General Solver Settings

General solver settings are available that relate to geometry tests and data storage precision.

On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon.

Geometry Tests

Activate normal geometry checking

This option allows geometry elements in the model to be analysed for typical user errors. These errors could be due to geometry parts that have not been unioned, or poor meshing such as wrong element sizes or meshing connection issues.

The following checks are performed:

- Verify that triangle elements on connecting surfaces have identical edge lengths.
- Verify that connection points coincide.
- Verify appropriate element sizes.
- Verify appropriate segment length to radius ratios.

Activate mesh element size checking
This option activates the verification of the mesh size in relation to the frequency.

*Export to the Feko *.out file*
This option allows the geometry data of the mesh elements to be written to the `.out` file.

Data Storage Precision

Single precision
This option sets certain memory-critical arrays to be stored in single precision. Single precision is the recommended and the default option.

Double precision
This option sets certain memory-critical arrays to be stored in double precision. Double precision is to be used when an error or warning message is displayed by the Solver suggesting that double precision be used. This could happen for example at very low frequencies where increased accuracy is required.

Thermal Analysis

Export files for thermal analysis (.epl, *.nas, *.map)*
This option allows the export of files for thermal analysis. The EM losses are exported to the element power loss (`.epl`) file and the geometry info is exported to a NASTRAN (`.nas`) file and label mapping (`.map`) file.

2.34.3 Advanced Solver Settings

Advanced solver settings are available to reduce the memory footprint or speed up a solution for specific types of models.

On the **Solve/Run** tab, in the **Solution Settings** group, click the  **Solver Settings** icon. On the **Solver settings** dialog, click the **Advanced** tab.

Factorisation for Parallel Execution

This option allows you to select between using standard full-rank factorisation or block low-rank (BLR) factorisation when using the parallel solver. Changing the factorisation method can reduce the memory footprint of the sparse LU-based preconditioners in some models where the solution methods are MLFMM or FEM.

Default
This option applies the predefined factorisation type adopted by the Solver.

Auto
This option applies automatically the optimum factorisation type based on the model.

Use standard full-rank factorisation
This option applies the standard full-rank factorisation.

Use block low-rank (BLR) factorisation
This option applies the block low-rank (BLR) factorisation.

> **Tip:** In most cases default gives the best performance.

Compression for Looped Plane Wave Sources

This option is an accelerated method that can be used to speed up the solution for a model that contains a plane wave source that loops over multiple directions (for example, when calculating RCS).

Related concepts

Preconditioners for MLFMM

Preconditioners for FEM

2.34.4 Store and Reuse Solution Files

For large models, the runtime can be reduced if the solution coefficients are saved during the solution phase and re-used in a subsequent solution.

> **Note:** For smaller models (where the run time is short), storage of the solution coefficients is typically not required. Storage of solution coefficients creates large files for models with many mesh elements.

On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon.

Save/read matrix elements
: This option allows you to save or read from a `.mat` file. The .mat file is used to store the matrix elements of a linear equation system (MoM models only.)

Save/read LU decomposed matrix
: This option allows you to save or read the `.lud` file. The `.lud` file is used to store the elements of the LU-decomposed MoM matrix.

Save/read currents
: This option allows you to save or read the `.str` file. The `.str` file is used by default to allow fast solutions in cases where only the output requests are modified (without changing the rest of the model).

Save/read cable per-unit-length parameters
: This option allows you to read or write a `.pul` file. The `.pul` files are used by default to allow the saving of cable per-unit-length parameters between frequency runs to allow fast solving of cable harnesses.

Store convergence data (.cgm file) for iterative solvers*
: This option allows the residue of the iterative solutions to be written to a `.cgm` file. Use this option to inspect convergence behaviour.

> **Note:** The data is saved but not re-used.

Note:

The saving of `.mat`, `.lud`, `.cgm` and `.pul` (for parallel runs) files is only possible if the model directory is on a shared location accessible by all processes.

2.34.5 Method of Moments (MoM)

The MoM is a full wave solution of Maxwell's integral equations in the frequency domain.

Related concepts
Which Solution Methods Support GPU Acceleration?
Related reference
Supported Solution Method and Technique Combinations

Surface Equivalence Principle (SEP)

The default solver in Feko is the method of moments (MoM) using surface equivalence principle (SEP).

The SEP introduces equivalent electric and magnetic currents on the surface of a closed dielectric body. The surface of such bodies can be arbitrarily shaped and is discretised using triangles.

Volume Equivalence Principle (VEP)

Volume equivalence principle (VEP) is an extension to the method of moments (MoM) for the modelling of dielectric bodies. The regions of such bodies can be arbitrarily shaped and are discretised into tetrahedra.

Solving a Model with VEP

To solve a model with the volume equivalence principle (VEP), you must activate VEP for each relevant region.

1. Select the region (or regions) in the 3D view or in the details tree that you want to solve with VEP.
2. In the details tree, from the right-click context menu, select **Properties**.
3. On the **Modify Region** dialog, click the **Solution** tab.
4. Under **Solution method**, from the drop-down list, select **MoM/MLFMM with volume equivalence principle (VEP)**.

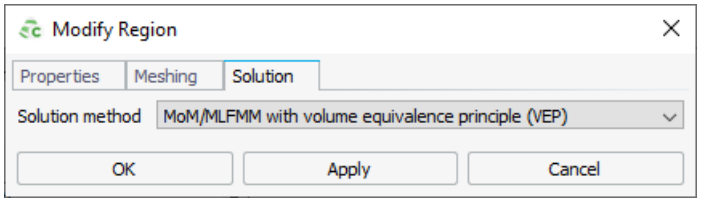


*Figure 345: The **Modify Region** dialog (**Solution** tab).*

5. Click **OK** to save the region properties and to close the dialog.

Higher Order Basis Functions (HOBF)

Higher order basis functions (HOBF) use higher order polynomial basis functions to model the currents on any particular mesh element.

HOBF is supported by the following solution methods:

- Method of moments (MoM) (including hybridisation with UTD and RL-GO)
- Multilevel fast multipole method (MLFMM)

Using HOBF allows the geometry to be meshed with larger triangles while obtaining the same solution accuracy. These larger and coarser mesh elements reduce the total number of mesh elements. In most cases the total unknowns are reduced. This leads to a reduction in solution time and memory.

Feko uses hierarchical basis functions to increase the basis function order of any triangle as required. Small geometric details of a model will be meshed with electrically small mesh elements, while larger details are meshed with coarser mesh elements. When the Solver automatically performs order selection for the model, higher order basis functions are applied to electrically large mesh elements, while lower order basis functions are applied to electrically smaller mesh elements. With this adaptive scheme, the Solver automatically ensures high fidelity MoM solutions, using as little memory as possible and fastest possible solution times.

Note: HOBF is also supported for curvilinear mesh elements.

Setting HOBF Globally on a Model

Enable higher order basis functions on a model to allow the model to be meshed with larger triangles.

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon.
2. On the **Solver settings** dialog, click the **General** tab.
3. Under **Basis function control**, select the **Solve MoM with higher order basis functions (HOBF)** check box.

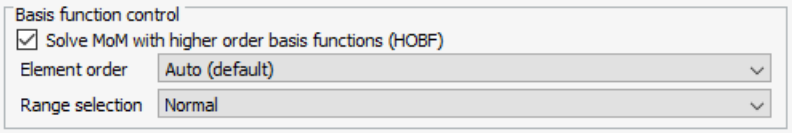


*Figure 346: A snippet of the **Solver Settings** dialog (**General** tab, **Basis function control** group).*

4. In the **Element order** drop-down list, select one of the following options:
   - To allow the Solver to select the most appropriate order, select **Auto (default)**. The order is chosen by the Solver based on the size of the element and neighbouring elements as well as the specified **Range selection**.
   - To specify the order of the basis function, select one of the predefined orders (0.5, 1.5, 2.5 and 3.5).
5. In the **Range selection** drop-down list, select one of the following options:
   - To allow the Solver to select the most appropriate range selection, select **Normal (recommended)**.
   - To allow the use of higher orders that result in a more accurate solution, but at the cost of an increase in runtime and memory, select **Prefer higher orders (more accurate, slower, more memory)**.
   - To allow the use of lower order basis functions that result in a less accurate solution, but with a shortened runtime and a decrease in memory, select **Prefer lower orders (less accurate, faster, less memory)**.
6. Click **OK** to close the dialog.

Setting HOBF Locally on a Face

Enable higher order basis functions on a face to allow the face to mesh with larger triangles.

> **Note:** Ensure HOBF is enabled globally, else the local HOBF setting will not be applied.

1. Select the part in the model tree (**Construct** tab).
2. In the details tree, select the face where you want to apply HOBF.
3. From the right-click context menu, click the **Properties** tab.
4. On the **Modify Face** dialog, click **Solution** tab.

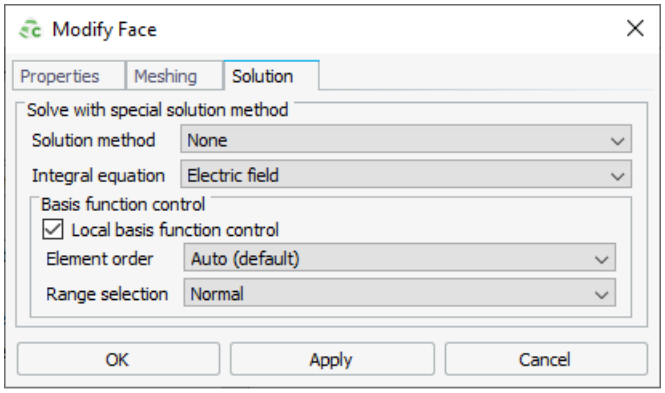


*Figure 347: The **Modify Face** dialog (**Solution** tab).*

5. In the **Element order** drop-down list, select one of the following options:
   - To allow the Solver to select the most appropriate order, select **Auto (default)**. The order is chosen by the Solver based on the size of the element and neighbouring elements as well as the specified **Range selection**.
   - To specify the order of the basis function, select one of the predefined orders (0.5, 1.5, 2.5 and 3.5).
6. In the **Range selection** drop-down list, select one of the following options:
   - To allow the Solver to select the most appropriate range selection, select **Normal (recommended)**.
   - To allow the use of higher orders that result in a more accurate solution, but at the cost of an increase in runtime and memory, select **Prefer higher orders (more accurate, slower, more memory)**.
   - To allow the use of lower order basis functions that result in a less accurate solution, but with a shortened runtime and a decrease in memory, select **Prefer lower orders (less accurate, faster, less memory)**.
7. Click **OK** to close the dialog.

Characteristic Basis Functions Method (CBFM)

Characteristic basis functions are special basis functions for the method of moments defined on large domains (blocks) that contain large numbers of sub-domains meshed into triangles.

The CBFM reduces the total number of unknowns by discarding a number of insignificant basis functions based on a certain threshold. This leads to a reduction in solution time and memory.

Note: Curvilinear mesh elements are not supported.

Solving a Model with CBFM

To solve a model with the CBFM, you must activate it for the entire model.

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon.
2. On the **Solver settings** dialog, **General** tab, in the **Characteristic basis function method (CBFM)** group, click the **Activate characteristic basis function method for MoM** check box. Select one of the following:

   MoM-based
   Use the MoM solution method to generate the CBFMs. Solution is more accurate, but at the cost of an increase in runtime and memory.

   PO-based
   Use the PO solution method to generate the CBFMs. Solution is less accurate, but with a shortened runtime and a decrease in memory.

3. Click **OK** to close the dialog.

Low-Frequency Stabilisation

At very low frequencies (frequency range where the largest dimension of the model is much smaller than a wavelength), the method of moments (MoM) solution can become numerically unstable and singular.

The default MoM solution uses single precision. When using double precision, the MoM solution is valid for lower frequencies than for single precision. If the solver gives a warning about the matrix stability when using double precision, then it is recommended to use low-frequency stabilisation.

Note: Low-frequency stabilisation is not required at higher frequencies and increases the runtime. Double precision uses double the memory of single precision.

Activating Low-Frequency Stabilisation for MoM

For very low-frequency method of moments (MoM) solutions, enable low-frequency stabilisation to ensure a valid solution over the full frequency range.

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon.
2. On the **Solver Settings** dialog, click the **General** tab.
3. Under **Low frequency modelling**, select the **Activate low frequency stabilisation for MoM** check box. From the drop-down list, select one of the following:

   Auto
   This option allows the Solver to determine automatically if the low frequency stabilisation should be used for the model.

   Always on
   This option enables the low frequency stabilisation for the model.

*Figure 348: A snippet of the **Solver settings** dialog (**General** tab, **Low frequency modelling** group).*

4. Click the **OK** to close the dialog.

Numerical Green's Function (NGF)

In the solution of large electromagnetic problems solved using the method of moments (MoM), sometimes a considerable part of the geometry remains unchanged, while only a small part changes. The unchanged part (static interaction matrix) can be saved to a `.ngf` file and reused to reduce CPU time.

> **Tip:** To obtain a reduction in CPU time, domain decomposition is recommended for MoM models consisting of a large static part and a smaller dynamic part.

A static part is indicated by the icon in the model tree.

The following restrictions apply with respect to the NGF:

- The NGF can only be activated on a part. Selecting a sub-part and activating the NGF will activate the NGF for the entire part.
- The NGF is not supported in conjunction with continuous frequency simulations.
- When the NGF is activated for a part, the following cannot be modified:
  - geometry
  - the solution method
  - media
  - ports added or deleted
  - loads
  - transmission lines
  - general networks

  The part is essentially "locked". It is allowed to add or remove sources from the part.

Using the Numerical Green's Function to Reduce CPU Time

For a method of moments (MoM) model, specify the static part. The part is saved and locked to prevent modification. Save the static interaction matrix to a `.ngf` file and reuse the file.

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **NGF** icon.

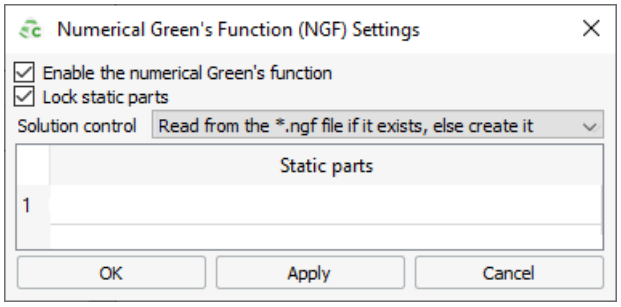


*Figure 349: The **Numerical Green's Function (NGF) Settings** dialog.*

2. Ensure the **Enable the numerical Green's function** check box is selected.
3. To lock the static part, ensure the **Lock static parts** check box is selected.

> **Note:** A locked part cannot be modified or deleted. If the frequency is changed or face properties are modified, it is recommended to unlock and remesh the part.

4. Add a part or model mesh to the list of static parts.
   a) Click on the relevant part in the 3D view or model tree.
5. Click **OK** to close the dialog.
   The static part is locked and cannot be modified. An active NGF part is indicated by the icon in the model tree.

> **Note:** To disable the NGF for a part, clear the **Enable the numerical Green's function** check box.

Defining an Aperture in an Infinite PEC Plane

Model a slot or aperture in an infinite plane using the planar Green's function aperture. The aperture is discretised instead of the surrounding ground plane, reducing the number of triangles and run time.

1. Create the geometry to model the aperture or slot in the infinite PEC plane. For example, create a rectangle to represent the aperture.
2. Select the “aperture” in the 3D view or in the model tree.
3. In the details tree, from the right-click context menu, select **Properties**.
4. On the **Modify Face** dialog, click the **Solution** tab.
5. Under **Solve with special solution method**, from the drop-down list, select **Planar Green's function aperture**.

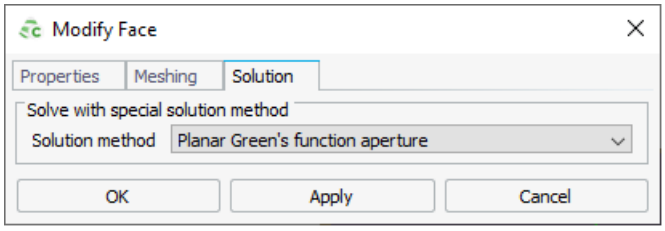


*Figure 350: The **Modify Face** dialog (**Solution** tab).*

6. Click **OK** to model the aperture as a planar Green's function aperture and to close the dialog.

2.34.6 Multilevel Fast Multipole Method (MLFMM)

The multilevel fast multipole method (MLFMM) is a current-based method applicable to electrically large structures.

Related reference
Supported Solution Method and Technique Combinations

Solving a Model with the MLFMM

To solve a model with the multilevel fast multipole method (MLFMM), you must activate MLFMM for the model.

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon.
2. On the **Solver settings** dialog, click the **MLFMM / ACA** tab.
3. Click **Solve model with the multilevel fast multipole method (MLFMM)**.

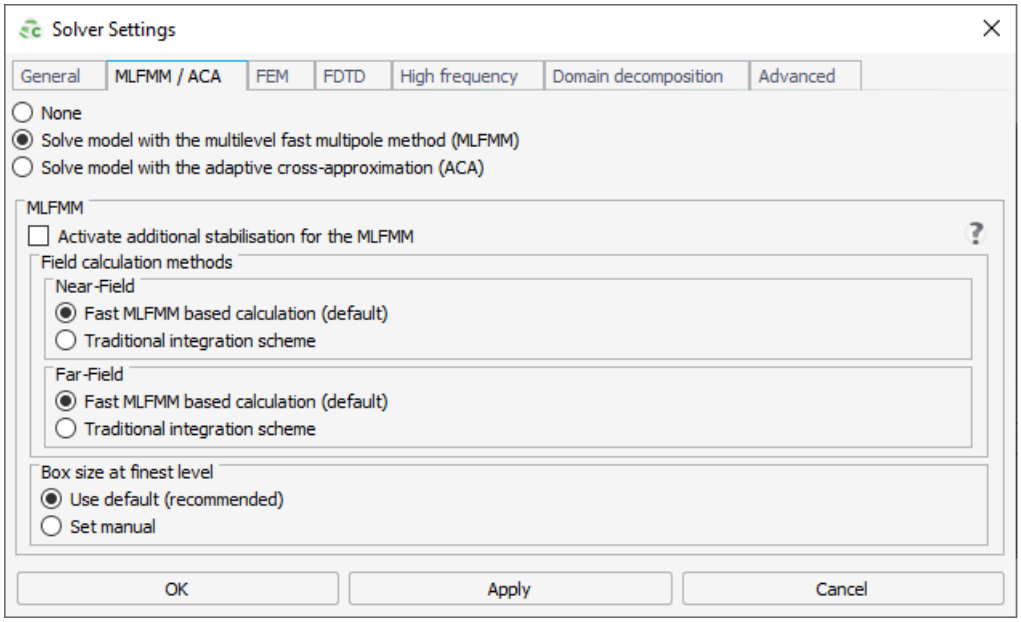


*Figure 351: The **Solver Settings** dialog (**MLFMM / ACA** tab).*

4. Click **OK** to close the dialog.

MLFMM Settings

A number of optional settings are available when using multilevel fast multipole method (MLFMM) to solve a model.

> **Note:** It is recommended to use the default settings. Modifying the advanced settings can impact accuracy and/or run time.

Activate additional stabilisation for the MLFMM
: Select this option to activate additional stabilisation for a model with severe convergence problems.

Field calculation methods

Near-field
: The MLFMM uses a fast near field calculation method (default), but in some cases the traditional integration method could be used.

Far field
: The MLFMM method uses a fast far field calculation method (default), but in some cases the traditional integration method could be used.

Box size at finest level
: The MLFMM is based on a hierarchical tree-based grouping algorithm and depending on the frequency and the model dimensions, the Solver automatically determines the number of levels in this tree and the size of the boxes at the finest level. This option allows you to adjust the box size. Adjusting the box size can improve convergence in some cases. The box size is specified in

terms of the wavelength. The default is 0.23 and the minimum value should be larger than 0.2. Increasing the box size increases the memory requirement.

> **Tip:** Make incremental changes of 0.03 at a time.

Preconditioners for MLFMM

A few preconditioners are available for the multilevel fast multipole method (MLFMM).

> **Note:**
> - It is recommended to use the default settings. Changing the preconditioner is recommended only for advanced users.
> - The number in brackets corresponds to the value of the I_3 field at the CG card in the model `.pre` file.

On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon. On the **Solver Settings** dialog, click the **Advanced** tab.

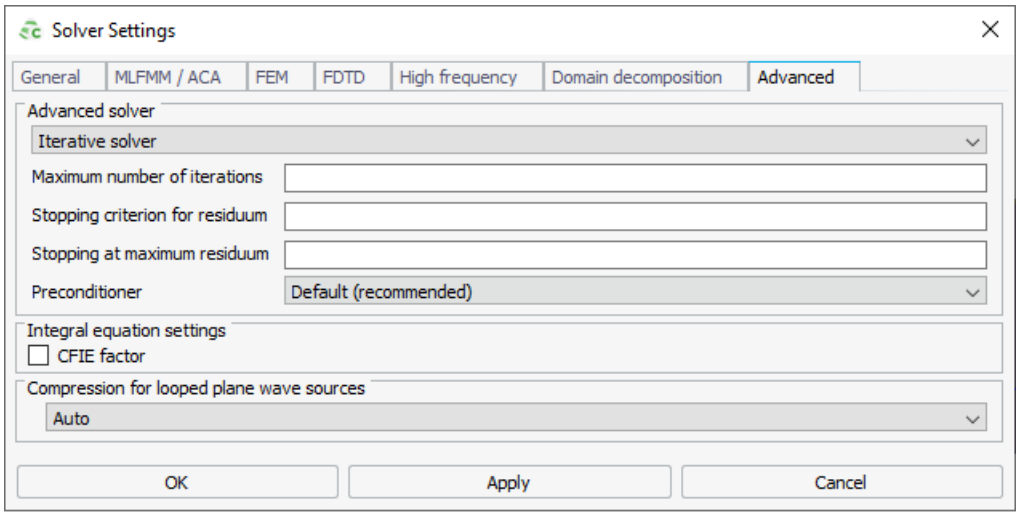


*Figure 352: The **Solver Settings** dialog (**Advanced** tab).*

Multilevel FEM-MLFMM LU/diagonal decomposition (2010)
: Preconditioner for a hybrid MLFMM / FEM solution that uses a multilevel sparse LU decomposition of the combined and partitioned system.

  When using the parallel Solver, the factorisation type, which slightly impacts runtime and memory, can be specified.

Sparse approximate inverse (SPAI) (8192)
: Preconditioner for an MLFMM solution. This preconditioner uses less memory than the default in most cases, but runtime could be longer.

Sparse LU (8193)

Preconditioner for an MLFMM solution that uses a sparse LU decomposition of the MLFMM near field matrix.

When using the parallel Solver, the factorisation type, which slightly impacts runtime and memory, can be specified.

Note: The SPAI (8192) and Sparse LU (8193) are the recommended preconditioners for MLFMM without FEM.

Related concepts
Factorisation for Parallel Execution
Related reference
CG Card

2.34.7 Modifying the Integral Equation Method

When using the MoM or MLFMM solution method, you can specify the integral equation to obtain faster iterative convergence.

1. If the model will be solved using the multilevel fast multipole method (MLFMM) activate the MLFMM.
2. Select the face(s) of the enclosed volume in the in the 3D view or in the details tree.
3. In the details tree, from the right-click context menu, select **Properties**.

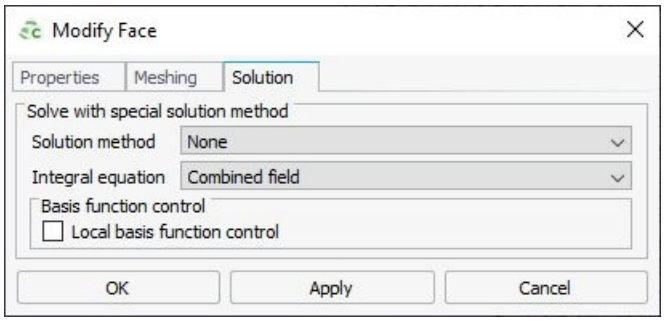


*Figure 353: The **Modify Face** dialog (**Solution** tab).*

4. On the **Modify Face** dialog, click the **Solution** tab.
5. From the **Integral equation** drop-down list, select one of the following options.
   - ***Combined field***

     Solve the model using a combination of the electric field integral equation (EFIE) and magnetic field integral equation (MFIE). This is known as the combined field integral equation (CFIE).

   - ***Electric field***

     Solve the model using the electric field integral equation (EFIE).

- ***Magnetic field***
  Solve the model using the magnetic field integral equation (MFIE).

> **Note:** The electric field integral equation (EFIE) is the default and is valid for all geometries (open, fully enclosed, metallic and dielectric).

6. Click **OK** to save the face properties and to close the dialog.

Related concepts
Integral Equation Methods (EFIE, MFIE and CFIE)
Related reference
Supported Solution Method and Technique Combinations

Using the CFIE For Closed PEC Regions

When solving an enclosed perfectly conducting metallic region using the MoM or MLFMM solution method, the CFIE can be used to improve the iterative solver convergence.

1. If the model will be solved using the multilevel fast multipole method (MLFMM), activate the MLFMM.
2. Select the faces of the enclosed volume in the in the 3D view or in the details tree.
3. In the details tree, from the right-click context menu, select **Properties**.
4. On the **Modify Face** dialog, click the **Solution** tab.
5. From the **Integral equation** drop-down list, select **Combined field**[44].
6. Click **OK** to save the face properties and to close the dialog.

Related concepts
Integral Equation Methods (EFIE, MFIE and CFIE)

Modifying the CFIE Factor

Activate the combined field integral equation (CFIE) for the model and specify the factor for the linear combination of the magnetic field integral equation (MFIE) and the electric field integral equation (EFIE).

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon.
2. On the **Solver Settings** dialog, click the **Advanced** tab.
3. Under **Integral equation settings**, select the **CFIE** factor check box and in the edit field, enter a value for the CFIE factor where 0 < CFIE factor < 1.

> **Note:** The default factor is 0.2. This is a ratio of 80% MFIE to 20% EFIE.

44. combined field integral equation (CFIE)

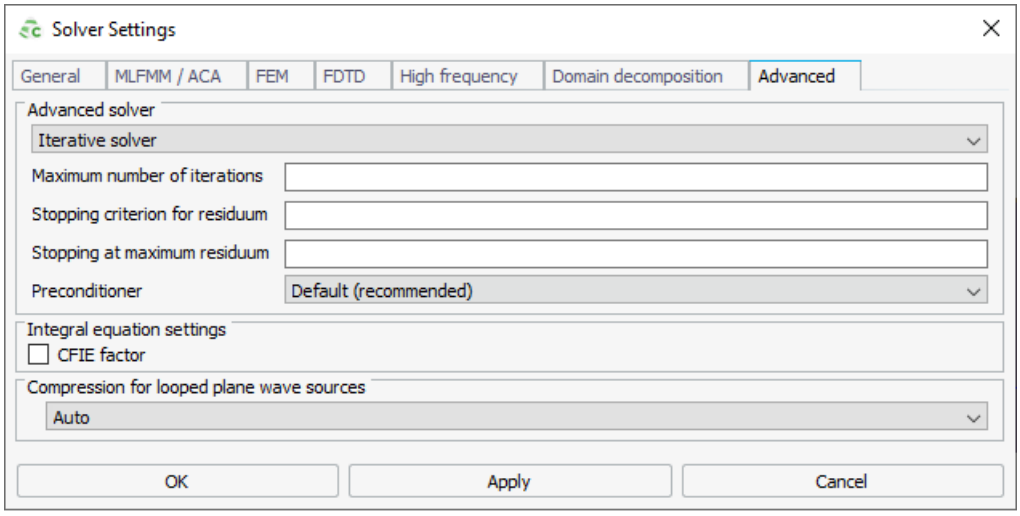


*Figure 354: The **Solver Settings** dialog (**Advanced** tab).*

4. Click **OK** to close the dialog.

Related concepts
Integral Equation Methods (EFIE, MFIE and CFIE)

2.34.8 Adaptive Cross-Approximation (ACA)

The adaptive cross-approximation (ACA) is a fast method, similar to multilevel fast multipole method (MLFMM). The method improves the solution of certain complex method of moments (MoM) problems using less memory and run-time.

The ACA method does not suffer from low-frequency breakdown and is also applicable to using the special Green's function.

Related reference
Supported Solution Method and Technique Combinations

Solving a Model with Adaptive Cross-Approximation (ACA)

For complex, method of moments (MoM) problems, solve the model with the adaptive cross-approximation (ACA) to reduce memory and run-time.

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon.
2. On the **Solver Settings** dialog, click the **MLFMM / ACA** tab.
3. Click **Solve model with adaptive cross-approximation (ACA)**.

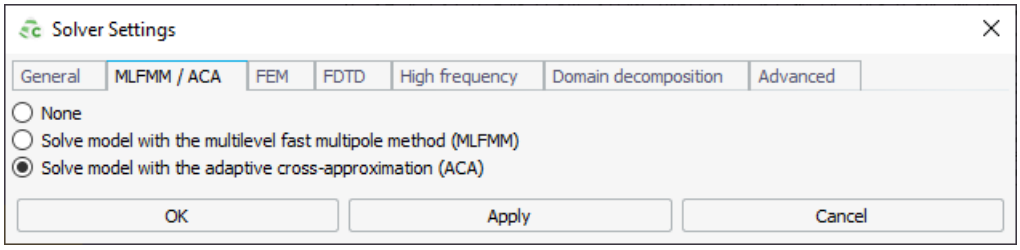


*Figure 355: The **Solver Settings** dialog (**MLFMM / ACA** tab).*

4. Click the **OK** button to close the dialog.

2.34.9 Finite Element Method (FEM)

The finite element method (FEM) is a volume meshing technique used to model electrically complex or inhomogeneous dielectric bodies.

Related reference
Supported Solution Method and Technique Combinations

Solving a Region with FEM

To solve a model with the finite element method (FEM), you must activate the FEM for each relevant region.

1. Select the region(s) in the 3D view or in the details tree that you want to solve with the FEM.
2. In the details tree, from the right-click context menu, select **Properties**.
3. On the **Modify Region** dialog, click the **Solution** tab.
4. Under **Solution method**, from the drop-down list, select **Finite Element Method (FEM)**.
5. [Optional] Under **Element order control**, select the **Local element order control** check box to specify the element order locally per region.

   This setting takes precedence over the element order set globally per model. Use **First order only (reduced accuracy)** to reduce the required memory and run-time for regions with fine details where the geometric details (and resulting mesh) are very fine.
6. Click **OK** to save the region properties and to close the dialog.

Related concepts
Element Order Per Model

FEM Parameters

Optional parameters can be used with the finite element method (FEM) to save memory and runtime in specific cases.

On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon. On the **Solver Settings** dialog, click the **FEM** tab.

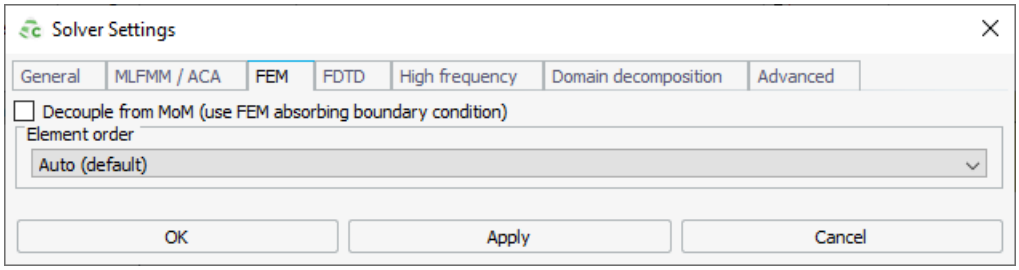


*Figure 356: The **Solver Settings** dialog (**FEM** tab).*

Decouple from MoM (use FEM absorbing boundary condition)
: This option removes the influence of the FEM region (the tetrahedral elements and any conducting surfaces on their boundaries) on the MoM solution. The runtime and memory will be less compared to a fully coupled (default) solution. Closed FEM problems (for example, completely confined by PEC and / or modal port boundaries such as waveguides) are highly suitable, automatically detected and the MoM solver will be deactivated. For other types of problems this option should be used with caution. For example, the input impedance of a dipole antenna close to a human head, where the dipole is solved using the MoM and the human head using FEM, will (incorrectly) be the same as that of the MoM dipole in free space.

> **Tip:** Use this option if the MoM and FEM regions are electrically far apart.

Element order
: This option allows you to specify the element order for the model. Use **First order only (reduced accuracy)** to reduce the required memory and run-time for regions with fine details where the geometric details (and resulting mesh) are very fine.

> **Note:** Use **First order** for very fine meshes to reduce memory and runtime.

Preconditioners for FEM

A few preconditioners are available for the finite element method (FEM).

> **Note:**
>
> - It is recommended to use the default settings. Changing the preconditioner is recommended only for advanced users.
> - The number in brackets corresponds to the value of the I_3 field at the CG card in the model `.pre` file.

On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon. On the **Solver Settings** dialog, click the **Advanced** tab.

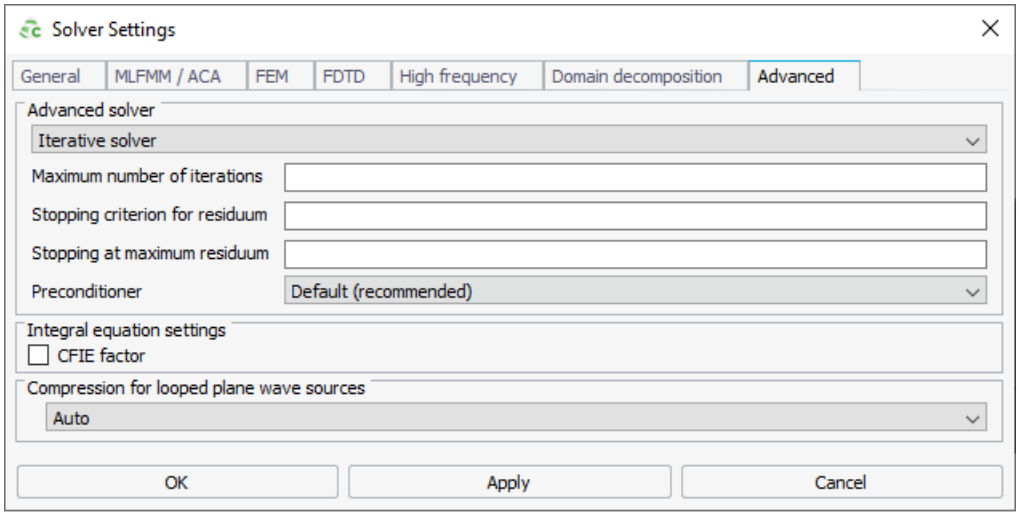


*Figure 357: The **Solver Settings** dialog (**Advanced** tab).*

Multilevel ILU/diagonal decomposition (512)
: Preconditioner for a hybrid FEM/MoM solution that uses a multilevel sparse incomplete LU-decomposition with threshold and controlled fill-in. Note, not available for parallel.

Multilevel FEM-MLFMM LU/diagonal decomposition (2010)
: Preconditioner for a hybrid FEM/MLFMM solution that uses a multilevel sparse LU decomposition of the combined and partitioned, FEM/MLFMM system. This is the default for a FEM/MLFMM solution.

  When using the parallel , the factorisation type, which slightly impacts runtime and memory, can be specified.

Multilevel LU/diagonal decomposition (2050)
: Preconditioner for a hybrid FEM/MoM solution that uses a multilevel sparse LU decomposition of the partitioned system. This is the default for a FEM/MoM solution.

  When using the parallel , the factorisation type, which slightly impacts runtime and memory, can be specified.

Related concepts
Factorisation for Parallel Execution
Related reference
CG Card

2.34.10 Physical Optics (PO) and Large Element Physical Optics (LE-PO)

The physical optics (PO) solver is an asymptotic high-frequency numerical solver based on currents. Use the method in instances where electrically very large metallic structures are modelled.

The large element physical optics (LE-PO) solution method is similar to the PO method but allows larger triangular mesh elements to be used.

Related reference
Supported Solution Method and Technique Combinations

Solving Faces with Physical Optics (PO)

To solve a model with physical optics (PO), you must activate PO for each relevant face.

1. Select the face(s) in the 3D view or in the details tree that you want to solve with PO.
2. In the details tree, from the right-click context menu, select **Properties**.
3. On the **Modify Face** dialog, click the **Solution** tab.
4. Under **Solve with special solution method**, from the drop-down list, select one of the following:
   - To use complete ray tracing, select **Physical optics (PO) - full ray-tracing**.
   - If the assumption can be made that all triangles on which the PO approximation is made, are illuminated, select **Physical optics (PO) - always illuminated**. Ray-tracing is switched off to reduce run time.
   - To use full ray tracing when the metallic triangles are only illuminated from the front (normals side), select **Physical optics (PO) - only illuminated from front**.
5. Click **OK** to save the face properties and to close the dialog.

Solving Faces with Large Element Physical Optics (LE-PO)

To solve a model with large element physical optics (LE-PO), you must activate LE-PO for each relevant face.

1. Select the face(s) in the 3D view or in the details tree that you want to solve with LE-PO.
2. In the details tree, from the right-click context menu, select **Properties**.
3. On the **Modify Face** dialog, click the **Solution** tab.
4. Under **Solve with special solution method**, from the drop-down list, select one of the following:
   - To use complete ray tracing, select **Large element PO - full ray-tracing**.
   - If the assumption can be made that all triangles on which the PO approximation is made, are illuminated, select **Large element PO - always illuminated**. Ray-tracing is switched off to reduce run time.
   - To use full ray tracing when the metallic triangles are only illuminated from the front (normals side), select **Large element PO - only illuminated from front**.

5. Click OK to save the face properties and to close the dialog.

PO and LE-PO Settings

Optional parameters can be used with the physical optics (PO) or large element physical optics (LE-PO) to save memory and runtime in specific faces.

On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon. On the **Solver Settings** dialog, click the **High frequency** tab.

Couple PO and MoM/MLFMM solutions (iterative technique, default)
: This option uses a hybrid iterative technique to determine the coupling (interaction) between the MoM or MLFMM region and the PO region. As a result, the currents in the PO region will have an effect on the current distribution in the MoM region.

Couple PO and MoM solutions (full coupling)
: This option takes into account the coupling between the MoM region and PO regions. This is a non-iterative technique and uses more memory compared to the iterative technique. The currents in the PO region will have an effect on the current distribution in the MoM region.

Decouple PO and MoM solutions
: This option ignores the PO regions when calculating the MoM currents. The runtime and memory will be less compared to a fully coupled (default) solution. This option should be used with caution. For example, the input impedance of a dipole antenna in close proximity to a metallic plate, where the dipole is solved using the MoM and the plate using PO, will (incorrectly) be the same as that of the MoM dipole in free space.

> **Tip:** Use this option where the MoM and PO regions are electrically far apart (for example a reflector antenna).

Maximum number of iterations
: This option limits the number of iterations for the iterative coupling technique.

Stopping criterion for residuum
: This option specifies the termination criterion for the normalised residue when using the iterative method. The iterative solution is terminated when the normalised residue is smaller than this value.

Store / reuse shadowing information
: During calculations using the PO formulation, a large amount of the runtime could be spent in determining which surfaces are illuminated from the source(s). This option saves the shadowing information to speed up subsequent runs. Re-use is only possible if the mesh remains unchanged.

> **Note:** Storage of the shadowing information could cause large `.sha` files on disk.

Use symmetry in ray-tracing (where possible)
This options allows symmetry to be used in full ray tracing when determining the shadowing to reduce runtime. For geometrical symmetry, select this option to utilise symmetry. For electric and magnetic symmetry, this speed up is always used.

2.34.11 Ray Launching Geometrical Optics (RL-GO)

The ray launching geometrical optics (RL-GO) solver is a ray-based solver that models objects based on optical propagation, reflection and refraction theory.

Related concepts
Which Solution Methods Support GPU Acceleration?
Related reference
Supported Solution Method and Technique Combinations

Solving a Model with RL-GO

To solve a model with ray launching geometrical optics (RL-GO), you must activate RL-GO for each relevant face.

1. Select the face (or faces) in the in the 3D view or in the details tree that you want to solve with RL-GO.
2. In the details tree, from the right-click context menu, select **Properties**.
3. On the **Modify Face** dialog, click the **Solution** tab.
4. Under **Solve with special solution method**, from the drop-down list, select **Ray launching - geometrical optics (RL-GO)**.
5. Click **OK** to save the face properties and to close the dialog.

RL-GO Settings

A number of optional settings are available when using ray launching geometrical optics (RL-GO) to solve parts of the model.

On the **Solve/Run** tab, in the **Solution Settings** group, click the  **Solver Settings** icon. On the **Solver Settings** dialog, click the **High frequency** tab.

Decouple from MoM solutions
This option ignores the RL-GO regions when calculating the MoM currents. The runtime and memory will be less compared to a fully coupled (default) solution. This option should be used with caution. For example, the input impedance of a dipole antenna close to a dielectric sphere, where the dipole is solved using the MoM and the sphere using RL-GO, will (incorrectly) be the same as that of the MoM dipole in free space.

> **Tip:** Use this option where the MoM and RL-GO regions are electrically far apart (for example an aircraft nose cone radome).

Maximum number of ray interactions
This option limits the number of ray interactions (reflection and diffraction combined). For example, if this parameter is set to 3, a ray can have three reflections, or two reflections and a transmission. If left empty, then the maximum number if ray interactions is determined automatically.

Edge and wedge diffractions
This option takes the diffraction on edges and wedges into account.

Export ray data for post-processing to

**.bof file (default)*
This option exports the rays during the RL-GO solution process to the `.bof` file for visualisation in POSTFEKO.

**.ray file*
This option exports the rays during the RL-GO solution process to a `.ray` file. This text file can be used for custom post-processing.

> **Note:** Large `.ray` files are possible when the MoM and RL-GO solution have not been decoupled and the MoM region contains a large number of mesh elements.

Adaptive ray launching settings
This option allows you to control the density of the rays launched, as well as when to stop tracing a ray based on the ray's decay.

- **High (more rays)**: The ray density is high. Results take longer to obtain but with higher accuracy.
- **Normal (default)**: The default ray density setting.
- **Low (fewer rays)**: The ray density is low. Results are fast to obtain but with lower accuracy.

> **Tip:** Start with **Low (fewer rays)** which uses the least computational resources. When the model appears to be performing as expected, use a higher setting.

Fixed grid increments
This option allows you to specify the angular or spatial resolution for ray launching. The resolution is specified by means of the increments in the U direction and V direction for a parallel ray front (plane wave source) or in the φ and ϑ directions (sources other than plane waves). Though the run-time for a problem involving RL-GO may be decreased using this option, it may influence the accuracy of the solution.

> **Note:** Manual specification of the increments should only be used after the implications have been carefully considered.

2.34.12 Uniform Theory of Diffraction (UTD)

The uniform theory of diffraction (UTD) is an asymptotic high-frequency numerical solver. The method is typically used for electrically extremely large PEC structures.

The Solver has two UTD solution methods:

- *Faceted uniform theory of diffraction (faceted UTD)*
  This method is well-suited for antenna placement on electrically large platforms with curved surfaces (such as aircrafts). It is a frequency independent solver which uses planar mesh triangles to approximate the structures, including surface curvature. The method takes into account multiple reflections, edge and wedge diffraction, corner diffraction and creeping waves.

- *Uniform theory of diffraction (UTD) with polygons or cylinder*
  This method is well-suited for antenna placement of electrically large platforms on flat surfaces. It is a frequency independent solver which uses polygons to approximate the structures, but it does not consider surface curvature. This method can also be used to solve a single canonical circular cylinder.

Related reference
Supported Solution Method and Technique Combinations

Solving a Flat Face with UTD

To solve a model with the uniform theory of diffraction (UTD), you must activate UTD for each relevant flat face.

1. Select the face(s) in the 3D view or in the details tree that you want to solve with UTD.
2. In the details tree, from the right-click context menu, select **Properties**.

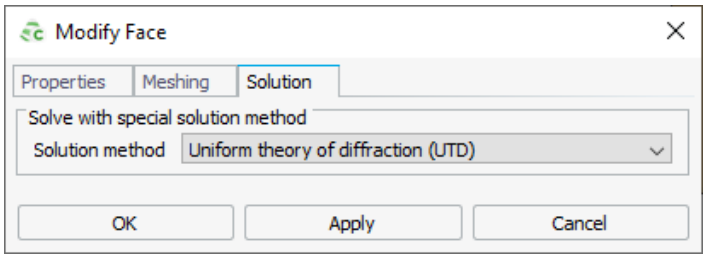


*Figure 358: The **Modify Face** dialog (**Solution** tab).*

3. On the **Modify Face** dialog, click the **Solution** tab.
4. Under **Solve with special solution method**, from the drop-down list, select **Uniform theory of diffraction (UTD)**.
5. Click OK to save the face properties and to close the dialog.

Solving a Curved Face with Faceted UTD

To solve a model with the faceted uniform theory of diffraction (faceted UTD), you must activate faceted UTD for each relevant curved face.

1. Select the face(s) in the 3D view or in the details tree that you want to solve with UTD.
2. In the details tree, from the right-click context menu, select **Properties**.

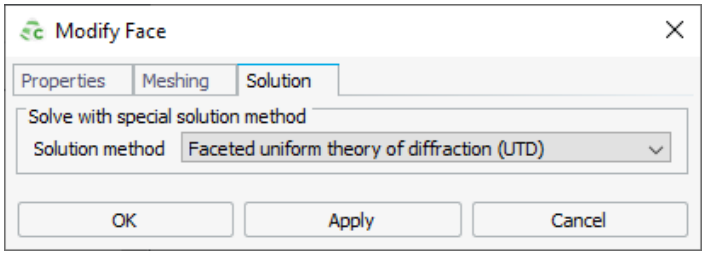


*Figure 359: The **Modify Face** dialog (**Solution** tab).*

3. On the **Face properties** dialog, click the **Solution** tab.
4. Under **Solve with special solution method**, from the drop-down list, select **Faceted uniform theory of diffraction (UTD)**.
5. Click OK to save the face properties and to close the dialog.

Creating a UTD Cylinder

Solve a cylinder with the uniform theory of diffraction (UTD). The cylinder can either be a finite, semi-infinite or infinite, depending on the termination type of the start cap and end cap.

1. Create a cylinder.
2. Select the cylinder in the 3D view or in the details tree.
3. In the details tree, from the right-click context menu, select **Properties**.
4. On the **Modify Region** dialog, click the **Solution** tab.

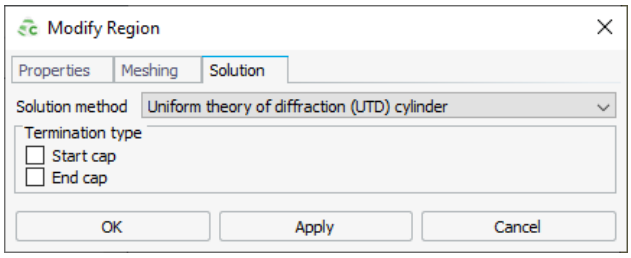


*Figure 360: The **Modify Region** dialog (**Solution** tab).*

5. Under **Solution method**, from the drop-down list, select **Uniform theory of diffraction (UTD) cylinder**.
6. Under **Termination type**, specify if the UTD cylinder is infinite or finite sized at the start and/or end cap.
   - To define a finite or semi-finite cylinder, select the **Start cap** and/or **End cap** check boxes.

- To define an infinite cylinder, clear the **Start cap** and **End cap** check boxes.

7. Click **OK** to save the region properties and to close the dialog.

UTD Settings

A number of optional settings are available for the uniform theory of diffraction (UTD) to solve a face or faces.

On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon. On the **Solver Settings** dialog, click the **High frequency** tab.

Decouple from MoM solutions

This option ignores the UTD surfaces when calculating the MoM currents. The runtime and memory will be less compared to a fully coupled (default) solution. For example, the input impedance of a dipole antenna in close proximity to a PEC plate, where the dipole is solved using the MoM and the plate using UTD, will (incorrectly) be the same as that of the MoM dipole in free space. If equivalent sources, such as far field or near field sources, are used, this option will have no effect.

> **Tip:** Use this option where the MoM and UTD regions are electrically far apart.

Maximum number of ray interactions

This option limits the number of ray interactions (reflection and diffraction combined). For example, if this parameter is set to 3, a ray can have three reflections, or two reflections and a diffraction. A value of 0 means that only direct rays are taken into account.

> **Note:** For faceted UTD, this setting only affects the wedge diffracted rays (up to 2 diffractions), reflected rays and combination of different effects.

Export ray data for post-processing to

**.bof file (default)*

This option exports the rays during the UTD solution process to the `.bof` file for visualisation in POSTFEKO.

**.ray file*

This option exports the rays during the UTD solution process to a `.ray` file. This text file can be used for custom post-processing.

> **Note:** Large `.ray` files are possible when the MoM and UTD solution have not been decoupled and the MoM part contains a large number of mesh elements.

Enable acceleration (for faceted UTD)

An acceleration technique for faceted UTD can be used to speed-up the search process for ray paths significantly but could result in some rays not being found in exceptional cases.

Auto
The Solver determines automatically if the acceleration technique should be used for the model (if the method is likely to speed up the solution).

On
This option enables the acceleration technique. Runtime decreases but technique could result in some rays not being found in exceptional cases.

Off
This option disables the acceleration technique at the expense of a runtime increase.

UTD Ray Contributions

A number of optional ray contribution parameters are available for the faceted UTD.

Faceted UTD

Direct field
This option takes into account the direct rays.

Edge and wedge diffraction
This option takes into account the diffraction on edges and wedges.

Surface reflection
This option takes into account the rays reflected by PEC and non-metallic planar and curved surfaces.

Creeping waves
This option takes into account the creeping waves on curved surfaces.

Surface transmission
This option takes into account the rays transmitted (refracted) by PEC and non-metallic planar and curved surfaces.

Combination of different effects
This option allows for multiple reflections plus one edge/wedge diffraction at any position along the ray path to be computed. This option is only active if the **Surface reflection** and **Edge and Wedge diffraction** check boxes are selected and the **Maximum number of ray interactions** is larger than 1.

Corner and tip diffraction
This option takes into account the diffraction at corners and tips.

UTD (Polygons and Cylinder)

Direct and reflected
This option takes into account both the direct rays and reflected rays.

Double diffraction
This option takes into account the double diffraction on edges and wedges and the combinations of reflections. Single diffraction rays are not included for this option.

Edge and wedge diffractions
This option takes into account the diffraction on edges and wedges. The ray may include an arbitrary number of reflections, but only one diffraction.

> **Note:** The total number of interactions (number of reflections) plus one for the diffraction may not be larger than the value specified in the **Maximum number of UTD ray interactions** field.

Creeping waves
This option takes into account the creeping waves on a cylinder.

Corner diffraction
This option takes into account the corner diffraction.

Cone tip diffraction
This option takes into account the diffraction at the tip of the cone.

2.34.13 Finite Difference Time Domain (FDTD)

The finite difference time domain (FDTD) solver is well suited to modelling inhomogeneous materials and models with wide bandwidths.

Related concepts
Which Solution Methods Support GPU Acceleration?
Related reference
Supported Solution Method and Technique Combinations

Solving a Model with FDTD

To solve a model with the finite difference time domain (FDTD), you must activate the solver for the model.

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **Solver Settings** icon.
2. On the **Solver Settings** dialog, click the **FDTD** tab.

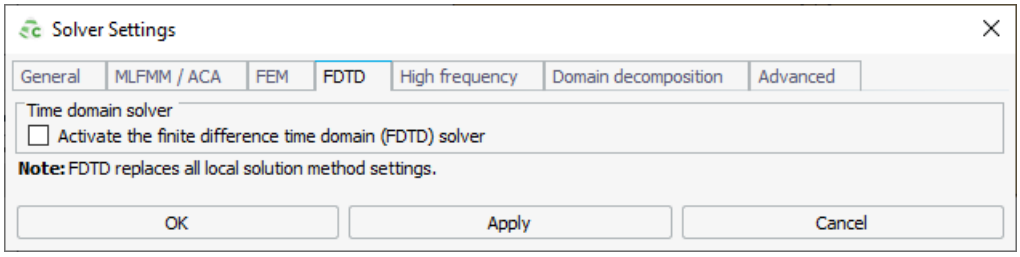


*Figure 361: The **Solver Settings** dialog (**FDTD** tab).*

3. Under **Time domain solver**, select the **Activate the finite difference time domain (FDTD) solver** check box.
4. Click the **OK** to close the dialog.

Specifying the FDTD Boundary Conditions

The boundary conditions define the size and type of boundaries of the volume solved by the finite difference time domain (FDTD) solver.

1. On the **Solve/Run** tab, in the **Solution Settings** group, click the **FDTD Boundary Conditions** icon.

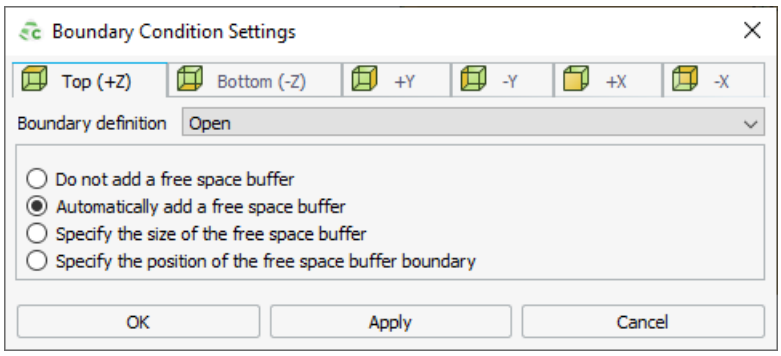


*Figure 362: The **Boundary Condition Settings** dialog (**Top (+Z)** tab).*

2. On the **Boundary Condition Settings** dialog, click the **Top (+Z)** tab to specify the boundary in the positive Z axis.
3. Specify the boundary definition by selecting one of the following from the **Boundary definition** drop-down list:
   - To specify an open radiating boundary, implemented as a convolutional perfectly matched layer (CPML), select **Open**.
   - To specify a PEC boundary that allows efficient simulation of infinitely large electrically conducting planes, select **PEC**.
   - To specify a PMC boundary that allows efficient simulation of infinitely large magnetically conducting planes, select **PMC**.
4. Enlarge a volume by adding a free space buffer[45] by selecting one of the following:
   - If no free space buffer is required, select **Do not add a free space buffer**.
   - To automatically add a free space buffer (perpendicular to the specific face), select **Automatically add a free space buffer**.
   - To specify the size of the free space buffer to be added to the specified face, select **Specify the size of the free space buffer**.
     - In the **Free space buffer region size** field, enter a value.
   - To specify the position of the free space buffer on the respective axis, select **Specify the position of the free space buffer boundary**.
     - In the **Position on the Z axis** field, enter a value.
5. Repeat Step 2 to Step 4 for the remaining five faces of the boundary.
6. Click **OK** to define the boundary condition and to close the dialog.

45. The buffer is the space between the bounding box of the model and the position of the FDTD boundary.

Note: A free space boundary condition is only displayed in the 3D view when the Configuration tab is selected.

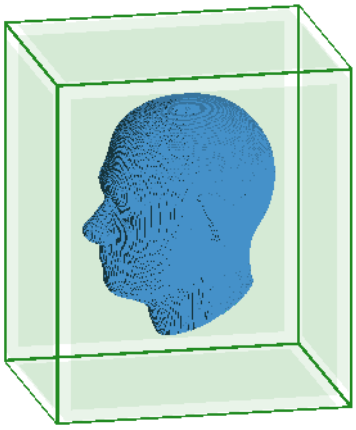


Figure 363: An example of the display for six free space boundary conditions.

2.34.14 Dielectric Surface Impedance Approximation

The dielectric surface impedance approximation is a solution method that can be applied to homogeneous, lossy dielectric regions. Use the solution method to compute SAR values for a homogeneous phantom.

Note: Restrictions apply when solving regions with the dielectric surface impedance approximation:

- The region must be in free space (cannot be contained inside another region).
- The region may not touch or intersect another region when the media properties differ.
- The boundary surface must be a closed dielectric surface without any metal parts.
- No sources may be located inside the dielectric region.

Related reference
Supported Solution Method and Technique Combinations

Solving a Region with the Dielectric Surface Impedance Approximation

Activate the dielectric surface impedance approximation solution method for an homogeneous, lossy dielectric region. Use the solution method to compute SAR values for a homogeneous phantom.

1. Select the region (or regions) in the 3D view or in the details tree that you want to solve with the dielectric surface impedance approximation.
2. In the details tree, from the right-click context menu, select **Properties**.

3. On the **Modify Region** dialog, click the **Solution** tab.
4. Under **Solution method**, from the drop-down list, select **Dielectric surface impedance approximation**.
5. Click **OK** to save the region properties and to close the dialog.

2.35 Component Launch Options

Specify the command-line parameters for the Feko components.

On the **Solve/Run** tab, in the **Run/Launch** group, click the dialog launcher.

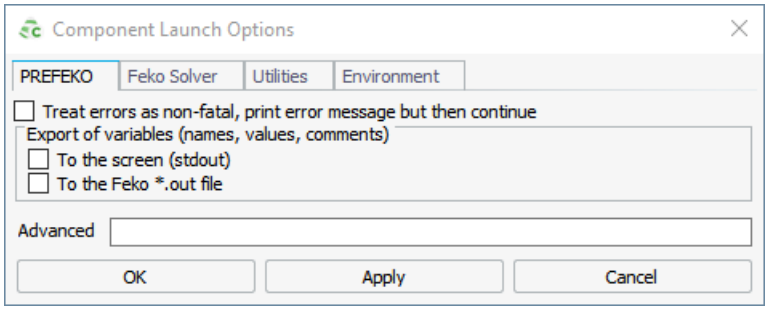


*Figure 364: The **Component Launch Options** dialog.*

2.35.1 PREFEKO Options

Specify PREFEKO command-line parameters using the GUI.

Treat errors as non-fatal, print error message but then continue
: This option allows PREFEKO to continue running after encountering an error.

Export of Variables (Names, Values, Comments)

To the screen (stdout)
: This option exports variables to the screen (stdout).

*To the Feko *.out file*
: This option exports variables to the `.out`.

Advanced

The **Advanced** field allows you to add command-line options similar to when using a command shell.

Related reference
Running PREFEKO

2.35.2 Feko Solver Options

Specify Solver command-line parameters using the GUI.

Only check the geometry
: This option allows you to perform geometry checks to check the model validity and exit before the full solution starts.

> **Tip:** First check the geometry of large models for clusters on a local machine.

Process priority
This option allows you to specify the priority of the Feko processes. If the priority is set to **Low**, the solution could take slightly longer, but the CPU will still be responsive to other work.

> **Note:** For parallel runs, all machines in the cluster operate at the speed of the slowest machine. Starting additional CPU-intensive jobs on a machine(s) in the cluster is generally not recommended.

Export SPICE MTL circuit files
This option allows you to export SPICE MTL harness files for further processing by a third-party SPICE simulator.

Graphics Processing Units (GPU)

Accelerate Solver runs using multiple NVIDIA GPUs based on the compute unified architecture (CUDA). GPU acceleration is only applicable if a compatible NVIDIA GPU device(s) is found.

> **Note:** Minimum requirements for the CUDA device:
> - Compute capability of at least 3.5
> - Driver installed on system must support CUDA 11.6.

Use GPU (graphical processing) for NVIDIA CUDA devices
This option allows you to make use of NVIDIA GPUs to accelerate Solver runs.

> **Note:** Not all solvers fully support GPU acceleration.

Number of GPUs (empty = all)
This option allows you to specify the number of GPUs to use (if multiple GPUs are available and supported by the solver).

List of GPUs (optional comma separated list)
Specify the list of available GPUs using a comma ",” as separation.

Remote Execution

Remote host (hostname or IP address)
Specify the machine to be used as the remote host.

ssh / rsh (must be installed on remote and local machine)
This option uses a remote shell (either RSH or SSH or similar) for launching the process. For copying of the files, SCP (or similar) is used. The remote machine must be able to serve such connection attempts (an SSH daemon must be set up and running with public key authentication). This method can be used between different platforms.

MPI (Windows only)
This option is only supported between Windows machines (both machines must run a Windows operating system). This method uses native windows file copy methods and a shared network folder on the remote machine for transferring the model files and results. The launching on the remote machine is done by the MPI daemon which is already installed during installation for parallel launching. Authentication is done by Windows internal mechanisms, as a result, the remote machine must be able to authenticate the current user either against a domain or its local user database to grant access.

Parallel Execution

Specify number of parallel processes
This option allows you to specify the number of parallel processes to be launched.

> **Note:** If the number of parallel processes is not specified, then the machines (with their specified number of parallel processes) as stated in the machines file, is used for launching.

Full CPU report with run times for individual processes
This option enables diagnostic tests and outputs a full CPU report with run times for individual processes. For normal runs, this option should be disabled to not degrade performance.

Output MFLOPS rate of each process (without network communication time)
This option enables diagnostic tests and outputs the MFLOPS rate of each process (without network communication time). For normal runs, this option should be disabled to not degrade performance.

Network latency and bandwidth
This option enables diagnostic tests and outputs the network latency and bandwidth. For normal runs, this option should be disabled to not degrade performance.

Parallel Authentication Method

Use encrypted credentials in registry (Windows only)
This option uses a previously stored encrypted username and password from the Windows registry. You have to save these credentials prior to starting a parallel simulation using the *Update parallel credentials* tool[46]

> **Note:** This setting is a per-user setting and must be updated after any changes to the user's credentials. If using remote-parallel launching, this must also be done on the remote host where the parallel Feko solution is started from.

Use SSPI (Active Directory) integration (Windows only, requires domain)
For this option all the machines must be a member of a Windows Active Directory (AD) domain and the user accounts must be domain accounts. The authentication is carried out using internal Windows mechanisms without having to encrypt anything into the registry.

46. The Update parallel credentials tool is available on the Launcher utility.

> **Note:** A one-time configuration of the settings may be required by the domain administrator to prepare the Windows domain for this kind of authentication.

Local run only (no authentication required)
This option runs the parallel job on the local host only and requires no authentication.

> **Tip:** Use this option for parallel runs on a single local machine.

Default (rsh / ssh for UNIX, registry for Windows)
This option always uses the default authentication method for the target operating system.

Windows
The Use encrypted credentials in registry (Windows only option) is considered the default.

UNIX
The public key authentication of rsh / ssh is used.

Advanced

The **Advanced** field allows you to add command line options similar to when using a command shell.

Related concepts
Which Solution Methods Support GPU Acceleration?

2.35.3 ADAPTFEKO Options

Specify ADAPTFEKO command-line parameters using the GUI.

Restart analysis number
This option can be used if the run was discontinued and the temporary files were not deleted. Restart the solution at the number of the first incomplete analysis number.

Delete temporary files
This option allows you to delete the temporary files created during the ADAPTFEKO run.

> **Tip:** Unselect this check box to allow resumption of an interrupted run.

2.35.4 OPTFEKO Options

Specify OPTFEKO command-line parameters using the GUI.

Restart from solver run
This option can be used if the optimisation process was terminated or interrupted and temporary files are available. Restart the optimisation at the number of the last incompleted optimisation number. No changes may be made to the model before restarting the optimisation process.

Delete all files (except optimum)
This option deletes all temporary files except the files related to the optimum solution.

> **Tip:** Unselect this check box to allow resumption of an interrupted run.

Number of processes to farm out
This option allows you to specify the number processes to use per optimisation iteration when farming out the solution.

Configure
Specify the hosts and processes to be used when farming out the kernel solution.

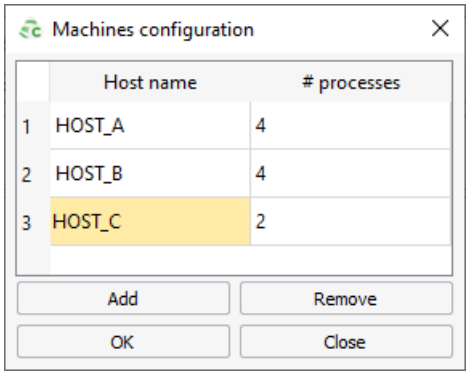


*Figure 365: The **Machines configuration** dialog.*

2.35.5 AMRFEKO Options

Specify AMRFEKO command-line parameters using the GUI.

Specify frequency (Hz)
This option can be used if the refinement should be run at a different frequency from the current model setting.

Specify maximum number of iterations
This option limits the number of times AMRFEKO will perform a refinement. More iterations will result in a longer runtime.

> **Warning:** If the number of iterations is limited, the mesh will not be refined sufficiently and can lead to inaccurate results.

Error threshold
For the adaptive mesh refinement, an error threshold is defined where the mesh is adapted until the error function drops below this threshold.

High
A smaller number of mesh elements is refined (least aggressive refinement).

Normal
Default

Low
A larger number of mesh elements is refined (most aggressive refinement) and will increase the runtime.

Delete temporary files
This option deletes all temporary files except the files related to the refined solution.

Skip final solver run
This option only saves the refined model and skips the solution thereof.

Advanced

The **Advanced** field allows you to add command line options similar to when using a command shell.

2.35.6 Environment Variables to Control the Solution

Add environment variables to be used during the launching of processes.

Specify an environment variable per single line, for example:

```
VARIABLE=VALUE
```

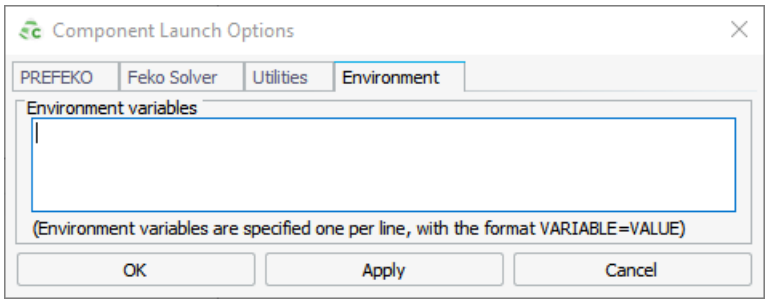


*Figure 366: The **Component Launch Options** dialog.*

2.36 Tools

CADFEKO has a collection of tools that allows you to quickly validate the model, for example, perform calculations using a calculator, measure distances, measure angles and export images.

2.36.1 Measuring a Distance

The measure distance tool allows you to measure or validate the physical distance between two points in a model.

1. On the **Tools** tab, in the **Tools** group, click the **Measure Distance** icon.
2. Under **Point1**, use Ctrl+Shift+left click to snap to points (for example, named points, geometry points, geometry face centre, geometry edge centre, mesh vertices and grid).
3. Repeat Step 2 for **Point 2**.
   The total distance, as well as the individual X axis, Y axis and Z axis distances, are displayed in the **Distance (D)**, **X distance**, **Y distance** and **Z distance** fields respectively.
4. Click **Close** to close the dialog.

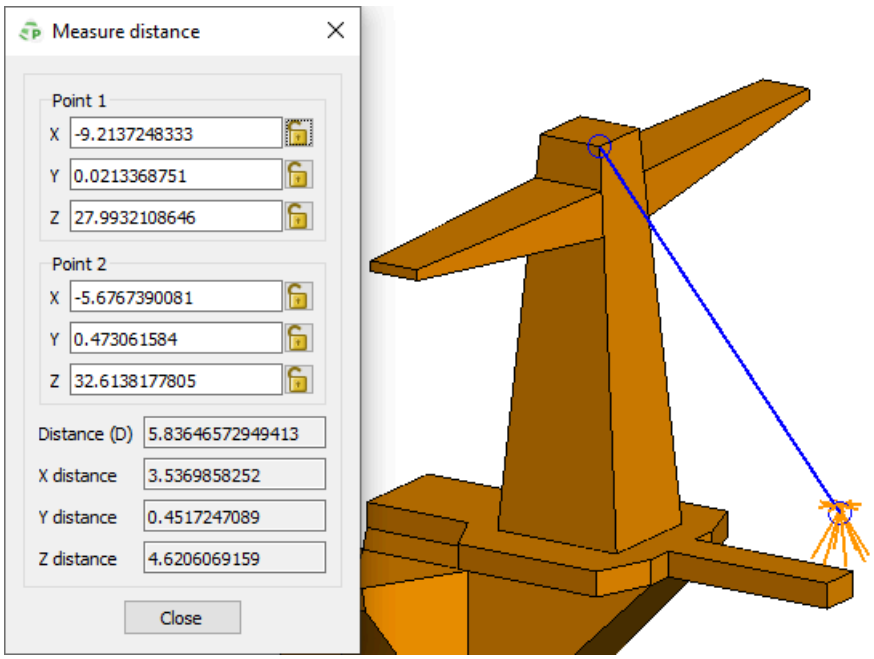


*Figure 367: The **Measure distance** tool.*

2.36.2 Measuring an Angle

Use the angle measuring tool to measure or validate the angle (in degrees) between three points in a model.

1. On the **Tools** tab, in the **Tools** group, click the **Measure Angle** icon.
2. Under **Point1**, use Ctrl+Shift+left click to snap to points (for example, named points, geometry points, geometry face centre, geometry edge centre, mesh vertices and grid).
3. Repeat Step 2 for **Point 2**.
4. Repeat Step 2 for **Point 3**.
   The angle in degrees is displayed in the **Angle (degrees)** field.
5. Click **Close** to close the dialog.

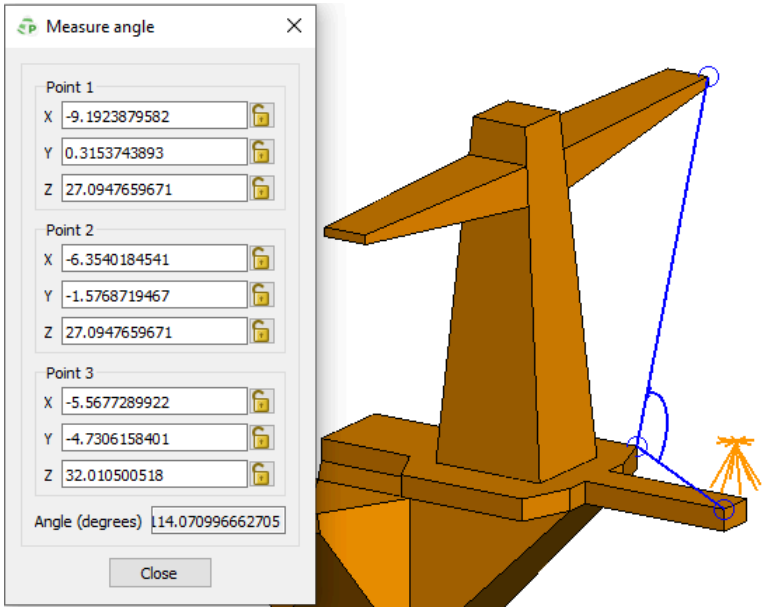


*Figure 368: The **Measure angle** dialog.*

2.36.3 Performing Calculations

Use the calculator to perform calculations using functions, predefined variables, user-defined variables and named points.

> **Tip:** Use the tool to evaluate variables and named points without modifying the model.

1. On the **Tools** tab, in the **Tools** group, click the **Calculator** icon.

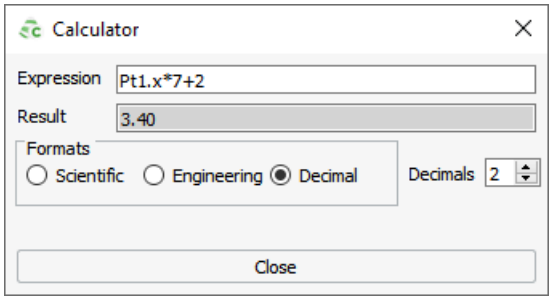


*Figure 369: The **Calculator** dialog.*

2. In the **Expression** field, you can add expressions that consist of functions, predefined variables, user-defined variables or named points.
3. [Optional] Change the number format for the result. Under **Formats**, select one of the following:
   - **Scientific**
   - **Engineering**
   - **Decimal**
4. [Optional] In the **Decimals** box, type the value or click the up or down arrows to specify the number of decimals.
5. Click **Calculate** or Enter to evaluate the expression and display the result in the **Result** field.
6. Click **Close** to close the dialog.

Related reference
Math Functions in CADFEKO
Predefined Variables

2.36.4 Exporting an Image

Export an image of the active view to file.

1. On the **Tools** tab, in the **Image Tools** group, click the **Export image** icon.

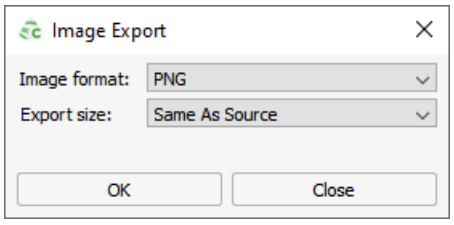


*Figure 370: The **Export image** dialog.*

2. Select a view to export.
3. From the **Image format** drop-down list, select one of the following:
   - **PNG**
   - **BMP**

- **CUR**
- **ICNS**
- **JPG**
- **PBM**
- **PGM**
- **TIF**
- **WBMP**
- **WEBP**
- **PDF**
- **EPS**
- **EMF**

4. From the **Export size** drop-down list, select one of the following:
   - **Same as source**
   - **QQVGA (160x120)**
   - **QVGA (320x240)**
   - **VGA (640x480)**
   - **SVGA (800x600)**
   - **XGA (1024x768)**
   - **SXGA (1280x1024)**
   - **Custom**
5. Click **OK**.
   The **Image export file name** dialog is displayed.
6. In the **File name** field, specify the file name of the exported file.
7. In the **Save as type**, specify the file type of the exported file.
8. Click **Save** to export the active view to file and to close the dialogs.

2.37 Model Tree Icons

View the list of icons that may be found in the model tree.

Icon	Definition
	Imported CAD body or a part that was converted to a primitive.
	Surface body (for example, created with a face copy or explode)
	Curve (edge/wire) body (for example, created with an edge copy or explode.)
	A mesh part in the model.
	The part/region/face/edge/wire contains faults.
	The target from which an object was subtracted from.
	This part contains a dielectric medium.
	This part contains anisotropic regions.
	This part is locked.
	This part is excluded.
	The active optimisation search.
	A protected model.
	Local mesh settings (that can be applied to a part) is specified.
	The default cutplanes.
	An adaptive mesh refinement is added to the model.
	A point mesh refinement is added to the model or a point refinement that forms part of an adaptive mesh refinement.
	A polyline mesh refinement is added to the model.

Icon	Definition
	Part was repaired.
	Part was repaired and faces sewn.

2.38 Details Tree Icons

View the list of icons that may be found in the details tree.

Icon	Definition
	This face lies on a dielectric region.
	This item is suspect — it could not be mapped.
	Local mesh properties set on regions, faces or edges.
	Local wire radius.
	• For wires and surfaces, the core medium. • For tetrahedra, the medium.
	The layered medium applied as a coating.
	• For surfaces, the medium on the normal side. • For wires, the surrounding medium.
	Only used for surfaces; the medium on the rear (opposite to normal) side.
	The solution method applied to the wire, edge, face or region.
	A face of the selected part.
	A wire or edge of the selected part.
	A mesh triangle of the selected part.
	A mesh segment of the selected part.
	A mesh tetrahedron of the selected part.
	A UTD mesh plate of the selected part.
	A UTD mesh cylinder of the selected part.

2.39 Math Functions in CADFEKO

View the list of supported math functions.

Shortcut Key	Description
ABS	Absolute value
ARCCOS	Arccosine (radians)
ARCCOT	Arccotangent (radians)
ARCSIN	Arcsine (radians)
ARCTAN	Arctangent (radians)
ATAN2	Arctangent of Y/X in the range -PI to PI Syntax: `ATAN2(X,Y)`
CEIL	Smallest integer value that is equal or greater than the argument
COS	Cosine (radians)
COSH	Hyperbolic cosine (radians)
COT	Cotangent (Radians)
DEG	Convert radians to degrees
EXP	Exponential function
FLOOR	Largest integer value that is equal to or smaller than the argument
FMOD	Resulting reminder of a/b. Syntax: `FMOD(a, b)`
IMAG	Determines the imaginary component
LN	Computes the natural logarithm Syntax: `LN(a)` Example: `LN(trace1)` where `trace1` is a trace in the result palette.
LOG	Computes logarithm base 10
MAX	Returns the largest of the two arguments

Shortcut Key	Description
	Syntax: `MAX(a, b)`
MIN	Returns the smallest of the two arguments Syntax: `MIN(a, b)`
PT	Point creation Syntax: `PT(x, y, z)`
RAD	Convert degrees to radians
SIN	Sine (Radians)
SINH	Hyperbolic sine (Radians)
SQRT	Square root
STEP	Step function Syntax: `STEP(X) = 0 for X <= 0` and `STEP(X) = 1 for X > 0`
TAN	Tangent (Radians)
TANH	Hyperbolic tangent (radians)

Predefined Variables

Table 18: List of predefined variables.

Variable	Description
c0	The speed of light in free space in m/sec.
eps0	The permittivity of free space in F/m.
mu0	The permeability of free space in H/m.
pi	The mathematical constant (Ludolph’s number).
zf0	The characteristic impedance of free space in Ohm.

2.40 Files Generated by CADFEKO

View the files associated and generated by CADFEKO.

Table 19: Files generated by CADFEKO

Argument	Description
`.cfx`	Contains the meshed and/or unmeshed CADFEKO model as well as the calculation requests. If an optimisation was run, a model will be created with the optimum values with a `_optimum` suffix.
`.cfm`	Contains information regarding the mesh of the CADFEKO model.
`.pre`	A `.pre` file is created when the CADFEKO model is saved.
`.fek`	The `.fek` file is created when running PREFEKO and it contains the geometry (solver mesh) of the CADFEKO model. This file can be opened in POSTFEKO to view the geometry and the calculation requests (for example, the near field request points are displayed if a near field calculation was requested). The mesh from a `.fek` file may be imported into CADFEKO.
`.opt`	An `.opt` file is created when optimisation settings have been defined for the CADFEKO model.
`.pfg`	Contains the relevant information used during optimisation (in conjunction with the `.pre` and `.cfx` files).

2.41 Default Shortcut Keys

View the default shortcut keys available for CADFEKO for faster and easier operation of CADFEKO.

Keyboard shortcut keys help you to save time accessing actions that you perform regularly. The shortcut key or key combination is displayed in the keytip that is displayed when you hover the mouse over the action on the ribbon.

Shortcut Key	**Description**
Alt+0	Run CADFEKO.
Alt+1	Run EDITFEKO.
Alt+2	Run PREFEKO.
Alt+3	Run POSTFEKO.
Alt+4	Run Solver.
Alt+6	Run OPTFEKO.
Alt+8	Open the Feko terminal.
General Editing	
F2	Rename selected item.
F9	Create workplane.
Del	Delete selected item.
Shift+Ins	Paste clipboard text.
Ctrl+Ins	Copy selected text.
Shift+Del	Cut selected text.
Ctrl+A	Select all entities (edge, wire, face or region) of the same type in the collection[47].
Ctrl+Shift+A	Select all entities (edge, wire, face or region) of the same type in the model.
Ctrl+C	Copy selected text / image.

47. For example, in the model tree, a collection can be geometry, meshes, ports, meshing rules, cutplanes and solution settings. In the details tree, a collection can be wires, edges, faces and regions.

Ctrl+E	Export image.
Ctrl+F	Edit project tree filter.
Ctrl+H	Show / Hide selected geometry and requests.
Ctrl+K	For mesh parts or solution items which allow multiple instances, create copies of the selected items. For geometry items (at any level) create new root-level parts as copies of the selected items.
Ctrl+M	Modify global mesh settings.
Ctrl+N	Create new model.
Ctrl+L	Open the component library.
Ctrl+3	Create new 3D view.
Ctrl+O	Open model.
Ctrl+S	Save model.
Ctrl+V	Paste.
Ctrl+X	Cut selected text.
Q+C	Select the smallest loop of edges containing the currently selected edges.
Ctrl+Y	Redo model creation / modification.
Ctrl+Z	Undo model creation / modification.
#	Create variable.
Ctrl+Shift	Point entry.
Alt+S	Search bar.
Create arcs / curves	
V,1	Create a straight line.
V,2	Create a polyline.
V,3	Create a fitted spline.
V,4	Create a Bézier curve.

V,5	Create an analytical curve.
A,1	Create an elliptic arc.
A,2	Create a parabolic arc.
A,3	Create a hyberbolic arc.
A,4	Create a helix.
Create Solids	
C,1	Create a cuboid.
C,2	Create a flare.
C,3	Create a sphere.
C,4	Create a cylinder.
C,5	Create a cone.
Create Surfaces	
S,1	Create a rectangle.
S,2	Create a polygon.
S,3	Create an ellipse.
S,4	Create a paraboloid.
S,5	Create a NURBS surface.
Transform / Modify	
B,1	Subtract selected object from another object.
B,2	Intersect the selected geometries.
B,3	Split selected items along a plane.
B,4	Stitch selected face parts together.
E,1	Spin selected items around a specified axis.
E,2	Sweep selected items in a specified direction.
E,3	Sweep selected item along a path.
E,4	Connect two profiles to form a loft surface or solid.
R	Re-evaluate the geometry tree.

U	Union the selected parts.
View	
F1	Context-sensitive help for the dialog / window that has focus.
F5	Zoom to extents.
0	Restore view.
2	Bottom view.
4	Left view.
5	Front view.
Ctrl+5	Back view.
6	Right view.
8	Top view.
3D View / Schematic View Interaction	
F5	Zoom to extents.
Shift & hold while scrolling mouse wheel.	Slow zoom (3D view).
Scroll mouse wheel.	Zoom (3D view).
Click & drag with middle mouse button.	Panning (3D view, schematic view).
Ctrl & click / drag.	Panning (3D view).
Left click & drag the mouse.	Rotation (3D view).
+	Zoom in (3D view, schematic view).
-	Zoom out (3D view, schematic view).
R	Rotate element (schematic view).
Script Editor	
Ctrl+N	New empty script.
Ctrl+O	Open script.
Ctrl++	Zoom in.
Ctrl+-	Zoom out.

Ctrl+G	Goto line.

POSTFEKO

3

POSTFEKO, the Feko post processor, is used to display the model (configuration and mesh), results on graphs and 3D views.

This chapter covers the following:

3.1 Introduction to POSTFEKO

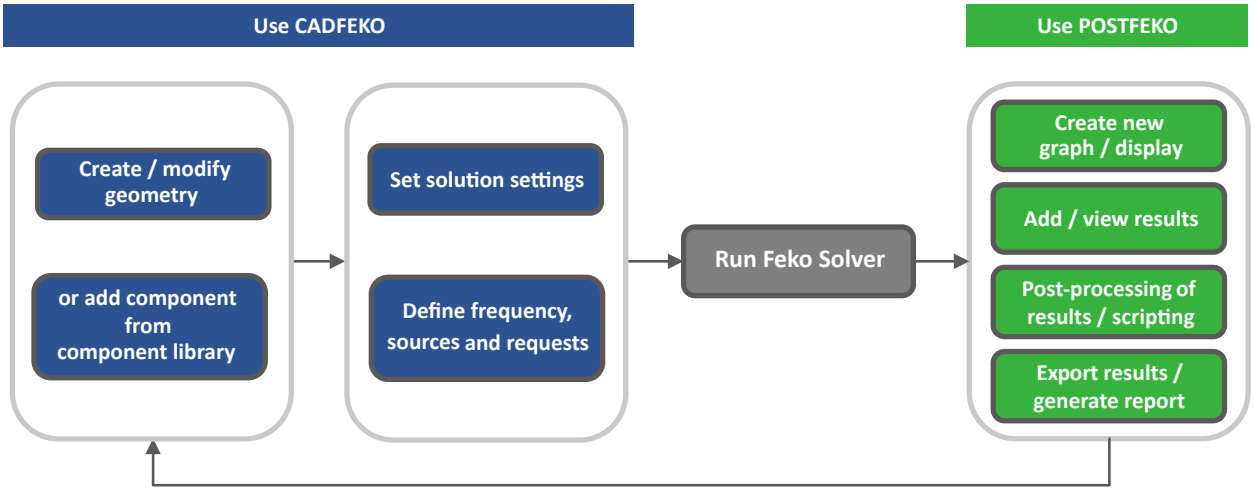
Use POSTFEKO to validate meshed geometry and analyse and post-process results.

POSTFEKO is the component that allows you to verify that your model is constructed and configured correctly before starting a simulation and analyse the results after the simulation completes. The POSTFEKO component is particularly useful to verify models created using EDITFEKO, but it is just as relevant for CADFEKO model verification.

Result post-processing and analysis is the primary function of POSTFEKO. Once a model has been simulated, POSTFEKO can be used to display and review the results. It is easy to load multiple models in a single session and compare them on 3D views, Cartesian graphs, Smith charts, polar graphs and surface graphs. Various measurement and other data formats are supported for comparison to the simulated results. A powerful scripting interface makes it easy to post-process results, automate repetitive tasks and create plug-in extensions that customise the interface and experience.

3.1.1 Feko Components and Workflow

View the typical workflow when working with the Feko components.



POSTFEKO

Create a new graph or 3D view and add results of the requested calculations on a graph or 3D view. Results from graphs can be exported to data files or images for reporting or external post-processing. Reports can be created that export all the images to a single document or a custom report can be created by configuring a report template.

After viewing the results, it is often required to modify the model again in CADFEKO and then repeat the process until the design is complete.

3.1.2 Launching POSTFEKO (Windows)

There are several options available to launch POSTFEKO in Windows.

Launch POSTFEKO using one of the following workflows:

- Open POSTFEKO using the Launcher utility.
- Open POSTFEKO by double-clicking a `.pfs`, `.fek` or `.bof` file.
- Open POSTFEKO from other components, for example, from inside CADFEKO and EDITFEKO.

> **Note:** If the application icon is used to launch POSTFEKO, no model is loaded and the start page is shown. Launching POSTFEKO from other Feko components, automatically loads the model.

Related tasks
Opening the Launcher Utility (Windows)

3.1.3 Launching POSTFEKO (Linux)

There are several options available to launch POSTFEKO in Linux.

Launch POSTFEKO using one of the following workflows:

- Open POSTFEKO using the Launcher utility.
- Open a command terminal. Launch POSTFEKO using the absolute path to the executable in the installation, for example:

```
/home/user/2024.1/altair/feko/bin/postfeko
```

- Open a command terminal. Source the "initfeko" script using the absolute path to it, for example:

```
. /home/user/2024.1/altair/feko/bin/initfeko
```

Sourcing `initfeko` ensures that the correct Feko environment is setup. Type `postfeko` and press Enter.

> **Note:** Take note that sourcing a script requires a dot (".") followed by a space (" ") and then the path to `initfeko` in order for the changes to be applied to the current shell and not a sub-shell.

Related tasks
Opening the Launcher Utility (Linux)

3.1.4 Command Line Arguments for Launching POSTFEKO

POSTFEKO can be called via the command line. Use command line arguments to pass configuration information to POSTFEKO.

If POSTFEKO is launched and a model (or set of models) is specified, the model is added to a new project (or sessions). Without any models specified, POSTFEKO will start and display the start page.

Command-line options:

```
postfeko [SESSION] [FILES] [OPTIONS]
```

SESSION
: A single session (`.pfs`) may be specified that may or may not exist

FILES
: Multiple model (`.fek`) files or result (`.bof`) files may be specified. Model files result in a 3D view being created automatically that displays the first configuration of the model.

OPTIONS

-h, --help
: Displays the help message.

--version
: Print the version information and then exit.

--non-interactive
: Special execution mode for running automation scripts without user interaction.

--run-script SCRIPTFILE
: Specifies an automation script to load and run.

--configure-script CONFIGSTRING
: Executes the string `CONFIGSTRING` before running the script specified in `SCRIPTFILE`. This options is only used with the "non-interactive" option.

--file-info [=OUTPUTFORMAT] SESSION
: Display the POSTFEKO version used to create the file.

```
postfeko startup.pfs --file-info[48]
```

```
postfeko startup.pfs --non-interactive --file-info |more[49]
```

```
postfeko startup.pfs --non-interactive --file-info > versions.txt[50]
```

48. Opens a dialog and displays the version information.
49. Writes the version information out to standard output stream (stdout).
50. Redirects the version information to the specified file.

=OUTPUTFORMAT

Optional argument that is used to specify the output format. If the argument is set to `xml`, version information is written out in XML format. XML will only be output to stdout, and only if `--non-interactive` was also specified.

```
postfeko startup.pfs --file-info=xml --non-interactive | more[51]
```

3.1.5 Start Page

The Feko start page is displayed when starting a new instance (no models are loaded) of CADFEKO, EDITFEKO or POSTFEKO.

The start page provides quick access to **Open a session**, **Open a model** and a list of **Recent projects**.

Links to the documentation (in PDF format), introduction videos and website resources are available on the start page. Click the icon to launch the Feko help.

Figure 371: The POSTFEKO start page.

51. Writes the version information in XML format in non-interactive mode, displaying the content one screen at a time.

3.2 Quick Tour of the POSTFEKO Interface

View the main elements and terminology in the POSTFEKO graphical user interface (GUI).

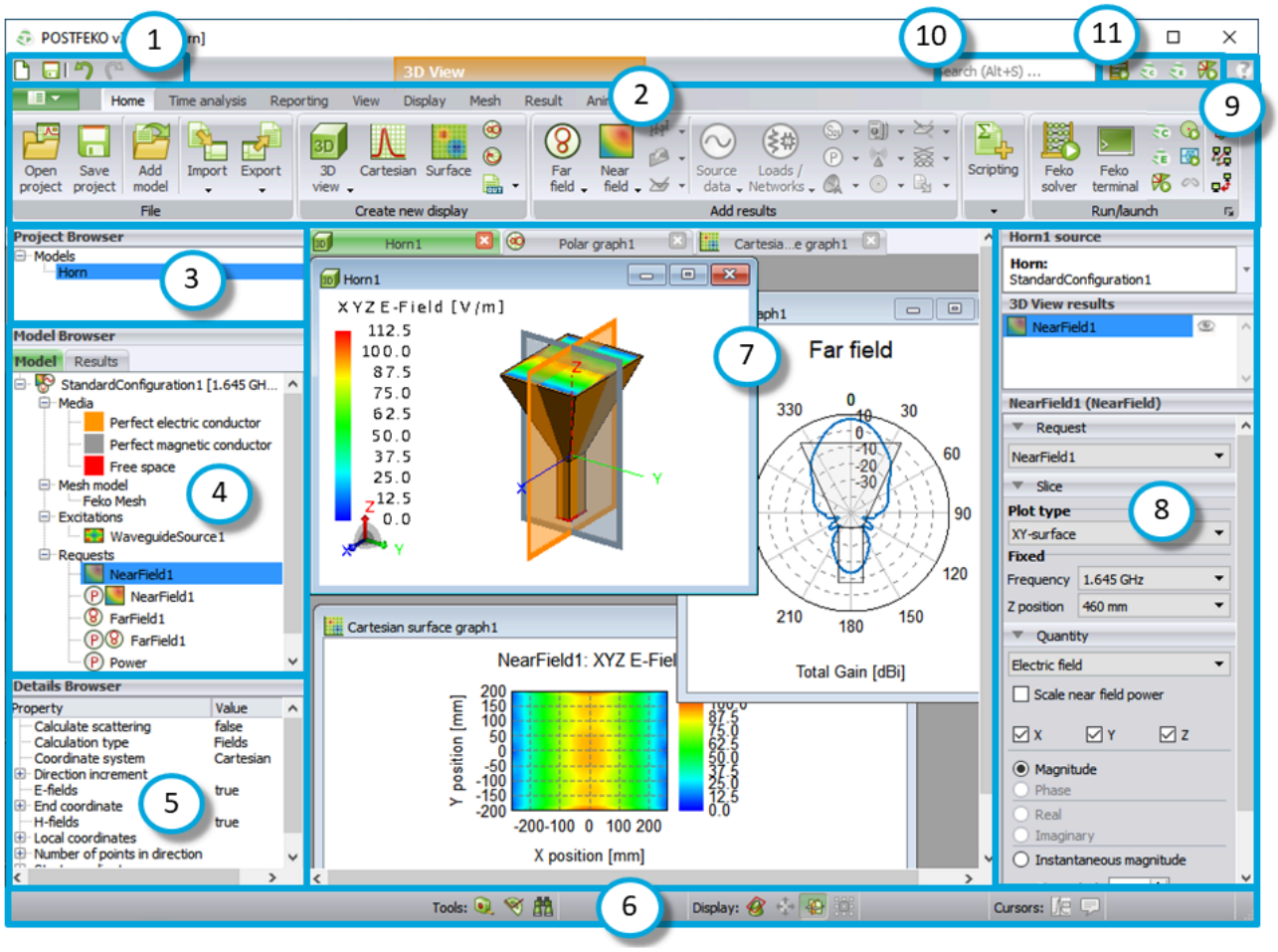


Figure 372: The POSTFEKO window.

1. Quick Access Toolbar
2. Ribbon
3. Project Browser
4. Model Browser
5. Details Browser
6. Status Bar
7. 3D Views and Graphs
8. Result Palette
9. Help
10. Search Bar
11. Application Launcher
12. Application Menu

3.2.1 Quick Access Toolbar

The quick access toolbar is a small toolbar that gives quick access to actions that are often performed.

The toolbar is located at the top-left corner of the application window, just below the title bar. It allows you to create a new model, open a model, save a model, undo a model operation or redo a model operation using fewer mouse clicks for a faster workflow. The actions available on the quick access toolbar are also available via the ribbon.

3.2.2 Ribbon

The ribbon is a command bar that groups similar actions in a series of tabs.

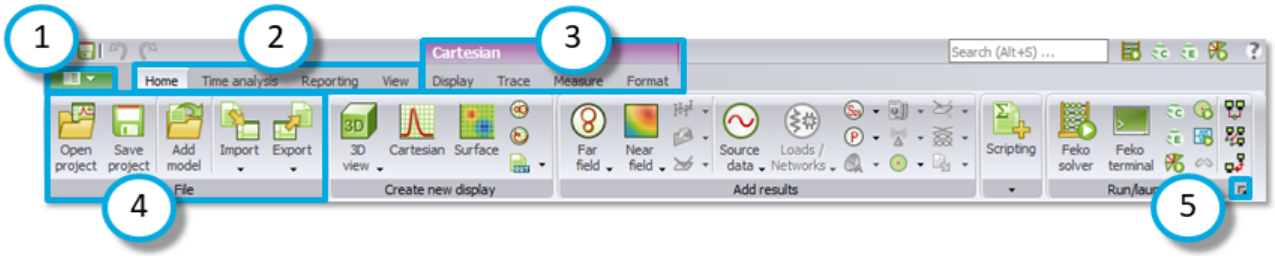


Figure 373: The ribbon in POSTFEKO.

1. Application menu
The application menu button is the first item on the ribbon. When the application menu drop-down button is clicked, the application menu is displayed. The menu allows saving and loading of models, import and export options as well as giving access to application-wide settings and a recent file list.

2. Core tabs
A tab that is always displayed on the ribbon, for example, the **Home** tab and **Reporting** tab.

The **Home** tab is the first tab on the ribbon and contains the most frequently used commands for quick access.

3. Contextual tab sets
A tab that is only displayed in a specific context.

For example, the **Cartesian** contextual tab set contains the **Format** contextual tab. Contextual tabs appear and disappear as the selected items such as a view or item on a view, change.

4. Ribbon group
A ribbon tab consists of groups that contain similar actions or commands.

5. Dialog launcher
Click the dialog launcher to launch a dialog with additional and advanced settings that relate to that group. Most groups don't have dialog launcher buttons.

Keytips
A keytip is the keyboard shortcut for a button or tab that allows navigating the ribbon using a keyboard (without using a mouse). Press F10 to display the keytips. Type the indicated keytip to open the tab or perform the selected action.

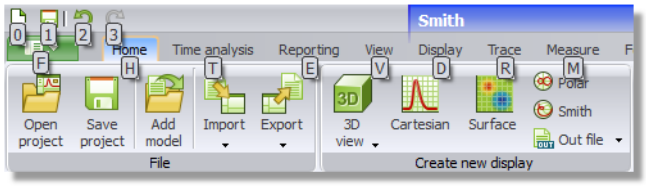


Figure 374: An example of keytips.

Application Menu

The application menu is similar to a standard file menu of an application. It allows saving and loading of models, print functionality and gives access to application-wide settings.

When you click on the application menu drop-down button, the application menu, consisting of two panels, is displayed.

The first panel gives you access to application-wide settings, for example:

- Creating a new model.
- Open models, open a project, saving a project and closing a project.
- Import
- Export
- Print
- Check for updates
- Settings
  - Preferences
  - 3D mouse sensitivity setting
  - Rendering options (for example, rendering mode and transparency mode)
  - Component launch options
- Feko help
- About
  - Version information about POSTFEKO
  - Information about Altair HyperWorks Products
  - Information about third-party libraries
- Exit

The second panel consists of a recent file list and is replaced by a sub-menu when a menu item is selected.

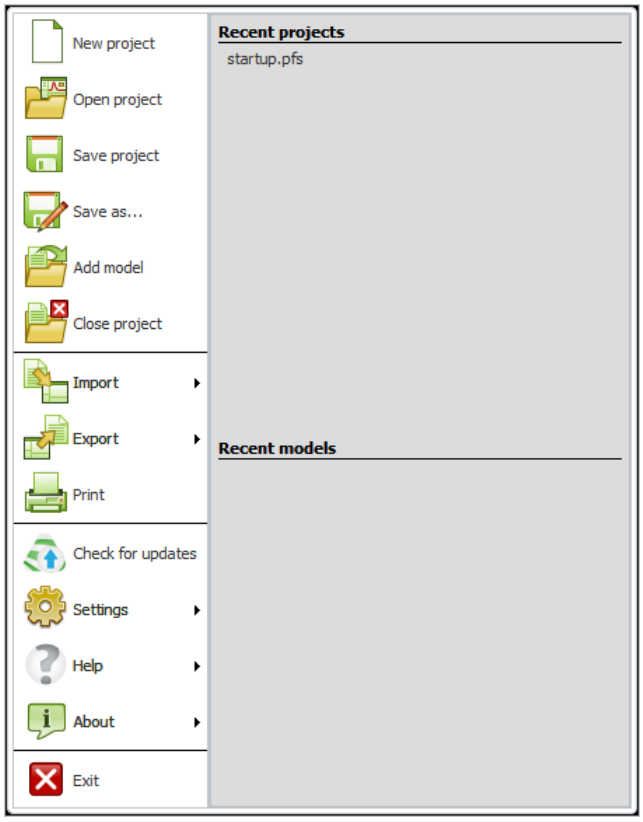


Figure 375: The application menu in POSTFEKO.

Home Tab

The **Home** tab is the first tab on the ribbon and contains the most frequently used operations.

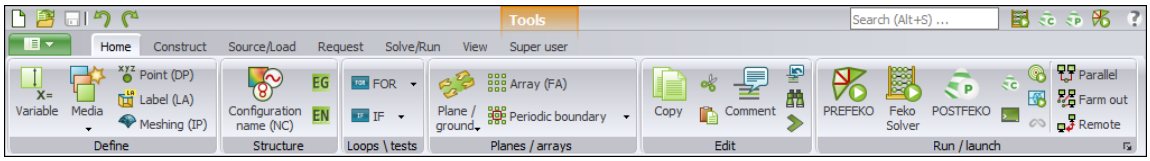


*Figure 376: The **Home** tab in POSTFEKO.*

3.2.3 Project Browser

The project browser is a panel that lists the models loaded in the current project, imported data, stored data and scripted data.

Collapse the project browser to expand the 3D view. On the **View** tab, in the **Show** group, click the **Project** icon.

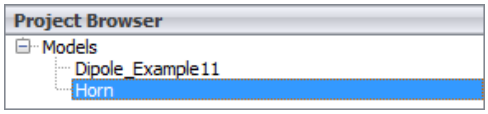


Figure 377: Project browser is showing the model file for the current session.

3.2.4 Model Browser

The model browser is a panel that organises the model information of the selected model in the project browser, into two separate tabs.

The model browser is separated into two tabs.

- The **Model** tab lists the model information and results for the selected model.
- The **Results** tab lists the results and solution information.

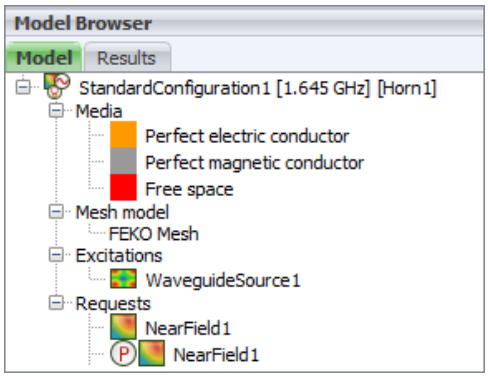


Figure 378: Model browser is showing the model information for the selected model.

3.2.5 Details Browser

The details browser is a panel that shows in-depth detail for the selected item in the model browser.

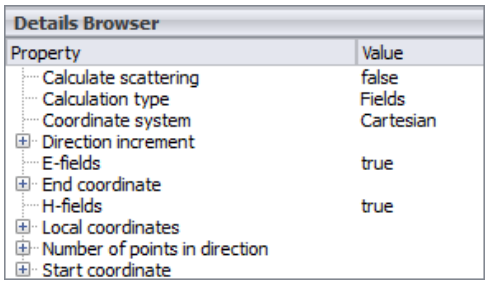


Figure 379: Details browser is showing the detail for a selected item in the model browser.

Tip: View the solution information for the selected model.

On the model browser, click **Solution information** to view:

- memory per process
- total CPU-time
- total runtime

3.2.6 Status Bar

The status bar is a small toolbar that gives quick access to general display settings, tools, and graph cursor settings.

The status bar is located at the bottom-right of the application window. Options on the status bar are also available on the ribbon, but since the status bar is always visible, they are easily accessible no matter which ribbon tab is selected.

3.2.7 3D Views and Graphs

3D views are used to display mesh, solution settings and interact with the model as well as view 3D results. Graphs are used to display 2D results.

3.2.8 Result Palette

The result palette is a panel that gives access to options that control the data in the 3D view or graph.

Collapse the result palette to expand the 3D view. On the **View** tab, in the **Show** group, click the **Palette** icon.

If different types of results are loaded, then the result palette layout and options update according to the selected data. If multiple results are simultaneously selected, settings common to all the results are available.

3.2.9 Help

The **Help** icon provides access to the Feko documentation.

Press F1 to access context-sensitive help. The context-sensitive help opens the help on a page that is relevant to the selected dialog, panel or view.

Tip: When no help context is associated with the current dialog or panel, the help opens on the main help page that allows you to navigate the documentation or search in the documentation for relevant information.

3.2.10 Search Bar

The search bar is a single-line text field that allows you to enter search terms and find relevant information in the GUI or the documentation.

The search bar is located at the top-right of the application window.

Tip:

- Enter a search term in the search bar to populate a drop-down list of actions as well as the location of the action on the ribbon or context menu.
- Click an item in the list to execute the action.
- Partial searches are supported.
- Search the documentation.

3.2.11 Application Launcher

The application launcher toolbar is a small toolbar that provides quick access to other Feko components.

3.2.12 Application Menu

The application menu is similar to a standard file menu of an application. It allows saving and loading of models, print functionality and gives access to application-wide settings.

When you click on the application menu drop-down button, the application menu, consisting of two panels, is displayed.

The first panel gives you access to application-wide settings, for example:

- Creating a new model.
- Open models, open a project, saving a project and closing a project.
- Import
- Export
- Print
- Check for updates
- Settings
  - Preferences
  - 3D mouse sensitivity setting
  - Rendering options (for example, rendering mode and transparency mode)
  - Component launch options
- Feko help
- About
  - Version information about POSTFEKO
  - Information about Altair HyperWorks Products

    - Information about third-party libraries
- Exit

The second panel consists of a recent file list and is replaced by a sub-menu when a menu item is selected.

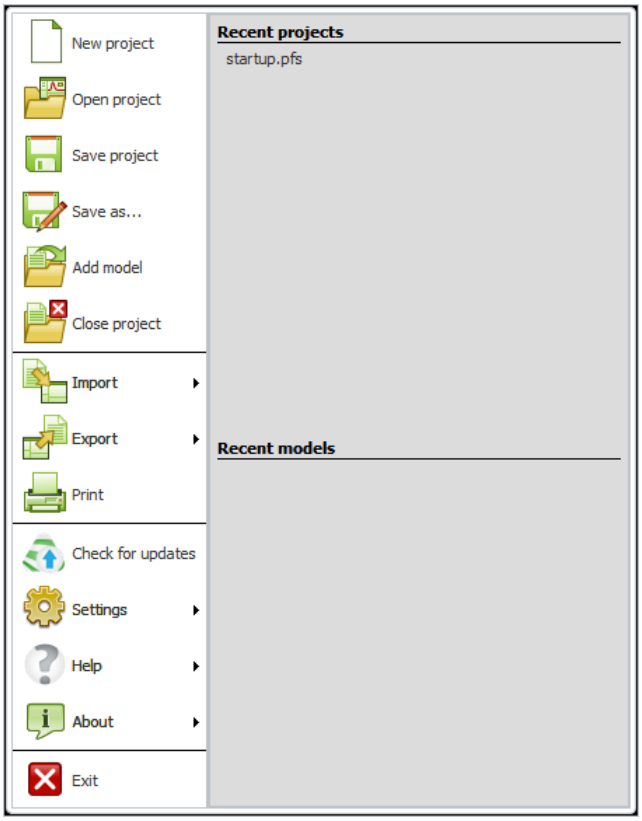


Figure 380: The application menu in POSTFEKO.

3.2.13 Scripting

Use the application programming interface (API) to control CADFEKO from an external script.

Scripting allows repetitive or complex tasks to be performed in a script that would have taken a long time to perform manually. Scripts are created and edited in the script editor or scripts can be recorded (macro recording) by enabling the recording and then performing the actions in the graphical interface. The recorded script can be modified to perform a more complex task. Scripts that are used regularly can be added to the ribbon providing easy access and hiding the complexity of the script. Forms (dialogs) can be created in the scripting environment that obtain input from the script user without having to edit the script.

Script Editor

The script editor allows you to create scripts based on the Lua language to control CADFEKO, POSTFEKO and other applications as well as manipulation of data to be viewed and analysed further in POSTFEKO.

On the **Home** tab, in the **Scripting** group, click the  **Script editor** icon.

The script editor includes the following IDE (integrated development environment) features:

1. Syntax highlighting.
2. Intelligent code completion.
3. Indentation for blocks to convey program structure, for example, loops and decision blocks in scripts.
4. Use of breakpoints and stepping in scripts to debug code or control its execution.
5. An active console to query variables or execute simple commands.

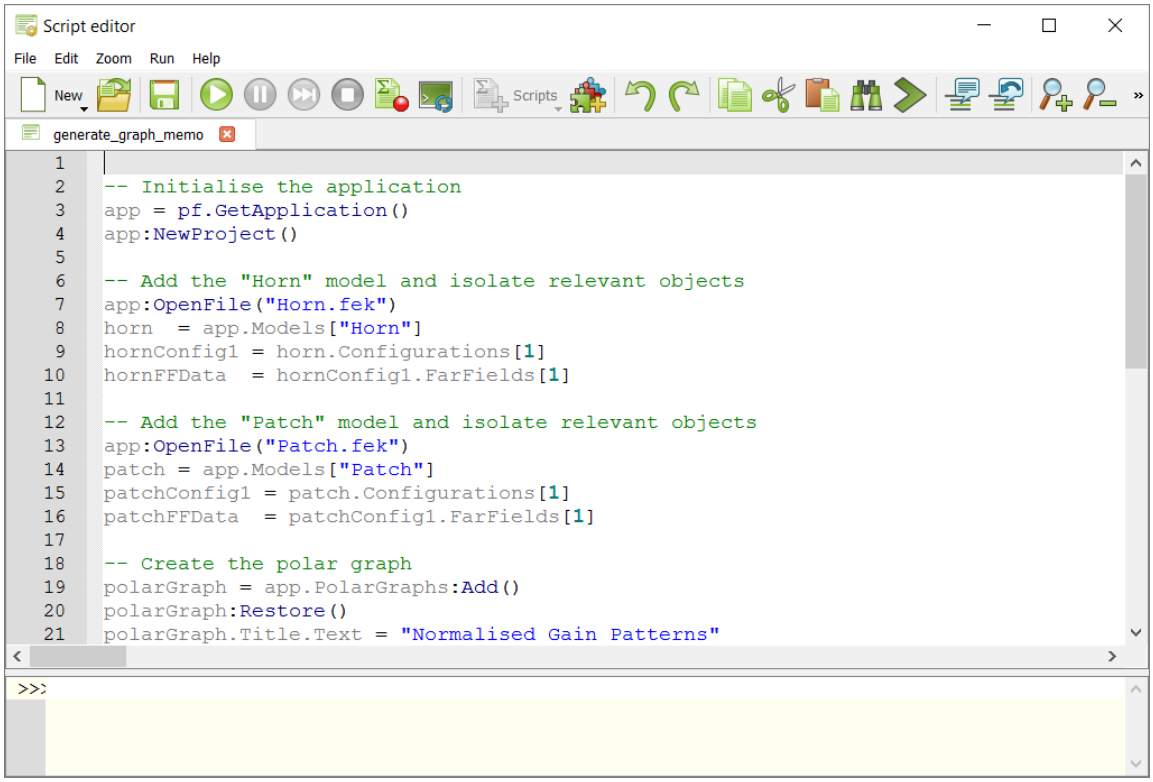


Figure 381: The script editor in POSTFEKO.

Macro Recording

Use macro recording to record actions in a script. Play the script back to automate the process or view the script to learn the Lua-based scripting language by example. Macro recording allows you to perform repetitive actions faster and with less effort.

On the **Home** tab, in the **Scripting** group, click the **Record Macro** icon.

Application Macros

An application macro is a reference to an automation script, an icon file and associated metadata.

Application macros are available directly or can be added, removed, modified or executed from the application macro library.

> **Tip:** A large collection of application macros are available in CADFEKO and POSTFEKO.

On the **Home** tab, in the **Scripting** group, click the **Application macro** icon.

Related concepts
CADFEKO Application Macros
POSTFEKO Application Macros

3.3 Preferences

POSTFEKO has various default settings that you can configure to customise it to your preference.

On the application menu, click **Settings** > **Preferences**. The settings can be reset to the default settings at any time, restoring the settings to the state of a new installation.

Many of the settings are applied immediately, but some of the settings such as 3D view font changes and rendering options require the application to be restarted before the changes take effect.

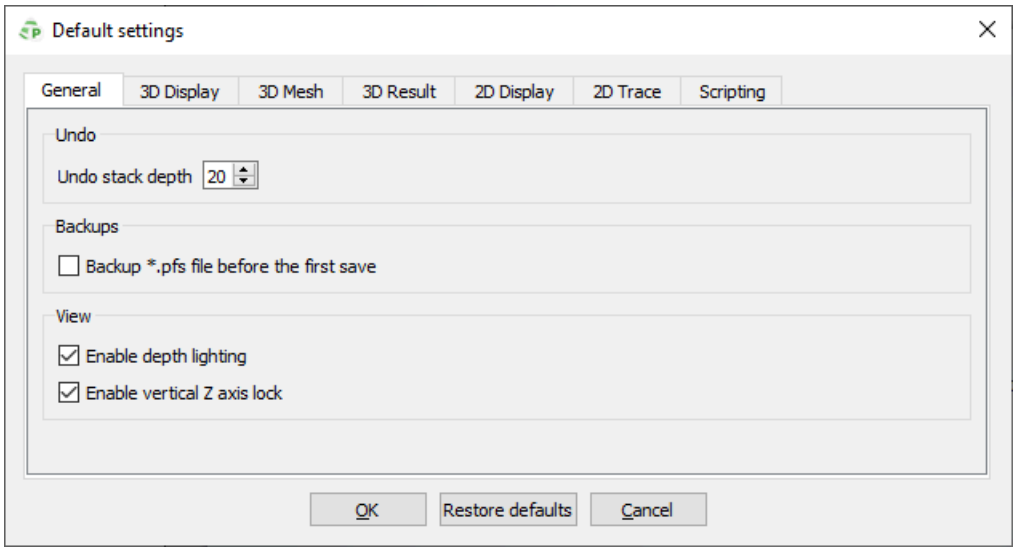


*Figure 382: The **Default settings** dialog.*

3.4 Rendering Options

A number of rendering options are available to ensure that 3D models and graphs (containing a large number of sample points) are rendered efficiently.

On the application menu, click **Settings** > **Rendering options**.

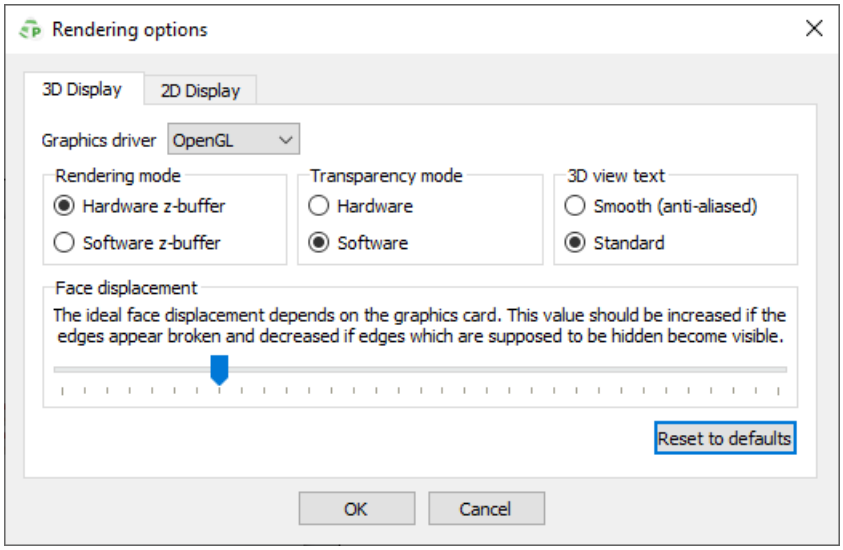


*Figure 383: The **Rendering options** dialog.*

3D Display

Graphics driver
: Specify the graphics driver used to render 3D graphics. The following options are available: **Auto select**, **OpenGL**, **OpenGL 2.0**, **DirectX 9**, **DirectX 11** and **Software**.

Rendering mode
: Z-buffering is an algorithm used in 3D graphics to determine if an object (or part of the object) is visible or if it is hidden from view and is used to increase rendering efficiency. These calculations can be done using the GPU (hardware) or using the CPU (software).

Transparency mode
: The transparency rendering of objects can be done using the GPU (hardware) or using CPU (software).

3D view text
: Text in the 3D view can be rendered using either **Smooth (anti-aliased)** or **Standard**. Anti-aliasing of text results in a font being displayed with smooth curves and makes it appear less jagged.

Note: Rendering settings changes are only applied to new views.

2D Display

Enable OpenGL rendering

Select the **Enable OpenGL rendering** check box to accelerate the rendering of graphs and graph manipulation (for example, zooming, restoring a view) for graphs containing thousands of sample points.

3.5 Model and Project Basics

You can add a model, open an existing project and save the project.

3.5.1 Adding a Model or Project

Load a `.fek` file (single file or multiple files) or load a single POSTFEKO session file that contains the settings, views and references to result files that were present at the time of save.

Load a model.

- Open a single or multiple `.fek` files. On the **Home** tab, in the **File** group, click the **Add model** icon.
- Open saved session (saved project). On the **Home** tab, in the **File** group, click the **Open project** icon.

> **Tip:** A model or session can also be opened from the start page.

3.5.2 Saving a Project

Store the view settings, views and references to result files to a `.pfs` file to reopen later.

On the **Home** tab, in the **File** group, click the **Save Project** icon.

3.5.3 Large Models

When a model containing more than 500 000 elements is opened, it may become difficult to work with the 3D model due to memory requirements for the 3D rendering and visualisation.

Should such a model be opened, you are prompted to select whether the model is to be displayed in the 3D view or only load the model into memory. You can still view and process the results in 3D views or graphs, just without any geometry visualisation. The model can be loaded at a later stage from the context menu in the project browser.

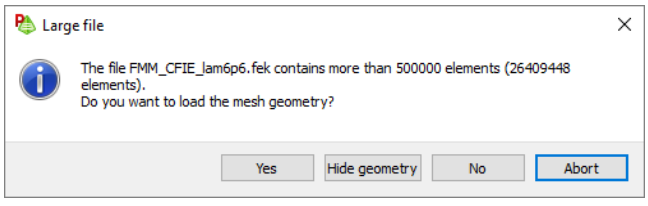


Figure 384: An information message is stating that the model contains more than 500 000 mesh elements.

3.6 Data Import

Import text files, native data files and Touchstone format files.

Multiple selected files located in the same folder can be imported in a single import, but only a single custom data file can be imported at a time.

Data that was imported into POSTFEKO can be added to a graph in the same manner as other any result. The project browser contains an entry for each import under **Imported files**. Imports can be deleted from the project should they no longer be required.

Saving a project with imported results stores it as part of the POSTFEKO session file (`.pfs`) file.

3.6.1 Supported File Formats for Import

POSTFEKO supports the import of native file formats as well as Touchstone file format.

On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list, select the file format to import.

The following file formats are supported for import:

Feko far field (.ffe)*
: The `.ffe` file is imported automatically. No further user input is required.

Feko near field (.efe, *.hfe)*
: When a near field is imported, specify whether an **Electric near field file (*.efe)** file or **Magnetic near field (*.hfe)** file or both files are to be imported.

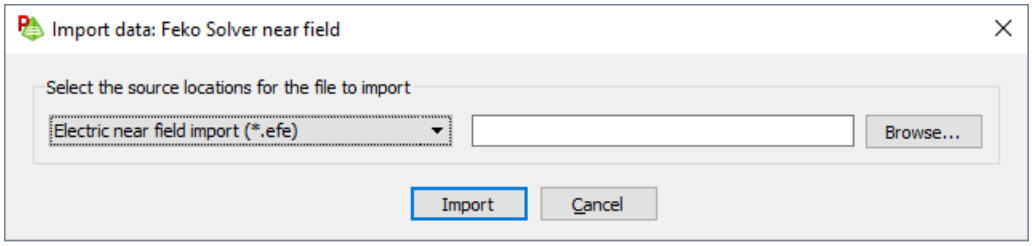


*Figure 385: The **Import data: Feko Solver near field** dialog.*

> **Note:** The `.efe` file and the corresponding `.hfe` file must have identical file names.

Touchstone (.snp)*
: The `.snp` file is imported automatically. No further user input is required.

Report template (.xml)*
: A report template in XML format can be imported.

POSTFEKO graph file (.pfg)*
: This file format contains the relevant information used during optimisation. This file type works in conjunction with the `.pre` and `.cfx` files.

Custom data file
When importing custom data, an import template needs to be defined.

Importing Custom Data

Import a single custom data file by defining the import template.

1. On the **Home** tab, in the **File** group, click the **Import** icon. From the drop-down list, click the **Custom Data File** icon.
2. Browse to the location of the file and select a custom data file.
3. Under **Delimiter**, select one of the following delimiters that separate the columns of data.
   - **Tab**
   - **Space**
   - **Comma**
   - **Other**

The data file may contain lines of text that are not part of the data to be imported.

4. In the **Start reading file at (line number)** field, enter the line number at which data should be imported.
5. In the **Specify number of lines to read** field, enter the number of data to read.
6. If the data contains column title, select the **Data contains column titles** check box.
7. Click **Next** to continue with the template.

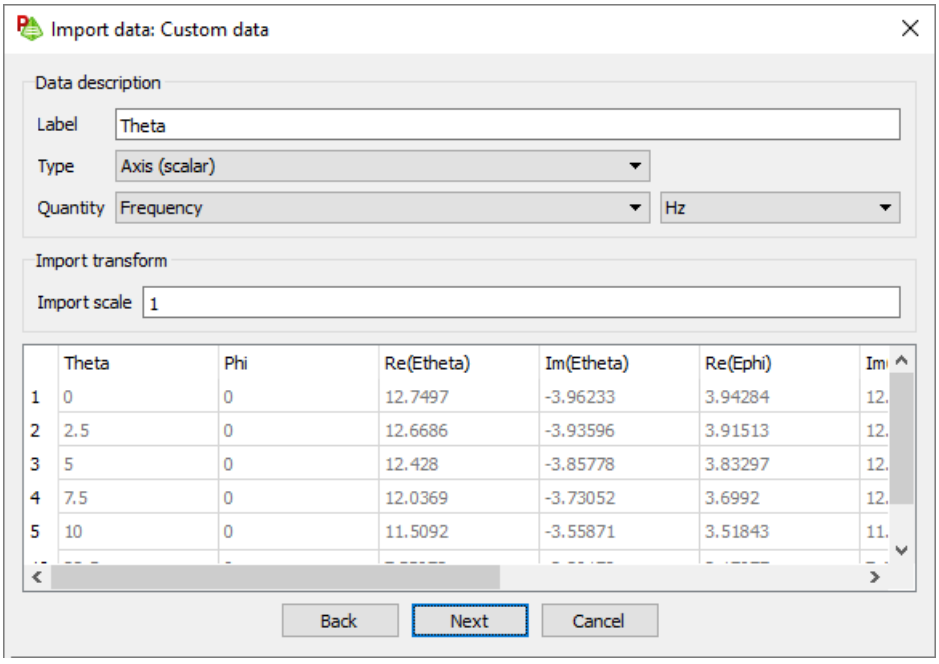


*Figure 386: The **Import data: Custom data** dialog.*

8. In the **Label** field, specify a descriptive label for the data.

9. In the **Type** drop-down list, select one of the following and then a relevant **Quantity** for the data column:
   - *Axis scalar*

     Select this option if the column is used as an independent axis on a graph.

     Quantity: frequency, position, radius, angle, time or a user-defined quantity.

   - *Scalar*

     Select this option if any scalar result type may be used.

     Quantity: far field, near field, voltage, current, power, specific absorption rate (SAR), impedance / admittance, scattering parameters, axial ratios, gain / directivity, radar cross section (RCS), voltage standing wave ratio, reflection coefficient, Poynting vector (magnitude) user-defined quantities and several other typical data types.

   - *Complex pair (Real + Imaginary)*

     Select this option of two adjacent columns contain the real and imaginary components of a complex number.

     Quantity: far field, near field, voltage, current, impedance / admittance, scattering parameters, reflection coefficient, or a user-defined quantity.

   - *Ignore*

     Select this option if a column is to be ignored during the import process.
10. In the **Import scale** field, enter a value to scale the data.
11. If the data in the column is in dB, select the **Data is in dB (not linear)** check box.
12. Repeat 9 to 11 for the remaining columns.
13. Click **Done** to import the data and to close the dialog.

A new **Imported files** entry, is created that is accessible from the project browser or the ribbon.

Refresh Imported Data

If after data was imported into POSTFEKO, the external file is modified, the imported data can be refreshed without the need for reimporting the data.

If changes occur to the external file, a refresh icon is displayed next to the file name.

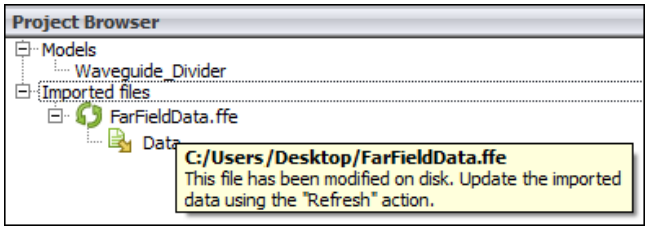


Figure 387: Example of an imported `FarFieldData.ffe` file that shows the refresh icon.

To refresh the external file, from the right-click context menu, select **Refresh**.

3.7 Data Export

POSTFEKO supports the export of native file formats. These files can be exported to use in other sessions or when further post-processing is required.

3.7.1 Supported File Formats for Export

POSTFEKO supports the export of native file formats and Touchstone file format.

On the **Home** tab, in the **File** group, click the  **Export** icon. From the drop-down list, select the file format to export.

The following formats are supported for export:

- Feko far field (`.ffe`)
- Feko near field (`.efe`, `.hfe`)
- Touchstone (`.snp`)
- Currents and charges (`.os`, `.ol`)
- Custom data (`.txt`)
- Graph data to file (`.dat`)
- Graph data to the clipboard for quick transfer to an external application

Exporting Data

To export data, select the model, its configuration and the specific results to export to a native file format.

1. On the **Home** tab, in the **File** group, click the  **Export** icon. From the drop-down list, select the file format to export.
2. In the **Source** panel, select the required configuration.
3. In the **Results** panel, select a result for the selected configuration.
4. Under **Result options**, specify the result-specific parameters.
5. Click **OK** to close the dialog.
6. Specify a file name and click **Save**.

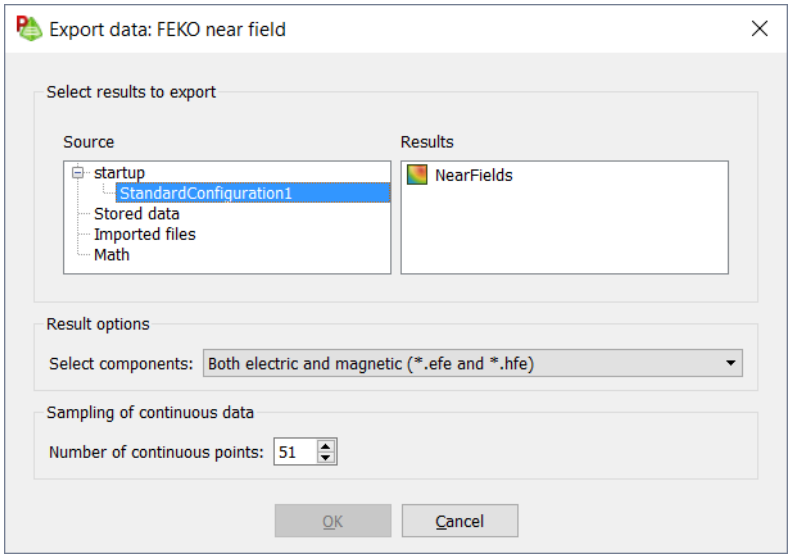


Figure 388: An example of the export dialog when exporting near field data.

3.8 Terminology

The terms, dataset, slice, trace and 3D result, are used extensively in the documentation. Review the definitions to get a better concept of these definitions.

Dataset

A dataset is any multi-dimensional data that can be used to define a full set of quantities over a full set of axes.

Slice

Slicing gives you control over which section / subset of the data is viewed.

Trace

A trace is a line plotted on a graph that represents a quantity relative to an independent axis. The styling of the trace as well as the representation of the data can be manipulated.

3D Result

A 3D result is any data that displays in three-dimensional space.

3.9 Graphs (Cartesian, Polar and Smith Charts)

Display result data on a graph to allow visual interpretation of the data in a human-readable format, as well as to communicate the results in reports and presentations.

When a trace is added to a graph and the Solver is run, POSTFEKO monitors the simulation results and updates the graphs as the results become available for discrete frequency results.

For adaptive frequency sampling results (continuous frequency), POSTFEKO displays the discrete results during the simulation and interpolate the results once the simulation is complete.

Related concepts
Trace (Terminology)
Continuous Frequency (CADFEKO)

3.9.1 Graph Types

POSTFEKO supports three types of graphs, namely Cartesian graph, polar graph and Smith chart. Each graph type represents data in a different way to make it easier to interpret for a given application.

Related concepts
Cartesian Surface Graphs

Creating a Cartesian Graph

A Cartesian graph is the classical line graph and most simple graph type. This graph type is used when you want to view closely related series of data. Any data can be viewed on a Cartesian graph.

On the **Home** tab, in the **Create new display** group, click the **Cartesian** icon.

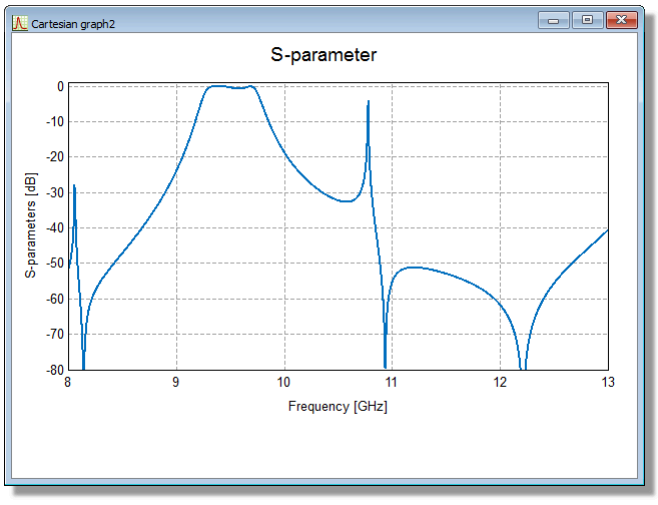


Figure 389: Example of a Cartesian graph with S-parameter results.

Creating a Polar Graph

A polar graph allows you to plot data that has at least one angular axis. You can either plot a full polar graph or only display a sector of the polar graph.

On the **Home** tab, in the **Create new display** group, click the **Polar** icon.

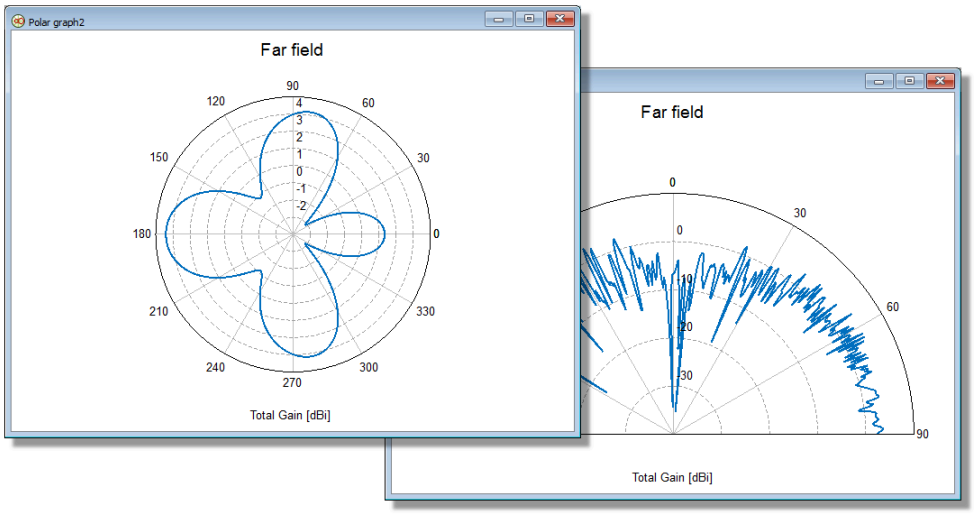


Figure 390: Example of a full polar graph and sector of a polar graph.

Creating a Smith Chart

A Smith chart allows you to view complex impedance, admittance, reflection coefficient and S-parameters.

On the **Home** tab, in the **Create new display** group, click the **Smith** icon.

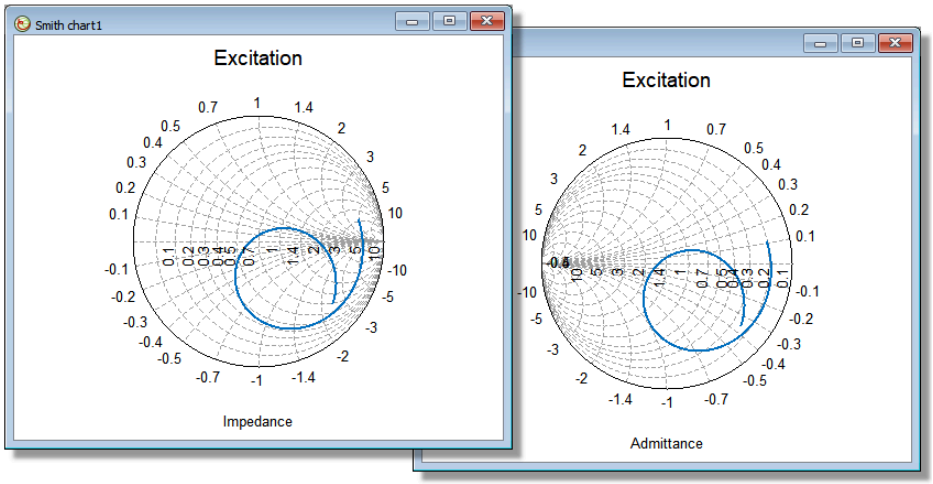


Figure 391: Examples of a Smith chart (impedance and admittance).

Changing the Grid Type to Admittance

Modify the default grid type from impedance to admittance.

Select the Smith chart that you want to modify.

On the **Display** tab, in the **Grid** group, click the **Grid type** icon.

Overlay Image for New Graphs

For the first five times that POSTFEKO is started after installation, an overlay image is displayed to guide you on how to add data. The overlay image is removed once data is added to the graph.

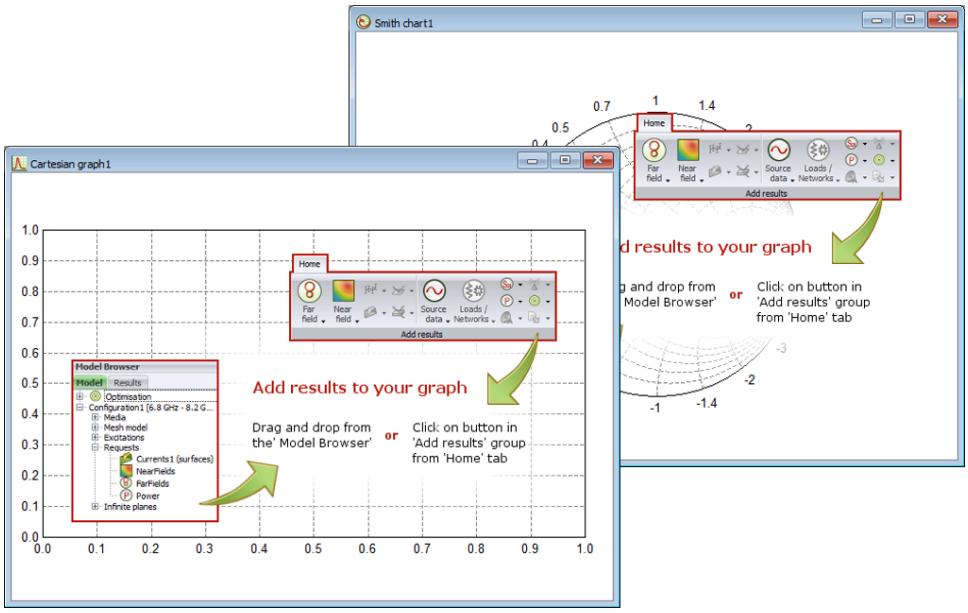


Figure 392: Overlay image when creating a new graph.

3.9.2 Graph Settings

A number of settings are available to customise a graph. From changing the font, font size, adding fill, changing the marker styling, adding shapes and text boxes, editing the graph title, footer and many more settings to obtain graphs that suits your styling.

Editing a Graph Title, Footer and Axes

Modify the graph title, graph footer, vertical axis label and horizontal axis label.

Select the graph where you want to change the title, footer, or axis labels.

A default title, footer, vertical axis and horizontal axis are assigned to a graph based on its content.

1. On the **Display** tab, in the **Display** group, click the **Chart text** icon.

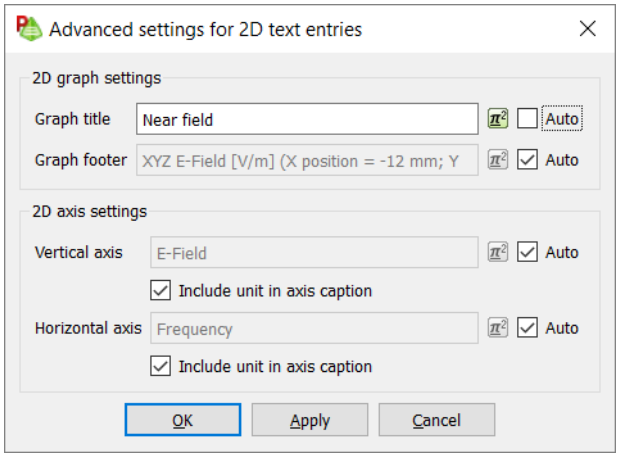


*Figure 393: The **Advanced settings for 2D text entries** dialog.*

2. Edit the graph title.
   a) Under **2D graph settings**, next to the **Graph title** field, clear the **Auto** check box.
   b) In the **Graph title** field, enter the text you want to add as the title.

   > **Tip:** Clear the **Graph title** field to remove the graph title.

3. Edit the graph footer.
   a) Under **2D graph settings**, next to the **Graph footer** field, clear the **Auto** check box.
   b) In the **Graph footer** field, enter the text you want to add as the title.

   > **Tip:** Clear the **Graph footer** to remove the graph footer.

4. Edit the vertical axis label (or the horizontal axis label).
   a) Under **2D axis settings**, next to the **Vertical axis** field, clear the **Auto** check box.
   b) In the **Vertical axis** field, enter the text you want to add as the title.
   c) [Optional] Clear the **Include unit in axis caption** check box if you do not want a unit to be assigned automatically to the axis based on the graph content.
5. Click **OK** to apply the changes and to close the dialog.

Adding Greek Symbols and Character Formatting to Text

Add Greek symbols to any text on a graph. Use rich text formatting on individual characters.

Greek symbols and individual character formatting are available for graph titles, axes titles, legend text, and text boxes.

1. Click on the **Rich text formatting** icon.
2. Modify the text.
3. Click **OK** to apply the changes and to close the dialog.

> **Tip:** For a complete set of Greek symbols, click on **More symbols**.

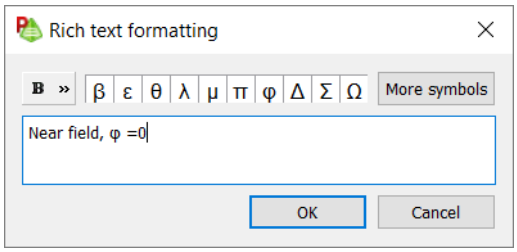


*Figure 394: The **Rich text formatting** dialog.*

Changing the Font

Modify the default font and font styling for text on a graph.

Select the graph where you want to change the font.

Change the default font.

1. Select the text you want to change.

   > **Tip:** To select multiple objects, press and hold Ctrl while you click the items.

   a) On the **Format** tab, in the **Font** group, select a font from the **Font** drop-down list.
2. Underline the text.
   a) Select the text you want to underline.
   b) On the **Format** tab, in the **Font** group, click the U **Underline** icon.

Adding a Fill to a Text Box or Shape

Add a colour fill to the interior of a text box or shape.

Select the graph where you want to change the look of a text box or shape.

1. Click the text box or shape that you want to fill.
2. On the **Format** tab, in the **Colour** group, select the **Flood fill** button.
3. Select one of the following:
   - To add or modify a fill colour, click the colour you want to use for the fill.
   - To add a colour that is not included as one of the basic colours, click **More colours**.
   - To remove the fill colour, click **No colour**.

Changing the Line Styling

Change the line style, line colour and line weight of a selected trace.

Select the graph where you want to change the line style, line colour and line weight and click the trace.

1. Change the line style of the selected trace.
   a) On the **Format** tab, in the **Line** group, click the **Line style** icon.
   b) Select one of the following:
      - To remove the line style, click **None**.
      - To modify the line style, click the line style you want to use.
2. Change the line colour of the selected trace.
   a) On the **Format** tab, in the **Line** group, click the **Line colour** icon.
   b) Select one of the following:
      - To modify a colour, click the marker colour you want to use.
      - To add a colour that is not included as one of the basic colours, click **More colours**.
3. Change the line weight for the selected trace.
   a) On the **Format** tab, in the **Line** group, click the **Line weight** icon.
   b) Select the line weight you want to use.

Changing the Marker Styling

Change the marker style, marker colour and marker size for a selected trace.

Select the graph where you want to change the marker style, marker colour and marker size and click the trace.

1. Change the marker style for the selected trace.
   a) On the **Format** tab, in the **Marker** group, click the **Marker style** icon.
   b) Select one of the following:
      - To remove markers, click .
      - To add markers, select the marker style you want to use.
2. Change the marker colour for the selected trace.
   a) On the **Format** tab, in the **Marker** group, click the **Marker colour** icon.
   b) Select the maker colour you want to use.
3. Change the marker size for the selected trace.
   a) On the **Format** tab, in the **Marker** group, click the **Marker size** icon.
   b) Select one of the following:
      - To select a specified marker size, click the marker size you want to use.
      - To specify a marker size that is not included as one of the default sizes, click **Custom**.
   c) Select the maker size you want to use.

Changing the Marker Placement

Change the marker placement for a trace to view the calculated points in a continuous frequency simulation or for aesthetic reasons.

Select the graph where you want to change the marker placement and click the trace.

1. On the **Format** tab, in the **Marker** group, click the **Marker placement** icon.
2. Select one of the following:
   - To place markers at the calculated points on the trace, select **Calculated points**.
   - To place markers sparsely-spaced on the trace, select **Sparsely spaced**.
   - To place markers densely-spaced on the trace, select **Densely spaced**.

> **Note:** The **Sparsely spaced** and **Densely spaced** trace options are always visible in a view, irrespective of the zoom level.

Adding a Shadow to Text or a Shape

Add a drop shadow to text or shape.

Select the graph where you want to add a drop shadow to text or a shape.

1. Add a drop shadow to text or a shape.
   a) Click the text or shape.
   b) On the **Format** tab, in the **Effects** group, click the **Drop shadow** icon.
2. Change the depth of the drop shadow.
   a) Click the text or shape.
   b) On the **Format** tab, in the **Effects** group, click the **Shadow depth** icon.
   c) Select the depth you want to use for the drop shadow.

Specifying the Major Axes Range

Specify the range for the major axes.

Select the graph where you want to change the axis range.

1. On the **Cartesian** context tab, on the **Display** tab, on the **Axes** group, click the **Axis settings** icon.

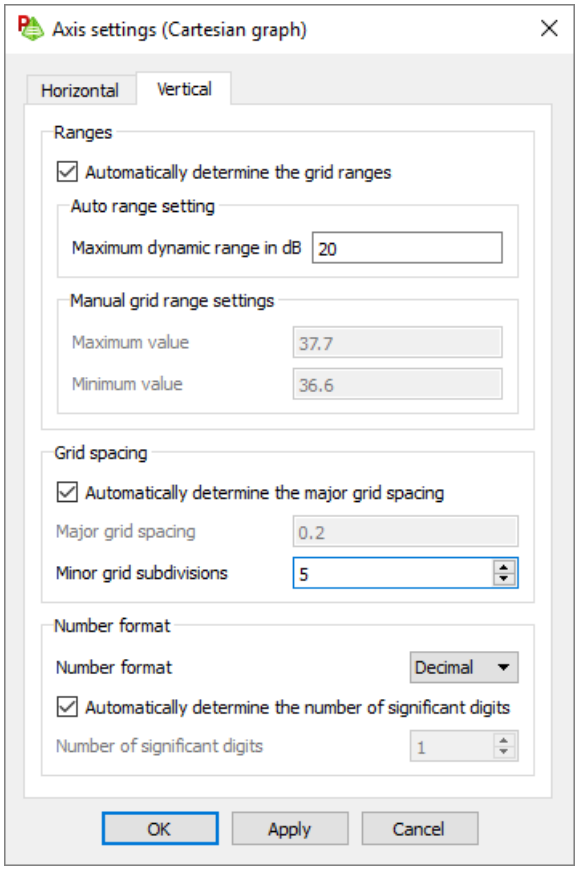


*Figure 395: The **Axis settings** dialog.*

2. Select the axis that you want to modify.
   - To modify the grid range for the horizontal axis, click **Horizontal**.
   - To modify the grid range for the vertical axis, click **Vertical**.
3. Under **Ranges**, select one of the following:
   - To automatically determine the grid range, select the **Automatically determine the grid range** check box.
     - To specify the dynamic range for the vertical axis, under **Auto range setting**, in the **Maximum dynamic range in dB** field, enter a value for the dynamic range in dB.

       Note: For the **Maximum value** the maximum value of the traces is used. For the **Minimum value** the minimum value of the traces is used, or the maximum value of the traces minus the specified dynamic range, whichever is larger.
   - To specify the grid range, clear the **Automatically determine the grid range** check box.
     - In the **Maximum value** field, enter a value for the upper limit of the graph.
     - In the **Minimum value** field, enter a value for the lower limit of the graph.
4. Click **OK** to apply the settings and to close the dialog.

Specifying the Grid Spacing

Specify the grid interval for the major (and minor) grid.

Select the graph where you want to change the grid spacing for the major grid (or minor grid).

1. On the **Cartesian** context tab, on the **Display** tab, on the **Axes** group, click the **Axis settings** icon.
2. Select the axis that you want to modify.
   - To modify the grid range for the horizontal axis, click **Horizontal**.
   - To modify the grid range for the vertical axis, click **Vertical**.

Modify the major grid spacing.

3. Under **Grid spacing**, select one of the following:
   - To automatically determine the major grid spacing for the graph, select the **Automatically determine the major grid spacing** check box.
   - To specify the major grid spacing, clear the **Automatically determine the major grid spacing** check box.
     - In the **Major grid spacing** field, enter a value for the major grid spacing.

Modify the minor grid spacing.

4. Under **Grid spacing**, select one of the following:
   a) In the **Minor grid subdivisions** field, enter a value for the minor grid spacing.
5. Click **OK** to apply the settings and to close the dialog.

Enable the minor grid to view the grid spacing.

6. On the **Cartesian** context tab, on the **Display** tab, on the **Minor grid** group, click the **Minor grid** icon.

Changing the Axis Scale to Logarithmic

Modify a graph to make use of a logarithmic scale when the data is spread over large range. A logarithmic (log) scale allows you to view the data on a non-linear scale.

A log scale can be applied to a Cartesian graph for both the horizontal and vertical axes. For a polar graph, a log scale can only be applied to the radial axis.

As an example, the horizontal axis of a Cartesian graph is changed to a log scale. The steps are similar for changing the vertical axis of a Cartesian graph or the radial axis of a polar graph.

1. Select the Cartesian graph where you want to enable the log scaling for the horizontal axis.
2. On the **Display** tab, in the **Axes** group, click the **Log (horizontal)** icon.

Reversing the Axis Order

Change the order in which values are plotted along the axis of a Cartesian graph.

The axis order can be reversed for both the horizontal axis and the vertical axis.

As an example, the vertical axis order of a Cartesian graph is reversed. The steps are similar for reversing the horizontal axis order.

1. Select the Cartesian graph where you want to reverse the vertical axis order.
2. On the **Cartesian** contextual tabs set, on the **Display** tab, in the **Axes** group, click the **Reversed order (vertical)** icon.

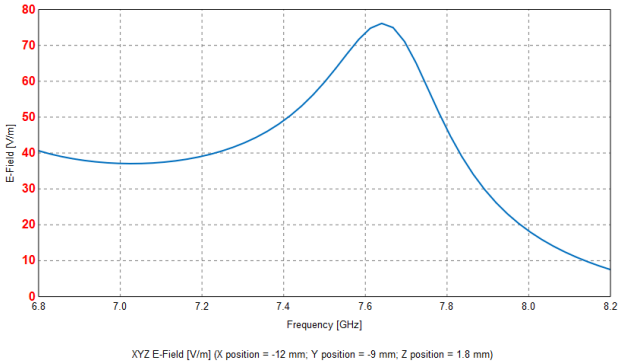


Figure 396: An example of a Cartesian graph.

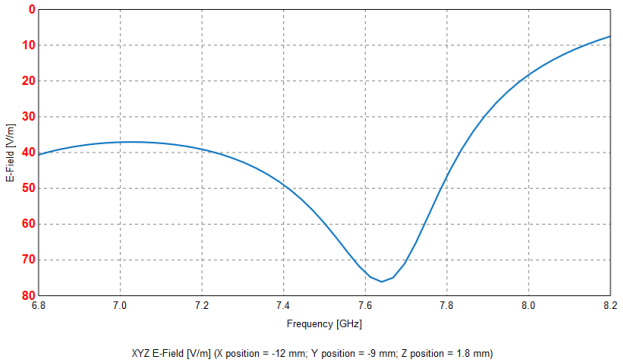


Figure 397: An example of a Cartesian graph where the order of the vertical axis was reversed.

Changing the Unit of an Axis

Modify the axes units to allow data to be plotted in a familiar unit (for example, to change dBmV/m to dBuV/m) or to shorten the axis text and make it more readable.

As an example, the vertical unit is changed. The step is similar for changing the unit of the horizontal axis.

1. Select the trace.
2. On the **Trace** tab, in the **Units** group, from the **Vertical unit** drop-down list, select an unit.

Specifying the Number Format for an Axis

Specify the number format for the axes and the number of significant digits that are display on the axes.

Select the graph where you want to change the major axis (or minor axis).

1. On the **Cartesian** context tab, on the **Display** tab, on the **Axes** group, click the **Axis settings** icon.
2. Select the axis that you want to modify.
   - To modify the grid range for the horizontal axis, click **Horizontal**.
   - To modify the grid range for the vertical axis, click **Vertical**.

Change the number format for the axis.

3. Under **Number format**, select one of the following:
   a) From the **Number format** drop-down list, select one of the following:
      - Decimal
      - Scientific

Modify the number of significant digits for the axis.

4. Under **Number format**, select one of the following:
   - To specify the number of significant digits that you want to see on the graph axis, clear the **Automatically determine the number of significant digits** check box.
     - From the **Number of significant digits** drop-down list, select the number of digits you want to view on the graph.
   - To determine the number of significant digits for the graph axis automatically, select the **Automatically determine the number of significant digits** check box.
5. Click **OK** to apply the settings and to close the dialog.

3.9.3 Graph Legend

A graph legend is a summary of the trace or traces displayed on the graph. The legend also indicates which colour represents each trace on the legend

Adding a Legend to a Graph

Add a legend to a graph, modify the legend position and specify the number of columns for the legend entries.

Select the graph where you want to modify the legend.

> **Note:** When you add a trace to a graph, a legend entry is added automatically.

1. Modify the legend position.
   a) On the **Display** tab, in the **Legend** group, click the **Position** icon.

b) From the drop-down list select a position where you want to place the legend.

The graph is automatically resized based on the legend position.

2. Modify the number of columns displayed for the legend.
   a) On the **Display** tab, in the **Display** group, click the **Number of columns** icon.
   b) From the drop-down list, select one of the following:
      - To create a legend with a specified number of columns in the legend, select the number you want to use.
      - To create a legend where the number of columns is determined automatically, select **Auto**.

Editing Legend Text

Modify or remove the legend entry text.

Select the graph where you want to modify the legend text and click the trace.

1. On the **Cartesian** context tab, on the **Display** tab, in the **Legend** group, click the **Trace text** icon.

*Figure 398: The **Legend entry settings** dialog.*

2. Specify the trace text for the graph legend.
   a) Clear the **Auto** check box.
   b) In the **Legend text** field, enter the trace text for the graph legend.

> **Tip:** You can also use one of the following workflows:
> - In the result palette or 3D view, select the trace. From the right-click context menu, select **Trace text**.
> - Press Shift+F2.

3. To remove a trace from the legend, select one of the following:
   - Clear the **Legend entry visible** check box.
   - Clear the **Auto** check box and delete the content in the **Legend text** field.

Use the trace label (displayed in the result palette) as the legend text.

4. Select the **Use trace label text** check box.
5. Click **OK** to apply the changes and to close the dialog.

Changing the Order of Legend Entries

Raise or lower a trace in the result palette to change the order of the legend entries (traces).

Related tasks
Raising and Lowering a Trace

3.9.4 Annotations and Cursors

Use annotations and cursors to read and interpret plotted results.

Annotations

Add an annotation to a trace to highlight values of interest. The annotation updates along with the data and always display the value according to its definition.

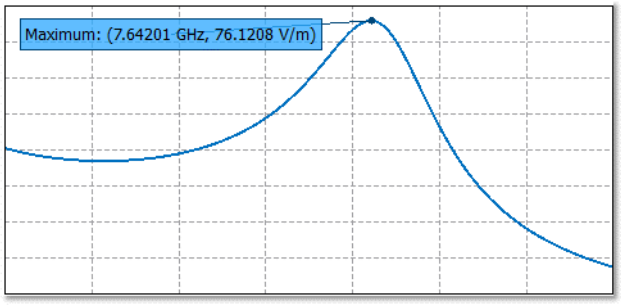


Figure 399: An example of an annotation on a Cartesian graph.

Cursors

Cursors are dynamic and allow you to interact and move the cursors. Drag the cursors until they are placed at the desired positions. Cursors allow data to be read off several traces simultaneously but suffer from the limitation that it cannot update along with the results.

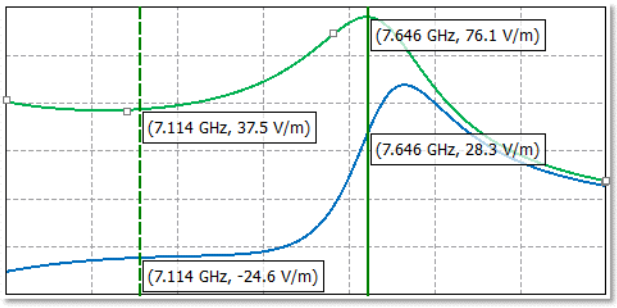


Figure 400: An example of cursors on a Cartesian graph.

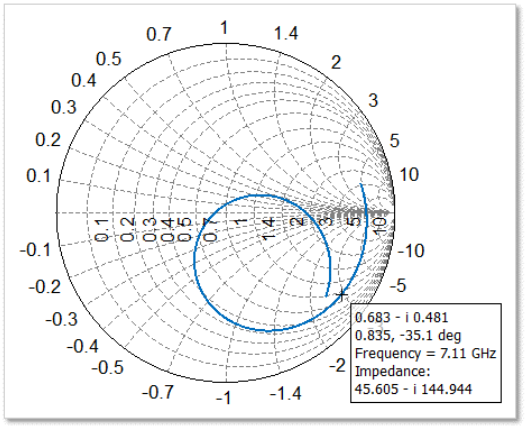


Figure 401: An example of a cursor on a Smith chart.

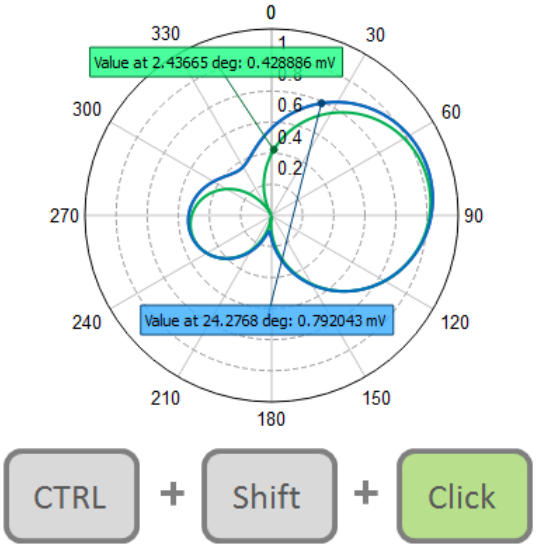
Note: A Smith chart has a single cursor appearing as a small table.

Related tasks
Adding a Custom Point Annotation
Adding Cursors and a Cursor Table

Adding a Quick Single Point Annotation

Read a point from a graph by adding a quick single point annotation.

1. Select the graph where you want to add the annotation.
2. Position the mouse cursor on the graph trace.
3. Press Ctrl+Shift+left click.



Note: A quick single point annotation is not available for a Smith chart.

Adding a Custom Point Annotation

Add a custom point annotation to read the value and highlight a point of interest on the graph.

Select the trace where you want to read the point.

1. Add a single point annotation to indicate the global maximum of the selected trace.
   a) On the **Cartesian** context tab, on the **Measure** tab, on the **Custom annotations** group, click the **Point** icon. From the drop-down list, click **Global maximum**.

   An annotation is added to the trace to highlight the maximum value. The annotation updates if the data changes.
2. Add a custom single point annotation to indicate the first local minimum to the left (relative to the global maximum).
   a) On the **Cartesian** context tab, on the **Measure** tab, on the **Custom annotations** group, click the **Point** icon. From the drop-down list, click **Other**.

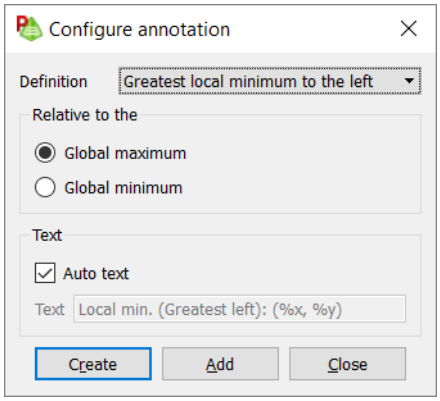


*Figure 402: The **Configure annotation** dialog.*

   a) In the **Definition field**, from the drop-down list, select **First local maximum to the left**.
   b) Under **Relative to**, click **Global maximum**.
   c) [Optional] Under **Text**, clear the **Auto text** check box and add the text you want displayed in the annotation.
3. Click **Create** to create the annotation and to close the dialog.

Custom Point Annotations

A number of custom point annotations definitions are available for a Cartesian graph that allows you the flexibility to annotate any point of interest on the graph.

On the **Cartesian** context tab, on the **Measure** tab, on the **Custom annotations** group, click the **Point** icon. From the drop-down list, select the type of annotation you want to use.

Table 20: The custom point annotation definitions available in POSTFEKO.

Icon	Icon text	Description
	**Global maximum**	Place an annotation at the global maximum.
	**Global minimum**	Place an annotation at the global minimum.
	**Specify independent axis value**	Place an annotation at a specified independent axis value.
	**Second maximum**	Place an annotation at the second maximum.
	**Second minimum**	Place an annotation at the second minimum.
	**Other**	Opens a dialog where you can specify any of the following custom point annotations: • Global maximum • Global minimum • First local maximum • First local maximum to the left • First local maximum to the right • Greatest local maximum • Greatest local maximum to the left • Greatest local maximum to the right • First local minimum • First local minimum to the left • First local minimum to the right • Greatest local minimum • Greatest local minimum to the left • Greatest local minimum to the right • Value at given horizontal position • Define independent value

Adding a Custom Annotation Between Two Points (Delta)

Add an annotation to highlight the difference between two points on a graph.

Add an annotation to the global maximum and the first local maximum to the left on the graph.

1. On the **Cartesian** context tab, on the **Measure** tab, on the **Custom annotations** group, click the **Delta** icon.

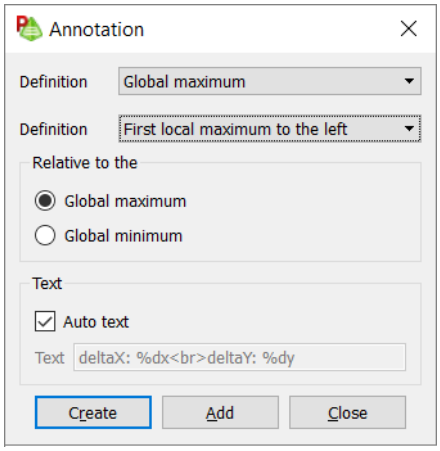


*Figure 403: The **Annotation** dialog.*

2. In the **Definition field**, from the drop-down list, select **Fist local maximum to the left**.
3. Under **Relative to the**, click **Global maximum**.
4. [Optional] Under **Text**, clear the **Auto text** check box and add the text you want displayed in the annotation.
5. Click **Create** to create the annotation and to close the dialog.

Adding a Custom Annotation to a Point and Its Derived Width

Specify a single point of interest. Add an annotation between adjacent points derived from the single point.

As an example, add an annotation to indicate the -3 dB transmission bandwidth.

Select the trace where you want to read the derived width.

1. On the **Cartesian** context tab, on the **Measure** tab, on the **Custom annotations** group, click the **Derived width** icon.

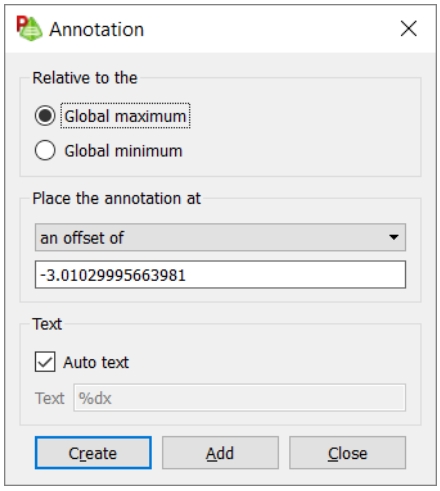


*Figure 404: The **Annotation** dialog.*

2. Under **Relative to the**, click **Global maximum**.
3. Under **Place the annotation at**, from the drop-down list select **an offset of**.
4. [Optional] Under **Text**, clear the **Auto text** check box and add the text you want displayed in the annotation.
5. Click **Create** to create the annotation and to close the dialog.

Adding Result Specific Annotations

For impedance results and far field results, custom annotations are available that allows you to quickly add annotations relevant to the data types, for example, reflection bandwidth, transmission bandwidth, beamwidth and sidelobe level.

Tip: Use custom annotations for custom data.

Annotations for Bandwidths

A specialised form of annotations is available for to annotate impedance results and to highlight bandwidth.

Due to the varying definitions of "bandwidth" between industries and applications, definitions for both transmission and reflection bandwidths are provided.

Reflection bandwidths are typically used in antenna modelling. Transmission bandwidths are used for filters and other multi-port problems.

Note: -3 dB bandwidth refers to the frequency point where the power is at 3 dB below the maximum value or half the maximum power.

Table 21: The reflection bandwidth annotation definitions available in POSTFEKO.

Icon	Icon text	Description
	**-3 dB Reflection bandwidth**	Place annotation to indicate the -3 dB half power reflection bandwidth.
	**-10 dB Reflection bandwidth**	Place annotation to indicate the -10 dB half power reflection bandwidth.
	**-15 dB Reflection bandwidth**	Place annotation to indicate the -15 dB half power reflection bandwidth.

Table 22: Transmission bandwidth annotations.

Icon	Icon text	Description
	**-3dB Transmission bandwidth**	Place annotation to indicate the -3 dB half power transmission bandwidth.
	**-10dB Transmission bandwidth**	Place annotation to indicate the -10 dB half power transmission bandwidth.
	**-15dB Transmission bandwidth**	Place annotation to indicate the -15 dB half power transmission bandwidth.

Highlighting the -3dB Bandwidth

Add an annotation to highlight the -3 dB reflection bandwidth for a source result.

Select the trace where you want to read the -3 dB bandwidth.

Add a **Reflection bandwidth** annotation.

a) On the **Measure** tab, in the **Source annotations** group, click the  **Reflection bandwidth** icon.

b) From the drop-down list, select **-3 dB**.

Annotations for Beamwidth and Sidelobe Level

A specialised form of annotations is available to annotate far field results and to highlight beamwidth and sidelobe level.

A number of annotations for beamwidth and sidelobe level are provided. The sidelobe level is defined as the ratio between the maximum beam strength divided by the second largest beam strength. Annotations for locating the first null and the bandwidth from null to null are also provided.

Table 23: The reflection bandwidth annotation definitions available in POSTFEKO.

Icon	Icon text	Description
	**Half power (-3dB)**	The -3 dB half power beamwidth.
	**First Null**	The first null beamwidth.
	**Null to Null**	The null to null beamwidth.
	**Sidelobe level**	The sidelobe level.

Highlighting the Half Power (-3dB) Beamwidth and Sidelobe Level

Add an annotation to highlight the half power (-3dB) beamwidth for the far field result.

Select the graph and trace to which you want to add the annotation.

1. Add an annotation to highlight the half power (-3 dB) for the plotted result.
   a) On the **Measure** tab, in the **Far field annotations** group, click the  **Beamwidth** icon.
   b) From the drop-down list select **Half power (-3dB)**.
2. Add an annotation to highlight the **Sidelobe level**.
   a) On the **Measure** tab, in the **Far field annotations** group, click the  **Sidelobe level** icon.

Adding Cursors and a Cursor Table

Use cursors and its cursor table to read and interpret information from a graph. Place cursors at predefined positions.

Select the graph where you want to read the information on the graph.

1. Enable cursors on the graph.
   a) On the **Measure** tab, in the **Measurement** group, click the  **Cursors** icon.
2. Add a cursor table to the graph.
   a) On the **Measure** tab, in the **Measurement** group, click the  **Cursor table** icon.

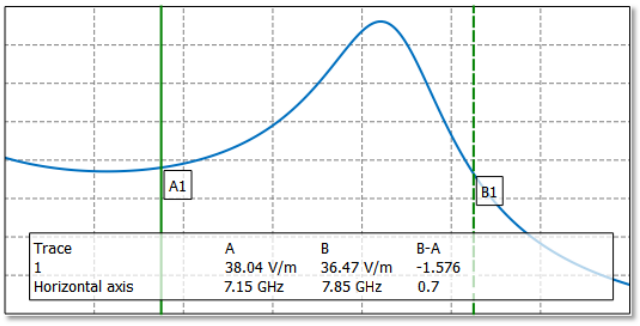


Figure 405: A Cartesian graph with cursor table.

Note: The table contains the data for the displayed points as well as the difference (indicated by **B-A**).

Tip: If you move the cursor outside of the visible region of a graph, a handle appears to retrieve the cursor.

3. Set the cursor position to a predefined position. For example, place the cursor at the global maximum of the selected trace.
   a) Select the trace where you want to find the global maximum.
   b) On the **Measure** tab, in the **Measurement** group, click the **Global maximum** icon.

Predefined Cursor Positions

View the available predefined cursor positions.

Table 24: Predefined cursor positions for graphs.

Icon	Icon text	Description
	Global max	Place the cursor at the global maximum
	Global min	Place the cursor at the global minimum.
	Local max to the left	Place the cursor at the next local maximum.
	Local max to the right	Place the cursor at the next local maximum.
	Local min to the left	Place the cursor at the next local minimum to the left.
	Local min to the right	Place the cursor at the next local minimum to the right.

Adding Text Boxes and Shapes

Add text boxes and shapes to a graph to add a comment and highlight results.

1. Add a text box to a graph.
   a) On the **Format** tab, in the **Drawing** group, click the **Text box** icon.
   b) In the **Text** field, enter the text you want to add to the graph.
   c) Click **Create** to create the text box and to close the dialog.
2. Change the direction of the text box.
   a) On the **Format** tab, in the **Drawing** group, click the **Text direction** icon.
   b) From the drop-down list select one of the following:
      - To place the text horizontally, select **Horizontal**.
      - To rotate the text clockwise by 90°, select **Top to bottom**.
      - To rotate the text counter-clockwise by 90°, select **Bottom to top**.
3. Add a shape to the graph.
   a) On the **Format** tab, in the **Drawing** group, click the **Shapes** icon.
   b) From the drop-down list select one of the following:
      - To create a line, select **Line**.
      - To create an arrow, select **Arrow**.
      - To create a double-arrow, select **Double arrow**.
      - To create a rectangle, select **Rectangle**.
      - To create a circle, select **Circle**.

> **Note:** Double-click a rectangle or circle to add text.

3.9.5 Overlaying an Image on a 2D Graph

Add an image to a Cartesian graph, polar graph or Smith chart to better interpret and understand the results.

Adding a Static Overlay Image

Overlay an image on a graph. The image can either be a 3D view or imported from a file.

Select the graph to which you want to add the image.

1. On the **Display** tab, in the **Image** group, click the **Chart image** icon.
2. From the drop-down list, select one of the following: an image from the **3D result view** (if available) or select **Import file** under **Existing image**.
   - To add an image of the 3D view to the graph, select the relevant 3D view.

- To import an image from a file, select **Import file**.

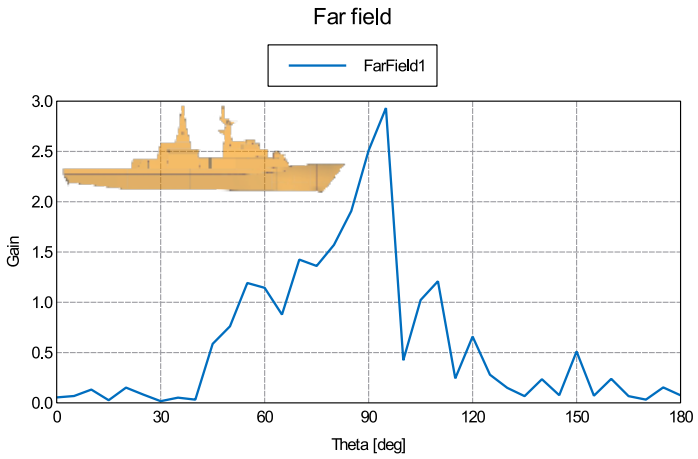


Figure 406: A Cartesian graph with overlay image of the 3D view. The image was moved to the top-left.

Adding Image as Reference to Data Cut Orientation

Add an overlay image to a polar graph as a reference to the data cut orientation.

Data must already be added to the polar graph for this option to be available.

1. Select the polar graph where you want to add the image.
2. On the **Display** tab, in the **Image** group, click the **Chart image** icon.
3. From the drop-down list select **Model reference to data cut orientation**.
4. Under **Model reference to data cut orientation**, select a far field request.

> **Note:** This type of image is automatically updated depending on the selected data cut for the trace.

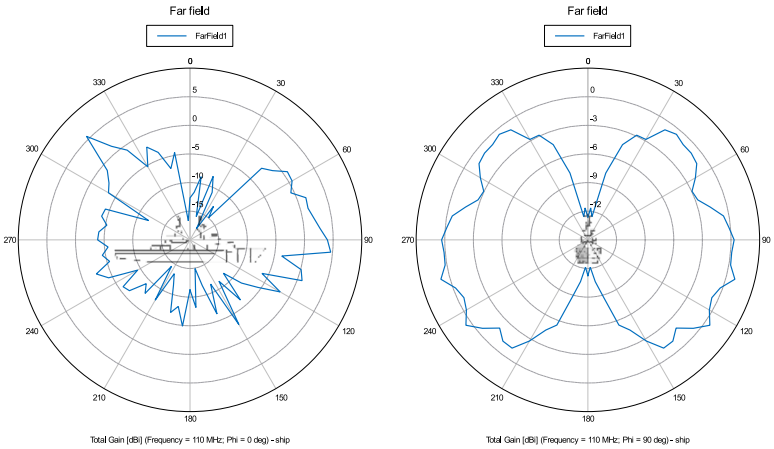


Figure 407: The image is added to the polar graph as q reference to the data cut orientation. On the left, phi was set to 0 degrees. On the right, phi was set to 270 degrees and the overlay image was automatically updated to reflect the changes.

Customising an Overlay Image

Resize, position and orientate an overlay image. Apply opacity to the image.

1. Position and resize the image
   a) Select the overlay image you want to customise.
   b) Move the image by dragging the image with your mouse.
   c) Resize the image by dragging the resize handles.
   d) To center the image, from the right-click context menu, select **Center image**.
2. Set the opacity of the overlay image.
   a) On the **Display** tab, in the **Image** group, click the **Opacity** icon.
   b) From the drop-down list, select a percentage or use **Custom** to set a custom percentage.
3. Change the orientation of static overlay image.

   Note: This option is only available for a static overlay image.

   a) On the **Display** tab, in the **Image** group, click the **Rotation** icon.
   b) From the drop-down list, select one of the predefined angles or select **Custom** to enter a rotation angle.

3.9.6 Duplicating a Graph

Create a duplicate view of a graph, complete with all settings.

> **Note:** Cursors on the graph are not duplicated.

Create a duplicate view.

a) Select the graph you want to duplicate.

b) On the **Display** tab, in the **Duplicate** group, click the **Duplicate view** icon.

3.9.7 Copying and Converting to a Different Graph Type

Create a copy of the graph and change the graph type (if the data is compatible with both). For example, create a polar graph copy from a Cartesian graph.

As an example, a polar graph copy is created from a Cartesian graph, but the steps are similar to derive any graph type.

1. Select the graph from which you want to create a derived copy.
2. On the **Display** tab, in the **Duplicate** group, click the **Polar copy** icon.

> **Note:** If the traces on the source graph are incompatible with the derived graph, an error is given stating the incompatible traces.

3.9.8 Trace Manipulation

A trace is a line plotted on a graph that represents a quantity relative to an independent axis. The styling of the trace as well as the representation of the data can be manipulated.

Duplicating a Trace

Make a copy of a trace.

1. Select the trace you want to duplicate.
2. On the **Trace** tab, in the **Manage** group, click the **Duplicate trace** icon.

> **Tip:** You can also use one of the following workflows:
>
> - In the result palette or 3D view, select the trace. From the right-click context menu, select **Duplicate trace**.
> - Press Ctrl+K.

Storing a Local Copy of a Data Set

Stores a local copy of the underlying data that is represented by the trace. By storing a local copy, you can modify the existing model and compare the old results to the new results.

1. Select the trace that you want to store a copy of the underlying data.

   > **Note:** Most results from a graph can be stored, except for cable probes, error estimates, imported data, rays, and currents and charges.

   A new entry under **Stored data** is created that is accessible from the project browser or the ribbon.

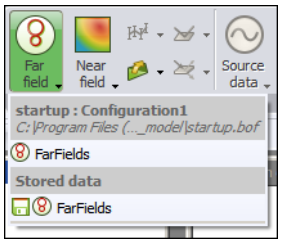


Figure 408: Accessing stored data from the ribbon.

2. On the **Trace** tab, in the **Manage** group, click the **Store a copy** icon.

Math Traces

A math trace is created to perform calculations on existing data or to create mathematically defined reference curves. These traces inherently contain no data and require other traces or mathematical equations to present information.

Related reference
Math Functions in POSTFEKO

Creating a New Math Trace

Use a math trace to define a mathematical reference curve. A math trace requires other traces or mathematical equations to present information.

1. On the **Trace** tab, in the **Manage** group, click the **New math** icon.
2. In the result palette, under **Maths**, type an equation in the **Maths** field.

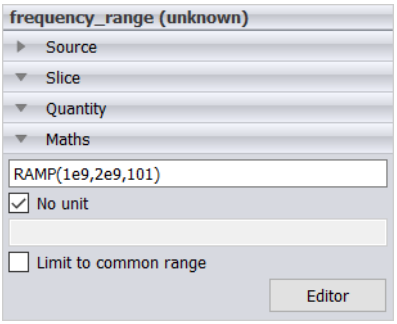


Figure 409: An example of a math trace created by using the RAMP function in the range 1e9 to 2e9 using 101 points.

> **Tip:** To use the built-in functions, and constants from POSTFEKO, click the **Editor** button to open the **Expression editor** dialog.

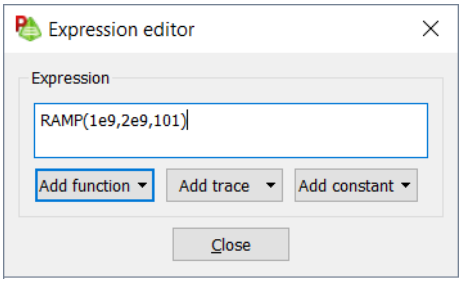


*Figure 410: The **Expression editor** dialog.*

Related reference
Math Functions in POSTFEKO

Performing Calculations on Data Sets

Create a math trace to perform calculations on existing data sets.

1. Select the trace you want to use in the calculation.
2. In the result palette, under **Maths**, select the **Enable maths** check box.

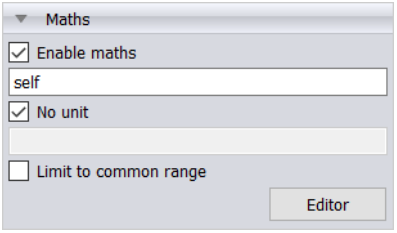


*Figure 411: Select the **Enable maths** check box to perform calculations on a trace.*

The text "self" appears in the text box.

3. Enter an equation using the term "self" to refer to the trace data.

> **Tip:** To use the built-in functions, and constants from POSTFEKO, click the **Editor** button to open the **Expression editor** dialog.

Related reference
Math Functions in POSTFEKO

Raising and Lowering a Trace

Re-order the trace sequence in the result palette. Raising and lower a trace in the result palette also changes the order of the legend entries.

Raise a trace.

1. Select the trace that you want to move.
2. On the **Trace** tab, in the **Rendering** group, click the **Raise trace** icon.

> **Tip:** You can also use one of the following workflows:
> - In the result palette, select the trace. From the right-click context menu, select **Raise trace**.
> - Press Ctrl++.

Lower a trace.

3. Select the trace that you want to move.
4. On the **Trace** tab, in the **Rendering** group, click the **Lower trace** icon.

> **Tip:** You can also use one of the following workflows:
> - In the result palette, select the trace. From the right-click context menu, select **Lower trace**.
> - Press Ctrl+-.

Related tasks
Changing the Order of Legend Entries

Transforming the Horizontal Axis

Stretch or shrink the independent axis or add an offset.

1. Select the graph for which you want to transform the axis.
2. On the **Trace** tab, in the **Units** group, click the **Transform axis (horizontal)** icon.
3. In the **Transform horizontal axis** dialog enter values for **Scale** and **Offset**.

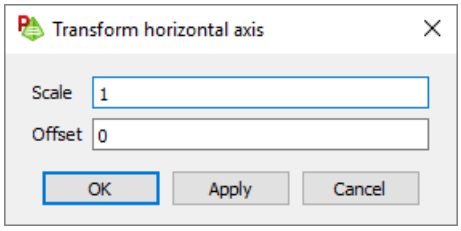


*Figure 412: The **Transform horizontal axis** dialog.*

4. Click **OK** to apply the changes and to close the dialog.

Normalising a Trace or Graph

Normalise a graph or trace to interpret results better.

On the **Display** tab, in the **Axes** group, click the Normalise To icon. Select one of the following:

- To normalise all traces to the maximum value found between all the traces, select the **Normalise to maximum of all traces** icon. Only one trace has a maximum value of 1.
- To normalise each trace to its maximum, select the **Normalise to maximum of individual traces**. All traces have a maximum value of 1.

Changing the Sampling Settings

Specify the number of samples for continuous results displayed on graphs.

The default rendering for continuous data on a graph is determined automatically. The sampling can be adjusted to display either the actual frequency samples or adjusted to display a set number of frequency points.

1. Select the graph for which you want to modify the sampling.
2. On the **Trace** tab, in the **Units** group, click the **Sampling settings** icon.

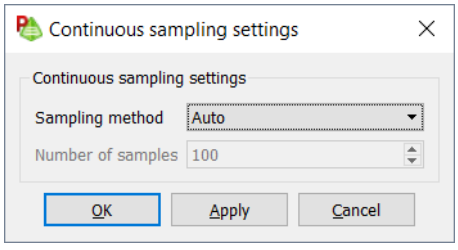


*Figure 413: The **Continuous sampling settings** dialog.*

3. From the **Sampling method** drop-down list, select one of the following:
   - To display the default rendering where the sampling is determined based on the data, select **Auto**.
   - To display only the actual samples, select **Discrete**.

- To resample the data and display a fixed number of discrete points, select **Specify number of samples**.

> **Tip:** To export continuous data, use **Specify number of samples** or **Discrete samples** to limit the number of samples (file size).

4. Click **OK** to set the sampling settings and to close the dialog.

Exporting Plotted Data to Clipboard or a Text File

Retrieve the plotted trace data and save to file.

1. On the graph, select the trace from which you want to export the plotted data.
2. Export the plotted data using one of the following workflows:
   - To export the data to a `.dat` file, from the right-click context menu select **to file (*.dat)** and specify the file name.
   - To export the data to clipboard, press Ctrl+X.

3.10 Cartesian Surface Graphs

A Cartesian surface graph is a flat colour plot with results plotted against two independent axes.

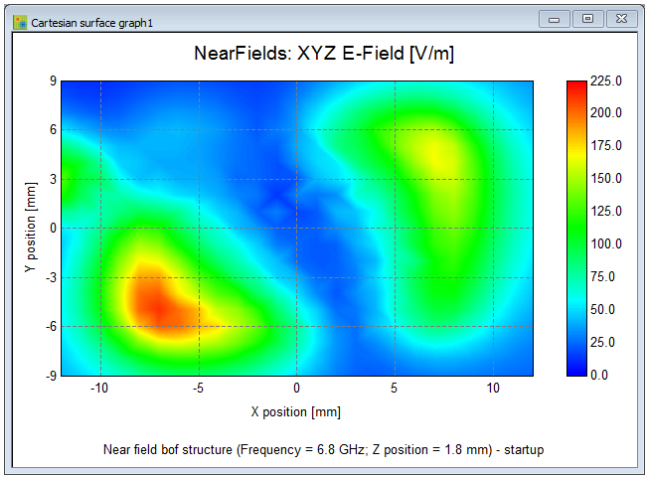


Figure 414: Example of a near field displayed on a Cartesian surface graph.

The surface graph allows you to plot quantities like radar cross section (RCS), gain or near fields as a function of two as a function of two independent parameters, such as angles theta and phi or frequency and position.

Note: Only a single plot per Cartesian surface graph is supported.

Table 25: Result types that can be viewed on a Cartesian surface graph.

Result type	Cartesian surface graph
Far fields (including RCS)	✓
Near fields	✓
Error estimates	
Currents	
Rays	
Sources	
Loads	
S-parameters	✓
Power	

Result type	Cartesian surface graph
Probes	
Transmission / reflection coefficients	
Characteristic modes	✓
Imported data	
Script data / custom datasets	✓
Optimisation	
Receiving antenna	
SAR	

3.10.1 Creating a Cartesian Surface Graph

Create a new Cartesian surface graph for displaying data.

On the **Home** tab, in the **Create new display** group, click the **Surface** icon.

3.10.2 Editing a Graph Title, Footer and Axes

Modify the graph title, graph footer, vertical axis label and horizontal axis label.

Select the graph where you want to change the title, footer, or axis labels.

A default title, footer, vertical axis and horizontal axis are assigned to a graph based on its content.

1. On the **Display** tab, in the **Display** group, click the **Chart text** icon.

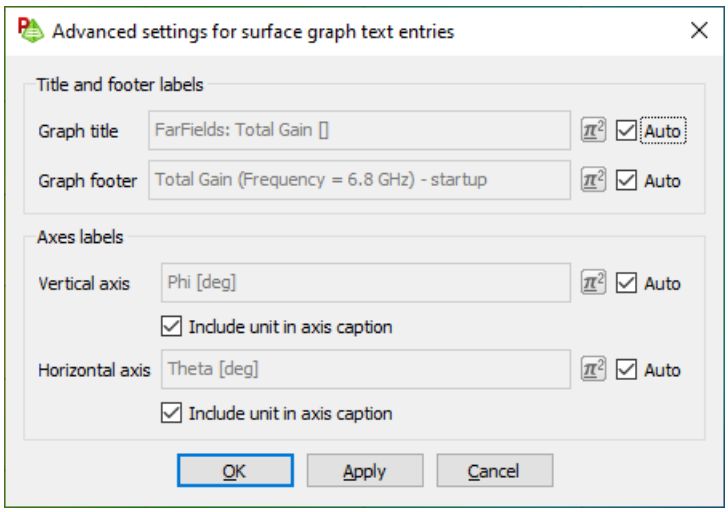


*Figure 415: The **Advanced settings for surface graph text entries** dialog.*

2. Edit the graph title.
   a) Under **Title and footer labels**, next to the **Graph title** field, clear the **Auto** check box.
   b) In the **Graph title** field, enter the text you want to add as the title.

   > **Tip:** Clear the **Graph title** field to remove the graph title.

3. Edit the graph footer.
   a) Under **Title and footer labels**, next to the **Graph footer** field, clear the **Auto** check box.
   b) In the **Graph footer** field, enter the text you want to add as the title.

   > **Tip:** Clear the **Graph footer** to remove the graph footer.

4. Edit the vertical axis label (or the horizontal axis label).
   a) Under **Axes labels**, next to the **Vertical axis** field, clear the **Auto** check box.
   b) In the **Vertical axis** field, enter the text you want to add as the title.
   c) [Optional] Clear the **Include unit in axis caption** check box if you do not want a unit to be assigned automatically to the axis based on the graph content.
5. Click **OK** to apply the changes and to close the dialog.

3.10.3 Enabling Grid Lines and Grid Labels

Enable the major and minor grid lines for surface graphs as well as the grid labels.

Enable major grid lines.

1. Select the surface graph for which you want to enable grid lines.
2. On the **Surface** contextual tabs set, on the **Display** tab, in the **Grid** group, click the **Grid** icon.

Enable minor grid lines.

3. On the **Surface** contextual tabs set, on the **Display** tab, in the **Grid** group, click the **Minor grid** icon.

   Note: Enable the major grid to view the minor grid.

Enable the minor grid labels.

4. On the **Surface** contextual tabs set, on the **Display** tab, in the **Grid** group, click the **Labels (horizontal)** icon.

   Note: Enable the minor grid to view the minor grid labels.

3.10.4 Changing the Line Styling

Change the line style, line colour and line weight of the title, footer or axes from a selected surface graph.

Select the surface graph where you want to change the line style, line colour and line weight and click the title, footer or axes.

1. Change the line style of the selected title, footer or axes.
   a) On the **Format** tab, in the **Line** group, click the **Line style** icon.
   b) Select one of the following:
      - To remove the line style, click **None**.
      - To modify the line style, click the line style you want to use.
2. Change the line colour of the selected title, footer or axes.
   a) On the **Format** tab, in the **Line** group, click the **Line colour** icon.
   b) Select one of the following:
      - To modify a colour, click the marker colour you want to use.
      - To add a colour that is not included as one of the basic colours, click **More colours**.
3. Change the line weight for the selected title, footer or axes.
   a) On the **Format** tab, in the **Line** group, click the **Line weight** icon.
   b) Select the line weight you want to use.

3.10.5 Locking the Aspect Ratio

Lock the proportional relationship between the independent axes on Cartesian surface graphs to keep the graph dimensions undistorted and true to the 3D model.

1. On the **Surface** contextual tabs set, on the **Display** tab, in the **Axes** group, click the **Lock aspect ratio** icon.
2. Select one of the following:
   - To view the true aspect ratio between the independent axes, click **Enable locked aspect ratio**. The full graph area is not utilised when displaying the surface graph.

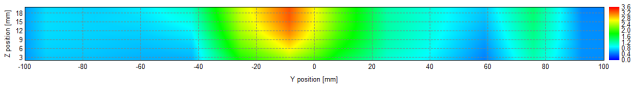


Figure 416: An example of a near field request where the aspect ratio is locked.

   - To resize the independent axes to allow the full graph area to be utilised, click **Disable locked aspect ratio**.

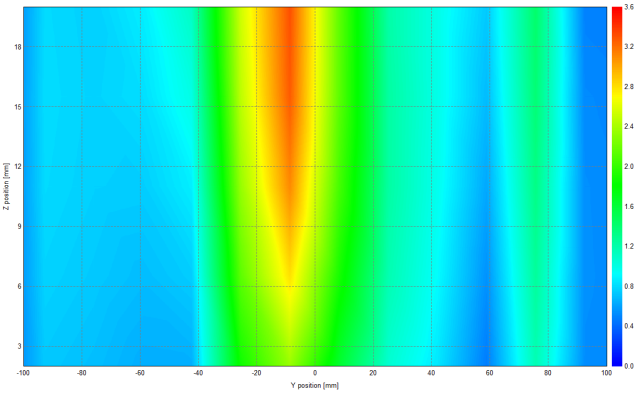


Figure 417: An example of a near field request where the aspect ratio is not locked.

   - To lock the original aspect ratio for the cases where the independent axes have the same units automatically, click **Auto lock aspect ratio**.

3.10.6 Swopping the Independent Axes

The two independent axes on a Cartesian surface graph can be interchanged when required.

In the result palette, click the icon to interchange the independent axes.

3.10.7 Specifying the Major Axes Range

Specify the range for the major axes.

Select the graph where you want to change the axis range.

1. On the **Surface** contextual tabs set, on the **Display** tab, in the **Axes** group, click the **Axis settings** icon.

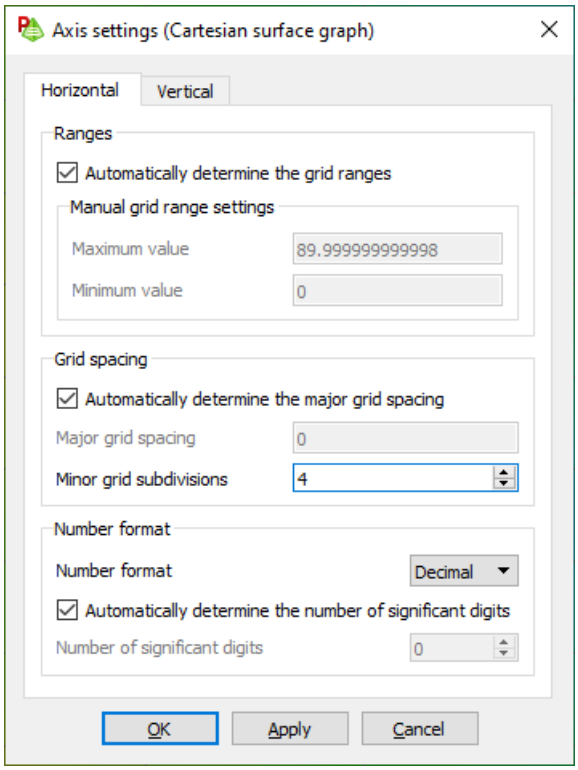


*Figure 418: The **Axis settings** dialog.*

2. Select the axis that you want to modify.
   - To modify the grid range for the horizontal axis, click **Horizontal**.
   - To modify the grid range for the vertical axis, click **Vertical**.
3. Under **Ranges**, select one of the following:
   - To automatically determine the grid range, select the **Automatically determine the grid range** check box.
   - To specify the grid range, clear the **Automatically determine the grid range** check box.
     - In the **Maximum value** field, enter a value for the upper limit of the graph.
     - In the **Minimum value** field, enter a value for the lower limit of the graph.
4. Click **OK** to apply the settings and to close the dialog.

3.10.8 Reversing the Axis Order

Change the order in which values are plotted along the axis of a Cartesian surface graph.

The axis order can be reversed for both the horizontal axis and the vertical axis.

As an example, the vertical axis order of a Cartesian surface graph is reversed. The steps are similar for reversing the horizontal axis order.

1. Select the Cartesian surface graph where you want to reverse the vertical axis order.
2. On the **Surface** contextual tabs set, on the **Display** tab, in the **Axes** group, click the **Reversed order (vertical)** icon.

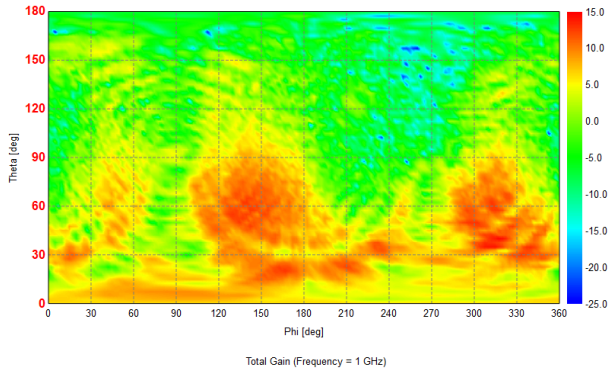


Figure 419: An example of a Cartesian surface graph.

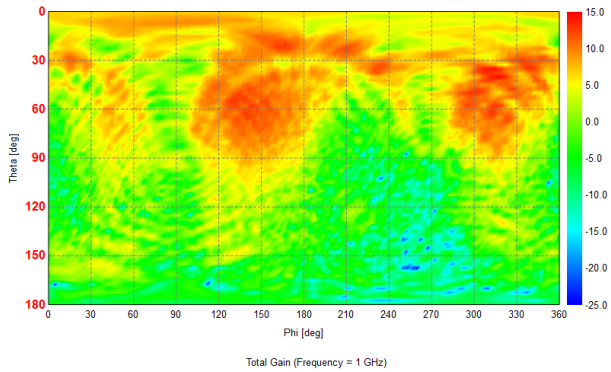


Figure 420: An example of a Cartesian surface graph where the order of the vertical axis was reversed.

3.10.9 Storing a Local Copy of a Data Set

Stores a local copy of the underlying data that is represented by the Cartesian surface graph. By storing a local copy, you can modify the existing model and compare the old results to the new results.

1. Select the surface graph result in the result palette that you want to store.
2. On the **Surface** contextual tabs set, on the **Result** tab, select the **Store a copy**.

A new entry under **Stored data** is created that is accessible from the project browser or the ribbon.

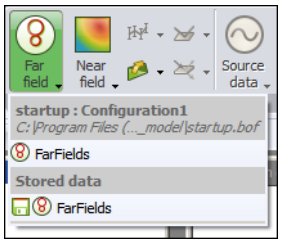


Figure 421: Accessing stored data from the ribbon.

3.10.10 Changing the Sampling Settings

Specify the number of samples for continuous results displayed on surface graphs.

The default rendering for continuous data on a graph is determined automatically. The sampling can be adjusted to display either the actual frequency samples or adjusted to display a set number of frequency points.

1. Select the graph for which you want to modify the sampling.
2. On the **Surface** contextual tabs set, on the **Result** tab, in the **Rendering** group, click the **Sampling settings** icon.

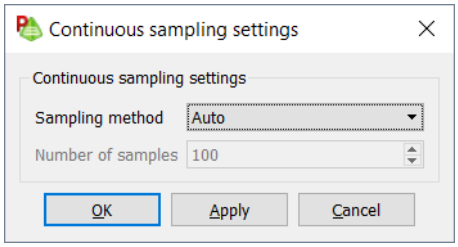


*Figure 422: The **Continuous sampling settings** dialog.*

3. From the **Sampling method** drop-down list, select one of the following:
   - To display the default rendering where the sampling is determined based on the data, select **Auto**.
   - To display only the actual samples, select **Discrete**.
   - To resample the data and display a fixed number of discrete points, select **Specify number of samples**.
4. Click **OK** to set the sampling settings and to close the dialog.

3.10.11 Adding a Quick Single Point Annotation

Read a single point from a Cartesian surface graph result by adding a quick single point annotation.

1. Select the Cartesian surface graph where you want to add the annotation.
2. Position the mouse cursor on the Cartesian surface graph result.
3. Press Ctrl+Shift+left click.

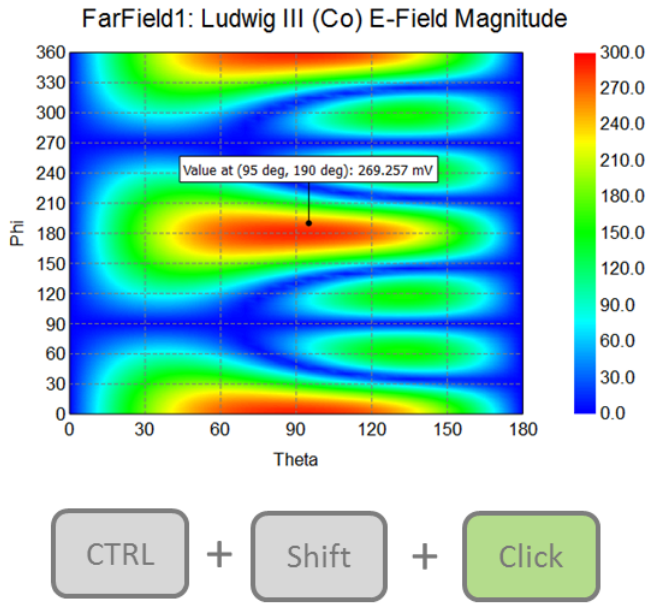


Figure 423: Press Ctrl+Shift+left click to add an annotation.

3.10.12 Adding a Custom Point Annotation

Add a custom point annotation to a Cartesian surface graph to read the value of a point and highlight this point of interest on the graph.

Select the Cartesian surface graph where you want to read the point.

1. On the **Surface** context tab, on the **Measure** tab, on the **Custom annotations** group, click the **Point** icon. From the drop-down list, select **Specify independent axis value**.

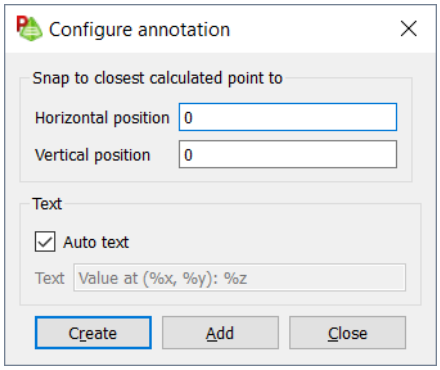


*Figure 424: The **Configure annotation** dialog.*

2. In the **Horizontal position** field, specify a value on the horizontal axis.
3. In the **Vertical position** field, specify a value on the horizontal axis.
4. Specify the text displayed in the annotation.
   - To specify text, clear the **Auto text** check box. In the **Text** field, enter the custom text.
   - To add text containing the X value and Y axis, select the **Auto text** check box.
5. Click **Create** to add the annotation and to close the dialog.

3.10.13 Duplicating a Cartesian Surface Graph

Make a copy of the current Cartesian surface graph complete with all the settings.

On the **Surface** contextual tabs set, on the **Display** tab, in the **Duplicate** group, click the **Duplicate view** icon.

3.10.14 Legend Range Settings

Adjust the legend range for the current active surface graph.

Cartesian surface graph data can be clamped between two values to help reveal changes in a result that would be missed with the default range. The colouring of the result is changed whereby blue corresponds to the minimum value and red to the maximum value.

On the **Surface** contextual tabs set, on the **Display** tab, in the **Legends** group, click the dialog launcher.

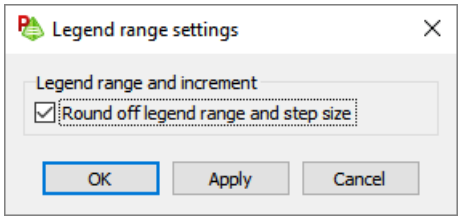


*Figure 425: The **Legend range settings** dialog.*

The following settings are available:

Round off legend range and step size	Data is often represented in a decimal form to represent a value accurately. The number of digits after the decimal point could result in a legend range that is difficult to read and interpret. Select this option for a more legible legend containing rounded off values. The original data range is contained within the rounded off range.

3.10.15 Scaling and Display Settings

A number of settings are available that affects how a result is scaled and displayed in a surface graph. These settings influences the colour scaling of the legend and displayed colour range.

On the **Surface** contextual tabs set, on the **Display** tab, in the **Legends** group, click the **Individual range** icon.

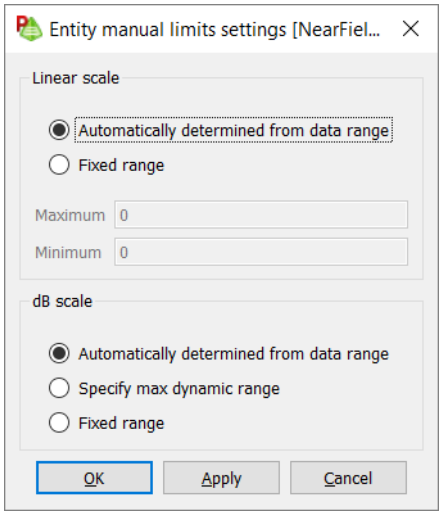


*Figure 426: The **Entity manual limits settings** dialog.*

Linear Scaling

For linear scaling, the following options are available to control the value range of the result:

Automatically determined from data range	This option is applicable when you want the value range to clamp between the maximum and minimum values of the result.
Fixed range	This option is applicable when you want to specify the maximum and minimum values of the data range.

dB Scale

For dB scaling, the following options are available to control the value range of the result:

Automatically determined from data range	This option is applicable when you want the value range to clamp between the maximum and minimum values of the result.
Fixed range	This option is applicable when you want to specify the maximum and minimum values of the data range.
Specify max dynamic range	This option is applicable when you want the maximum value of the result data to be used as the upper limit for the legend values. The minimum value of the result data is the maximum value of the result data minus the dynamic range value entered or the minimum value of the result data, whichever is larger.

Note: These settings affect the dynamic range limits.

3.10.16 Exporting Plotted Data to Clipboard or a Text File

Retrieve the plotted data and save to file.

1. Select the Cartesian surface graph that you want to export.
2. Export the plotted data using one of the following workflows:
   - To export the data to a `.dat` file, from the right-click context menu select **to file (*.dat)** and specify the file name.
   - To export the data to clipboard, press Ctrl+X.

3.11 3D Views

View the simulation data in a 3D view to allow visual interpretation of the data in a human-readable format, as well as to communicate the results in reports and presentations. The 3D view can also be used to verify that the CADFEKO or EDITFEKO model is correct.

When a 3D result is added to a 3D view and the Solver is run, POSTFEKO monitors the simulation results and updates the 3D view as the results become available for discrete frequency results.

For adaptive frequency sampling results (continuous frequency), POSTFEKO displays the discrete results during the simulation and interpolate the results once the simulation is complete.

Related concepts
3D Result (Terminology)
Continuous Frequency (CADFEKO)

3.11.1 Creating a New 3D view

Create a new 3D view to verify the model and visualise the results.

1. On the **Home** tab, in the **Create new display** group, click the **3D view** icon.
2. From the drop-down list select one of the following:
   - To select a configuration to add to a new 3D view, select a configuration from the list.
   - To create a new empty view, select **3D view**.

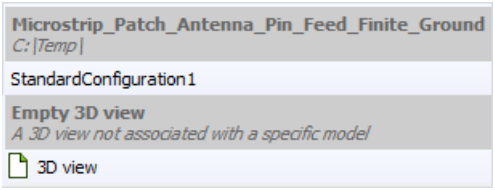


Figure 427: The drop-down list when a new 3D view is requested.

> **Tip:** You can create multiple 3D views where each view has different settings.

3.11.2 Adding a Cutplane

Create a sectional view of the model by using a cut plane to show internal details that would otherwise be hidden. Multiple cutplanes are supported.

1. On the **3D View** contextual tabs set, on the **Display** tab, in the **Display** group, click the **Cutplanes** icon.
2. Click the **Plane definition** tab.

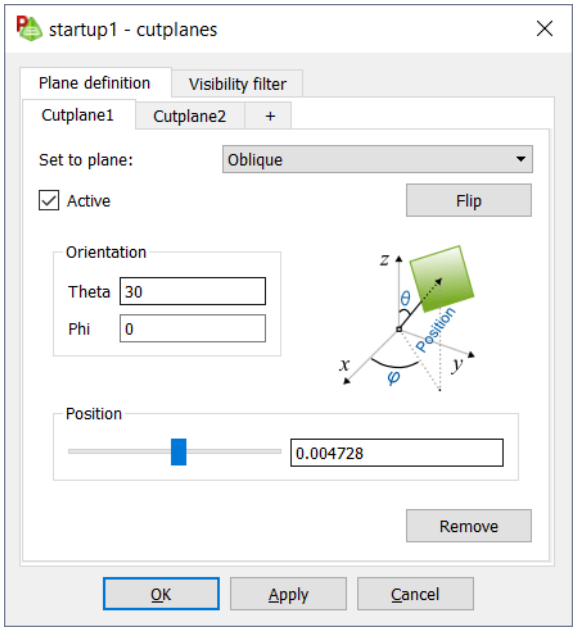


*Figure 428: The **Cutplanes** dialog, **Plane definition** tab.*

a) [Optional] To create additional, click **+**.

> **Note:** Click **Remove** to delete the cutplane.

b) In the **Set to plane** drop-down list, select one of the following for the orientation of the cutplane:
   - **Global YZ**
   - **Global XZ**
   - **Global XY**
   - **Oblique**
     - In the **Theta** field, specify the theta angle in degrees.
     - In the **Phi** field, specify the phi angle in degrees.

c) [Optional] Click the **Flip** to alternate the normal direction of the cutplane, which in turn determines which side of the plane is hidden.

d) Under **Position**, use the slider to place the cutplane at a specific location.

By default, everything in the model is affected by the cutplane. Entities that should be left uncut, can be specified.

3. [Optional] Click the **Visibility filter** tab.

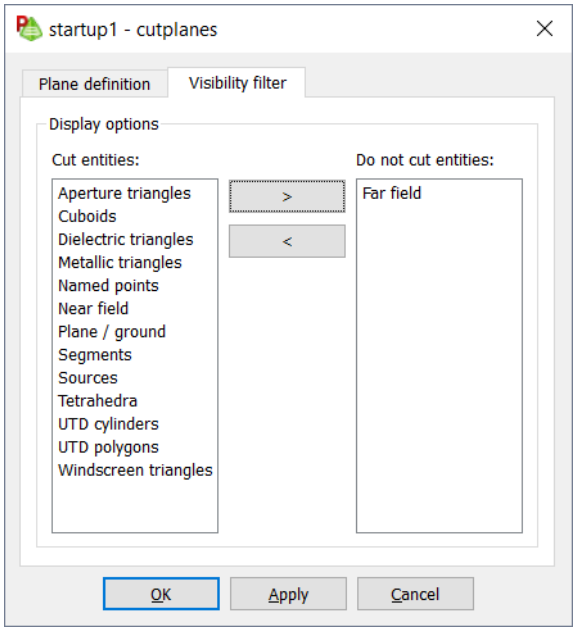


*Figure 429: The **Cutplanes** dialog, **Visibility filter** tab.*

a) To prevent an entity from being cut, in the **Cut entities** panel, select the entity and click **>**.

4. Click **OK** to define the cutplane and close the dialog.

3.11.3 Adding Legends to a 3D View

Add up to four legends to a predefined location on the 3D view. Bind the legend to a specific entity (for example, far field data or mesh display), based on the results displayed in the 3D view.

A legend can be placed top left, top right, bottom left and bottom right on the 3D view. As an example, the legend is placed top left, although the steps are similar to placing the legend at one of the three predefined locations.

1. Select the 3D view where you want to add a legend.
2. On the **3D View** contextual tabs set, on the **Display** tab, in the **Legends** group, click the **Top left** icon.
3. From the drop-down list, select a result to link to legend.

> **Note:** To remove a legend, select **None** from the drop-down list.

The legend is placed on the 3D view.

Legends Based on Multiple Results

Use a dialog for the active 3D view legend to make a selection if more than 19 results.

An active 3D view can display multiple result items. When a legend is added, a dialog is provided to select from all the displayed results.

If the number of results exceeds 19 items, click **More...** at the bottom of the legend drop-down list.

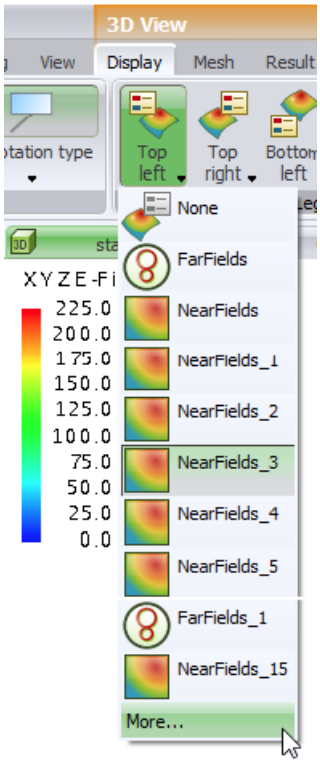


Figure 430: A legend with many result items.

This opens the **More...** dialog where all the results can be selected from.

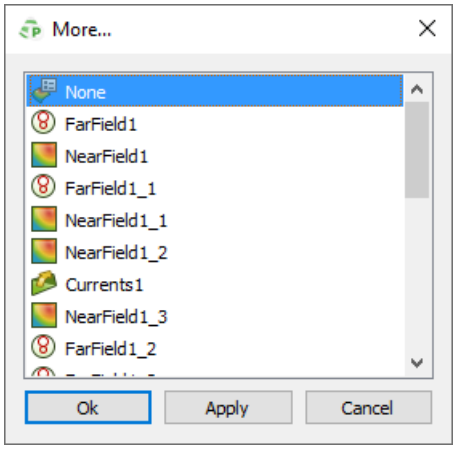


*Figure 431: The **More...** dialog.*

Legend Range Settings

Adjust the legend range for the current active 3D view.

3D view data can be clamped between two values to help reveal changes in a result that would be missed with the default range. The colouring of the result is changed whereby blue corresponds to the minimum value and red to the maximum value.

On the **3D View** contextual tabs set, on the **Display** tab, in the **Legends** group, click the dialog launcher.

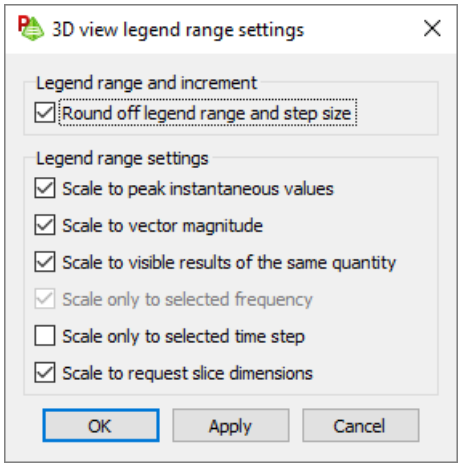


*Figure 432: The **3D view legend range settings** dialog.*

The following settings are available:

Round off legend range and step size	Data is often represented in a decimal form to represent a value accurately. The number of digits after the decimal point could result in a legend range that is difficult to read and interpret. Select this option for a more legible legend containing rounded off values. The original data range is contained within the rounded off range.
Scale to peak instantaneous values	This option is applicable when viewing the magnitude of a result along with the instantaneous phase. The minimum and maximum value limits (and therefore the colours) remain constant for each phase step. This option makes it simpler to see how the magnitude changes over phase. Clear the **Scale to peak instantaneous values** check box when the magnitudes at the given phase are of interest to synchronise the range limits with the displayed data.
Scale to vector magnitude	This option is applicable when comparing two components of a vector with one another. It is simpler to compare relative magnitudes of the components if displayed relative to the same maximum. Use this option to render the components relative to

	the total vector magnitude. This option allows a relatively small component to be distinguished from a larger component.
Scale to visible results of the same quantity	This option is applicable when a model contains several near field or far field requests. Each request has its minimum and maximum value. For multiple results on a view, POSTFEKO display the data relative to the minimum and maximum found over all the results added to the view. Deselecting this option renders each result according to its minimum, and maximum value.
Scale only to selected frequency	This option is only enabled when discrete frequency data is available in the model. The range limits are determined by the minimum and maximum values found over all the calculated frequency points. For example, it is more convenient to view the change in far field gain over frequency when the far field is scaled according to the maximum / minimum found over all calculated points.
Scale only to selected time step	This option is applicable for time domain analysis where the range limits are determined by the minimum and maximum values of the displayed data for a selected time signal sample.
Scale to request slice dimensions	This option is applicable when the range values are to be determined by the displayed slice[52].

Note: Each setting is applied independently, meaning that a wide variety of combinations are possible to help display and interpret the data in the desired manner.

Scaling and Display Settings

A number of settings are available that affects how a result is scaled and displayed in a view. These settings influences the colour scaling of the legend and displayed colour range.

On the **3D View** contextual tabs set, on the **Display** tab, in the **Legends** group, click the **Individual range** icon.

52. For example, if you have a 3D near field result with data in the X axis, Y axis and Z axis, a slice of data is a cut at a specific X value and Y value.

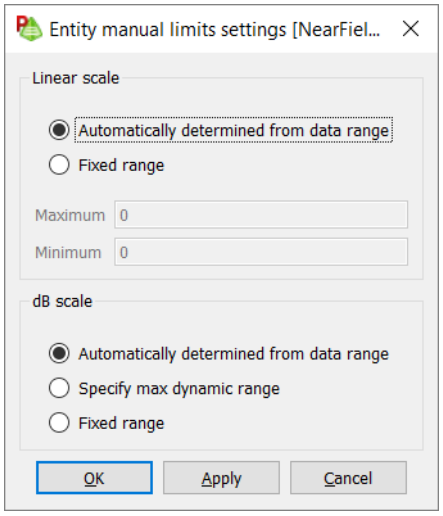


*Figure 433: The **Entity manual limits settings** dialog.*

Linear Scaling

For linear scaling, the following options are available to control the value range of the result:

Automatically determined from data range	This option is applicable when you want the value range to clamp between the maximum and minimum values of the result.
Fixed range	This option is applicable when you want to specify the maximum and minimum values of the data range.

dB Scale

For dB scaling, the following options are available to control the value range of the result:

Automatically determined from data range	This option is applicable when you want the value range to clamp between the maximum and minimum values of the result.
Fixed range	This option is applicable when you want to specify the maximum and minimum values of the data range.
Specify max dynamic range	This option is applicable when you want the maximum value of the result data to be used as the upper limit for the legend values. The minimum value of the result data is the maximum value of the result data minus the dynamic range value entered or the minimum value of the result data, whichever is larger.

Note: These settings affect the dynamic range limits.

3.11.4 Adding a Quick Single Point Annotation

Read a single point from a 3D view result by adding a quick single point annotation.

1. Select the 3D view where you want to add the annotation.
2. Position the mouse cursor on the 3D view result.
3. Press Ctrl+Shift+left click.

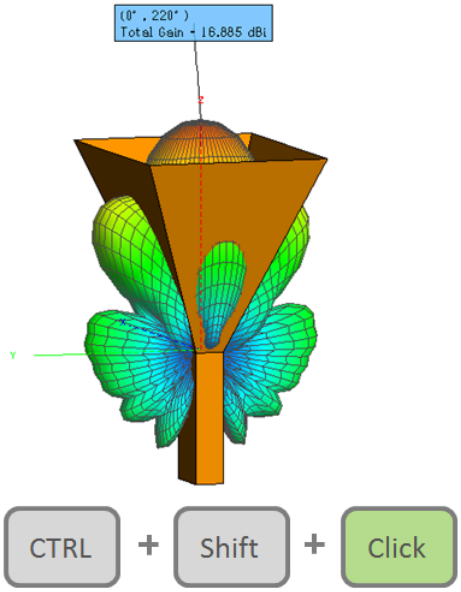


Figure 434: Press Ctrl+Shift+left click to add an annotation.

3.11.5 Adding Annotations to a 3D View Result

Add a single point annotation to a 3D view result. Multiple annotations can be added to a single 3D view.

1. Select the 3D view where you want to add a single annotation or multiple annotations.
2. On the **3D View** contextual tabs set, on the **Display** tab, in the **Display** group, click the **Annotation type** icon to enable.
3. From the **Annotation type** list, select one of the following:
   - To highlight and add an annotation to an element, select the **Highlight and annotate** icon.
   - To only highlight an element, select the **Highlight elements** icon.

3.11.6 Duplicating a 3D View

Create a duplicate view of a 3D view, and copy the display settings.

1. Select the 3D view you want to duplicate.
2. On the **3D View** contextual tabs set, on the **Display** tab, in the **Duplicate** group, click the **Duplicate view** icon.

3.11.7 Duplicating a 3D Simulation Result

Make a copy of the 3D simulation result.

1. Select the 3D view and 3D view simulation result (either in the 3D view or the result palette).
2. On the **3D View** contextual tabs set, on the **Result** tab in the **Manage** group, click the **Duplicate component** icon.

> **Tip:** You can also use one of the following workflows:
> - In the result palette or 3D view, select the trace. From the right-click context menu, select **Duplicate component**.
> - Press Ctrl+K.

3.11.8 Storing a Local Copy of a Dataset

Store a local copy of the underlying data that is represented by the 3D simulation result. By storing a local copy, you can modify the existing model and compare the old results to new results.

1. Select the 3D view and 3D view simulation result (either in the 3D view or the result palette) that you want to store.
2. On the **3D View** contextual tabs set, on the **Result** tab in the **Manage** group, click the **Store a copy** icon.
   A new entry under **Stored data** is created that is accessible from the project browser or the ribbon.

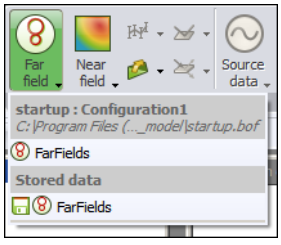


Figure 435: Accessing stored data from the ribbon.

3.11.9 Display Settings for 3D View

A number of display settings are available to customise the display of simulation results in the 3D view.

Visibility of Entities

All 3D entities are visible by default (except for named points), provided the model contains the entity.

The entity display options are available on the **3D view** context tab, on the **Display** tab, in the **Entities** group.

Table 26: Display options for 3D entities.

Icon	Name	Description
	Sources	Show / hide sources, such as voltage sources, current sources and plane waves.
	Loads	Show / hide loads.
	Point	Show / hide named points.
	Probes	Show / hide probes, such as voltage probes and current probes.
	Cables	Show / hide cables.
	Networks	Show / hide general non-radiating networks.
	TX Line	Show / hide non-radiating ideal transmission lines.
	RX antenna	Show / hide receiving antennas, such as far field receiving antennas and near field receiving antennas.

Display Settings for Sources and Loads

Show or hide specific types of sources or loads in the 3D view. A source type can be displayed while also coloured and scaled according to magnitude.

On the **3D View** contextual tabs set, on the **Display** tab, in the **Entities** group, click the ↘ dialog launcher.

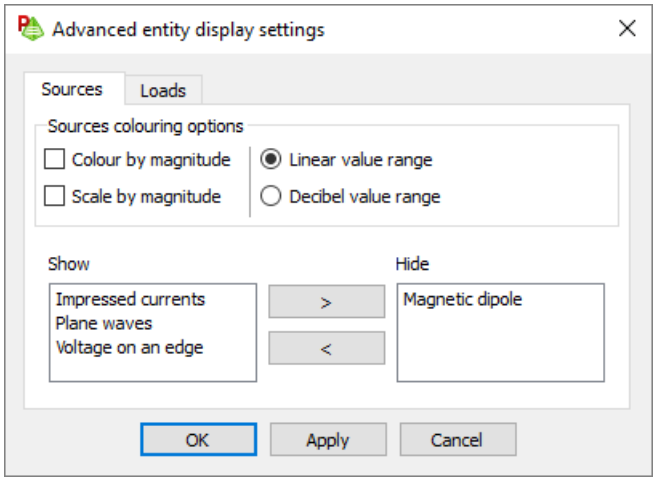


*Figure 436: The **Advanced entity display settings** dialog.*

You can use the **Show** and **Hide** panels to show or hide sources and loads selectively. Additional options for sources are **Colour by magnitude** and **Scale by magnitude**. These options are often used in conjunction with aperture sources, electric dipoles, magnetic dipoles and impressed currents.

Note: Sources and loads are only displayed if the visibility for these entities is enabled, regardless if they are placed in the **Show** panel.

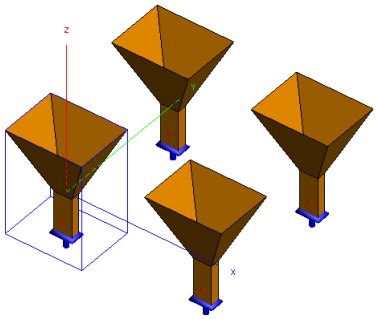
Visibility of Symmetry, FDTD Boundary, PBC Boundary and Array Base Element

Show or hide the display of symmetry, the finite difference time domain (FDTD) boundary, the periodic boundary condition (PBC) boundary and the array base element for finite arrays.

The method display options are available on the **3D view** context tab, on the **Display** tab, in the **Method display** group.

Table 27: Display options for symmetry, finite antenna arrays, PBC and FDTD boundary condition.

Icon	Icon text	Description
	Array Base Element	Show / hide the finite antenna array base element.

Icon	Icon text	Description
		 *Figure 437: A blue bounding box indicates the base element of the finite antenna array.*
	**Symmetry**	Show / hide defined symmetry.
	**Periodic Boundary Conditions**	Show / hide periodic boundary conditions.
	**FDTD boundary type**	Show / hide the FDTD boundary.

Axes Settings

Show or hide the display of the main axes, mini axes and tick marks.

The axes display options are available on the **3D view** context tab, on the **Display** tab, in the **Axes** group.

Table 28: Display options for axes and tick marks.

Icon	Icon text	Description
	**Main axes**	Show / hide the main axes.
	**Mini axes**	Show / hide the mini axes.
	**Tick Marks**	Show / hide tick marks on the main axes.

Specifying the Axis Size and Tick Marks

The axis size in relation to the 3D view can be specified as well as the placement of tick marks on the axes.

Specify the axis size.

1. On the **3D View** contextual tabs set, on the **Display** tab, in the **Axes** group, click the dialog launcher.

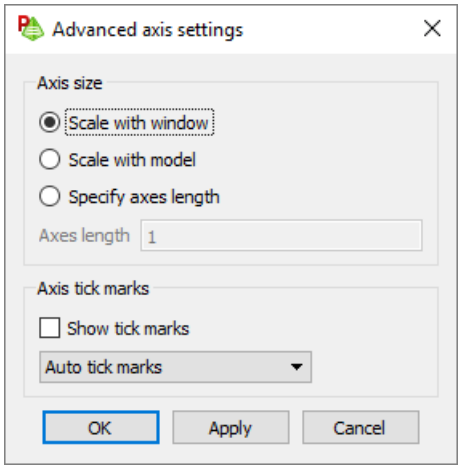


*Figure 438: The **Advanced axis settings** dialog.*

2. Under **Axis size**, select one of the following:
   - To scale the axis dynamically with the size of the 3D view, select **Scale with window**.
   - To scale the axis along with the model size, select **Scale with model**.
   - To specify the length of the axes, select **Specify axes length** and specify the axes length.

Specify the axis tick marks.

3. To display the axes tick marks, select the **Show tick marks** check box.
4. Specify the placement of the tick marks.
   - To place the tick marks at the default location on the axis, select **Auto tick marks**.
   - To specify the number of tick marks on the axis, select **Number of tick marks**.
   - To specify the increment between the tick marks on the axis, select **Increment between tick marks**.
5. Click **OK** to close the dialog.

Mesh Display Settings

A number of mesh display settings are available to give you full control of the mesh rendering in the 3D view. These settings are useful if you want to verify the model (simulation mesh).

Mesh Rendering Options

View the display options for mesh colour, model outline, segment radius, wire coating, anisotropic media, windscreen layers and mesh normals.

The mesh rendering options are available on the **3D View** contextual tabs set, on the **Mesh** tab in the **Rendering** group.

> **Note:** Only the rendering options relevant to the model are available. For example, if the model does not contain any windscreens, the **Windscreen Layers** icon is disabled.

Table 29: Display options for mesh colour, model outline, segment radius, wire coating, anisotropic media, characterised surfaces, windscreen layers and mesh normals.

Icon	Icon text	Description
	**Mesh colour**: Element face media	The mesh faces are coloured according to their media.
	**Mesh colour**: Element region media	The mesh is coloured according to the surrounding region. For example, a model in free space is displayed in red to indicate that the outside region is free space.
	**Mesh colour**: Element label	The mesh is coloured according to mesh labels.
	**Mesh colour**: Element normal	An element normal defines the two sides of a triangle face (normal side and reverse side). The element normal for a triangle face is determined by order of the triangle vertices according to the right-hand rule. The normal side is coloured green, while the reverse side is coloured brown.
	**Mesh colour**: Element type	The mesh is coloured according to mesh types contained in the mesh. For example, different colours are be used for wire segments, metallic triangles and dielectric triangles.
	**Model outline**	Highlight the edges of the model faces.
	**Segment radius**	Specify the display size for the segment radius.
	**Coating**	Show / hide visibility of coatings on mesh wires and triangles.

Icon	Icon text	Description
	Orientation	Display the principal direction (for anisotropic layers), coordinate system (for 3D anisotropic media) or the reference vector orientation for characterised surfaces.
	Windscreen Layers	Show / Hide the display of the windscreen layer thickness.
	Normals	Show / Hide the display of mesh normals.

Viewing Anisotropic Layers

Validate your model that contains multi-layered anisotropic composite media, by viewing the principal direction.

1. On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Rendering** group, click the **Orientation** icon. From the drop-down list select the **Layer** icon.

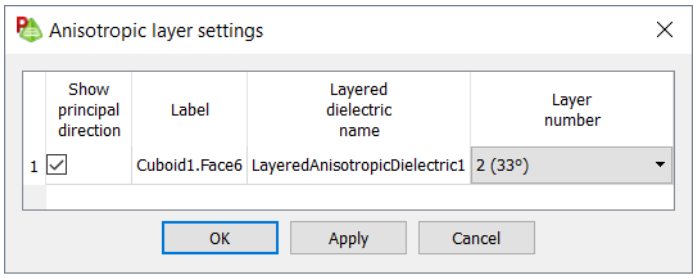


*Figure 439: The **Anisotropic layer settings** dialog.*

Each face that has an anisotropic layer applied to is listed on the **Anisotropic layer settings** dialog.

> **Note:** An anisotropic layer is applied to a label.

2. For each face, you can select the **Show principal direction** check box and specify the **Layer number** to show the direction in the 3D view.
3. Click **OK** to close the dialog.

Viewing the Coordinate System for Anisotropic (3D) Media

Validate your model that contains anisotropic (3D) media, by viewing the principal direction for each medium.

1. On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Rendering** group, click the **Orientation** icon. From the drop-down list select the **Media (3D)** icon.

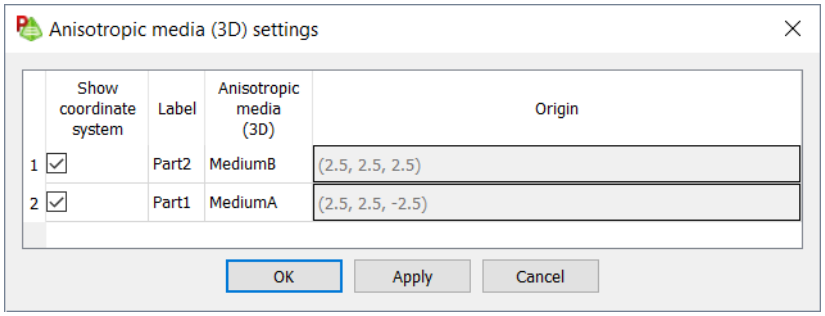


*Figure 440: The **Anisotropic media (3D) settings** dialog.*

Each region that has an anisotropic (3D) medium applied to is listed on the **Anisotropic media (3D) settings** dialog.

Note: An anisotropic (3D) medium is applied to a label.

2. For each region, you can select the **Show coordinate system** check box to show the coordinate system in the 3D view.
3. Click **OK** to close the dialog.

Viewing the Reference Vector Orientation for Characterised Surface Mesh Elements

Validate your model that contains characterised surface mesh elements, by viewing the reference vector orientation.

1. On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Rendering** group, click the **Orientation** icon. From the drop-down list select the **Characterised Surface** icon.

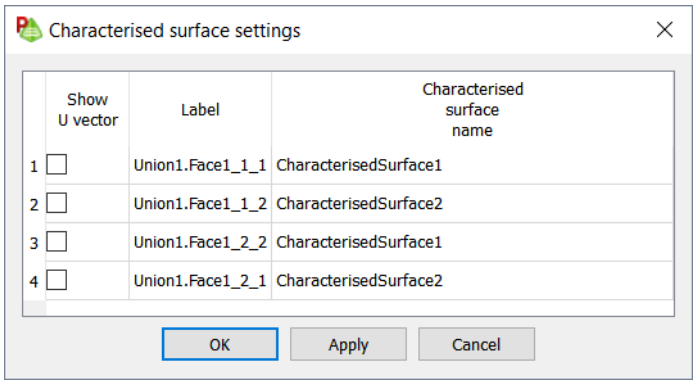


*Figure 441: The **Characterised surface settings** dialog.*

Each face that has a characterised surface applied to, is listed on the **Characterised surface settings** dialog.

2. For each face, you can select the **Show U vector** check box to show the direction in the 3D view.
3. Click **OK** to close the dialog.

The start of the vector (the coordinate system origin) is indicated by a yellow dot. The vector is displayed as a blue line to indicate that it is aligned with the U reference direction.

Mesh Opacity Settings

Specify the mesh opacity as well as the opacity of windscreens triangles and aperture triangles.

The mesh opacity settings are available on the **3D View** contextual tabs set, on the **Mesh** tab in the **Opacity** group.

Note: Only the opacity settings relevant are available. For example, if the model does not contain any windscreens the **Windscreen** icon is disabled.

Table 30: Display options for mesh opacity, windscreens triangles and aperture triangles.

Icon	Icon text	Description
	Mesh opacity	Specify the opacity for mesh elements.
	Windscreen	Specify the opacity for windscreen elements.
	Aperture	Specify the opacity for aperture elements.

Note: For all the opacity settings, a drop-down list is available to specify a custom opacity level. A level of 100% is equivalent to setting no opacity, and 0% is equivalent to full transparency.

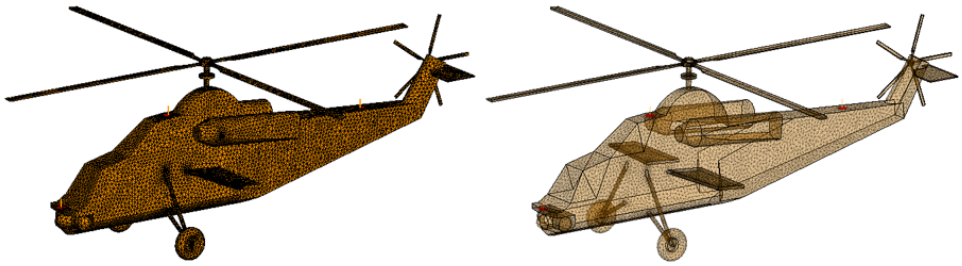


Figure 442: Mesh opacity: 100% (left) and 20% (right).

Mesh Visibility Settings

View the visibility settings for segments, triangles, apertures, windscreens, tetrahedra, voxels, cuboids and uniform theory of diffraction (UTD) polygons and cylinders.

The mesh visibility settings are found on the **3D View** contextual tabs set, on the **Mesh** tab in the **Visibility** group.

Note: Only the visibility settings relevant to the model will be available. For example, if the model does not contain any voxels the **Voxels** icon is disabled.

Table 31: Display options for segments, triangles, apertures, windscreens, tetrahedra, voxels, cuboids and uniform theory of diffraction (UTD) polygons and cylinders.

Icon	Icon text	Description
	**Segments**	Show / hide the surfaces, lines and vertices of mesh segments.
	**Metal**	Show / hide the faces, edges and vertices of metal triangles.
	**Dielectric**	Show / hide the faces, edges and vertices of dielectric triangles.
	**Aperture**	Show / hide the faces, edges and vertices of aperture triangles.
	**Windscreen**	Show / hide the faces, edges and vertices of windscreen triangles.
	**Tetrahedra**	Show / hide the faces, edges, vertices and volumes of tetrahedra.
	**Voxels**	Show / hide the faces, edges, wire lines, wire surfaces, volumes and grid of voxels.
	**Cuboids**	Show / hide the faces, edges and vertices of cuboids.[53]
	**UTD polygons**	Show / hide the faces, edges and vertices of UTD polygons.
	**UTD cylinders**	Show / hide faces and edges of UTD cylinders.

Note: For all of these visibility settings, a drop-down list is available to individually set the visibility for the faces, edges and vertices of the elements. For volumetric elements, an additional volume option is provided.

Visibility Filter

The visibility filter provides additional control over the visibility of mesh elements. With this filter, mesh regions with specific labels or specific media can be filtered out.

53. Cuboidal mesh elements can only be created in EDITFEKO.

On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Visibility** group, click the **Visibility filter** icon.

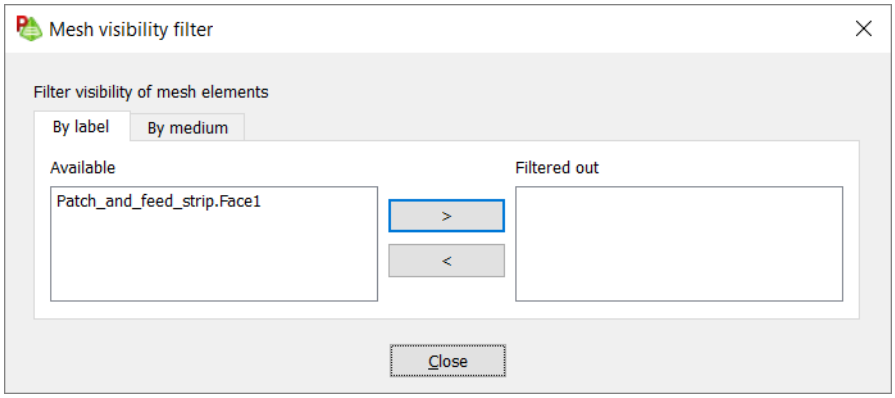


*Figure 443: The **Mesh visibility filter** dialog.*

Result Settings

A number of display settings are available to customise 3D view results.

Rendering Settings for Results

View the display settings for simulation results of far fields, near fields, currents and error estimates.

The result rendering settings are available on the **3D View** contextual tabs set, on the **Result** tab, in the **Rendering** group.

> **Note:** Only the rendering settings relevant to the displayed results are enabled. For example, if there are no near field iso-surfaces displayed, the **Colour** icon is disabled.

You can apply the rendering options to far fields, near fields, currents and error estimates.

Table 32: Display options for far fields, near fields, currents and error estimates.

Icon	Icon text	Description
	Grid	Overlays a mesh grid on the result. Provides a sense of dimension to the 3D results.
	Surface	Show / hide the coloured surface for the result.
	Sampling settings	Adjusts the sampling settings for 3D continuous far fields.
	Discrete	Removes the interpolated colouring of a surface and uses a predefined set of colours to represent the surface.

Icon	Icon text	Description
	**Colour**	Specify the colour of the 3D near field iso-surfaces.
	**Offset**	Opens the **Set display offset** dialog for moving the display of the far field origin.
	**Opacity**	Specify the amount of transparency.
	**Size**	Scale the display of the far field.
	**Extrusion**	Specify the extrusion for near field surfaces.

Sampling Settings for Continuous Far Fields

View the sampling options for sampling 3D continuous far fields.

On the **3D View** contextual tabs set, on the **Result** tab, in the **Rendering** group, click the  **Sampling settings** icon.

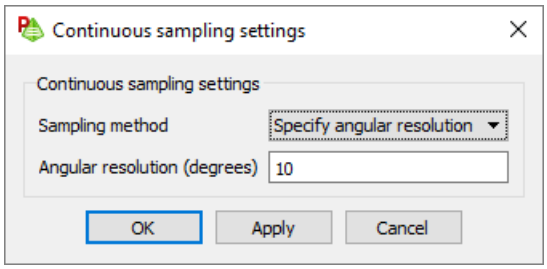


*Figure 444: The **Continuous sampling settings** dialog.*

The following sampling options are available:

Table 33: Sampling settings for continuous far fields.

Sampling Setting	Description
Auto	The default rendering option automatically determined from sampled data.
Request points	Disables continuous sampling and shows only the requested far field points.
Specify angular resolution	Specify a custom sampling interval.

Moving the Display of the Far Field Origin

Move the display of the far field origin to be more consistent with the location of the source.

Far fields are displayed in POSTFEKO around the global origin. When the source of the radiator is located away from the origin, the far field origin can be moved to show a more intuitive result.

1. Select the 3D view and far field result that you want to modify.
2. On the **3D View** contextual tabs set, on the **Result** tab, in the **Rendering** group, click the **Offset** icon.
3. On the **Set display offset** dialog specify new values for the U, V and N coordinates.

> **Tip:** You can use Ctrl+Shift+left click to click on the new origin for the far field.

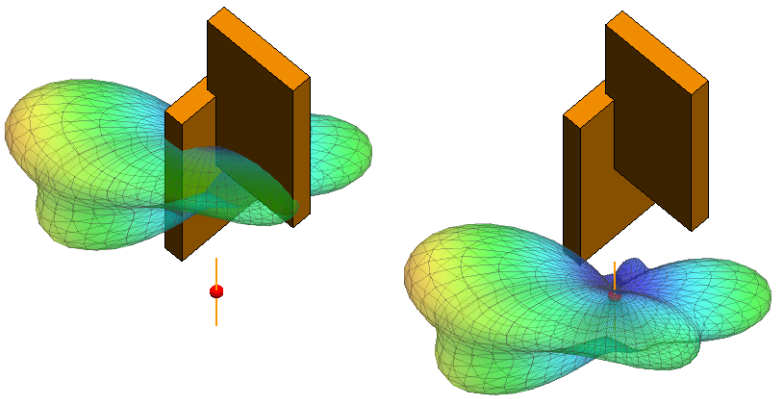


Figure 445: A dipole and PEC structure with a far field with no offset (on the left) and a far field with an offset (to the right).

Extruding Far Fields or Near Fields

Change the extrusion of a field result to view the data in a different way. By adding or removing depth to a surface, the relative impact of field values can be better understood.

Extrusion applies to far fields and near field surfaces that lie in a flat plane.

1. Select the 3D view and field result that you want to extrude.
2. On the **3D View** contextual tabs set, on the **Result** tab, in the **Rendering** group, click the **Extrusion** icon.
3. From the drop-down list, select one of the following:
   - Select a predefined percentage from the list.
   - To specify a custom percentage value, select **Custom**.
   - To allow POSTFEKO to decide on an extrusion value automatically, select **Auto**.

Table 34: The effect of extrusion on far fields and near fields.

Setting	Effect on Near fields	Effect on Far fields
0%	Flat surface.	A fixed radius sphere.
100%	A surface with a height dependent on the near field value.	A surface with a radius dependent on the far field value.
Auto	Same as **0%** setting.	Electric field, gain, realised gain and directivity - same as **100%** setting. Axial ratio and handedness - same as **100%** setting.
Custom	Dependent on user setting.	Dependent on user setting.

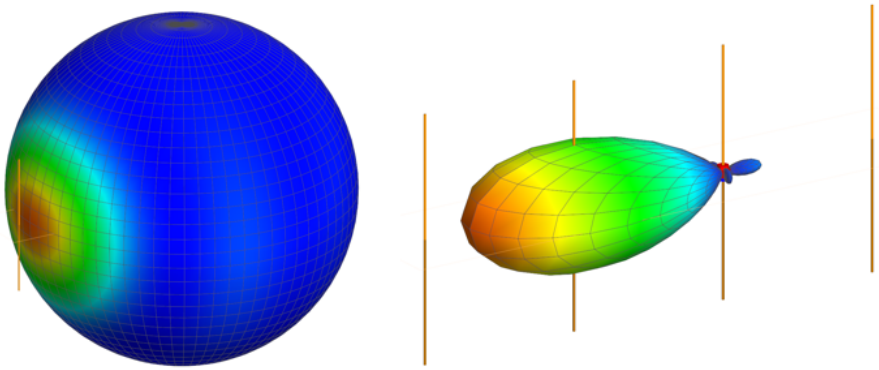


Figure 446: Examples of far field extrusion, 0% (on the left) and 100% (to the right).

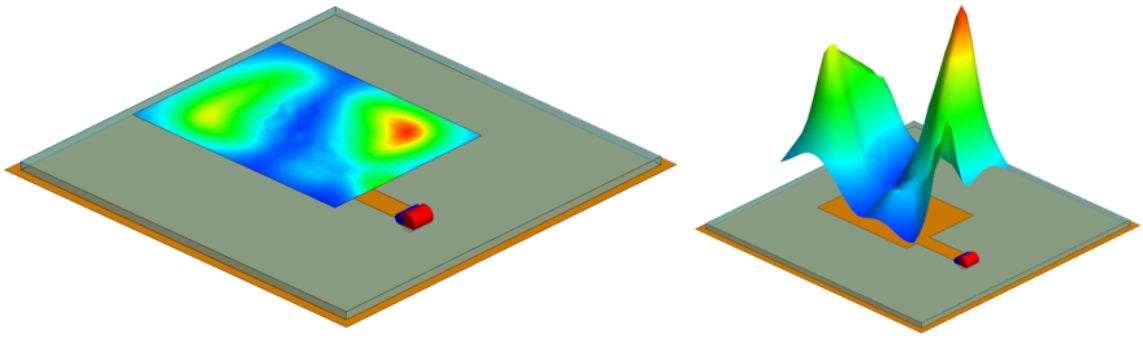


Figure 447: Example of near field extrusion, 0% (on the left) and 100% (to the right).

Display Options for Requests Points

Before a simulation is run, it is good practice to validate that the data were requested at the correct locations. Request points and the display of the near field boundary are used to verify that far fields and near fields requests are correct.

Requests points are displayed automatically if no result data is present. Once data becomes available, the result data is displayed and the request points are hidden.

The requests display settings are available on the **3D View** contextual tabs set, on the **Result** tab, in the **Requests** group.

Table 35: Display options for request points and near field boundary.

Icon	Icon text	Description
	**Auto request points display**	Request points are displayed when no result data is present. Once data is available, the request points are hidden.
	**Display request points**	Request points are always displayed.
	**Don't display request points**	Request points are never displayed.
	**Settings**	Specify the display type (points, lines or surfaces) colour and marker size for the request point.
	**Boundary**	Show / hide the near field boundary.
	**Axes**	Show / hide the local axes of the selected result.

Changing the Default Request Points Styling

Modify the request point type, colour and marker size.

1. Select the 3D view and the result where you want to change the request points styling.
2. On the **3D View** contextual tabs set, on the **Result** tab, in the **Requests** group, click the  **Settings** icon.

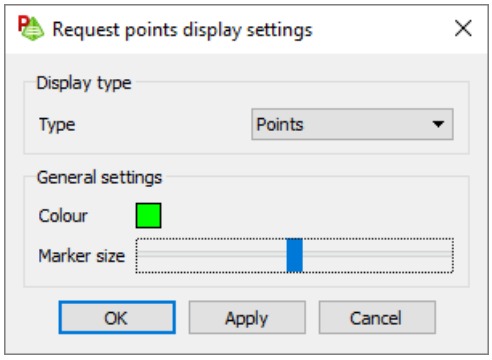


*Figure 448: The **Request points display settings** dialog.*

3. In the **Type** field, select one of the following:
   - **Points**
   - **Lines**
   - **Surface**
4. Next to **Colour**, click the colour block to specify the colour for the request points.
5. Next to **Marker size**, move the slider to specify the size. Left to right maps to small to large.
6. Click **OK** to apply the settings and to close the dialog.

Display Options for Contours

View the display options for contours. Contour lines are curves that connect points where a function has identical values.

The contour display settings are available on the **3D View** contextual tabs set, on the **Result** tab, in the **Contours** group.

Table 36: Display options for contours.

Icon	Icon text	Description
	Show contours	Show / hide contour lines.
	Colour	Set the colour of the contour to any value, or the colour is linked to the magnitude of the displayed value.
	Position	Specify the number of contours or the contour values.

Changing the Default Contour Positions

Specify the number of contours and its location on a 3D result to view points of equal value.

Enable the display of contours.

1. On the **3D View** contextual tabs set, on the **Result** tab, on the **Contours** group, click the **Show contours** icon.
2. On the **3D View** contextual tabs set, on the **Result** tab, in the **Contours** group, click the **Position** icon.

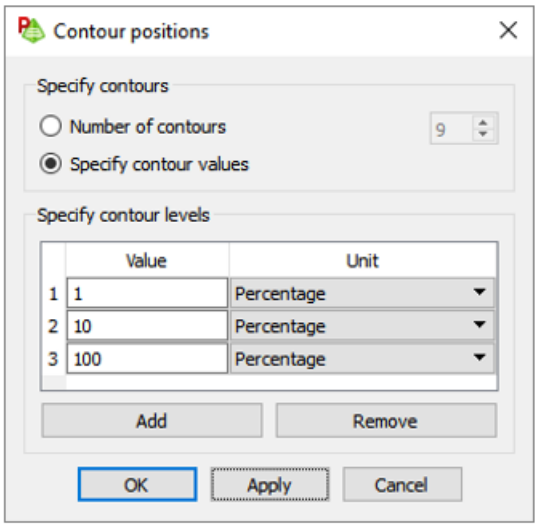


*Figure 449: The **Contour positions** dialog.*

3. Select one of the following workflows:
   - To specify the number of contours, select **Number of contours**. The contour values are evenly distributed over the result range.
   - To specify the number of contours coincident with a specific location, select **Specify contour values**. The contour location can either be defined by its magnitude value, or by specifying a percentage of the value range.
4. Click **OK** to close the dialog.

Display Options for Arrows

Display arrows to indicate the current flow direction or the field direction.

The arrow display settings are available on the **3D View** contextual tabs set, on the **Result** tab, in the **Arrows** group.

> **Note:** Arrows can be plotted if the instantaneous magnitude of a current result or a field result is displayed at a specific phase value.

Table 37: Display options for instantaneous vectors.

Icon	Icon text	Description
	Show arrows	Show / hide arrows.

Icon	Icon text	Description
	**Colour**	Sets the colour of the arrows to a user defined colour or dependent on the result magnitude.
	**Fixed size**	Enables or disables arrows displayed with a fixed size.
	**Arrow size**	Specify the arrow size as a percentage for the selected result

Display Options for Rays

View the ray display options for ray launching geometrical optics (RL-GO) and uniform theory of diffraction (UTD).

The UTD ray display settings are available on the **3D View** contextual tabs set, on the **Result** tab, in the **Rays** group.

The options are enabled on the POSTFEKO ribbon if a ray result is displayed in the 3D view. To obtain the ray file, you must select the option to export the ray file in CADFEKO.

Table 38: Display options for UTD rays.

Icon	Icon text	Description
	**Ray lines**	Show / hide ray lines.
	**Ray numbers**	Show / hide ray numbers. **Note:** A ray number is a unique number or ID associated with a ray.
	**Group numbers**	Show / hide the ray group numbers. **Note:** A ray group number is a unique number or ID associated with a group of rays which all belong to the same source, start at the same location or end at the same observation point.
	**Intersections**	Show / hide ray intersection points. The following abbreviations are used in the 3D view: • ▪: Creeping wave intermediate point on geometry surface • **B**: Diffraction at an edge

Icon	Icon text	Description
		• **D**: Diffraction at a corner or a tip • **E**: Diffraction at a corner (of an edge) • **K**: Diffraction at a wedge • **Q**: Source point • **R**: Reflection • **S**: Observation point • **C**: Creeping wave attaching and shedding point on geometry surface • **V**: Reflection at the shadow boundary of a creeping wave
	**Threshold**	Specify the visibility threshold of the rays as a percentage. • **0%**: All rays are displayed. • **100%**: All rays are hidden.
	**Colour magnitude**	Enables / disables the display of rays in colour according to its magnitude.

3.12 Frequency Domain Results

The Solver contains a number of frequency domain solution methods, as well as a time domain solution method. By default, all simulation results are obtained in the frequency domain, unless explicitly using the time analysis tool in POSTFEKO to convert the results to the time domain.

3.12.1 Result Types

View the result types that can be added to a 3D view or a graph.

The results types are available on the **Home** tab, in the **Add results** group.

Table 39: Result types that can be added to a graph or 3D view.

Icon	Icon text	Description
	**Far field**	The far field results, for example, electric field, gain, axial ratio and RCS.
	**Near field**	The near field results, for example, such as electric field, flux density, SAR and isosurfaces.
	**Error estimate**	Colours the mesh in the 3D view according to the results for the error estimation.
	**Currents**	The currents results such as electric currents, magnetic currents and charges.
	**Currents**	The currents results for wire segments such as electric currents and charges.
	**Rays**	Ray information for a model solved with the RL-GO or UTD.
	**Source data**	The source results, for example, input impedance, reflection coefficient and VSWR.
	**Loads / Networks**	The results for loads and networks, for example, voltage, current and power.
	**S-matrix**	The S-parameter results.
	**Power**	The results for source power, for example, power in the far field and loss power.
	**RX antenna**	The results for the receiving antenna, for example, received power and phase.

Icon	Icon text	Description
	**RX antenna**	The results for the far field receiving antenna, for example, received power and phase.
	**RX antenna**	The results for the near field receiving antenna, for example,received power and phase.
	**RX antenna**	The results for the spherical modes receiving antenna, for example, such as received power and phase.
	**Probes**	The probe data from a cable schematic such as voltage and current.
	**SAR**	The SAR data such as 1g, 10, and volume average.
	**Optimisation**	The optimisation data, for example, individual and global goal and parameter data.
	**Transmission / reflection**	The transmission and reflection coefficients.
	**Characteristic modes**	The characteristic modes data, for example, model excitation coefficient, eigenvalue and modal significance.
	**Imports + Scripts**	Results from imported data and Lua scripts.

POSTFEKO only enable the icons for results available in the current model or project.

Restrictions on the Display of Data

The result types that can be displayed depends on the view type (or graph).

3D views can display the following:

- far fields
- near fields
- error estimates
- currents
- rays
- SAR (only when calculated at a specific location)

Cartesian graphs can display all data, except for error estimates and currents on triangles (currents on segments can however be displayed).

Polar graphs can display data that varies according to angle (theta or phi). Only near fields and far fields meet this requirement.

Smith charts can display complex source data such as impedance and S-parameters.

Data can be imported for all graphs provided the data is consistent with the graph type.

Table 40: Summary of result types that can be plotted on various graphs types.

Result type	Cartesian	Smith	Polar
Characteristic modes	✓		
Far fields	✓		✓
Impedances	✓	✓	✓
Loads	✓	✓	✓
Near fields	✓		✓
Networks	✓		✓
Optimisation results	✓		
Power	✓		✓
Probes	✓		✓
Sources	✓	✓	✓
Parameters	✓	✓	
Transmission / reflection coefficients	✓	✓	✓
Wire segment data (Charges / Currents / Error estimates)	✓		✓

3.12.2 Adding a Result from the Ribbon

Add a result to a 3D view or graph using the ribbon.

1. Select the 3D view or graph where you want to add a simulation result.
2. On the **Home** tab, in the **Add results** group, click the relevant icon (if results are available).
3. If more than one result of the same request type are available, select a result.

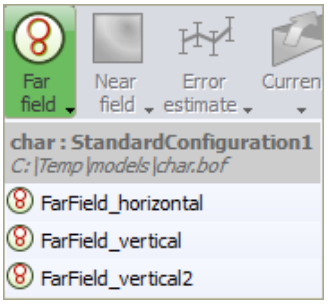


Figure 450: Example of adding a far field result from the ribbon and selecting a specific result.

4. If multiple models are loaded into the same session, POSTFEKO will collapse the panel.
5. [Optional] Click **Show more entries** to view all the available results.

3.12.3 Adding a Result from the Project Browser

Add a result to a 3D view or graph using the project browser.

1. Select the 3D view or graph where you want to add a simulation result.
2. In the project browser, select the model.
3. In the model browser, click the **Results** tab. Use one of the following workflows:
   - Drag a result onto the 3D view or graph.
   - From the right-click context menu and select **Add to active window** or click **Add to new** to create a new 3D view or graph.

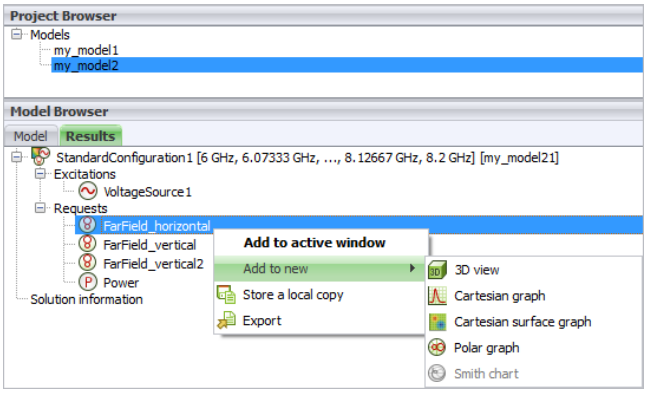


Figure 451: Example of adding a far field result from the project browser and selecting a result.

Far Fields

View the quantities and properties that are available for a far field request.

On the **Home** tab, in the **Add results** group, click the **Far field** icon.

Table 41: Properties for far field requests.

Quantity	Properties
Electric field	Total
Gain	Theta
Realised gain	Phi
Directivity	Ludwig III (Co)
Radar cross section (RCS)	Ludwig III (Cross)
	LHC
	RHC
Axial ratio	Minor / Major
	Major / Minor
Handedness	

The options available for far fields:

Total The total value independent of the polarisation.

Theta The vertical (or θ) component.

Phi The horizontal (or ϕ) component.

Ludwig III (Co) The reference polarisation as defined by Ludwig for conventional measurement configurations. An antenna that is Z directed implied for which the reference polarisation is intended along the $\phi = 90^{\circ}$ cut.

$$LIII_{Co}(\theta, \phi) = E(\theta, \phi) \cdot \left[\sin(\phi)\hat{\mathbf{i}}_{\theta} + \cos(\phi)\hat{\mathbf{i}}_{\phi}\right] \quad (16)$$

Ludwig III (Cross) The cross polarisation as defined by Ludwig for conventional measurement configurations. An antenna that is Z directed implied for which the reference polarisation is intended along the $\phi = 0^{\circ}$.

$$LIII_{Cross}(\theta, \phi) = E(\theta, \phi) \cdot \left[\cos(\phi)\hat{\mathbf{i}}_{\theta} - \sin(\phi)\hat{\mathbf{i}}_{\phi}\right] \quad (17)$$

Conventions for the Ludwig coordinate system are defined by the following parameters:

θ and ϕ
: Rotational angles in the spherical coordinate system as defined in Feko.

$\hat{\mathbf{i}}_\theta$
: Directional unit vector in the θ direction.

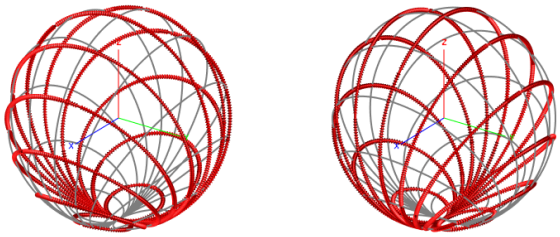


Figure 452: The reference and cross polarisations in 3D space.

LHC
: The left hand circularly polarised component. The polarisation vector rotates counter clockwise when viewed from a fixed position in the direction of propagation.

RHC
: The left hand circularly polarised component. The polarisation vector rotates counter clockwise when viewed from a fixed position in the direction of propagation.

Z (+45°)
: When viewed in the direction of propagation, the θ unit vector points downwards and the $\phi = 90^\circ$ unit vector to the left. The Z-polarisation vector is then

$$\hat{\mathbf{i}}_Z = \frac{\left(\hat{\mathbf{i}}_\theta + \hat{\mathbf{i}}_\phi\right)}{\sqrt{2}} \tag{18}$$

which lies along an axis rotated +45 degrees from horizontal (in a counter clockwise direction) — coinciding with the direction of the diagonal line of the Z.

S (-45°)
: The S-polarisation unit vector is

$$\hat{\mathbf{i}}_S = \frac{\left(\hat{\mathbf{i}}_\theta + \hat{\mathbf{i}}_\phi\right)}{\sqrt{2}} \tag{19}$$

which rotated by -45° from horizontal and lies in the direction approximated by the diagonal of the S.

Minor/Major
: Displays the magnitude of the axial ratio using the axes specification, **Minor/Major**.

Major/Minor
: Displays the magnitude of the axial ratio using the axes specification, **Major/Minor**.

Handedness	Displays the sign information for axial ratio on a sphere using different colours for left hand rotating, linear and right rotating fields.

Near Fields

View the quantities and properties that are available for a near field request.

On the **Home** tab, in the **Add results** group, click the **Near Fields** icon.

Table 42: Properties for near field requests.

Quantity	Properties
Electric field	Scale near field power
Magnetic field	X
Electric flux density	Y
Magnetic flux density	Z
Poynting vector	Rho
	Phi
	Z
	R
	Theta
SAR	Scale near field power

When any quantity (with the exception of **SAR**) and **Magnitude** is selected, POSTFEKO displays the vector magnitude of all the selected components. If only one component is selected, POSTFEKO can display the **Phase**, **Real** or **Imaginary** part of this component.

Currents and Charges

View the quantities and properties that are available for a current request.

On the **Home** tab, in the **Add results** group, click the **Currents** icon.

Table 43: Properties for current requests.

Quantity	Properties
Electric currents	Magnitude
Magnetic currents	Instantaneous magnitude
Charges	

Note: Magnetic currents are only applicable on dielectric surfaces modelled with the surface equivalence principle (SEP).

Error Estimation

View the quantities and properties that are available for an error estimation request.

On the **Home** tab, in the **Add results** group, click the **Error Estimation** icon.

The following quantities are available for error estimation:

- All mesh elements
- Triangles
- Segments

UTD / RL-GO rays

View the quantities and properties that are available when UTD or RL-GO rays are requested.

On the **Home** tab, in the **Add results** group, click the **Rays** icon.

Note: UTD or RL-GO rays are not stored by default due to possible large file sizes. The rays must be explicitly requested.

Source Data

View the quantities that are available for voltage, current and waveguide sources as well as FEM modal ports.

On the **Home** tab, in the **Add results** group, click the **Source data** icon.

Table 44: Source quantities.

Quantity	Voltage and Current sources	Waveguide sources and FEM modal ports
Impedance	✓	
Admittance	✓	
Voltage	✓	
Current	✓	
Reflection coefficient	✓	✓
VSWR	✓	
SWR		✓
Source power	✓	
Power loss due to mismatch	✓	
Mismatch loss	✓	✓

Note: The quantities listed are only available for sources on ports. For ideal sources, such as plane waves and electric dipoles, or equivalent sources such as far field and near field sources, source data is not available since these sources are, per definition, not connected to any geometry.

Loads and Networks

View the quantities and properties that are available for loads and networks.

On the **Home** tab, in the **Add results** group, click the **Loads/Networks** icon.

Table 45: Quantities for loads and networks

Quantity	Networks	Loads
Impedance	✓	✓

Quantity	Networks	Loads
Voltage	✓	✓
Current	✓	✓
Power	✓	✓
Power in	✓	

Note: The result palette for loads have a similar layout except there is no **Port number**.

S-parameters

View the settings for S-parameters.

On the **Home** tab, in the **Add results** group, click the **S-matrix** icon.

The **S-parameter** is the only quantity for this request. The number of selectable results for S-parameter will depend on the ports that were selected to be included in this request.

Power

View the quantities and properties that are available for power.

On the **Home** tab, in the **Add results** group, click the **Power** icon.

The available quantities for power differ substantially based on the selected entity. For example a load does not have the **Active power** quantity.

Table 46: Quantities for power

Quantity	Power	FarField	NearField
Active power	✓		
Loss power	✓		
Efficiency	✓		
Total radiated power		✓	

Quantity	Power	FarField	NearField
Power transmitted through surface			✓

> **Note:** The result palette for loads has a similar layout except there is no **Port number**.

Receiving Antennas

A receiving antenna data described by spherical modes can be added to a valid view.

On the **Home** tab, in the **Add results** group, click the **RX antenna** icon.

The following quantities are available for receiving antennas:

- Active power
- Loss power
- Efficiency
- Received signal phase

Probes

The data for a voltage probe, current probe or a SPICE probe can be added to a graph.

On the **Home** tab, in the **Add results** group, click the **Probes** icon.

> **Note:** Request probe data from the cable schematic view in CADFEKO.

The following quantities are available for probes:

- Voltage
- Current

SAR (Specific Absorption Rate)

SAR does not have any special quantities or properties.

POSTFEKO can display specific absorption rate (SAR) values from near field calculations, but if spatial peak SAR of either an 1g or 10g cube is required, a SAR calculation must be requested.

The result of the SAR calculation is displayed in the result palette. For peak SAR calculations, the position is shown as a cube in the 3D view

> **Note:** The cube for peak SAR calculations is only visible if the geometry is transparent or cut away.

When viewing SAR results on a graph, the power lost or dissipated per medium is displayed in an info box in the result palette.

Related reference
SAR Standards

Characteristic Modes

A characteristic mode results can be added to a valid view or graph. The mode can either be untracked or tracked by correlating modes between frequency runs.

On the **Home** tab, in the **Add results** group, click the **Characteristic Modes Configuration** icon.

The following quantities are available for characteristic modes:

- Eigenvalue
- Modal significance
- Characteristic angle
- Modal excitation coefficient
- Modal weighting coefficient

Note: For the independent axis you can plot the results versus **Frequency**, **Mode index** or **Mode index (untracked)**.

Imports and Scripts

Imported data or data generated by a script can be added to any view on which the data is valid.

On the **Home** tab, in the **Add results** group, click the **Imports + Scripts** icon.

The quantities for imports and scripts depend entirely on the imported data or data created by the script.

Optimisation

Optimisation data such as optimised parameters, goals, global goals and masks can be viewed on a Cartesian graph after OPTFEKO was used to calculate the results.

On the **Home** tab, in the **Add results** group, click the **Optimisation** icon.

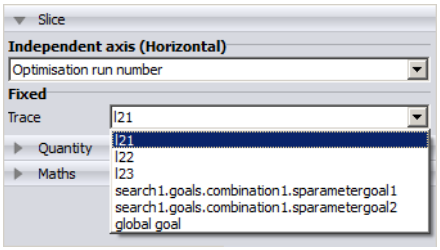


*Figure 453: Example of the result palette for optimisation. The **Independent axis (Horizontal)** is set as the **Optimisation run number**.*

Viewing the Mask on a Cartesian Graph

Display the piece-wise linear mask used to define the optimisation goal, on a Cartesian graph.

Note: Access masks from the project browser.

1. In the project browser, select the model containing the defined mask.
2. In the model browser, select the **Model** tab.
3. In the tree, expand **Optimisation**.
4. Under **Optimisation**, expand **Masks**.
5. Select a mask. From the right-click context menu, select one of the following options:
   - To add the mask to the currently selected Cartesian graph, click **Add to active window**.
   - To create a new Cartesian graph and add the mask, click **Send to new Cartesian graph**.

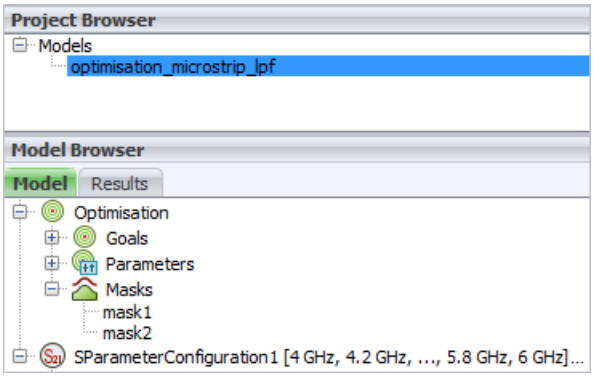


Figure 454: The project browser containing two mask definitions.

Note: Scale the mask trace to view the mask on the same axis as the goal.

On the **Trace** tab, in the **Units** group, click the **Transform axis (horizontal)** icon.

3.13 Time Domain Results

With the time analysis tool in POSTFEKO, electromagnetic scattering problems can be analysed in the time domain. The time domain results are obtained by applying an inverse fast Fourier transformation (IFFT) on the frequency domain simulation results.

3.13.1 Guidelines for Defining a Time Signal

The simulated frequency range and frequency sampling affects the time signal that can be created.

Note:

- If part of the time signal does not fall within the same frequency range as the simulation, it is possible that the windowing effect can introduce numerical artefacts in the time domain results.
- The time signal repeats due to the application of an inverse fast Fourier transformation (IFFT) on the frequency domain simulation results. Care should be taken that the repeating time signal corresponds to the desired time signal.

Follow these basic guidelines when defining a time signal:

Total Signal Duration (s_d)

For a given total signal duration of s_d, the lowest frequency to be simulated is given by:

$$f_{min} = \frac{1}{s_d} \tag{20}$$

The total signal duration should allow for the response signal to decay sufficiently before the time signal repeats.

Note: The duration of the response signal decay is model dependent and only required when the signal is not intended as a repeating time signal pulse.

Time Sampling (d_t)

The time step d_t will be given by:

$$d_t = \frac{1}{(2 \times f_{max})} \tag{21}$$

where f_{max} is the highest frequency to be simulated.

Number of Time Samples (N_t)

The number of time samples is derived from:

$$N_t = \frac{s_d}{d_t} \tag{22}$$

Number of Positive Frequency Samples

The number of frequency samples (positive) excluding zero is given by:

$$N_f = \frac{N_t}{2} \tag{23}$$

Bandwidth Calculation of Time Signals

The total power content of the time signal is calculated as follows:

$P_t = P_1 + P_2 + P_i + \ldots + P_n$

where

$P_i = (\text{SpectrumValue}(F_{(i-1)})^2 + \text{SpectrumValue}(F_{(i)})^2) * 0.5 * (F_{(i)} - F_{(i-1)})$.

The bandwidth of the time signal is then calculated as follows:

$BW = [F_{start}, F_{end}]$

where $F_{start} = F_i$ for the lowest frequency where $P_i > 10\% * P_t$

and $F_{end} = F_i$ for the lowest frequency where $P_i > 90\% * P_t$.

In summary, the model should be simulated from f_{min} to f_{max}. The POSTFEKO power bandwidth calculation as described above gives an indication of the frequency range where most of the signal power is contained.

POSTFEKO will use this power bandwidth to give a warning when there is insufficient overlap between the simulated frequency range and the power bandwidth.

Related reference
Math Functions in POSTFEKO

3.13.2 Defining the Input Time Pulse

Create a time signal to analyse frequency domain results in the time domain. A list of predefined time signals are available.

1. Obtain a frequency domain solution over the required bandwidth for the relevant requests.
2. On the **Time analysis** tab, in the **Time signal** group, click the **New time signal** icon.
3. On the **Create time signal** dialog, from the **Signal type** drop-down list, select one of the following time signals:
   - **Define pulse mathematically**
   - **Double exponential difference pulse**
   - **Double exponential piecewise pulse**
   - **Gaussian pulse (normal distribution)**
   - **Ramp**
   - **Specify points manually**

- **Triangular pulse**

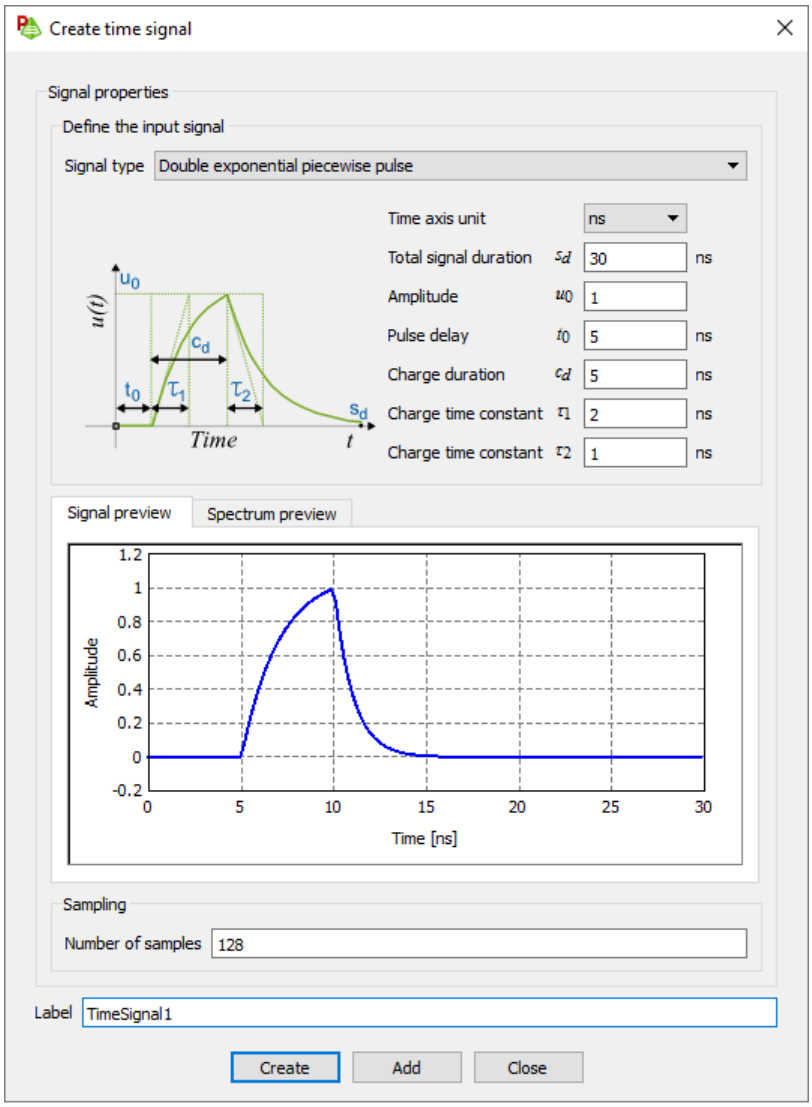


*Figure 455: The **Create time signal** dialog.*

4. Modify the time signal parameters to adjust the time signal.
5. Click **Create** to create the time signal and to close the dialog.

Define Pulse Mathematically

Define a time pulse using an analytical equation.

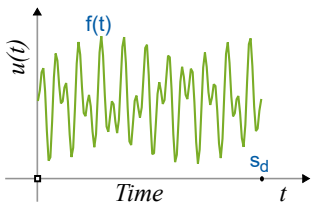


Figure 456: Define a time signal using an analytical equation.

Time axis unit	Specify the unit to be used for the time axis.
Total signal duration (s_d)	The total length of the signal in the specified units.
f(t)	Analytical equation describing the input pulse, where "t" can be used as the input time variable.
Number of samples	The number of samples taken from the signal's analytical equation.

Related concepts
Example: Define a Sine Wave Pulse
Related reference
Math Functions in POSTFEKO
Define Pulse Mathematically
Double Exponential Difference Pulse
Double Exponential Piecewise Pulse
Gaussian Pulse (Normal Distribution)
Ramp Pulse
Specify Points Manually
Triangular Pulse

Example: Define a Sine Wave Pulse

Define a sine wave pulse with a delay of 0.3 ns and a duration of 0.5 ns.

Define Step(t) = 1 for t > 0.3 ns

On the **Create time signal** dialog, in the **f(t)** field, add the following to define a step function with a delay of 0.3 ns:

$$f(t) = step(t - 300e-12) \quad (24)$$

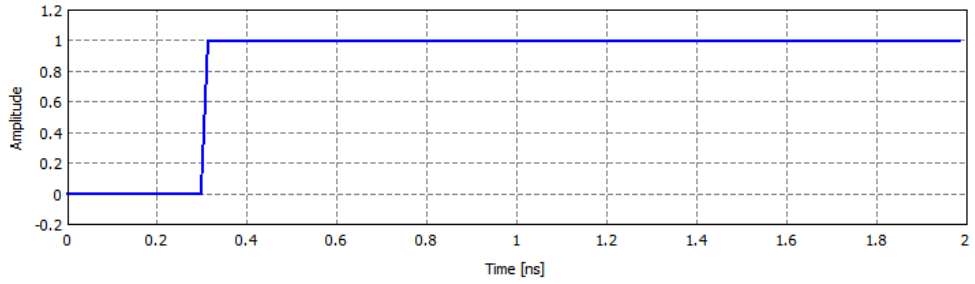


*Figure 457: The **Signal preview** shows the step function with a delay of 0.3 ns.*

Define a Rectangular Pulse

In the **f(t)** field, define a rectangular pulse with a duration of 0.5 ns by extending Equation 24 to:

$$f(t)=step(t-300e-12)-step(t-800e-12) \quad (25)$$

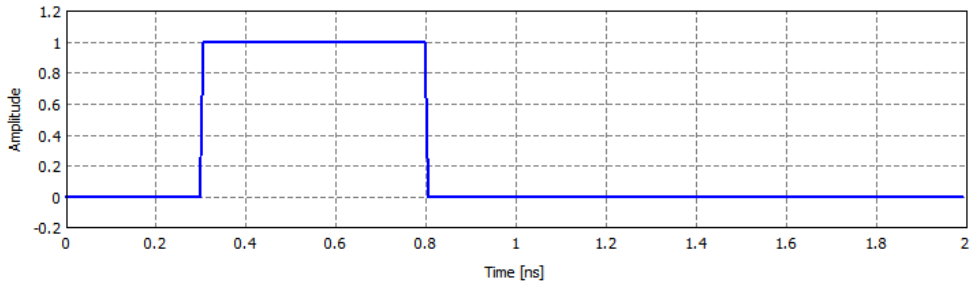


*Figure 458: The **Signal preview** shows the rectangular pulse with a delay of 0.3 ns with a duration of 0.5 ns.*

Define a Sine Wave Pulse

- In the **f(t)** field, define a sine wave pulse with a delay of 0.3 ns and a duration of 0.5 ns by extending Equation 25 to:

$$f(t)=(step(t-300e-12)-step(t-800e-12))*\sin(2*3.14*(t)*7e9) \quad (26)$$

> **Note:** Predefined variables are not supported. Use a value of 3.14 instead of *pi*.

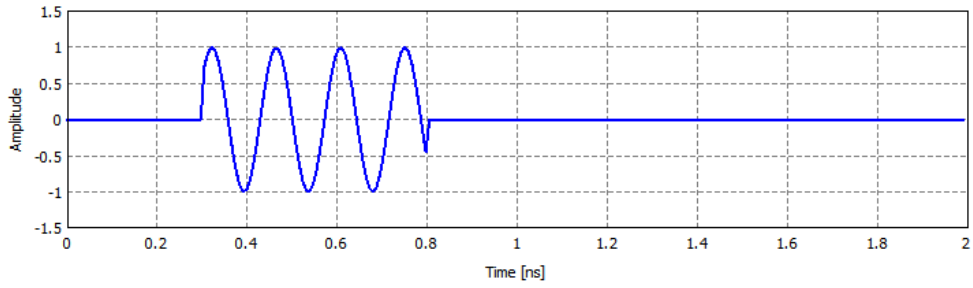


*Figure 459: The **Signal preview** shows the sine wave pulse with a delay of 0.3 ns and a duration of 0.5 ns.*

Note: Equation 26 and Figure 459 correspond to a modulated step signal at 7e9 GHz.

- Add a delay of 0.3 ns to the sine wave by extending Equation 26 to:

$$f(t)=\left(step(t-300e-12)-step(t-800e-12)\right)*\sin(2*3.14*(t-300e-12)*7e9) \quad (27)$$

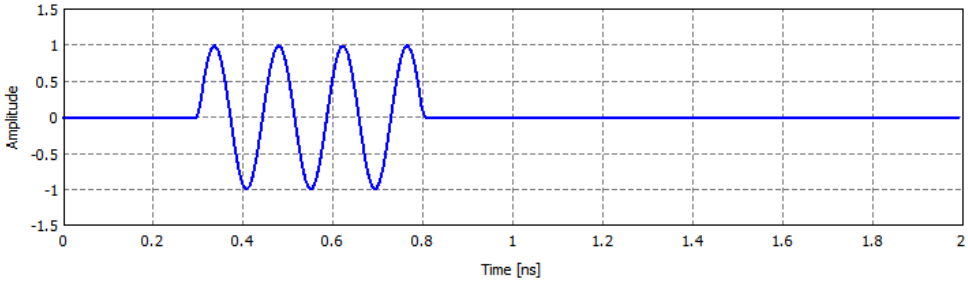


*Figure 460: The **Signal preview** shows the final sine wave pulse.*

Double Exponential Piecewise Pulse

Define a double exponential piecewise time pulse.

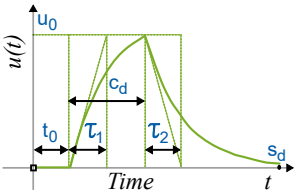


Figure 461: Define a double exponential difference time pulse.

Time axis unit
: Specify the unit to be used for the time axis.

Total signal duration (s_d)
: The total length of the signal in the specified units.

Amplitude (u_0)
: The amplitude of the time signal.

Pulse delay (t_0)
: The pulse delay is the time until the peak of the time signal envelope.

Charge duration (c_d)
: The charge duration is the time from the pulse delay has ended until the signal begins to discharge.

Charge time constant (τ_2)
: The time that would be required to discharge the signal down to 36.8% of its full potential (u_0).

Charge time constant (τ_1)
The time that would be required to charge the signal up to 63.2% of its full potential (u_0).

Number of samples
The number of samples taken from the signal's analytical equation.

$$u(t)=\begin{cases} 0 & \text{for} \quad t\le t_0 \\ u_1\left(1-e^{-\frac{t-t_0}{\tau_1}}\right) & \text{for} \quad t_0\le t\le c_d+t_0 \\ u_2 e^{-\frac{t-t_0}{\tau_2}} & \text{for} \quad t\ge c_d+t_0 \end{cases} \tag{28}$$

$$u_1=\frac{u_0}{1-e^{-\frac{c_d}{\tau_1}}} \qquad u_2=\frac{u_0}{e^{-\frac{c_d}{\tau_2}}} \tag{29}$$

The Fourier transform is as follows:

$$U(f)=\begin{cases} \left[u_1\left(\frac{1-e^{-j2\pi f c_d}}{j2\pi f}+\frac{\tau_1(e^{-c_d(j2\pi f+\frac{1}{\tau_1})}-1)}{1+j2\pi f\tau_1}\right)\cdots \right. \\ \left. \quad +\frac{u_2\tau_2}{(1+j2\pi f\tau_2)}e^{-c_d(j2\pi f+\frac{1}{\tau_2})}\right]e^{-j2\pi f t_0} \quad \text{for } f>0 \\ u_1(\tau_1(e^{\frac{-c_d}{\tau_1}}-1)+c_d)+u_2\tau_2 e^{\frac{-c_d}{\tau_2}} \quad \text{for } f=0 \end{cases} \tag{30}$$

Double Exponential Difference Pulse

Define a double exponential difference time pulse.

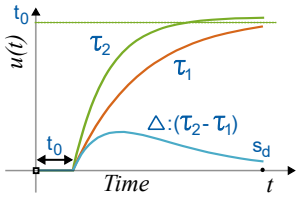


Figure 462: Define a double exponential difference time pulse.

Time axis unit
Specify the unit to be used for the time axis.

Total signal duration (s_d)
The total length of the signal in the specified units.

Amplitude (u_0)
The amplitude of the time signal.

Pulse delay (t_0)
The pulse delay is the time until the peak of the time signal envelope.

Time constant (τ_1)
The pulse is defined as the difference of two exponentially charging pulses. The value of τ_1 describes the time that would be required for the subtracted signal to reach 63.2% of its full potential (u_0).

Time constant (τ_2)
The value of τ_2 describes the time that would be required for the base signal to reach 63.2% of its full potential (u_0).

Number of samples
The number of samples taken from the signal's analytical equation.

$$u(t)=\begin{cases} 0 & \text{for} \quad t \le t_0 \\ u_0\left(e^{-\frac{t-t_0}{\tau_1}}-e^{-\frac{t-t_0}{\tau_2}}\right) & \text{for} \quad t > t_0 \end{cases} \tag{31}$$

The Fourier transform is as follows:

$$U(f)=u_0\left(\frac{\tau_1}{1+j2\pi f\tau_1}-\frac{\tau_2}{1+j2\pi f\tau_2}\right)e^{-j2\pi f t_0} \tag{32}$$

Gaussian Pulse (Normal Distribution)

Define a Gaussian time pulse with a normal distribution.

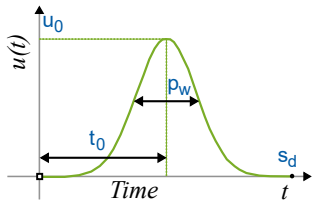


Figure 463: Define a Gaussian time pulse with a normal distribution.

Time axis unit
Specify the unit to be used for the time axis.

Total signal duration (s_d)
The total length of the signal in the specified units.

Amplitude (u_0)
The amplitude of the time signal.

Pulse delay (t_0)
The pulse delay is the time until the peak of the time signal envelope.

Pulse width (p_w)
This is the half-amplitude pulse width of the signal. The pulse width is the total length of time that the signal is above 50% of its peak value (u_0).

Number of samples
The number of samples taken from the signal's analytical equation.

$$u(t)=\begin{cases} 0 & \text{for} \quad t \le t_0 \\ u_0\left(e^{-\frac{t-t_0}{\tau_1}}-e^{-\frac{t-t_0}{\tau_2}}\right) & \text{for} \quad t > t_0 \end{cases} \tag{33}$$

The Fourier transform is as follows:

$$U(f)=u_0\left(\frac{\tau_1}{1+j2\pi f\tau_1}-\frac{\tau_2}{1+j2\pi f\tau_2}\right)e^{-j2\pi f t_0} \tag{34}$$

Ramp Pulse

Define a ramp time pulse.

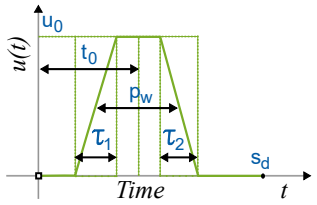


Figure 464: Define a ramp time pulse.

Time axis unit
Specify the unit to be used for the time axis.

Total signal duration (s_d)
The total length of the signal in the specified units.

Amplitude (u_0)
The amplitude of the time signal.

Pulse delay (t_0)
The pulse delay is the time until the peak of the time signal envelope.

Pulse width (p_w)
This is the half-amplitude pulse width of the signal. The pulse width is the total length of time that the signal is above 50% of its peak value (u_0).

Rise time (τ_1)
The time required for the pulse to reach its peak value (u_0) from rest.

Fall time (τ_2)
The time required for the pulse to reach the rest value from its peak (u_0).

Note: The discharge time will be determined by the pulse width (p_w).

Number of samples

The number of samples taken from the signal's analytical equation.

$$u(t)=\begin{cases} 0 & \text{for} \quad t\le t_1+t_0 \\ u_0\left(1-\frac{|t-t_2-t_0|}{\tau_1}\right) & \text{for} \quad t_1+t_0\le t\le t_2+t_0 \\ u_0 & \text{for} \quad t_2+t_0\le t\le t_3+t_0 \\ u_0\left(1-\frac{|t-t_3-t_0|}{\tau_2}\right) & \text{for} \quad t_3+t_0\le t\le t_4+t_0 \\ 0 & \text{for} \quad t\ge t_4+t_0 \end{cases} \tag{35}$$

$$t_1=-\frac{p_w+\tau_1}{2} \qquad t_2=-\frac{p_w-\tau_1}{2} \qquad t_3=\frac{p_w-\tau_2}{2} \qquad t_4=\frac{p_w+\tau_2}{2} \tag{36}$$

The Fourier transform is as follows:

$$U(f)=\begin{cases} u_0\left[\left((p_w-\frac{\tau_1+\tau_2}{2})\text{sinc}\left((p_w-\frac{\tau_1+\tau_2}{2})f\right)\right)\ldots\right. \\ \quad +\frac{1}{\tau_1(2\pi f)^2}\left(e^{-j2\pi f\tau_1}(1+j2\pi f\tau_1)-1\right)e^{+j2\pi f(\frac{p_w}{2}+\tau_1)}\ldots \\ \left.+\frac{1}{\tau_2(2\pi f)^2}\left(e^{+j2\pi f\tau_2}(1-j2\pi f\tau_2)-1\right)e^{-j2\pi f(\frac{p_w}{2}+\tau_2)}\right]e^{-j2\pi f t_0} & \text{for } f>0 \\ u_0 p_w & \text{for } f=0 \end{cases} \tag{37}$$

Triangular Pulse

Define a triangular time pulse.

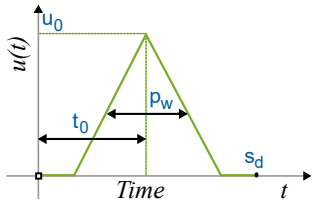


Figure 465: Define a triangular time pulse.

Time axis unit

Specify the unit to be used for the time axis.

Total signal duration (s_d)

The total length of the signal in the specified units.

Amplitude (u_0)

The amplitude of the time signal.

Pulse delay (t_0)

The pulse delay is the time until the peak of the time signal envelope.

Pulse width (p_w)
: This is the half-amplitude pulse width of the signal. The pulse width is the total length of time that the signal is above 50% of its peak value (u_0).

Rise time (τ_1)
: The time required for the pulse to reach its peak value (u_0) from rest.

Fall time (τ_2)
: The time required for the pulse to reach the rest value from its peak (u_0).

> **Note:** The discharge time will be determined by the pulse width (p_w).

Number of samples
: The number of samples taken from the signal's analytical equation.

$$u(t)=\begin{cases} u_0\left(1-\dfrac{|t-t_0|}{p_w}\right) & \text{for} \quad |t-t_0|\le p_w \\ 0 & \text{for} \quad \text{otherwise} \end{cases} \tag{38}$$

The Fourier transform is as follows:

$$U(f)=2u_0p_w\text{sinc}^2(2fp_w)e^{-j2\pi ft_0} \tag{39}$$

Specify Points Manually

Define a time pulse by specifying the points manually.

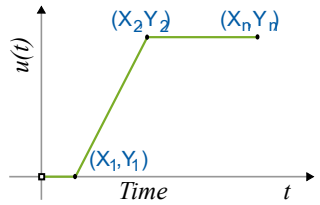


Figure 466: Define a time signal by specifying the points.

Time axis unit
: Specify the unit to be used for the time axis.

Scale time axis
: A scale factor applied to the time axis values.

Scale amplitude
: A scale factor applied to the amplitude axis values.

[Time, Amplitude]
: Specify the Time (X) and Amplitude (Y) coordinates of the time signal. The pulse will be resampled using number of specified samples, where linear interpolation between the defined points will be used.

> **Note:** The list of points can be imported from any comma separated value file.

Number of samples
The number of samples taken from the signal's analytical equation.

3.13.3 Adding Time Domain Results to a View

Add the time results to a 3D view or graph.

You must have already solved the model over the required frequency range and defined a suitable time signal.

1. Select the graph or 3D view to which you want to add time results.
2. On the **Time analysis** tab, in the **Add time domain results** group, click the relevant request type. From the drop-down list select the request.

3.13.4 Time Domain Results

The following time domain results can be added to a valid 3D view or graph.

Table 47: The time result types that can be added to a graph or 3D view.

Icon	Icon text	Description
	Time signal	Adds the defined time signal to a graph.
	Far field	Adds a far field time analysis result to a graph or 3D view.
	Near field	Adds a near field time analysis result to a graph or 3D view.
	Sources	Adds a source time analysis result to a graph.
	Sparameters	Adds an S-parameter time result to a graph.
	Loads	Adds a load time analysis result to a graph.
	Networks	Adds a networks time analysis result to a graph.
	Currents	Adds a currents time analysis result to a 3D view or graph.
	SPICE probes	Adds a SPICE probe time analysis result to a graph.

3.13.5 Spectral Extrapolation Techniques

When performing a time analysis where lower frequencies are not simulated and need to be estimated, different options are available to extrapolate the spectral component of the simulation result to 0 Hz.

On the **Time analysis** tab, in the **Add time domain results** group, click the dialog launcher.

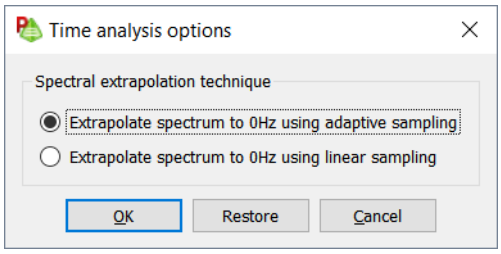


*Figure 467: The **Time analysis options** dialog.*

Note: The adaptive sampling technique provides more accurate low frequency extrapolation than linear interpolation but can be less predictable.

3.14 Animation

Use animation to obtain a better understanding of results or export the animation to use in a presentation or report.

Animate the following properties:

- phase (requires a result)
- frequency (requires a result)
- camera angle (requires geometry)
  - phi
  - theta
  - theta and phi

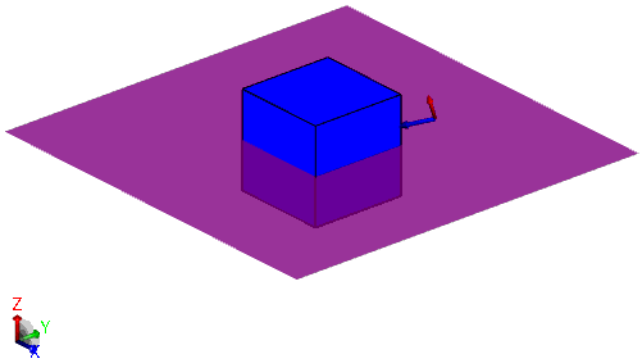


Figure 468: An example of animation (not supported in PDF User Guide).

3.14.1 Animating a Result

Gain insight into a result by animating a result.

For this example, the result is animated over time step. A time signal and time result are required to animate over time.

1. Select the 3D view and the result that you want to animate.
2. On the **3D View** contextual tabs set, on the **Animate** tab, on the **Settings** group, click the **Type** icon. From the drop-down list, select the **Time step** icon.
3. On the **3D View** contextual tabs set, on the **Animate** tab, on the **Control** group, click the **Play** icon.

> **Note:** To stop the animation, click the **Play** icon.

3.14.2 Exporting an Animation

Export an animation of a model to use in a presentation or report.

1. On the **3D View** contextual tabs set, on the **Animate** tab, on the **Animation** group, click the **Export animation** icon.

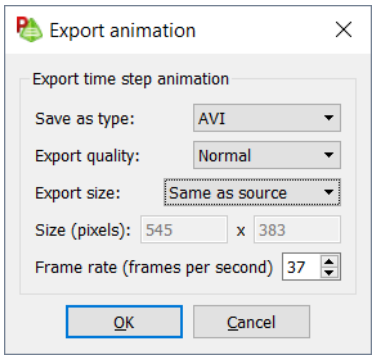


*Figure 469: The **Export animation** dialog.*

2. From the **Save as type** drop-down list, select one of the following:
   - **AVI**
   - **MOV**
   - **GIF**
   - **MKV**
3. From the **Export quality** drop-down list, select one of the following:
   - **High**
   - **Normal**
   - **Low**

   Setting the quality affects the compression ratio for the specified screen size. For very high-quality exports, it is good practice to reduce the screen size to as small as is need and setting the **Export quality** to **High**.
4. From the **Export size** drop-down list, select one of the following:
   - **Same as source**
   - **QQVGA (160x120)**
   - **QVGA (320x240)**
   - **VGA (640x480)**
   - **SVGA (800x600)**
   - **XGA (1024x768)**
   - **SXGA (1280x1024)**
   - **Custom**
5. In the **Frame rate (frames per second)** field, specify the frame rate. Setting the frame rate affects how "smooth" the animation appears.
6. Click **OK**.

The **Animation export file name** dialog is displayed.

7. In the **File name** field, specify the name of the exported animation file.
8. In the **Save as type**, specify the file type of the exported animation file.
9. Click **Save** to export the animation to file and to close the dialog.

3.14.3 Animation Controls and Settings

View the controls available to control the animation.

On the **3D View** contextual tabs set, on the **Animate** tab, on the **Control** group, click the **Play** icon.

Table 48: Animation controls and settings.

Icon	Icon text	Description
	Play	Start / stop the animation.
	Faster	Increases the speed of the animation - more changes per second of viewing.
	Slower	Decreases the speed of the animation - less changes per second of viewing.
	Type	Specify the animation type: frequency, phase or camera angle.
	Legend	Show / hide the display the animation legend.
	Settings	Advanced animation settings.
	Frequency	Animation over frequency.
	Phase	Animation over phase.
	Time step	Animation over time step. A time step animation requires a time signal and a time domain result added to the 3D view.
	Phi Rotate	Animation over camera angle phi.
	Theta rotate	Animation over camera angle theta.
	Theta and Phi rotate	Animation over camera angle theta and phi

3.14.4 Advanced Animation Settings

Specify the speed and resolution with which a variable animates when animating a property.

On the **3D View** contextual tabs set, on the **Animate** tab, on the **Settings** group, click the **Animation settings** icon.

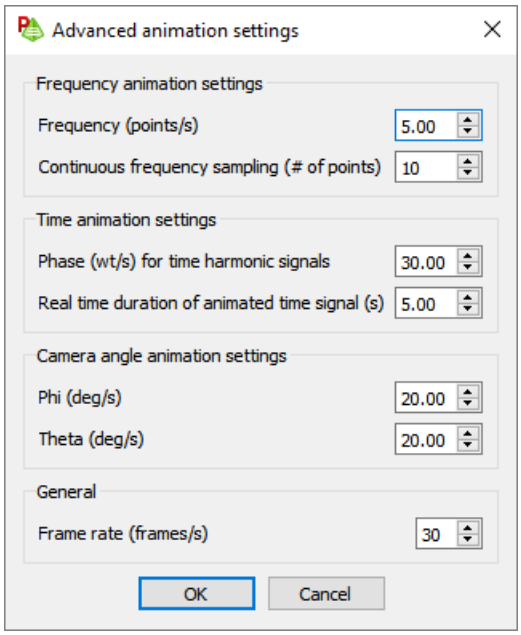


*Figure 470: The **Advanced animation settings** dialog.*

Frequency Animation Settings

Frequency (points/s)
Specify the animation speed.

Continuous frequency sampling (# of points)
For continuous frequency models, the frequency range is broken into a number of discrete steps, thereby specifying the sampling resolution.

Time Animation Settings

Phase (wt/s) for time harmonic signals
Specify the phase increment per second.

For example, setting phase (wt/s) = 30° will result in the phase incrementing by 30° each second and a complete 360° loop in 12 seconds.

Real time duration of animated time signal(s)
Specify the time duration of the animation before it starts to loop.

Camera Angle Animation Settings

Phi (deg/s)
Specify the phi angle for the camera angle during animation.

Theta (deg/s)
Specify the theta angle for the camera angle during animation.

General Settings

Frame rate (frames/s)
Specify the rate at which the consecutive images are displayed.

3.15 Generating Reports

POSTFEKO is a useful tool to help analyse and present data in a useful format. It is often required to use the processed results in a report or presentation. To help make it easier to generate these reports, several tools are available in POSTFEKO.

Note: For Microsoft PowerPoint and Microsoft Word, you need to have Microsoft Office 2003 or later installed.

3.15.1 Exporting an Image

Export an image of the active view to file.

1. On the **Reporting** tab, in the **Export images** group, click the **Export image** icon.

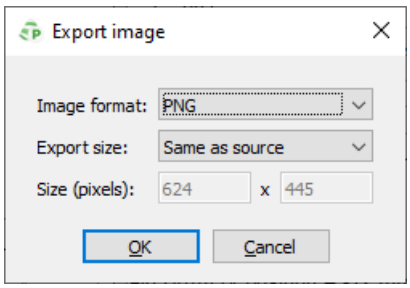


*Figure 471: The **Export image** dialog.*

2. Select a view to export.
3. From the **Image format** drop-down list, select one of the following:
   - **PNG**
   - **BMP**
   - **CUR**
   - **ICNS**
   - **JPG**
   - **PBM**
   - **PGM**
   - **TIF**
   - **WBMP**
   - **WEBP**
   - **PDF**
   - **EPS**
   - **EMF**
4. From the **Export size** drop-down list, select one of the following:

- **Same as source**
- **QQVGA (160x120)**
- **QVGA (320x240)**
- **VGA (640x480)**
- **SVGA (800x600)**
- **XGA (1024x768)**
- **SXGA (1280x1024)**
- **Custom**

5. Click **OK**.
   The **Image export file name** dialog is displayed.
6. In the **File name** field, specify the file name of the exported file.
7. In the **Save as type**, specify the file type of the exported file.
8. Click **Save** to export the active view to file and to close the dialogs.

3.15.2 Generating a Quick Report

Generate a report with minimal effort using selected images and headers from a POSTFEKO session using a predefined report template.

1. On the **Reporting** tab, in the **Reports** group, click the **Generate quick report** icon.

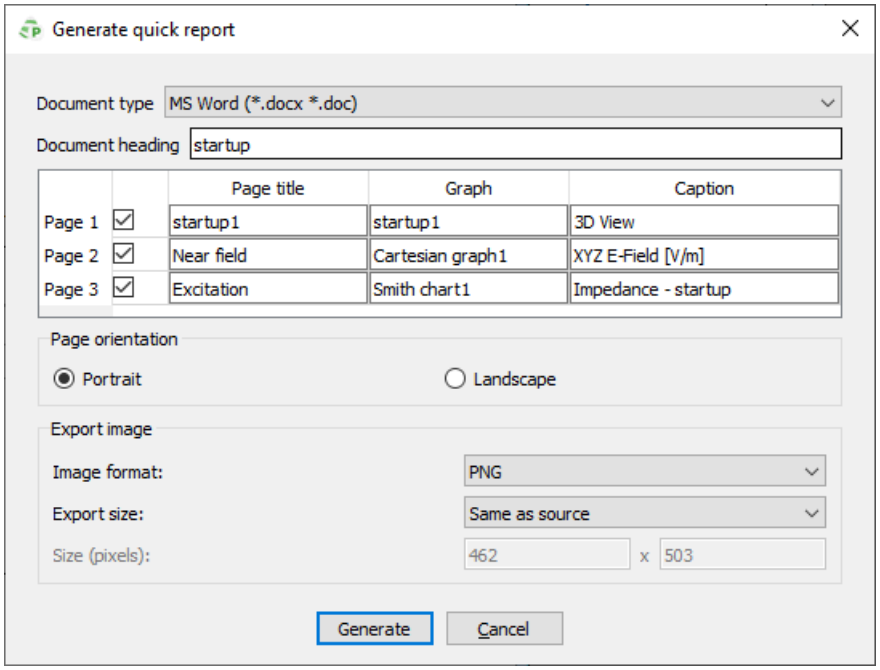


*Figure 472: The **Generate quick report** dialog.*

2. From the **Document type** drop-down list, select one of the following:
   - **MS PowerPoint (*.pptx)**

- **MS Word (*.docx *.doc)**
- **PDF (*.pdf)**

3. In the **Document heading** field, specify the report title.
4. In the table, specify the page titles, graphs to include and the graph captions.
5. For Microsoft Word and PDF reports, specify the **Page orientation**.
6. From the **Image format** drop-down list, select one of the following:
   - **PNG**
   - **BMP**
   - **CUR**
   - **ICNS**
   - **JPG**
   - **PBM**
   - **PGM**
   - **TIF**
   - **WBMP**
   - **WEBP**
   - **PDF**
   - **EPS**
   - **EMF**
7. From the **Export size** drop-down list, select one of the following:
   - **Same as source**
   - **QQVGA (160x120)**
   - **QVGA (320x240)**
   - **VGA (640x480)**
   - **SVGA (800x600)**
   - **XGA (1024x768)**
   - **SXGA (1280x1024)**
   - **Custom**
8. Click **Generate** to generate the report.

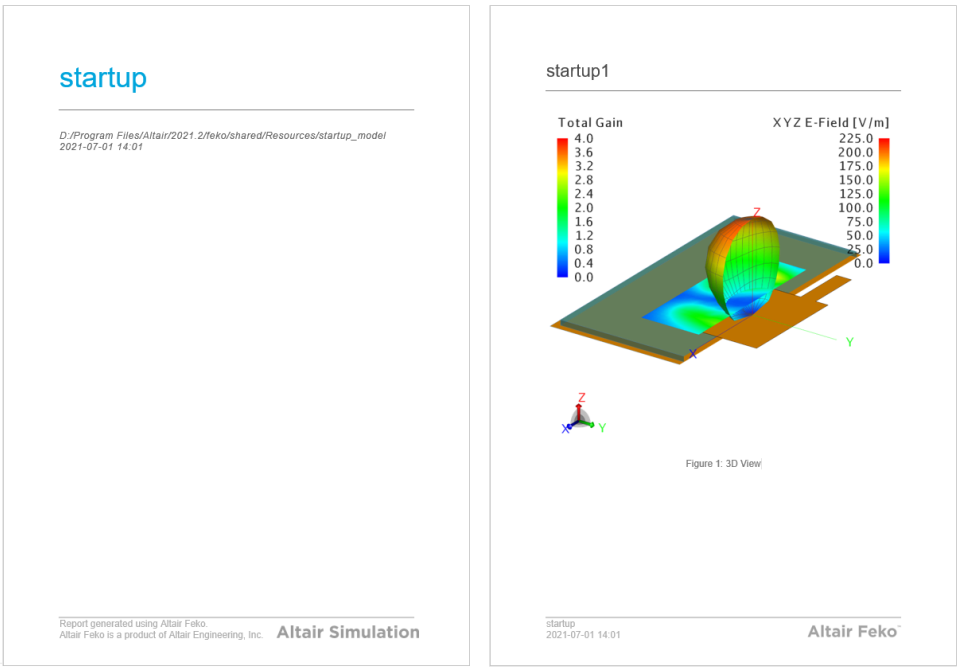


Figure 473: Example showing the Feko template for the quick report (Microsoft Word document).

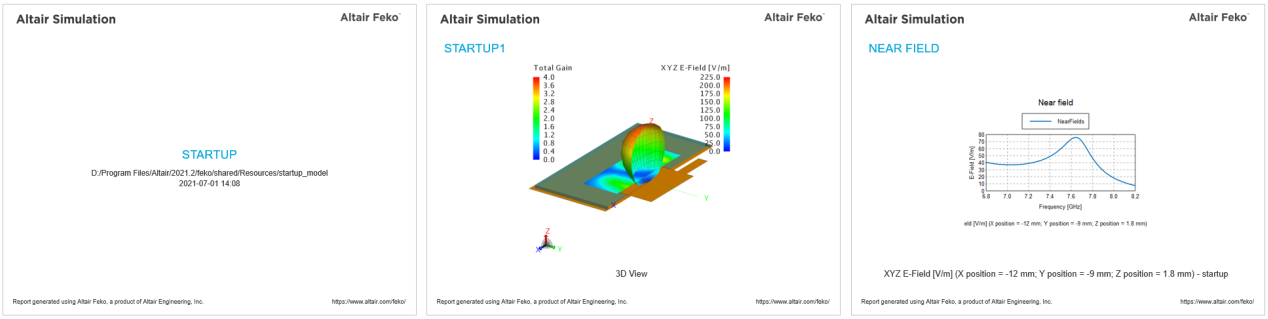


Figure 474: Example showing the Feko template for the quick report (Microsoft PowerPoint document).

3.15.3 Defining a Microsoft Template

Decide on a Microsoft template to define the theme, company logo and branding to use when creating a POSTFEKO report template.

These styled templates can be obtained from Microsoft or you can create a template with a specific theme, company logo and branding.

- For Microsoft Office 2010 and onward, use “content controls”.

  Note: “Content controls” only applicable to Microsoft Word.

- For Microsoft Office 2007 and older, use “rectangular shape placeholders”.

Defining a Microsoft Word Template Using Content Controls

Create a report template in Microsoft Word that uses content controls to create structured content that can be reused each time you generate a report.

1. Create a Microsoft Word (`.dotx`) template using one of the following workflows:
   - Use one of the predefined templates provided by Microsoft.
   - Create a template with the required styling.

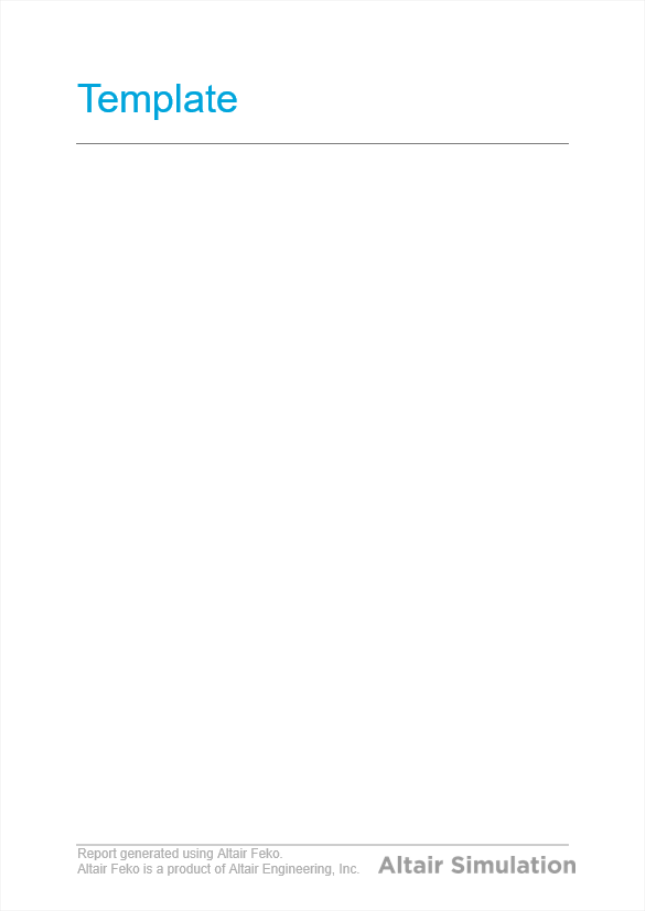


Figure 475: Example of a Microsoft Word template (`.dotx` file) with styling.

2. In POSTFEKO, decide on the graphs and 3D views to be added to the report.

   For this example, the `startup` model is used. The required views are the 3D view and the graphs are:
   - *startup1*
   - *Cartesian graph1*
   - *Smith chart1*

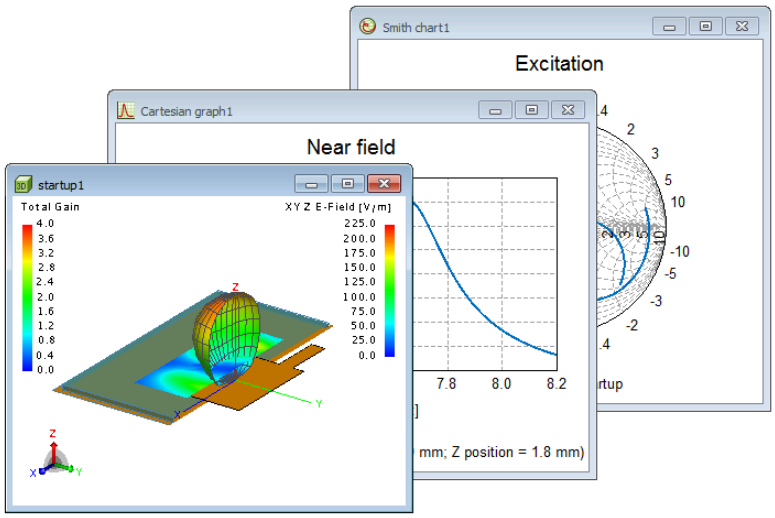


Figure 476: The startup model with the 3D view (startup1), Cartesian graph (Cartesian graph1) and Smith chart (Smith chart1) which will be required for the report.

3. In Microsoft Word, activate the **Developer** tab.

a) On the application menu, click **Options** > **Customize Ribbon** and select the *Developer* check box.

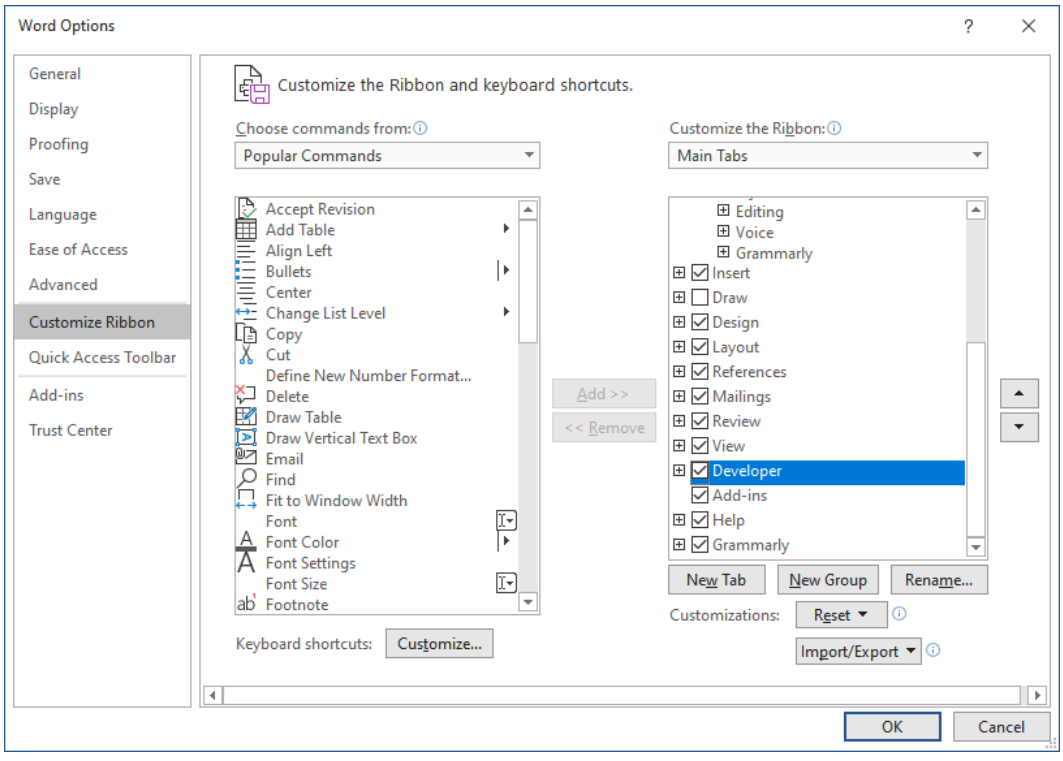


*Figure 477: The **Word Options** dialog in Microsoft Word. Select the **Developer** check box to enable the **Developer** tab in Microsoft Word.*

4. Add content controls to the Microsoft Word template.

a) In Microsoft Word, on the ribbon click the **Developer** tab.

b) Add a **Picture Content Control** (**Controls** group) to the template at each location in the template where a graph or 3D view is to be added.

5. Enable **Design Mode** in Microsoft Word.

On the **Developer** tab, in the **Controls** group, click the **Design Mode** icon.

6. Add tags to the Microsoft Word template.

a) For each content control, on the **Developer** tab, in the **Controls** group, click **Properties**.

b) On the **Content Control Properties** dialog, add the tag that links to a specific POSTFEKO graph.
For this example, the tag is *TagFor3dView*.

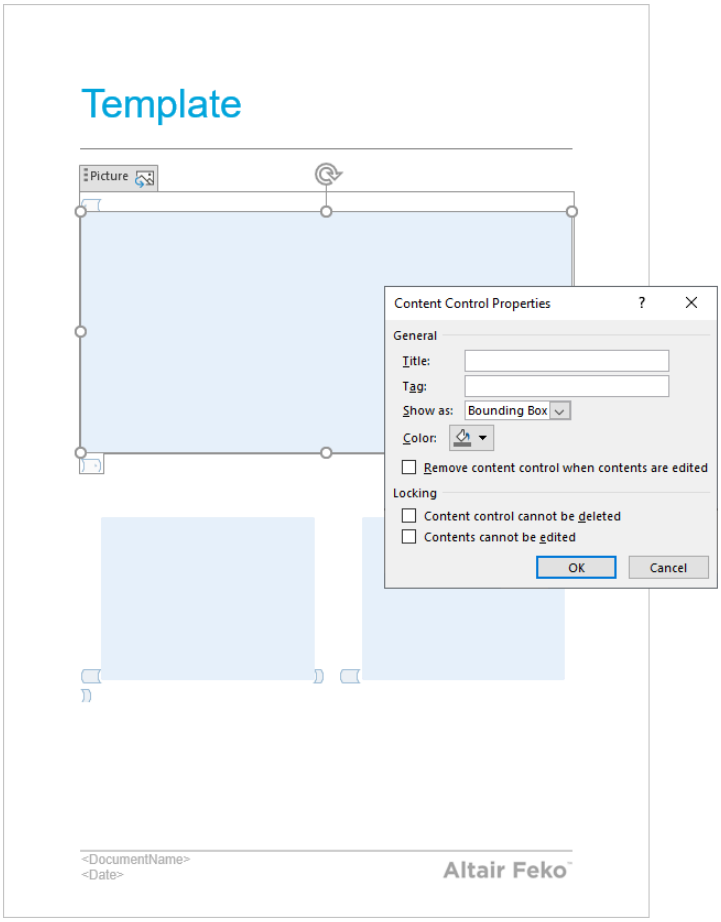


*Figure 478: The **Content Control Properties** dialog in Microsoft Word where the tag for the POSTFEKO graph is specified.*

7. Save the Microsoft Word (`.dotx`) template.

Defining a Microsoft Template Using Rectangular Placeholders

Create a report template in Microsoft PowerPoint or Microsoft Word that uses rectangular placeholders to create structured content that can be reused each time you generate a report.

1. Create a Microsoft PowerPoint (`.potx`) or Microsoft Word (`.dotx`) template using one of the following workflows:
   - Use one of the predefined templates provided by Microsoft.
   - Create a template with the required styling.
2. In POSTFEKO, decide on the graphs and 3D views to be added to the report. For this example, the `startup` model will be used. The required views are the 3D view and the graphs are *startup1*, *Cartesian graph1* and *Smith chart1*.
3. In Microsoft Word or Microsoft PowerPoint add "placeholders" to the template.
   a) Add a rectangle (**Shapes**) to the template for each required graph. It acts as a placeholder for the graph.
   b) Select a placeholder and from its right-click context menu select **Add text**.
   c) Add the text `<feko:image:tag>`, where tag is a unique label linking to a specific graph or 3D view.

      For this example, the `startup` model is used. The required views are the 3D view and the graphs are:
      - *startup1*
      - *Cartesian graph1*
      - *Smith chart1*

   d) [Optional] Add text descriptions and title for the graphs and 3D views.

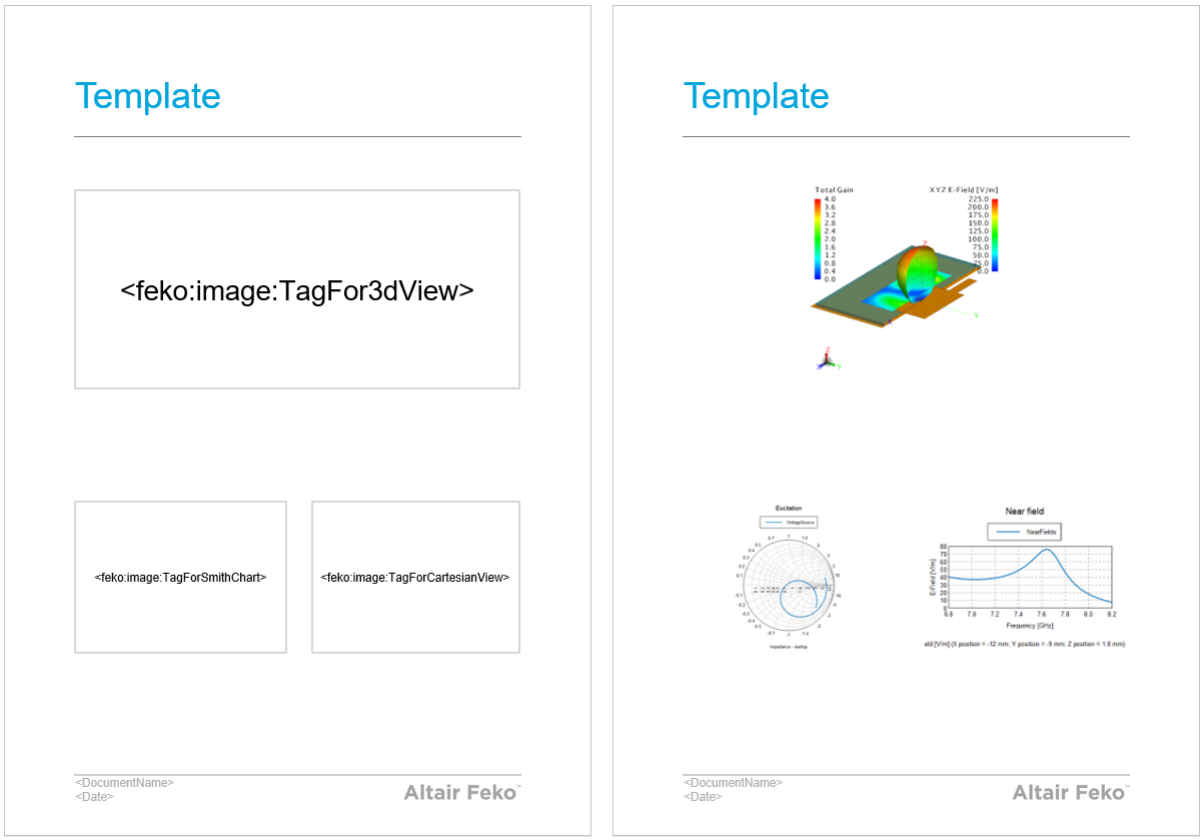


Figure 479: The template with "rectangular placeholders" at the positions in the template where the 3D view and graphs will be required and (b) the report that will be generated when using this template.

4. Save the Microsoft PowerPoint (`.potx`) or Microsoft Word (`.dotx`) template.

3.15.4 Defining a POSTFEKO Report Template

Create a POSTFEKO report template when it is required to create consistent reports. The reports are generated from a preconfigured POSTFEKO report template using styling from aMicrosoft PowerPoint or a Microsoft Word template.

1. On the **Reporting** tab, in the **Reports** group, click the **Define template** icon.

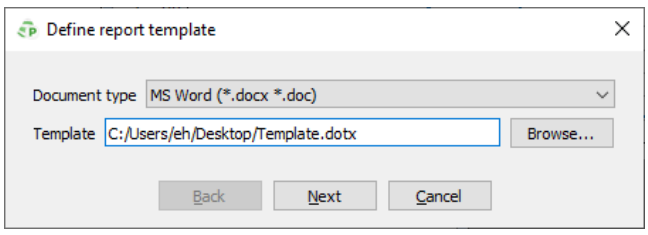


*Figure 480: The **Define report template** dialog.*

2. In the **Document type** drop-down list, select one of the following:

- **MS Word (*.docx *.doc)**
- **MS PowerPoint (*.pptx)**

3. In the **Template** field, specify the Microsoft template to be used for the report generation.
4. Click **Next**.

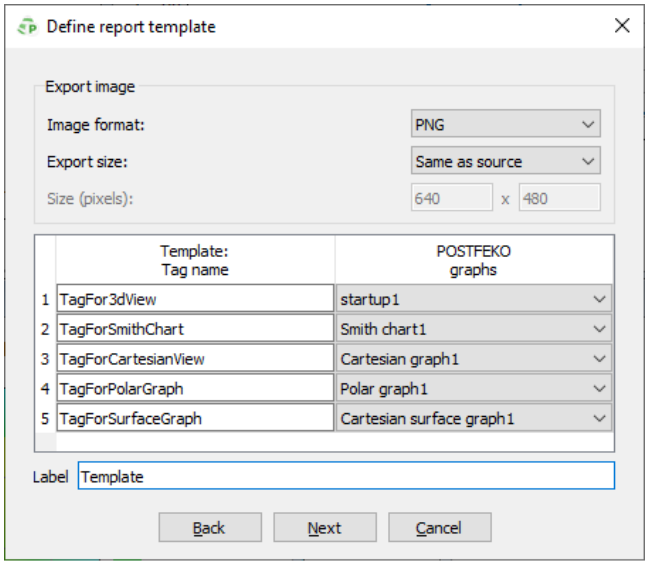


*Figure 481: The **Define report template** dialog.*

5. In the **Image format** drop-down list specify the image format.
6. In the **Export size** drop-down list select the export size of the images used in the report.
7. In the table specify the POSTFEKO graph or 3D view for each tag used in the Microsoft template.
8. Click **Next** > **Done**.

> **Tip:** View the report template in the project browser under **Report templates**.

9. [Optional] Modify the report template.
   a) On the **Reporting** tab, in the **Reports** group, click the **Modify template** icon.

3.15.5 Generating a Report From a POSTFEKO Report Template

After a report template was defined, create a report using the report template.

1. On the **Reporting** tab, in the **Reports** group, click the **Generate quick report** icon.
2. From the drop-down list, select one of the following workflows:
   - Select an existing defined report template from which to create the report.
   - To set up a report template, select **Define report template**.

3.15.6 Exporting a Report Template for Reuse

A report template can be exported to XML format for reuse in another POSTFEKO session.

1. On the **Reporting** tab, in the **Reports** group, click the **Import / Export template** icon.
   From the drop-down list select the **Export report template (*.xml)** icon.

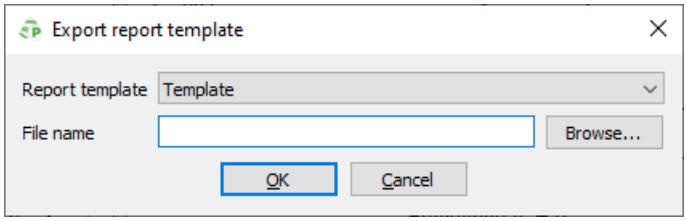


*Figure 482: The **Export report template** dialog.*

2. From the **Report template** drop-down list, select a defined report template.
3. In the **File name** field enter a file name to be used for the exported template.
4. Click **OK** to export the report template and to close the dialog.

3.15.7 Importing a Report Template for Reuse

A report template can be imported from an XML file to reuse in the current POSTFEKO session.

1. On the **Reporting** tab, in the **Reports** group, click the **Import / Export template** icon.
   From the drop-down list select the **Import report template (*.xml)** icon.

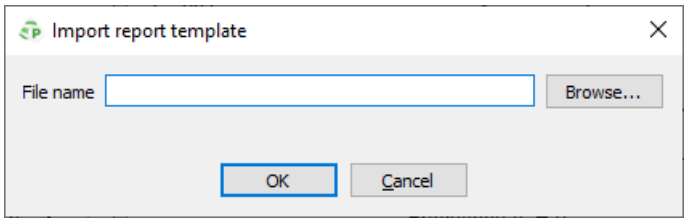


*Figure 483: The **Import report template** dialog.*

2. In the **File name** field, browse for the XML file to import.
3. Click **OK** to import the XML file and to close the dialog.

3.15.8 Using LuaCOM to Control Microsoft Word and Microsoft Excel

Use a Lua script to generate Microsoft Word or Microsoft Excel documents with specified content without having to open the applications.

Ensure that you are using the Windows operating system and that Microsoft Word and/or Microsoft Excel is installed on the machine.

1. Open the script editor.
2. Create a new empty script.
3. As an example, load one of the scripts below into the script editor.
4. Run the script.

```
-- MS WORD
require "luacom"

-- Open Word
local msword = luacom.CreateObject("Word.Application")
assert(msword, "Could not open MS Word")

-- Initialise the document
msword.Visible = true
doc = msword.Documents:Add()

-- Add content
insertionPoint = doc.ActiveWindow.Selection
insertionPoint.Style = "Heading 1"
insertionPoint:TypeText( "Feko Says..." )
insertionPoint:TypeParagraph()
insertionPoint:TypeText( "Hello world!" )
```

```
-- MS EXCEL
require "luacom"

-- Open Excel
local excel = luacom.CreateObject("Excel.Application")
assert(excel, "Could not open MS Excel")

-- Initialise the worksheet
excel.Visible = true
workbook = excel.Workbooks:Add()
worksheet = workbook.Worksheets:Add()

-- Populate the data and display the contents of cell A3
worksheet.Range( "A1", "A1" ).Value2 = [[hello]]
worksheet.Range( "A2", "A2" ).Value2 = [[world]]
worksheet.Range( "A3", "A3" ).Value2 = [[=CONCAT(A1," ",A2,"!!")]]
feko.Form.Info( "Excel says...", worksheet.Range( "A3", "A3" ).Value2 )

-- Change an input value and display A3 once again
worksheet.Range( "A2", "A2" ).Value2 = [[everybody]]
feko.Form.Info( "Excel says...", worksheet.Range( "A3", "A3" ).Value2 )
```

3.16 Lua Scripting

Feko provides a powerful scripting language that allows you to create scripts that control CADFEKO and POSTFEKO.

3.16.1 Script Editor

The script editor allows you to create scripts based on the Lua language to control CADFEKO, POSTFEKO and other applications as well as manipulation of data to be viewed and analysed further in POSTFEKO.

On the **Home** tab, in the **Scripting** group, click the **Script editor** icon.

The script editor includes the following IDE (integrated development environment) features:

1. Syntax highlighting.
2. Intelligent code completion.
3. Indentation for blocks to convey program structure, for example, loops and decision blocks in scripts.
4. Use of breakpoints and stepping in scripts to debug code or control its execution.
5. An active console to query variables or execute simple commands.

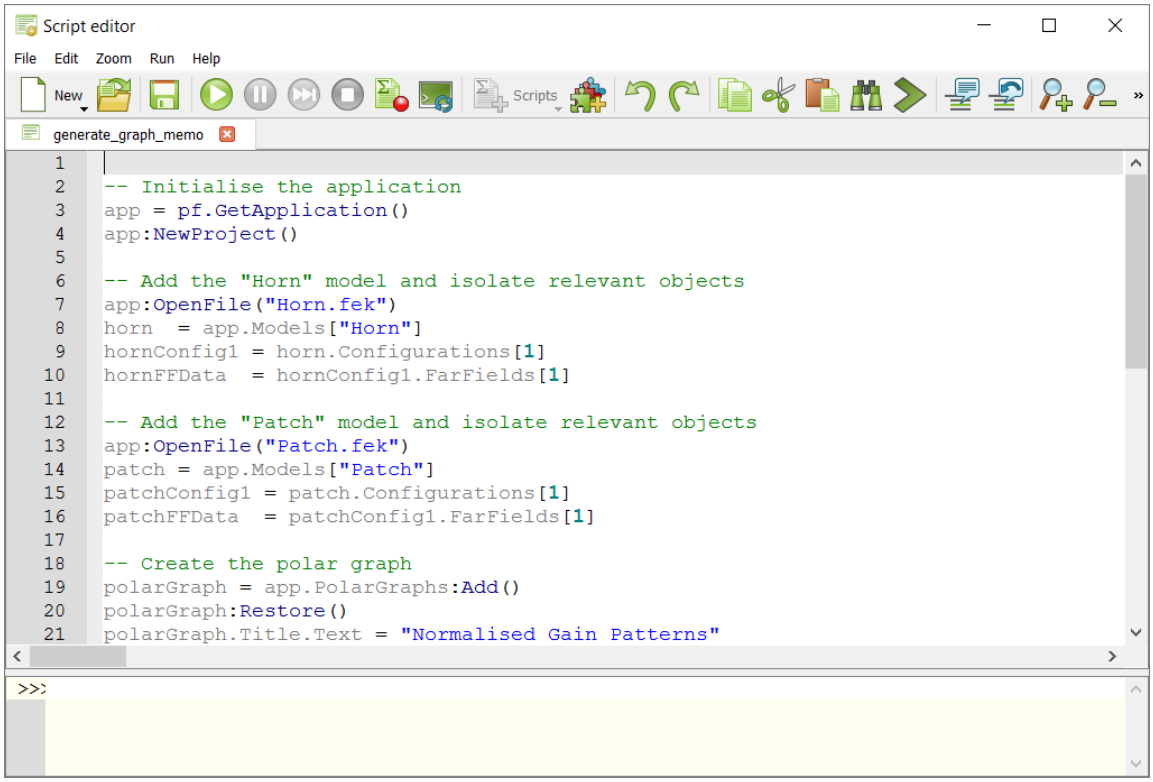


Figure 484: The script editor in POSTFEKO.

3.16.2 Application Macros

An application macro is a reference to an automation script, an icon file and associated metadata.

Application macros are available directly or can be added, removed, modified or executed from the application macro library.

> **Tip:** A large collection of application macros are available in CADFEKO and POSTFEKO.

On the **Home** tab, in the **Scripting** group, click the **Application macro** icon.

Related concepts

CADFEKO Application Macros

POSTFEKO Application Macros

3.17 Tools

POSTFEKO has a collection of tools that allows you to quickly validate the model, for example, measure distances, measure angles and finding specific mesh elements.

3.17.1 Measuring a Distance

The measure distance tool allows you to measure or validate the physical distance between two points in a model.

1. On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Tools** group, click the **Measure Distance** icon.
2. Under **Point1**, use Ctrl+Shift+left click to snap to points (for example, named points, geometry points, geometry face centre, geometry edge centre, mesh vertices and grid).
3. Repeat Step 2 for **Point 2**.
   The total distance, as well as the individual X axis, Y axis and Z axis distances, are displayed in the **Distance (D)**, **X distance**, **Y distance** and **Z distance** fields respectively.
4. Click **Close** to close the dialog.

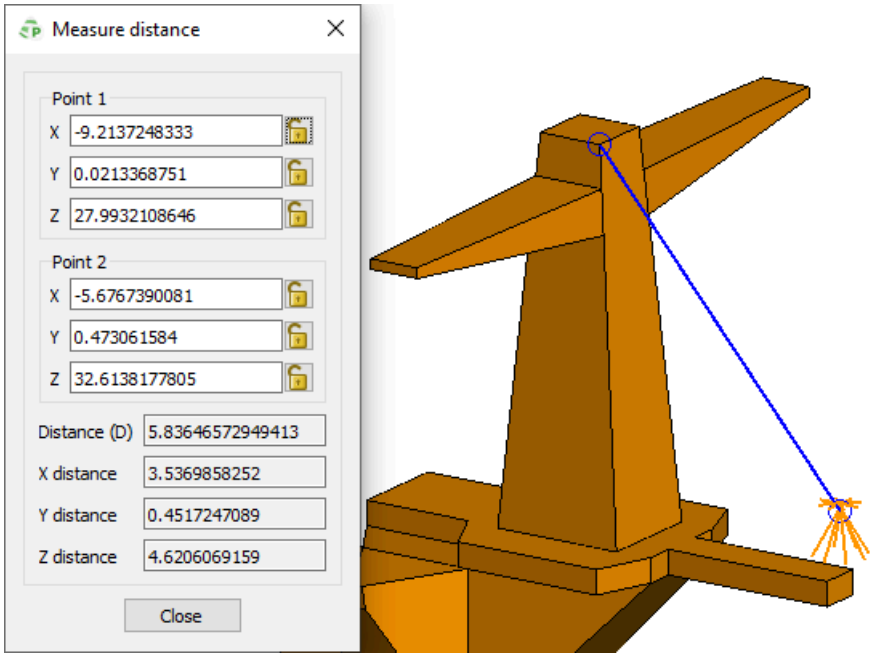


*Figure 485: The **Measure distance** tool.*

3.17.2 Measuring an Angle

Use the angle measuring tool to measure or validate the angle (in degrees) between three points in a model.

1. On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Tools** group, click the **Measure Angle** icon.
2. Under **Point1**, use Ctrl+Shift+left click to snap to points (for example, named points, geometry points, geometry face centre, geometry edge centre, mesh vertices and grid).
3. Repeat Step 2 for **Point 2**.
4. Repeat Step 2 for **Point 3**.
   The angle in degrees is displayed in the **Angle (degrees)** field.
5. Click **Close** to close the dialog.

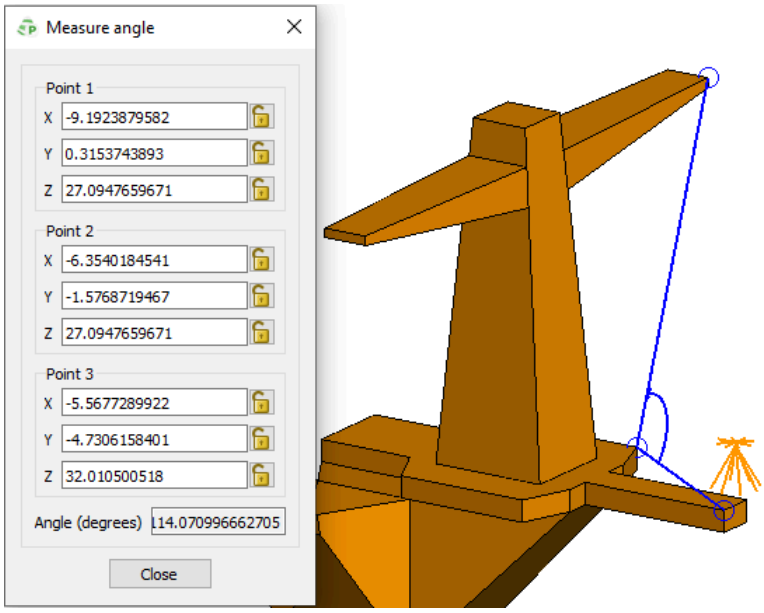


*Figure 486: The **Measure angle** dialog.*

3.17.3 Highlighting the Non-Included Angle

Use the include angle highlighting tool to show the non-included angle between flat mesh triangles.

On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Tools** group, for the **Include angle** icon, specify the non-included angle 210° and click the icon.

The non-included angle is highlighted in the 3D view with pink color where applicable.

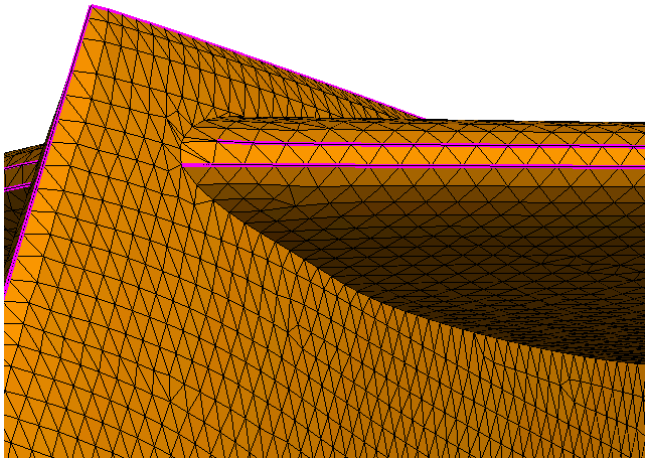


Figure 487: Example of non-included angle displayed in POSTFEKO.

3.17.4 Finding Elements

Locate specific mesh elements by element number (ID) in the 3D view.

When a warning or error message is obtained during the solution of a model, in some cases the message is related to a specific mesh element.[54]. With the **Find elements** tool, you can find and view the location of the mesh element.

1. On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Tools** group, click the **Find Elements** icon.
2. From the **Element type** drop-down list, select the type of mesh element you want to find.
3. In the **Element ID(s)** field, enter the element number(s) you want to find.

> **Tip:** Search for multiple elements by separating the element numbers with a comma.

54. The mesh element ID(s) would be given in the `.out` file

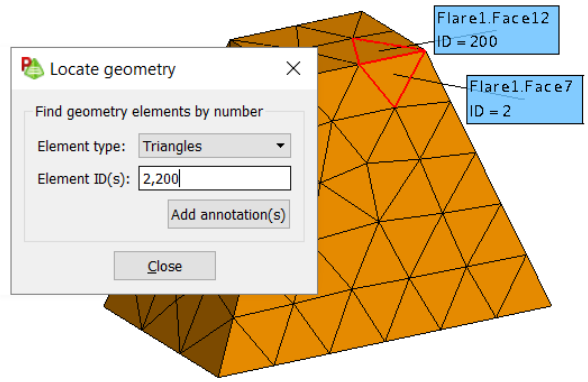


Figure 488: Finding two mesh triangles by ID (number).

4. [Optional] To retain the annotations, click **Add annotation(s)**.
5. Click **Close** to close the dialog.

3.17.5 Confirming Mesh Connectivity

The mesh connectivity tool allows you to view free edges in the 3D view.

Free edges can be used to confirm if a mesh is connected.

> **Note:** A free edge is an edge that is only on the boundary of a single face.

On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Tools** group, click the **Connectivity** icon.

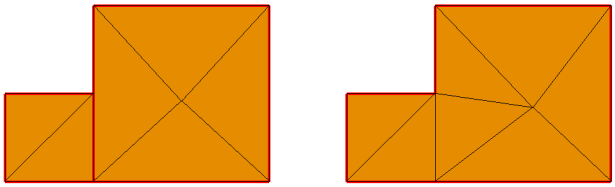


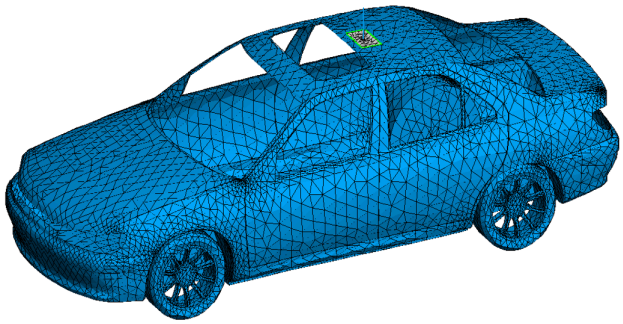
Figure 489: On the left, an example of two unconnected rectangles. To the right, the two rectangles are unioned. Edges displayed in red indicate free edges.

3.17.6 Viewing Domain Connectivity

The domain connectivity tool allows you to view parts in the 3D view that are "connected" using domain connectivity.

On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Tools** group, click the **Domain Connectivity** icon.

The edges “connected” by means of domain connectivity are indicated in green.



Related tasks
Defining Domain Connectivity

3.17.7 Highlighting Specific Mesh Elements

The mesh highlight tool allows you to view areas of the mesh where specific model settings are applied.

On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Tools** group, click the **Highlight** icon. From the drop-down list select one of the following:

- *None*
  No mesh elements are highlighted.
- *Lossy metal*
  Highlight mesh elements (faces, wires) with a metallic medium and thickness applied to it.
- *Coating*
  Highlight mesh elements (faces, wires, edges) with a coating (layered dielectric) applied to it.
- CFIE *CFIE / MFIE*
  Highlight mesh elements (faces) with either a combined field integral equation (CFIE) or magnetic field integral equation (MFIE) applied to it.
- EFIE *EFIE*
  Highlight mesh elements (faces) with the electric field integral equation (EFIE) applied to it.
- *Impedance sheet*
  Highlight mesh elements (wires, faces) with an impedance sheet applied to it.
- *Surface impedance approximation*
  Highlight faces that bound a region set to the dielectric surface impedance approximation.

-  *Physical Optics*

  Highlight mesh elements (faces) with the physical optics (PO) solution method applied to it.

-  *Physical Optics (Fock regions)*

  Highlight mesh elements (faces) with the physical optics (PO) solution method applied to a Fock region.

-  *Ray Launching GO*

  Highlight mesh elements (faces) with the ray launching geometrical optics (RL-GO) solution method applied to it.

-  *Uniform Theory of Diffraction*

  Highlight mesh elements (faces) with the uniform theory of diffraction (UTD) solution method applied to it.

-  *Faceted Uniform Theory of Diffraction*

  Highlight mesh elements (faces) with the faceted uniform theory of diffraction (faceted UTD) solution method applied to it.

-  *FEM*

  Highlight mesh elements (regions) with the finite element method (FEM) solution method applied to it.

-  *VEP*

  Highlight mesh elements (regions) with the volume equivalence principle (VEP) solution method applied to it.

-  *Windscreen solution elements*

  Highlight mesh elements (faces, wires) that are specified as windscreen solution elements (windscreen antenna elements).

-  *Aperture*

  Highlight a slot or aperture in an infinite plane with the planar Green's function aperture applied to it.

-  *Numerical Green's Function*

  Highlight mesh elements defined as the static part using the numerical Green's function.

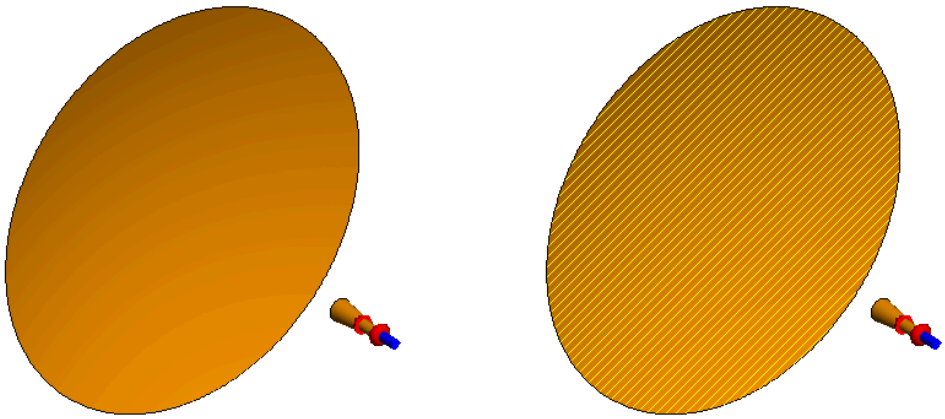


Figure 490: On the left, a 3D view of a horn and a reflector with no highlighting applied. To the right, the reflector is highlighted in yellow to indicate that PO solution method is applied to the face.

3.18 Math Functions in POSTFEKO

View the list of supported math functions.

Shortcut Key	Description
ABS	Absolute value
ARCCOS	Arccosine (radians)
ARCCOT	Arccotangent (radians)
ARCSIN	Arcsine (radians)
ARCTAN	Arctangent (radians)
ATAN2	Arctangent of Y/X in the range -PI to PI Syntax: `ATAN2(X,Y)`
CEIL	Smallest integer value that is equal or greater than the argument
COS	Cosine (radians)
COSH	Hyperbolic cosine (radians)
COT	Cotangent (Radians)
DD	Polynomial derivative of a function Syntax: `DD(function)` Example: `DD(trace1)` where `trace1` is a trace in the result palette.
DEG	Convert radians to degrees
EXP	Exponential function
FFT	Fast Fourier transforms of a real function. The bandwidth can be provided as additional argument. Syntax: `FFT(a,[,bandwidth])` Example: `FFT(U_re, 1e3)` where `U_re` is a trace in the result palette.
FLOOR	Largest integer value that is equal to or smaller than the argument
FMOD	Resulting reminder of a/b. Syntax: `FMOD(a, b)`

Shortcut Key	Description
IFFT	Inverse Fast Fourier transform of a complex function. The resulting range can be specified as additional argument. Syntax: `IFFT(a[,T])` Example: `IFFT((U_re)*w+i*(U_im)*w,1e-3)` where `U_re`, `U_im` and `w` are traces in the result palette.
IMAG	Determines the imaginary component
LN	Computes the natural logarithm Syntax: `LN(a)` Example: `LN(trace1)` where `trace1` is a trace in the result palette.
LOG	Computes logarithm base 10
MAX	Returns the largest of the two arguments Syntax: `MAX(a, b)`
MIN	Returns the smallest of the two arguments Syntax: `MIN(a, b)`
NORM	Normalise a function Syntax:**`NORM(a)`** Example: `NORM(trace1 + trace2)` where `trace1` and `trace2` are traces in the result palette.
ONES	Creates a list of points with a constant value 1 Syntax: `ONES(from, to, samplePoints)`
PEAK	Returns the peak value in the given range of values Syntax: `PEAK(a)` Example: `trace1*PEAK(trace2)` where `trace1` and `trace2` are traces in the result palette.
PHASE	Determines the phase
RAD	Convert degrees to radians

Shortcut Key	Description
RAMP	Ramp function Syntax: `RAMP(from, to, samplePoints)` Example: `RAMP(0,1e-3,2001)`
REAL	Determines the real component
SELF	A reference to the current trace
SIN	Sine (Radians)
SINH	Hyperbolic sine (Radians)
SQRT	Square root
STEP	Step function Syntax: `STEP(X) = 0 for X <= 0` and `STEP(X) = 1 for X > 0`
TAN	Tangent (Radians)
TANH	Hyperbolic tangent (radians)
ZEROS	Creates a list of points with a constant value of 0 Syntax: `ZEROS(from, to, samplePoints)`

Predefined Variables

Table 49: List of predefined variables.

Variable	Description
i	Imaginary unit
j	Imaginary unit
c0	The speed of light in free space in m/sec.
eps0	The permittivity of free space in F/m.
mu0	The permeability of free space in H/m.
pi	The mathematical constant (Ludolph’s number).

Variable	Description
zf0	The characteristic impedance of free space in Ohm.

3.19 Files Generated by POSTFEKO

View the files associated and generated by POSTFEKO.

Table 50: Files generated by POSTFEKO

Argument	Description
`.fek`	POSTFEKO reads the `.fek` to display the geometry and the calculation requests (for example the near field request points will be displayed if a near field calculation was requested).
`.bof`	POSTFEKO reads the `.bof` file to display the results as obtained by the Solver. Incomplete `.bof` files can be loaded and the results displayed. The results for discrete frequency calculations are displayed as they become available. This allows simulations that terminated due to system power failure to be loaded and displayed, showing the results which were calculated prior to the failure.
`.out`	The `.out` file may be displayed to view information regarding the Solver version, date, memory usage and results obtained by the Solver and any errors and warnings etc.
`.pfs`	Contains the POSTFEKO workspace, for example, views, graphs, models, settings and references to result files which were present at the time of save.
`.pfg`	The `.pfg` file is used to store optimisation process information used for graphing in POSTFEKO after / during an optimisation run.

3.20 Default Shortcut Keys

View the shortcut keys available for POSTFEKO for faster and easier operation of POSTFEKO.

Keyboard shortcut keys help you to save time accessing actions that you perform regularly. The shortcut key or key combination is also displayed in the keytip that is displayed when you hover the mouse over the action on the ribbon.

Shortcut Key	**Description**
Alt+0	Run CADFEKO.
Alt+1	Run EDITFEKO.
Alt+2	Run PREFEKO.
Alt+4	Run Solver.
Alt+6	Run OPTFEKO.
Alt+8	Open the Feko terminal.
General Editing	
F1	Context-sensitive help for the dialog / window that has focus.
Ctrl+C	Copy image to clipboard.
Ctrl+X	Copy data to clipboard.
Ctrl+F	Locate geometry (3D view).
Ctrl+F	Find and replace text (script editor).
Ctrl+E	Export image.
Ctrl+P	Print current window.
Ctrl+Shift+O	Open POSTFEKO project file.
Ctrl+N	Create a new session.
Ctrl+O	Add a model.
Ctrl+S	Save POSTFEKO session file.
Ctrl+Q	Quit POSTFEKO.
Ctrl+Z	Undo

Shortcut Key	Description
Ctrl+Y	Redo.
Alt+B	Show / hide the project browser.
Alt+B	Show / hide the visibility the project browser.
Ctrl+K	Duplicate trace / component.
F2	Rename trace / result.
Ctrl+F2	Change the labels, title and footer of a graph
Ctrl+Shift+left click	Add annotation in 3D view or to a Cartesian graph or polar graph.
Shift+F2	Edit trace text.
Ctrl++	Raise trace.
Ctrl+-	Lower trace.
Del	Delete selected items.
View	
F5	Zoom to extents.
0	Restore view.
8	Top view.
2	Bottom view.
5	Front view.
Ctrl+5	Back view.
4	Left view.
6	Right view.
3D View Interaction	
F5	Zoom to extents.
Shift + hold while scrolling mouse wheel	Slow zoom (3D view).
Scroll mouse wheel	Zoom (3D view).
Click + drag with middle mouse button	Panning (3D view, schematic view).

Shortcut Key	**Description**
Ctrl + click / drag	Panning (3D view).
Left click + drag mouse	Rotation (3D view).
Script Editor	
Ctrl+N	Create a new empty script.
Ctrl+O	Open script.
Ctrl++	Zoom in.
Ctrl+-	Zoom out.
Ctrl+G	Goto line.

EDITFEKO

4

EDITFEKO is used to construct advanced models (both the geometry and solution requirements) using a high-level scripting language which includes loops and conditional statements.

This chapter covers the following:

4.1 Introduction to EDITFEKO

EDITFEKO is a scripting interface for advanced users to construct models using a high-level scripting language, which includes `FOR` loops and conditional `IF-ELSE` statements.

EDITFEKO can also be used for advanced editing of a model created in CADFEKO. Most models do not require the use of EDITFEKO, but some advanced options are not available in EDITFEKO and would require EDITFEKO.

When creating a model in EDITFEKO, the model geometry and calculation requests are entered on separate lines in the `.pre` file and are referred to as cards. Each card has a number of parameters that must be specified in a specific order. The language used in EDITFEKO is known as the PREFEKO language (the PREFEKO application translates the cards into a format understood by the solver and saves it to a .fek file).

Important: The order of the cards in the `.pre` file controls the order of the steps during the simulation.

4.1.1 EDITFEKO Workflow

View the typical workflow when working with the Feko component - EDITFEKO.

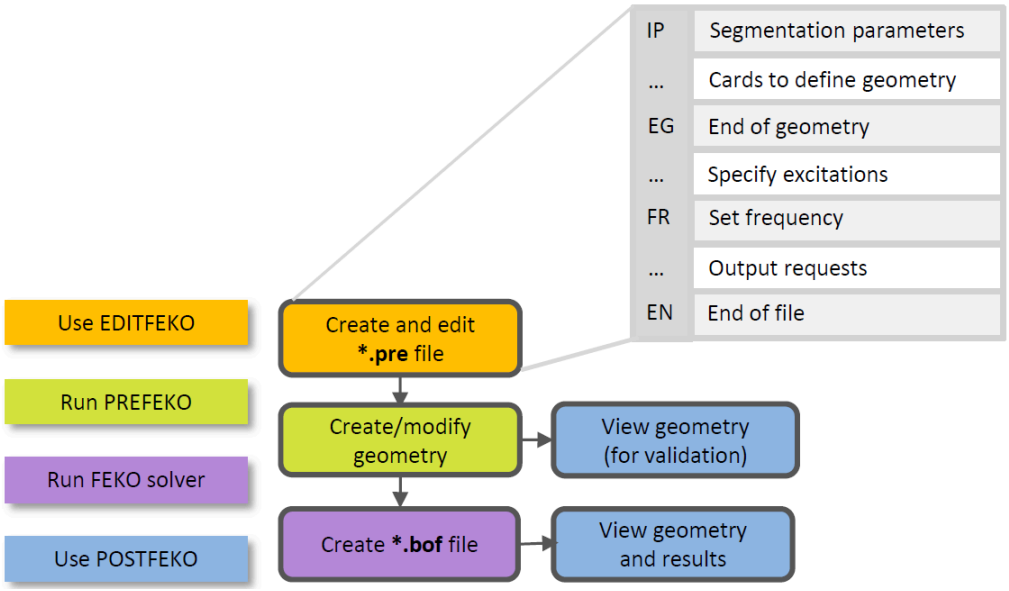


Figure 491: Illustration of the EDITFEKO workflow.

Create the PRE File

Define the `.pre` file containing the mesh parameters, geometry, excitations, frequency and output requests.

Verify the Model

Although PREFEKO is run as part of running Feko, it is recommended to first run PREFEKO to verify the commands and syntax of the `.pre` file. The `.pre` file does not have to be complete, but requires at least an EG card and EN card.

If no error is given, view the (partial) model mesh, settings and requests in POSTFEKO.

Run the Solver

Run the Feko solver to obtain simulation results for the output requests. Take note of notes and warnings to ensure that the model setup correctly. Any errors will terminate the simulation and has to be corrected.

View Model and Results in POSTFEKO

The completed model and results can be viewed in POSTFEKO on a 3D view or 2D graphs. The ASCII .out file produced during the simulation can also be viewed in POSTFEKO.

Related concepts
Structure of the PRE File

4.1.2 Launching EDITFEKO (Windows)

There are several options available to launch EDITFEKO on Windows.

Launch EDITFEKO using one of the following workflows:

- Open EDITFEKO using the Launcher utility.
- Open EDITFEKO by double-clicking a `.pre` file.
- Open EDITFEKO from other components, for example, from inside CADFEKO and POSTFEKO.

> **Note:** If the application icon is used to launch EDITFEKO, no model is loaded and the start page is shown. Launching EDITFEKO from other Feko components, automatically loads the model into the editor.

Related tasks
Opening the Launcher Utility (Windows)

4.1.3 Launching EDITFEKO (Linux)

There are several options available to launch EDITFEKO on Linux.

Launch EDITFEKO using one of the following workflows:

- Open EDITFEKO using the Launcher utility.
- Open a command terminal. Use the absolute path to the location where the EDITFEKO executable resides, for example:

```
/home/user/2024.1/altair/feko/bin/editfeko
```

- Open a command terminal. Source the "initfeko" 3D view using the absolute path to it, for example:

  ```
  . /home/user/2024.1/altair/feko/bin/initfeko
  ```

  Sourcing `initfeko` ensures that the correct Feko environment is setup. Type `editfeko` and press Enter.

  > **Note:** Take note that sourcing a script requires a dot (".") followed by a space (" ") and then the path to `initfeko` in order for the changes to be applied to the current shell and not a sub-shell.

Related tasks
Opening the Launcher Utility (Linux)

4.1.4 Command Line Arguments for Launching EDITFEKO

EDITFEKO can be launched via the command line. Use command line arguments to pass information on how EDITFEKO is to be launched.

Syntax using the command-line options:

```
editfeko [FILES] [OPTIONS]
```

FILES
Loads the specified `.pre` files. Any number of `.pre` files can be loaded.

OPTIONS

-h, --help
Displays the help message.

--version
Print the version information and exit.

If EDITFEKO is launched without providing a filename, no model is loaded and the start page is shown. Launching EDITFEKO with a .pre file, it loads the model into the editor.

4.1.5 Start Page

The Feko start page is displayed when starting a new instance (no models are loaded) of CADFEKO, EDITFEKO or POSTFEKO.

The start page provides quick access to **Create a new model**, **Open an existing model** and a list of **Recent models**.

Links to the documentation (in PDF format), introduction videos and website resources are available on the start page. Click the icon to launch the Feko help.

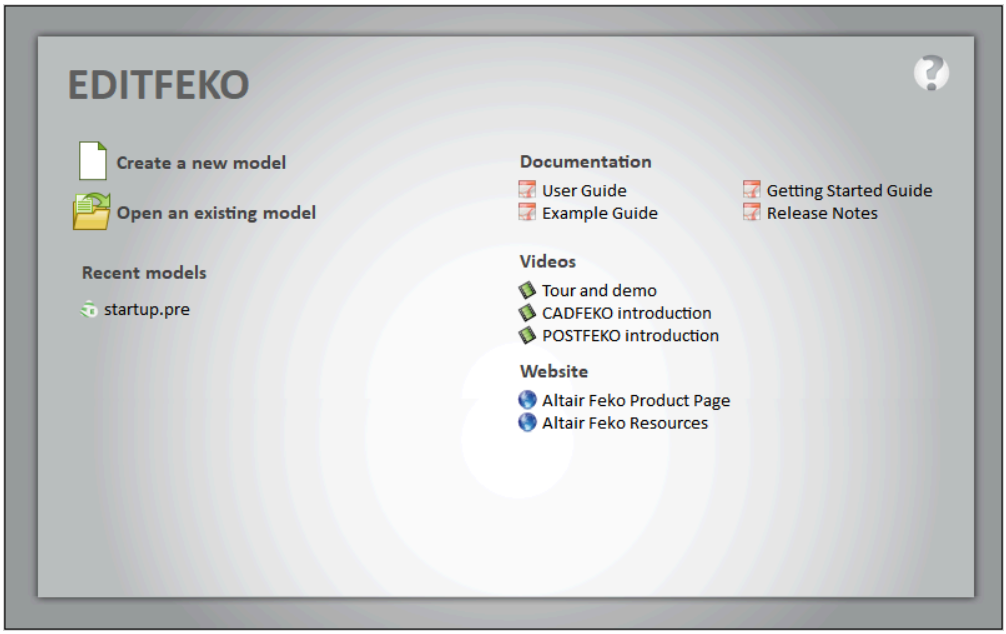


Figure 492: The EDITFEKO start page.

4.2 Quick Tour of the EDITFEKO Interface

View the main elements and terminology in the EDITFEKO graphical user interface (GUI).

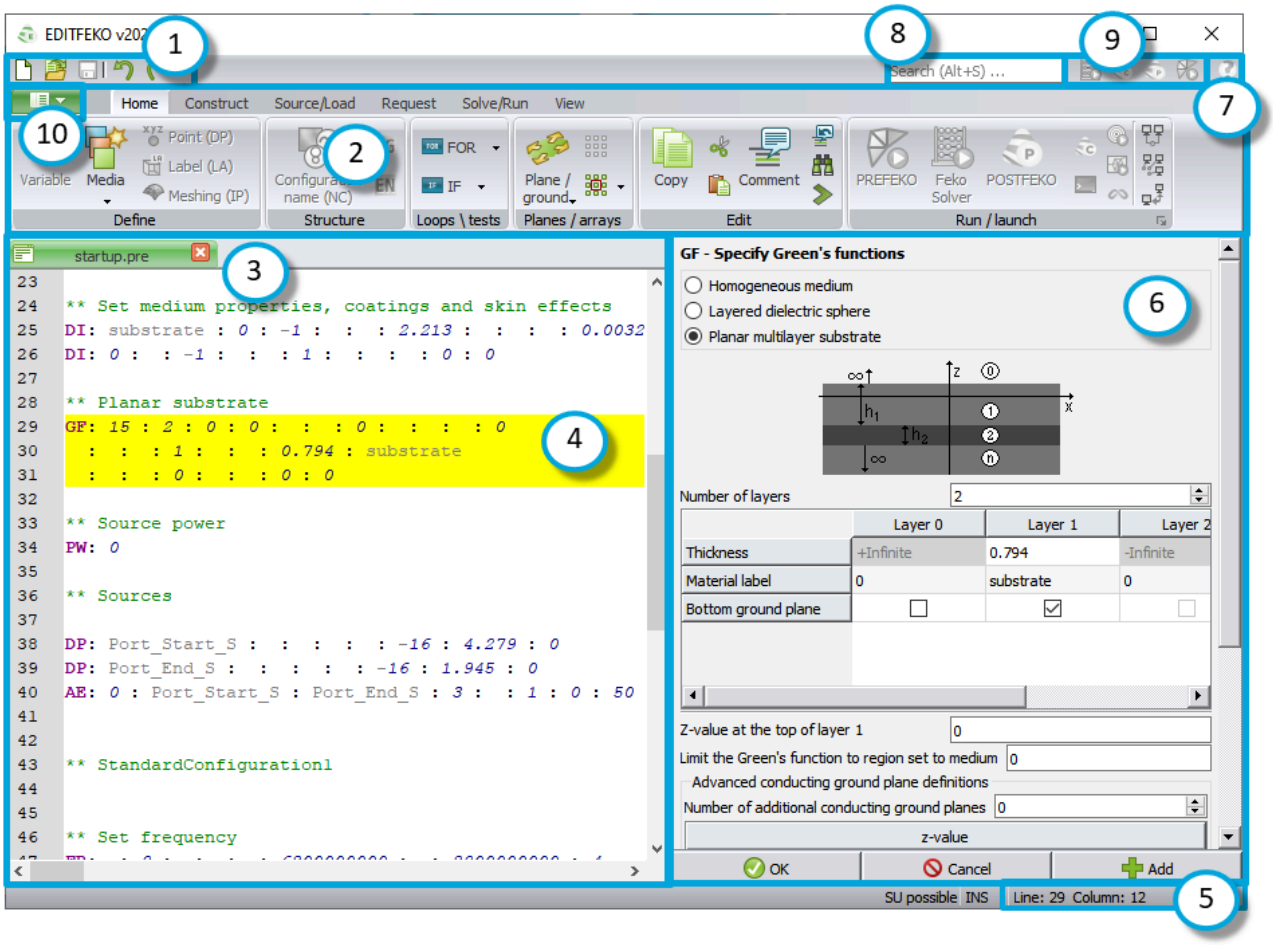


Figure 493: The EDITFEKO window.

1. Quick Access Toolbar
2. Ribbon
3. Script Editor Area
4. Edit Card
5. Status Bar
6. Card Panel
7. Help
8. Search Bar
9. Application Launcher
10. Application Menu

4.2.1 Quick Access Toolbar

The quick access toolbar is a small toolbar that gives quick access to actions that are often performed.

The toolbar is located at the top-left corner of the application window, just below the title bar. It allows you to create a new model, open a model, save a model, undo a model operation or redo a model operation using fewer mouse clicks for a faster workflow. The actions available on the quick access toolbar are also available via the ribbon.

4.2.2 Ribbon

The ribbon is a command bar that groups similar actions in a series of tabs.

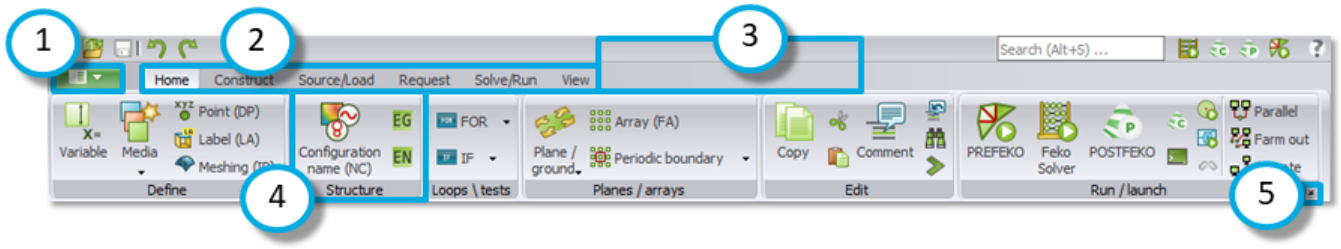


Figure 494: The ribbon in EDITFEKO.

1. Application menu
The application menu button is the first item on the ribbon. When the application menu drop-down button is clicked, the application menu is displayed. The menu allows saving and loading of models, import and export options as well as giving access to application-wide settings and a recent file list.

2. Core tabs
A tab that is always displayed on the ribbon, for example, the **Home** tab and **Construct** tab.

The **Home** tab is the first tab on the ribbon and contains the most frequently used commands for quick access.

3. Contextual tab sets
A tab that is only displayed in a specific context.

EDITFEKO does not have any contextual tabs.

4. Ribbon group
A ribbon tab consists of groups that contain similar actions or commands.

5. Dialog launcher
Click the dialog launcher to launch a dialog with additional and advanced settings that relate to that group. Most groups don't have dialog launcher buttons.

Keytips
A keytip is the keyboard shortcut for a button or tab that allows navigating the ribbon using a keyboard (without using a mouse). Press F10 to display the keytips. Type the indicated keytip to open the tab or perform the selected action.

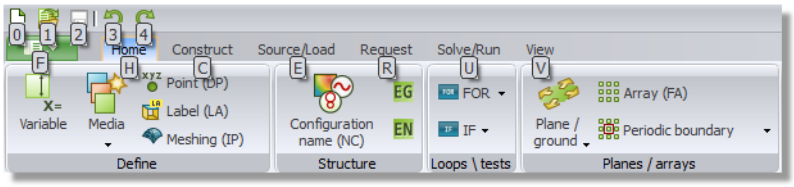


Figure 495: An example of keytips.

Application Menu

The application menu is similar to a standard file menu of an application. It allows saving and loading of models, print functionality and gives access to application-wide settings.

When you click on the application menu drop-down button, the application menu, consisting of two panels, is displayed.

The first panel gives you access to application-wide settings, for example:

- Creating a new model.
- Opening a model, saving a model and closing a model.
- Print
- Check for updates
- Settings
  - Preferences
  - Component launch options
- Feko help
- About
  - Version information about EDITFEKO
  - Information about Altair HyperWorks Products
  - Information about third-party libraries
- Exit

The second panel consists of a recent file list and is replaced by a sub-menu when a menu item is selected.

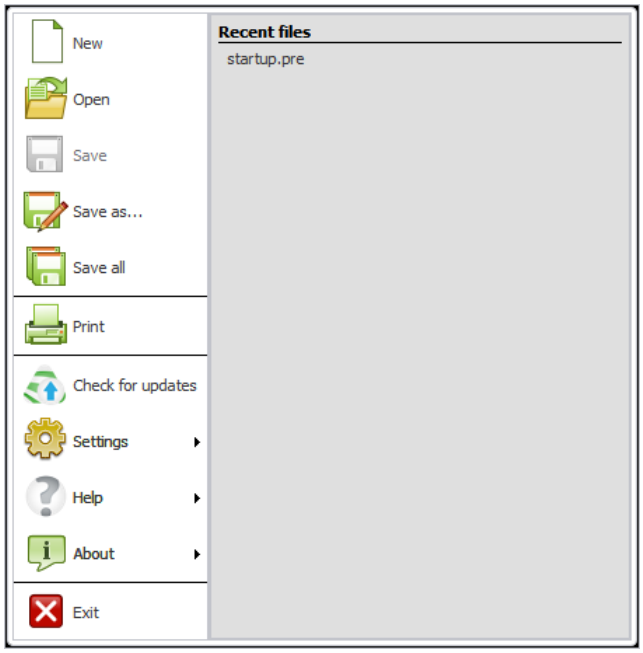


Figure 496: The application menu in EDITFEKO.

Home Tab

The **Home** tab is the first tab on the ribbon and contains the most frequently used operations.

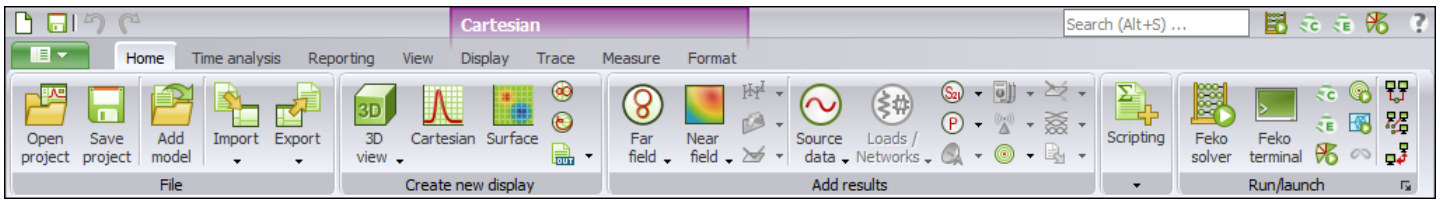


*Figure 497: The **Home** tab in EDITFEKO.*

4.2.3 Script Editor Area

The editor area allows you to edit `.pre` files. The script editor includes syntax highlighting and each file is contained in its own tab.

> **Tip:**
>
> - Re-order the window tabs by simply dragging the tab to the desired location.
> - View the path to the open `.pre` file by hovering with mouse cursor over the window tab.
> - Drag-and-drop functionality is supported.

The editing tools are available on the **Home** tab, in the **Edit** group.

Table 51: Editing tools.

Icon	Icon text	Description
	**Copy**	Copy the selected text to clipboard. Shortcut: Ctrl+C
	**Cut**	Cut the selected text to clipboard. Shortcut: Ctrl+X
	**Paste**	Paste the text from clipboard. Shortcut: Ctr+V
	**Comment**	Block comment the selected items. Shortcut: Alt+C
	**Uncomment**	Uncomment the selected items. Shortcut: Alt+U
	**Find / Replace**	A find and replace tool with the following text search functionality: **Find next**, **Find previous**, **Replace**, **Replace all**, **Close**. Shortcut: Ctrl+F
	**Goto line**	A tool which allows you to find a specific line in the script. This is useful when PREFEKO reports an error with a corresponding line number.

4.2.4 Edit Card

Press F1 on a card to highlight the card entry in the editor area and display the full card definition in the card panel.

> **Note:** A yellow background for a card entry indicates that the selected card is in editing mode.

4.2.5 Status Bar

The status bar is a small toolbar that shows the line and column number for the current cursor position as well as the setting for the text editor (insert or overwrite).

The status bar is located at the bottom-right of the application window. Options on the status bar are also available on the ribbon, but since the status bar is always visible, they are easily accessible no matter which ribbon tab is selected.

4.2.6 Card Panel

The card panel contains the full card definition and provides editing of the card parameters.

Press F1 on a card to highlight the card entry in the editor area and display the full card definition in the card panel. A new card can be created by clicking on the corresponding button on the ribbon.

Card panels make it easy to edit cards and enter data in the correct card fields. The panels support cards that span multiple lines and they automatically use the correct card format (column or colon delimited). Once the panel has been populated with the data, click on **OK** to apply the changes, write the card and close the panel.

Tip:

- Click the **OK** button to add the card to the `.pre` file and close the card panel.
- Click the **Add** button to add the card to the `.pre` file, but keep the card panel open.

4.2.7 Help

The **Help** icon provides access to the Feko documentation.

Press F1 to access context-sensitive help. The context-sensitive help opens the help on a page that is relevant to the selected dialog, panel or view.

The first time you press F1 on a card, the panel for the card will be opened. Pressing F1 on an open panel will access context-sensitive help. The documentation for the card will provide information regarding the different options on the panel and the meaning of the settings.

Tip: When no help context is associated with the current dialog or panel, the help opens on the main help page that allows you to navigate the documentation or search in the documentation for relevant information.

4.2.8 Search Bar

The search bar is a single-line text field that allows you to enter search terms and find relevant information in the GUI or the documentation.

The search bar is located at the top-right of the application window.

Tip:

- Enter a search term in the search bar to populate a drop-down list of actions as well as the location of the action on the ribbon or context menu.
- Click an item in the list to execute the action.
- Partial searches are supported.
- Search the documentation.

4.2.9 Application Launcher

The application launcher toolbar is a small toolbar that provides quick access to other Feko components.

4.2.10 Application Menu

The application menu is similar to a standard file menu of an application. It allows saving and loading of models, print functionality and gives access to application-wide settings.

When you click on the application menu drop-down button, the application menu, consisting of two panels, is displayed.

The first panel gives you access to application-wide settings, for example:

- Creating a new model.
- Opening a model, saving a model and closing a model.
- Print
- Check for updates
- Settings
  - Preferences
  - Component launch options
- Feko help
- About
  - Version information about EDITFEKO
  - Information about Altair HyperWorks Products
  - Information about third-party libraries
- Exit

The second panel consists of a recent file list and is replaced by a sub-menu when a menu item is selected.

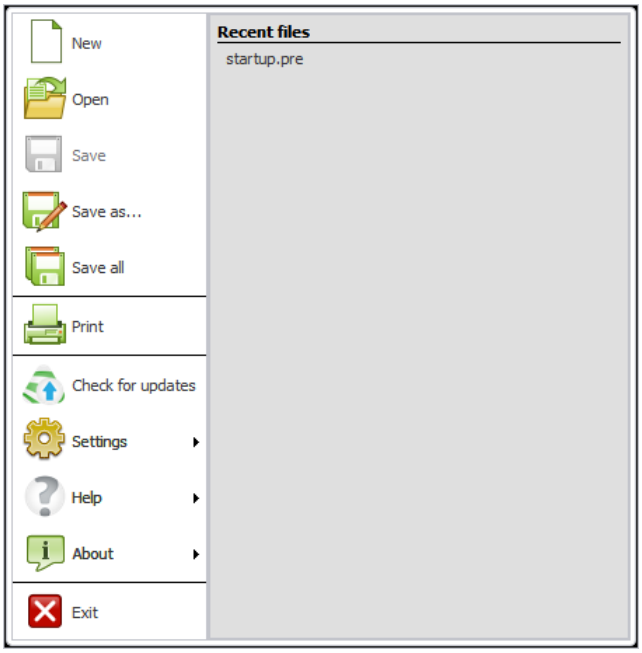


Figure 498: The application menu in EDITFEKO.

4.3 PREFEKO Language Concepts

The language used to create and modify models in EDITFEKO is the PREFEKO language.

EDITFEKO is the editor or integrated development environment (IDE) used to create models in an ASCII format, but the language is PREFEKO. PREFEKO also refers to the application that translates `.pre` files into `.fek` files that is read by the Feko solver. In order to create models with EDITFEKO, it is vital to understand the language concepts in the PREFEKO language.

4.3.1 Comments

Comments are descriptive text added to the .pre file to help understand and follow the code execution.

Comments can be added to the script by inserting "**" followed by a space, for example:

```
** This is a comment
```

A comment may also be added after the last column of a card, after the comment indicator ("**").

> **Note:** Some cards use comments at the end of the card to indicate the name of the source, load or request. Care should be taken not to mistake these for comments.

4.3.2 Structure of the PRE File

The order of the cards in the .pre file specifies the order of the steps during the simulation.

There are two main types of cards in EDITFEKO:

Geometry cards
: Cards used to create geometry and affect the meshing. These cards are used above the EG card.

Control cards
: Cards used to define sources, loads, request and control the simulation. These cards are generally below the EG card, but can usually be used above the EG card as well.

The structure of the `.pre` file consists of the following sections:

1. Specify the mesh parameters.
   a. Define the IP card. All cards following the IP card inherit the mesh settings set with this card.
2. Create the geometry.
   a. Use the geometry cards to define the geometry of the model.
   b. End with the EG card to indicate that the geometry creation is complete.
3. Specify the excitations, loads, frequency and output requests.
   a. Use the control cards to define excitations, specify the frequency and add output requests.
   b. End with the EN card to indicate the end of the file.

For control cards that define solution requests, "**" is used often as a label for that card. The label of a card is used by OPTFEKO to identify specific results. The label is also used by POSTFEKO for the identification of the solutions and output requests when post-processing simulation results.

For example:

```
**   Comments at the start the input file
...  Cards that define the geometry ** Comments
EG   End of the geometry
...  Control cards that define sources, special solution options
     and indicate which quantities to calculate ** Card labels / comments
EG   End of the input file
```

Note:

All input and output parameters are in SI units (for example, lengths are in metres, potential in volts). All angles are in degrees.

See the SF, TG and IN cards to enter dimensions in different units and scale to metres.

Related concepts
Card Formats
Related reference
IP Card
EG Card
EN Card
SF Card
TG Card
IN Card

4.3.3 Card Formats

Two card formats are supported namely column-based and colon-separated. This is relevant to users who externally generate Feko input files and can be ignored by users using EDITFEKO to modify the cards.

Note: The two formats may be mixed in a single input file.

Column Based Format

This format separates the individual integer and real parameters in columns, see Figure 499. The upper numbers indicate the columns. The name field ("xx") in columns 1 and 2 specifies the type of the card (all cards start with a unique two character combination). This is followed by five integer parameters I_1 to I_5 (these input fields may also contain text such as node names) containing five digits each, and eight real-value parameters R_1 to R_8 containing ten digits each.

1	6	10	15	20	25	30	40	50	60	70	80	90	100	110
xx		I_1	I_2	I_3	I_4	I_5	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
		INT STR	INT STR	INT STR	INT STR	INT STR	REAL	REAL	REAL	REAL	REAL	REAL	REAL	REAL

Figure 499: Column based card format in EDITFEKO. The numbers above the table (1, 6, 10, 15, 20...110) indicate the columns.

Colon Separated Format

The colon separated format separates the individual integer and real parameters by a colon character. It is a less restrictive format than the column-based format. Unlike the column-based format, integer and real input fields are not restricted to 5 or 10 characters respectively. Note that the card name is still located in columns 1 and 2. The name is followed by a colon in column 3. The rest of the card has no spacing limitations.

For example:

```
DP: S1 : : : : : #x : #y : #z
BP: S2 : S2 : S3 : S4
```

4.3.4 Variables

Variables are parameters that help to create easily adjustable models such as the investigation of structures with varying geometry.

Symbolic Variables

Symbolic variables can contain expressions to calculate specific parameters of the model and have specific syntax requirements.

Instead of using numerical values in the cards, it is possible to use predefined variables. The name of a variable always consists of the "#" symbol followed by a string consisting of the characters "a-z", "A-Z", "0-9" and the special character "_".

The following are examples of valid variable names:

- *#height*
- *#a*
- *#STARTINGFREQUENCY*
- *#a_1*
- *#P5_7f*

The following are examples of invalid variable names:

- *#a?1*
- *#value2.1*

Note: There is no distinction between upper and lower case characters.
For example, *#a* and *#A* are interpreted as the same variable.

It is important to note that in CADFEKO variables are used without the "#" character whereas PREFEKO requires the "#" character to distinguish between variables and functions (such as cosine).

When CADFEKO writes the variables to the .cfm file, it prepends the "#" character so that the variables can be used after the IN card in the .pre file. When using OPTFEKO in a model that has no CADFEKO geometry defined, the .cfm file mesh import command must be removed, or variables that are defined in the .cfm file must be excluded from the import, as these variable values will override the values assigned by OPTFEKO if they are included.

Expressions and functions are used when defining variables so that direct calculations can be carried out. The variables must be defined before they can be used in the respective cards. It is possible to use expressions such as `2*#radius` in the input fields subject to the maximum allowed length for the column based format (10 characters for real values, 5 characters for integer values). For larger expressions additional variables must be defined or the colon based format can be used.

Examples of variables:

```
#2pi = 2*#pi
#vara = 1 + sqrt(2)
#varb = #vara * 2.3e-2 * (sin(#pi/6) + sin(rad(40)) + #vara^2)
#sum = #vara+#varb
```

Note: The "#" character must appear in the first column to define a variable.

Variable Editor

The # card presents a list of supported functions and operations. Use this card to calculate the value of the variable as it would be evaluated by PREFEKO at this point.

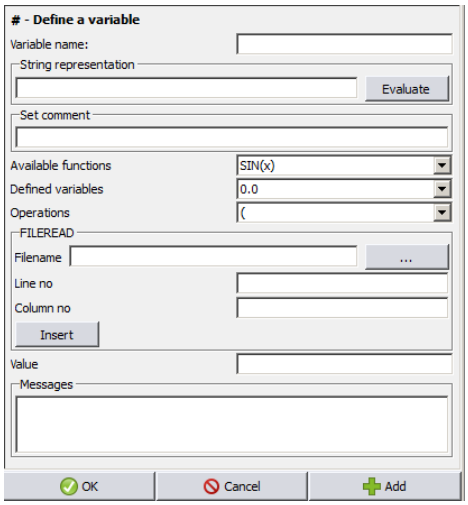


*Figure 500: The **# - Define a variable** card.*

Arrays

Arrays provide functionality to allocate a series of numbers to a parameter.

Arrays are supported and indexed with the notation such as `#a[5]`. More complex arrays are supported where the array is constructed from an expression:

```
#am_0[3*#i+ceil(#r[2])]
```

The expression between the square brackets must evaluate to an integer number, which can also be negative. The implementation of using arrays is such that they do not need to be allocated, however they need to be initialised.

Consider the following lines of code:

```
!!for #i = 10 to 20
#array_a[#i] = 3*#i-10
#array_b[-#i] = 0
!!next
```

It would be possible to use `#array_a[10]` or `#array_a[17]` or also `#array_b[-12]` in other expressions. But, trying to use for instance `#array_a[5]` or `#array_b[0]` results in an error message that an undefined variable is used.

Predefined Variables

The PREFEKO language includes a number of predefined variables. Generally, these variables remain constant but may be overwritten by re-assignments.

Table 52: Predefined variables list

Name	Value	Description
`#pi`	3.14159265358979...	The constant π
`#esp0`	$8.85418781761 \cdot 10^{-7}$	Dielectric constant ε_0 of free space.
`#mu0`	$4\pi \cdot 10^{-7}$	Dielectric constant μ_0 of free space.
`#c0`	$\frac{1}{\sqrt{\mu_0 \varepsilon_0}}$	The speed of light in free space.
`#zf0`	$\sqrt{\frac{\mu_0}{\varepsilon_0}}$	The intrinsic impedance of free space.
`#true`	1	Used for logical true.
`#false`	0	Used for logical false.

PREFEKO also supports a logical function `DEFINED(#variable)` which returns `TRUE` if the variable *#variable* has been defined, and `FALSE` if not. This is useful in `.pre` files used for OPTFEKO or ADAPTFEKO runs. These two components insert variables at the top of the file, but it may be required to define the variable in the file for preview purposes.

For example, if a `.pre` file is used for optimisation with respect to the variable *#a*, this variable could be defined as follows:

```
!!if (not(defined(#a))) then
#a = 200.0e-3
!!endif
```

Logical and Mathematical Operators

Logical operations are supported and a specific order of precedence is followed.

PREFEKO allows the use of logical operations. It supports the function `NOT()` that returns `TRUE` if the argument is `FALSE` and `FALSE` when the argument is `TRUE`. PREFEKO also supports the delimiters >, <, >=, <=, =, <>, `AND` and `OR`. When boolean operations are applied to variables, a value of 0 is taken as `FALSE` and everything else is interpreted as `TRUE`. Similarly, in the result of a logical operation, `FALSE` is mapped to 0 and `TRUE` to 1.

The order of precedence is as follows:

1. single number, expressions in brackets
2. function calls
3. + and - (when used as a sign)
4. ^
5. * and /
6. + and -
7. >, <, >= and <=
8. = and <>
9. AND
10. OR

There are three other special variables *#!x*, *#!y* and *#!z* that are useful for the connection of complex wire structures. The three variables specify the Cartesian coordinates of the end point of the wire segment most recently defined. This enables the correct and easy connection of a straight wire to a curved length of wire, as the next extract from an input file demonstrates:

```
CL .....
DP   A                        #!x       #!y       #!z
#z = #!z + 0.5
DP   B                        #!x       #!y       #z
BL   A    B
```

The following example demonstrates the use of variables:

```
** A dielectric sphere in the field of an incident wave

** Define the variables
#r = 1          **   Radius of the sphere
#betrad = 1     **   Electrical size of the sphere
#epsr = 15      **   The relative dielectric constant
#maxlen = 0.7   **   The maximum edge length

** Define segmentation parameters
IP                                    #maxlen

** The corner points
DP     A                       0         0         0
DP     B                       0         0         #r
DP     C                       #r        0         0

** Select the medium
ME     1    0

** Generate an eighth of the sphere
KU     A    B    C             0         0         90        90        #maxlen

** Use symmetry in all three coordinate planes
**   yz-plane: ideal electrically conducting plane
**   xz-plane: ideal magnetically conducting plane
**   xy-plane: only geometrically symmetric
SY   1    2    3    1

** End of the geometry
EG   1    0    0    0    0
```

```
** Assigning the dielectric's properties
DI                            #epsr     1.0

** Incident plane wave excitation
#freq = #betrad * #c0/(2*#pi*#r)
FR   1    0                   #freq
A0   0         1    1         1.0       0.0      -180.0

** Near fields along the Z axis
FE   1    1    1    25   0     0.0       0.0      -1.98     0.0      0.0      0.04
FE   4    1    1    50   0     0.0       0.0      -0.98     0.0      0.0      0.04
FE   1    1    1    25   0     0.0       0.0       1.02     0.0      0.0      0.04

** End
EN
```

The use of variables makes the investigation of structures with varying geometry (such as the variable distance of the antenna in front of a reflector) an easy process, because only one variable (the distance parameter) needs to be changed. It also allows `FOR` loops and `IF` conditions.

Mathematical Functions

Various trigonometric, Bessel and miscellaneous functions are built into Feko to help construct geometry, expressions and calculate parameters.

Trigonometric Functions

The following trigonometric functions are supported:

Table 53: Trigonometric functions

`SIN`	sine (argument in radians)
`COS`	cosine (argument in radians)
`TAN`	tangent (argument in radians)
`COT`	cotangent (argument in radians)
`ARCSIN`	arcsine (argument in radians)
`ARCCOS`	arccosine (argument in radians)
`ARCTAN`	arctangent (in radians)
`ATAN2`	This function has two arguments `atan2(#y,#x)` - it yields `arctan(#y/#x)` in the range $-\pi...\pi$
`ARCCOT`	arccotangent
`SINH`	hyperbolic sine
`COSH`	hyperbolic cosine

`TANH`	hyperbolic tangent

Bessel Functions

The following Bessel functions are supported:

`BESJ(n,x)`	Bessel function $J_n(x)$ of integer order $n \geq 0$ and real argument x.
`BESY(n,x)`	Neumann function $Y_n(x)$ of integer order $n \geq 0$ and real argument x
`BESI(n,x)`	Modified Bessel function of the first kind $I_n(x)$ of integer order $n \geq 0$ and real argument x
`BESK(n,x)`	Modified Bessel function of the second kind $K_n(x)$ of integer order $n \geq 0$ and real argument x

Miscellaneous Functions

The following miscellaneous functions are supported:

Table 54: Miscellaneous functions

`SQRT`	Square root
`LOG`	Logarithm to base 10.
`LN`	Natural logarithm
`EXP`	Exponential function
`ABS`	Absolute value
`DEG`	Convert radians into degrees.
`RAD`	Convert degrees into radians.
`STEP`	Step function, `STEP(x) = 0` for $x \leq 0$ and `STEP(x) = 1` for x > 0.
`CEIL`	Smallest integer value that is equal to or greater than the argument.
`FLOOR`	Largest integer value that is equal to or smaller than the argument.
`MAX`	Returns the largest of the two arguments — called as `max(#a,#b)`.

MIN	Returns the smallest of the two arguments — called as `min(#a,#b)`.
FMOD	This function has two arguments `fmod(#a,#b)` and returns the remainder of the division `#a/#b`.
RANDOM	This function returns a random value in the range 0 . . . 1. If the argument `X` of `RANDOM()` is -1, then a random number is returned. For any other argument `X` in the range 0 . . . 1 this value is used to set the seed, and then a random number is created using this seed. (Using the same seed allows one to create a deterministic and reproducible random number series). If `RANDOM(-1)` is called before any seed is set in the `.pre` file, then the returned values are random and not reproducible. (The internal seed is used based on the time when PREFEKO is executed).

Coordinate functions provide access to the individual X coordinate, Y coordinate and Z coordinate of a Cartesian coordinate in 3D space.

Coordinate Functions

The following coordinate functions are supported:

Table 55: Coordinate functions

X_COORD	This function returns the X coordinate of a point previously defined by a DP card.
Y_COORD	This function returns the Y coordinate of a point previously defined by a DP card.
Z_COORD	This function returns the Z coordinate of a point previously defined by a DP card.

The `X_COORD`, `Y_COORD` and `Z_COORD` functions are used by passing the name of the point, in quotation marks, as an argument to the function. For example, the following code sets the parameter *#x* equal to 1.234.

```
DP   PNT01                    1.234     0.4567    #z
#x = x_coord("PNT01")
```

4.3.5 Labels in Feko

In the PREFEKO language and Feko in general, items in a model are identified by their labels. Operations are performed and electromagnetic properties are applied to the items through their labels.

Labels are set either directly in CADFEKO or by preceding geometry cards in EDITFEKO with the LA card. When importing specific mesh formats by means of the IN card then labels can also be imported (for instance the NASTRAN property gets converted into an Altair label).

Labels can consist of any or a combination of the following:

- A positive integer number, including zero (for example, 0, 1, 2 ..9).
- Any valid expression (for example, `3*#i+2`). The expressions are evaluated, and the resultant numerical value is used in the label.
- A string of characters (valid are, a..z, A..Z, 0..9 and the underscore "_"), optionally followed by a variable (which starts with the "#" sign). Such variables at the end are evaluated and replaced by the corresponding numerical value (rounded to an integer).

> **Note:** String labels are case insensitive. The labels "*Roof*" and "*ROOF*" are treated identically.

For example, the following labels are valid:

```
23
5*#k+#j/2
LeftWing
Front_Door
Part#i
```

The following labels are invalid:

```
Left+Wing (invalid character '+')
-23 (negative integer)
Part_#i_#k (two variables)
```

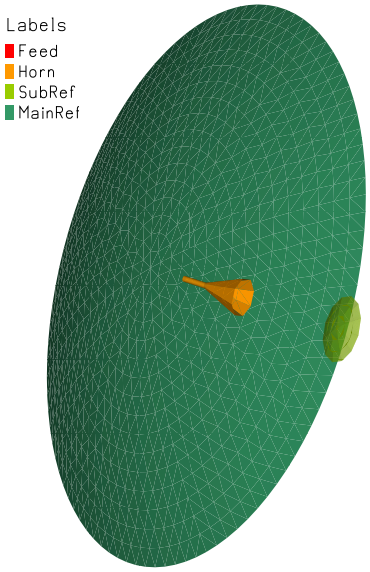


Figure 501: Example demonstrating the usage of labels (display of labels in colour with legend in POSTFEKO.

You can use the CB card in EDITFEKO to change labels (for example, after having imported geometry). A powerful wild card algorithm (expanding a non-specific label name containing a wild card character into a set of specific labels) is supported. Some Feko cards allow you to specify label ranges while other cards allow labels for created geometry to be derived from other labels (for example when using symmetry with the SY card). It is therefore important to understand how the label algorithm works.

The labelling algorithm first evaluates expressions or replaces variables, and then the label is split into the associated number and the remaining base string. The associated number is split off from the back of the label, and if there are no digits, this is set to zero.

Table 56: Examples of splitting a label into its base string and associated number.

Label	Base string	Associated number
`1`		`1`
`Roof`	`Roof`	`0`
`Part_17`	`Part_`	`17`

When incrementing labels, the base string is kept and the associated number is incremented. There is just one exception: the label zero will always remain zero.

Table 57: Label incrementing example (increment by two).

Label	Incrementing label
1	3

Label	Incrementing label
Roof	Roof
Part_17	Part_19

Related concepts
Reference Elements
Related reference
LA Card
IN Card
CB Card
SY Card

4.3.6 Conditional Statements

Conditional statements provide functionality for changing parameters inside a loop, or for a parameter(s) to depend on other parameter(s).

FOR / NEXT Loops

`FOR` / `NEXT` loops provide the flexibility of varying parameters inside a loop.

Some cards in EDITFEKO implicitly use loops (such as when an FR card with multiple frequencies is used). This does not always provide the flexibility which may be required. For example, to change the material parameters inside the loop. Another example would be the use of a loop to create a complex geometry object(s).

For general loops, PREFEKO allows the construct:

```
!!for #var = #start to #end step #delta
!!next
```

where a simple example would be as follows:

```
** Loop for the relative permittivity
!!for #eps_r = 1 to 5 step 0.5
** Set material parameters
GF   0    1          #epsr
** Compute fields etc.
FE
** End of loop
!!next
```

The syntax requirements of `FOR` / `NEXT` loops are as follows:

- The `!!` characters must be located in the first two columns of the line. This is followed by a number of optional spaces and the keyword `FOR` (it is not case sensitive, so also “For” or “for” are accepted).

- The keyword FOR is followed by the name of the loop variable (starting with "#").
- Next follows an expression for the initial value of the loop (a constant, variable or formula).
- This is followed by the keyword TO and the terminating value of the loop variable (again a constant, variable or formula).
- The default increment of the loop variable is 1, but it can be changed by using the keyword STEP followed by an expression. Negative increments are allowed.
- The loop is terminated by a line of the form !!NEXT (spaces are allowed between !! and NEXT but not before the !!). All instructions and input cards between !!FOR and !!NEXT are evaluated repeatedly inside the loop.
- Loops can be nested.

A more complicated example:

```
#end = 3+sin(4)
!!for #x1 = sqrt(5) + 2*3 to 2*#end step -#end/10
!! for #x2 = 1.23 to 2*#x1 ** this is the inner loop
#x3 = #x1 + #x2
DP ....
.... (more commands)
!! next
!!nex
```

Related concepts
FILEREAD Function
Related reference
Mathematical Functions

IF / ELSE / ENDIF Constructs

The IF, ELSE and ENDIF constructs allow different control cards to be used under certain conditions.

The syntax requirements of IF / ELSE / ENDIF constructs are as follows:

- The !! characters must be in the first two columns of the line. This is followed by an arbitrary number of spaces, the keyword IF, the expression to be evaluated and the keyword THEN.

  Note: Keywords are case insensitive, for example, "*Then*" or "*then*" are also valid

- The block is terminated by a line of the form !!ENDIF (again spaces are allowed between !! and ENDIF but not before the !!).
- An optional line of the form !!ELSE (the !! must be in the first two columns and spaces are allowed before the keyword, which is not case sensitive).
- All instructions and input cards between !!IF and !!ENDIF (or !!ELSE if it is present) are processed if the expression is TRUE. If it is present, all lines between !!ELSE and !!ENDIF are processed if the expression is FALSE.

  As an example:

```
!!if #a > 5 then
...
```

```
!!endif
```

Another example is as follows:

```
#l = (#a+5 > 21) and (#a < 100)
!!if ( (3*#a+5 >= #x/2) and not(#l) ) then
...
!!else
!! if (sin(#x/10) > 0.5) then
...
!! else
...
!! endif
!!endif
```

Related concepts
FILEREAD Function
Related reference
Mathematical Functions

EXIT Command

Use the `EXIT` command to break execution of a loop.

This command is useful for checks such as the following:

```
!!if #a < 2*#b then
!! exit
!!endif
```

4.3.7 PRINT Command

The `PRINT` command prints strings, numbers and other information to screen or the `.out` file to display the solution progress and for debugging purposes.

The following print commands are available:

!!print
: Prints text to screen.

!!print_warning
: Prints the warning messages to the screen.
  - For terminal runs, the string "WARNING" precedes the warning message.
  - For runs from the GUI, the warning message is displayed in colour.

!!print_error
: Prints the warning messages to the screen. To stop execution use the `!!EXIT` statement.

!!print_to_out
: Writes the text to the `.out` file while the Feko kernel is run.

The print commands accept multiple arguments separated by commas.

Example 1

Print the error message and exit if the variable *#a* is < 2**#b*:

```
!!print "2*#b = ", 2*#b
!!if #a < 2*#b then
!! print_error "The value of #a is too small:", #a, " (exiting now)"
!! exit
!!endif
```

Example 2

Print the value of *#b* to the .out file at the location where it appears in the .pre file.

```
!!print_to_out "This run was done with #b = ", #b
```

4.3.8 FILEREAD Function

The `FILEREAD` function reads data from an arbitrary ASCII file.

Read a numerical value using the following general syntax:

```
fileread("Filename", Line, Column)
```

The `FILEREAD` command contains the file name, the line number to read from and the column to read. The data in the respective columns of any line are separated by one or more spaces or tab characters.

For example, consider a data file containing a list of frequencies and a load impedance for each frequency:

```
Frequency in MHz    Re(load) in Ohm    Im(load in Ohm)
100                   22.54               -12.56
150                   25.07               -6.54
200                   27.42                0.23
```

The frequency and loading can be imported directly from this file using the following example code:

```
#numfreq = 3 ** Number of frequencies
!!for #i = 1 to #numfreq
** Define the frequency (conversion from MHz to Hz)
#freq = 1.0e6*fileread("datafile.dat", #i+1, 1)
FR   1    0                    #freq
** Define the load
#Zr = fileread("datafile.dat", #i+1, 2)
#Zi = fileread("datafile.dat", #i+1, 3)
LZ                  0          #Zr       #Zi
** Computations ...
!!next ** End of frequency loop
```

4.3.9 Symbolic Node Names

Symbolic node names or named points can be constructed as either single points or as an array of points. Arrays of points can then be referenced by only specifying the array name. This is useful when a large number of points are required.

Single Node Names

A node is a point in 3D space. Nodes can be constructed and referenced using variable names. Use a loop to construct multiple nodes.

When defining or using node names, simple variable names of the form *A#i* can be used to define the node. If a hash character ("`#`") is found in a node point name, this character and everything that follows is interpreted as a variable string, evaluated and rounded to the nearest integer.

As an example, `#k=15` and a point defined as *P#k*, is equivalent to using *P15* as point name. The length of the node name string (before and after expansion) is limited to 5 characters.

For example, the points *P1* to *P20* are defined inside a loop as follows:

```
!!for #k = 1 to 20
DP P#k
!!next
```

These defined points can then be used individually or inside another loop.

Node Name Arrays

Symbolic node names or named points can be constructed as either single points or as an array of points. An array of nodes is useful when creating a polygonal surface with multiple points.

For example, when creating polygonal surfaces using the PY card and PM card containing many points, only the node name can be specified instead of each individual point. Expressions such as `A[2*#i+3]` can be used to index the array.

A symbolic node name array can be defined in a loop as follows:

```
!!for #k = 1 to 20
DP P[2*#k+3]
!!next
```

When using node names, the nodes can be referenced using only `P`. Single node names can be referenced by indexing the array.

4.4 Creating Geometry in EDITFEKO

Create geometry in EDITFEKO according to the guidelines to ensure that different mesh parts are electrically connected. Meshes can also be imported to create a model.

4.4.1 Importing Meshes

Include a CADFEKO mesh or any other external mesh in the `.pre` file using the IN card.

4.4.2 Guidelines for Mesh Connectivity

Meshing guidelines are given to ensure electrical connectivity in the mesh.

Elements must be connected at edges or vertices to ensure electrical connectivity. Most of these rules are automatically complied with when creating Feko models in CADFEKO. However, adherence to these rules should be maintained when combining CADFEKO models with EDITFEKO scripting (for example attaching an antenna modelled with geometry cards on an aircraft meshed in CADFEKO), or when creating the geometry only in EDITFEKO, or when working with imported meshes.

> **Note:** Cuboidal volume elements used to model volume dielectrics (with the DK, DZ and QU cards), do not need to be connected in this manner.

When creating structures with scripting commands, wires are divided into segments that are equal to or shorter than the specified segment length. For surfaces the triangle edges along the boundary of the surface are always equal to or shorter than the specified edge length. Therefore, meshing the same line with the same mesh size will always give the same number of divisions of equal length. The internal edges may, however, be longer than the specified edge length. This is not necessarily the case with CADFEKO meshes where the specified mesh size is the average size and the internal structure influences the placement of vertices along the surface boundaries.

When creating wire junctions as shown in the figure below, it is important to ensure that the wire AB have a vertex at point C. The best option is to construct this as two wires, one from A to C and the other from C to B.

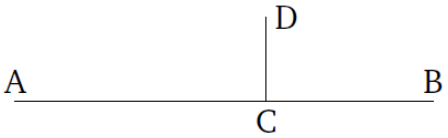


Figure 502: Example of a wire structure.

Similarly, where two surfaces touch, the common edge must be part of both surfaces. For example, the surface in the figure below should not be created as two rectangles ABFG and CDEF. If done in this manner, it is highly unlikely that there will be an ohmic connection along the line BF. There are a number of ways to correctly create this structure. It can be created from the rectangles ABFG, CDHB and BHEF or the quadrangles ABEG and BCDE. In both cases the contacting edges are common

and will be meshed correctly. The simplest way to mesh this structure is to create a single polygon ABCD(H)E(F)G.

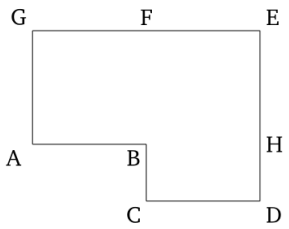


Figure 503: Example of a wire structure.

A connection point between a segment and one or more triangles is only recognised when the beginning or the end of the segment is coincident with the vertex or vertices of the triangles. In the figure below an incorrect connection is depicted on the left and a correct connection on the right (where the segment is connected to six triangles).

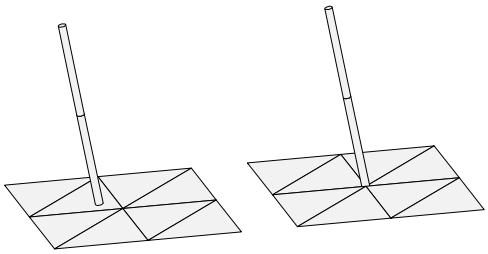


Figure 504: Incorrect (left) and correct (right) connection between a segment and triangles.

When curved structures (such as circles, cylinders, spheres and so forth) are modelled, a finer mesh may be used along the curved edges to get a more accurate representation of the geometry. In this case the same edge length should be used on both edges and the reference points should be identical as depicted in the figure below.

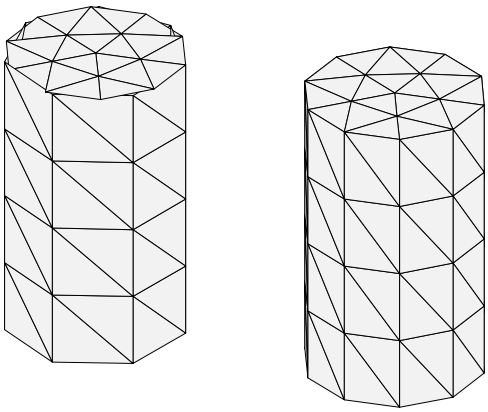


Figure 505: Incorrect (left) and correct (right) connection between a segment and triangles.

Related reference

DK Card

DZ Card

QU Card

4.4.3 Discontinuous Mesh and Geometry Parts

Use domain connectivity (discontinuous Galerkin domain decomposition method) to "connect" a static mesh to a dynamic parameterised geometry without requiring node connectivity. The parts may be separated by a small gap.

> **Note:** Mesh connectivity between parts is achieved only when the parts have a common interface with shared vertices (a continuous mesh).

A simulation model is often assembled from different parts. For example, a car model meshed using Altair HyperWorks and imported into CADFEKO. An antenna geometry is then placed on the imported mesh. To avoid re-meshing the full model to align the vertices on the common interface, the domain connectivity approach can be used to "connect" the discontinuous mesh and geometry parts.

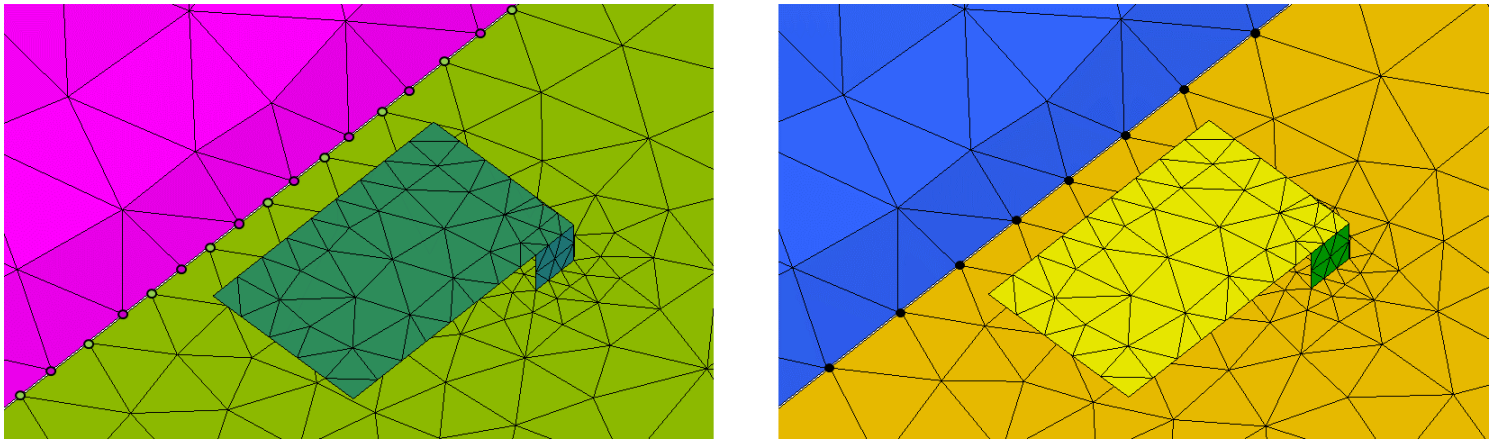


Figure 506: Left: Mesh parts that have no mesh connectivity (triangles do not share vertices on a common interface); Right: Mesh parts that have mesh connectivity.

Apply the domain connectivity to discontinuous meshes where the gap between the meshes is small in relation to the wavelength. For high accuracy, choose the gap distance g as $g = \lambda/1000$. Increasing the gap distance, the result becomes less accurate and margins $g > \lambda/100$ should be avoided.

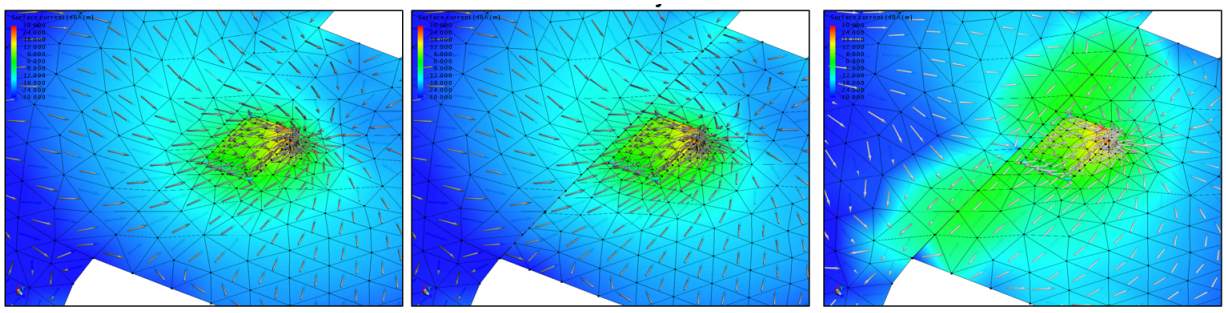


Figure 507: Left: Mesh parts that have mesh connectivity; Center: Mesh parts that do not have mesh connectivity but have domain connectivity defined; Right: Mesh parts that have no mesh connectivity.

Related concepts

Workflow for Connecting Discontinuous Mesh and Geometry Parts

Domain Connectivity

Related tasks
Defining Domain Connectivity in CADFEKODefining Domain Connectivity in CADFEKO
Related reference
DC Card

Workflow for Connecting Discontinuous Mesh and Geometry Parts

View the workflow for domain connectivity (discontinuous Galerkin domain decomposition method) to connect discontinuous mesh and geometry parts using CADFEKO to set up the model and EDITFEKO to define the domain connectivity.

> **Note:** Use this workflow when setting up domain connectivity in EDITFEKO.
> Domain connectivity can also be defined in CADFEKO.

Connect discontinuous mesh and geometry parts using the following workflow:

1. Assemble the model in CADFEKO and define all settings and requests.
2. Save the model. CADFEKO creates the `.pre` file.
3. Open the `.pre` file in EDITFEKO.
4. Define the domain connectivity using the DC card.
   - Specify the number of connections.
   - Define for each connection the labels of the corresponding faces.

     > **Tip:** To view the label names for the relevant faces, view the model in POSTFEKO and colour the mesh by label[55]. Add a legend to view the labels per colour[56].

   - Define for each connection a tolerance distance, in order to distinguish between regions, where a gap in the model is desired (or not desired).
5. Run PREFEKO to create the `.fek` file for the simulation.
6. Run the Solver to simulate the model.

Related concepts
Discontinuous Mesh and Geometry Parts
Domain Connectivity
Related tasks
Defining Domain Connectivity in CADFEKODefining Domain Connectivity in CADFEKO
Related reference
DC Card

55. On the **3D View** contextual tabs set, on the **Mesh** tab, in the **Rendering** group, click the **Mesh colour** icon. From the drop-down list, click **Mesh colour**.
56. On the **3D View** contextual tabs set, on the **Display** tab, in the **Legends** group, click the **Top left** icon.

4.4.4 Reducing Mesh Sizes

Mesh subdivision is a method to reduce the number of mesh elements created / rendered, while still solving the correctly intended mesh.

For very large models (or at very small wavelengths) it is possible that CADFEKO or POSTFEKO cannot create / display the required mesh. This problem can be alleviated by creating a mesh of larger elements in CADFEKO and using the RM card in EDITFEKO to subdivide the mesh to obtain the correctly sized elements.

Note: The original mesh should use much larger elements than the desired mesh. If this is not the case, the subdivision may result in an unnecessary large number of elements.

4.5 Preferences

EDITFEKO has various default settings that you can configure to customise it to your preference.

On the application menu, click **Settings** > **Preferences**. The settings can be reset to the default settings at any time, restoring the settings to the state of a new installation.

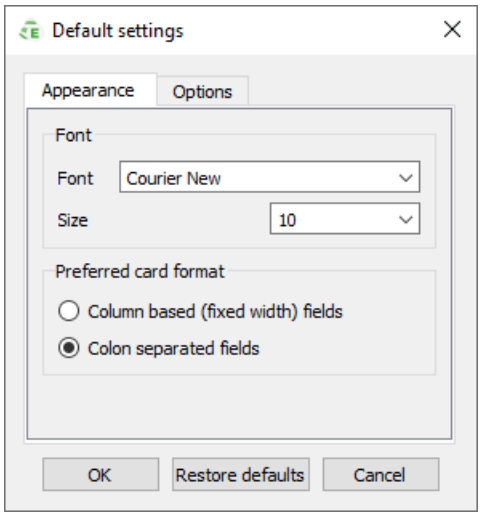


*Figure 508: The **Default settings** dialog.*

4.6 Files Generated by EDITFEKO

View the files associated and generated by EDITFEKO.

Table 58: File type(s) generated by EDITFEKO

Files	Description
`.pre`	A `.pre` file is created when the EDITFEKO model is saved.

Note: After saving a model in CADFEKO, CADFEKO generates a `.pre` file.

4.7 Shortcut Keys

View the shortcut keys available for EDITFEKO for faster and easier operation of EDITFEKO.

Keyboard shortcut keys help you to save time accessing actions that you perform regularly. The shortcut key or key combination is also displayed in the tooltip that is displayed when you hover the mouse over the action on the ribbon.

Table 59: EDITFEKO shortcut keys

Shortcut Key	**Description**
Feko Components	
Alt+0	Run CADFEKO
Alt+2	Run PREFEKO.
Alt+3	Run POSTFEKO.
Alt+4	Run Solver
Alt+6	Run OPTFEKO
Alt+8	Open the Feko terminal.
General Editing and Construction	
F1	Edit card in panel. If a panel is already open, open contextual help for the card.
Ctrl+A	Select all (text).
Ctrl+C or Ctrl Ins	Copy to clipboard.
Ctrl+G	Goto line.
Ctrl+N	New `.pre` file.
Ctrl+O	Open file.
Ctrl+P	Print.
Ctrl+Q	Exit.
Ctrl+S	Save.
Ctrl+V or Shift+Ins	Paste from clipboard
Ctrl+X or Shift+Del	Cut to clipboard.

Ctrl+Z	Undo.
Ctrl+Y	Redo.
Alt+C	Comment line(s).
Alt+U	Uncomment line(s).
Ctrl+F	Find and replace.
F3	Find next.
Ctrl+Left Arrow	Move the cursor to the previous word.
Ctrl+Right Arrow	Move the cursor to the next word.
Home	Move to beginning of line.
End	Move to end of line.
Ctrl+Home	Move to beginning of file.
Ctrl+End	Move to end of file.

Feko Solution Methods

5

One of the key features in Feko is that it includes a broad set of unique and hybridised solution methods. Effective use of Feko features requires an understanding of the available methods.

This chapter covers the following:

5.1 Basic Concepts

Basic antenna and EM concepts are given that provide a foundation for understanding the different solver methods in Feko.

What is an Antenna?

An antenna is a conducting structure consisting of surfaces and/or wires designed to be of specific characteristic dimensions to radiate or receive an electromagnetic wave.

The primary purpose of an antenna is to make an impedance match between a signal (electrical current or wave) travelling in a coaxial cable (transmission line) or wave guide and waves travelling in free space. A secondary purpose is to send the signal in a specific direction. [57][58]

Figure 509: A Yagi-Uda antenna.

What is a Far Field?

When calculating fields at a specific point P in space over a great distance from the radiator, the following assumptions hold:

- Differences in the distances from P to the different points on the radiator have a negligible effect on the magnitude of the field.
- Differences in the distances from P to the different points on the radiator should be accounted for when calculating the phase, but certain assumptions could be made.
- All field components that decay faster than $\frac{1}{r}$ can be considered negligible compared to those that decay with $\frac{1}{r}$.[59]

The far field of an antenna is the minimum distance from the antenna where the field components do not contain reactive components, or where these components can be considered negligibly small. This distance is generally written as:

$$r = \frac{2D^2}{\lambda} \tag{40}$$

where D is the largest dimension of the antenna.

57. Fields and Waves in Communication Electronics, Third Edition, Ramo, Whinnery and Van Duzer, p. 584-586; 599
58. Electromagnetic Fields and Energy. Haus and Melcher, p. 547
59. Fields and Waves in Communication Electronics, Third Edition, Ramo, Whinnery and Van Duzer, p. 593

In the far field, the antenna is considered a point source. Equation 40 is derived under the assumption that the varying distances to the radiator do not contribute to phase errors larger than 22.5°.[60]

What is a Near Field?

The near field of an antenna is the region in close proximity to the antenna where the electric and magnetic fields are not in phase. The fields are reactive and there is also a strong radial component. The radial component of the field has no $\frac{1}{r}$ dependency but does have $\frac{1}{r^2}$, $\frac{1}{r^3}$ and even higher dependencies. Naturally, these field components vanish very quickly with increasing distance.[61]

What is a Transmission Line?

A transmission line is a coaxial cable, microstrip, stripline, waveguide or some specialized structure designed to conduct a radio frequency signal. The frequency of the signal is high enough, such that the wave behaviour of the signal cannot be ignored. While there are several purposes, in radio frequency engineering, transmission lines are typically used to connect transmitters, receivers and antennas.

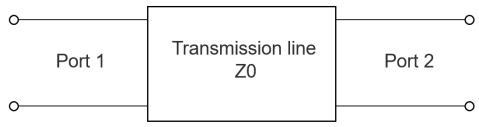


Figure 510: A basic representation of a transmission line.

From a radio frequency engineering point of view, typical parameters of interest are the input reflection coefficient and the voltage standing wave ratio (VSWR).

What are S-Parameters?

S-parameters characterize the relationship between the input and output ports of a system in terms of power waves. While the relationship could also be described in terms of other network parameters such as ABCD, Z and Y-parameters, calculating these parameters require the termination of the ports in open or short circuits. Achieving purely open or short circuits, especially over wide bands, are not feasible. In addition, some devices are not stable if they are open or short-circuited.

However, when calculating S-parameters, it only requires termination of the ports in the system impedance.[62]

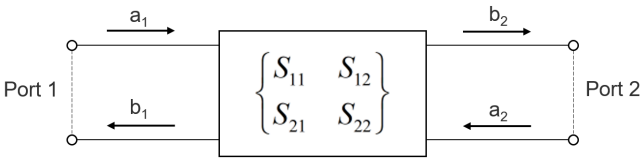


Figure 511: Two-port S-parameters representation.

60. Antenna Engineering Handbook, Fourth Edition, John L. Volakis, p. 1-8
61. Advanced Engineering Electromagnetics, Second Edition, Constantine A. Balanis, p. 283
62. High-Frequency Circuit Design and Measurements, Peter C.L. Yip, p. 33-34

What is the Reflection Coefficient?

The reflection coefficient is a quantity or figure describing how much of an electromagnetic wave is reflected due to an impedance mismatch (discontinuity) in the transmission line or transmission medium. The reflection coefficient is calculated as the ratio of the magnitude of the reflected wave to the incident wave. It can be calculated from the characteristic impedance of the transmission medium (line), Z_0, and the impedance of the discontinuity (often the load impedance at the end of a transmission line), Z_l, as follows:

$$\Gamma = \frac{Z_l - Z_0}{Z_l + Z_0} \tag{41}$$

For the special case of a one-port device, the reflection coefficient is the same as the S-parameter, S_{11}. In the case of a device with two ports or more, the parameters, S_{nn} (for example S_{11}, S_{22}), is the same as the reflection coefficient if all the ports are loaded with the port or system impedances.

> **Note:** A Feko model with a single port does not require an S-parameter request. The input reflection coefficient is the same as the S-parameter request.

What is a Smith Chart?

A Smith chart is a graphical representation of impedance, admittance, phase, wavelength and reflection coefficient. The Smith chart consists of a family of normalized resistance circles and reactance circles. The circles represent the value of the input impedance of some system (network, circuit or load) as measured a certain distance away from the system over a transmission line. [63] The Smith chart provides a transformation between reflection coefficient and impedance over a transmission line. It represents wave behaviour on a transmission line.[64] The advantage of the Smith chart is that you can represent all complex values from positive to zero and negative infinity for the real and imaginary parts.

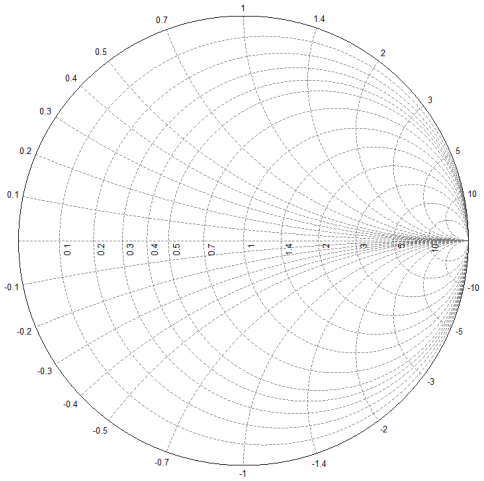


Figure 512: An empty Smith chart from POSTFEKO.

63. High-Frequency Circuit Design and Measurements. Peter C.L. Yip, p. 15
64. High-Frequency Amplifiers, Second Edition, Ralph S. Carson, p. 56

What is CEM?

Computational electromagnetics (CEM) refers to a numerical solution or a computer-based approximation of the currents or the fields. In the numerical solution the currents or fields are firstly divided into many small parts. Subsequently the physical equations (typically Maxwell's equations such as Ampère's law, Faraday's law) that describe the relationships between fields, currents and charges are then used to obtain the magnitude and phase of each current or field element. Finally the summing (integration) of these current or field elements yield antenna parameters such as input impedance and far fields.

Consider the Yagi-Uda antenna shown in Figure 513. Only the current-carrying parts of the antenna are shown. More specifically, the antenna is represented as an assembly of subdivided sections where each section (or rather, each junction between sections) carries a small but uniform current.

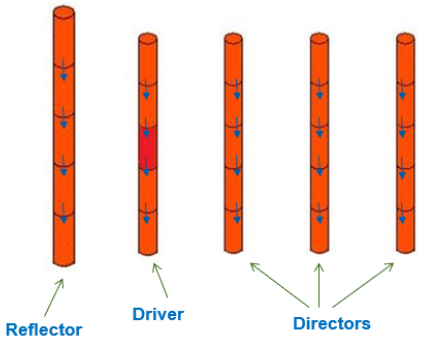


Figure 513: A Yagi-Uda antenna, current-carrying parts only, divided into small sections with uniform current elements across the junctions.

It is initially assumed that on each junction between sections, a constant current is flowing, but the magnitude and phase of these currents are not known. Maxwell's equations are used to find the magnitude and phase of each small current element.

Once the magnitude and phase of each current element is known, all the currents are added together (integrated). A transformation of the currents then give the electric and magnetic near and far fields as well as other antenna parameters.

In post-processing, the fields as seen in Figure 514 can be visualised.

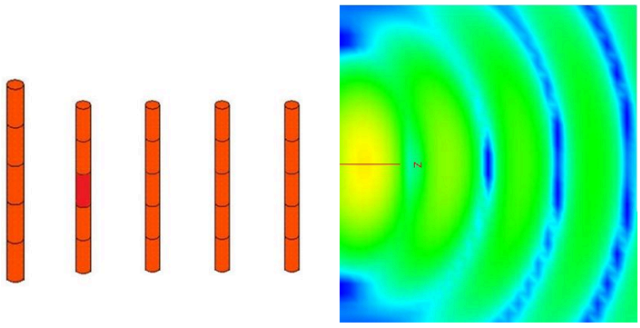


Figure 514: Yagi-Uda antenna near fields visualised in POSTFEKO.

5.2 Source Methods and Field Methods

Solver methods can be categorized as either source-based methods or field-based methods. Understanding the main differences between these two categories helps to understand and choose an appropriate solution method for each application.

Discretization

In source methods, only the structure is discretized (meshed) but not the free-space regions between the structures. In field methods, the whole solution domain is discretized, that is the structure as well as the free-space region between structures.

Consider a dipole and cuboid in Figure 515. In the source method, only the surfaces of the model are discretized. Here the cuboid is discretized into triangles and the dipole into wire segments. A surface mesh could also consist of quadrangles.

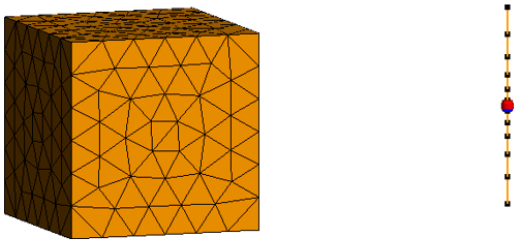


Figure 515: A dipole and cuboid discretized for a source-based solution.

In field methods, the whole solution space is discretized into, for example, voxels (small cuboids shown in Figure 516) or it could be tetrahedra. The field-based mesh is displayed partially transparent to show that the internal volume of the cuboid is meshed. In addition, the surrounding free-space is also meshed, but only the mesh of the outer boundary of free-space is displayed (also transparent).

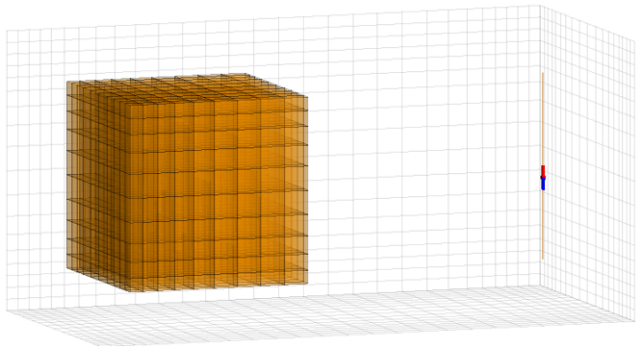


Figure 516: A dipole and cuboid discretized for a field-based solution.

Note: For the source-based mesh displayed in Figure 515 the chosen source-based solution method is the method of moments and for the field-based in Figure 516 it is the finite difference time domain.

Boundary Conditions

In field-based methods, the propagating fields, and therefore the fields' associated mesh, requires a proper termination (truncation). This is not a problem for closed regions such as waveguides or cavities where the PEC boundary provides a proper termination. However, for open radiating problems such as shown in Figure 515 and Figure 516, the mesh would be required to extend to infinity. An artificial absorbing region within the mesh is used to solve this problem. This termination or absorbing region is denoted a boundary condition. Source-based methods do not require a termination of the mesh (boundary condition). A special function (denoted the Green's function) built into the method automatically accounts for the field behaviour at infinity or any point in space.[65]

Solvable Model Size

Field-based methods are generally more limited in terms of the size, specifically the electrical size (in wavelengths), of models they can solve. This is because a growing model size implies a larger **volume** of mesh elements to mesh and solve. For source-based methods it is only a larger **surface** area of mesh elements. It is assumed, however, that acceleration techniques for the source-based method is employed such as the multilevel fast multipole method. In addition it must be noted that increasing usage of GPU acceleration is increasing the solvable sizes of field-based models.

65. Computational Electromagnetics for RF and Microwave Engineering, Second Edition, David B. Davidson, p. 14

5.3 Full Wave and Asymptotic Solution Methods

The Solver includes multiple frequency and time domain solution methods. True hybridisation of some of these methods enables efficient analysis of a broad spectrum of electromagnetic problems. You can also use more than one solver method for cross-validation purposes.

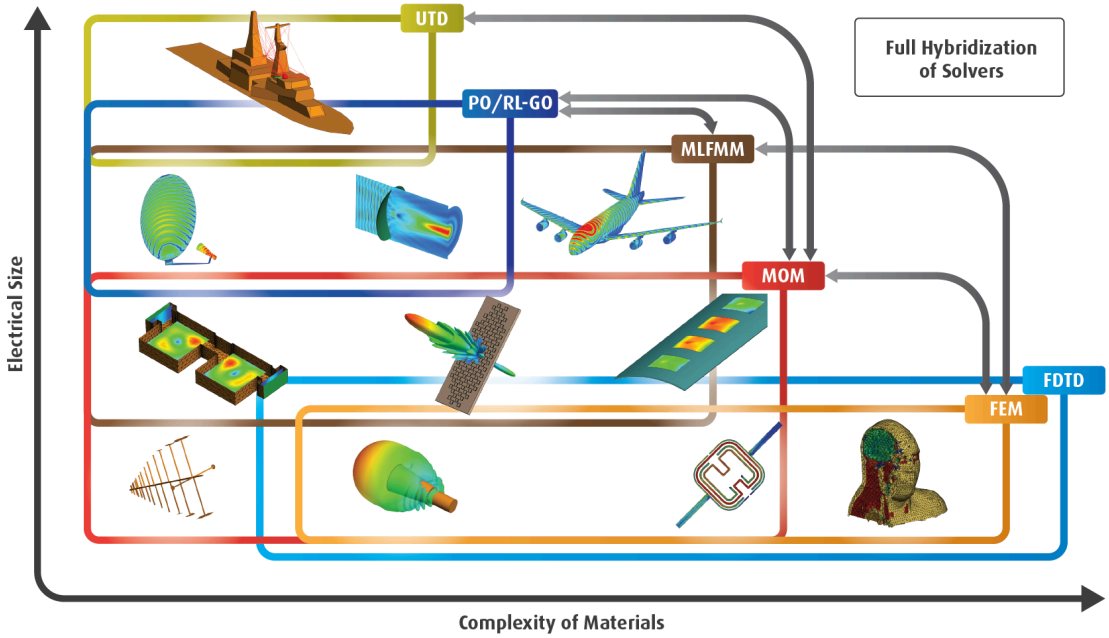


Figure 517: Illustration of the numerical analysis techniques in Feko.

The following solution methods are supported:

- Full wave frequency domain solution methods:
  - MoM (method of moments)
  - FEM (finite element method)
  - MLFMM (multilevel fast multipole method)
- Full wave time domain solution methods:
  - FDTD (finite difference time domain)
- Asymptotic solution methods:
  - PO (physical optics)
  - LE-PO (large element physical optics)
  - RL-GO (ray launching geometrical optics)
  - UTD (uniform theory of diffraction)

5.3.1 Full Wave Solutions

Full wave solutions rigorously solve Maxwell's equations without making any assumptions regarding the nature of the electromagnetic problem. The solution can be either in the frequency or the time domain.

Introduction to the Method of Moments

The MoM is the default solver in Feko. A simple electrostatic example is used to convey the basics of the solver.

The Charge Distribution of a Straight Wire at a Constant Electric Potential of 1 V.

The basic Yagi-Uda antenna shown in Figure 513 consists of a few straight wires. Consider the solution of the charge distribution of a single straight wire of length *l* and diameter 2*a* shown in Figure 518.

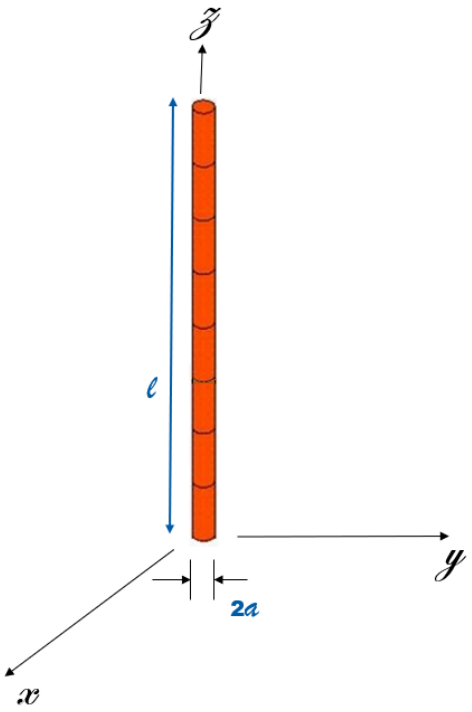


Figure 518: A segmented straight wire charged to a constant potential.

According to [66], a linear electric charge distribution $\rho(\mathbf{r}')$ will create an electric potential $V(\mathbf{r})$ as follows:

$$V(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0}\int_{\text{sources (charge)}} \frac{\rho(\mathbf{r}')}{R} dl' \qquad (42)$$

where $\mathbf{r}'$ represents the source coordinates and $\mathbf{r}$ denotes the observation coordinates, dl' is the path of integration and R is the distance from any point on the source to the observation point which can also be written as

66. Advanced Engineering Electromagnetics, Second Edition, Constantine A. Balanis, p. 680

$$|r-r'|=\sqrt{(x-x')^2+(y-y')^2+(z-z')^2} \tag{43}$$

Note: Equation 42 is valid on the wire and in free space. This is the so-called "boundary condition" for this particular problem.

Even though the charge distribution on arbitrarily shaped objects are not generally known, the straight wire example is useful for an introduction to the MoM.

Assume the wire is charged to a constant electric potential of 1 V. For convenience, the wire is oriented parallel to the Z axis. To solve Equation 42 on a computer, the wire is divided into smaller segments and the charge distribution can be approximated as follows:

$$\rho(z')=\sum_{n=1}^{N} a_n g_n(z') \tag{44}$$

The functions, $g_n(z')$, often referred to as basis functions, are chosen to accurately model the unknown quantity (here the charge on a wire segment) as well as for computational efficiency. For simplicity, constant functions over each segment are assumed. More specifically, each $g_n(z')$ function is equal to 1 over one segment only, and zero elsewhere. The assumption of a constant function implies that the segment length should be short enough for this assumption to hold.

Note: A rule of thumb is to make segments $\frac{1}{10}$th of a wavelength.

Therefore Equation 42 can be approximated as follows:

$$4\pi\varepsilon_0=\sum_{n=1}^{N} a_n \int_0^l \frac{g_n(z')}{\sqrt{(z-z')^2+a^2}}\,dz' \tag{45}$$

As shown in Figure 513, the wire is divided into N uniform segments where each segment is of length $\Delta=\frac{l}{N}$.

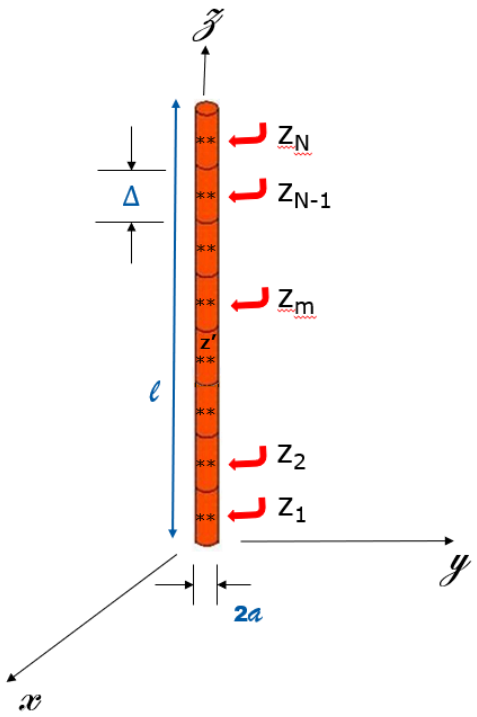


Figure 519: A segmented straight wire charged to a constant potential.

Since Equation 42 is valid everywhere, z can be chosen to be located at fixed points, z_m, on the surface of the wire segments with radii, a. This choice simplifies Equation 45 to only a function of z', allowing the calculation of the integral. Furthermore, since the wire was divided into N segments, Equation 45 can be written as one equation with N unknowns (a_n) as follows:

$$4\pi\varepsilon_0 = a_1\int_0^{\Delta}\frac{g_1(z')}{R(z_m,z')}dz' + a_2\int_{\Delta}^{2\Delta}\frac{g_2(z')}{R(z_m,z')}dz' + \ldots$$
$$+ a_n\int_{(n-1)\Delta}^{n\Delta}\frac{g_n(z')}{R(z_m,z')}dz' + a_N\int_{(N-1)\Delta}^{l}\frac{g_N(z')}{R(z_m,z')}dz' \qquad (46)$$

An equation of N unknowns requires N equations where each equation stands linearly independent from each other. These N equations can be constructed by selecting the observation points z_m in the centre of each segment of length Δ as shown in Figure 519.

> **Note:** The selection of observation points is denoted "testing" or "sampling" $V(z, \rho = a)$ and the method is referred to as "point-matching" or "collocation".

Performing the selection of points N times reduces Equation 46 to the following:

$$4\pi\varepsilon_0 = a_1\int_0^{\Delta}\frac{g_1(z')}{R(z_1,z')}dz' + \ldots + a_N\int_{(N-1)\Delta}^{l}\frac{g_N(z')}{R(z_1,z')}dz'$$
$$\vdots \qquad (47)$$
$$4\pi\varepsilon_0 = a_1\int_0^{\Delta}\frac{g_1(z')}{R(z_N,z')}dz' + \ldots + a_N\int_{(N-1)\Delta}^{l}\frac{g_N(z')}{R(z_n,z')}dz'$$

Equation 47 can be more readily written in matrix form as:

$$[V_m] = [Z_{mn}][I_n] \qquad (48)$$

In Equation 48 each Z_{mn} term can be written as:

$$Z_{mn} = \int_{(n-1)\Delta}^{n\Delta}\frac{1}{(z_m - z')^2 + a^2}dz' \qquad (49)$$

In addition, we can write the remaining two terms:

$$[I_n] = [a_n] \qquad (50)$$

$$[V_m] = [4\pi\varepsilon 0] \qquad (51)$$

The V_m matrix consists of 1 row and N columns and all entries are equal to $4\pi\varepsilon_0$. The a_n values are the unknown coefficients for the charge distribution. To solve Equation 48, the matrix requires inversion where

$$[I_n] = [Z_{mn}]^{-1}[V_m] \qquad (52)$$

Note: A well-known and computationally cheaper inversion procedure, LU decomposition, is followed. The matrix is factored into an upper and lower triangular matrix. Then a process similar to Gaussian elimination is followed to solve the matrices.

Figure 520 shows the line charge density for a wire of length 1 m discretized into 50 segments.

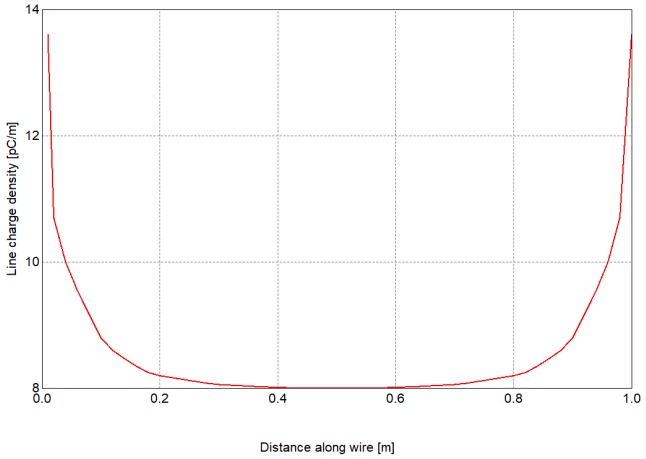


Figure 520: Line charge density of a straight wire charged to a potential of 1 V.

For more complex problems, the integrals cannot be reduced to approximations such as those made here.

The MoM for Full-Wave Solutions

For general open-radiating problems for scatterers of arbitrary shape, a procedure similar to the solution of the charge on the straight wire is followed.

This procedure can be summarized as follows:

1. Specify the relevant integral equation.
2. Apply boundary conditions to manipulate the integral equations into a solvable form.
3. Discretize the unknowns on the scatterer - in this section, we will work with currents.
4. Test the integral equation to create the same number of equations as the number of unknowns.
5. Solve the matrix equation to obtain current coefficients.
6. Sum (integrate) the vector currents to obtain output such as far fields and impedance.

Specifying the Relevant Integral Equation

Specify the integral equation by decomposing the fields into two parts.

Incident and Scattered Fields

Basic laws of physics dictate when an electromagnetic field encounters an object, currents are excited on the object. These currents will subsequently re-radiate. This behaviour is referred to as "electromagnetic scattering".

Maxwell's equations are linear equations which allows them to be decomposed into the sum (superposition) of the "incident" field and "scattered" field. The total field can, therefore, be written as:

$$\mathbf{E}_{tot} = \mathbf{E}_{inc} + \mathbf{E}_{scat} \tag{53}$$

The incident field is typically a plane wave or it could be a voltage source. The incident field is that field that exists in the absence of the conducting body.

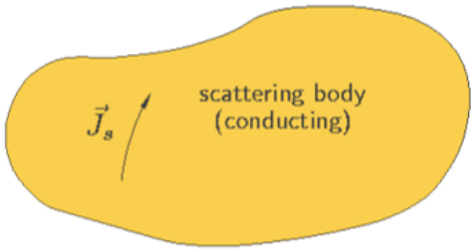


Figure 521: An arbitrary shaped conducting body.

Finding the Incident Field

In RCS applications, the incident field is a plane wave. For example, a plane wave incident from the negative X axis with the electric field z-polarized gives the incident field as:

$$\mathbf{E}_{inc} = e^{-jkx}\hat{z} \tag{54}$$

In antenna problems, the incident field, also denoted the "excitation," is usually a voltage source. A simple form of excitation is the "delta-gap" feed. For an impressed voltage *V* at the terminals of an antenna over a gap of length δ the incident field can be written as:

$$\mathbf{E}_{inc} = \pm\frac{V}{\partial} \tag{55}$$

If this feed is applied to a wire, the length of the gap is typically the length of a wire segment. Other types of incident fields are magnetic frills and elementary Hertzian dipoles.

Finding the Scattered Fields

To find the scattered fields an integral equation is applied to the surface currents. This is written in a simple notation as follows: $\mathscr{L}\{\mathbf{J}_{scat}\}$ where $\mathscr{L}$ represents the integral operator and $\{\mathbf{J}_{scat}\}$ are the unknown currents to be found.

Applying Boundary Conditions

Boundary conditions refer to already known properties of the physics of the problem. These help to derive the solvable integral equations.

Boundary conditions differ depending on the problem to be solved. A dielectric body would have different boundary conditions compared to a PEC body. For the arbitrary shaped PEC body shown in Figure 521 the boundary condition states that the electric field tangential to the surface is zero all over the surface. In terms of the incident and scattered fields (Equation 53) we can then write:

$$\mathbf{E}_{scat,\tan} = -\mathbf{E}_{inc,\tan} \tag{56}$$

This equation is also denoted the electric field integral equation (EFIE).

It was previously shown that an integral operation applied to the surface currents leads to the scattered fields. Therefore we can write in simple notation:

$$\mathscr{L}\left\{\mathbf{J}_{scat}\right\}_{\tan} = -\mathbf{E}_{inc,\tan} \tag{57}$$

where $\mathscr{L}$ represents the integral operator and $\left\{\mathbf{J}_{scat}\right\}$ are the unknown currents to be found.

Discretizing the Currents

Discretizing the object or solution space is commonly referred to as meshing. It is a necessary step to solve the integral equations.

Similar to the procedure followed to solve the charge distribution on the straight wire, the surface of the PEC body is discretized into triangles. Therefore the currents on the triangles are approximated as follows (similar to Equation 44):

$$\mathbf{J}_{scat} = \sum_{n=1}^{N} a_n \mathbf{g}_n \tag{58}$$

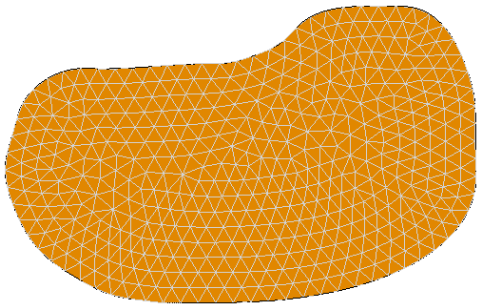


Figure 522: The arbitrary shaped conducting body meshed into triangles.

The equation for the discretized currents can then be substituted into the EFIE (Equation 56) to yield:

$$\sum_{n=1}^{N} a_n \mathscr{L}\left\{\mathbf{g}_n\right\}_{\tan} = -\mathbf{E}_{inc,\tan} \tag{59}$$

In Equation 59, the current coefficients represented by a_n are the only unknown quantity.

Before proceeding to the next step, it is necessary to take a closer look at basis functions.

Basis Functions

Basis functions are elementary functions for the modelling of the unknown quantity on a mesh element.

Categories of Basis Functions

There are two main categories of basis functions:

- entire-domain basis functions
- sub-domain (sub-sectional) basis functions

Entire-domain basis functions are defined over the entire surface of the scatterer - they are non-zero over the entire domain. The formulation of these functions is deemed rather trivial, provided the shape of the scatterer is regular. For most practical applications, the shape of the scatterer is irregular and

the formulation of such basis functions is near impossible. This requires the usage of sub-domain basis functions.

In the application of sub-domain basis functions the entire surface of the scatterer is subdivided into small surfaces. On each subdivided surface a simple function is employed to represent the unknown quantity (such as charge or current). Sub-domain basis functions are non-zero on only a small part of the entire domain.

> **Note:** For FEM and VEP, the **volume** is subdivided and on each volumetric element a simple function is employed to represent the field.

Types of Sub-Domain Basis Functions

The different types of basis functions are distinguished from each other based on their spatial variations. A few well-known ones are as follows:

- constant (also known as pulse or stair-step)
- linear
- polynomial
- piecewise sinusoidal

The Rao-Wilton-Glisson (RWG) element

The MoM in Feko is based on a triangular mesh. Triangular meshes can approximate surfaces much better than for example, rectangular patches. Feko makes use of linear roof-top basis functions introduced by Rao, Wilton and Glisson in 1982. [67] These basis functions enforces current continuity over a common edge of a triangle pair.

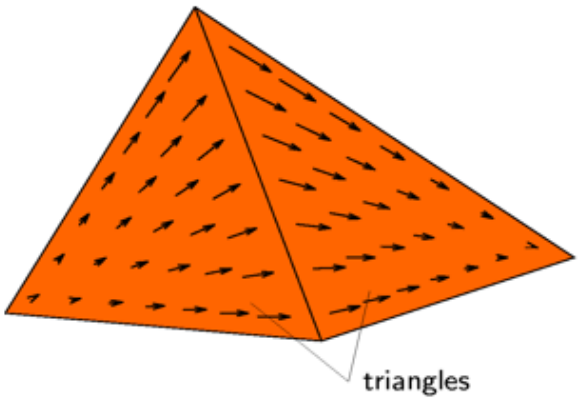


Figure 523: A triangle pair showing the current flow across the common edge as modelled by the RWG basis function.

In Figure 523, only two triangles are shown sharing a common edge. Each triangle also has two other edges. If these edges are connected to triangles, then additional basis functions would be required.

67. S.M. Rao, D.R. Wilton and A.W. Glisson. "Electromagnetic scattering by surfaces of arbitrary shape," IEEE Trans. Antennas Propagation, 30, 409-418, May 1982.

Therefore for a triangle connected on all three sides, a total of three basis functions would be defined. Within the triangle element the total current would then be the sum of these three basis functions.

In Figure 513, the Yagi-Uda was modelled with wire segments. Similar to triangle pairs, linear roof-top basis functions are used across vertices between wire segment pairs.

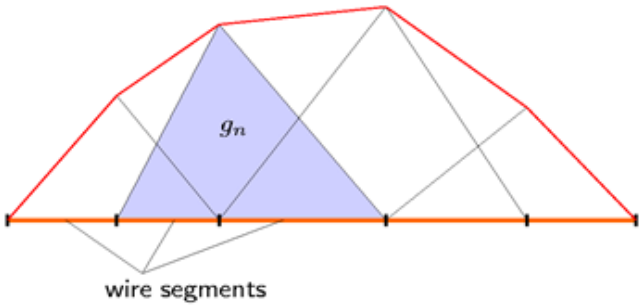


Figure 524: Linear roof-top basis functions for wires modelling current across the wire vertices.

Testing and Solving the Integral Equation

The testing of the integral equation applies the integral equation over each triangle edge to obtain *N* equations with *N* unknowns which can readily be solved on a computer.

For arbitrarily shaped bodies the integral operation is much more complicated than that for a straight wire. It involves several mathematically complex derivations and pitfalls to navigate around of which some are as follows:

- When the integral equations are tested, the so-called self-terms are problematic. The testing of the integral equation at or very near the same position as the unknown leads to a (near) singularity in the matrix equation. In Feko, a computationally efficient methodology is adopted to deal with this problem.
- The testing of the integral equation applies the boundary condition (zero tangential electric field all over the surface of the conductor) at discrete points. Between these points the boundary conditions are not satisfied and this deviation is denoted the "residual". Naturally, this residual introduces deviations from the exact physical solution. One way to minimize the residual is to minimize the average residual all over the structure. For this purpose, a set of vector weighting functions are defined. Different weighting functions were proposed and the implementation in Feko is beyond the scope of this document.

  > **Note:** Minimizing the deviation from the boundary conditions is denoted the "method of weighted residuals" or more commonly, the method of moments.

The testing of Equation 59 results in a square matrix very similar to Equation 48.

This equation can be solved for the current coefficients by using LU decomposition routines.

Integrating the Currents

Summing or integrating the vector currents is the last step in the MoM procedure. This step leads to specific output parameters such as far fields and impedance.

The Free Space Green Function

The free space Green's function is essential to the MoM to allow calculation of fields at arbitrary points in 3D space. Without going into the finer technical details of the equation, it can be stated that the Green function is contained inside the integral operator $\mathscr{L}$ operating on the surface currents,

$$\mathscr{L}\{\mathbf{J}_{scat}\}_{\tan} = -\mathbf{E}_{inc,\tan} \tag{60}$$

Consider an infinitesimally small current element *J* in free space at a point *r*' radiating an electric field *E* and a magnetic field *H*.

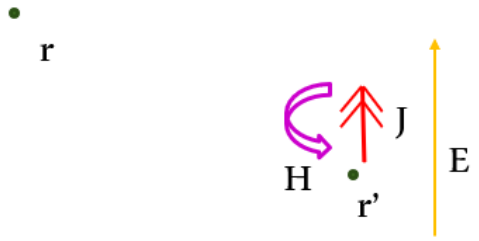


Figure 525: An infinitesimally small current element in free space at a point r' radiating an E and H field. Its potential at the point r is given by the Green Function.

The Green's function (Equation 61) gives the spatial response to a spatially impulsive current source. This means that for the current element (source) located at the point **r**', the Green's function gives the potential of this source at the point **r**, or any required point in 3D space.

$$G(r,r') = \frac{e^{-j\beta|r-r'|}}{|r-r'|} \tag{61}$$

with

$$|r-r'| = \sqrt{(x-x')^2+(y-y')^2+(z-z')^2} \tag{62}$$

the distance from the source to the field point. When there are multiple of these sources distributed in space, such as over the arbitrary PEC body, the response at the point *r* is given by summing all the sources (integration over all the sources).[68]

68. Computational Electromagnetics for RF and Microwave Engineering, Second Edition, David B. Davidson, p.265

MoM Computational Resources Scaling

The usage of a dense matrix in the MoM implies a limit to the size of the problem that can be solved. The limit is determined by the available computational resources.

Although the MoM efficiently discretizes the model by only requiring the bounding surface to be meshed, the method uses a dense matrix. As a result, the memory scaling is proportional to N^2 and CPU-time to N^3, where N is the number of unknowns.

This is best illustrated by comparing the asymptotic behaviour of the memory and CPU-time scaling of a model solved at one frequency and at double the frequency.

- At the higher frequency, the triangle patches are required to have half the edge lengths. The number of elements increases by a factor of four. The number of unknowns is proportional to the number of elements and the memory required to solve the problem increases by a factor of 16.
- When solving the problem at double the frequency, the simulation time increases by a factor of 64.

As the frequency and structure size increases, special techniques such as the multilevel fast multipole method, higher order basis functions and asymptotic techniques are required to obtain a solution efficiently.

Alternately, higher order basis functions for triangular elements could be used. Higher order basis functions have more unknowns per element, but they allow larger mesh elements to be used. The net result is that less memory is required.

> **Note:** Use larger triangular elements provided the larger triangles describe the model geometry accurately.

Higher order basis function elements can be represented with curvilinear triangle patches that allow second order descriptions of the triangular patch boundaries. The curvilinear elements allow a further reduction in the number of elements required for an accurate representation of the model.

Other Methods Based on the MoM

Specific methods, which can also be categorized as MoM methods, are tailor-made for solving dielectric bodies.

Two methods specifically designed for solving dielectric bodies are as follows:

1. The surface equivalence principle (SEP) which is the default method for solving dielectric materials in the MoM.
2. The volume equivalence principle (VEP) is an extension to the MoM for modelling finite dielectric objects using a volume mesh.

Surface Equivalence Principle (SEP)

The surface equivalence principle (SEP) introduces equivalent electric and magnetic currents on the surface of a closed dielectric body. The surface of such bodies can be arbitrarily shaped and is discretised using triangles.

The surface equivalence principle (SEP) is the default method for solving dielectric materials when using the method of moments. The SEP introduces equivalent electric and magnetic currents on the boundary of the dielectric body (opposed to only using only equivalent electric currents on a perfectly electric conducting body).

The SEP can model homogeneous dielectric bodies efficiently, but becomes inefficient when the material is inhomogeneous, as is the case when modelling biological tissue or multiple thin layers of dielectric.

Feko includes other solution methods for efficient treatment of inhomogeneous dielectric structures. These include volume equivalence principle, the finite element method and support for planar multi-layered media.

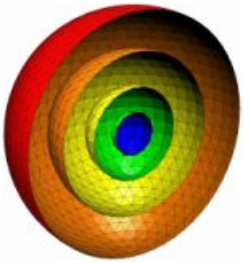


Figure 526: An example of a layered dielectric body consisting of closed bounding surfaces.

Volume Equivalence Principle (VEP)

The volume equivalence principle (VEP) is an extension to the method of moments (MoM) for modelling finite dielectric objects using a volume mesh (tetrahedra and cuboidal[69] elements).

The volume equivalence principle (VEP) is not used by default and would only be used when the solution requires an alternative to the default surface equivalence principle (SEP). More basis functions are usually required compared to the surface equivalence principle (SEP), but neighbouring cuboids or tetrahedra may have differing electric and magnetic properties.

The VEP is associated with a volume mesh, and general usability is inhibited by the order $O(N^{2..3})$ memory and CPU-time scaling with the number of unknowns N. There are however cases where the VEP is advantageous over the SEP or the FEM/MoM:

- The formulation is stable at low frequencies.
- The formulation is stable when modelling dielectrics with very high permittivity (high dielectric constant).
- It displays good stability and convergence properties for an iterative solution with the MLFMM.
- It is well-suited to inhomogeneous and thin dielectric bodies.

69. Only supported in EDITFEKO

Note: Tetrahedral VEP is not supported together with other dielectric modelling methods (SEP, FEM, VEP with cuboids, special Green's functions) or periodic boundary conditions (PBC).

Additional Features and Extensions for the Method of Moments

Numerous features and optimised electromagnetic (EM) analysis options for the method of moments (MoM) are available.

Planar Green's Functions for Multi-Layered Media

Multi-layered dielectric media can be modelled with Green's functions. Example of structures that are efficiently modelled with the planar multi-layer substrate method include printed circuit boards (applications using microstrip and stripline structures).

The special Green's function formulation implements 2D infinite planes with a finite thickness to model each layer of the dielectric and optional conducting layers. Conducting surfaces and wires inside the dielectric layers must be discretised, but not the dielectric layers themselves. Although the layers are infinite in extent, it is often a good approximation to practical antennas as planar structures.

Metallic surfaces and wires can be arbitrarily oriented in the media and can cross multiple layers if they are discretised on layer boundaries (vertices have to coincide with the layer boundaries).

The planar multi-layered media can be defined with in a bounded region, allowing models with many fine layers to be modelled efficiently without requiring the same layers throughout the entire model. Limiting the Green's function to a specific dielectric region would require a region to be defined to encapsulate the layered media, increasing the number of unknowns, but it can be much more efficient when compared to the alternative of modelling each layer individually.

Good performance is achieved since calculations using Green's functions are accelerated by using interpolation tables.

Numerical Green's Function

The numerical Green's function can be used for problems containing static and dynamic parts, allowing re-use of the static part of the solution in subsequent simulations to improve overall performance.

The numerical Green's function, also sometimes referred to as macro basis functions, allows users to group a subsection of the model as being "static" and the rest of the model is then considered "dynamic". During the solution phase, the static part is grouped together during the matrix fill phases and the matrix elements are stored to file. Any subsequent simulations use the static part instead of re-calculating it, allowing the simulation to complete faster. An example where this is useful would be when a rotor blade needs to be simulated in multiple positions or different positions of a source on a large platform. The structure supporting the rotor blade (helicopter, aeroplane, wind turbine tower) could be huge, much larger than the rotor blade itself, and reusing the calculations of the static part can be a tremendous saving.

All the segmentation rules still apply and the mesh elements and vertices need to align where the static and the dynamic parts touch. Care should be taken to ensure that the meshing is done that allows all the variations of interest in the dynamic part.

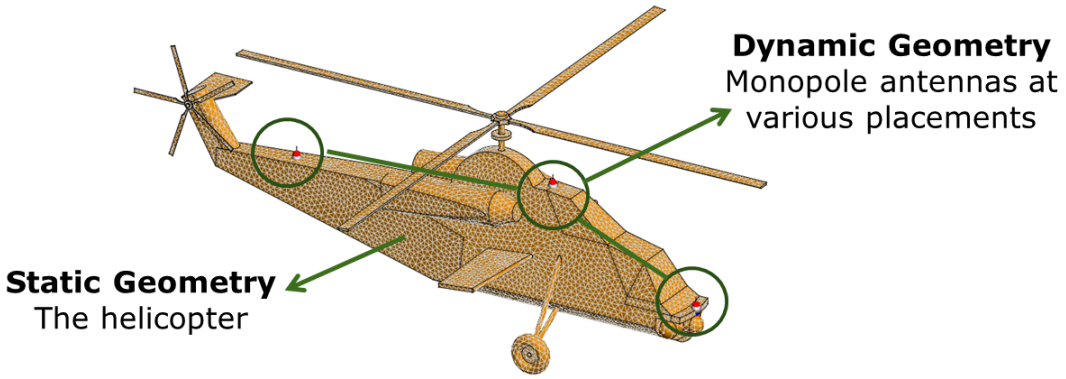


Figure 527: Example of a large platform (static) with source (dynamic) locations that need to be investigated.

Thin Dielectric Sheets

Multiple layers of thin dielectric sheets and anisotropic sheets can be analysed using a single meshed surface. Typical applications are the analysis of radome covered antennas.

Dielectric Coated Wires

The effect of dielectric-coated wires can be modelled using an equivalent impedance or as an equivalent volume current.

Feko implements two methods for modelling dielectric and magnetic coatings on wires:

- Popovic's formulation modifies the radius of the metallic wire core to change the capacitive loading on the wire, while simultaneously adding a corresponding inductive load. The method requires that the loss tangent of the layer be identical to the loss tangent of the surrounding medium.
- Pure dielectric layers (for example, the relative permeability of the layer equal to the surrounding medium) should be modelled with the equivalence theorem where the effect of the dielectric layering is accounted for by a volume polarisation current.

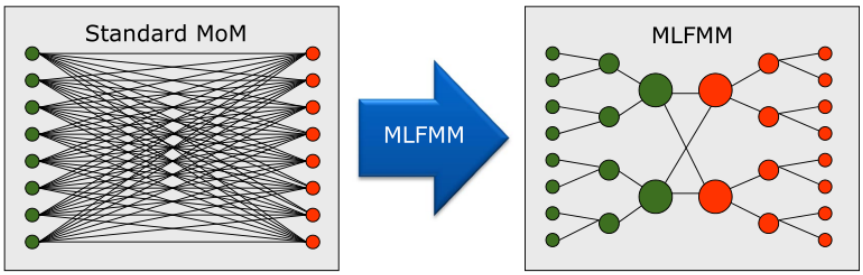
> **Note:** Layers must be non-magnetic.

Real Ground

A real ground can be modelled with the reflection coefficient approximation or the exact Sommerfeld formulation. Real grounds are used to model the effect of non-ideal grounds such as the earth (wet or dry ground).

Multilevel Fast Multipole Method (MLFMM)

The multilevel fast multipole method (MLFMM) is an alternative formulation of the technology behind the method of moments (MoM) and applies to much larger structures (in terms of the wavelength) than the MoM, making full-wave current-based solutions of electrically large structures a possibility.



The MLFMM can be applied to large models that were previously treated with the MoM without having to change the mesh.

The agreement between the MoM and MLFMM is that basis functions model the interaction between all triangles. The MLFMM differs from the MoM in that it groups basis functions and computes the interaction between groups of basis functions, rather than between individual basis functions and in so doing reduces the number of interactions that need to be calculated. The MLFMM never really calculates the matrix that is used during the method of moments calculation and as a result there is no direct solution for the MLFMM. An iterative solution utilising a fast matrix-vector product is used during the MLFMM solution phase.

Feko employs a boxing algorithm that encloses the entire computational space in a single box at the highest level, dividing this box in three dimensions into a maximum of eight child cubes and repeating the process iteratively until the side length of each child cube is approximately a quarter wavelength at the lowest level. Only populated cubes are stored at each level, forming an efficient tree-like data structure.

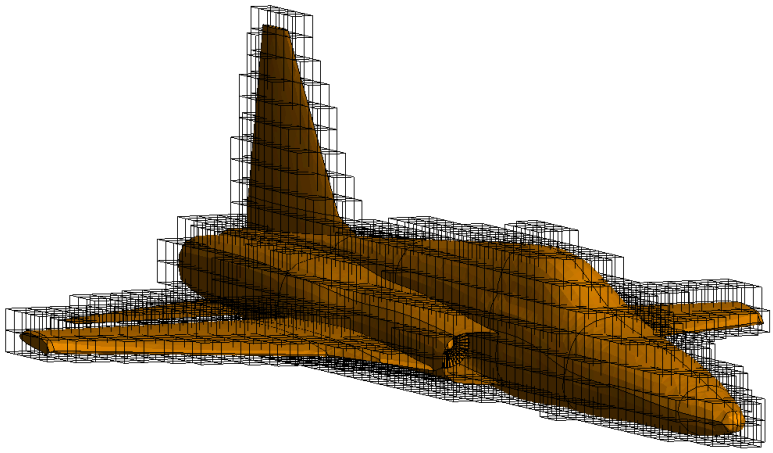


Figure 528: MLFMM boxes at the third level.

In the MoM framework, the MLFMM is implemented through a process of aggregation, translation and disaggregation of the different levels.

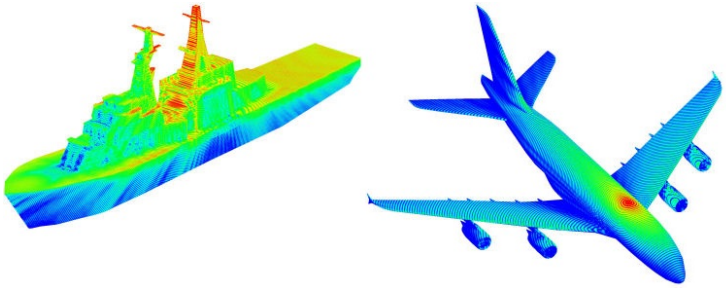


Figure 529: MLFMM analysis of a ship (on the left) and antenna placement modelling on a commercial aircraft (on the right).

Integral Equation Methods (EFIE, MFIE and CFIE)

The relevant integral equation method can be used to solve a model to either obtain faster iterative or higher numerical accuracy when using the MoM or MLFMM.

When solving a structure that consists of perfectly conducting surfaces (PEC), the solution can be accelerated using either the electric field integral equation (EFIE) or the magnetic field integral equation (MFIE).

The EFIE method has a higher numerical accuracy and applicable to open structures, whereas the MFIE method has much faster iterative convergence and applicable to enclosed, perfectly conducting metallic regions.

To obtain both accuracy and faster iterative convergence for the solution using less memory, the combined field integral equation (CFIE) method is used.

Note: The CFIE / MFIE is only applicable to enclosed, perfectly conducting metallic regions.

When specifying the CFIE factor, you are specifying if the solution must be one of the following:

- A pure EFIE solution (CFIE factor = 1)
- A pure MFIE solution (CFIE factor = 0)
- A combination of EFIE and MFIE, which is a CFIE solution (0 < CFIE factor < 1)

Note: The EFIE is the default solution in Feko.

Related tasks
Modifying the Integral Equation Method

Adaptive Cross-Approximation (ACA)

The adaptive cross-approximation (ACA) is a fast method similar to the multilevel fast multipole method (MLFMM) but is also applicable to low-frequency problems or when using a special Green's function.

The adaptive cross-approximation (ACA) approximates the impedance matrix by constructing a sparse H-matrix (only a few selected elements are computed). The ACA is similar to the MLFMM in the sense that they are both used for large models where the method of moments has become too resource intensive, but they are quite different in implementation and applications. Models with many unknowns, where the model is electrically small (less than a wavelength) can be solved with ACA. For electrically large structures (multiple wavelengths) the multilevel fast multipole method is much better suited.

Finite Element Method (FEM)

The finite element method (FEM) is a solution method that employs tetrahedra to mesh arbitrarily shaped volumes accurately where the dielectric properties may vary between neighbouring tetrahedra.

The FEM applies to the modelling of inhomogeneous dielectric bodies. It is also well suited to non-radiating microwave components such as shielded filters.

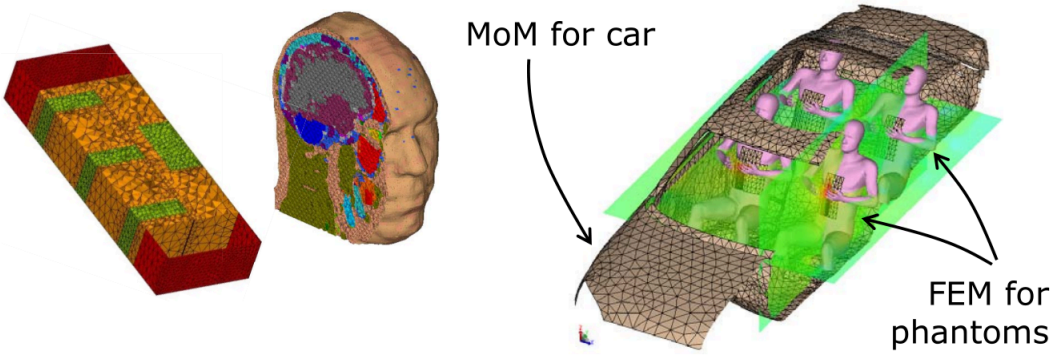


Figure 530: Examples of MoM / FEM hybrid.

For radiating surfaces as well as wires the hybrid FEM / MoM or FEM / MLFMM is invoked and not a pure FEM analysis. The FEM / MoM hybridisation features full coupling between metallic wires and surfaces in the MoM region and heterogeneous dielectric bodies in the FEM region. The MoM part of the solution is calculated first, which results in equivalent magnetic and electric currents that form the radiation

boundary of the FEM region. This hybrid method incorporates the strengths of both the MoM and the FEM.

When a structure is bounded only by PEC surfaces and FEM modal ports, Feko recognises that the problem can be solved by just the FEM (fully sparse matrix solution), resulting in a reduction in memory and runtime. For electrically large problems, the hybridised FEM / MLFMM can be used where the MLFMM solves the MoM part of the FEM / MoM problem efficiently.

Finite Difference Time Domain (FDTD)

The finite difference time domain (FDTD) is a full wave time domain solution method, and Fourier transforms are applied to convert the native time domain results to the frequency domain.

The finite difference time domain (FDTD) solution technique has gained popularity in computational electromagnetics (CEM). Much of this popularity comes from its relatively straightforward formulation, where electric and magnetic fields are computed on two offset rectilinear grids and marched in time. This approach allows for the use of central differencing to approximate Maxwell's equations. It can achieve second-order accuracy using first-order numeric differentiation.

The solution method is best suited to problems that include highly inhomogeneous materials and is, therefore, a popular choice in biomedical applications for the modelling of human phantoms. It is also a highly efficient solution for wideband problems, and is well suited to analyse broadband antennas. A single FDTD simulation with a pulsed excitation can be used to characterise a wideband frequency response of an antenna.

Also, the method lends itself well to various parallelisation techniques, including the use of accelerators such as GPUs to obtain significant speedups.

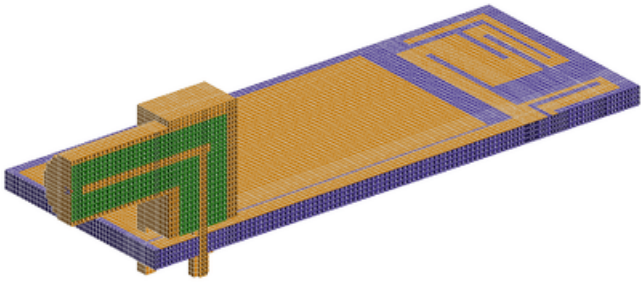


Figure 531: The FDTD voxel mesh of a GSM antenna.

5.3.2 Asymptotic Solutions

Asymptotic solution methods solve Maxwell's equations, but make certain assumptions regarding the nature of the problem. Feko provides various high frequency asymptotic solution methods that assume the frequency of interest is high enough that the structure is much larger than the wavelength.

Ray Launching Geometrical Optics (RL-GO)

The ray launching geometrical optics (RL-GO) is a ray-based method intended for modelling electrically large dielectric and perfect electrically conducting structures in applications such as lens antennas and radar cross section (RCS) analysis.

The RL-GO is formulated for use in instances where electrically very large (> 20λ) metallic or dielectric structures are modelled. RL-GO is inherently well suited to the solution of large scattering problems such as radar cross section (RCS) analysis since the “shooting and bouncing rays” approach are highly efficient for an arbitrary number of multiple reflections.

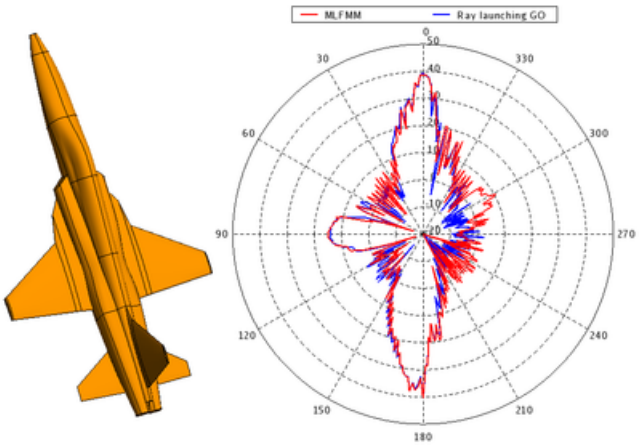


Figure 532: RCS of an aircraft at 1 GHz in the elevation plane: Comparison between MLFMM and RL-GO. RL-GO required 33 times less memory.

Feko integrates the RL-GO method with the current-based MoM, by launching rays from each radiating MoM element. The ray interactions with metallic and dielectric structures are then modelled using Huygens sources placed on each ray contact point (for reflected, refracted and transmitted rays) on the material boundaries. The runtime and memory requirements scale almost perfectly for parallel processing, resulting in multi-core CPUs or cluster computers operating highly efficiently while solving RL-GO problems.

A typical application of the MoM / RL-GO hybrid method is the analysis of dielectric lenses. The source structure (for example a metallic antenna under a lens), may be modelled with the MoM and the large dielectric lens may be modelled with the RL-GO.

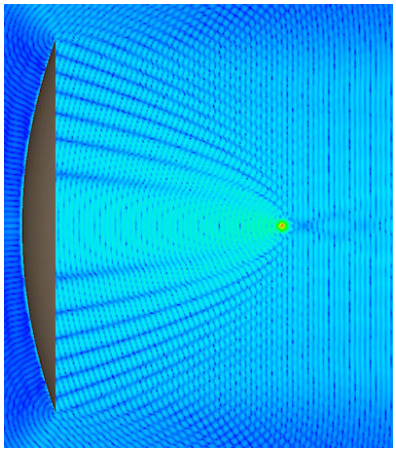


Figure 533: Reflector near field calculated with RL-GO.

Physical Optics (PO)

The PO solution method is an asymptotic high-frequency numerical method of the same nature as the UTD but based on currents and not rays.

The PO solution method is formulated for use in instances where electrically very large metallic or dielectric structures are modelled. Feko hybridises the current-based accurate MoM with PO including bidirectional coupling between the MoM and PO regions. It discretises a PO region, as it would for a MoM solution, making it a simple task to switch between solution methods. In cases where the MoM part of the problem is electrically large, the PO hybridised with the MLFMM provides an efficient solution.

A practical example for PO would be to calculate the effect on the input impedance of a horn antenna (treated with the MoM) when placed near a large structure (treated with the PO).

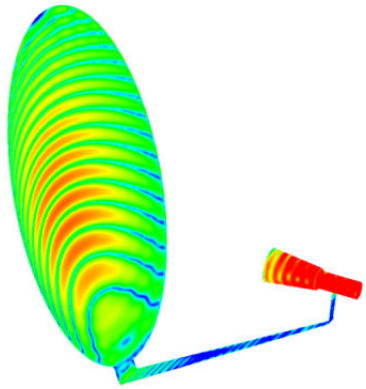


Figure 534: PO modelling of a reflector antenna with MoM modelling of the feed.

Large Element Physical Optics (LE-PO)

The large element physical optics (LE-PO) solution method is similar to the PO method but allows larger elements to be used.

The large element physical optics (LE-PO) is formulated for use in instances where electrically very large structures are modelled. This method should only be used when there are no discontinuities in the incident field (the field incident on the LE-PO face should closely represent a plane wave). LE-PO is similar to PO in that it is an asymptotic high-frequency numerical method of the same nature as the UTD.

The high-frequency large element physical optics method is applicable for large smooth areas when calculating near and far fields.

Uniform Theory of Diffraction (UTD)

The uniform theory of diffraction (UTD) is formulated for modelling electrically extremely large structures. The UTD is an asymptotic high-frequency numerical method similar to the PO.

Users typically attempt a solution with the MoM, and when they realise that the structure is electrically too large to solve with their available resources (platform memory and time), they turn to the MLFMM. If the required resources are still too large, the PO, UTD or ray launching geometrical optics (RL-GO) can be used.

The Solver contains the following two UTD-based solvers:

Uniform theory of diffraction (UTD) with polygons and cylinders

Feko hybridises the current-based accurate MoM with the UTD. Bidirectional coupling between the MoM and UTD is maintained in the solution (through modification of the interaction matrix) to ensure accuracy. Frequency does not affect the memory resources required for solving a structure with UTD, given that only points of reflection from surfaces and diffraction from edges or corners are considered without meshing the structure.

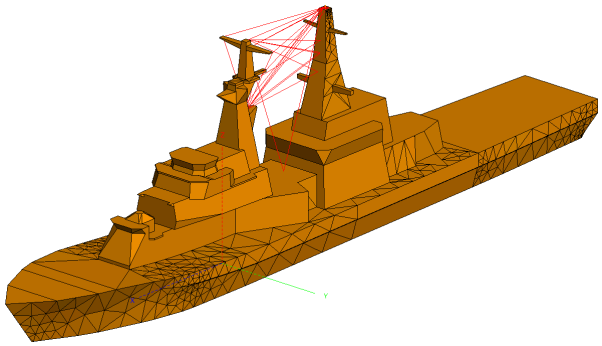


Figure 535: UTD modelling of cross-coupling on the superstructure of a modern naval vessel.

Multiple reflections, edge/wedge and corner diffraction and creeping wave effects on curved surfaces are considered. Insight into the propagation of rays are provided in POSTFEKO during post-processing. Currently, the numerical formulation of the UTD only allows it to be applied to flat polygonal plates with minimum edge lengths in the order of a wavelength, where surface curvature is not considered. A single canonical circular cylinder can be included in the model.

Creeping waves are only considered on the cylinder. The UTD is well suited to the analysis of ships at radar frequencies but less appropriate for analysing complex objects with curved surfaces (such as automobiles).

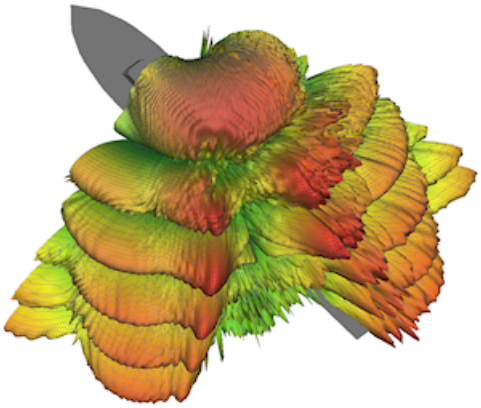


Figure 536: Analysis of the transmission patterns of an X-band radar mounted on a ship.

Faceted uniform theory of diffraction (faceted UTD)

The faceted-UTD solver can be used to calculate fields and radiation patterns for antenna placement applications at high frequencies. It supports impressed sources and a planar triangular surface mesh. The resource requirements are independent of frequency, but depend on the number of mesh elements required to accurately represent the geometry and the number of field observation points. Multiple reflections, diffraction and creeping wave effects on curved surfaces are considered.

The faceted-UTD solver is well suited for antenna placement on electrically large platforms with planar or curved surfaces (such as aircrafts).

5.3.3 Solution Methods per Application

A solution method is selected based on the electrical size of a problem, the geometrical complexity and available computational resources.

●	Solution method is ideally suited to the problem.
○	Solution method could be used, but an alternative method is better suited to the problem.
<empty space>	Indication that the respective solution method should not be used.

Table 60: The electromagnetic solution methods suited to the various applications.

	Geometrically complex			**Electrically large**				
	MoM	FEM, FEM/MoM	FDTD	MLFMM	FEM/MLFMM	PO, MoM/PO, MLFMM/PO	RL-GO, MoM/RL-GO	UTD, MoM/UTD
Wire antennas	●		○	○				
Microstrip antennas	●	●	●	○	○			
Aperture antennas	●	○	○	○	○			
Reflector antennas	○	○	○	●	○	●	●	
Windscreen antennas	●			●				
Conformal antennas	●	●		●	●			
Broadband antennas	●	●	●	●	●			
Array antennas	●	●		●	●			
Lens antennas	○	○	○	○	○		●	
Radomes	○	○	○	●	●		●	
Antenna placement (radiation pattern)	●	○	○	●	●	●	●	●
Antenna placement (coupling)	●	○	○	●	●	○	○	○
Biomedical	○	●	●	○	●			
RADHAZ zones	●	●	○	●	●	●	●	○

	Geometrically complex			Electrically large				
	MoM	FEM, FEM/MoM	FDTD	MLFMM	FEM/MLFMM	PO, MoM/PO, MLFMM/PO	RL-GO, MoM/RL-GO	UTD, MoM/UTD
Periodic structures FSS, metamaterials	●	○		○	○			
Scattering with plane wave source (RCS)	●	○	○	●	○	●	●	
Scattering with localised source	●	○		●	○	●	●	●
EMC/EMI shielding and coupling	●	●	○	●	○			
Propagation environment	○	○	○	○	○	○	○	●
Cable bundle coupling	●			●				
Waveguide components	●	●			○			
Connectors	●	●	○					
Microstrip circuits	●	●	●	○	○			

5.3.4 Supported Solution Method and Technique Combinations

When selecting a solution method or technique in Feko, it is important to know which method and technique combinations may be used in a model, else CADFEKO and the Solver will give an error.

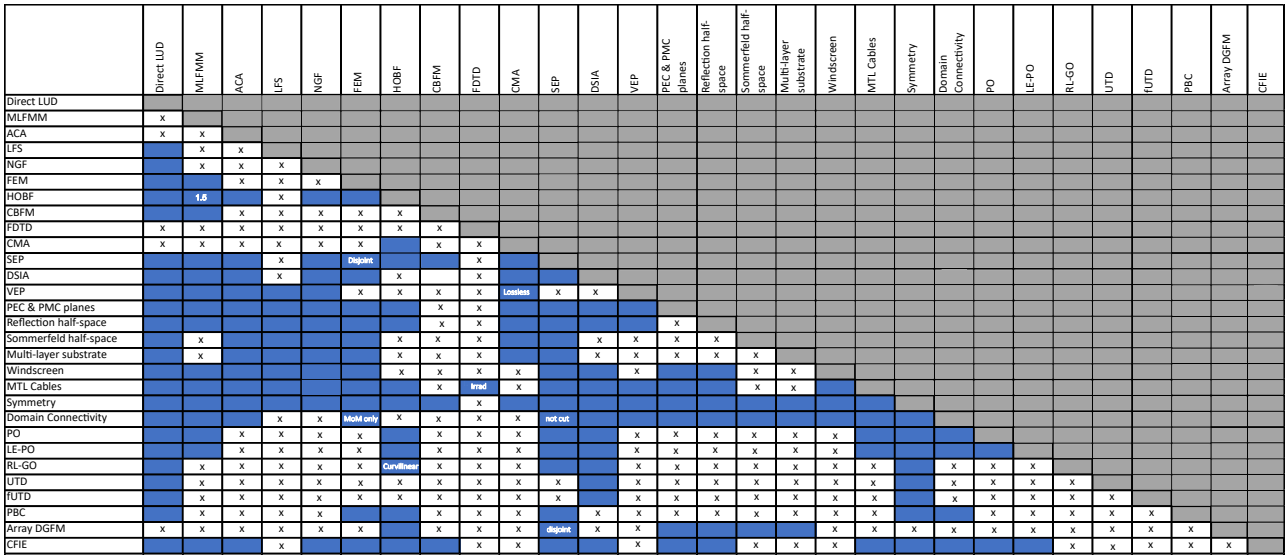


Figure 537: Summary of solution method and technique combinations that may used in a model.

5.3.5 Cable Coupling

Model complex cable-bundle networks using full-wave simulations.

When modelling cables, two methods are available:

- multiconductor transmission line (MTL)
- combined method of moments / multiconductor transmission line (combined MoM / multiconductor transmission line)

Results from the above methods may differ due to the effect of the additional combined MoM / MTL termination segments. Make provision for the non-idealities of the combined MoM / MTL method in the MTL model by adding equivalent parasitic circuit elements to the MTL cable model.

Multiconductor Transmission Line (MTL)

An arbitrary cable (shielded or unshielded) can be solved using the multiconductor transmission line (MTL) solution method hybridised with the MoM, MLFMM or FDTD (only irradiation).

MTL theory is used to model the cable problem, while the MoM or MLFMM is used for the solution of the external fields and currents that couple to or from the cable harness.

MTL theory is limited in application to situations where cables run close to a ground plane. The cable path should be within $\frac{\lambda}{5}$ (ideally within $\frac{\lambda}{10}$) of the conducting surface. Cables that are solved with the

MTL may not be connected directly to MoM geometry. The connection between the cable circuit and ground is modelled using non-radiating networks.

Method of Moments (MoM), Only For Shielded Cables

An arbitrary shielded cable can be solved using the combined method of moments and multiconductor transmission line (MoM / MTL) solution method.

Any arbitrary cable path can be defined, and there is no restriction on the cable path's proximity to a ground plane.

For the combined MoM / MTL method, the outer shield is replaced by physical MoM segments.

If a cable is not connected directly to MoM geometry, additional segments are added from the shield to ground. The inner cable cross-section geometry is solved with the MTL. The shield inner current is transformed via the shield transfer impedance, resulting in distributed voltage sources that are applied to the shield segments that are included internally in the MoM part of the solution.

Cables can be connected to one another using a shielded enclosure that has a physical size and include radiation and grounding effects between the cable and the antenna or circuit board.

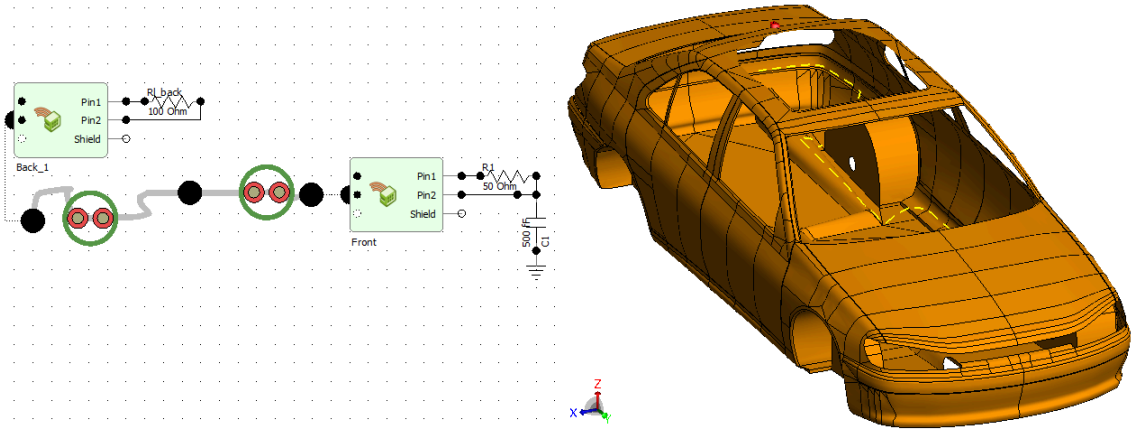


Figure 538: A generic car with a cable harness.

5.3.6 Arrays and Infinite Periodic Structures

Infinite and finite periodic structures are efficiently modelled using special features available in Feko.

Arrays and finite periodic structures can be modelled using any of the full wave solution methods in Feko, but this can lead to long simulation times and high resource requirements, making these simulations impractical.

Finite Antenna Arrays

Finite antenna arrays where the elements are identical and the feed magnitude is similar can be modelled using the domain Green's function method (DGFM). The method is based on perturbation of the results from a single element and requires the elements to identical and similar (not identical) in terms of currents flowing on the elements.

Large finite arrays can also be approximated by infinite arrays by calculating the currents for the infinite structure, but then only taking a finite number of elements into account when calculating the far and near fields. The larger the array, the more accurate the approximations, since the error is the greatest at the edges of the array. These edge effects can also be taken into account (approximately) by modelling a finite array and using the currents of different elements (centre, edge, corner) to reconstruct the large array using radiating antenna sources.

Infinite Arrays

The periodic boundary conditions in Feko allow infinite 1D and 2D arrays to be modelled very efficiently. Users can either define the excitation or the phase difference between the elements.

Frequency Selective Surfaces

Frequency selective surfaces are also infinite structures and their properties can be investigated using the periodic boundary conditions method. Once their properties are known, approximations to these frequency selective surfaces are used to model complex, large, but finite, structures.

Domain Green's Function Method (DGFM) for Large Finite Arrays

The domain Green's function method (DGFM) is a perturbation approach where the mutual coupling between array elements is taken into account when calculating the Green’s function for each element. The current distribution on the entire array geometry is obtained by solving each element independently, leading to a significant saving in both runtime and memory usage.

The method also takes into account the edge effects attributed to the finite size of the array, complex excitations with non-linear phase shift and is not limited to periodic array configurations.

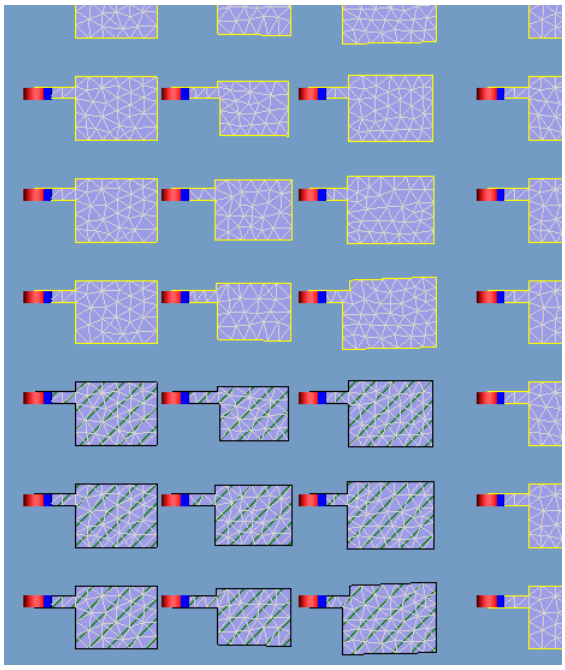


Figure 539: An array of non-identical patch antennas to be solved with the DGFM.

Periodic Boundary Conditions (PBC)

Periodic boundary conditions allow for analysing large, uniformly spaced, repetitive linear and planar structures, for example, frequency selective surfaces (FSS).

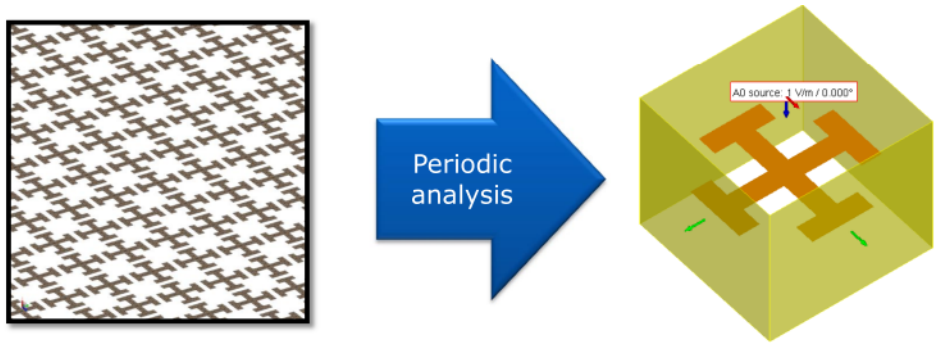


Figure 540: Example of applying PBC to a frequency selective surface, for example a Jerusalem cross.

5.3.7 General Non-Radiating Networks

Complex feed networks can be simplified by including them as a circuit representation using general network blocks.

General networks (defined using network parameter matrices) can be used to model a feed network. These non-radiating networks may be interconnected (cascaded) and excited or loaded directly at the ports.

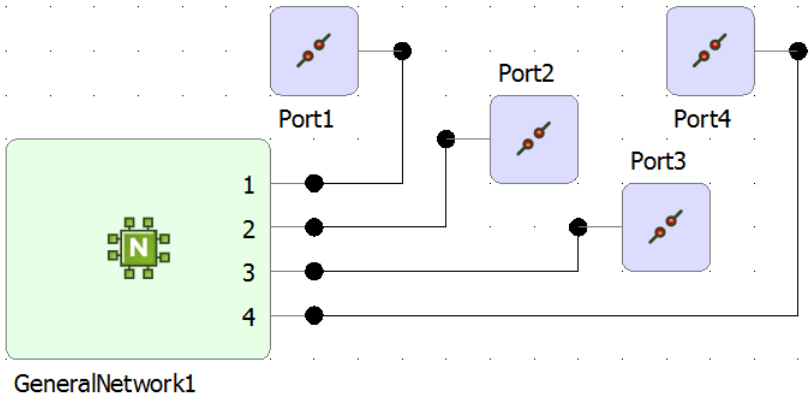


Figure 541: A four-port general non-radiating network.

The voltages and currents at the ports of these ideal representations of networks may interact with currents and voltages on parts of the model that are solved using other solution methods, though no radiation-based coupling is taken into account.

5.3.8 Windscreen Modelling

The windscreen antenna solution method reduces the computational requirements by meshing only metallic elements while analysing the behaviour of the integrated windscreen antennas within their operating environment. The analysis can take into account the physical features of windscreen antennas and their surroundings.

The following physical features are taken into account when analysing windscreen antennas:

- Finite sized windscreens
- Arbitrarily curved (no extreme curvature) windscreens
- Multiple dielectric windscreen layers (glass, plastic and other dielectric materials)
- Multiple windscreens in a vehicle (multiple glass definitions supported)
- The vehicle body
- The presence of real ground

The analysis is based on the MoM and can be used in conjunction with the multilevel fast multipole method (MLFMM). It is an efficient approach due to only including the vehicle body and metallic antenna elements in the MoM mesh. The windscreen layers are not discretised.

Numerous electromagnetic characteristics of the windscreen antenna can be computed, including:

- Current distribution on the antenna and vehicle
- Input impedance bandwidth and scattering parameters
- Near field distributions and far field radiation patterns

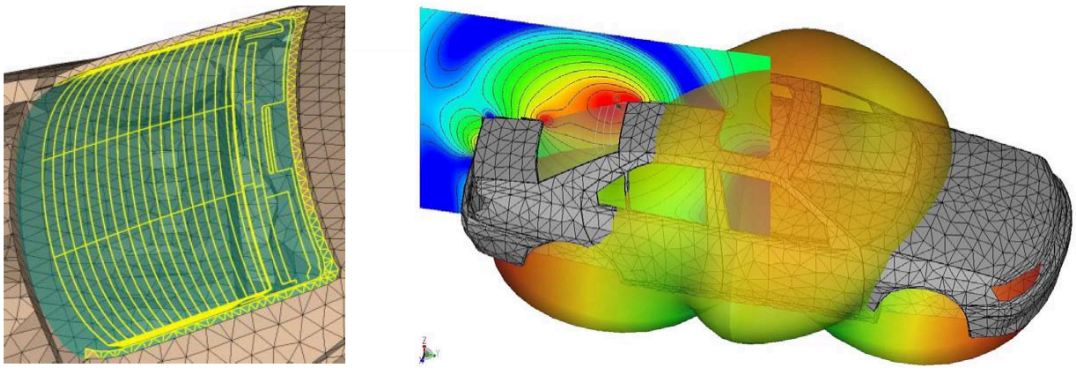


Figure 542: An example of a windscreen antenna (on the left) and an automobile with a far field and near field result (on the right).

5.3.9 Symmetry Planes

Geometric symmetry, electric symmetry and magnetic planes of symmetry in a model can be exploited to reduce runtime and memory requirements.

Symmetry in a model applies to the method of moments (MoM) and all hybrid techniques where the MoM is involved, but not in conjunction with the multilevel fast multipole method (MLFMM).

A symmetric model without geometric symmetry defined is not guaranteed to have a symmetric mesh. Such a setup leads to non-symmetric current distributions on the structure.

Geometric Symmetry

The structure must be symmetric concerning the symmetry plane, while the sources may be arbitrarily located.

Electric Symmetry

To define an electric symmetry plane, the following must be true:

- The model must be geometric symmetry at the plane.
- The electric current density must be anti-symmetric.
- The magnetic current density must be symmetric.

For example, a physical interpretation of an electric symmetry plane is a plane which can be replaced by a perfect electric conductor (PEC) wall without changing the field distribution. The tangential component of the electric field and the normal component of the magnetic field are zero at such a plane.

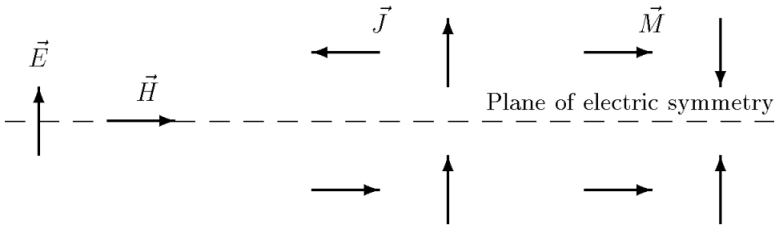


Figure 543: Electric symmetry plane

Magnetic Symmetry

To define a magnetic symmetry plane, the following must be true:

- The model must be geometric symmetry at the plane.
- The electric current density must be symmetric.
- The magnetic current density must be anti-symmetric.

For example, a physical interpretation of a magnetic symmetry plane is a plane which can be replaced by a perfect magnetic conductor (PMC) wall without changing the field distribution. The normal component of the electric field and the tangential component of the magnetic field are zero at such a plane.

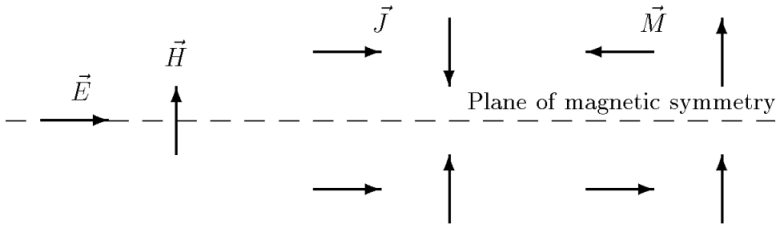


Figure 544: Magnetic symmetry plane

Computational Benefits of Using Symmetry

Exploiting symmetry in model affects the calculation of the matrix equation, which can lead to a reduction in runtime and memory requirements.

Geometric Symmetry

The arbitrarily placed sources lead to unsymmetrical current distributions. As a result, all unknown coefficients on all the mesh must be solved. The matrix equation being solved is, as a result, the same as it would have been, without symmetry being considered.

The computation time is however reduced for setting up the matrix equation. This reduction is achieved by exploiting the interaction between any two basis functions is the same as that between their symmetrical counterparts.

Electric / Magnetic Symmetry

When using electric / magnetic symmetry, less computational time is required to calculate the matrix equation entries. The major benefit of using symmetry is that the number of unknown coefficients is reduced by a factor of two. The system of linear equations to be solved has only half of the dimension, in comparison to a model without electric / magnetic symmetry.

The impact for the method of moments (MoM) is a reduction by a factor four (=2*2) in memory requirement, as the MoM has fully populated matrices.

The impact for the finite element method (FEM) is a reduction by a factor two in memory requirement, as the FEM leads to sparsely populated matrices. The reduction in unknowns also leads to a dramatic lowering of matrix equation solution time.

5.3.10 Media

Provided are the formulations and concepts to define frequency-dependent dielectric media and anisotropic media (3D).

Dielectric Media (Frequency-Dependent)

The frequency-dependent dielectric media formulations supported in the Solver are Debye relaxation, Cole-Cole, Havriliak-Negami, Djordjevic-Sarkar and frequency list (linear interpolation).

To define a dielectric, you need to define both the dielectric properties (dielectric modelling) and magnetic properties (magnetic modelling) of the medium.

Dielectric Modelling

The dielectric properties of the dielectric is defined.

Frequency Independent

The media is defined in terms of the relative permittivity (ε_r), relative permeability (μ_r), magnetic loss tangent ($\tan\delta_u$), and the dielectric loss tangent ($\tan\delta$) or conductivity (σ).

For example, low loss dielectric substrates are typically specified in terms of the loss tangent, while human tissue (used in specific absorption rate studies) are specified in terms of conductivity.

The effective permittivity of the dielectric is given by:

$$\varepsilon_{eff} = \varepsilon_0 \varepsilon_r \left(1 - j \tan\delta\right) \tag{63}$$

or

$$\varepsilon_{eff} = \varepsilon_0 \varepsilon_r - j\frac{\sigma}{\omega} \tag{64}$$

Debye Relaxation

The Debye relaxation[70] describes the relaxation characteristics of gasses and fluids at microwave frequencies. It has been derived for freely rotating spherical polar molecules in a predominantly non-polar background. The method is defined in terms of the relative static permittivity (ε_s), relative high frequency permittivity (ε_∞) and the relaxation frequency (f_r).

$$\varepsilon^* = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + j\frac{f}{f_r}} \tag{65}$$

Cole-Cole

The Cole-Cole[71] model is similar to the Debye model, but uses one additional parameter to describe the material. The model is defined in terms of the relative static permittivity (ε_s), relative high frequency permittivity (ε_∞), relaxation frequency (f_r) and the attenuation factor (a).

$$\varepsilon^* = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + \left(j\frac{f}{f_r}\right)^{1-a}} \tag{66}$$

Havriliak-Negami

The Havriliak-Negami[72] is a more general model and should be able to successfully model liquids, solids and semi-solids. It is defined in terms of the relative static permittivity (ε_s), relative high frequency permittivity (ε_∞), relaxation frequency (f_r), attenuation factor (a) and the phase factor (β).

$$\varepsilon^* = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{\left[1 + \left(j\frac{f}{f_r}\right)^{1-a}\right]^{\beta}} \tag{67}$$

70. R.Coelho, Physics of dielectrics for the Engineer, 1st ed. Elsevier Scientific Publishing Company, 1979.
71. K.S. Cole and R.H. Cole, “Dispersion and absorption in dielectrics,” Journal of Chemical Physics,vol.9, pp.341-351, 1941
72. J. Baker-Jarvis, M. D. Janezic, J. H. Grosvenor, and R.G. Geyer, “Transmission/reflection and short-circuit line methods for measuring permittivity and permeability: Technical note 1355-r,” National Institute of Standards and Technology, Tech. Rep., 1994

Djordjevic-Sarkar

The Djodervic-Sarkar[73] model is particularly well suited as a broadband model for composite dielectrics. It is defined in terms of the variation of real permittivity ($\Delta\varepsilon$), relative high frequency permittivity (ε_∞), conductivity (σ), lower limit of angular frequency (ω_1) and the upper limit of angular frequency (ω_2).

$$\varepsilon^* = \varepsilon_\infty + \left[\frac{\Delta\varepsilon}{log_{10}\frac{\omega_2}{\omega_1}}\right]\frac{ln\frac{\omega_2+j\omega}{\omega_1+j\omega}}{ln(10)} - \frac{j\sigma}{\omega\ \varepsilon_0} \tag{68}$$

Frequency List (Linear Interpolation)

Data points at a range of frequencies are specified. Values for the dielectric properties are then linearly interpolated to obtain the dielectric properties at frequency points other than specified. Parameters required are frequency, relative permittivity (ε_r) and either the loss tangent ($tan\delta$) or conductivity (σ).

Magnetic Modelling

The magnetic properties of the dielectric is defined.

Non-Magnetic

The relative permeability (μ_r) is set to 1.0 and the magnetic loss tangent ($tan\delta_u$) is set to 0.0.

Frequency Independent

The media is defined in terms of the relative permeability (μ_r) and the magnetic loss tangent ($tan\delta_u$).

The effective permeability of the dielectric is given by:

$$\mu_{eff} = \mu_0\mu_r(1 - jtan\delta_\mu) \tag{69}$$

Frequency List (Linear Interpolation)

The dielectric properties of the material are defined at various frequency points. Values for the dielectric properties are then linearly interpolated to obtain the dielectric properties at frequency points other than specified.

Data points at a range of frequencies are specified. Values for the dielectric properties are then linearly interpolated to obtain the dielectric properties at frequency points other than specified. Parameters required are frequency, relative permeability (μ_r) and the magnetic loss tangent ($tan\delta_u$).

73. Djordjevic, R.M. Biljic, V.D. Likar-Smiljanic, T.K. Sarkar, Wideband frequency-domain characterization of FR4 and time-domain causality, IEEE Transactions. on Electromagnetic Compatibility, vol. 43, no.4, 2001, p.662-667

Anisotropic Media (3D)

The anisotropic media formulations supported in the Solver are diagonalised tensor, full tensor, complex tensor and Polder tensor (for ferrites).

Note: Only passive media are supported. Passive media can be either lossless or lossy.[74]

Diagonalised Tensor

The permittivity along the UU, VV and NN axes are described by diagonal tensor:

$$\boldsymbol{\varepsilon} = \varepsilon_0 \boldsymbol{\varepsilon}_\mathbf{r} = \begin{bmatrix} \varepsilon_{uu} & 0 & 0 \\ 0 & \varepsilon_{vv} & 0 \\ 0 & 0 & \varepsilon_{nn} \end{bmatrix} \tag{70}$$

The permeability along the UU, VV and NN axes are described by diagonal tensor:

$$\boldsymbol{\mu} = \mu_0 \boldsymbol{\mu}_\mathbf{r} = \begin{bmatrix} \mu_{uu} & 0 & 0 \\ 0 & \mu_{vv} & 0 \\ 0 & 0 & \mu_{nn} \end{bmatrix} \tag{71}$$

Full Tensor

The permittivity along the UU, UV, UN, VU, VV, VN, NU, NV and NN axes are described by the dyadic tensor:

$$\boldsymbol{\varepsilon} = \varepsilon_0 \boldsymbol{\varepsilon}_\mathbf{r} = \begin{bmatrix} \varepsilon_{uu} & \varepsilon_{uv} & \varepsilon_{un} \\ \varepsilon_{vu} & \varepsilon_{vv} & \varepsilon_{vn} \\ \varepsilon_{nu} & \varepsilon_{nv} & \varepsilon_{nn} \end{bmatrix} \tag{72}$$

The permeability along the UU, UV, UN, VU, VV, VN, NU, NV and NN axes are described by the dyadic tensor:

$$\boldsymbol{\mu} = \mu_0 \boldsymbol{\mu}_\mathbf{r} = \begin{bmatrix} \mu_{uu} & \mu_{uv} & \mu_{un} \\ \mu_{vu} & \mu_{vv} & \mu_{vn} \\ \mu_{nu} & \mu_{nv} & \mu_{nn} \end{bmatrix} \tag{73}$$

Complex Tensor

The permittivity along the UU, UV, UN, VU, VV, VN, NU, NV and NN axes are described by the dyadic tensor:

$$\boldsymbol{\varepsilon} = \varepsilon_0 \boldsymbol{\varepsilon}_\mathbf{r} = \varepsilon_0 \begin{bmatrix} \varepsilon r_{uu} & \varepsilon r_{uv} & \varepsilon r_{un} \\ \varepsilon r_{vu} & \varepsilon r_{vv} & \varepsilon r_{vn} \\ \varepsilon r_{nu} & \varepsilon r_{nv} & \varepsilon r_{nn} \end{bmatrix} \tag{74}$$

The permeability along the UU, UV, UN, VU, VV, VN, NU, NV and NN axes are described by the dyadic tensor:

74. A lossless passive medium allows fields to pass through the medium without attenuation. In a lossy passive medium, a fraction of the power is transformed to heat, as an example.

$$\boldsymbol{\mu} = \mu_0 \boldsymbol{\mu}_\mathbf{r} = \mu_0 \begin{bmatrix} \mu_{r_{uu}} & \mu_{r_{uv}} & \mu_{r_{un}} \\ \mu_{r_{vu}} & \mu_{r_{vv}} & \mu_{r_{vn}} \\ \mu_{r_{nu}} & \mu_{r_{nv}} & \mu_{r_{nn}} \end{bmatrix} \tag{75}$$

To create the full permittivity and permeability tensors, create up to nine dielectrics constituting the medium properties along the UU, UV, UN, VU, VV, NU, NV and NN axes.

If no linear dependencies exist between two axes, add a zero (0) entry.

Important:

- An entry in the tensor must be a complex number, pure real number or a pure imaginary number.
- An entry may not be 0.

Polder Tensor

The ferrimagnetic[75] material is described by the permittivity tensor (where the static magnetic field is orientated respectively along the U, V and N axis):

$$\boldsymbol{\varepsilon} = \varepsilon_0 \boldsymbol{\varepsilon}_\mathbf{r} = \varepsilon_0 \begin{bmatrix} \varepsilon_r(1 - j\tan\delta) & 0 & 0 \\ 0 & \varepsilon_r(1 - j\tan\delta) & 0 \\ 0 & 0 & \varepsilon_r(1 - j\tan\delta) \end{bmatrix} \tag{76}$$

The ferrimagnetic material is described by the permeability tensors (where the static magnetic field is orientated respectively along the U, V and N axis):

$$\boldsymbol{\mu} = \mu_0 \boldsymbol{\mu}_\mathbf{r} = \begin{bmatrix} \mu_0 & 0 & 0 \\ 0 & \mu & j_\kappa \\ 0 & -j_\kappa & \mu \end{bmatrix} \quad \text{(X directed)} \tag{77}$$

$$\boldsymbol{\mu} = \mu_0 \boldsymbol{\mu}_\mathbf{r} = \begin{bmatrix} \mu & 0 & j_\kappa \\ 0 & \mu_0 & 0 \\ -j_\kappa & 0 & \mu \end{bmatrix} \quad \text{(Y directed)} \tag{78}$$

$$\boldsymbol{\mu} = \mu_0 \boldsymbol{\mu}_\mathbf{r} = \begin{bmatrix} \mu & j_\kappa & 0 \\ -j_\kappa & \mu & 0 \\ 0 & 0 & \mu_0 \end{bmatrix} \quad \text{(Z directed)} \tag{79}$$

Where μ and κ elements of the permeability tensor are given by

$$\mu = \mu_0 \left(1 + \frac{\omega_0 \omega_m}{\omega_0^2 - \omega^2}\right) \tag{80}$$

75. D. M. Pozar, “Theory and Design of Ferrimagnetic Components” in “Microwave Engineering”, 2nd ed., New York: Wiley, 1997, ch 9, pp. 497-508

$$\kappa = \mu_0 \frac{\omega \omega_m}{\omega_0^2 - \omega^2} \tag{81}$$

and where,

operating frequency: ω

Lamor (precession) frequency: $\omega_0 = \mu_0 \gamma H_0$

forced precession frequency: $\omega_m = \mu_0 \gamma M_s$

gyromagnetic ratio: γ

magnetic bias field: H_0

DC saturation magnetisation: M_s.

To account for magnetic loss, the resonant frequency can be made complex by introducing a damping factor (α) into Equation 80 and Equation 81. The damping factor and the field line width (ΔH), the width of the imaginary susceptibility curve against the bias field at half its peak value, are related by

$$\alpha = \frac{\mu_0 \gamma \Delta H}{2\omega}. \tag{82}$$

Note: The Polder tensor is defined using CGS[76] units in terms of:

- saturation magnetisation (Gauss): $4\pi M_s$
- line width (Oersted): ΔH
- DC bias field (Oersted): H_0
- field direction.

76. CGS is the system of units based on measuring lengths in centimetres, mass in grams and time in seconds.

Optimisation in Feko

6

Feko offers state-of-the-art optimisation engines based on generic algorithm (GA) and other methods, which can be used to automatically optimise the design and determine the optimum solution.

This chapter covers the following:

The CADFEKO interface supports optimisation searches. Refer to the optimiser OPTFEKO for information regarding optimisation algorithms and related options. An optimisation example can be found in the Feko Getting Started Guide.

> **Note:** Continuously sampled results (generated using ADAPTFEKO) cannot be used in an optimisation. Only single or discretely sampled frequency settings are allowed.

Related concepts
OPTFEKO

6.1 Optimisation Workflow in CADFEKO

The workflow for setting up an optimisation in CADFEKO is explained.

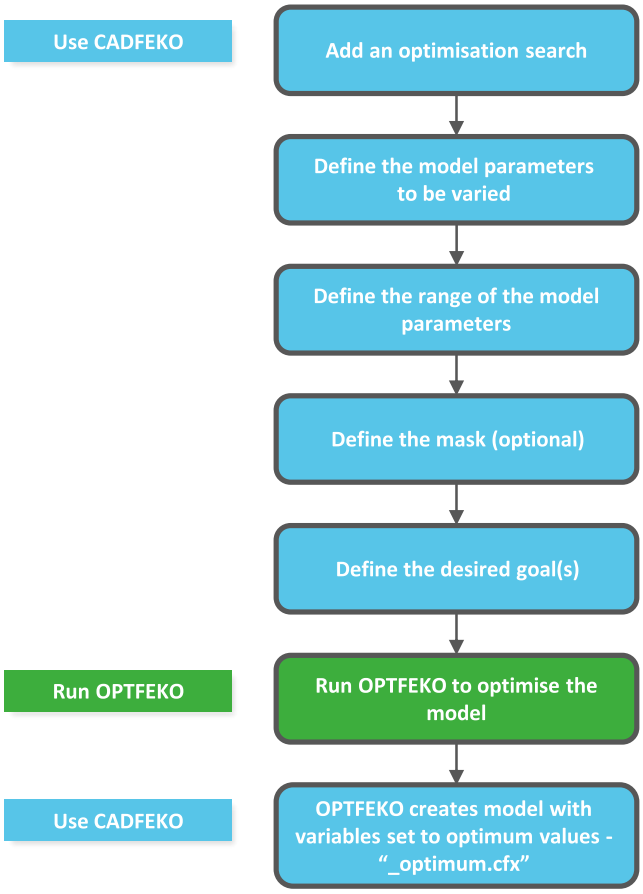


Figure 545: The workflow for defining an optimisation search in CADFEKO.

Select the Optimisation Method

The following optimisation methods are available::

- Simplex (Nelder-Mead)
- Particle swarm optimisation (PSO)
- Genetic algorithm (GA)
- Adaptive response surface method (ARSM)
- Global response surface method (GRSM)
- Grid search

Select the Model Parameters

The model parameters are the variables defined by the user with which certain characteristics of the model can be varied, for example, its length, spacing between parts, and height. In this step of the workflow, the variables used in the optimisation are selected from a drop-down list.

Define the Parameter Range

Define the range over which each selected parameter varies by specifying the **Min value**, **Max value** and optionally the **Start value**.

Define an Optimisation Mask

This step is optional. An optimisation mask is a set of user-specified values that form a continuous line to which the optimal solution is fitted to. The optimised solution is specified to be either less than, equal or greater than the mask. During the calculation of the optimal solution, the goal values are compared to the mask. If the mask criterion is satisfied, the values are added to an array of values.

Define the Optimisation Goal

Define the goal(s) that specify the desired state of the model that the optimisation process should attempt to achieve by varying the specified model parameters.

Run the Optimisation (OPTFEKO)

Run OPTFEKO to calculate the optimum solution for the specified parameters.

View the Optimum Model

After the optimisation completed, a CADFEKO model is created with the optimum parameters. The file is given a "_optimum" suffix.

> **Note:** Continuously sampled results (generated using ADAPTFEKO) cannot be used in an optimisation. Only single or discretely sampled frequency settings are allowed.

Related concepts

Optimisation Methods and Stopping Criteria

6.2 Launching OPTFEKO (Windows)

Use the most suitable option for launching OPTFEKO.

Launch OPTFEKO using one of the following options:

- On the **Solve/Run** tab, in the **Run/Launch** group, click the OPTFEKO icon.
- From the command line in a terminal environment.
  1. On the desktop, click the Windows **Start** button.
  2. Type `Feko + WinProp 2024.1`.
  3. Select the **Feko + WinProp 2024.1** icon, from the list of filtered options.
  4. On the **Tools** tab, select the **Feko Terminal** icon.
  5. In the terminal (assuming the model with file name `dipole.cfx` is to be optimised) type the following command

     ```
     optfeko dipole
     ```

     and press Enter.

> **Note:** The above steps launches OPTFEKO without any special settings. It is also possible to use parallel processing for optimisation.

Related reference

Command Line Arguments for Launching OPTFEKO

6.3 Launching OPTFEKO (Linux)

Use the most suitable option for launching OPTFEKO in Linux.

Launch OPTFEKO using one of the following options:

- On the **Solve/Run** tab, in the **Run/Launch** group, click the  OPTFEKO icon.
- Launch OPTFEKO from the command line in a terminal environment.
  1. Open a command terminal. Source the script "initfeko" using the absolute path to `. /home/user/2024.1/altair/feko/bin/optfeko`
  2. In the terminal (assuming the model with file name `dipole.cfx` is to be optimised) type the following command

     ```
     optfeko dipole
     ```

     and press Enter.

> **Note:** The above steps will launch OPTFEKO without any special settings. It is also possible to use parallel processing for optimisation.

6.4 Command Line Arguments for Launching OPTFEKO

Use the command line arguments to pass additional information to OPTFEKO when it is launched.

Table 61: Command line arguments for launching OPTFEKO.

Argument	Description
`--version`	Displays the current version of OPTFEKO.
`-r`	All interim model files are deleted after each analysis. The optimum results are, however, not deleted, and are available with the string (`_optimum`) appended to the file name. This saves disk space during and after the optimisation process.
`--restart x`	Resumes an optimisation process that has been stopped, provided that all of the interim optimisation files (`.fek`, `.bof` and `.cfx`) are still available (for example, the previous optimisation has been stopped by pressing `Ctrl+C` or due to a power failure or a Feko error).
`-np x`	The number of processors used for farming out of the individual optimisation steps.
`--machines-file machname`	The file `machname` is the machines file with the node names and the number of CPUs used for farming of the individual optimisation steps. This machines file is used for both farming and parallel execution when farming and parallel execution is used simultaneously.
`--eval-aim-only x`	The value of the goal function is calculated only for one existing file (x) — no optimisation is done. (This is mostly used for debugging.)

`--runfeko-options`	After this option one can specify additional options which is used in the launcher RUNFEKO for the Solver. For example, to use the parallel Solver during the optimisation, the command
	`optfeko file --runfeko-options -np 2`
	or
	`optfeko file --runfeko-options -np 2 --machines-file m`

Argument	Description
	where `m` is the machines file. For a remote execution of the Feko runs during the optimisation on another host, the suitable command is `optfeko file --runfeko-options --remote-host hostname` Additional options for ADAPTFEKO and PREFEKO is included in the OPTFEKO command as part of the RUNFEKO options. The options are passed to the relevant component by RUNFEKO as needed. This allows for control of all of the Feko components during the optimisation process.

6.5 Optimisation Methods and Stopping Criteria

The duration and accuracy of an optimisation depends on the selected optimisation method and stopping criteria.

On the **Request** tab, in the **Optimisation** group, click the **Add Search** icon.

After adding the optimisation search it is visible in the model tree . To change the optimisation search method and settings double-click or open the right-click context menu for the relevant search (the default label is **Search1**) in the **Optimisation** tree.

The following optimisation method types are supported (see Table 62).

Automatic: A method is automatically chosen by the optimiser.

Simplex (Nelder-Mead): A gradient-based or "hill-climbing" method.

Particle swarm optimisation (PSO): A swarm-based global search method.

Genetic algorithm (GA): An evolutionary global search method.

Grid search: This method searches over a predefined grid of parameter sets.

Adaptive response surface method (ARSM): This method internally builds a response surface that is updated as more sample points are added.

Global response surface method (GRSM): This method internally builds a response surface that is updated as more sample points are added and continues to test different areas of the design space.

Table 62: Optimisation methods overview

Method	Description	Number of variables	Convergence	Accuracy	Farming
Simplex	local search, optimum strongly dependent on starting point	low	fast	locally high, globally low	initial/ recreating simplex
PSO	population-based stochastic global search	high	slow	medium/high	yes
GA	robust, stochastic global search	high	slow/medium	medium/high	yes
ASRM	response surface based approach	medium	fast	low/medium	no

Method	Description	Number of variables	Convergence	Accuracy	Farming
GSRM	response surface based approach, good balance between local and global	high	medium	high	yes

Create Optimisation Search - Options Tab

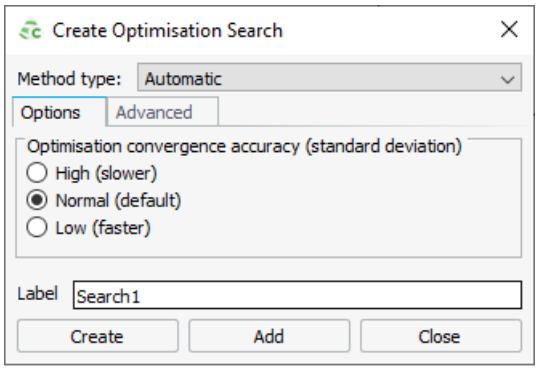


*Figure 546: The **Create Optimisation Search** dialog, **Options** tab*

Note: The layout of the **Options** tab depends on the selected optimisation **Method type**.

Optimisation convergence accuracy (standard deviation)
This setting controls the level of accuracy required by the search algorithm to converge. The three options, **High (slower)**, **Normal (default)** and **Low (faster)** modify the conditions under which the search algorithm converges, and is also dependent on which optimisation **Method type** is chosen, since some techniques have a predetermined number of samples.

Default number of points
Only applicable when the **Method type** is set to **Grid search**. Specify the number of grid points to use for each optimisation parameter in the predefined grid. This value is used for the **Grid points** on the **Optimisation parameters** dialog if no values are specified.

Add Optimisation Search - Advanced Tab

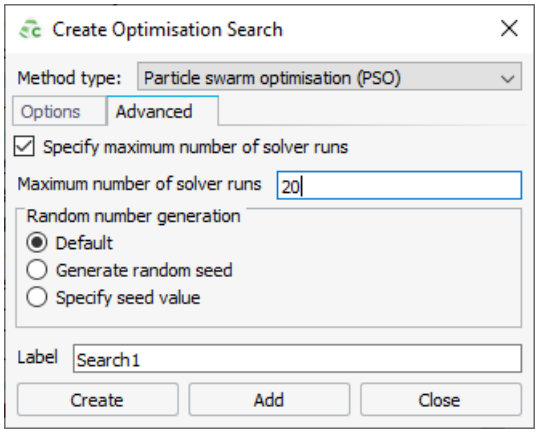


*Figure 547: The **Add optimisation search** dialog, **Advanced** tab.*

Note: The layout of the **Advanced** tab depend on the selected optimisation **Method type**.

Specify maximum number of solver runs

The optimisation process is terminated when the Feko Solver is launched, the specified number of times during the optimisation process.

For the PSO and GA methods, should a full swarm or generation not be generated within the allowable number of allocated runs, the optimisation may terminate before the indicated number of solver runs.

When an optimisation process terminates due to reaching the value in **Specify maximum number of solver runs**, the optimum solution found up to that point and the optimisation process information are made available.

Random number generation

This group is visible for those methods that make use of randomised sampling and allows setting the seed value.

Default	The seed value is set equal to a fixed default.
Generate random seed	The seed value is set equal to a random integer number.
Specify seed value	The seed value is entered as a positive integer.

Multiple Searches

If multiple searches are defined in a model, and is represented as individual branches below the **Optimisation** heading in the model tree. Only one optimisation search may be activated at a time. If only one search is defined in the model, then the search is active. The settings for each search are independent, and only the settings specified in the active search are saved to the `.opt` and `.pfg` files for use during an optimisation run.

To activate a specific search, from the right-click context menu select **Activate** or select the **Request** tab and click the **Activate** icon.

The active search is indicated by the  icon in the model tree.

Related concepts

Simplex (Nelder-Mead)

Particle Swarm Optimisation (PSO)

Genetic Algorithm (GA)

Grid Search

Adaptive Response Surface Method (ARSM)

Global Response Surface Method (GRSM)

6.6 Optimisation Parameters

Specify the variables to alter during the optimisation run.

In the model tree (**Construction** tab), select the relevant search. On the **Request** tab, in the **Optimisation** group, click the **Parameters** icon.

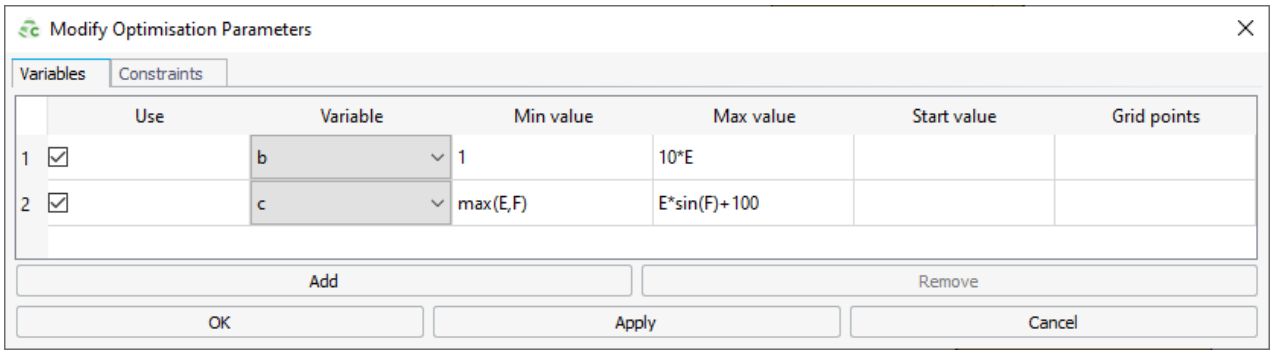


*Figure 548: The **Optimisation parameters** (**Variables** tab) dialog.*

The optimisation parameters are local to each optimisation search and a valid search must contain at least one active parameter. Any variable defined in CADFEKO is available as an optimisation parameter, for example, physical dimensions, loads and sources (amplitude and phase), provided that a dependency is not implied between optimisation parameters in the same search. Optimisation parameters are added or removed from the list by using the **Add** and **Remove** buttons.

For each optimisation parameter a **Min value** and **Max value** is required. Optionally a **Start value** in the variable range can be specified. The starting value effects the optimisation process when randomised techniques are used, for instance, particle swarm optimisation or genetic algorithm. If the **Start value** is not specified by the user, the value at the centre of the range will be taken as the starting point for the optimisation.

Related concepts
Particle Swarm Optimisation (PSO)
Genetic Algorithm (GA)

6.6.1 Constraints Between Optimisation Parameters

A dynamic boundary is defined for an optimisation parameter by specifying a constraint.

A constraint is defined by specifying two parameters and their dependency on one another, see Figure 549. The following dependencies are available: **!=**, **<**, **<=**, **>** and **>=**.

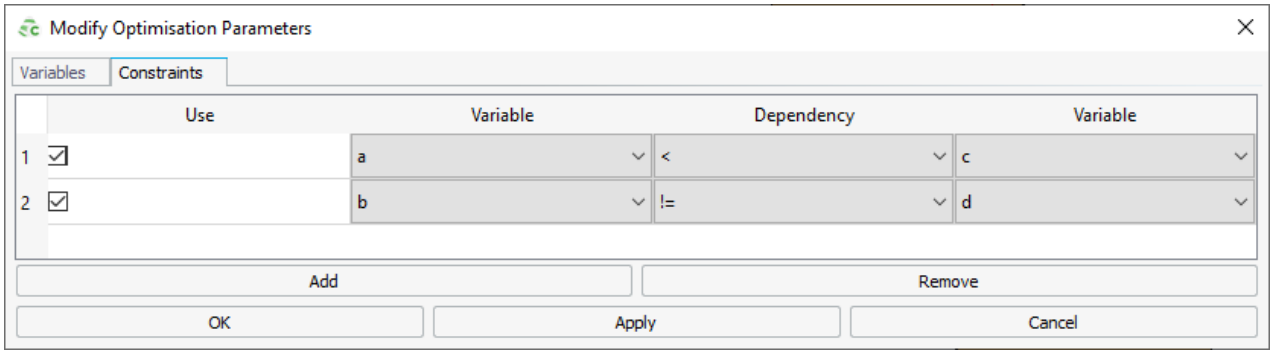


*Figure 549: The **Optimisation parameters** (**Constraints** tab) dialog.*

Parameter and Constraint Deactivation

For each parameter in the parameter list or constraint in the parameter constraints list, a **Use** check-box is used to include or exclude each specific parameter or constraint in the optimisation search process. If the **Use** check box for a specific parameter or constraint is not selected then that parameter or constraint is excluded in the `.opt` or `.pfg` files and does not influence the optimisation search. If a parameter is deactivated, the value of the variable as specified in the CADFEKO variables list is used as if it is not defined as an optimisation parameter.

Note: All parameter and constraint settings are local to each search. Deactivating a specific parameter or constraint in the parameter settings of one search does not deactivate that parameter or constraint in any other search.

6.7 Optimisation Masks

An optimisation mask is a graphical approach to define an optimisation search. More complex scenarios are handled by a mask, where the goal shape is visually known, for example, the desired bandpass and bandstop regions of a filter.

6.7.1 Defining an Optimisation Mask

An optimisation mask is a set of values that form a continuous line (or trace). Use a mask for specifying a specific performance curve for a requested output.

1. On the **Request** tab, in the **Optimisation** group, click the **Add Mask** icon.
2. Specify the **X** and **Y** coordinates of the mask points using one of the following methods:
   1. Enter the coordinates in the **X** and **Y** text boxes.
   2. Import the values from an external file by clicking the **Import points** button.

      Note: The import starts from the first coordinate point and overwrites any existing coordinate definitions. Changes in the external file requires a re-import of the values.
3. Enter a label for referencing the mask in the optimisation objective.
4. Press **Create** to create and close the dialog or press **Add** to add another mask.

Note: The same mask can be used in multiple optimisation searches.

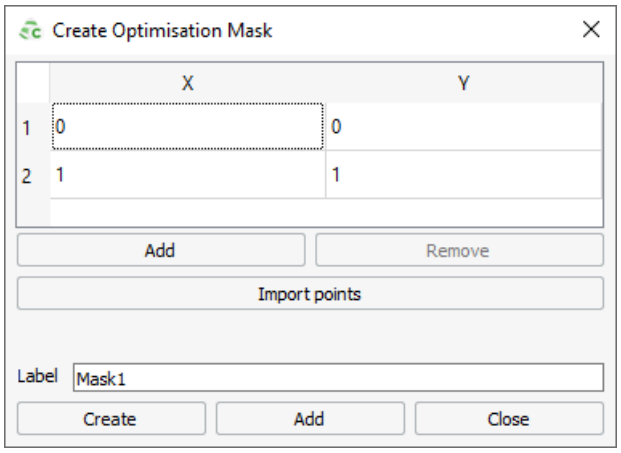


*Figure 550: The **Create Optimisation Mask** dialog.*

6.7.2 How Masks are Used for Optimisation

Perform complex optimisations with masks by specifying a variable goal. Apply the mask correctly to avoid undesired results.

See Figure 551 for a graphical representation of the mask. This is useful for validating that the mask data is correct, particularly when working with a large number of data points imported from an external file.

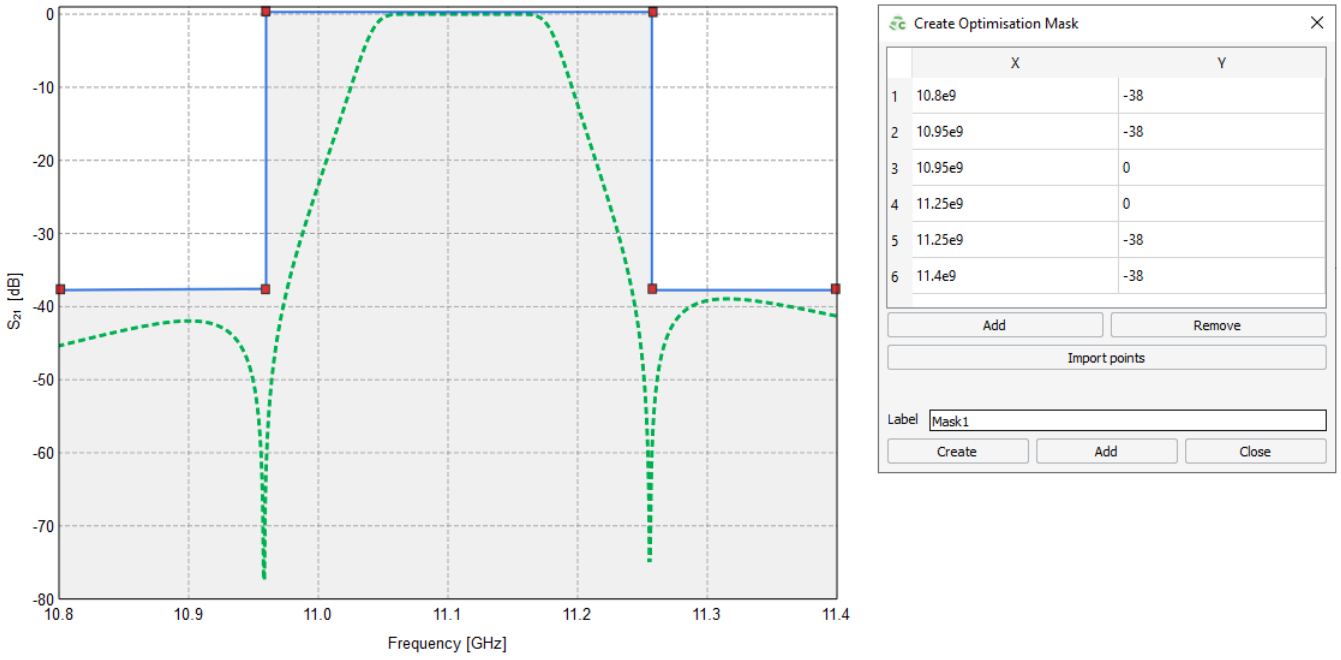


*Figure 551: The desired frequency response of a Ku-band waveguide filter (indicated in green) with a mask (indicated in blue) and (b) the **Create optimisation mask** dialog.*

All the calculated points that satisfy the criteria for the goal type and name are added to a long array of values and then compared to the mask. The following examples illustrate the usage of masks during optimisation.

Example 1: Optimisation of a Far Field Pattern at a Single Frequency

Create the mask with the required shape and the far field request that is compared to the mask. The first point (angle) in the far field calculation map to the first point in the mask and the last point (angle) in the far field calculation map to the last point in the mask. All other points of the far field is compared to points in the mask (linear interpolation is used to ensure a continuous mask).

Example 2: Optimisation for a Specific Gain Profile over Frequency

Create the far field request containing a single far field point. There after create the mask that contains the gain profile over frequency. The gain at the first frequency map to the first point in the mask and the gain at the last frequency map to the last point in the mask. The gain at the frequency values within the range are compared to the values in the mask using linear interpolation.

Example 3: Optimisation for a Varying Far Field (Gain) Profile Over Frequency

Using a combination of the two examples above create a complex optimisation requiring a predefined far field/gain pattern that changes as a function of frequency. Create the multi-point far field requests for each frequency. There after create a mask that the first point of the mask map to the first point in the far field request at the first frequency. The last point in the mask map to the last point of the far field request of the last frequency. If the required far field pattern is unchanged over frequency then the mask contains the same far field pattern repeated N times, where N is the number of frequency points.

Warning: Optimisation does not fail due to an incorrect mask, but the optimum results could be unexpected.

6.8 Defining an Optimisation Goal

An optimisation goal defines the request type to be optimised and the goal it will attempt to achieve.

To define an optimisation goal, you first need a defined optimisation search.

1. In the model tree under **Optimisation**, select a search.
2. On the **Request** tab, in the **Optimisation** group, click the **Add Goal Function** icon.
3. From the drop-down list, select one of the following:
   - **Impedance Goal**
   - **Near Field Goal**
   - **Far Field Goal**
   - **Power Goal**
   - **Receiving Antenna Goal**
   - **S-Parameter Goal**
   - **Transmission / Reflection Goal**
   - **SAR Goal**

6.8.1 Structure of an Optimisation Goal

Each part of the definition in an optimisation goal serves a specific purpose and should be correctly understood and applied for the desired optimisation outcome.

All optimisation goals (irrespective of type) have the same basic structure. They are divided into four basic parts.

Goal focus
: The part of the Feko solution to be considered for optimisation. The **Focus type** is based on a quantity computed by the Feko solver. It is uniquely identified based on the request **Label**. If the solution request was defined in CADFEKO, select the request label from the drop-down list. If the solution request was defined in EDITFEKO, enter the label of the request.

Focus processing steps
: A number of conversion steps or mathematical operations to be carried out on the Focus before the Goal is evaluated. Processing steps may be specific to the focus and goal type, while other processing steps are generic to all focus and goal types. The number, order and type of processing steps can be freely chosen by the user to provide flexibility in the goal definition.

Goal operator
: The operator indicates the desired relationship between the focus and the objective.

Goal objective
: The objective describes a state that the optimisation process should attempt to achieve. The objective is predefined and assumes the same unit as the focus.

Weight
: The weight modifies the contribution of the goal's error relative to other errors at the same tree level during the fitness evaluation. The error at each level is computed by multiplying the evaluated error of each goal with the associated weighting factor and then summing all of the weighted errors. The global error is the summation of the weighted errors at the highest level of the tree.

Label
: The label for each goal identifies the simulation results to be considered during the evaluation of the goal.

6.8.2 Optimisation Goal Types

Select an optimisation goal consistent with the requested output.

Impedance Goal

Optimise the impedance or admittance of a voltage/current source, solved as part of the Feko model.

On the **Request** tab, in the **Optimisation** group, click the **Add Goal Function** icon. From the drop down list, select **Impedance goal**.

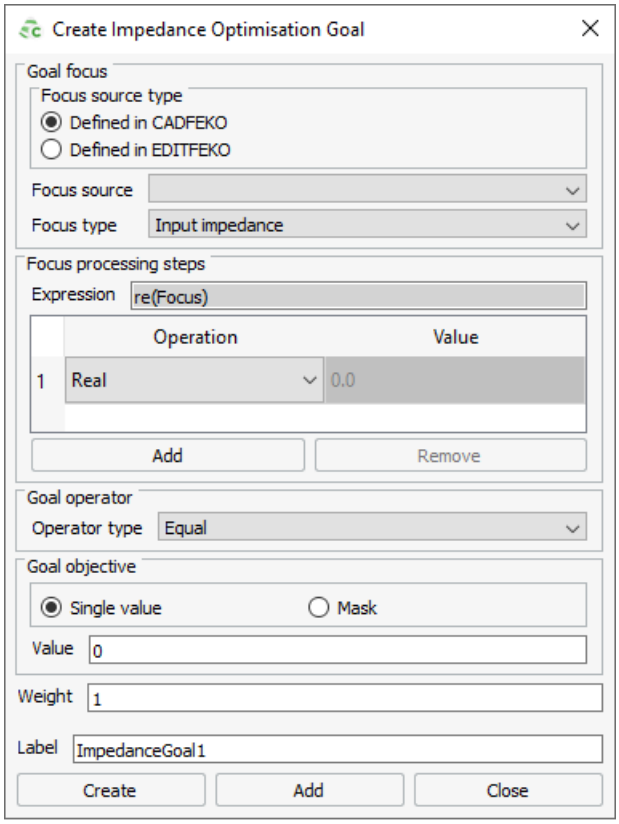


*Figure 552: The **Create Impedance Optimisation Goal** dialog.*

The **Focus source** is identified based on the label of a voltage source or current source in CADFEKO. The **Focus source label** is identified based on the or a card-defined source, for example, the A1, A2, A3, AF and AN card in EDITFEKO.

Focus Types

The following focus types are available:

Input impedance/Input admittance
: Both of these are complex quantities that represent the load characteristics (based on the currents and voltages at the source points). As these focus types always consist of complex values, the focus processing options require that there be at least one general processing step indicating the selection of one of the complex components.

Reflection coefficient (S11)
: The reflection coefficient is computed with respect to the **Reference impedance**. For the impedance goal, the reflection coefficient is computed directly from the observed input impedance. This value is then in effect the "active" reflection coefficient (Γ) and may differ from the S11 computed during an S-parameter calculation in a multi-port model.

Transmission coefficient
: The transmission coefficient ($1-\Gamma$) is considered with respect to the **Reference impedance**.

VSWR

The voltage standing wave ratio ($VSWR = \frac{1+|S_{nn}|}{1-|S_{nn}|}$) for the observed input impedance is considered with respect to the **Reference impedance**.

Return losses

The return loss ($-20\log|\Gamma|$) for the observed input impedance is considered with respect to the **Reference impedance**.

Current

The current flowing through the segment on which the selected voltage source is located. In order to use this optimisation goal, a port with a source or an A1 card must be applied to the segment of interest. The source should be set to zero magnitude, and a suitable **Source name** (label) entered.

Reference Impedance

The impedance to be used during the calculation of the relevant focus types can be specified here. The impedance must be a single non-complex value and is local to each impedance goal (different reference impedances may be used for different impedance goals in the same optimisation search).

Related reference

A1 Card
A2 Card
A3 Card
AF Card
AN Card

Near field Goal

Optimise the near fields that are solved as part of the Feko model.

On the **Request** tab, in the **Optimisation** group, click the **Add Goal Function** icon. From the drop down list, select **Near Field Goal**.

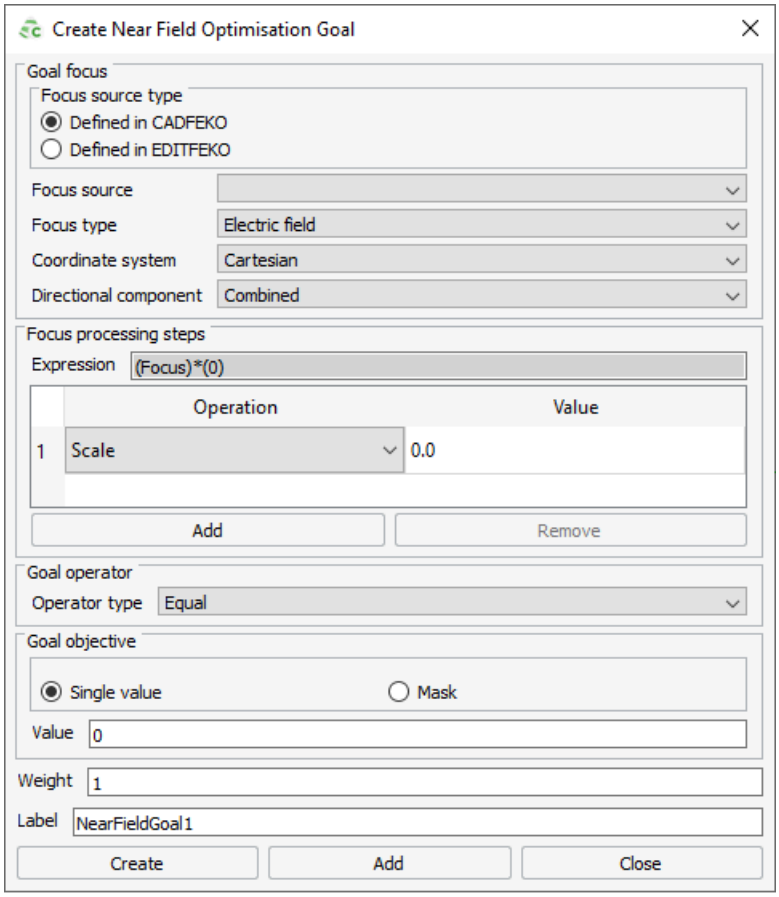


*Figure 553: The **Create Near Field Optimisation Goal** dialog.*

The **Focus source** is identified based on the label of a **Near fields** request in CADFEKO or the **Focus source label** of an FE card in EDITFEKO.

The following focus types can be optimised:

Electric field
: The electric field part of the near field is considered.

Magnetic field
: The magnetic field part of the near field is considered.

Electric flux density (normalised)
: The **Electric flux density (normalised)** considers the electric field scaled by the relative permittivity of the medium where the near field is calculated. These are normalised quantities, with the electric flux densities scaled by the permittivity of free space. The normalisation prevents the goal function from having values that are small enough for the optimiser to consider them to be zero.

Magnetic flux density (normalised)
: The **Magnetic flux density (normalised)** considers the magnetic field scaled by the relative permeability of the medium where the near field is calculated. These are normalised quantities, with the magnetic flux densities scaled by the permeability of free space. The normalisation

prevents the goal function from having values that are small enough for the optimiser to consider them to be zero.

> **Note:** If the focus type attempts to access a part of the near field output request that was not requested (for example only electric fields requested but optimisation is for magnetic fields) then an error will be returned during the evaluation of the goal in the first optimisation iteration.

Coordinate System

The coordinate system in which the directional component of the near field is required must be selected. The available coordinate systems are **Cartesian**, **Cylindrical(X)/(Y)/(Z)**, **Spherical** and **Conical**. This coordinate system selection defines the options available in the **Directional component** drop-down list.

> **Note:** The coordinate system chosen here differs from the coordinate system chosen as part of the near field computation request.

The coordinate system choice in the near field goal is related to the near field component of interest, while the coordinate system chosen in the near field output request dialog is related to the positioning of the sample points for the near field calculation. This distinction makes it possible to consider the near field component in any direction independently of the physical placement of the near field sampling points.

Directional Component

The options available in the **Directional component** drop-down list depends on the choice of **Coordinate system**, but are independent of the near field request sampling point positions.

Radial or X/Y/Z/Phi/Theta-directed

In the chosen **Coordinate system**, the field in any of the 3 coordinate directions may be requested. Each individual component of the electric or magnetic near field is a complex quantity, and the selection of a specific field component requires that there be at least one general processing step which indicates the selection of one of the complex components.

Combined

In addition to the individual components in the coordinate directions, the **Combined** near field value may be requested. This value is computed by combining all 3 directional components of the field at each point as follows (shown for Cartesian components):

$$E_{combined} = \sqrt{|F_x|^2 + |F_y|^2 + |F_z|^2} \quad (83)$$

The choice of **Coordinate system** has no effect on the value of the **Combined** component. The combined field is always a non-complex value (or an array of non-complex values) and it is therefore not required that any further processing is performed.

Related reference

FE Card

Far Field Goal

Optimise the far fields that are solved as part of the Feko model.

On the **Request** tab, in the **Optimisation** group, click the **Add Goal Function** icon. From the drop down list, select **Far Field Goal**.

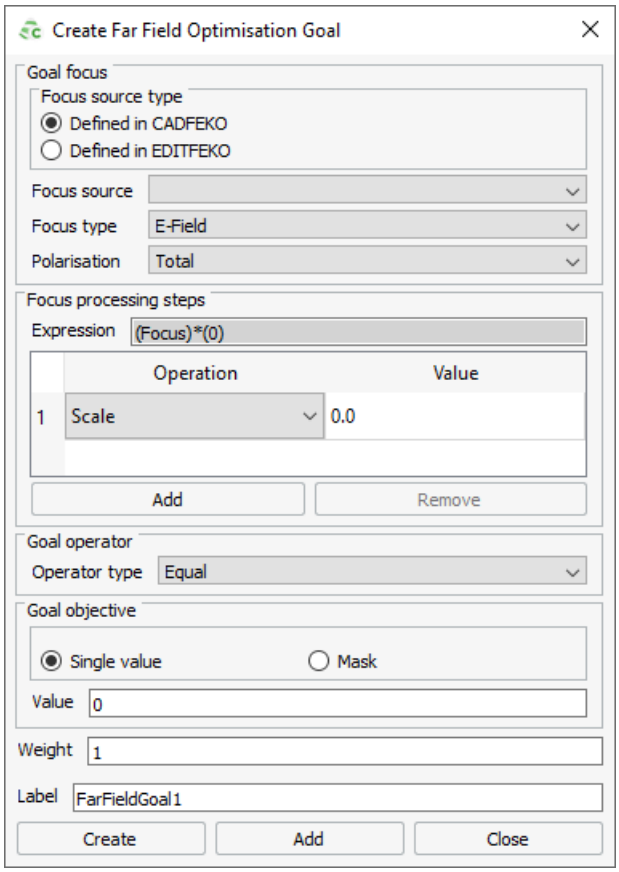


*Figure 554: The **Create Far Field Optimisation Goal** dialog.*

The **Focus source** is identified based on the label of a **Far fields** request in CADFEKO or the **Focus source label** of an FF card in EDITFEKO.

The following focus types can be optimised:

E-field
: The **E-field** focus type considers the radiated fields associated with a specific far field solution request directly. The fields are considered according to the settings of the far field request. For example if only the scattered fields from a single object are requested, then only these will be taken into account in the goal evaluation.

Directivity, Gain and Realised gain
: With this focus type, only the directivity, gain or realised gain of the model is considered. This option can only be based on a far field request where the **Calculate fields as specified** option is chosen and is independent of whether **Directivity** or **Gain** is selected in the far field request.

Radar cross section (RCS):

This focus type is only valid for far field solutions that have been computed with a plane wave source. The **RCS** focus type delivers non-complex values (or an array of non-complex values) representing the derived RCS (see section) according to the options set in the far field calculation request. If no valid RCS information is found in the computation output, an error will be generated during the goal evaluation.

Note: Fields that are requested in invalid directions (for example fields requested below an infinite ground plane) are ignored during the Goal evaluation. If no valid far field results with the correct request label are found in the solver output, an error will be generated during the Goal evaluation phase of the first optimisation iteration.

Polarisation

The Polarisation option allows the specification of the far field component to be considered in the goal.

Total

For the **E-field** focus type, the **Total** option provides a magnitude combination of the θ- and ϕ-components of the far field. The total field is calculated as:

$$E_{total} = \sqrt{|E_\theta|^2 + |E_\phi|^2} \tag{84}$$

This value is representative of the power in the far field.
For the **Directivity**, **Gain** and **Realised gain** focus types, the polarisation-independent quantities are considered. This is the only **Polarisation** option for **RCS**.

Horizontal (Phi)/Vertical (Theta)

These options allow specific selection of the θ- and ϕ-directed components of the far field. For the **Directivity**, **Gain** and **Realised gain** focus types, only the component of the field with the selected polarisation is used in the calculation of the required quantity, delivering a non-complex value (or array of non-complex values).

LHC/RHC

These options allow specific selection of the left-hand-circular and right-hand-circular components of the far field (see section). For the **Directivity**, **Gain** and **Realised gain** focus types, only the component of the field with the selected polarisation is used in the calculation of the required quantity, delivering a non-complex value (or array of non-complex values).

S/Z

These options allow specific selection of the S- or Z-polarised components of the far field (see section). For the **Directivity**, **Gain** and **Realised gain** focus types, only the component of the field with the selected polarisation is used in the calculation of the required quantity, delivering a non-complex value (or array of non-complex values).

Axial ratio

This option is only available for the **E-field** focus type. This provides the ratio between the magnitudes of the θ- and ϕ-directed field components (see section).
For the purposes of optimisation, an additional sign is added to the **Axial ratio** value considered by the optimiser. The sign indicates the handedness of the radiated field, with a negative sign

implying left-handedness, and a positive sign implying right-handedness. This makes provision for the inclusion of the required handedness directly in the **Axial ratio** optimisation.

Ludwig III (Co and Cross)
: These options allow specific selection of the **Ludwig III (Co)** and **Ludwig III (Cross)** polarised components of the far field (see section).

Related reference
FF Card

S-Parameter Goal

Optimise the S-parameters that are solved as part of the Feko model.

On the **Request** tab, in the **Optimisation** group, click the **Add Goal Function** icon. From the drop down list, select **S-Matrix Goal**.

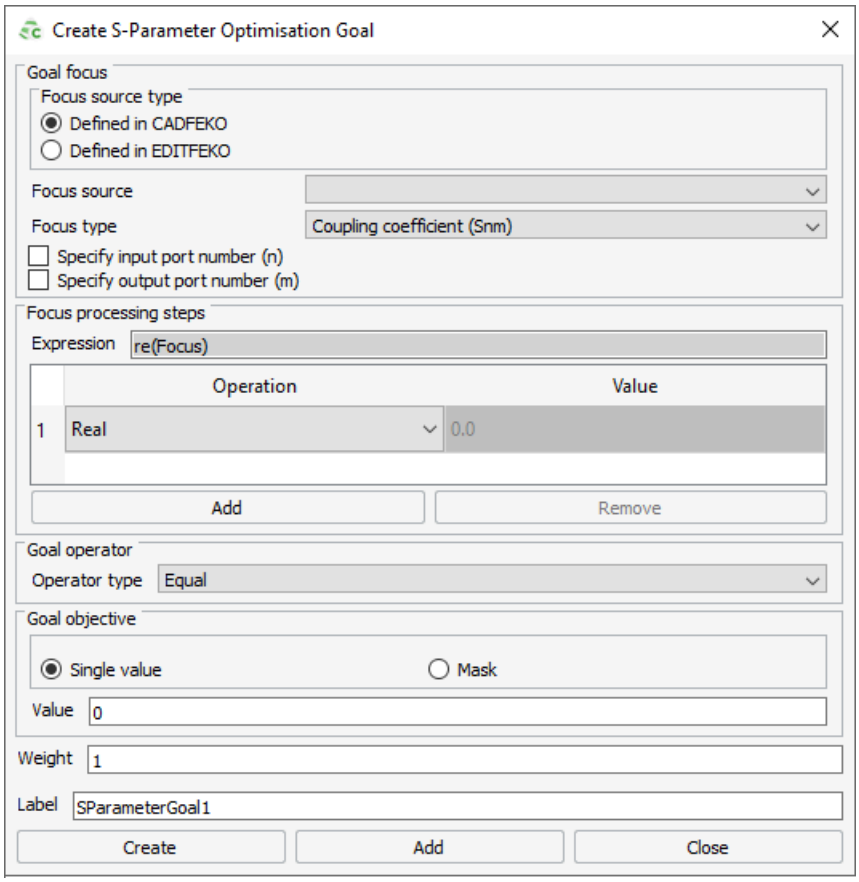


*Figure 555: The **Create S-Parameter Optimisation Goal** dialog.*

The **Focus source** is identified based on the label of a **Multiport S-parameter** request in CADFEKO or the **Focus source label** of an SP card in EDITFEKO.

Quantity

Coupling coefficient (S_{mn})

Only the coupling between different ports will be considered in the optimisation (all S-parameter values where the port indices are not equal).

> **Note:** If Snm and Smn are computed in an S-parameter request, then both of these values will be considered in the goal evaluation. If the coupling in one direction is required, the relevant port should be deactivated in the S-parameter calculation request (CADFEKO) or the source set to zero magnitude (EDITFEKO).

Reflection coefficient (S_{nn})

Only the reflection at the port(s) will be considered in the optimisation. The reflection coefficient at all ports that are active for the S-parameter computation will be considered.

Return loss

The return loss at the port(s) will be considered in the optimisation. Return loss is calculated from the reflection coefficient at each active port as:

$$RL = -20\log|S_{nn}| \tag{85}$$

Transmission coefficient

The transmission coefficient at the port(s) will be considered in the optimisation. The transmission coefficient is calculated from:

$$\Upsilon = 1 - |S_{nn}| \tag{86}$$

VSWR

The voltage standing wave ratio at the port(s) will be considered in the optimisation. Return loss is calculated from the reflection coefficient at each active port as:

$$VSWR = \frac{1+|S_{nn}|}{1-|S_{nn}|} \tag{87}$$

Port Selection

Specify input port number (n)

By default all of the active ports will be considered during the goal evaluation. When activated, this option allows the selection of a single port to be used as the input port. For example, if all of the S-parameters in a 3-port device are computed and the **Focus** quantity is chosen as **Coupling coefficient (Smn)**, then if the input port is specified as `2`, only the values of S12 and S32 will be considered during the goal evaluation.

Specify output port number (m):

In a similar manner to the input port selection option, when this option is selected it allows the selection of a single port to be used as the output port. For example, if all of the S-parameters in a 3-port are computed and the **Focus** quantity is chosen as **Coupling coefficient (Smn)**, then by specifying the output port as `2`, only the values of S21 and S23 will be considered during the goal evaluation.

Related reference
SP Card

SAR Goal

Optimise the SAR that are solved as part of the Feko model.

On the **Request** tab, in the **Optimisation** group, click the **Add Goal Function** icon. From the drop-down list, select **SAR Goal**.

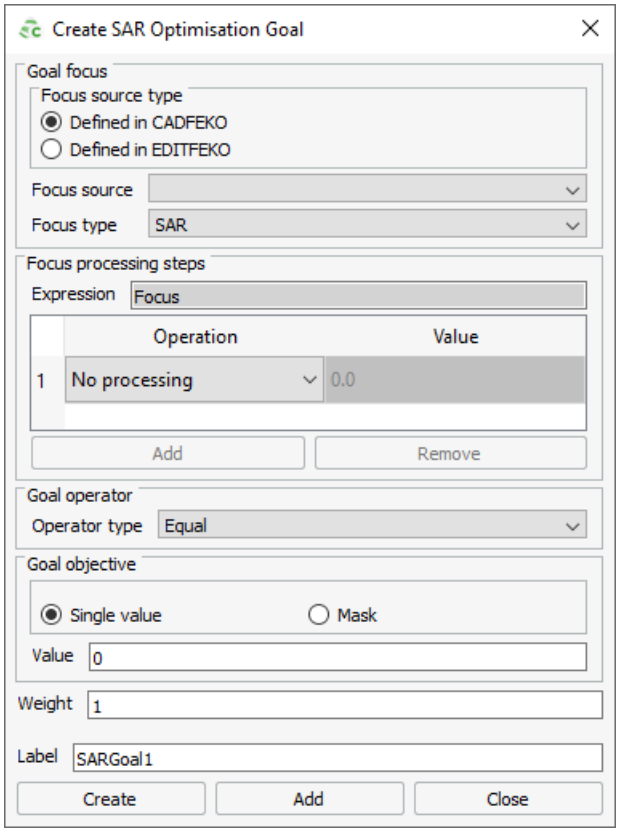


*Figure 556: The **Create SAR Optimisation Goal** dialog.*

The **Focus source** is identified based on the label of a **SAR** request in CADFEKO or the **Focus source label** of an SA card in EDITFEKO. The SAR focus delivers a non-complex value (or an array of non-complex values) based on the Feko solution.

Related reference
SA Card

Power Goal

Optimise the power that is solved as part of the Feko model.

On the **Request** tab, in the **Optimisation** group, click the **Add Goal Function** icon. From the drop-down list, select **Power Goal**.

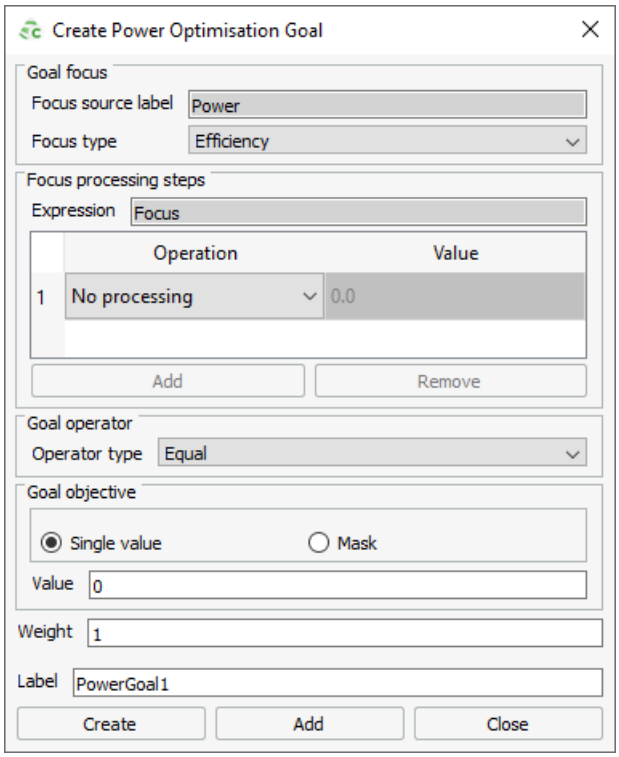


*Figure 557: The **Create Power Optimisation Goal** dialog.*

The **Power** goal allows optimisation of the total antenna efficiency, total power and power loss. The power goal does not accept any request name since it operates on the total power in the model.

Receiving Antenna Goal

Optimise the receiving antenna that is solved as part of the Feko model.

On the **Request** tab, in the **Optimisation** group, click the **Add Goal Function** icon. From the drop-down list, select **Receiving Antenna Goal**.

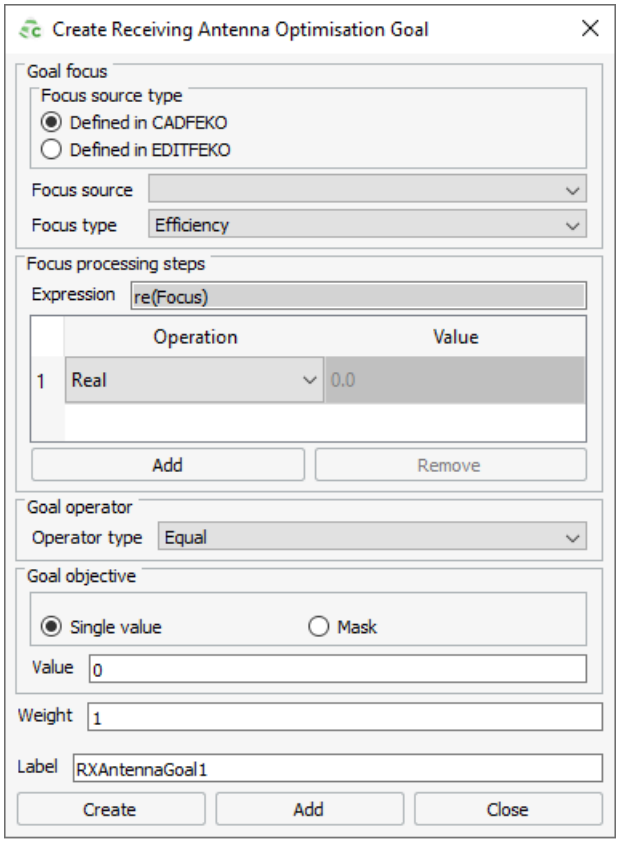


*Figure 558: The **Create Receiving Antenna Optimisation Goal** dialog.*

The **Focus source** is identified based on the label of a **Receiving antenna** request in CADFEKO or the **Focus source label** of a RA card in EDITFEKO.

Related reference
RA Card

Transmission / Reflection Goal

Optimise the transmission and reflection quantities that are solved as part of the Feko model.

On the **Request** tab, in the **Optimisation** group, click the **Add Goal Function** icon. From the drop down list, select **Transmission / Reflection Goal**.

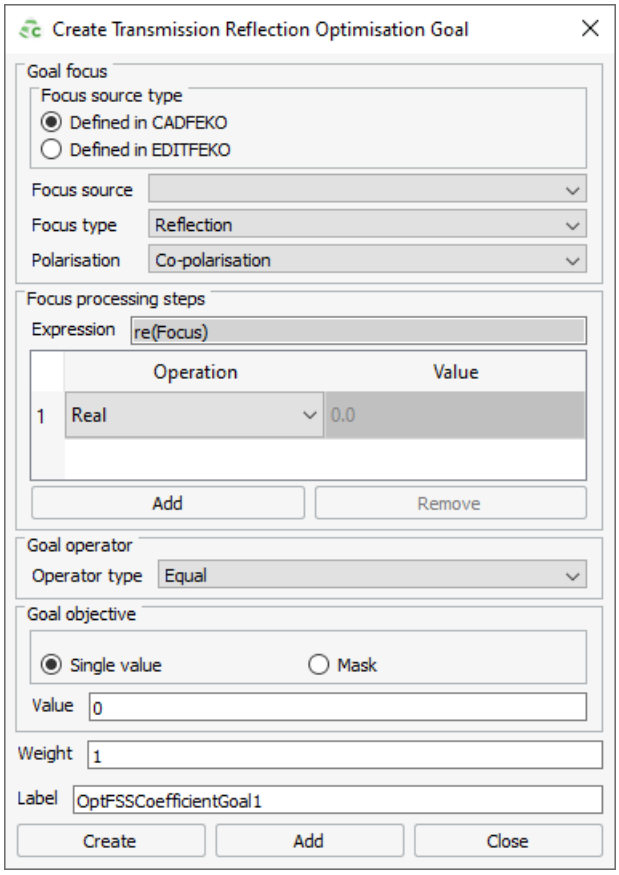


*Figure 559: The **Create Transmission Reflection Optimisation Goal** dialog.*

The **Focus source** is identified based on the label of a **Transmission/Reflection coefficient** request in CADFEKO or the **Focus source label** of a TR card in EDITFEKO.

Focus Type

Transmission

The transmission coefficient is calculated as

$$\tau = \frac{E_t}{E_i} \tag{88}$$

Reflection

The reflection coefficient is calculated as

$$\rho = \frac{E_r}{E_i} \tag{89}$$

Polarisation

Choose to optimise **Co-polarisation** or **Cross-polarisation**.

Related reference

TR Card

6.8.3 Focus Processing Options

Specify what operation should be performed on the selected goal.

The following processing steps are common to all goals.

No processing

Where the focus is non-complex, no processing steps are required. In order to consider the focus directly, the **No processing** option is provided.

$$x \rightarrow x \tag{90}$$

Real/Imaginary/Magnitude/Phase

Selects a specific component of a complex focus type. For an array, the complex component of each array element is taken, delivering a non-complex array.

$$x \rightarrow \mathrm{Re}(x)\ ;\ x \rightarrow \mathrm{Im}(x)\ ;\ x \rightarrow \mathrm{Phase}(x)\ ;\ x \rightarrow \mathrm{Mag}(x) \tag{91}$$

Unwrap

Unwraps a phase component. For a phase array, the whole array is considered in the unwrap process. This operator is applied directly after selecting **Phase**.

$$x \rightarrow \mathrm{unwrap}(x) \tag{92}$$

Absolute value

Takes the absolute value. For an array, the absolute value of each element is taken.

$$x \rightarrow |x| \tag{93}$$

Average/Minimum/Maximum

Finds the average, minimum or maximum value of an array. This has no effect on a single value.

$$x \rightarrow \mathrm{ave}(x)\ ;\ x \rightarrow \mathrm{min}(x)\ ;\ x \rightarrow \mathrm{max}(x) \tag{94}$$

Normalise

Normalises to the largest value in an array. For a single value, "1" will be returned.

$$x \rightarrow \frac{x}{\mathrm{max}(x)} \tag{95}$$

Log

Takes the base-10 logarithm. For an array, the base-10 logarithm of each element of the array is taken. This operator is only available for non-complex values or arrays.

$$x \rightarrow \log_{10}(x) \tag{96}$$

Offset

Adds a specified non-complex value. For an array, the value is added to each element of the array. This operator is only available for non-complex values or arrays.

$$x \rightarrow x + n \tag{97}$$

Scale
Multiplies by a specified scale factor. For an array, each element of the array is multiplied by the scaling factor.

$$x \rightarrow nx \tag{98}$$

Exponent
Applies an exponent. For an array of values, the exponent of each value in the array is taken.

$$x \rightarrow x^n \tag{99}$$

Undefined
When a processing step is modified and the step becomes invalid, the processing step reverts to an **Undefined** state. Delete or redefine all **Undefined** steps before applying the changes to the goal.

6.8.4 Goal Operator

Specify how the focus is compared with the goal objective and the goal operator.

There are five operator types that are common to all goals.

Equal
Indicates that the processed focus should be equal to the object.

$$\text{error} = \sum_{n=1}^{N} \left| \text{Focus}(n) - \text{Objective} \right| \tag{100}$$

Greater than
Indicates that the processed focus should be greater than the objective.

$$\text{error} = \sum_{N}^{n=1} \begin{cases} \left| \text{Focus}(n) - \text{Objective} \right| & \text{for Focus} < \text{Objective} \\ 0 & \text{for Focus} \geq \text{Objective} \end{cases} \tag{101}$$

Less than
Indicates that the processed focus should be less than the objective.

$$\text{error} = \sum_{N}^{n=1} \begin{cases} \left| \text{Focus}(n) - \text{Objective} \right| & \text{for Focus} > \text{Objective} \\ 0 & \text{for Focus} \leq \text{Objective} \end{cases} \tag{102}$$

Maximise
Indicates that the processed focus should be maximised (no objective is required for this operator).

Minimise
Indicates that the processed focus should be minimised (no objective is required for this operator).

When a goal is evaluated, a single value error representation of the goal is extracted according to the operator type. When the focus remains an array after the processing steps, an error is evaluated at

each point in the array, and the cumulative error is taken. For the comparative operator types (**Equal**, **Greater than** and **Less than**), where the relationship between the focus and objective satisfies the operator, the contribution to the error representation is zero.

6.8.5 Goal Objective

Choose between a single value or range of values defined by a mask to which the required output is compared with.

Single Value Objective

This objective is defined in the **Value** text-box. The optimisation error (convergence accuracy) is evaluated by comparing this value to the processed focus value according to the defined operator. Where the focus remains an array after the processing steps are applied, the objective value is compared to each of the array values separately, and the cumulative error is extracted according to the operator type by a summation of all of the errors.

Mask Objective

A 2D mask may be predefined and used as the objective of an optimisation goal. This allows for the comparison of an array of calculated data with a predefined array in the evaluation of the fitness of the optimisation step. This type of objective is typically used when a quantity varies with position, observation angle or frequency within one optimised simulation result. The length of the mask array is not required to be the same length as the computed data array it is compared to. The optimiser uses a piece-wise linear fitting on the mask array to determine the values for comparison with the correct points (output points as calculated according to the solution setup).

6.9 Global Goal: Combining and Weighting of Multiple Goals

Specify the method for combining multiple goals. A weight or importance factor is assigned to the combined goal and optimised according to the combination type, for example, the average of the goals combined for the specified data.

6.9.1 Combining Goals

Use the goal combination tool to extract a single error value from a set of goals.

On the **Request** tab, in the **Optimisation** group, click the **Combine Goals** icon.

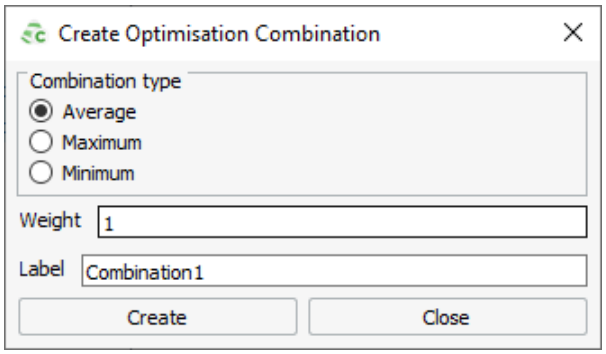


*Figure 560: The **Combine goal** dialog.*

The extraction type can be chosen as **Maximum**, **Minimum** or **Average**. When a set of goals are combined using this tool, only the minimum, maximum or average value of all of the errors of all of the goals in the set is taken.

In order to combine goals using the combination tool, goals in the same search in the same tree level should be selected. The **Combine goals** dialog (see Figure 560) is launched in which the **Combination type** is chosen. Goals are added to an existing combination by right-clicking on the combination in the tree, and selecting the type of goal to add. Goals are removed from the combination by deleting them. If all goals in a combination are deleted, then the combination is automatically removed. Goals can be copied out of a combination to the root of the goals tree by opening the right-click context menu for the particular goal and selecting **Copy**.

The **Average**, **Minimum** and **Maximum** options define how the evaluated errors of the goals in the combination should be reduced to one error value. For example, if **Average** is chosen, then the average error of the goals in the combination are returned, while **Maximum** returns the maximum error. Each combination is assigned a weighting that indicates how the error should be combined with other goals and combinations in the same level of the tree during the global fitness evaluation. The combination tools may be nested to as many levels as required.

6.9.2 Goal Weighting

Specify a weighting for the combination of multiple goals.

This weighting is used to modify the contribution of the combination goals error to the global error during the fitness evaluation. The global error in each level of the tree is computed by taking the evaluated error of each goal, multiplying it by the indicated weighting factor, and then summing all of the resultant weighted errors in each branch-level of the tree.

Note: The weighting of each goal is shown in brackets in the model tree.

6.10 Optimisation Using .PRE File modifications

Use optimisation with modifications made in the `.pre` file in EDITFEKO.

The optimiser operates on solution requests specified in CADFEKO. However, for advanced users, it is possible to make use of the optimiser after making modifications to the `.pre` file in EDITFEKO.

The optimiser operates on the labels of the solution requests. These labels are usually created in CADFEKO, but are copied to the `.pre` file. The optimiser reads the labels from the `.pre` file.

Consider the patch antenna on a finite substrate in Figure 561.

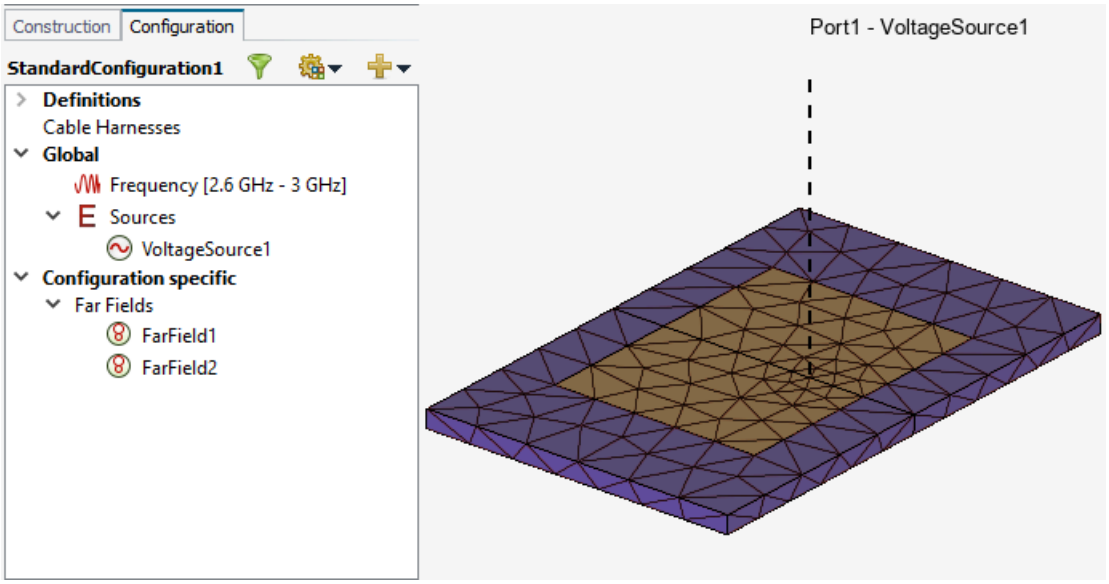


Figure 561: A patch antenna on a finite substrate fed with a voltage source on a wire port.

The antenna is fed with a voltage source on a wire port. Unless edited by the user, the voltage source is created with a default label of **VoltageSource1** in CADFEKO. When the CADFEKO model is saved, a `.pre` file is created by CADFEKO.

To open the `.pre` file, run EDITFEKO from within CADFEKO.

In EDITFEKO the `.pre` file contains an A1 card with the label, "MyCustomVoltageSource1" (label following after "**"), see Figure 562.

```
A1: 0 : Line1 :  :  :  : 1 : 90 **MyCustomVoltageSource1
```

Figure 562: Screenshot of the A1 card (voltage source on a wire segment) in the `.pre` file.

New sources and output requests can be added to the `.pre` file. For each request that will be used for optimisation, a unique label must be specified. After the `.pre` file is edited and saved, the optimisation setup is completed in CADFEKO by manually entering the label of the parameter to be optimised in the **Focus source label** text box in the **Goal focus** group.

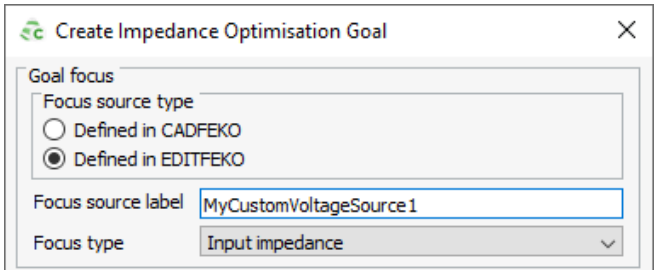


Figure 563: A snippet of the impedance optimisation goal dialog using a custom label.

Related reference

A1 Card

6.11 Optimisation Solver Settings

Make adjustments to the optimisation settings for a more computationally efficient solution.

On the **Solve/Run** tab, in the **Run/Launch** group, click the dialog launcher.

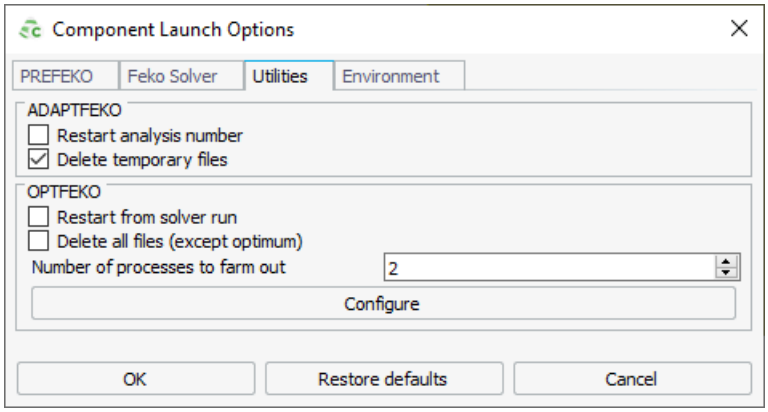


*Figure 564: The **Component Launch Options** (**Utilities** tab) dialog.*

Special options related to OPTFEKO are set on the **Utilities** tab, **OPTFEKO** group. These settings are as follows:

Restart from solver run
: This option may be used if a previous optimisation was interrupted. If this option is selected, then OPTFEKO will attempt to restart the optimisation process from the iteration number provided in the **Restart analysis number** text box. The optimisation can only be restarted if the temporary files have been kept during a previous optimisation run, see **Delete all files (except optimum)** below. If solution files are missing for a specific optimisation iteration, OPTFEKO runs the Feko solver to recreate the missing files. If any changes have been made to the model, solution or optimisation settings, OPTFEKO ignores all existing results, and re-compute all results as required.

Delete all files (except optimum)
: If this option is selected, then all of the temporary files are deleted during the optimisation process. When the optimisation process is completed (or if the optimisation process is interrupted), the original model, as well as the optimum are available along with all related simulation results. The optimum model and results are indicated by the addition of the string (`_optimum`) at the end of the file names. If this option is unchecked then no model or result files are deleted during the optimisation process.

> **Note:** This option must be unselected in order to use the **Restart from solver run** option (above).

Number of processes to farm out
: This option allows the specification of the distributed computing system when farming out the solutions during an optimisation. The **Configure** button launches the **Machines configuration** dialog where the machines in the cluster as well as the number of processes to be launched on each machine is specified. This dialog is identical to cluster configuration for parallel launching.

Feko Utilities

7

The Feko utilities consist of PREFEKO, OPTFEKO, ADAPTFEKO, the Launcher utility, Updater and the crash reporter.

This chapter covers the following:

7.1 The Preprocessor PREFEKO

Use PREFEKO to perform meshing and to prepare the input files for the Feko solver.

The component PREFEKO performs three tasks:

1. PREFEKO creates the mesh for the Feko solver based on geometry input from the user.
2. PREFEKO imports meshed geometry, usually constructed in CADFEKO.
3. All the mesh and requested control and output requests specified by the user is integrated by PREFEKO into the final Feko input file.

With regards to meshing PREFEKO subdivides surfaces into elementary surfaces (usually triangles) while wires are subdivided into segments. The mesh size (density) is dependent on the wavelength and medium parameters, which should be specified by the user.

This section describes the principal workings of the PREFEKO component. Assuming the user is specifying the geometry in a `.pre` file (usually with EDITFEKO), the user first defines the location of points in space with the DP card. Structures are then defined in terms of these points. For example, two points may be joined to form a line (BL card), or four points for a parallelogram (BP card).

7.2 Running PREFEKO

Use PREFEKO with the correct syntax and optional parameters for advanced control.

PREFEKO creates a `.fek` file ready for solving by the Feko solver from a `.pre` input file. PREFEKO is started using the following command:

```
prefeko example
```

where `example` is the `.pre` input file.

The component PREFEKO allows a number of options, which are mainly used for debugging purposes. Entering PREFEKO without arguments will give an overview of the syntax and supported options.

The options available for PREFEKO are as follows:

--version Print the version information and then exit.

--fek-format x Write the `.fek` file in the xth file format.

-#var=value Set a variable #var to the value `value`.

> **Note:**
>
> When a variable is set and evaluated using the command line and then a `.pre` file is run where the same variable is defined with a different value, it will override the value set using the command line.
>
> A solution is to check in the `.pre` file if the variable was defined from the command line using, for example:
>
> ```
> !!if not defined(#var1) then
> #var1 = 1
> !!endif
> ```

--ignore-errors Treat error messages as non-fatal. PREFEKO will continue with the processing after encountering errors. This can result in more errors as a consequence of the first one, but it could also be useful to see all geometry modelling errors at once, and not only the first one.

--print-variables Print a list of all variables (name, value, comment) to stdout. The output also includes info whether the variable is set for the first time or whether the value of an existing variable is changed.

--print-variables-to-out Print a list of all variables (name, value, comment) to the Feko output file (`.out`). The output also includes info whether the variable is set for the first time or whether the value of an existing variable is changed.

When defining variables from the command line, for example calling PREFEKO with

```
prefeko filename -#variable1=value1 -#variable2=value2 ...
```

it is recommended to use the `!!print_to_out` command to write these variables to the output file in order to keep a record of their values.

Related concepts

`!!print_to_out`

7.3 The Solver

The Solver is the electromagnetic solver component that calculates the specified output requests.

7.3.1 Running the Sequential Version

Run the sequential version of Feko with optional parameters.

It is recommended to run the Feko kernel directly from the GUI components CADFEKO, EDITFEKO or POSTFEKO. Once a session or model has been loaded, the sequential Feko solver can be started from the **Solve/Run** tab, by selecting **Feko** (the shortcut key Alt+4 can also be used).

While the Feko kernel is running, the status of the calculation phases is indicated on the **Executing runfeko** dialog, see Figure 565. The output generated by the Feko kernel is hidden by default. The Feko kernel output may be viewed by clicking on the **Details** button and selecting the **Output** tab. Similarly, notices, warnings and errors can be viewed by selecting the **Notices**, **Warnings** and **Errors** tabs respectively.

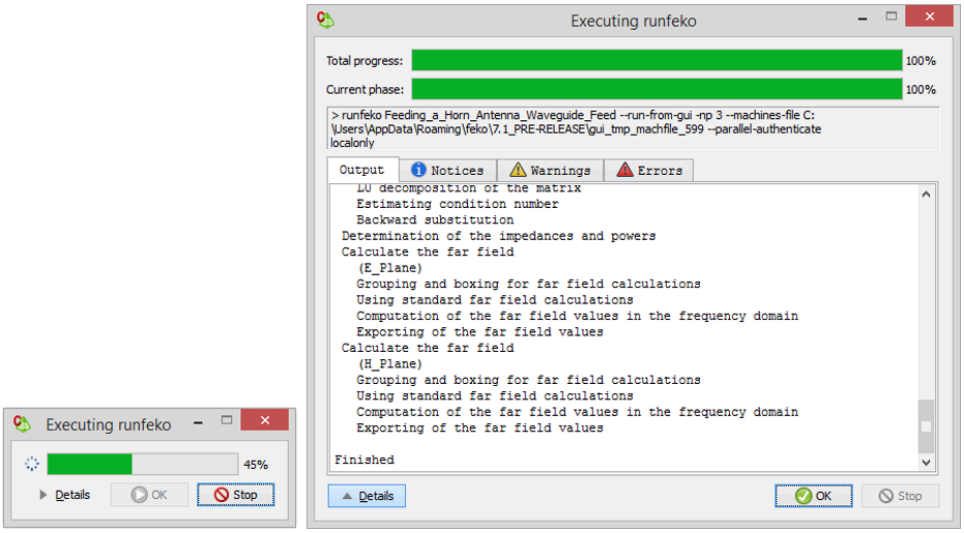


*Figure 565: The **Executing runfeko** dialogs. Output generated by the kernel is hidden by default. Solver output may be viewed by clicking on the **Details** button.*

When the Feko kernel is not executed from within the GUI, it can be started in a command window (on a Windows PC) or a shell (in UNIX) by executing the command:

```
runfeko example08
```

where `example08` must be a valid Feko input file (either `.cfx` or `.pre`/`.cfm` or `.fek` etc., there are internal time checks to run `cadfeko_batch` and/or PREFEKO as required to generate missing files or replace older ones).

RUNFEKO accepts the optional parameters listed below. More information regarding additional options for launching and controlling the parallel version of the solver can be found in Running the Parallel Version. Additional options for the remote launching of Feko are found in Running on a Remote Host.

In CADFEKO these settings are available by selecting the **Solve/Run** tab and clicking on the dialog launcher button on the **Run/launch** group. For POSTFEKO, select the **Home** tab and click on the dialog launcher button on the **Run/launch** group.

--version	Print the version information and then exit.
--priority x	The value x specifies the CPU usage priority of the Feko run: 0 = idle 1 = below normal 2 = normal 3 = above normal 4 = high If not specified, the default is 2. This option might not be available for specific systems or for specific Feko versions. In this case it is just ignored.
--use-gpu	[NUM_GPUS][:GPU_1[,GPU_N]] Execute Feko using GPU acceleration. The optional parameters are: NUM_GPUS: The number of devices to use. GPU_1 ,GPU_N: A comma separated list of specific devices to use. If the option is specified without the optional parameters, all available GPU resources are used. If NUM_GPUS is specified, the first NUM_GPUS devices in the system will be selected. Specifying `--use-gpu 0` will completely disable GPU detection and prevent NOTE 35179 from being printed. Example usage is as follows: `--use-gpu 2:0,2` which uses the first (device 0) and third (device 2) GPU in the system. This is equivalent to `--use-gpu :0,2`.
--remote-use-mpi	Activates the MPI method on Windows.
--execute-cadfeko_batch	Always execute CADFEKO_BATCH first (by-pass automatic checks based on file existence and date stamps.)
--no--execute-cadfeko_batch	CADFEKO_BATCH will not be run to create a new `.cfm` and `.pre` file.
--execute-prefeko	Always execute PREFEKO even if the existing `.fek` file is newer than the `.pre`.

--no--execute-prefeko PREFEKO will not be run to generate a new `.fek` file before the Feko solver is launched, even if the `.fek` and/or `.cfm` files are older than the existing `.fek` file.

--use-job-scheduler Run the parallel Feko kernel within a queuing system and obtain the number of parallel processes as well as the host list directly from the respective job scheduler.

> **Note:**
>
> The Intel MPI library supports the following job schedulers:
>
> *Microsoft Windows*
>
> - Altair PBS Professional
> - Microsoft HPC Pack
>
> *Linux*
>
> - Altair PBS Professional
> - Torque
> - OpenPBS
> - IBM Platform LSF
> - Parallelnavi NQS
> - SLURM
> - Univa Grid Engine

-d Debug mode with extra output (can be used to troubleshoot errors).

--prefeko-options All options following, up to the next –xxx-options, are passed to PREFEKO.

--feko-options All options following, up to the next –xxx-options, are passed to Feko.

--adaptfeko-options All options following, up to the next –xxx-options, are passed to ADAPTFEKO.

The optional command line parameters for Feko (specified after `--feko-options`) are listed below.

--check-only Feko processes and checks the model, but does not start a solution. This is useful to, for example, check an input file on a local computer before submitting it to a cluster.

--estimate-resource-requirements-only Feko processes the model and provides an estimate for the memory consumption. The estimated value is provided at the end of the `.out` file.

Note: An estimate is only available for:

- MoM
- MLFMM
- PO (not hybridised with any other solution methods).

Tip: For a more accurate estimate, run the estimation with the intended number of processes on the intended host(s).

-e ENV=value	This has the same effect as starting Feko with the environment variable ENV set to `value`. More than one `-e …` argument is allowed.
--data-export-format n	Use the nth version format for the data export files (`.efe`, `hfe`, `.ffe`, `.os`, `.ol`). Allowed values for n are `1` and `2` where `2` is the latest version (since Feko 6.1). If not specified, the default is to use the latest supported version.
--mtl-circuit-export	Special execution mode to export SPICE MTL circuit files.

Related concepts
How to Estimate Memory Requirements for the MLFMM

7.3.2 Running the Parallel Version

Run the parallel version of Feko with optional parameters for an efficient solution.

The parallel version of Feko may be used on any system that is licensed to run multiple Feko processes concurrently. If a system has a multi-core CPU (for example a quad-core CPU) then a sequential Feko licence will allow a parallel solution with up to 4 parallel processes to be launched. For systems with multiple CPUs (for example, a system with 2 separate dual-core CPUs) a 2-CPU parallel Feko licence will be required in order to run parallel solutions using all 4 of the available cores.

In order to use the parallel version of Feko from the GUI, it is required to configure the host names and number of processes that will be used for each node. (This is initially be set up during installation of Feko, meaning that reconfiguration is only necessary if changes are made.)

On the **Solve/Run** tab, in the **Run/Launch** group, click the dialog launcher.

On the **Feko Solver** tab, under **Parallel execution**, click the **Configure** button. In the dialog (see Figure 566), the host names and number of processes to be started on each host must be entered. Usually one process per available core on each machine should be chosen, for example 4 processes for a quad-core machine. It is also possible to use this option to implement a crude load balancing system: running more processes on hosts with faster CPUs or more memory. Nodes may be added or removed from the current cluster setup using the **Add** and **Remove** buttons respectively.

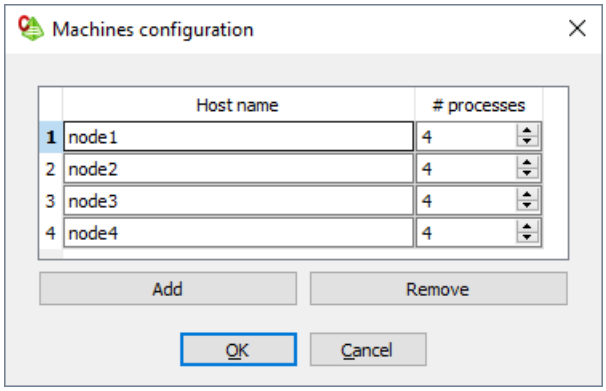


*Figure 566: The **Machines configuration** dialog for the parallel host configuration.*

Notice: For parallel Windows PC clusters, Feko must be installed at the same location on each host.

It is recommended that the parallel job is started from a PC that forms part of the cluster and that this host is listed first.[77]

After clicking **OK** the hosts are saved to a `machines.feko` file in the directory specified by the environment variable FEKO_USER_HOME. This file is then used in the actual parallel process launching.

77. It is possible to launch the job without including the local machine. The `.fek` input file must be located on the first PC in the list and the `.out` and `.bof` output files are created on this PC in the same directory as the project directory on the local machine. It is the user's responsibility to transfer the files between the local machine and the first machine in the list if these are not the same. Alternately remote parallel launching can be used where Feko does this copying explicitly.

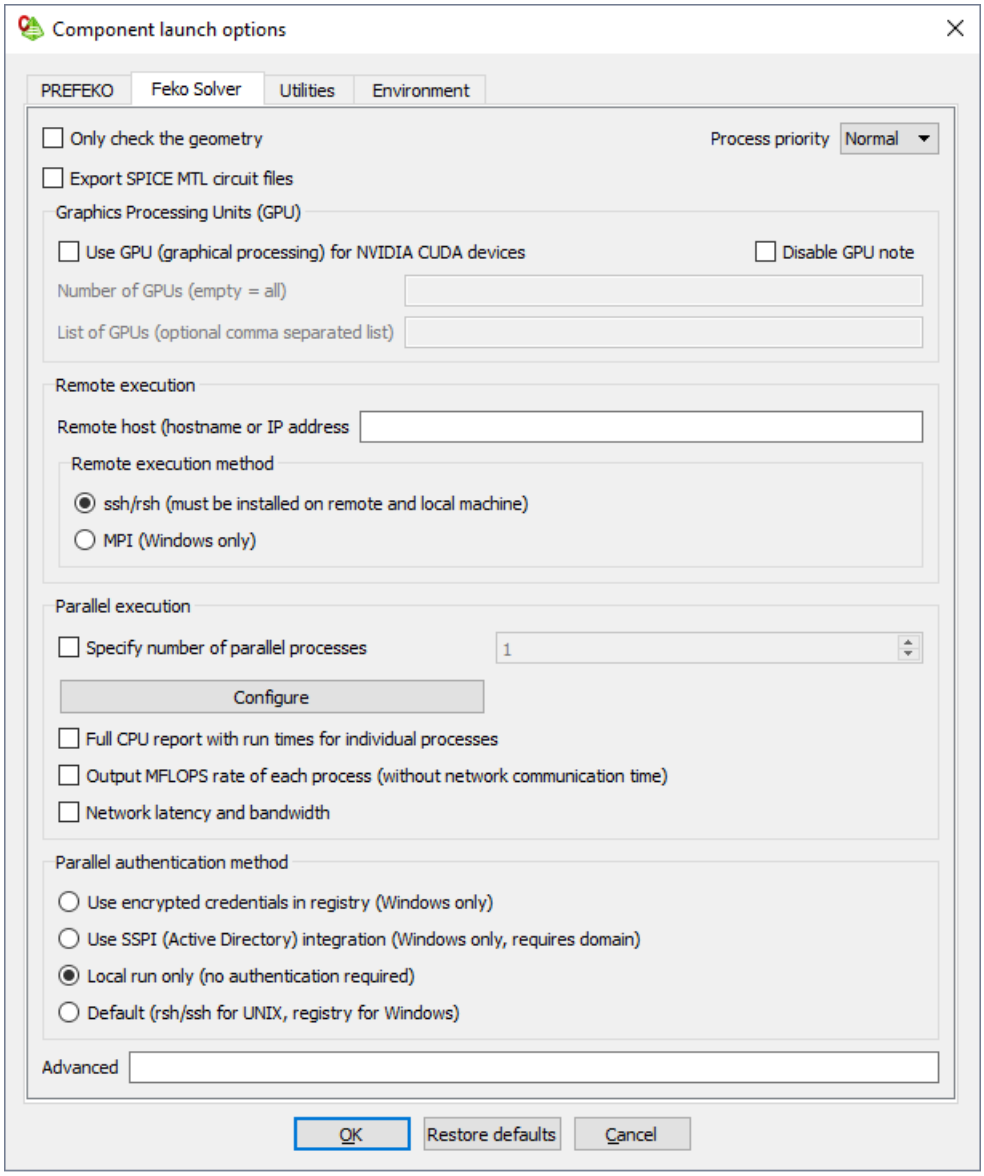


*Figure 567: The **Component launch options** dialog.*

On the Feko tab of the **Component launch options** dialog (shown in Figure 567), further settings with regards to the Feko solution can be made.

Tip: Use the **Output MFLOPS rate** . . . and **Network latency and bandwidth** options to ensure an optimum configuration for the nodes.

Feko will print a table giving the performance of the various nodes. These checks are repeated each time Feko calculates the solution.

CAUTION: A significant amount of time may be required if the test file contains multiple frequencies. These options should therefore not be kept selected after the initial setup, except for debugging purposes.

The target priority of the Feko run may also be set on this tab. Setting the priority below normal will allow other interactive work on the same computer. However, all machines in the cluster operate at the speed of the slowest node - starting other CPU-intensive jobs on one of the nodes in a cluster is generally not recommended.

In order to use parallel solving after it has been set up, do the following: On the **Solve/Run** tab, in the **Run/Launch** group, click the  **Parallel** icon. A check mark will be displayed next to the menu option. Any Feko solver runs that are launched while this option is checked will use the parallel version of Feko.

From the command line (for example on a UNIX workstation), the parallel Feko version is started as follows:

```
runfeko example1 -np x
```

where the parameter `x` following `-np` indicates the required number of processes to be used in the parallel solution. In addition to the arguments listed in Running the Sequential Version, the parallel version accepts the following optional parameters:

-np x	Start the parallel Feko version with `x` processes. The `-np all` option is also supported when all available processors in the machines file should be used.
--machines-file machname	The file `machname` is the machines file with the node names and the number of CPUs (see below).
--mpi-options ...	Unless another `--xxx-options` parameter is used, all options following this are passed to the MPI launcher (for example mpirun or mpiexec).
--parallel-authenticate <method>	Sets the authentication method to be used for parallel Feko runs. The following authentication methods are available: default: Platform dependent default (same as if option not specified). localonly: Run the parallel job on local host only and thus no authentication is required sspi: Windows Active Directory (SSPI) authentication is used. This option is available on Windows only. registry: Encrypt the credentials (username and password) into the registry. This option is available on Windows only.

78. For more details refer to the Intel MPI and MPICH documentation in mpi\<platform>\<mpi-version>\doc directory of the Feko installation.
79. Note that additional (one time) configuration settings might be required by the domain administrator to prepare the Windows domain for this kind of authentication requests.

The number of processes to launch on each available host is specified in a machines file with the following general syntax:

```
Hostname:Number of processes
```

For example assume that `host1` has 4 processors and `host2` has 8, then the machines file will be as follows:

```
host1:4
host2:8
```

With this machines file, if 6 parallel processes are requested then Fekowill use 4 processes on host1 and 2 processes on host2.

If only one process is to be started on any host, then instead of the entry `host3:1` in the machines file, the shorter form `host3` may be used.

The machines file (`machines.feko`) is located in `%FEKO_HOME%\shared\mpi` and is automatically created during the installation of the parallel version of Feko. This file is the default machines file used by Feko. If a different distribution of the processes is required, this file can be manually edited (however this action is strongly discouraged).

> **Tip:** Create a separate machines file with the syntax described above if a different number of parallel processes are required than what was specified during the installation.

The environment variable `FEKO_MACHFILE` can be used to force RUNFEKO to use this file instead of the default. The required commands assuming the desired machines file is, for example `machname`, are as follows (for the sh shell):

```
FEKO_MACHFILE=./machname
export FEKO_MACHFILE
runfeko example1 -np 6
```

Alternatively the name of the machines file can be passed as an argument to RUNFEKO on the command line as follows:

```
runfeko example1 -np 6 --machines-file ../../machname
```

Using RUNFEKO is independent of the respective platforms and MPI implementations (the discussion of the environment variable FEKO_WHICH_MPI contains more information). For certain applications or experienced users it may be necessary to pass additional options to MPI[79]. These options are added after the argument `--mpi-options`. For example on a ScaMPI cluster (assuming FEKO_WHICH_MPI=6), the call

```
runfeko example_08 -np 6 --mpi-options -immediate_handling \
threaded -smtrace 5-6
```

(all on one line) is interpreted internally and Feko is executed with the command

```
/opt/scali/bin/mpimon -export env -immediate_handling threaded \
-smtrace 5-6 /opt/feko/bin/feko.csv example_08 -- host1 4 \
host2 2
```

Note: `host1` and `host2` are examples only—the actual information is taken from the machines file.

In addition to using the `--mpi-options` command line option, the MPI environment can be controlled by setting certain environment variables. For instance, when using Intel MPI the environment variable I_MPI_DEVICE[80] is quite important to control which device should be used (sockets or shared memory, or RDMA device). Such environment variables is set up internally by means of Lua initialisation scripts. See the Feko Installation Guide for more information.

Feko employs shared memory extensively for parallel runs. On Linux shared memory uses a bind mount to the `/dev/shm` partition and its size is set to 50% of the physical memory by default. The exact size can be queried using the `df -h` command.

If the shared memory size limit is exceeded during a parallel Feko solver run, it could lead to an error message:

```
Bus error.
```

Tip: Adjust the shared memory size temporarily using the mount command, or permanently by editing the `/etc/fstab` file.

7.3.3 Running on a Remote Host

Run Feko remotely with automatic file transfer between the local machine and host.

Remote launching allows for example the user to run the Feko GUI on a local Windows PC, but start a sequential or parallel Feko job directly from one of the GUI components or a terminal on a remote workstation or cluster. There are two main mechanisms for remote launching: The SSH/RSH based method and the MPI based method.

The SSH/RSH method
: This remote launching method is cross platform capable, for example, it is possible to launch a remote job from a Windows PC on a UNIX workstation or vice-versa. In order to use the remote launching facility, SSH must be available with public key authentication.

The MPI method
: This method is currently only available between Windows hosts. It is based purely on Windows commands and relies on a network share for copying files and uses the MPI daemon (as shipped with Feko) for starting the remote process. Also for this method to work properly, the related option must have been selected during installation of Feko on the remote machine. This consists of creating a shared network directory.

For more information regarding the setup requirements for remote launching using either method, please see the detailed installation and setup instructions in the Feko Installation Guide.

80. For more details refer to the Intel MPI and MPICH documentation in mpi\<platform>\<mpi-version>\doc directory of the Feko installation.

General settings and usage

On Windows and Linux, this remote launching facility can be used directly from within the GUI components, CADFEKO, EDITFEKO or POSTFEKO. As described for parallel launching, open the **Component launch options** dialog.This dialog is shown in Figure 567. Enter the hostname or IP address of the remote host in the **Remote host** input field under **Remote execution** and select the appropriate **Remote execution method**. In order to use remote launching after it has been set up, do the following: On the **Solve/Run** tab, in the **Run/Launch** group, click the  **Remote** icon. A check mark will be displayed next to the menu option. Runs of the Feko solver (either sequential or parallel if **Parallel** is also checked) will employ Remote Feko execution on the remote host while this option remains checked.

In order to use the remote launching facility from the command line, the following command can be used:

```
runfeko example1 --remote-host h
```

The parameter `h` following `--remote-host` gives the host name or the IP address of the remote host. This will automatically use the SSH based remote launching method.

In order to use the MPI based method, the following command can be used:

```
runfeko example1 --remote-host h --remote-use-mpi
```

This command line option of RUNFEKO may be combined with other options, for example using the following command:

```
runfeko example1 --remote-host h -np 4 --machines-file m
```

The above command would launch a parallel job with 4 processes using the nodes as listed in the machines file `m`, and the parallel job is then launched from the remote host `h` (typically the control node of a cluster).

As previously mentioned, the remote launching facility has an automatic file transfer feature included, negating the requirement to work on a shared network drive. On the remote host, Feko will create a temporary sub-directory in the user's home directory with the name `remote_FEKO_job_xxx` (`xxx` is a unique number) and all the Feko files will be placed there for the duration of the Feko solution. After the completion of the remote execution, all files will be copied back to the client and this temporary subdirectory on the remote machine will be removed.

Important notes regarding remote launching of parallel Feko jobs

- If a machines file is specified while launching the job locally, this will also be used on the remote host (it will be copied to the remote host). In this way a parallel job can be configured on the local client (on for example, two hosts node1 and node2) but the Feko solution can be launched remotely on another computer - which will then be the control node of the parallel solution. This makes sense when launching a parallel Feko job from a Windows PC on a Linux cluster.
- If no local machines file is specified when launching a remote solution it is important to note that default options as set on the remote host will be used. Therefore Feko will read the `machines.feko` file on the remote host, and not on the local host where jobs are launched.

In addition note that when launching a remote job from the GUI the machines file will always be present, but not from the command line unless explicitly included in the command. If the machines file is omitted the remote parallel hosts are then found using the default mechanism (from the environment variable FEKO_MACHFILE, default location for the file `machines.feko` and so forth). More information can be found in Running the Parallel Version).

7.4 OPTFEKO

OPTFEKO is the component that controls the optimisation process. The optimisation parameters are usually associated with geometric dimensions, material properties, excitations and loadings. For example, the gain of a horn antenna is maximised by varying the size of the horn aperture.

OPTFEKO requires two components for successful execution:

1. A parametric model that consists of at least a `.pre` file, or a `.cfx` file (or possibly both).
2. An `.opt` file that specifies how the model is optimised.

All options relating to optimisation are specified through the CADFEKO interface and is stored in the .opt file. The parametric model is prepared using CADFEKO and or EDITFEKO.

Optimisation is based on, comprising a number of parts:

- *Method* to be used for the search (including method settings regarding accuracy and stopping criteria).
- *Parameters* define the range in which the search will be performed.
- *Goals* specify the desired result of the optimisation process.

> **Note:** Multiple *Parameters* and *Goals* may be defined as part of a single *Search*. The goals are combined into a single representative function that is minimised or maximised.

7.4.1 Optimisation Methods

OPTFEKO provides various optimisation methods, each one with different characteristics.

Selecting the appropriate optimisation search method to apply to a given problem is not a trivial task. It is a function of the number and range of the parameters, the required outcome of the optimisation, the model size and the resources available.

Simplex (Nelder-Mead)

The Simplex Nelder-Mead Algorithm can be categorised as a local or hill-climbing search method, where the final optimum relies strongly on the specified starting point.

The geometric figure formed by a set of $N+1$ points in an N-dimensional space is called a simplex. The basic idea in the simplex method is to compare the values of the combined goals at the $N+1$ points of a general simplex (where each point represents a single set of parameter values) and move the simplex gradually toward the optimum point during an iterative process. The movement of the simplex is achieved using three operations: reflection, contraction and expansion.

The initial *simplex* in a 2-dimensional search-space is represented by the points X_1, X_2 and X_3.

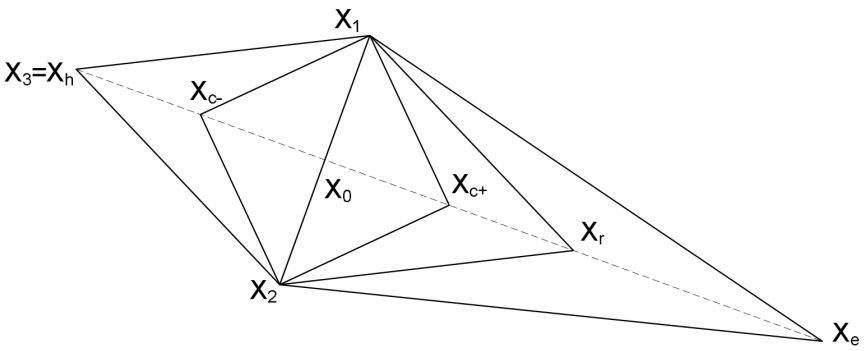


Figure 568: Reflection, expansion and contraction for the Simplex method.

Reflection

Consider the diagram in Figure 568. If X_h is the point corresponding to the poorest fitness value among the points of the initial simplex, it may be expected that the point X_r obtained by reflecting the point X_h around the axis defined by the other points in the simplex (X_1 and X_2) may (when evaluated according to the optimisation goals) provide a better fitness value. If this is the case, a new simplex can be constructed by rejecting the point X_h from the simplex and including the new point X_r. This process is illustrated in Figure 568 where the points X_1, X_2 and X_r form the new simplex. Since the direction of movement of the simplex is always away from the worst result, movement will always be in a favourable direction. If the global goal function does not have steep valleys within the space defined by the parameter ranges, repetitive application of the reflection process will lead to a zigzag path in the general direction of the optimum.

Expansion

If a reflection process finds a point X_r which is a better fitness than any point in the simplex (a new optimum point), it may be expected that the best fitness value may be improved even further by moving along the direction pointing from X_0 to X_r. An expansion is therefore performed from X_r to X_e.

If the evaluated fitness at X_e is better than the fitness at X_r, the expansion was successful; X_h his then replaced with X_e and the reflection process is restarted. If the evaluated fitness at X_e is poorer, the expansion attempt has failed; X_h is replaced by X_r (as identified in the previous reflection operation) and the reflection process is continued.

Contraction

If the reflection process finds a point X_r with a better fitness than the second-best point in the current simplex (X_{nh}), a contraction operation will be performed.

If the contraction process produces a point X_c which has a better fitness than a point in the simplex, the contraction was successful and X_h is replaced with X_c before continuing with the reflection process. If the contraction process produces a point X_c which has a poorer fitness, the contraction process has failed and the simplex base is reduced by scaling all the points in the simplex by an internal factor before restarting with the reflection process.

Table 63: A summary of the Simplex operations. The F(X) operator represents the evaluation of the fitness at the point X in the parameter space.

Objective Function	Operation
$F(X_r) < F(X_l)$	Expansion
$F(X_l) \leq F(X_r) < F(X_{nh})$	Reflection
$F(X_{nh}) \leq F(X_r) < F(X_h)$	Positive contraction
$F(X_h) \leq F(X_r)$	Negative contraction

Error treatment and termination

The Simplex Nelder-Mead terminates naturally when:

- The maximum number of Feko solver runs has been reached
- The standard deviation between the simplex vertices is small enough
- The simplex base is small enough
- The optimisation goal has been reached

Text log

During an optimisation, OPTFEKO maintains a text log of the optimisation process in the project `.log` file. The structure of this file is primarily determined by the optimisation method.

Section 1: General information regarding the optimisation setup.

```
=========================  L O G - FILE - OPTFEKO  =========================

Version: 13.22 of 2007-05-08
Date: 2007-06-06 16:45:43
File: test

OPTIMISATION WITH ALtair Feko

=============== Optimisation variables ===============

No.  Name                                  Beg.value          Minimum
 Maximum
  1 sigma                           3.503500000e+07  1.000000000e+07  5.000000000e
+08


=============== Optimisation goals ===============

No.  Name                              Expression
  1 search1.goals.nearfieldgoal1     nearfieldgoal1
```

Section 2: Information regarding the Simplex method parameters.

```
=============== Optimisation method: SIMPLEX NELDER-MEAD ===============
```

```
Maximum number of iterations:                        1000
Base of the simplex:                    1.500000000e-01
Reduction factor of the base:           5.000000000e-01
Termination at minimal base:            1.000000000e-03
Termination at standard deviation:      1.000000000e-04
Standard reflection coefficient (R):    1.000000000e+00
Contraction coefficient (-C, +C):       5.000000000e-01
Expansion coefficient (E):              2.000000000e+00
```

Section 3: Information regarding the parameter values, goal values and Simplex operations at each iteration.

```
=============== SIMPLEX NELDER-MEAD: Intermediate results ===============

 No.  sigma             nearfieldgoal1   global goal      operation   global best aim
   1  3.503500000e+07  6.488107157e-02  3.488107157e-02       -----  3.488107157e-02
   2  3.774988237e+07  5.929294284e-02  2.929294284e-02       -----  2.929294284e-02
   3  4.516707895e+07  6.328463622e-02  3.328463622e-02       -----
   4  4.788196133e+07  5.795280540e-02  2.795280540e-02           R  2.795280540e-02
   5  5.430544199e+07  5.500882669e-02  2.500882669e-02   E success  2.500882669e-02
   6  4.153550651e+08  3.036429062e-02  3.642906239e-04           R  3.642906239e-04
   7  4.356175335e+08  2.964433899e-02  3.556610122e-04   E success  3.556610122e-04
   8  4.457496125e+08  2.929870383e-02  7.012961666e-04           R
   9  4.356179559e+08  2.964446921e-02  3.555307926e-04  +C success  3.555307926e-04
```

Section 4: Information regarding the termination reason and optimisation results. If sufficient information was available for a sensitivity analysis to be completed, the results of the sensitivity analysis are also given.

```
=============== SIMPLEX NELDER-MEAD: Finished ===============

Optimisation finished (Standard deviation small enough:   5.020322005e-06)

Optimum found for these parameters:
  sigma                          =   4.356179559e+08

Optimum aim function value (at no.  9):   3.555307926e-04
No. of the last analysis:  9

Sensitivity of optimum value with respect to each optimisation parameter,
i.e. the gradient of the aim function at 1% variation from the optimum:
Parameter                        Sensitivity
  sigma                          8.344260771e-01
```

Particle Swarm Optimisation (PSO)

Particle swarm optimisation (PSO) is a population-based stochastic evolutionary computation technique based on the movement and intelligence of swarms. As a global search algorithm, the technique has, in certain instances, outperform other methods of optimisation like genetic algorithms (GA).

PSO can be best understood through an analogy similar to the one that led to the development of the PSO. Imagine a swarm of bees in a field. Their goal is to find in the field the location with the highest density of flowers. Without any a priori knowledge of the field, the bees begin in random locations with random velocities (direction and speed) looking for flowers. Each bee can remember the location at

which it found the most flowers, and is aware of the locations at which each of the other bees has found an abundance of flowers.

Based on its own experience (local best, *pbest*) and the known best position (global best, *gbest*) found so far, each bee in turn adjusts its trajectory (position and velocity) to fly somewhere between the two points depending on whether nostalgia or social influence dominates its decision. When each bee is done flying, it communicates its new-found information to the rest of the swarm who in turn adjust their positions and velocities accordingly.

Along the way, a bee might find a place with a higher concentration of flowers than it had found previously. It would then be attracted to this new location as well as the location of the most flowers found by any bee in the whole swarm. Occasionally, one bee may fly over a place with more flowers than have thus far been encountered by any bee in the swarm. The whole swarm would then be drawn toward that location in addition to the location of own personal best discovery. In this way the bees explore the field: overflying locations of greatest concentration, then being attracted back toward them. Eventually, the bees' flight leads them to the one place in the whole field with the highest concentration of flowers.

Population size and number of iterations

The default swarm / population size is set to 20 and the number of iterations to 50, resulting in a default maximum allowed number of Feko solver runs of 1000. While too small a swarm size prevents the search algorithm from properly traversing the parameter space, larger swarm sizes require more computational time. Compared to GA, the PSO technique tends to converge more quickly with smaller population sizes.

When the maximum number of solver runs, (C), is specified by the user, this needs to be converted into a population size (A) and number of iterations (B), with $A*B \leq C$. A is selected as a function of the number of parameters (N_p), with an internal upper limit, while the requirement that $B \geq 5$ must be satisfied.

Error treatment and termination

PSO terminates naturally when:

- The maximum number of Feko solver runs has been reached
- The standard deviation between the best positions of the swarm is small enough
- The optimisation goal has been reached

Failure during re-evaluation and meshing (in the CADFEKO batch meshing tool or in PREFEKO) for a specific set of parameters is treated by writing an appropriate error message to the `.log` file before computing a new parameter set to replace the failed one. If too many consecutive parameter set failures occur, then the optimisation will terminate with a message indicating this. The `.log` file for the optimisation can be consulted for further information.

Due to the nature of the technique, the parameters naturally adhere to boundaries defined in the parameter space.

The text log of the PSO method

During an optimisation, OPTFEKO maintains a text log of the optimisation process in the project `.log` file. The structure of this file is primarily determined by the optimisation method.

Section 1: General information regarding the optimisation setup.

```
========================  L O G - FILE - OPTFEKO  ========================

Version: 13.22 of 2007-05-08
Date: 2007-06-06 16:32:51
File: test

OPTIMISATION WITH Feko

=============== Optimisation variables ===============

No.  Name                                    Beg.value          Minimum
 Maximum
  1 zf0                                2.000000000e+00  1.000000000e+00  1.000000000e
+01



=============== Optimisation goals ===============

No.  Name                              Expression
  1 search1.goals.farfieldgoal1     farfieldgoal1
```

Section 2:

```
=============== Optimisation method: PSO ===============

Maximum number of iterations:                        3
Population size:                                     1
Acceleration constant 1:                2.800000000e+00
Acceleration constant 2:                1.300000000e+00
Termination at standard deviation:      1.000000000e-04
Pseudorandom number generator seed:                  1
```

Section 3:

```
=============== PSO: Intermediate results ===============

 No.  zf0              search1.goals.f  global goal      local best aim   global best
 aim
   1  2.000000000e+00  2.267123373e-01  7.732876627e-01  7.732876627e-01
 7.732876627e-01
   2  2.000000000e+00  2.267123373e-01  7.732876627e-01
   3  2.000000000e+00  2.267123373e-01  7.732876627e-01
```

Section 4:

```
=============== PSO: Finished ===============

Optimisation finished (Maximum number of analyses reached: 3)

Optimum found for these parameters:
  zf0                           =   2.000000000e+00

Optimum aim function value (at no. 1):   7.732876627e-01
No. of the last analysis: 3

Sensitivity of optimum value with respect to each optimisation parameter,
i.e. the gradient of the aim function at 1% variation from the optimum:
Parameter                          Sensitivity
```

```
  zf0                                8.344260771e-01
```

Genetic Algorithm (GA)

Genetic algorithm (GA) optimisers are robust, stochastic search methods modelled on the Darvinian principles and concepts of natural selection and evolution. Like the particle swarm optimisation (PSO) method, GA's are classified as global optimisers. Feko employs a real genetic algorithm (RGA).

GA optimisation borrows from the natural world in a number of ways. Conceptually, during a GA optimisation, a set of trial solutions (a generation) is chosen. This generation is assigned the role of "parents", from which a new generation of "children" are derived. In an evolutionary "survival-of-the-fittest process", each consecutive generation moves toward an optimal solution under the selective pressure of the fitness/goal function criteria.

Population size and number of iterations

As a default, the generation size for the GA method is set to 20 and the maximum number of iterations to 50, resulting in a maximum allowed number of Feko solver runs of 1000.

If the user specifies the maximum number of solver runs (), this needs to be converted into a generation size () and number of iterations (), with $A*B \geq C$. A is selected as a function of the number of parameters in the optimisation problem (N_p), with an internal upper limit. It is also internally required that B be chosen such that $B \geq 5$.

Error treatment and termination

The GA algorithm terminates naturally when:

- The maximum number of Feko solver runs has been reached
- The standard deviation between the current generation chromosomes is small enough
- The optimisation goal has been reached

Failure during re-evaluation and meshing (in the CADFEKO batch meshing tool or in PREFEKO) for a specific set of parameters is treated by writing an appropriate error message to the `.log` file before computing a new random parameter set to replace the failed one. Due to the nature of the technique, the parameters naturally adhere to boundaries defined in the parameter space.

The text log of the GA method

During an optimisation, OPTFEKO maintains a text log of the optimisation process in the project `.log` file. The structure of this file is primarily determined by the optimisation method.

Section 1: General information regarding the optimisation setup.

```
=========================  L O G - FILE - OPTFEKO  =========================

Version: 13.22 of 2007-05-08
Date: 2007-06-06 16:32:51
File: test

OPTIMISATION WITH Feko

=============== Optimisation variables ===============
```

```
No.  Name                                    Beg.value          Minimum
 Maximum
  1 zf0                                 2.000000000e+00  1.000000000e+00  1.000000000e
+01



=============== Optimisation goals ===============

No.  Name                              Expression
  1 search1.goals.farfieldgoal1     farfieldgoal1
```

Section 2: Information regarding the GA method parameters.

```
=============== Optimisation method: RGA ===============

Maximum number of iterations:                              3
Population size:                                           1
Creep mutation with probability:            5.000000000e-01
Elitism, i.e. best individual replicated into next generation
Enforce niching
Uniform crossover with probability:         5.000000000e-01
Termination at standard deviation:          1.000000000e-04
Pseudorandom number generator seed:                        1\
```

Section 3: Information regarding the parameter values, goal values and GA aims at each iteration.

```
=============== RGA: Intermediate results ===============

 No.  zf0              search1.goals.f  global goal      global best aim
   1  2.000000000e+00  2.267123373e-01  7.732876627e-01  7.732876627e-01
   2  1.357202892e+00  2.267123373e-01  7.732876627e-01
   3  1.095988516e+00  2.267123373e-01  7.732876627e-01
```

Section 4: Information regarding the termination reason and optimisation results. If sufficient information was available for a sensitivity analysis to be completed, the results of the sensitivity analysis are also given.

```
=============== RGA: Finished ===============

Optimisation finished (Maximum number of analyses reached: 3)

Optimum found for these parameters:
  zf0                          =   2.000000000e+00

Optimum aim function value (at no. 1):   7.732876627e-01
No. of the last analysis: 3

Sensitivity of optimum value with respect to each optimisation parameter,
i.e. the gradient of the aim function at 1% variation from the optimum:
Parameter                         Sensitivity
  zf0                             8.344260771e-01
```

Grid Search

The optimisation parameters are linearly varied between their minimum and the maximum values in a predefined number of steps.

This method is strictly speaking not an optimisation method. This can be useful to investigate the parameter space before beginning an optimisation. Due to the required computational time, it is not generally recommended that this method be applied for problems containing more than two parameters.

During application of the grid search method, optimisation goals are evaluated at each of the specified grid points, and a fitness is assigned to each evaluation. Though this fitness has no effect on the selection of the ensuing parameter set, an optimum result on the predefined grid will be identified and the solutions at each of the grid points can be compared to evaluate their performance based on fitness.

Error treatment and termination

Failure during re-evaluation and meshing (in the CADFEKO batch meshing tool or in PREFEKO) for a specific set of parameters is treated by writing an appropriate error message to the `.log` file before continuing with the next set of parameters in the grid.

The grid search method terminates naturally only when the maximum number of Feko solver runs has been reached. The number of solver runs can be computed based on the number of parameters and the number of steps per parameter.

$$N_{\text{solver runs}} = \prod_{i=1}^{N_{\text{parameters}}} N_{\text{points}}(i) \tag{103}$$

Internally, a limit of 10 000 is placed on the maximum number of allowed solver runs. For 4 parameters this would mean a maximum of only 10 points per parameter (this indicates how quickly the algorithm can scale in terms of the number of required solver runs for multi-parameter problems.)

The text log of the Grid search method

During an optimisation, OPTFEKO maintains a text log of the optimisation process in the project `.log` file. The structure of this file is primarily determined by the optimisation method.

Section 1: General information regarding the optimisation setup.

```
=========================  L O G - FILE - OPTFEKO  =========================

Version: 13.22 of 2007-05-08
Date: 2007-06-06 16:32:51
File: test

OPTIMISATION WITH Feko

=============== Optimisation variables ===============

No.  Name                                    Beg.value          Minimum
 Maximum
  1 zf0                                2.000000000e+00  1.000000000e+00  1.000000000e
+01
```

```
=============== Optimisation goals ===============

No.  Name                              Expression
  1 search1.goals.farfieldgoal1     farfieldgoal1
```

Section 2: Information regarding the grid search parameters.

```
=============== Optimisation method: GRID SEARCH ===============

No.  Name                          Quantity          Minimum          Maximum
   Step
  0 zf0                                   3  1.000000000e+00  1.000000000e+01
 4.500000000e+00
```

Section 3: Information regarding the parameter values and goal values at each step.

```
=============== GRID SEARCH: Intermediate results ===============

 No.  zf0              search1.goals.f  global goal
   1  1.000000000e+00  2.267123373e-01  7.732876627e-01
   2  5.500000000e+00  2.267123373e-01  7.732876627e-01
   3  1.000000000e+01  2.267123373e-01  7.732876627e-01
```

Section 4: Information regarding the termination reason and best results found on the search grid.

```
=============== GRID SEARCH: Finished ===============

Optimisation finished (Maximum number of analyses reached: 3)

Optimum found for these parameters:
  zf0                            =   1.000000000e+01

Optimum aim function value (at no. 3):   7.732876627e-01
No. of the last analysis: 3
```

Adaptive Response Surface Method (ARSM)

The adaptive response surface method (ARSM) works by internally building response surfaces and adaptively updating them as new evaluations become available.

The first response surface it builds is a linear regression polynomial, then it finds the optimum on this surface and validates it with the exact simulation. If the response values from the response surface and the exact simulation are not close; ARSM updates the surface with the new evaluation and finds the optimum in this updated surface. ARSM repeats this loop until it meets one of the convergence criteria.

Error treatment and termination

The ARSM terminates when one of the following conditions is met:

- One of the convergence criteria is satisfied.
- The maximum number of allowable analyses is reached.

The text log of the ARSM method

Section 1: General information regarding the optimisation setup.

```
================  L O G - FILE - OPTFEKO  ================

Version: 14.0.430-24 of 2016-08-22
Date: 2016-08-30 11:34:29
File: dipole_arsm

OPTIMISATION WITH Feko

================= Optimisation variables =================

No.  Name                                  Beg.value          Minimum
 Maximum
  1 h                                2.000000000e+00  1.600000000e+00  2.400000000e
+00
  2 radius                           2.000000000e-03  5.000000000e-04
 5.000000000e-03



=================== Optimisation goals ===================

No.  Name                             Expression
  1 arsm.goals.impedance_mag73ohm   mag(inputimp(impedance(source)))
```

Section 2: Information regarding the ARSM parameters.

```
=== Optimisation method: ADAPTIVE RESPONSE SURFACE METHOD (HyperOpt) ===

On failed analysis:              Ignore failed analysis (=1)
Initial linear move:             By DV initial (=1)
Maximum iterations:              22
Response surface:                SORS (=0)
Number of sample points:         3
ARSM solver:                     SQP (=1)
Use SVD:                         No; terminate at soft convergence (=0)
ARSM algorithm version:          A; normal (=0)
Absolute convergence:             1.0000000000e-04
Constraint screening (%):         5.0000000000e+01
Constraint violation tol. (%):    2.5000000000e-01
Design variable convergence:      1.0000000000e-03
Initial DV perturbation:          1.1000000000e+00
Move limit fraction:              1.5000000000e-01
Relative convergence (%):         5.0000000000e-01
Minimal move factor:              1.0000000000e-01
Constraint threshold              1.0000000000e-04
```

Section 3: Information regarding the parameter values and goal values at each step.

```
=========== ADAPTIVE RESPONSE SURFACE METHOD (HyperOpt): Intermediate results
 ===========

 No.    h                radius           arsm.goals.impe  Global goal      Global
 best aim
     1  2.000000000e+00  2.000000000e-03  8.223708649e+01  9.237086490e+00
 9.237086490e+00
     2  2.330000000e+00  2.000000000e-03  2.584129574e+02  1.854129574e+02
```

```
     3  2.000000000e+00  2.495000000e-03  8.276434015e+01  9.764340145e+00

     4  1.700000000e+00  1.550000000e-03  1.362858864e+02  6.328588642e+01

     5  1.930006139e+00  1.550000000e-03  6.806363939e+01  4.936360614e+00
 4.936360614e+00
     6  1.927744177e+00  2.000000000e-03  6.805981842e+01  4.940181579e+00

     7  1.931082122e+00  1.788235587e-03  6.821262088e+01  4.787379122e+00
 4.787379122e+00
     8  1.931204660e+00  1.791996361e-03  6.822294926e+01  4.777050742e+00
 4.777050742e+00
     9  1.931283820e+00  1.793912982e-03  6.822946010e+01  4.770539896e+00
 4.770539896e+00
```

Section 4: Information regarding the termination reason and best results found on the search grid.

```
================= ADAPTIVE RESPONSE SURFACE METHOD (HyperOpt): Finished
 =================

Optimisation finished (ARSM optimizer achieved convergence. Relative change in
 objective function over
two last iterations is smaller than 0.500E-02 and max. constraint violation is below
 permitted level.)

Optimum found for these parameters:
  h                             =   1.931283820e+00
  radius                        =   1.793912982e-03

Optimum aim function value (at no. 9):   4.770539896e+00
No. of the last analysis: 9
```

Global Response Surface Method (GRSM)

The global response surface method (GRSM) is a response surface based approach. During each iteration, the response surface based optimisation generates a few designs. Additional designs are generated globally to ensure a good balance on local search capability and global search capability.

Error treatment and termination

The response surface is adaptively updated with the newly generated designs to have a better fit of the model.

The text log of the GRSM method

Section 1: General information regarding the optimisation setup.

```
================  L O G - FILE - OPTFEKO  ================

Version: 14.0.430-24 of 2016-08-22
Date: 2016-08-30 11:36:21
File: dipole_grsm

OPTIMISATION WITH Feko

================= Optimisation variables =================
```

```
No.  Name                                    Beg.value          Minimum
 Maximum
  1 h                                  2.000000000e+00  1.600000000e+00  2.400000000e
+00
  2 radius                             2.000000000e-03  5.000000000e-04
 5.000000000e-03



=================== Optimisation goals ===================

No.  Name                               Expression
  1 grsm.goals.impedance_mag73ohm   mag(inputimp(impedance(source)))
```

Section 2: Information regarding the GRSM parameters.

```
=== Optimisation method: GLOBAL RESPONSE SURFACE METHOD (HyperOpt) ===

On failed analysis:             Ignore failed analysis (=1)
Maximum iterations:             50
Random seed:                    1
Initial sample points:          4
Points per iteration:           2
Constraint violation tol. (%):   5.0000000000e-01
Constraint threshold             1.0000000000e-04
```

Section 3: Information regarding the parameter values and goal values at each step.

```
=========== GLOBAL RESPONSE SURFACE METHOD (HyperOpt): Intermediate results
 ===========

 No.    h                radius           grsm.goals.impe  Global goal      Global
 best aim
     1  2.000000000e+00  2.000000000e-03  8.223708649e+01  9.237086490e+00
 9.237086490e+00
     2  2.348800000e+00  4.964000000e-03  2.488655725e+02  1.758655725e+02

     3  1.678080000e+00  4.942400000e-03  1.120701608e+02  3.907016075e+01

     4  2.394880000e+00  9.464000000e-04  3.227940059e+02  2.497940059e+02

     5  1.928320000e+00  2.401760000e-03  6.825584105e+01  4.744158945e+00
 4.744158945e+00
  ...
```

Section 4: Information regarding the termination reason and best results found on the search grid.

```
================ GLOBAL RESPONSE SURFACE METHOD (HyperOpt): Finished
 ================

Optimisation finished (GRSM optimiser stopped. Number of analysis reached the maximum
 allowed number of
50)

Optimum found for these parameters:
  h                             =   1.957379988e+00
  radius                        =   3.243705368e-03

Optimum aim function value (at no. 47):   4.087626847e-02
No. of the last analysis: 50
```

7.4.2 Sensitivity Analysis

OPTFEKO calculates upon termination of an optimisation, a sensitivity analysis of the goal function with relation to each parameter.

The sensitivity analysis is calculated using the particle swarm optimisation (PSO), generic algorithm (GA) or Simplex method, if sufficient information is available. The calculated sensitivity values are indicated on the screen output, and stored in the text `.log` file. If no sensitivity analysis is performed, the reason is indicated on the screen output, but no indication is written to the text `.log` file.

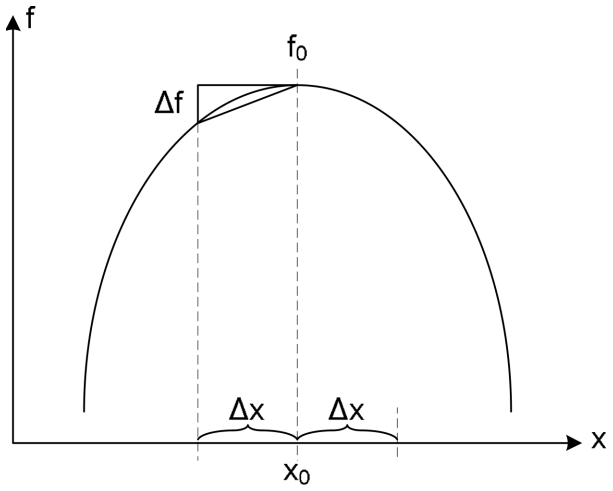


Figure 569: Sensitivity analysis of the goal function f with relation to the parameter x.

Figure 569 shows an example goal function f that varies as a function of the parameter *x*. The sensitivity with relation to the parameter *x* can be described by the following equation:

$$S(x) = \left.\frac{\partial f}{\partial x}\right|_{x_0 \pm \Delta x} \tag{104}$$

with Δx equal to 1

$$\Delta x = 0.01(x_{max} - x_{min}) \tag{105}$$

Solving the equation, however, gives a near zero value when the solution space is well converged. We therefore rather compute the second derivative from which the sensitivity parameter can be computed through integration

$$S(x) \approx \left.\frac{\partial f}{\partial x}\right|_{x_0 \pm \Delta x} \approx \left.\frac{\partial^2 f}{\partial x^2}\right|_{x_0} \cdot \Delta x \tag{106}$$

to finally give the sensitivity with relation to *x* as

$$S(x) \approx f''(x_0) \cdot 0.001(x_{max} - x_{min}) \tag{107}$$

A sensitivity analysis will only be performed if at least $2N+1$ samples are available for a problem with N parameters and these samples should all be within a 5% radius of the optimum. If the samples under consideration are scattered outside of a 5% radius of the optimum, the stored data is considered insufficient for proper sensitivity analysis. It should also be realised that as this computation makes use of already computed samples only, the accuracy of the reported sensitivity number depends on how well the algorithm has converged.

7.4.3 Farming

Farming out of the steps of an optimisation involves the concurrent solution of various optimisation steps on a number of available processors or hosts.

> **Note:** Farming not supported with the *Simplex* method.

When farming out the individual optimisation steps, the number of processes to start on each available host is specified in the *machines file*. This machines file has a syntax identical to that used for parallel runs. The basic syntax is:

```
Hostname:Number of processes
```

using a new line for each host.

During optimisation, new model files are continuously created by adding the string `_opt_` and a sequentially incremented number to the file name of each relevant component file of the parametric model.

Example

Two hosts are available with names *host1* and *host2*[81], with 4 and 8 processors respectively.

The machines file:

```
host1:4
host2:8
```

Launch an optimisation run with 12 processors for farming, results in the first 4 optimisation steps to be launched on *host1* and the next 8 steps to be launched on *host2*.

```
optfeko <x> -np <n_farming> --machines-file <m>
```

where:

<x>
: File name of the model

<n_farming>
: Number of farming processes

<m>
: File name of the *machines* file

81. this is the output of the UNIX command `hostname`)

Farming in Conjunction with Parallel Computing

Large problems may require that Feko be run in parallel (simulation spread over more than one host or processor - not the same as farming.)

Example

Run the Solver in parallel, while farming out the optimisation steps using a specified number of processes. For example:

```
optfeko <x> -np <n_farming> --machines-file <m> --runfeko-options -np <n_parallel>
```

where:

<x>
: File name of the model

<n_farming>
: Number of farming processes

<n_parallel>
: Number of parallel processes

<m>
: File name of the *machines* file

Note: For optimisation using the Solver in parallel, but not farming:

```
optfeko <x> --runfeko-options -np <n_parallel> --machines-file <m>
```

7.4.4 Optimisation Output

Information on the optimisation process is stored in a `.log` file.

Note: Using the remote launching facility or farming out of optimisation steps, the actual optimisation is done on the local machine, only the Solver runs (which are the time and memory consuming part) are done on the remote machine(s).

The optimisation process may be interrupted at any time by clicking the **Stop** in the GUI process information window, or Ctrl+C in a command line. If an optimisation is interrupted, all of the interim files created during the optimisation will be kept, except if the **Delete all files** option was selected when running from the GUI, or if the `-r` option was added when running from a command line. If the files were kept, the optimisation can be restarted at a later stage using the `-restart x` option.

7.5 ADAPTFEKO

ADAPTFEKO is the adaptive frequency utility used to automatically select smaller frequency steps near narrow resonances and larger steps where the results are relatively smooth.

For each frequency, ADAPTFEKO creates a `.pre` file and calls PREFEKO and the Solver. The file names are derived from the original name plus `_fr_n_ada_m` where `n` and `m` are incremental numerical values (for example, the new files associated with `test.pre` are `test_fr1_ada_1.pre`, `test_fr1_ada_2.pre`, ...).

7.5.1 Command Line Arguments for Launching ADAPTFEKO

ADAPTFEKO is started automatically by RUNFEKO if a continuous (interpolated) frequency range is specified.

The syntax is:

```
runfeko filename
```

or

```
runfeko filename --adaptfeko-options [options]
```

where the optional argument options in the second line may be:

-h,--help
: Displays the help message.

--version
: Output only the version information to the command line and terminate.

--keep-files
: All solution files (for example, `.pre`, `.fek`, `.out`) are preserved.

--restart x
: Restart an adaptive frequency analysis using results for the frequency points 1...(x-1) obtained in a previous run. (Then the previous run must have used `-keepfiles`.)

Related concepts
Setting a Continuous (Interpolated) Frequency Range
Related reference
FR Card

7.5.2 The PRE File for Adaptive Sampling

During solution, the variable `#adaptfreq` is defined automatically at the start of the single frequency input files generated by the ADAPTFEKO utility. This variable may be used to allow for frequency-based variation (for example, adaptive meshing).

You should not directly assign this name to a variable inside the `.pre` file or in CADFEKO as this will overwrite the value specified by ADAPTFEKO. If this variable is needed (for example to run PREFEKO during model setup when using adaptive meshing), the `DEFINED` function should be used in the .pre file, for example:

```
** define a frequency variable if it is not already defined by an ADAPTFEKO run
!!if (not(defined(#adaptfreq))) then
#adaptfreq = 250.0E6
!!endif
```

Note: Care must be taken when using adaptive meshing with ADAPTFEKO. Small discontinuities may result from changes in the mesh can have a dramatic effect on the convergence and accuracy of the adaptive sampled results.

7.6 AMRFEKO

AMRFEKO is the adaptive mesh refinement utility used to automatically refine the mesh.

The utility calculates the error estimates for the mesh, and based on these estimates, applies mesh refinements. This process is repeated until the error estimates are sufficiently reduced. The last refined model is saved with a new file name. Lower error estimates yield more accurate results. The current implementation is based on error estimates for the FEM.

> **Tip:** Use the FEM solution method when solving a model.

7.6.1 Command Line Arguments for Launching AMRFEKO

Run AMRFEKO to perform automatic mesh refinements.

The syntax is:

```
amrfeko filename
```

or

```
amrfeko filename [options]
```

where the optional argument options in the second line may be:

-h,--help
: Displays the help message.

--version
: Output only the version information to the command line and terminate.

--keep-files
: Files pertaining to each iteration of the refinement process are retained.

--skip-final-solve
: The final refined model is saved only but not solved.

--frequency FREQUENCY
: Set the frequency to use for refinement (also sets the number of frequencies to 1, ignoring any supplied setting for the number of frequencies).

--error-threshold HIGH | NORMAL | LOW
: Set error threshold (high / normal / low). A low threshold requires smaller errors before it terminates the mesh refinement. It could be considered an aggressive mesh refinement compared to the other options.

--runfeko-options OPTION_1 [OPTION_2 ...]
: All following options will be passed to RUNFEKO.

For each adaptive mesh refinement iteration, the Solver is run. After the final adaptive mesh refinement is reached, the refined model is saved. If a `.cfx` file was provided, the refined model is saved as `<model>_refined.cfx`.

AMRFEKO can also be run on an EDITFEKO model (`.pre` file) or a `.fek` file. When there is no `.cfx` file present, the refined model is saved as `<model>_fek_refined.fek`.

Example

The following examples shows how to call AMRFEKO:

```
amrfeko Example.cfx --error-threshold low
```

```
amrfeko Example.pre --error-threshold low --keep-files
```

```
amrfeko Example.fek --frequency 1e9
```

Related tasks

Refining the Mesh Adaptively Using Error Estimates

7.7 Environment Initialisation Script - initfeko

The `initfeko.bat` (batch file on Windows) and `initfeko` (bash shell script on Unix / Linux) scripts are executed from a terminal to configure the Feko environment. From this environment, the Feko applications can be launched without using their full path.

The scripts are provided for convenience and it is not required to use them. All Feko applications can be launched directly by using the full path to the application and the environment will be configured correctly.

The command line parameters for `initfeko` are as follows:

-v	Verbose mode (prints some information output).
-d	Shows additional debug output while setting the environment.
-t	Timeout on error (default is to wait for user).
-terminal	Mode to setup a complete standalone Feko terminal.[82]

Note: View the Environment Settings Overview for more information on how to set up the Feko environment using Lua commands and internal functions.

82. Note that -console is also supported

7.8 Launcher Utility

The Launcher utility is a single application that allows you quick access to the shortcuts for the Feko components, WinProp components (Microsoft Windows only), newFASANT, WRAP components (Microsoft Windows only), documentation, Altair license utility and updating parallel credentials. Pin the application to the taskbar for quick launching.

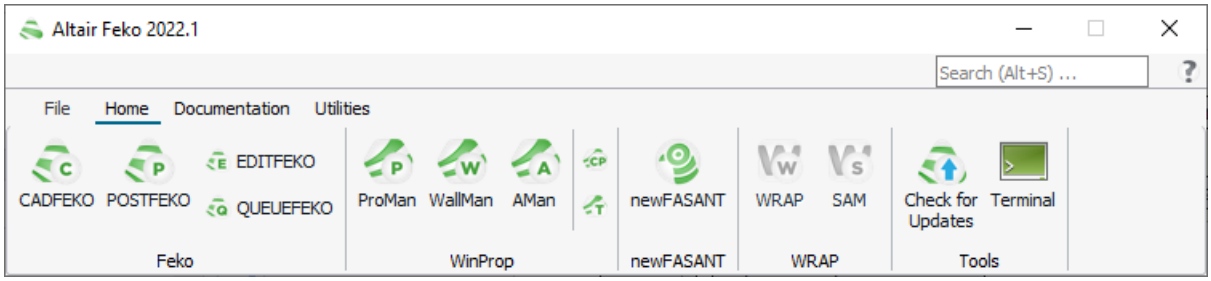


Figure 570: The Launcher utility allows quick access to Feko, WinProp (Microsoft Windows only), newFASANT, WRAP components (Microsoft Windows only), documentation and utilities.

Note: When WRAP is installed in an existing Altair Feko installation, the WRAP components are enabled on the Launcher utility.

7.8.1 Opening the Launcher Utility (Windows)

There are several options available to open the Launcher utility in Windows.

Open the Launcher utility using one of the following workflows:

- Open the Launcher utility from the Windows start menu:
  1. On the desktop, click the Windows **Start** button.
  2. Type `Feko` or `WinProp`.
  3. Select **Feko 2024.1** from the list of filtered options.
- Open the Launcher utility using the desktop shortcut (if you selected the option to install shortcuts during installation).

7.8.2 Opening the Launcher Utility (Linux)

There are several options available to open the Launcher utility in Linux.

Open the Launcher utility using one of the following workflows:

- Open a command terminal. Use the absolute path to the location where the Launcher utility executable resides (for example, `/home/user/2024.1/altair/feko/bin/feko_launcher`).
- Open a command terminal. Source the script "initfeko" using the absolute path to . `/home/user/2024.1/altair/feko/bin/feko_launcher`. Type `feko_launcher` and press Enter.

Note: Take note that sourcing a script requires a dot (".") followed by a space (" ") and then the path to `initfeko` in order for the changes to be applied to the current shell and not a sub-shell.

7.9 Updater

The `feko_update_gui` utility and the `feko_update` utility allows you the flexibility to install an update containing features, minor software enhancements and bug fixes on top of an existing base installation for Altair Feko (which includes Feko, newFASANT and WinProp).

7.9.1 Version Numbers

Each major release, upgrade or update is assigned a version number. A version number contains a unique set of numbers assigned to a specific software release for identification purposes. You can determine from the version number if its an initial release, update or upgrade.

The following terminology is used to define a version number:

```
Feko <Major>.<Minor>.<Patch>
```

for example:

```
Feko 2019.1.2
```

2019

Indicates the major release version. A major release is made available roughly once a year and has a minor and patch version of "0".

> **Note:**
> - The update utility does not support upgrades between major versions.
> - A major release requires a new installer.

1

Indicates the minor release version and is referred to as an upgrade. Large feature enhancements and bug fixes are included in the upgrade. Minor upgrades are released quarterly, for example "1" indicates the first minor upgrade after the initial release. Use the update utility to upgrade to a newer minor version (when available).

2

Indicates the patch version and is referred to as an update or "hot fix". Minor feature enhancements and bug fixes are included in the update. Patch updates are released between minor upgrades, for example "2" indicates the second patch update after an upgrade.

7.9.2 GUI Update Utility

Use the `feko_update_gui` to check for new versions of the software and install an update using a graphical user interface (GUI).

Click on **Application menu** > **Check for updates** to do a forced check for updates[83].

When either CADFEKO, EDITFEKO or POSTFEKO is launched and the scheduled interval time has elapsed, the update utility (GUI mode) automatically checks for updates. By default the schedule is set to check for updates once a week. If updates are available, the update utility displays a notification alert as well as giving you the option to select and install updates.

The GUI update utility can be started from the command line using:

```
feko_update_gui
```

Updates can be installed from a web repository[84] or a local repository. During an update a list containing the latest software is retrieved and compared to installed components.

> **Note:** No information is collected during an update.

Viewing the Installed Component Versions

View the version numbers of the installed Feko components.

1. Open the Updater using the Launcher utility.
2. On the **Altair Feko update** dialog, click the **Installed versions** tab.
3. View the **Component**, **Version** and **Date** information for the current installation.

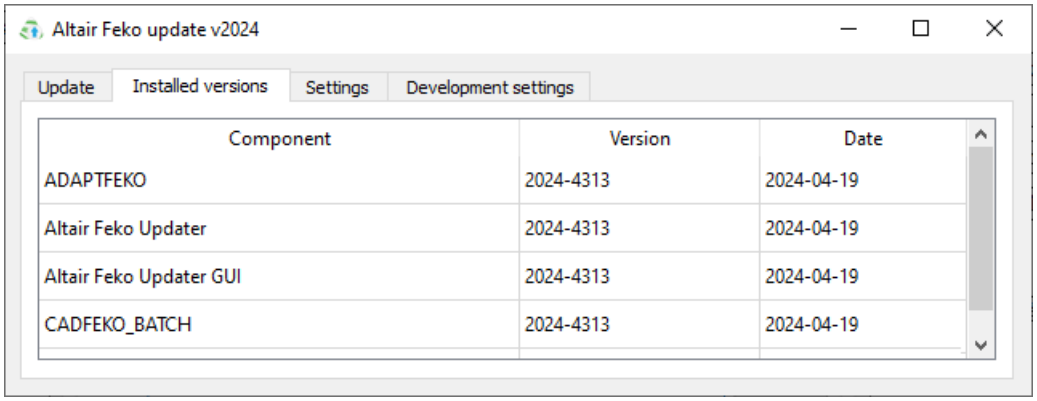


*Figure 571: The **Altair Feko update** dialog - **Installed versions** tab.*

4. Click the **Update** tab and click **Close** to exit the **Altair Feko update** dialog.

83. A forced update can also be done from the application menu in CADFEKO, POSTFEKO and EDITFEKO.
84. Requires internet access.

Updating or Upgrading to a New Version

Updating and upgrading refers to the process of installing a new version containing features, minor software enhancements and bug fixes on top of an existing base installation.

1. Open the Updater using the Launcher utility.
2. On the **Altair Feko update** dialog, click the **Update** tab.
3. Click the **Refresh** button to view the available Feko versions for download.
4. Select a version to view the available components and their individual file size in the table.

> **Tip:** Click **Details** to view the release notes in the message window.

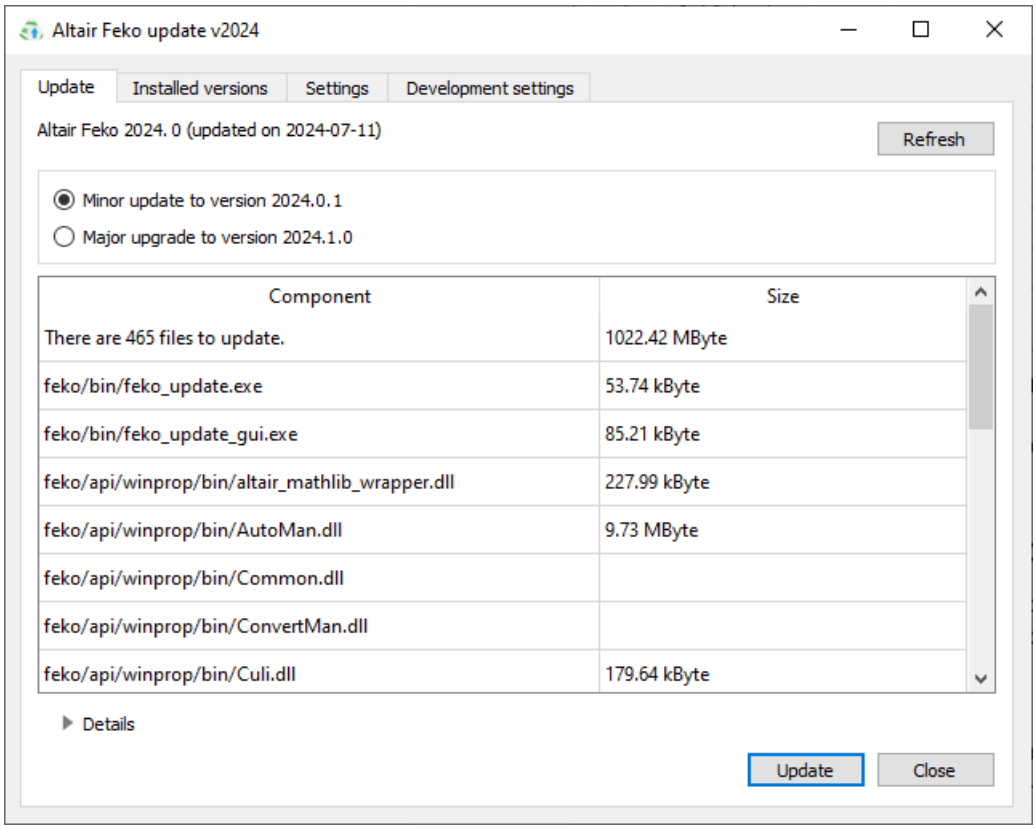


*Figure 572: The **Altair Feko update** dialog - **Update** tab.*

5. Click **Update** to update or upgrade to the selected version.
   a) Before an upgrade is started, you will be asked to confirm the upgrade from the current version to the selected version. Click **Continue with upgrade** to allow the update/upgrade process to proceed.
   b) During the update process, click **Details** to expand the message window and view detailed information regarding the update process.
6. When the update or upgrade is complete, click **Close**.

Updating From a Local Repository (GUI)

Update (or upgrade) from a local repository using the graphical user interface.

1. Open the Updater using the Launcher utility.
2. On the **Altair Feko update** dialog, click the **Settings** tab.
3. Under **Update from**, click **Local repository** to update from a local repository.

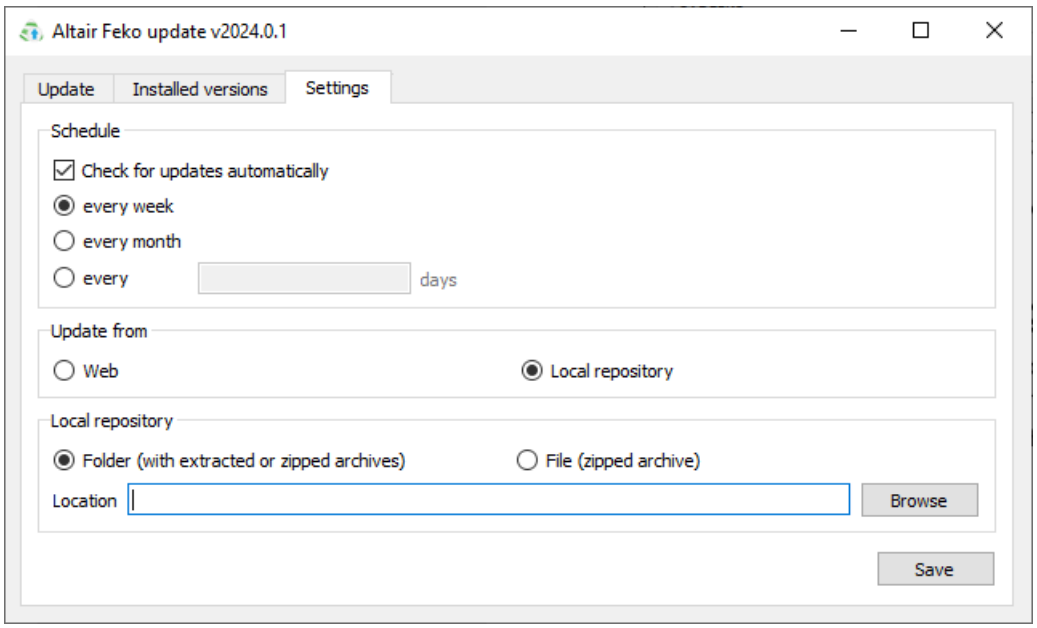


*Figure 573: The **Altair Feko update** dialog - **Settings** tab.*

4. Under **Local repository**, select one of the following:
   - If the local repository contains extracted archives or multiple zipped archives, select **Folder (with extracted or zipped archives)** and specify the folder.

     The path for the local Feko update repository must be an absolute file path which can point to an unmapped network share (Windows), mapped (mounted) network share or a directory on a local drive.

     Warning: Point the local repository path to the root folder of the updates.

     Example: The Feko updates for the Windows and Linux platforms were extracted and merged to `C:\Updates`. The path to the local repository points to `C:\Updates`.

     ```
     C:\Updates
         └─FEKO_2024.1.x
                     └─WIN64_X86_64
                     └─LINUX_X86_64
     ```

   - If the local repository contains a single zipped archive, select **File (zipped archive)** and specify the zip file.
5. Click **Save** to save the local repository settings.
6. Update or upgrade to a new version.

Troubleshooting: `Error 16700: Unable to find the file 'XX/YY/manifest.xml.gz' in the local repository.`

Error 16700 indicates that the path to the local repository is incorrect. The path must point to the root folder of the local update repository and the folders should not be modified.

Scheduling Automatic Updates

Schedule and configure an automatic Feko update.

1. Open the Updater using the Launcher utility.
2. On the **Altair Feko update** dialog, click the **Settings** tab.

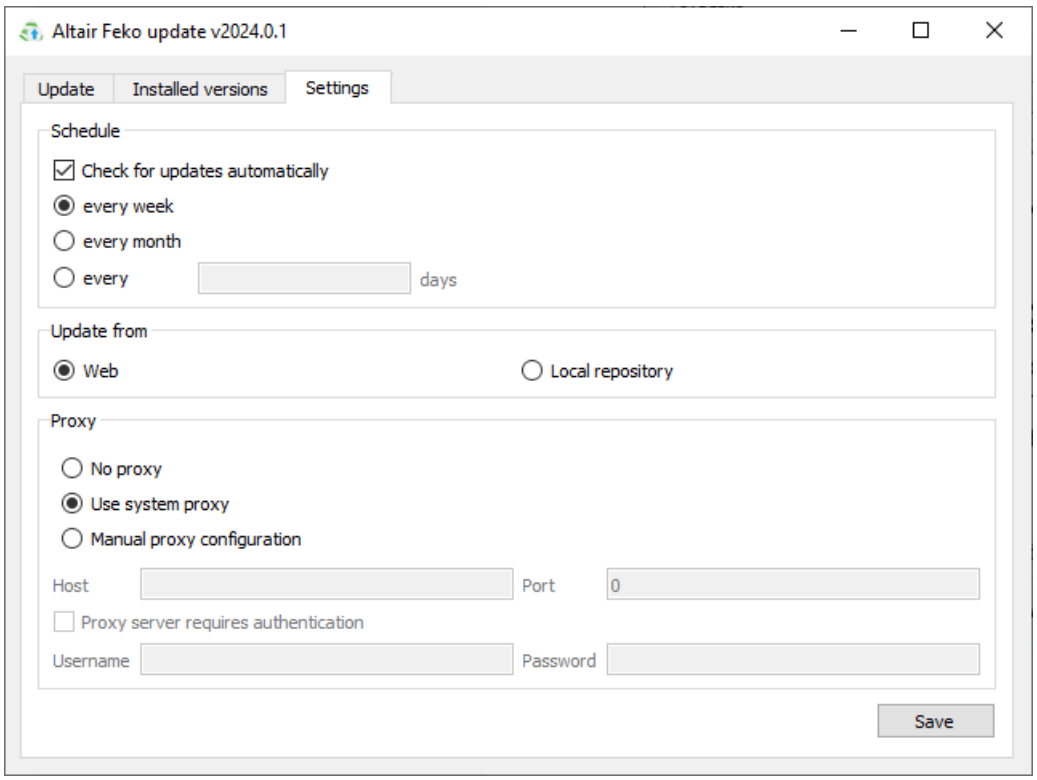


*Figure 574: The **Altair Feko update** dialog - **Settings** tab.*

3. Select the **Check for updates automatically** check box to automatically check for updates. Select one of the following options:
   - every week
   - every month
   - every N days
4. Select the download location under **Update from** group box.

   Web

   The updates are downloaded from the web repository.

Local repository
This option is recommended when the computer network or cluster has no internet access due to security reasons or only limited available bandwidth. The updates may be downloaded from the Connect website by the system administrator and placed at a location accessible for the computer network or cluster.

5. Optional: Specify the proxy server and authentication when the web is specified as the repository under **Proxy** group box.
6. Click **Save** to save the new settings.

7.9.3 Command Line Update Utility

Use the `feko_update` utility for scripted updates or updates from a Feko terminal.

The command line update utility is called from the command line using:

```
feko_update
```

-h,--help
Displays the help message.

--version
Output only the version information to the command line and terminate.

UPGRADE_OPTION
Argument that allows a specific major patch version to be specified. This option is used to view the Feko component changes for a specific major patch version, their respective download size and the release notes. *UPGRADE_OPTION* can be any of the following:

1-9
Indicates the major patch version.

latest
This option selects the largest valid major patch version that has a repository.

--check [UPGRADE_OPTION] [[USER:PASSWORD@]PROXY[:PORT]]
The update utility checks if new versions are available. If *UPGRADE_OPTION* was not specified and new versions are available, it will list the version and its associated *UPGRADE_OPTION* value. For example:

```
Update/upgrade options are available (UPGRADE_OPTION):
0: Minor update to version 2024.1.0.1
```

If the computer is behind a proxy server, the proxy server address and the login details can be supplied as required.

--check-from LOCATION [UPGRADE_OPTION]
The update utility checks if new versions are available. Here the update source is the local repository specified by *LOCATION*. If *UPGRADE_OPTION* was not specified and new versions are available, it will list the version and its associated *UPGRADE_OPTION* value.

--update [USER:PASSWORD@]PROXY[:PORT]]
: The update utility checks if new versions are available within the current patch major version from the web repository. If an update is available, download and install the new version. If the computer is behind a proxy server, the proxy server address and the login details can be supplied as required. If updates are available, the following information is printed to the screen:
  - Print each file which is being downloaded (only available when the update does not contain many files).
  - Print each file which is being updated (only available when the update does not contain many files).
  - Print a message stating that the update was successful and exit.

--update-from LOCATION
: The update utility checks if new versions are available within the current patch major version and installs the new version. Here the update source is the local repository specified by *LOCATION*. The path must be an absolute file path which can point to an unmapped network share (Windows), mapped (mounted) network share or a directory on a local drive that can contain either extracted archives, multiple zipped archives or a single zipped archive.

--upgrade UPGRADE_OPTION [[USER:PASSWORD@]PROXY[:PORT]]
: The update utility checks if new patch major versions are available from the web repository. If an upgrade is available, download and install the new version.

--upgrade-from LOCATION UPGRADE_OPTION
: The update utility checks if new patch major versions are available from the web repository. If an upgrade is available, it will download and install the new version. Here the update source is the local repository specified by *LOCATION*. The path must be an absolute file path which can point to an unmapped network share (Windows), mapped (mounted) network share or a directory on a local drive that can contain either extracted archives, multiple zipped archives or a single zipped archive.

--no-progress
: Suppress the download progress when updating from a web repository.

--no-proxy
: Suppress the use of a proxy (including the system proxy).

Updating From a Local Repository (Command Line)

Download a new software update (or upgrade) from a local repository using the command line utility.

1. Open a Feko terminal using the Launcher utility.

   Note: If a script is used to call the Feko updater, do one of the following:
   - Run the script from a Feko terminal.
   - Include `%FEKO_HOME%\bin` in the `PATH` environment variable.
   - Call the Feko updater using the full path, for example: `C:\Program Files\Altair\2024.1\feko\bin\feko_update.exe`.

2. Download the latest version using one of the following workflows:

- To update (if an update is available) within the current minor version, type:

```
feko_update --update-from LOCATION
```

- To upgrade to a new minor version, type:

```
feko_update --upgrade-from LOCATION VERSION
```

where `LOCATION` is either an absolute file path which can point to an unmapped network share (Windows), mapped (mounted) network share or a directory on a local drive that can contain either extracted archives, multiple zipped archives or a single zipped archive.

The `version` is the minor version that you would like to upgrade to and would usually be `1`, `2` or `3`, but it is possible to use `latest` to upgrade to the latest version.

The command line updater has many options to check for updates without updating or update to the latest version. Use the following command to see a list of options:

```
feko_update --help
```

7.9.4 Proxy Settings Overview

The `feko_update_gui` utility and `feko_update` utility (GUI and command line) use the system proxy by default, although it may be changed or the use of a proxy suppressed.

Windows

The proxy used is the same as is used by Internet Explorer. The proxy can be specified or by using a proxy auto-config (PAC) file.

Linux

The system proxy is defined by the environment variable *http_proxy*. If the environment variable *http_proxy* is not defined, then no proxy will be used.

Suppressing the Use of a Proxy

The parameter *--no-proxy* bypasses the system settings and use a direct connection.

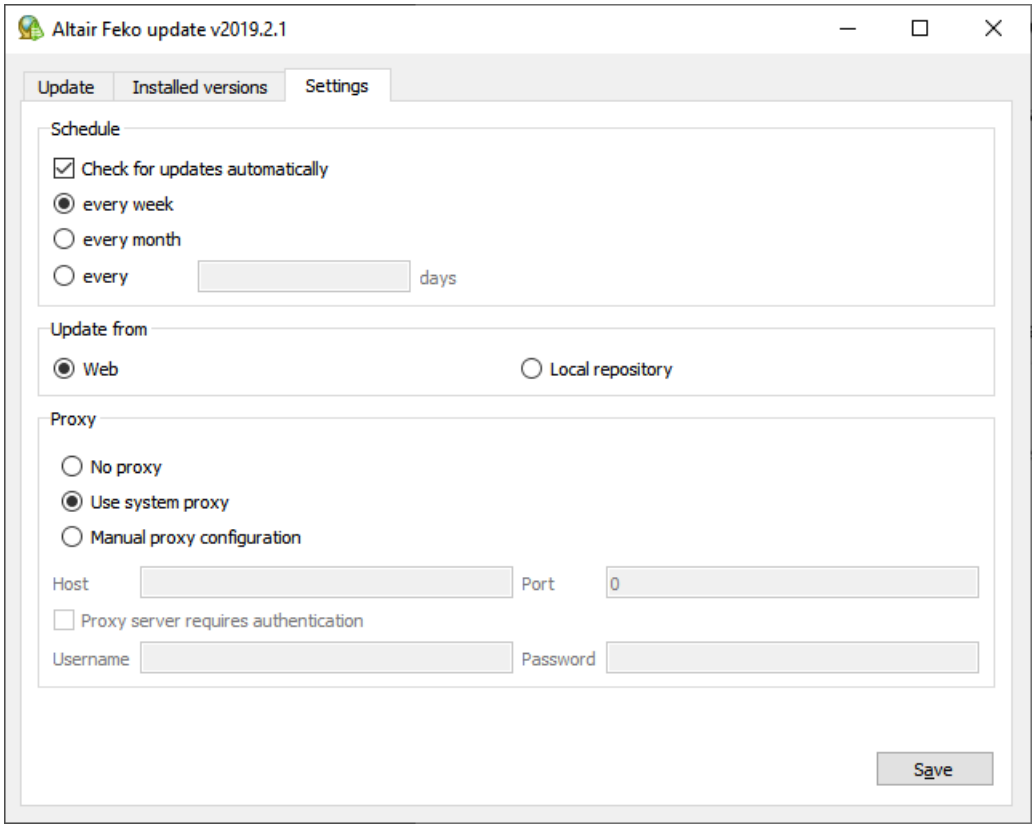


*Figure 575: The **Altair Feko update** dialog - **Settings** tab.*

7.9.5 Creating a Local Update Repository

Create a local Feko update repository to allow users to update without internet access or to limit the list of update versions that users can use. Local update repositories can also be used to reduce the amount of data being downloaded by downloading a repository once and making it available to many local machines or compute clusters.

A local repository folder can be set up using:

1. downloaded and extracted archives
2. downloaded, zipped archives

Using Extracted or Zipped Archives for Repo

Create a local Feko update repository using extracted or multiple archives.

1. Create the local repository folder, for example, `C:\Updates`.

   Note: If you already have an update repository for the same version, delete previous updates located in this folder.

2. Download the updates for the required platforms from Altair Connect.

   For example, if both the Windows and Linux platforms are required, download the following:

   - `FEKO_2024.1_WIN64_X86_64.zip`
   - `FEKO_2024.1_LINUX_X86_64.zip`

3. Unzip the downloaded archive(s) to the local repository folder.

   Important: Keep the structure of the `.zip` file intact.

   Important: If multiple platforms are downloaded, the platform updates must be located at the same folder (grouped by version) and "merged" (see example below).

Example: Extracting platform zip files with structure:

```
FEKO_2024.1_WIN64_X86_64.zip
    └FEKO_2024.1.x
        └WIN64_X86_64
            └...
```

to `C:\Updates`:

```
C:\Updates
    └FEKO_2024.1.x
        └WIN64_X86_64
            └...
        └LINUX_X86_64
            └...
```

Using a Single Zip Archive for Repo

Create a local Feko update repository using a single zip archive.

1. Create the local repository folder, for example, `C:\Updates`.

   > **Note:** If you already have an update repository for the same version, delete previous updates located in this folder.

2. Download the updates for the required platforms from Altair Connect.

   For example, if both the Windows and Linux platforms are required, download the following:

   - `FEKO_2024.1_WIN64_X86_64.zip`
   - `FEKO_2024.1_LINUX_X86_64.zip`

3. Copy the zipped archives to the local repository without extracting the files.

7.10 The Multiport Processor

The multiport processor allows you to calculate results for changes in the port excitations and loading without rerunning the Solver.

Through the multiport combinations configuration (`.mcc`) file, define the new excitations and loading for each port in the multiport data package (`.mdp`). Results supported are the scaled far fields, scaled near fields and specific port parameters, for example, the scaling coefficients, the voltage, current and impedance of each port.

7.10.1 Multiport Processor Workflow

The basic workflow using the multiport processor is described.

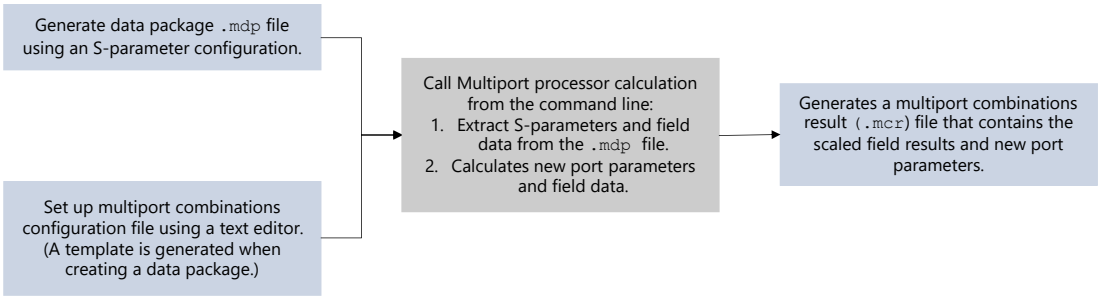


Figure 576: Basic workflow for the multiport processor calculator.

1. Generate a multiport data package using an S-parameter configuration. A template `.mcc` file is generated with creating a multiport data package `.mdp` file.
2. Set up the multiport combinations configuration file using a text editor[85].
3. Launch a calculation using the command line arguments[86].

   Note: The `.mcr` file is a HDF5 file.

4. Process the results using other Altair tools, for example, Altair Compose to read the results from the `.mcr` file.

85. See Multiport Combinations Configuration (.MCC)
86. See Using the Multiport Processor Calculator

7.10.2 Command Line Arguments for the Multiport Processor

The multiport processor is called via the command line to do a multiport calculation or to archive and extract data from a multiport data package `.mdp` file.

Using the Multiport Processor Calculator

Use the following command line parameters to launch the multiport processor as a calculator:

```
multiport_processor filename [OPTIONS]
```

FILENAME

The name of the multiport combinations configuration `.mcc` file.

OPTIONS

--version

Output only the version information to the command line and terminate.

--scale-factors-only

Only compute the scale-factor coefficients and terminate.

> **Note:** No scaling of quantities are performed, for example, far fields.

--combination name

Only process the results for a specified combination in the `.mcc` file.

--quantities [quantity_1{,quantity_i}]

Specify the quantities that should be processed by the computed scale-factor coefficients. *<quantity>* can be one of the following:

- FarFields

> **Note:** If no quantities are specified, then all quantities are processed.

--resultsfile fname

Specify the name of the multiport combinations result `.mcr` file.

Using the Multiport Processing Archiver

Use the following command line parameters to use the multiport processor as a archiver:

```
multiport_processor [OPTIONS]
```

OPTIONS

--create-package fname

Create a multiport data package `.mdp` file and terminate.

fname
The name of the multiport data manifest `.mdm` file which lists all the files to be added to the archive.

--expand-package fname [dirname]
Expand a multiport data package `.mdp` file and then terminate.

fname
Name of the multiport data package `.mdp` file.

dirname
The destination directory where to expand the files from the `.mdp` file.

7.11 S-Parameter Port De-embedding Calculator

The command line utility accepts an augmented S-parameter network definition, de-embeds the specified ports using 2-port calibration definitions and exports the result to an S-parameter Touchstone file.

The syntax is:

```
deembed filename [options]
```

where the optional argument options may be:

--version
: Output only the version information to the command line and terminate.

--pall fname
: De-embed all ports using the network data in `<fname>`.

--px fname
: De-embed port <x> using the network data in `<fname>`.

--export-format format
: Data format of the exported network parameters:

```
RI Real/Imaginary
MA Magnitude/Phase
```

--resultsfile fname
: Name of the results file.

Example

The following example shows how to de-embed a four way power divider:

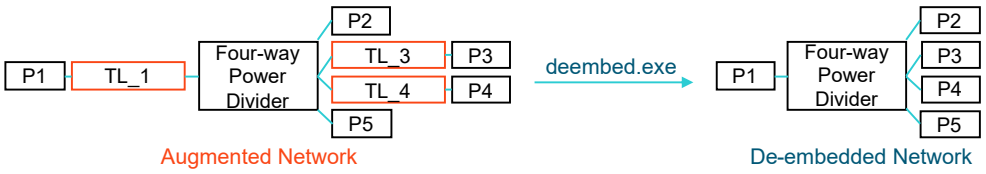


Figure 577: An example of how to use the utility.

```
deembed Four_Way_Power_Divider_SP.s5p --p1 TL_1_SP.s2p --p3 TL_3_SP.s2p --p4
 TL_4_SP.s2p
```

7.12 Crash Report Utility

In the event of a crash occurring in CADFEKO, POSTFEKO or EDITFEKO, the crash report utility generates a crash report.

The crash report gives you the option to provide the Altair Feko development team with details regarding the location where the crash occurred. This information is not always enough to identify and correct the problem. Providing a model and the steps that reproduce the crash is more useful to determine and correct the problem.

Important:

- To send a crash report to the Altair Feko development team is voluntary.
- No model files are attached without your consent.

Related tasks

Sending a Crash Report While Connected to the Internet

Exporting a Crash Report When Not Connected to the Internet

7.12.1 Sending a Crash Report While Connected to the Internet

You have the option to send the crash report to the Altair Feko development team with a description of the steps leading up to the crash. Crash reports can help the development team locate and correct problems faster.

In the event of a crash occurring in CADFEKO, POSTFEKO or EDITFEKO, the crash report utility generates a crash report.

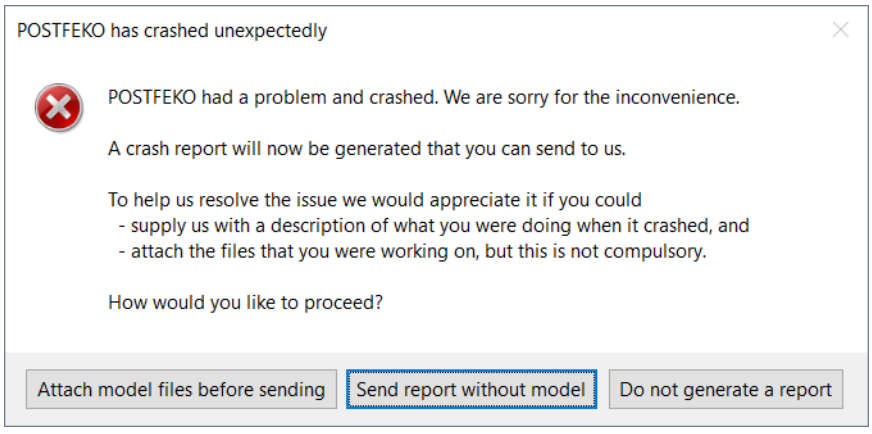


*Figure 578: The **POSTFEKO has crashed unexpectedly** dialog.*

1. Select one of the following options:
   - To attach the model files and any related files to the crash report, click **Attach model files before sending**.

- If you are working on a confidential model and do not want to send the model files, click **Send report without model**.
- If you do not want to generate a report nor attach model files, click **Do not generate report**, and the crash report utility will exit.

The **POSTFEKO has stopped working** dialog is displayed while the details regarding the location where the crash occurred are collected.

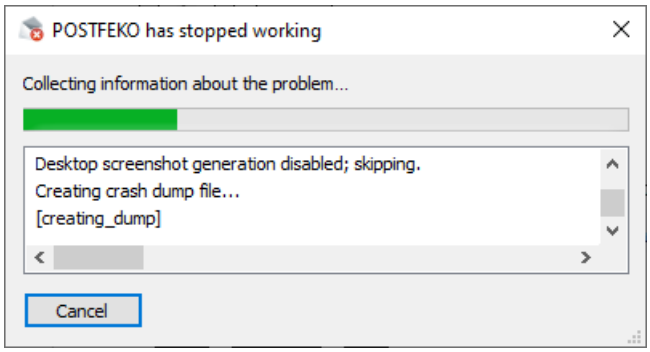


*Figure 579: The **POSTFEKO has stopped working** dialog.*

When the crash report is generated, the **Error Report** dialog is displayed.

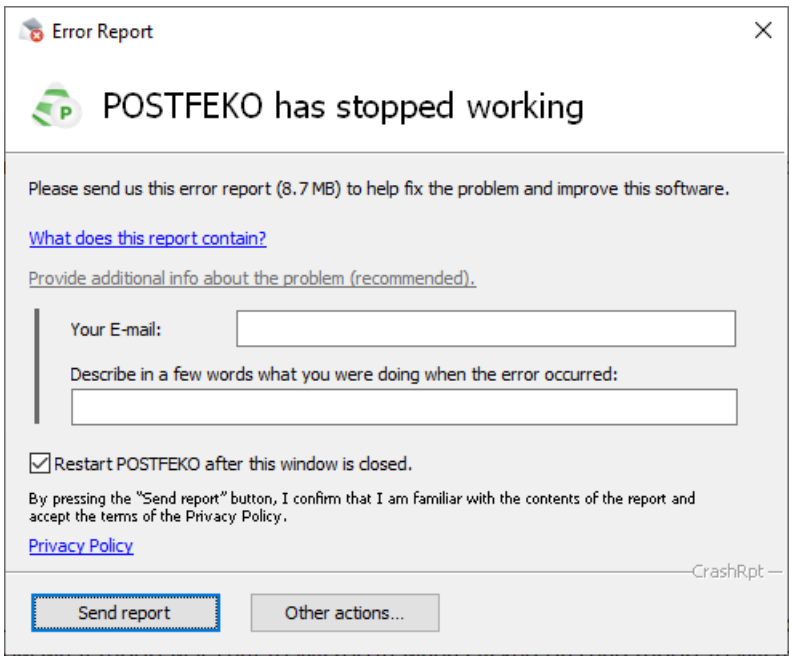


*Figure 580: The **Error Report** dialog.*

Note: The file size of the report is indicated at the top of the dialog.

2. [Optional] Click **What does this report contain?** to view the list of files in the crash report.
3. [Optional] In the **Your E-mail** field, enter your email if you would like feedback when the crash has been resolved.

4. [Optional] In the **Describe in a few words...** field, give a description of the steps that you followed at the time of the crash to help the Altair Feko development team resolve the issue.
5. [Optional] Select the **Restart after this window is closed** check box to restart the software after the window is closed.
6. Close the crash report utility by selecting one of the following workflows:
   - To send the report immediately and close the dialog, click **Send report**.
     - The report is sent by e-mail using a built-in SMTP client. If the network that the computer is on does not allow this, an attempt will be made to send the report using the default e-mail client installed on the computer.
   - To close the dialog and send the report later, click **Other actions** > **Close the program and send report later**.
   - To close the dialog and not send the report, click **Other actions** > **Close the program**.

     This option should be used when the computer is temporarily disconnected from the internet.

Note: Refer to the Privacy policy[87] for more information regarding how we use the information obtained from the crash report.

7.12.2 Exporting a Crash Report When Not Connected to the Internet

A crash report can be exported to a `.zip` file and emailed to Altair Technical Support. Use this workflow when the machine where the crash occurred is not connected to the internet.

In the event of a crash occurring in CADFEKO, POSTFEKO or EDITFEKO, the crash report utility generates a crash report.

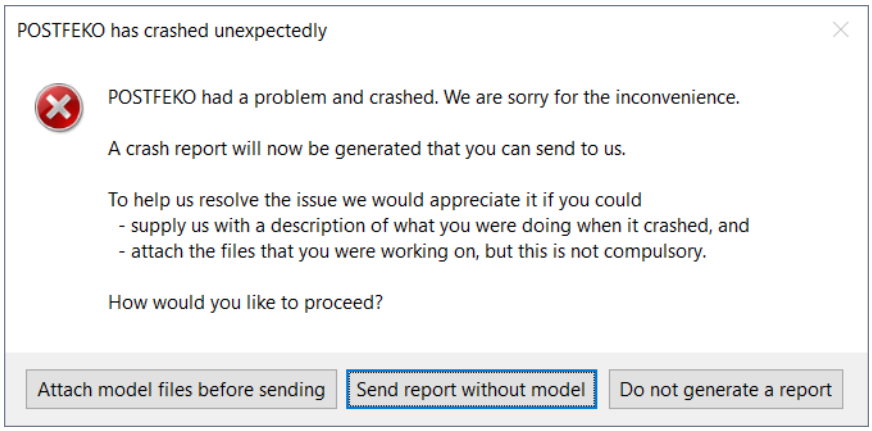


*Figure 581: The **POSTFEKO has crashed unexpectedly** dialog.*

1. Select one of the following options:

87. https://www.altair.com/privacy-shield/

- To attach the model files and any related files to the crash report, click **Attach model files before sending**.
- If you are working on a confidential model and do not want to send the model files, click **Send report without model**.

The **POSTFEKO has stopped working** dialog is displayed while the details regarding the location where the crash occurred are collected.

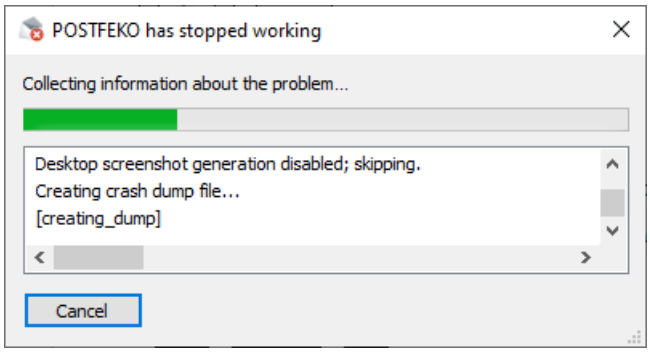


*Figure 582: The **POSTFEKO has stopped working** dialog.*

When the crash report is generated, the **Error Report** dialog is displayed.

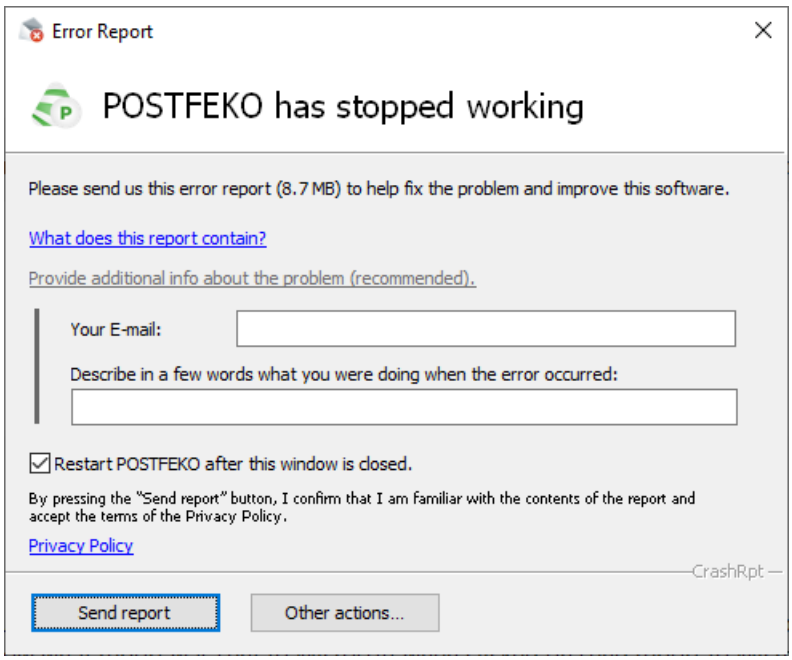


*Figure 583: The **Error Report** dialog.*

Note: The file size of the report is indicated at the top of the dialog.

2. On the **Error Report** dialog, click **What does this report** contain to view the files contained in the crash report.
   The **Error Report Details** dialog is displayed.

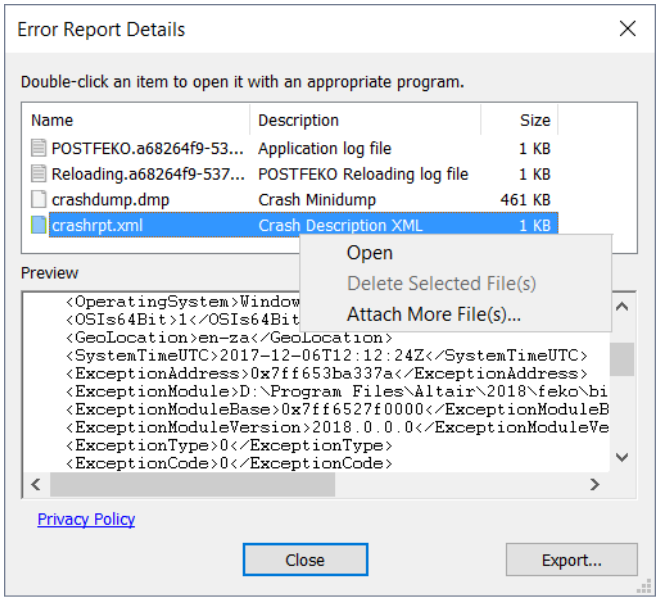


*Figure 584: The **Error report details** dialog.*

3. On the **Error report details** dialog, view the list of files and export to a `.zip` file.
   a) To view the data contained in a file, delete a file or add more files, click the file and from the right-click context menu, select the relevant option.
   b) Click **Export** to export the files listed on the **Error Report Details** dialog to a `.zip` file. Browse to the desired file location and specify a file name.
   c) Click **Close** to close the **Error Report Details** dialog.
4. On the **Error Report** dialog, click **Other actions** > **Close the program**.
5. Copy the exported `.zip` file to a machine that is connected to the internet and email the file to Altair Technical Support.

Note: Refer to the Privacy policy[88] for more information regarding how we use the information obtained from the crash report.

Related reference
Technical Support

88. https://www.altair.com/privacy-shield/

7.13 QUEUEFEKO

QUEUEFEKO is a graphical user interface (GUI) application that can create a package which you can transport to a remote queuing system. Created packages can be extracted once the simulation on the queuing system has been completed.

7.13.1 QUEUEFEKO Overview

The QUEUEFEKO graphical interface is used on the local machine to create the package that is then transferred to the compute cluster. On the compute cluster, use the `queuefeko` script to add a package to an execution queue.

These packages are placed by the queuefeko script (called `queuefeko`) in an execution queue (such as PBS) and executed when time and other resources become available. All information required to run Feko on the cluster is included in the package. The package is extracted on the remote machine and repackaged once the simulation is complete. Results can be viewed by copying the correct package back from the cluster and extracting the contents.

> **Note:** The `queuefeko` script is not a queuing system. The script handles the task of extracting, adding the job to the queue and packaging the results. A compatible queuing system must be set up separately.

QUEUEFEKO

QUEUEFEKO[89] is a graphical interface that allows users to create and extract packages. Use QUEUEFEKO on the local machine to configure the resource requirements and create a package for simulation and afterwards, to extract the package.

`queuefeko` script

The `queuefeko` script is a console application (editable script) that adds a package to an execution queue and takes care of all the management tasks required for the successful execution of the simulation. The `queuefeko` script runs on the remote cluster and oversees the simulation described in the package. Modifications to the script may be required to accommodate difference queueing systems.

89. The name of the binary is queuefeko_gui.

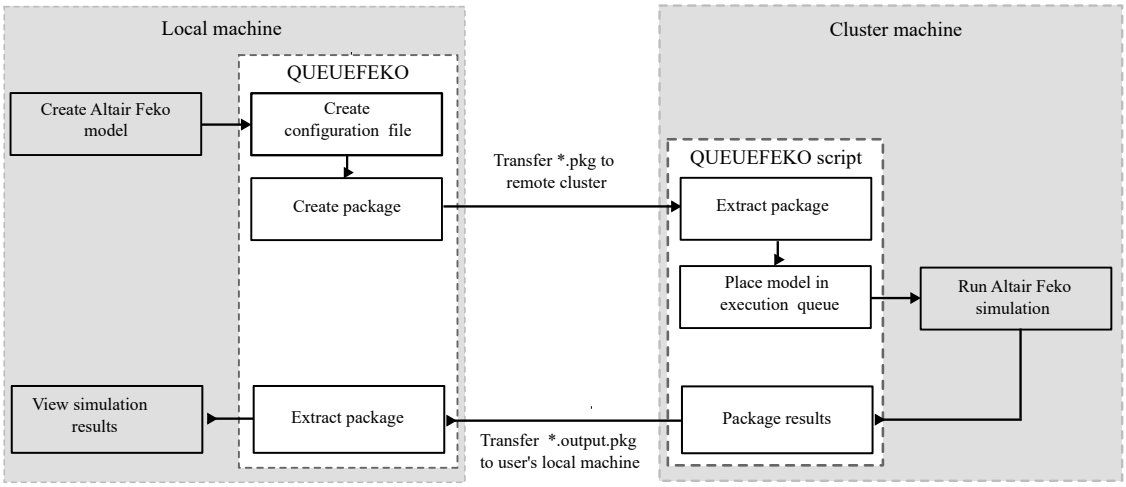


Figure 585: Remote execution scenario highlighting the role of QUEUEFEKO.

7.13.2 Creating and Extracting Packages

The steps for creating and extracting a package for remote execution, are explained.

1. Create a new configuration file (or edit an existing configuration file).
2. Set the configuration options.
3. Generate the package.
4. Add the package to the execution queue.
5. Extract the package containing the simulation results.

Package Configuration Files

A package configuration file is not the package itself but includes the settings to create a package. The file can be reused to create packages with similar settings.

QUEUEFEKO provides access to all the settings required to control the simulation and the queueing process. Once all the settings have been configured, package creation is as simple as choosing a name for the package.

Creating a Package Configuration File

Specify the settings to control the simulation and queueing process.

1. Launch QUEUEFEKO.

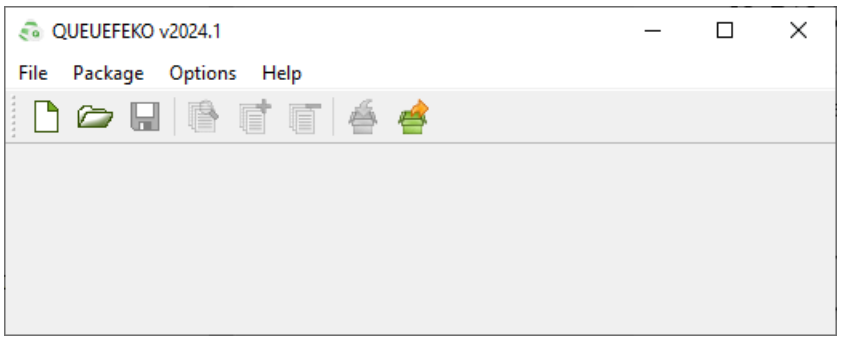


Figure 586: The component, QUEUEFEKO.

2. Click **File** > **New configuration**.

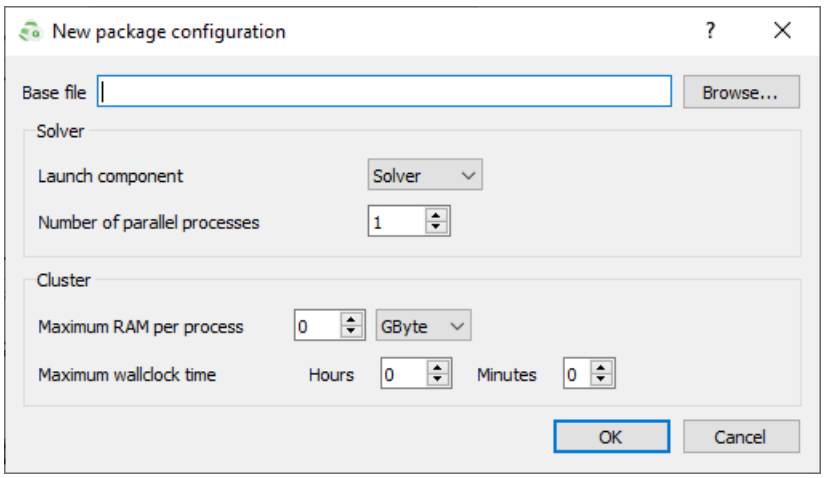


*Figure 587: The **New package configuration** dialog.*

3. In the **Base file** field, select the base `.pre` file or `.cfx` file.
4. In the **Launch component** field, select one of the following:
   - To run the Solver, select **Solver**.
   - To do an optimisation, select **OPTFEKO**.
5. In the **Number of parallel processes** field, enter the number of parallel processes allocated on the local machine.
6. In the **Maximum RAM per process** field, enter the maximum number of allowed RAM to be used on the cluster machine.
7. In the **Maximum wall clock time** field, enter the maximum allocated time to run the simulation on the cluster machine.
8. Click **OK** to close the dialog.

Adding Additional Files to the Package

The `.pre` file is read to determine the required files to perform the simulation. Files that form part of the Feko project (but not required for the simulation) can be added manually.

To add files to a package, create or open a package configuration file.

1. Click **Package** > **File list**.
2. Select one of the following options:
   - To replace the base file that was specified in the package configuration file, select **Replace base file**.
   - To add additional files to the package configuration file, select **Add file(s)**.
   - To remove a file from the package configuration file, click the **Package files** tab and select the file that you want to remove. Click **Package** > **File list** > **Remove file**.

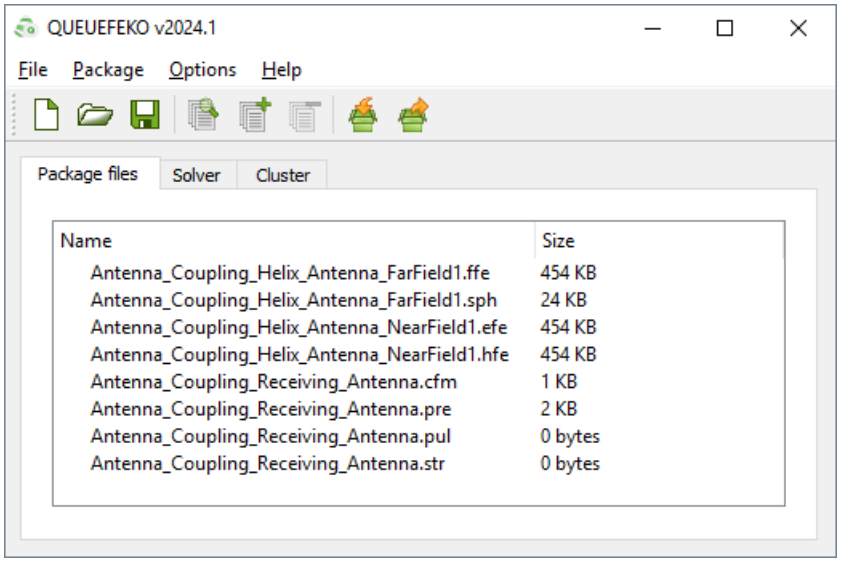


Figure 588: QUEUEFEKO is showing the list of files included in the package.

Specifying the Solver Options

For the remote cluster, you can specify the Solver options used.

1. Click the **Solver** tab.
2. In the **Launch component** field, select one of the following:
   - To run the Solver, select **Solver**.
   - To do an optimisation, select **OPTFEKO**.
     1. Click the **OPTFEKO** tab.
     2. Select the **Restart analysis number** check box if the run was discontinued and the temporary files are present. The solution can be restarted at the number of the first interrupted model.

3. Select the **Delete temporary files** check box to delete the temporary files once optimisation is complete.

   > **Note:** The optimum model and solution files are not considered as temporary files and are not deleted.

4. In the **Number of processes to be farmed out** field, specify the number of processes allocated to farming.
5. In the **Advanced** field, you can specify additional command line parameters.

3. In the **Number of parallel processes**, enter the number of parallel processes that will be used on the local machine.
4. In the **Advanced** field, specify additional command line parameters.

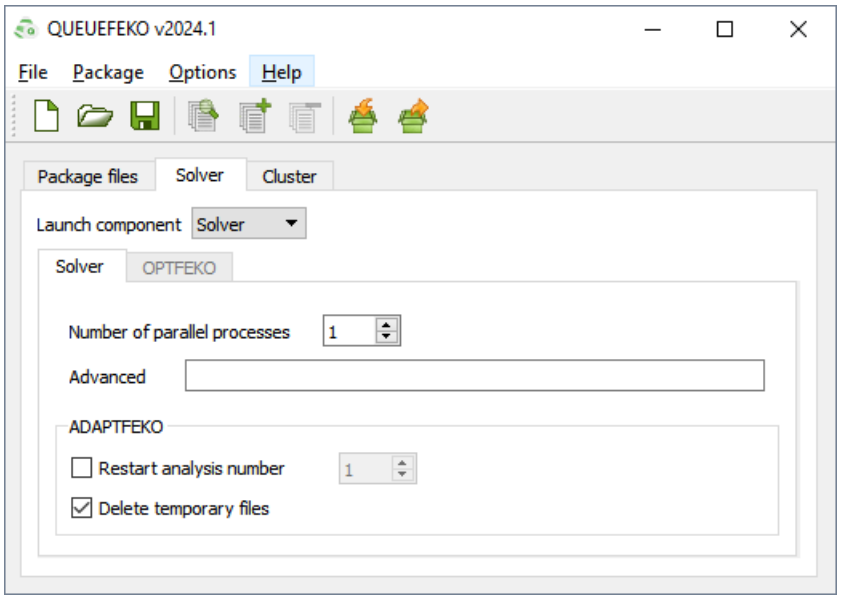


*Figure 589: The **Solver** component dialog.*

Specifying the Cluster Options

For the remote cluster, you can specify the specific job queue that is to be used and set up email notifications when the job starts or completes.

1. Under **Batch options**, specify the following:
   a) In the **Maximum RAM per process** field, enter the maximum number of RAM allocated on the cluster machine.
   b) In the **Maximum wallclock time** field, enter the maximum allocated time to simulate on the cluster machine.
   c) [Optional] In the **Queue** field, specify the job queue.
2. [Optional] Under **Notifications**, specify the following:

a) To receive email notifications when the job starts and completes, select the **Send job status information** check box.

b) In the **Email address** field, specify the email address for the notifications.

c) Under **Notifications events**, select one or more of the following options:

- **Start**
- **Abort**
- **End**

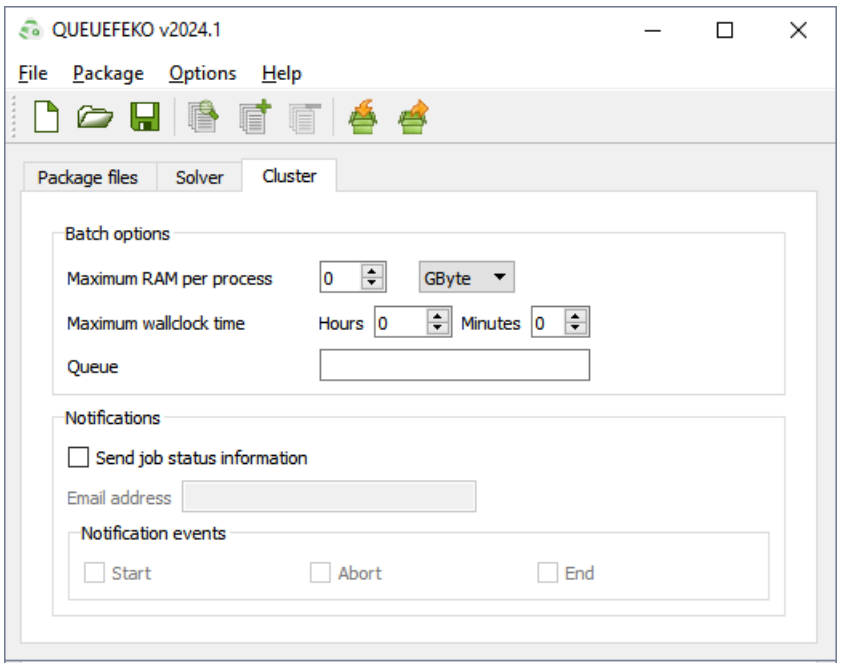


*Figure 590: The **Cluster** component dialog.*

Generating a Package

Create the package that is to be added to the execution queue.

1. Click **Package** > **Generate package**.
2. On the **Generate package file** dialog, enter the destination path of the newly created package. A package is created with a `.pkg` extension.

Adding the Package to the Execution Queue

Transfer the package to the remote cluster where it is placed in an execution queue.

If you are using the *CrunchYard* website, you can upload the package to the website and mark the package for execution. For other clusters, copy the file and manually add it to the PBS queue.

Add the package to the cluster machine using the command:

```
queuefeko mypackage.pkg
```

Simulation of the model in the package will commence as specified in the package configuration file.

Extracting the Package

After the simulation has completed, a new output package (`.output.pkg`) is available for download from the remote cluster machine to the local machine. Extract the package using QUEUEFEKO.

1. Click **Package** > **Extract package**.
2. On the **Extract package file** dialog, browse to the location of the package file.
3. On the **Extract package to** dialog, browse to the location where the contents of the package should be extracted to.
   The package is extracted. The directory contents as it was on the remote machine is made available on the local machine.

View Results

Results obtained from the remote cluster machine can be viewed in POSTFEKO on a local machine as if the simulation was run locally.

7.13.3 Setting Preferences

Specify the PDF viewer used when opening the Feko documentation.

1. Click **Options** > **Preferences**.

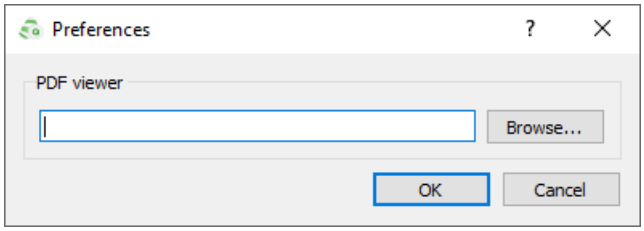


*Figure 591: The **Preferences** dialog.*

2. Under **PDF viewer**, browse to the PDF viewer of choice.
3. Click **OK** to close the dialog.

Description of the Output File of Feko

8

Feko writes all the results to an ASCII output file `.out` as well as a binary output file `.bof` for usage by POSTFEKO. Use the `.out` file to obtain additional information about the solution.

This chapter covers the following:

8.1 Geometric Data

Geometric data consist of mesh data for triangles, segments, connections between triangles and segments, dielectric cuboids as well tetrahedra for the FEM and VEP methods.

> **Note:** Geometric data is given in the .out file if it has been requested in CADFEKO or in the EG card.

8.1.1 Data For Triangles

Data for triangles consist of metallic triangles, edges, symmetry, dielectric triangles as well as advanced information for corner and end points.

Metallic Triangles

For the metallic triangles the following extract is written:

```
                        DATA OF THE METALLIC TRIANGLES

no.        label      x1 in m      y1 in m      z1 in m       edges
          medium      x2 in m      y2 in m      z2 in m
          medium      x3 in m      y3 in m      z3 in m
                        nx           ny           nz          area in m*m
        1      0   0.0000E+00   0.0000E+00   0.0000E+00            1
          Free s   0.0000E+00   2.0000E-01   0.0000E+00
          Free s   3.3333E-02   0.0000E+00   0.0000E+00
                   0.0000E+00   0.0000E+00  -1.0000E+00       3.3333E-03
        2      0   3.3333E-02   2.0000E-01   0.0000E+00           -1          2          3
          Free s   3.3333E-02   0.0000E+00   0.0000E+00
          Free s   0.0000E+00   2.0000E-01   0.0000E+00
                   0.0000E+00   0.0000E+00  -1.0000E+00       3.3333E-03
```

The first column gives the number of the triangle. The second column gives the label followed by the medium in which the triangle is situated. A **0** means that the triangle is in free space. The next three columns are the X coordinate, Y coordinate and Z coordinate of the three corner points of the triangles.

In the first row of each triangle follows another list of the numbers of the edges of the adjacent triangles. A positive sign indicates that the positive current direction is away from the triangle. A negative sign indicates that the positive current direction is towards the triangle. The area of the triangle is given below the edges in m^2.

Metallic Triangle Edges

The data for the metallic triangle edges is given after the metallic triangles. Such an edge is generated wherever two triangles have two common vertices. An additional line (or row) gives the components (nx, ny, nz) of the normal vector of each triangle.

```
                        DATA OF THE METALLIC EDGES (with MoM)

                                    triangle no.  points of tr.  information on symmetry
  no.  type  length/m   media     KORP  KORM  POIP   POIM  yz   xz   xy     status
   1    1   2.0276E-01  Free s    -1     1     2      1     1    0    0   0  unknown
   2    1   2.0000E-01  Free s    -1     2     3      3     3    0    0   0  unknown
```

```
    3    1    3.3333E-02  Free s    -1      2      7      2      2     0     0    0  unknown
```

Note: In the above table the spacing between columns was reduced to facilitate rendering the rows as single lines of data.

Each edge is assigned a consecutive number, which appears in the first column. The second column indicates the type of the edge. The third column gives the length of the edge and the fourth column gives the medium in which the edge is found. On an edge there are exactly two triangles. The columns **KORP** and **KORM** give the numbers of these two triangles and the positive current direction is from the triangle **KORP** to the triangle **KORM** . The column **POIP** gives the number of the corner point of the triangle **KORP** which is opposite to the edge. The same applies to the column **POIM**.

The next four columns contain information regarding the symmetry. The column yz gives the number of the edge corresponding to the X=0 plane (YZ plane) of symmetry. A positive sign indicates that the currents are symmetric and a negative sign indicates that the currents are anti-symmetric. If there is a **0** present in this column then a symmetric edge does not exist. The same applies to the next columns **xz** and **xy** concerning the Y=0 plane and the Z=0 plane.

If the last column with the heading **status** displays **unknown** then the edge has an unknown status. This means that the applicable coefficient of the current basis function cannot be determined from the symmetry, but has to be determined form the solution of the matrix equation. If a **0** is displayed instead then the coefficient of the current basis function is **0** due to electric or magnetic symmetry and does not have to be determined.

If there is any other number in the **status** column then this number indicates another edge for which the coefficient is equal to (positive sign in the **status** column) or the negative of (negative sign in the **status** column) the coefficient of the current basis function. From symmetry the coefficient of the current triangle does not have to be determined.

Dielectric Triangles

The data of the dielectric triangles (SEP method) is very similar to that of metallic triangles.

```
                          DATA OF THE DIELECTRIC TRIANGLES

no.        label      x1 in m      y1 in m      z1 in m       edges
          medium      x2 in m      y2 in m      z2 in m
          medium      x3 in m      y3 in m      z3 in m
                        nx           ny           nz          area in m*m
        1       0   7.1978E-01   0.0000E+00   7.1978E-01            1          2          3
                1   9.4044E-01   0.0000E+00   3.8954E-01
          Free s    8.6886E-01   3.5989E-01   3.8954E-01
                    8.2033E-01   1.6317E-01   5.4812E-01      7.2441E-02
        2       0   9.4044E-01   0.0000E+00   3.8954E-01            4          5          6
                1   1.0179E+00   0.0000E+00   0.0000E+00
          Free s    9.4044E-01   3.8954E-01   0.0000E+00
                    9.6264E-01   1.9148E-01   1.9148E-01      7.8817E-02
```

Dielectric Edges

For the dielectric edges the extract is as follows:

```
                          DATA OF THE DIELECTRIC EDGES (with MoM)

                                 triangle no.  points of tr.  electr. info on symmetry ...
no. type  length/m    media       KORP  KORM   POIP   POIM    yz    xz    xy   status    ...
```

```
  1    3    3.6694E-01   Free s   1    1      3       1       3     40    75    141   unknown    ...

  2    3    5.1069E-01   Free s   1    1      4       2       1     41    76    142   unknown    ...

  3    3    3.9718E-01   Free s   1    1     45       3       2     42    -3    143      0       ...

                                                                 magnet. info on symmetry
                                                                  yz      xz      xy   status
                                                                  40      75     141   unknown
                                                                  41      76     142   unknown
                                                                  42      -3     143   unknown
```

Note: In the above table the spacing between columns was reduced to facilitate convenient rendering of line breaks in the rows of data.

The symmetry information is shown for the basis functions for both the equivalent electric and magnetic current densities.

8.1.2 Data for Wire Segments

Data for wire segments consist of data for segments and data for nodes between segments.

Wire Segments

The data for the segments follow the data for the triangles.

```
                                DATA OF THE SEGMENTS

No.          label      x1 in m      y1 in m      z1 in m       nodes
           medium       x2 in m      y2 in m      z2 in m    length in m   radius in m
         1       0   0.0000E+00   0.0000E+00   0.0000E+00           1

           Free s    0.0000E+00   0.0000E+00   1.4286E-01   1.4286E-01    2.0000E-02
         2       0   0.0000E+00   0.0000E+00   1.4286E-01          -1             2

           Free s    0.0000E+00   0.0000E+00   2.8571E-01   1.4286E-01    2.0000E-02
         3       0   0.0000E+00   0.0000E+00   2.8571E-01          -2             3

           Free s    0.0000E+00   0.0000E+00   4.2857E-01   1.4286E-01    2.0000E-02
```

A consecutive number is assigned to each segment.

Note: This number assignment differs from the numbering from the CADFEKO `.cfm` file generated by CADFEKO.

The label of the segment is given in the second column and below that the number of the medium in which the segment is located. A zero **0** means free space (vacuum). The next columns provide the coordinates of the start and end points. The numbers of the adjacent nodes are given next to the start and the end point columns in the first row for the segments. A positive sign for the node number indicates that the positive current direction is defined away from the segment and vice versa for the negative number. The length of the segment appears in the second row, followed by the radius.

Nodes between Segments

The data of the nodes between the segments is given in a data table in the output file.

```
                    DATA OF THE NODES BETWEEN THE SEGMENTS

                no. of segment       points of segm.       info of symmetry
   No.        ISEGP      ISEGM       KNOP      KNOM       yz        xz        xy   status
       1          1          2          2         1        0         0         0   unknown

       2          2          3          2         1        0         0         0   unknown

       3          3          4          2         1        0         0         0   unknown
```

Note: In the above table the spacing between columns was reduced to facilitate rendering the rows as single lines of data.

The first column gives the consecutive numbers of the nodes. Next the numbers ISEGP and ISEGM are for the two connected segments indicating the direction of current flow: a positive current direction is defined from the segment ISEGP to the segment ISEGM. The column KNOP indicates whether the starting point (KNOP=1) of the segment ISEGP connects to the node or whether it is the end point (KNOP=2). Similarly, the column KNOM indicates whether the starting point (KNOM=1) of the segment ISEGM connects to the node or whether it is the end point (KNOM=2). The case ISEGM=0 and KNOM=0 is for half basis function connections over a single wire segment only (typically applicable to wire connections to PEC ground or to UTD faces and so forth). The following four columns contain the data for the symmetry and are the same as for the metallic triangles.

8.1.3 Connections Between Triangles and Segments

Data for connections between triangles and segments are given for triangles and segments that share connection points.

Note: The below data is only given if there are connections between triangles and segments.

```
             GEOMETRIC DATA OF CONNECTIONS SEGMENTS - TRIANGLES

     Data of triang.data of segm.                  info of symmetry
no.    DRENUM DREPOI  SEGNUM SEGPOI    angle        yz     xz     xy  status
    1                     11      1  360.0000        0      0      0  unknown
          15      1                   45.0000
          33      1                   45.0000
          55      1                   45.0000
```

Each connection point is assigned a consecutive number which is given in the first column. The column **DRENUM** gives the number of the triangle at the connection point, while the connecting vertex (1 to 3) is listed in the column **DREPOI**. Likewise the column **SEGNUM** gives the connecting segment's number and the connecting end in the **SEGPOI** column. If the start point of the segment is connected, **SEGPOI**=1, else the end point is connected and **SEGPOI**=2. The column angle gives the angle that is formed by the triangle at the connection point (in degrees). The meaning of the symmetry information in the last four columns is the same as that of the metallic triangles.

8.1.4 Dielectric Cuboids

Data for dielectric cuboids consist of the geometry information such as the medium, label and corner points as well as the basis functions.

If dielectric volume elements (cuboids) are used, then the following data block is given in the output:

```
                        DATA OF THE DIELECTRIC CUBOIDS

   No.        x1 in m      y1 in m      z1 in m
   label      x2 in m      y2 in m      z2 in m
   medium     x3 in m      y3 in m      z3 in m
              x4 in m      y4 in m      z4 in m
      1    0.0000E+00   0.0000E+00   0.0000E+00
    Cube    1.0000E-01   0.0000E+00   0.0000E+00
    wood    0.0000E+00   1.0000E-01   0.0000E+00
            0.0000E+00   0.0000E+00   1.0000E-01
      2    0.0000E+00   0.0000E+00   1.0000E-01
    Cube    1.0000E-01   0.0000E+00   1.0000E-01
    wood    0.0000E+00   1.0000E-01   1.0000E-01
            0.0000E+00   0.0000E+00   2.0000E-01
      3    0.0000E+00   0.0000E+00   2.0000E-01
    Cube    1.0000E-01   0.0000E+00   2.0000E-01
    wood    0.0000E+00   1.0000E-01   2.0000E-01
            0.0000E+00   0.0000E+00   3.0000E-01
```

Note: In the above data the cuboids are assigned the label “Cube” and the dielectric medium with label “wood.”

Each cuboid is given a consecutive number. The x , y and z corner point coordinates are given in the first three columns. The first row is the reference point. The second row is the corner point which gives the direction of the first basis function (defined from the reference point). Similarly, the third and fourth rows define the next two basis functions with respect to the reference point.

Each dielectric cuboid contains three basis functions, one in each coordinate direction. The data of these basis functions are given in the following format:

```
          DATA OF THE BASIS FUNCTIONS FOR DIELECTRIC CUBOIDS

                              Symmetry information
 No.   cuboidno. direc.     yz      xz      xy     status
  1        1        1        0       0       0     unknown
  2        2        1        0       0       0     unknown
  3        3        1        0       0       0     unknown
  4        4        1        0       0       0     unknown
  5        5        1        0       0       0     unknown
```

The first column gives the number of the basis function in consecutive numbers. The next column indicates the number of the cuboid. The column **direc.** indicates the direction of the basis function in the respective cuboid: the number **1** indicates that the basis function is defined from the reference point to the second corner point. The last four columns contain information regarding the symmetry properties of the cuboid where the structure and the meaning is the same as with the other basis functions.

8.1.5 Tetrahedra

Data for tetrahedra solved with the FEM or VEP methods consist of the label, medium, coordinates and solution method.

The data for the tetrahedral volume elements are printed in a table as follows:

```
                        DATA OF THE TETRAHEDRAL VOLUME ELEMENTS

no.     label     x1 in m     y1 in m     z1 in m       nodes
        medium    x2 in m     y2 in m     z2 in m       faces
        type      x3 in m     y3 in m     z3 in m       edges
                  x4 in m     y4 in m     z4 in m     volume in m*m*m
1       DRA1    1.0000E+00  0.0000E+00  0.0000E+00      1      2      3      4
        air     2.0000E+00  0.0000E+00  0.0000E+00      1      2      3      4
         0      2.0000E+00  1.0000E+00  0.0000E+00      1      2      3      4      5      6
                2.0000E+00  0.0000E+00  7.5000E-01      1.2500E-01
2       DRA1    1.0000E+00  0.0000E+00  0.0000E+00      1      4      5      6
        air     2.0000E+00  0.0000E+00  7.5000E-01      5      6      7      8
         0      1.0000E+00  0.0000E+00  7.5000E-01      3      7      8      9      10     11
                1.0000E+00  1.0000E+00  7.5000E-01      1.2500E-01
3       DRA1    2.0000E+00  1.0000E+00  0.0000E+00      3      4      6      7
        air     2.0000E+00  0.0000E+00  7.5000E-01      9      10     11     12
         0      1.0000E+00  1.0000E+00  7.5000E-01      6      12     13     10     14     15
                2.0000E+00  1.0000E+00  7.5000E-01      1.2500E-01
```

Note: In the above table the spacing between columns was reduced to facilitate rendering the rows as single lines of data.

The consecutive numbers of the elements are given in the first column. Column two contains 3 entries in the following order:

1. The label of the element.
2. The medium name of the element.
3. The type of tetrahedra.

The type of tetrahedra makes the distinction between the solution method and whether the element is a dielectric and/or a magnetic element as follows:

Table 64: Types of Tetrahedral Volume Elements

Type	Description/Method
0	FEM element (dielectric and/or magnetic)
1	VEP dielectric element
2	VEP magnetic element
3	VEP dielectric and magnetic element
5	Metallic FEM element

Columns 3, 4 and 5 provide the X coordinate, Y coordinate and Z coordinate of the vertices of the element. The numbers of each node, face and edge bounding the tetrahedral element follow in the last columns.

8.1.6 Data for Memory Usage

The data for the memory usage shows the number of mesh elements and basis functions which can be an indication of the memory usage.

In the table below it is also indicated how many basis functions have the status “unknown”, that is, how many basis functions have to be determined by solving the matrix equation.

```
                              DATA FOR MEMORY USAGE

Number of metallic triangles:          0          max. triangles:    MAXNDR    = 176
Number of dielectric triangles:      176
Number of aperture triangles:          0
Number of RL-GO triangles:             0
Number of windscreen triangles:        0
Number of FEM surface triangles:       0
Number of modal port triangles:        0
Number of metallic segments:           0          max. segments:     MAXNSEG   =   0
Number of combined MoM/MTL segments:   0
Number of dielectr./magnet. cuboids:   0          max. cuboids:      MAXNQUA   =   0
Number of tetrahedra:                  0          max. tetrahedra:   MAXNTETRA =   0
Number of edges in PO region:          0          max. edges:        MAXPOKA   =   0
Number of wedges in PO region:         0          max. wedges:       MAXPOKL   =   0
Number of Fock regions:                0          max. Fock regions: MAXFOGE   =   0
Number of polygonal surfaces:          0          max. surfaces:     MAXPOLYF  =   0
                                                  max. corner pts.:  MAXPOLYP  =   0
Number of UTD cylinders:               0

Number of metallic edges (MoM):        0  unknown:   0 (electr.) max. edges: MAXNKA=264
                                       0  unknown:   0 (magnet.)
Number of metallic edges (PO):         0  unknown:   0 (electr.)
                                          unknown:   0 (magnet.)
Number of dielectric edges (MoM):    264  unknown:  66 (electr.)
                                     264            66 (magnet.)
Number of dielectric edges (PO):       0  unknown:   0 (electr.)
                                          unknown:   0 (magnet.)
Number of aperture edges (MoM):        0  unknown:    0 (magnet.)
Number of edges FEM/MoM surface:       0  unknown:    0 (electr.)
                                       0              0 (magnet.)
Number of nodes between segments:      0  unknown:    0      max. nodes: MAXNKNO   =   0
Number of connection points:           0  unknown:    0      max. conn.: MAXNV     =   0
Number of dielectric cuboids:          0  unknown:    0      max. cuboids: MAXNQUA =   0
Number of magnetic cuboids:            0  unknown:    0
Number of dielectric faces (VEP):      0  unknown:    0
Number of magnetic faces (VEP):        0  unknown:    0

Number of basis funct. for MoM:      528  unknown: 132    max. basisf.  MAXNZEILE = 528
Number of basis funct. for PO:         0  unknown:   0    max. basisf.  MAXNKAPO  =   0
```

Note: In the above table the spacing between characters and entries was reduced to facilitate rendering the rows as single lines of data.

8.2 Excitation

Excitation data consist of voltage sources, plane waves, waveguide sources, equivalent sources and Hertzian dipoles.

Voltage Sources on Segments

A voltage source on a segment generates the following data block:

```
                    EXCITATION BY VOLTAGE SOURCE AT A SEGMENT

Name:
Excitation index:                1
Frequency in Hz:                 FREQ =    7.49481E+07
Wavelength in m:                 LAMBDA =  4.00000E+00
Open circuit voltage in V:       |U0| =    1.00000E+00
Phase in degrees:                ARG(U0) =        0.00
Attached to port:                Segment port
Port at segment with label:      1
Absolute number of segment:      11
Radius of segment in m:          2.00000E-03
Location of the port in m:       x =  0.00000E+00, y =  0.00000E+00, z =  0.00000E+00
Positive direction:              x =  0.00000E+00, y =  0.00000E+00, z =  1.00000E+00
```

Similar information is provided for other voltage sources (such as a voltage source on an edge port and on a microstrip port and a voltage source connected to a general network).

Waveguide Ports and Waveguide Sources

Data for the port and the source are split into three blocks. The geometrical data for the port is given first:

```
                          DATA FOR WAVEGUIDE PORTS

Waveguide port label: Port
Port type: Rectangular
Port dimensions
  Width:          1.29600E-01 m
  Height:         6.48000E-02 m
Port reference points in m:
  Point S1:               x = -2.37927E-01, y = -6.48000E-02, z = -3.24000E-02
  Point S2:               x = -2.37927E-01, y =  6.48000E-02, z = -3.24000E-02
Direction of propagation: x =  1.00000E+00, y =  0.00000E+00, z =  0.00000E+00
```

Subsequently follows the data for the modes.

```
                       MODE EXPANSION DATA OF A WAVEGUIDE PORT

Waveguide port label: Port
Frequency:         1.64500E+09 Hz

Mode indices  Cutoff freq.  Transverse wave impedance  Propagation factor  Description
      m   n     in Hz      real part   imag. part   real part  imag. part
TE    0   1   2.3132E+09  0.0000E+00  3.8105E+02  3.4085E+01  0.0000E+00   Evanescent
TE    0   2   4.6264E+09  0.0000E+00  1.4331E+02  9.0626E+01  0.0000E+00   Evanescent
TE    1   0   1.1566E+09  5.2979E+02  0.0000E+00  0.0000E+00  2.4515E+01   Propagating
TE    1   1   2.5862E+09  0.0000E+00  3.1053E+02  4.1826E+01  0.0000E+00   Evanescent
TE    1   2   4.7688E+09  0.0000E+00  1.3845E+02  9.3812E+01  0.0000E+00   Evanescent
TE    2   0   2.3132E+09  0.0000E+00  3.8105E+02  3.4085E+01  0.0000E+00   Evanescent
TE    2   1   3.2713E+09  0.0000E+00  2.1916E+02  5.9264E+01  0.0000E+00   Evanescent
TE    2   2   5.1725E+09  0.0000E+00  1.2637E+02  1.0277E+02  0.0000E+00   Evanescent
```

```
TM    1   1   2.5862E+09  0.0000E+00  4.5703E+02  4.1826E+01  0.0000E+00   Evanescent
TM    1   2   4.7688E+09  0.0000E+00  1.0251E+03  9.3812E+01  0.0000E+00   Evanescent
TM    2   1   3.2713E+09  0.0000E+00  6.4758E+02  5.9264E+01  0.0000E+00   Evanescent
TM    2   2   5.1725E+09  0.0000E+00  1.1230E+03  1.0277E+02  0.0000E+00   Evanescent
```

Lastly the data for the impressed waveguide mode is provided:

```
                    EXCITATION BY IMPRESSED WAVEGUIDE MODE

Name:
Excitation index:                1
Frequency in Hz:                 FREQ =    1.64500E+09
Wavelength in m:                 LAMBDA =  1.82245E-01
Impressed mode:                  TE 1 0
Amplitude in A/m:                 1.00000E+00 * PWFAKTOR (see below)
Phase in degrees:                        0.00
Transmitted power in W:           1.13772E+00 * PWFAKTOR^2 (see below)
Attached to port label:          Port
```

FEM Current Source

For a FEM excitation (impressed current source) the following information is provided:

```
                    EXCITATION BY IMPRESSED CURRENT ELEMENT (FEM)

Name:                            CurrentSource1
Excitation index:                1
Frequency in Hz:                 FREQ =    3.00000E+09
Wavelength in m:                 LAMBDA =  6.54669E-02
Amplitude in A:                  |I| =     1.00000E+00
Phase in degrees:                ARG(I) =     0.00
Attached to port:                FEM line port
Start point of the port in m:    x =  0.00000E+00, y =  6.50000E-03, z = -1.00000E-03
End point of the port in m:      x =  0.00000E+00, y =  6.50000E-03, z =  0.00000E+00
Port length in m:                1.00000E-03
```

FEM Modal Source

The data for this source is split into two blocks of data. The mode expansion for the port is given first:

```
                        MODE EXPANSION DATA OF A MODAL PORT

FEM modal port: Port1
Frequency:        3.00000E+09 Hz

                       Eigenvalues computed with ARPACK [Z]

              Mode            Propagation factor       Description
             counter      real part   imaginary part

                1       0.00000E+00      9.59751E+01   Propagating (fundamental mode)
```

Next follows the data for the source:

```
                        EXCITATION BY IMPRESSED MODAL PORT MODE

Name:                            FEMModalSource1
Excitation index:                1
Frequency in Hz:                 FREQ =    3.00000E+09
Impressed mode:                  Fundamental
Amplitude in V/m:                 1.00000E+00
Phase in degrees:                        0.00
Transmitted power in W:           5.00000E-01
Attached to port:                Port1
```

Plane Wave Source

If an incident plane wave is used then the output file has the following format:

```
               EXCITATION BY INCIDENT PLANE ELECTROMAGNETIC WAVE

Name:
Excitation index:                1
Frequency in Hz:                 FREQ =    1.49896E+07
Wavelength in m:                 LAMBDA =  2.00000E+01
Direction of incidence:          THETA = -180.00  PHI =    0.00
Polarisation:                    LINEAR
Axial ratio:                     V =      0.0000
Polarisation angle:              ETA =    0.00
Direction of propag.:            BETA0X =  0.00000E+00
  (unit vector)                  BETA0Y =  0.00000E+00
                                 BETA0Z =  1.00000E+00
Wave number:                     BETA0  = ( 3.14159E-01 +j 0.00000E+00)
Phase reference point in m:      x =  0.00000E+00, y =  0.00000E+00, z =  0.00000E+00
Field strength in V/m:           |E0X| =  1.00000E+00   ARG(E0X) =     0.00
  (Phase in degrees)             |E0Y| =  0.00000E+00   ARG(E0Y) =     0.00
                                 |E0Z| =  0.00000E+00   ARG(E0Z) =     0.00
```

The vector **β**, whose components are given, is the vector which points in the direction of propagation. The vector $\mathbf{E_0}$ represents the direction of the electric field.

Near Field Source

For an impressed near field (aperture) source, the following information is given:

```
                          EXCITATION BY NEAR FIELD SOURCE

Name:                            NearFieldSource1
Aperture number:                 1
Frequency in Hz:                 FREQ =    1.64500E+09
Wavelength in m:                 LAMBDA =  1.82245E-01
Number of electric dipoles:      90 (of which 20 suppressed)
Number of magnetic dipoles:      90 (of which 32 suppressed)
Extent of the source:            X = -5.83200E-02 ...  5.83200E-02 m
                                 Y = -2.59200E-02 ...  2.59200E-02 m
                                 Z = -2.56439E-01 ... -2.56439E-01 m
```

Note: No specific information regarding the magnitude and phase of the dipole elements that make up the excitation is given in the output.

Impressed Radiation Pattern

Excitation by an impressed radiation pattern point source is shown in the output as follows:

```
                     EXCITATION BY A FAR FIELD POINT SOURCE

Name:                            RadiationPattern1
Excitation index:                1
Frequency in Hz:                 FREQ =    1.60000E+09
Wavelength in m:                 LAMBDA =  1.87370E-01
Max. field strength * dist. in V:   5.91419E+01 * PWFAKTOR (see below)
Radiated power in W:                1.12755E+00 * PWFAKTOR^2 (see below)
Directivity of the antenna in dB:      17.138
Distance for far field cond. in m:  1.96442E+00
Source position in m:            x =  0.00000E+00, y =  0.00000E+00, z =  0.00000E+00
Number of grid points  NTHETA = 37
                       NPHI   = 73
```

```
Angular range THETA in degrees:    0.00 ... 180.00
              PHI   in degrees:    0.00 ... 360.00
```

Note: No specific information regarding the magnitude and phase of the impressed pattern that make up the excitation is given in the output.

Spherical Mode Source

For an impressed spherical mode source, the following information is written to the output:

```
                  EXCITATION BY AN IMPRESSED SPHERICAL MODE

Name:                            SphericalModesSource1
Excitation index:                1
Frequency in Hz:                 FREQ =    6.25000E+09
Wavelength in m:                 LAMBDA =  4.79668E-02
Source position in m:            x =  0.00000E+00, y =  0.00000E+00, z =  2.54000E-01
Rotation about the axes:         X = -180.00
                                 Y =   48.46
                                 Z =    0.00
Number of modes:                   880 (of which 38 suppressed)
Propagation direction:         C = 4 (outwards)
          mode indices            coefficient in sqrt(Watt)   rad. power(Watt)
      J       S       M       N         magn.        phase         power
      1       1      -1       1      2.54050E-05    -25.23      3.22706E-10
      2       2      -1       1      4.50632E-05      1.04      1.01535E-09
      3       1       0       1      5.41621E-07     40.48      1.46677E-13
```

Hertzian Dipoles

Point source type (Hertzian dipole) excitations result in the following information output:

```
                        EXCITATION BY ELECTRIC DIPOLE

Name:                            ElectricPointSource1
Excitation index:                1
Frequency in Hz:                 FREQ =    3.00000E+08
Wavelength in m:                 LAMBDA =  9.99308E-01
Amplitude in Am:                 |IL| =    1.00000E+00
Phase in degrees:                ARG(IL) =        0.00
Dipole position in m:            x =  0.00000E+00, y =  0.00000E+00, z =  0.00000E+00
Orientation of dipole:           THETA =    0.00
                                 PHI   =    0.00
```

The above output is for an electric dipole. The magnetic dipole will have similar output.

8.3 Currents and Charges

Currents and charges data are supported for triangles and wire segments and can be requested from CADFEKO or with the OS card.

Currents on Triangles

The currents and charges data for triangles are provided as follows:

```
            VALUES OF THE CURRENT DENSITY VECTOR ON TRIANGLES in A/m (no averaging)


Triangle      centre                                    JX                JY          ...
number  x/m          y/m          z/m          magn.     phase    magn.    phase   ...
 1    1.11111E-01  1.11111E-01  0.00000E+00 1.099E-02  147.51 1.099E-02  147.51   ...
 2    1.01903E-01  8.28341E-01  0.00000E+00 1.955E-04   62.22 4.668E-03  121.95   ...
 3    1.99385E-01  6.49031E-01  0.00000E+00 2.036E-03  131.69 5.377E-03  128.94   ...
 4    9.40494E-02  1.16528E+00  0.00000E+00 1.019E-04   29.26 3.685E-03  109.98   ...
 5    8.89886E-02  1.49041E+00  0.00000E+00 8.437E-05  -33.65 2.744E-03  100.59   ...

                                          JZ                Current magnitude in the
                                    magn.     phase              3 corner points
                                 1.099E-02  147.51    1.495E-02  3.067E+00  1.495E-02
                                 4.668E-03  121.95    4.520E-03  4.695E-03  4.842E-03
                                 5.377E-03  128.94    5.591E-03  6.001E-03  5.709E-03
                                 3.685E-03  109.98    3.697E-03  3.718E-03  3.660E-03
                                 2.744E-03  100.59    2.768E-03  2.630E-03  2.850E-03
```

The current density vector **J** in the complex form is given at the position (X, Y, Z). The last three columns indicate the value for the surface current density in the three vertices of the triangles.

> **Note:** The value of the current written in the `.out` file will be affected if averaging of the currents is de-activated in the OS card. If averaging is requested, the average of the current at the vertices of all three adjacent triangles is shown.

Charges on Triangles

If the current is requested, the charge on each triangle is also written to the output file. Only the charge is given as the position of each triangle is the same as written for the currents.

```
          VALUES OF THE SURFACE CHARGE DENSITY ON TRIANGLES in As/m^2

Triangle            SIGMA
number         magn.      phase
        1    2.03115E-11  165.31
        2    8.14289E-12  152.22
        3    1.00211E-11  160.36
        4    4.59629E-12  119.26
        5    4.02388E-12   56.35
```

Currents on Wire Segments

The current on the segments is written as follows:

```
                    VALUES OF THE CURRENT IN THE SEGMENTS in A

Segment centre                                  IX             IY             IZ
number  x/m        y/m      z/m         magn.  phase   magn.   phase   magn.    phase
```

```
1   0.000E+00   0.000E+00   1.66551E-01   0.00E+00   0.00 0.00E+00   0.00 1.837E-02   -31.39
2   0.000E+00   0.000E+00   4.99654E-01   0.00E+00   0.00 0.00E+00   0.00 1.410E-02   -33.86
3   0.000E+00   0.000E+00   8.32757E-01   0.00E+00   0.00 0.00E+00   0.00 5.366E-03   -35.06
```

Note: In the above table the spacing between columns and the number of significant digits were reduced to facilitate rendering the rows as single lines of data.

Charges on Wire Segments

If the currents on segments are requested, the charges are also written to the output file as follows:

```
          VALUES OF THE LINE CHARGE DENSITY ON SEGMENTS in As/m

Segment                Q
number          magn.       phase
        1    1.32488E-11  -90.69
        2    4.30863E-11 -120.06
        3    6.83730E-11 -125.06
```

Currents and Associated Data for Voltage Sources

For every voltage source the current at the feed point is determined and therefore the impedance and other related parameters as follows:

```
                        DATA OF THE VOLTAGE SOURCE NO. 1

                   real part  imag. part   magnitude       phase
 Current   in A    1.6718E-02 -9.5781E-03  1.9268E-02      -29.81
 Admitt.   in A/V  1.6718E-02 -9.5781E-03  1.9268E-02      -29.81
 Impedance in Ohm  4.5034E+01  2.5800E+01  5.1901E+01       29.81
 Inductance  in H  5.4750E-08
```

8.4 Finite Conductivity

Finite conductivity data consists of the material parameters and associated data such as skin effect penetration depth, conductor impedance as well as the power losses per label.

The block with the set of characteristics for the single labels is displayed first as follows:

```
                          DATA FOR LABELS

Label Cuboid1.Face3:
         Metallic conductor (skin effect)
         Surface thickness:  5.00000E-03 m
         Sigma =   1.000E+02 S/m   Mue_r =   1.000E+00   tan(delta_mue) =   0.000E+00
         Penetration depth of the skin effect:  5.81365E-03 m
         Conductor impedance due to the skin effect: (  2.098E+00 +j   9.799E-01) Ohm
```

After the calculation of the currents the losses that result from finite conductivity are displayed as follows:

```
                             POWER LOSS METAL (in Watt)

Results for labels     |                 in the segments                  | in the
  Label                | skineffect  conc.load   distr.load   coating     | triangles
         Cuboid1.Face3 | 0.0000E+00  0.0000E+00  0.0000E+00   0.0000E+00  | 8.9357E-06
         Cuboid1.Face4 | 0.0000E+00  0.0000E+00  0.0000E+00   0.0000E+00  | 1.2433E-06
         Cuboid1.Face1 | 0.0000E+00  0.0000E+00  0.0000E+00   0.0000E+00  | 1.4940E-06
         Cuboid1.Face6 | 0.0000E+00  0.0000E+00  0.0000E+00   0.0000E+00  | 4.0082E-07
         Cuboid1.Face2 | 0.0000E+00  0.0000E+00  0.0000E+00   0.0000E+00  | 1.2433E-06
         Cuboid1.Face5 | 0.0000E+00  0.0000E+00  0.0000E+00   0.0000E+00  | 1.4940E-06
  Sum:                 | 0.0000E+00  0.0000E+00  0.0000E+00   0.0000E+00  | 1.4811E-05

         Total loss in segments:                      0.0000E+00 W
         Total loss in triangles:                     1.4811E-05 W

         Sum of losses in metallic elements:          1.4811E-05 W


                          SUMMARY OF LOSSES

         Metallic elements:                           1.4811E-05 W
         Dielectric:                                  0.0000E+00 W
         Mismatch at feed:                            0.0000E+00 W
         Non-radiating networks:                      0.0000E+00 W
         Cables:                                      0.0000E+00 W
         Backward power at passive waveguide ports:   0.0000E+00 W
         Backward power at passive modal ports:       0.0000E+00 W
                                                     -------------
               Sum of all losses:                     1.4811E-05 W

         Efficiency of the antenna:                      99.7162 %
         (based on a total active power:              5.2196E-03 W)
```

For the power losses, in the first column the label is displayed while the lowest row displays the sum of the losses over all labels.

8.5 Near Fields and SAR

Near fields and SAR data consist of values for the electric and magnetic field strength as well as specific absorbtion rate (SAR) data.

Electric Field Strength

The position as well as the individual components of the electric and the magnetic field strength are given. Unless otherwise requested in the request, the total value of the field, that is the sum of the incident wave and the scattered field, is given.

```
                        VALUES OF THE ELECTRIC FIELD STRENGTH in V/m

                             (total field, incident and scattered)

              LOCATION                                  EX                      EY          ...
medium   X/m         Y/m          Z/m          magn.      phase       magn.      phase ...
 0  -1.00000E-01  0.0000E+00  1.25600E-01  2.0790E+01 -136.15  1.5550E+02   3.60  ...
 0  -8.87711E-02  0.0000E+00  1.25600E-01  2.1755E+01 -144.07  1.6368E+02  -5.91  ...
 0  -7.75422E-02  0.0000E+00  1.25600E-01  2.2504E+01 -150.94  1.6937E+02  -14.44 ...

                                                      EZ
                                                 magn.      phase
                                              1.46636E+02 -155.96
                                              1.53094E+02 -165.59
                                              1.57508E+02 -174.30
```

> **Note:** In the above table the spacing between columns was reduced to facilitate convenient rendering of line breaks in the rows of data.

The magnetic field strength data will have very similar contents.

Electric Fields inside Dielectric Cuboids

If the electric fields inside dielectric cuboids are requested (**Electric field and SAR values** request in the FE card) then the value for the SAR and the cuboid number are also given as follows:

```
             VALUES OF THE ELECTRIC FIELD STRENGTH in V/m

                  inside the dielectric cuboids

   LOCATION            EX                EY                EZ           SAR  cuboid no.
 X/m    Y/m    Z/m     magn.   phase     magn.    phase    magn.    phase    in W/kg
 0.128  0.128  0.128 6.129E-01  -5.68  3.481E-04   -1.66 3.623E-02  87.60  0.0E+00   1
 0.128  0.128 -0.128 6.127E-01   4.80  3.479E-04   -0.42 3.622E-02  91.27  0.0E+00   2
 0.128  0.128  0.383 6.046E-01  -16.07 3.582E-04   16.84 3.651E-02  85.05  0.0E+00   3
 0.128  0.128 -0.383 6.042E-01   15.19 3.594E-04  -18.87 3.648E-02  93.84  0.0E+00   4
 0.128  0.128  0.637 5.851E-01  -26.50 2.032E-03  156.46 3.657E-02  79.21  0.0E+00   5
```

> **Note:** In the above table the spacing between columns and the number of significant digits were reduced to facilitate convenient rendering of line breaks in the rows of data.

SAR

For specific SAR solution requests, the following output is shown (note that the extract shown below is representative for a spatial peak SAR calculation. The output for other options like volume average SAR calculations will differ.)

```
                            SPATIAL-PEAK SPECIFIC ABSORPTION RATE in W/kg
                              for 10.0 g tissue in the shape of a cube

  Search includes entire domain
  Maximum volume fraction of air allowed in a SAR averaging cube:  20.0 %

   cube centre                            cube edge      mass       AR    tissue content
 x in m        y in m       z in m        in m          in g       in W/kg      in %
1.0255E-01  3.3344E-02  1.0833E-02  2.1728E-02   9.9082E+00  2.82865E+00    93.07

          orientation unit vectors of the SAR cube
             x in m        y in m        z in m
           5.98050E-01   1.03718E-01  -7.94719E-01
           8.00460E-01  -2.78161E-02   5.98740E-01
           3.99940E-02  -9.94218E-01  -9.96574E-02
```

Note: In the above table the spacing between columns and the number of significant digits were reduced to facilitate convenient rendering of the data.

8.6 Far Fields and Receiving Antennas

Far fields and receiving antennas data consist of the electric far field data, RCS, gain, directivity and radiated power.

Far Fields and Polarisation Types

If the far field is calculated, the following block in this form is displayed:

```
      VALUES OF THE SCATTERED ELECTRIC FIELD STRENGTH IN THE FAR FIELD in V
                    Factor e^(-j*Re{BETA}*R)/R not considered

    LOCATION          ETHETA             EPHI              directivity in dB      ...
 THETA   PHI      magn.    phase     magn.    phase     vert.     horiz.    total   ...
 0.00    0.00   2.626E-16 -178.03  2.321E-16  22.06  -308.6129 -309.6881 -306.1070 ...
 2.00    0.00   7.271E-02  104.04  0.000E+00   0.00   -19.7678 -999.9999  -19.7678 ...
 4.00    0.00   1.449E-01  104.02  0.000E+00   0.00   -13.7772 -999.9999  -13.7772 ...

                                                 POLARISATION
                                               axial r. angle   direction
                                               0.1758   138.76   RIGHT
                                               0.0000   180.00   LINEAR
                                               0.0000   180.00   LINEAR

    Gain is a factor of  1.00000E+00 (    0.00 dB) larger than directivity

      The directivity/gain is based on an active power of  8.35911E-03 W
      and on a power loss of  0.00000E+00 W
```

Note: In the above table the spacing between columns and the number of significant digits were reduced to facilitate convenient line breaks and rendering of the data.

For incident plane waves, the displayed values are the values of the scattered field, that is the incident field is not taken into account. However, for any other sources (such as elementary Hertzian dipoles or impressed radiation patterns), the fields radiated by the source are included.

In the far field a complex field strength $\mathbf{E}_{\text{far}}$ is defined using the relation

$$\lim_{R\to\infty} \mathbf{E}(\mathbf{r}) = \frac{e^{-j\beta_0 R}}{R}\mathbf{E}_{\text{far}} \tag{108}$$

with a large distance $R=|\mathbf{r}|$ which tends to infinity (and which in the Feko calculations is identical to infinity).

Note: The dimension of $\mathbf{E}_{\text{far}}$ is voltage due to this extra distance factor R.

In the `.out` file the ϑ (vertical) and φ (horizontal) components of $\mathbf{E}_{\text{far}}$ are tabulated by magnitude and phase, that is $\mathbf{E}_{\text{far},\vartheta}$ and $\mathbf{E}_{\text{far},\varphi}$.

Using POSTFEKO results for other polarizations can be extracted. The corresponding formulas are as follows:

S-polarisation:

$$\mathbf{E}_{far,S}=\frac{1}{\sqrt{2}}\left(\mathbf{E}_{far,\varphi}-\mathbf{E}_{far,\vartheta}\right) \tag{109}$$

Z-polarisation:

$$\mathbf{E}_{far,Z}=\frac{1}{\sqrt{2}}\left(\mathbf{E}_{far,\varphi}+\mathbf{E}_{far,\vartheta}\right) \tag{110}$$

left-hand circular polarisation:

$$\mathbf{E}_{far,LHC}=\frac{1}{\sqrt{2}}\left(\mathbf{E}_{far,\varphi}+j\mathbf{E}_{far,\vartheta}\right) \tag{111}$$

right-hand circular polarisation:

$$\mathbf{E}_{far,RHC}=\frac{1}{\sqrt{2}}\left(\mathbf{E}_{far,\varphi}-j\mathbf{E}_{far,\vartheta}\right) \tag{112}$$

If a plane wave is included, the results will include the radar cross section. In the case of other sources without a plane wave source, the gain or directivity is included.

Note: If a plane wave is combined with, for example, a voltage source, the active RCS is obtained, but the gain/directivity will not be computed.

Radar Cross Section

For the radar cross section, the incident plane wave with complex amplitude $\mathbf{E}_0$ carries a power density of

$$S_i=\frac{1}{2}\cdot\frac{|\mathbf{E}_0|^2}{Z_{F0}} \tag{113}$$

where Z_{F0} denotes the wave impedance of the surrounding medium. The incident wave gets scattered on the object and a wave is reflected with the scattered power density

$$S_s=\frac{1}{2}\cdot\frac{|E_\vartheta|^2+|E_\varphi|^2}{Z_{F0}} \tag{114}$$

The radar cross section (RCS) σ is then defined as follows:

$$\sigma_{Total}=\lim_{R\to\infty}4\pi R^2\frac{S_s}{S_i}=\lim_{R\to\infty}4\pi\frac{|RE_\vartheta|^2+|RE_\varphi|^2}{|\mathbf{E}_0|^2}=4\pi\frac{|E_{far,\vartheta}|^2+|E_{far,\varphi}|^2}{|\mathbf{E}_0|^2} \tag{115}$$

$$\sigma_{Horizontal}=\lim_{R\to\infty}4\pi\frac{|RE_\varphi|^2}{|\mathbf{E}_0|^2}=4\pi\frac{|E_{far,\varphi}|^2}{|\mathbf{E}_0|^2} \tag{116}$$

$$\sigma_{Vertical}=\lim_{R\to\infty}4\pi\frac{|RE_\vartheta|^2}{|\mathbf{E}_0|^2}=4\pi\frac{|E_{far,\vartheta}|^2}{|\mathbf{E}_0|^2} \tag{117}$$

Gain and Directivity

For antenna and general radiation problems Feko computes either the gain or the directivity depending far field request setting.

> **Note:** The gain/directivity setting applies to the values tabulated in the `.out` file only. Any quantity can be selected in POSTFEKO.

Assume that P_t is the source power and P_v are losses in the structure (such as dielectric losses), then a power of $P_r = P_t - P_v$ will be radiated. The directivity (relative to an isotropic point source) is then defined as follows:

$$D_{Total} = 4\pi R^2 \frac{S_s}{P_r} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\vartheta}|^2 + |E_{far,\varphi}|^2}{P_r} \tag{118}$$

$$D_{Horizontal} = \frac{2\pi}{Z_{F0}} z \frac{|E_{far,\varphi}|^2}{P_r} \tag{119}$$

$$D_{Vertical} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\vartheta}|^2}{P_r} \tag{120}$$

For the gain a similar definition is used, except that the source power P_t and not the radiated power P_r is acting as reference as follows:

$$G_{Total} = 4\pi R^2 \frac{S_s}{P_t} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\vartheta}|^2 + |E_{far,\varphi}|^2}{P_t} \tag{121}$$

$$G_{Horizontal} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\varphi}|^2}{P_t} \tag{122}$$

$$G_{Vertical} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\vartheta}|^2}{P_t} \tag{123}$$

Between gain and directivity the following relation holds:

$$\frac{G}{D} = \frac{P_r}{P_t} = \frac{P_t - P_v}{P_t} = \eta \tag{124}$$

where η represents the antenna efficiency.

Polarisation and Axial Ratio

The last three columns of the far field output give the polarisation information of the scattered wave. In general the polarisation is elliptical as shown in the figure.

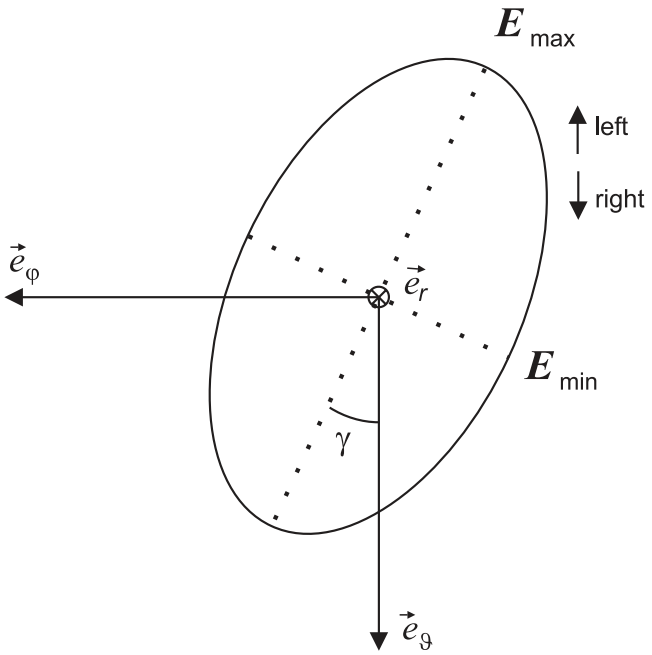


Figure 592: Elliptic polarisation in the far field.

The coordinates are $\vec{e}_r$, $\vec{e}_\vartheta$ and $\vec{e}_\varphi$, and the view is in the direction of the propagation of the wave ($\vec{e}_r$).

To evaluate these quantities, the magnitude and phase of the far field components are defined as follows:

$$E_{far,\vartheta} = A \cdot e^{j\alpha} \qquad E_{far,\varphi} = B \cdot e^{j\beta} \tag{125}$$

Using the abbreviation $\tau = \omega t - \beta_{0r}$ the temporal field strength vector in space can be written as:

$$\vec{E}(\tau) = \frac{A}{r} \cdot \cos(\tau + \alpha) \cdot \vec{e}_\vartheta + \frac{B}{r} \cdot \cos(\tau + \beta) \cdot \vec{e}_\varphi \tag{126}$$

This equation describes the polarisation ellipse depicted in the figure.

The minimum and maximum values of the field strength magnitude can be found at following times:

$$\tau_1 = -\frac{1}{2} \cdot \arctan \frac{A^2 \cdot \sin(2\alpha) + B^2 \cdot \sin(2\beta)}{A^2 \cdot \cos(2\alpha) + B^2 \cdot \cos(2\beta)} \tag{127}$$

and

$$\tau_2 = \tau_1 + \frac{\pi}{2} \tag{128}$$

Let $E_1 = |\mathbf{E}(\tau_1)|$ and $E_2 = |\mathbf{E}(\tau_2)|$ and assume that $E_1 > E_2$, then according to the figure $E_{max} = E_1$ and $E_{min} = E_2$.

The axial ratio (Minor/Major) is defined as

$$v = \frac{E_{min}}{E_{max}} = \frac{E_2}{E_1} \tag{129}$$

The axial ratio (Major/Minor) is defined as

$$v = \frac{E_{max}}{E_{min}} = \frac{E_1}{E_2} \tag{130}$$

A ratio (Minor/Major) of 0 means that the wave is a linearly polarised wave, but if the ratio (Minor/Major) has a value of 1 then it is a circularly polarised wave. The direction of rotation is right hand circular (RHC) for $0 < \alpha - \beta < \pi$ and left hand circular (LHC) for $\pi < \alpha - \beta < 2\pi$.

Feko also computes and prints the polarisation angle γ. It is the angle between the major axis of the polarisation ellipse and the unit vector $\vec{e}_\vartheta$ and can be computed using

$$\gamma = \arctan\frac{B \cdot \cos(\tau_1 + \beta)}{A \cdot \cos(\tau_1 + \alpha)} \tag{131}$$

Poynting Vector and Radiated Power

If the far field request is set to request 2 or more points for both the theta and phi directions, then the Poynting vector is integrated over the specified sector (see the detailed discussion for the FF card). This integration provides the radiated power and is given below the field values.

When analyzing an antenna the source power (calculated from the input impedance) should equal the integral of the radiated power over the surface of a sphere minus any losses such as dielectric losses and finite conductivity.

> **Tip:** Use the power integration as a partial validation of the result.

It is also possible to set the far field request to only integrate the far field power without writing the field values to the output file.

The output file will produce the following output (a full 3D far field request was set to generate the below output):

```
     VALUES OF THE SCATTERED ELECTRIC FIELD STRENGTH IN THE FAR FIELD in V
                  Factor e^(-j*Re{BETA}*R)/R not considered

  Integration of the normal component of the Poynting vector in the angular
  grid DTHETA =    5.00 deg. and DPHI =    5.00 deg. (2701 sample points)
    angular range THETA        angular range PHI       radiated power
   -2.50 .. 182.50 deg.      -2.50 .. 362.50 deg.     8.19957E-03 Watt
    0.00 .. 180.00 deg.       0.00 .. 360.00 deg.     8.08720E-03 Watt
  Polarisation dependent radiated power:
       horizontal polarisation:    4.81599E-09 Watt (  0.00 %)
       vertical polarisation:      8.08719E-03 Watt (100.00 %)
       S polarisation:             4.04360E-03 Watt ( 50.00 %)
       Z polarisation:             4.04360E-03 Watt ( 50.00 %)
       left hand circular pol.:    4.04360E-03 Watt ( 50.00 %)
       right hand circular pol.:   4.04360E-03 Watt ( 50.00 %)
```

Feko gives two values for the total power:

1. The first line gives the total power assuming that each specified point is located at the center of an incremental integration area. The effective area is therefore slightly larger than the area defined by the start and end angles.
2. The second line gives the total power integrated over an area defined by the start and end angles.

For example, assuming a far field integration from $\varphi = 0°$ to $\varphi = 350°$ and $\vartheta = 5°$ and $\vartheta = 175°$ both in $10°$ increments then the first total will give the total power through the sphere. It is also possible to set

the request from $\varphi = 0°$ to $\varphi = 360°$ and $\vartheta = 0°$ to $\vartheta = 180°$ in which case the second total will give the correct power through the sphere.

The polarisation dependent power displayed in the second block of data is calculated according to the effective area of the second line.

Receiving Antenna

When using a receiving antenna, the received power and phase of the received signal is given as follows:

```
Receiving antenna (far field pattern) with name: FarFieldReceivingAntenna1

               RECEIVED POWER FOR IDEAL RECEIVING ANTENNA (FAR FIELD PATTERN)

         Received power (ideal match assumed):    2.6019E-03 W

         Relative phase of received signal:      -8.6549E+01 deg.
```

8.7 S-parameters

S-parameters data consist of the S-parameters for all active ports as well as a table of reference impedances used at each port.

S-parameters are requested with the SP card or with an S-parameter configuration in CADFEKO. Two tables of data are printed to the output file. The first lists the impedance at each port.

```
                    S-PARAMETER REFERENCE IMPEDANCES AT PORTS

                           port     impedance in Ohm
                              1        5.00000E+01
                              2        1.00000E+02
```

Further into the output file the S-parameters are listed for each source as shown below. Note that inactive ports are only used as sink ports, that is, they are not excited and no data block is created for these.

> **Note:** Ports are set to inactive in CADFEKO in the **Request S-parameters** dialog and in EDITFEKO the source (such as the A1 or AE card) must be set to zero amplitude.

All the ports are loaded and Feko therefore also writes the loading information to the output file. For example, for an S-parameter request using edge ports, the following data will be written to the output file:

```
                     DATA FOR EDGE LOADS

Name:
Load index:                    1
Load type:                     Resistor
Complex impedance:             ( 5.00000E+01 +j 0.00000E+00) Ohm (freq. dep.)
Attached to port:              Edge port
Port between triangle labels:
  Union1.Face53_1  Union1.Face53_2
Port edge length in m:         4.60000E-03
Number of edges:               2
Indices of the edges:          809   814
```

For the S-parameter data, the second data line below gives S21 or the coupling to port 2 when port 1 is excited.

```
                            SCATTERING PARAMETERS

      ports                                          magnitude          phase
   sink source     real part   imag. part      linear      in dB     in deg.
S     1      1   8.20232E-01 -3.08302E-01   8.76260E-01   -1.1473    -20.60
S     2      1  -1.09955E-02 -5.19575E-03   1.21613E-02  -38.3004   -154.71
        Sum |S|^2 of these S-parameters:    7.67979E-01   -1.1465
```

8.8 Computation Time and Peak Memory

Computation time and memory data consist of the computation time for the different stages of the solution such as checking the geometry and matrix calculation time. The peak memory and memory per process is also provided.

The final section in the output file gives an overview of the computation time, in seconds, in a tabular format:

```
                  SUMMARY OF REQUIRED TIMES IN SECONDS

                                            CPU-time       runtime
Reading and constructing the geometry          0.184         0.184
Checking the geometry                          0.095         0.095
Initialisation of the Green's function         0.000         0.000
Calcul. of coupling for PO/Fock                0.000         0.000
Transformation to equivalent sources           0.000         0.000
Ray launching/tracing phase of RL-GO           0.000         0.000
Calcul. of matrix elements                    18.036        18.037
Calcul. of right-hand side vector              0.001         0.000
Preconditioning system of linear eqns.         0.437         0.439
Solution of the system of linear eqns.         3.367         3.365
Eigensolution for characteristic modes         0.000         0.000
Determination of surface currents              0.000         0.000
Calcul. of impedances/powers/losses            0.045         0.045
Calcul. of averaged SAR values                 0.000         0.000
Calcul. of power receiving antenna             0.000         0.000
Calcul. of cable coupling                      0.000         0.000
Calcul. of error estimates                     0.000         0.000
Calcul. of electric near field                 0.000         0.000
Calcul. of magnetic near field                 0.000         0.000
Calcul. of far field                           0.000         0.000
                           other               0.127         0.128
                                           ------------  ------------
                      total times:            22.292        22.293
             (total times in hours:            0.006         0.006)
```

This table is followed by an output of the peak memory usage (main memory, excluding possible out-of-core files) which Feko encountered during any solution phase:

```
Specified CPU-times are referring to the master process only
 Sum of the CPU-times of all processes:        89.173 seconds (     0.025 hours)
 On average per process:                       22.293 seconds (     0.006 hours)

 Peak memory usage during the whole solution: 60.218 MByte
 (refers to the master process only)
 Sum of the peak memory of all processes:     233.845 MByte
 On average per process:                      58.461 MByte
```

Feko Application Macros

9

A large collection of application macros are available for CADFEKO and POSTFEKO.

This chapter covers the following:

9.1 Application Macros

An application macro is a reference to an automation script, an icon file and associated metadata.

Application macros are available directly or can be added, removed, modified or executed from the application macro library.

> **Tip:** A large collection of application macros are available in CADFEKO and POSTFEKO.

On the **Home** tab, in the **Scripting** group, click the **Application macro** icon.

9.2 Application Macro Library

The application macro library allows commonly used application macros to be stored in a repository.

The application macro library are stored at the following locations:

- Feko home directory for global access: `<FEKO_SHARED_HOME>\installedapplicationmacrolibrary`
- Feko user directory for local access: `<FEKO_USER_HOME>\applicationmacrolibrary`

Note:

- User defined application macros are stored and managed in the <FEKO_USER_HOME>.
- Only application macros stored locally in <FEKO_USER_HOME> may be modified or removed.

Related reference

Environment Variables: FEKO_SHARED_HOME, FEKO_USER_HOME

9.2.1 Adding a Macro to the Application Macro Library

Extend the application macro library by adding an application macro.

1. On the **Home** tab, in the **Scripting** group, click the **Application macro** icon. From the drop-down list, select the **Macro Library** icon.

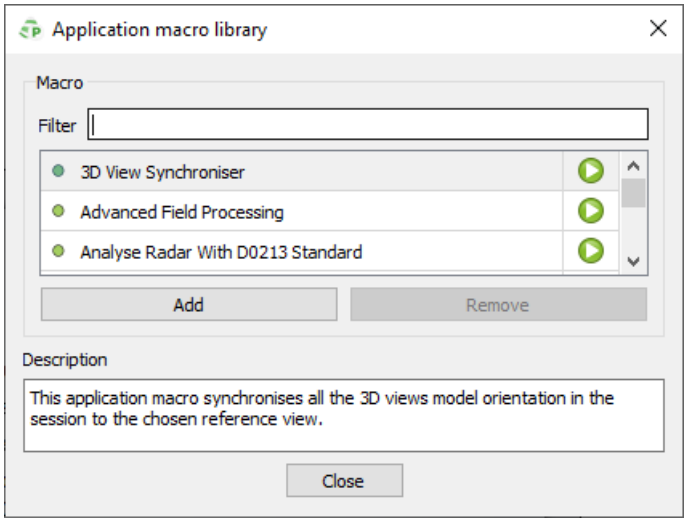


*Figure 593: The **Application macro library** dialog.*

2. On the **Application macro library** dialog, click **Add**.
3. In the **Script location** field, browse to the location of the application macro that you want to add to the library.
4. Under **Description**, add a comment to describe the purpose of the macro.

5. In the **Label** field, specify the macro name.
6. From the **Icon** drop-down list select one of the following:
   - Select a standard icon.
   - Browse to the location of a custom image.

   Note:
   - The image may be any size as it is scaled
   - Multiple image file formats are supported.
7. Click **Create** to add the application macro to the library and to close the dialog.

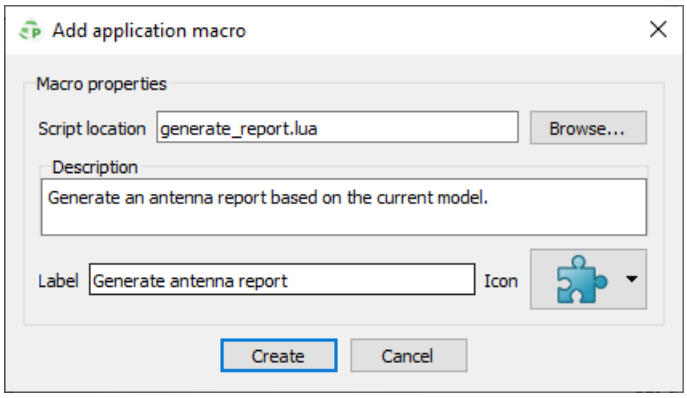


*Figure 594: The **Add application macro** dialog.*

9.2.2 Running a Macro from the Application Macro Library

Run a script that is located in the application macro library.

1. On the **Home** tab, in the **Scripting** group, click the **Application macro** icon. From the drop-down list, select the **Macro Library** icon.
2. Select the application macro that you want to run by using one of the following workflows:
   - In the **Filter** field, enter the macro name to narrow down the search.
   - In the table select the relevant macro.
3. Run the script by selecting one of the following workflows:
   - Click the button.
   - From the right-click context menu, click **Run**.

9.3 CADFEKO Application Macros

A collection of Lua application macros are available to automate repetitive tasks in CADFEKO.

9.3.1 Transfer User Configurations

This application macro transfers settings and application macros between different versions of Feko. This may be useful if you have customised one installation of Feko and want to use the same settings in a concurrent installation.

Using the Application Macro

Execute the application macro in CADFEKO to transfer user-defined settings to another Feko version.

1. Open CADFEKO and run the macro.
   The **Transfer user configurations** dialog is shown.

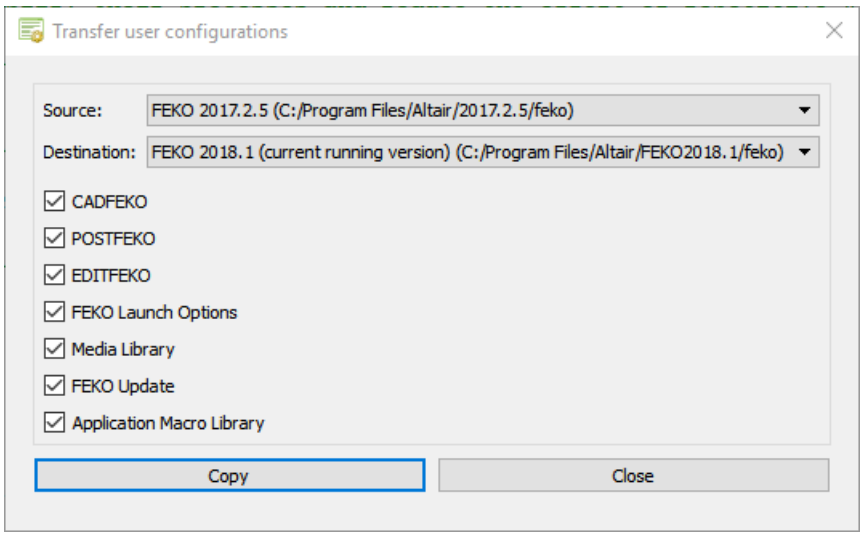


*Figure 595: The **Transfer user configurations** dialog.*

2. From the **Source** list, select the version to copy settings from.
3. From the **Destination** list, select the version to copy settings to.

> **Note:** Only Feko versions installed in the same directory as the current running version, for example, `C:/Program Files/Altair`, are displayed in the **Source** and **Destination** lists.

4. [Optional] Clear the check boxes for the settings that are not required for transfer.

> **Note:** If **Application Macro Library** is selected, the application macro locations are transferred. (The application macro files are not copied to a new location.)

5. Click **Copy**.
6. Finish the process by clicking **OK** to acknowledge the messages.

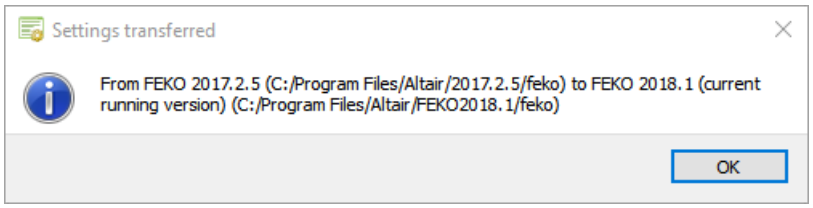


*Figure 596: The **Settings transferred** dialog.*

- If the **Destination** is the current running version, a new CADFEKO session is started with the transferred settings.

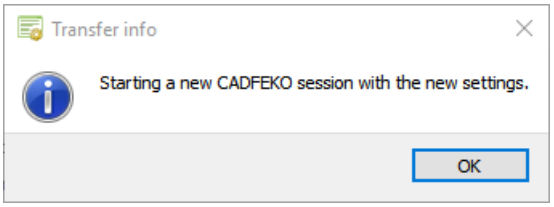


*Figure 597: The **Transfer info** dialog.*

After the new session is started, click **OK** to close the old CADFEKO session.

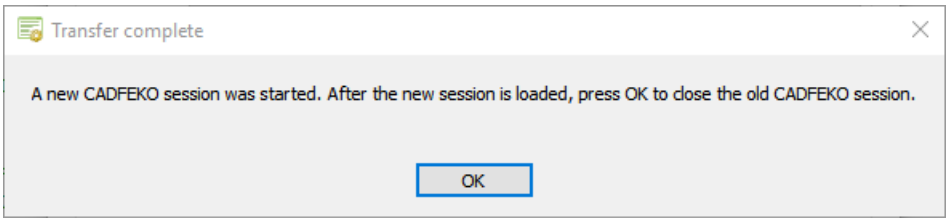


*Figure 598: The **Transfer complete** dialog.*

- If the **Destination** is not the current running version, click **Close** to dismiss the **Transfer user configurations** dialog.

The new settings are now available.

9.3.2 Generate Antenna Array

The application macro allows you to create an array of elements. Specify the number of elements and offset between the elements, or import the coordinates from file to create an irregular-spaced array. For the array amplitude distribution, select a mathematical distribution (such as cosine) or import the magnitude and phase for each element from file.

The array tool application macro has additional functionality when comparing to the *Finite array* tool[90] in CADFEKO that allows you to create custom and complex antenna arrays with ease. Create a `.cfx` file that contains the base element (antenna). Then use the application macro to copy the base elements (including all its ports, sources and loads) to create the array. Each source and load are given a unique name, allowing you to modify any source or load for an array element after the application macro was run. You can also specify the configuration to which the sources and loads should be copied.

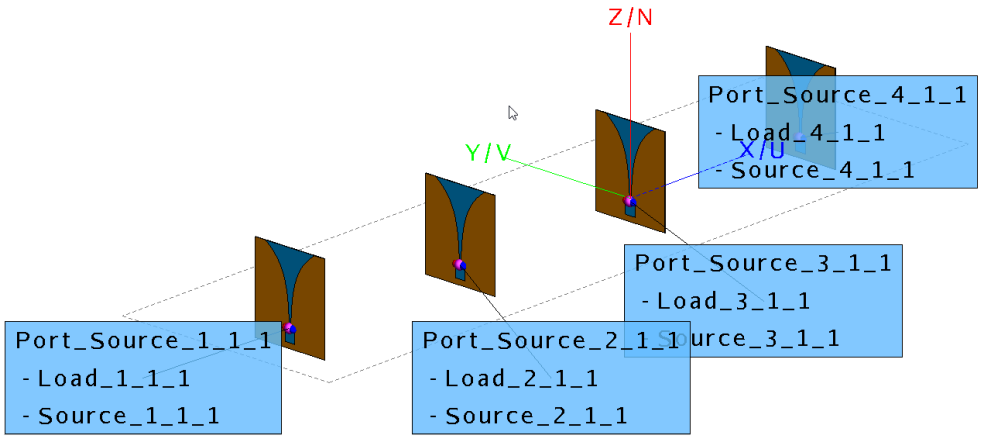


Figure 599: The antenna array created by copying the base element. Each element has a unique label for the port, source and load.

Tip: Find the examples in the <FEKO_SHARED_HOME> directory:

`<FEKO_SHARED_HOME>/installedapplicationmacrolibrary/CADFEKO/ArrayTool/examples`.

90. The *Finite antenna array* tool in CADFEKO copies the base element, but the same sources and loads as used for the base element, are used for all antenna array elements. For the array amplitude distributions, you can either specify the uniform amplitude distribution or specify the magnitude and phase per element.

Defining a Linear or Planar Antenna Array

Use the application macro to create a linear or planar antenna array from a specified base element.

1. Open `vivaldi_base_element.cfx` or any other `.cfx` file in CADFEKO.

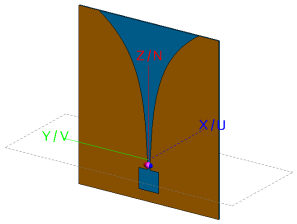


Figure 600: The base element in `vivaldi_base_element.cfx`.

2. Run the **Generate antenna array** application macro from the application macro library.
   The **Generate array** dialog is displayed.

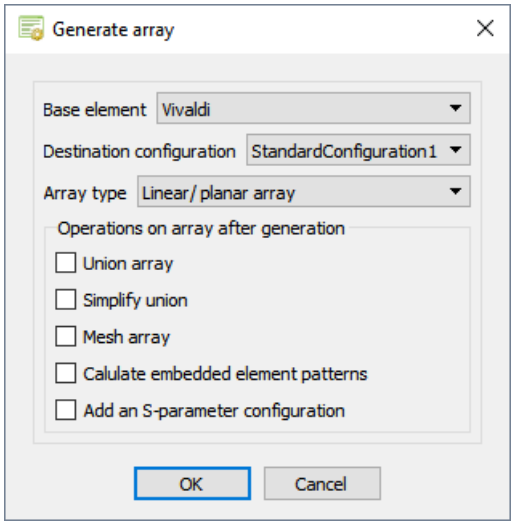


*Figure 601: The **Generate array** dialog.*

3. From the **Base element** drop-down list, select the geometry part[91] or model mesh part that is the antenna.
4. From the **Destination configuration** drop-down list, select the configuration where to the duplicated sources and loads are added.
5. From the **Array type** drop-down list, select **Linear/planar array**.
6. Under **Operations on array after generation**, select any of the following options:
   a) [Optional] Select the **Union array** check box to apply the union operation to the array automatically after the array is created.
   b) [Optional] Select the **Simplify union** check box to apply the simplify operation to the array automatically after the antenna array is created.

91. The highest-level items in the model tree are referred to as “parts”.

c) [Optional] Select the **Mesh array** check box to mesh the array automatically after the antenna array is created.

d) [Optional] Select the **Calculate embedded element patterns** check box to create *N* configurations (where *N* is the number of antenna elements in the array) with all requests duplicated for each configuration. Each configuration has an active port while the other ports are terminated with 50 ohm load.

e) [Optional] Select the **Add an S-parameter configuration** check box to add an S-parameter configuration automatically with all ports included and set active for the array.

7. Click **OK** to close the **Generate array** dialog.
The **Array layout** dialog is displayed.

Specifying the Array Layout

Specify the number of elements and offset between the elements.

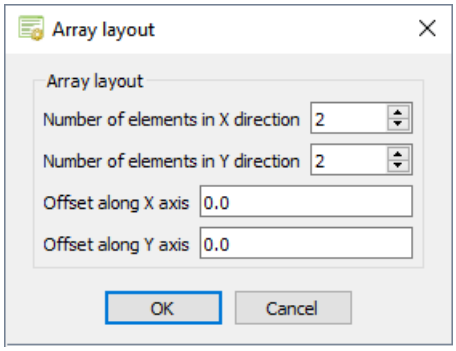


*Figure 602: The **Array layout** dialog.*

1. In the **Number of elements in X direction** field, enter a value for the number of elements.
2. In the **Number of elements in Y direction** field, enter a value for the number of elements.
3. In the **Offset along X axis** field, enter a value for the offset between the elements.
4. In the **Offset along Y axis** field, enter a value for the offset between the elements.
5. Click **OK** to close the **Array layout** dialog.
The **Import excitation values** dialog is displayed.

Specifying the Amplitude Distribution

Define the amplitude distribution for the antenna array.

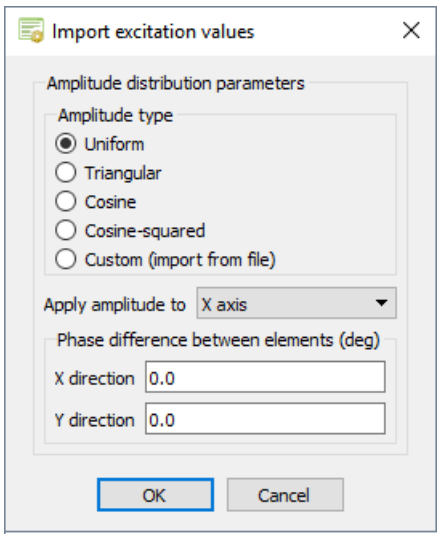


*Figure 603: The **Import excitation values** dialog.*

1. Under **Amplitude type**, select one of the following:
   - **Uniform**
   - **Triangular**
   - **Cosine**
   - **Cosine-squared**
   - **Custom (import from file)**

   > **Note:** If you have selected **Custom (import from file)**, continue to Importing Array Parameters From File.

2. From the **Apply amplitude type to** drop-down list, select one of the following:
   - **X axis**
   - **Y axis**
   - **X and Y axes**

   > **Note:** If only a single axis is specified[92], a uniform distribution is applied to the second axis.

3. [Optional] In the **X direction** field, specify the phase difference in degrees between the elements.
4. [Optional] In the **Y direction** field, specify the phase difference in degrees between the elements.
5. Click **OK** to close the dialog.

92. For example, if a cosine distribution is applied to the X axis, a uniform amplitude is applied to the Y axis.

The application macro creates an array as specified.

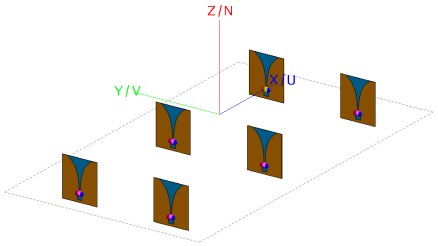


Figure 604: An example of a planar antenna array created from a single base element.

Defining an Irregular-Spaced Array

Use the application macro to create an irregular-spaced antenna array from a specified base element.

1. Open `vivaldi_base_element.cfx` or any other `.cfx` file in CADFEKO.

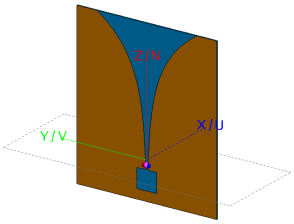


Figure 605: The base element in `vivaldi_base_element.cfx`.

2. Run the **Generate antenna array** application macro from the application macro library.
The **Generate array** dialog is displayed.

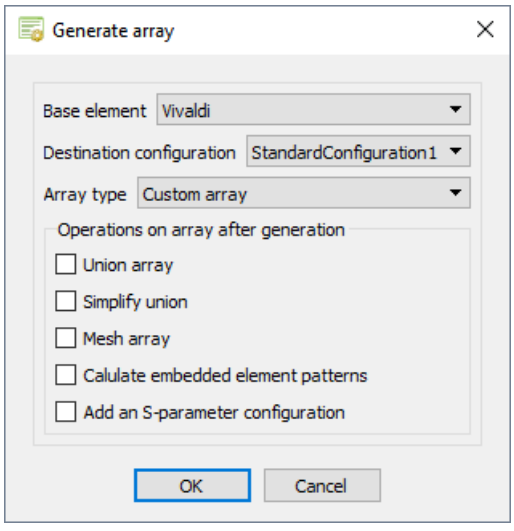


*Figure 606: The **Generate array** dialog.*

3. From the **Base element** drop-down list, select the geometry part[93] or model mesh part that is the antenna.
4. From the **Destination configuration** drop-down list, select the configuration where to the duplicated sources and loads are added.
5. From the **Array type** drop-down list, select **Custom array**.
6. Under **Operations on array after generation**, select any of the following options:
   a) [Optional] Select the **Union array** check box to apply the union operation to the array automatically after the array is created.
   b) [Optional] Select the **Simplify union** check box to apply the simplify operation to the array automatically after the antenna array is created.
   c) [Optional] Select the **Mesh array** check box to mesh the array automatically after the antenna array is created.
   d) [Optional] Select the **Calculate embedded element patterns** check box to create *N* configurations (where *N* is the number of antenna elements in the array) with all requests duplicated for each configuration. Each configuration has an active port while the other ports are terminated with 50 ohm load.
   e) [Optional] Select the **Add an S-parameter configuration** check box to add an S-parameter configuration automatically with all ports included and set active for the array.
7. Click **OK** to close the **Generate array** dialog.
   The **Import array parameters** dialog is displayed.

Importing Array Parameters From File

Specify a file to import an array with a user-defined distribution, or to import an irregular-spaced array or both.

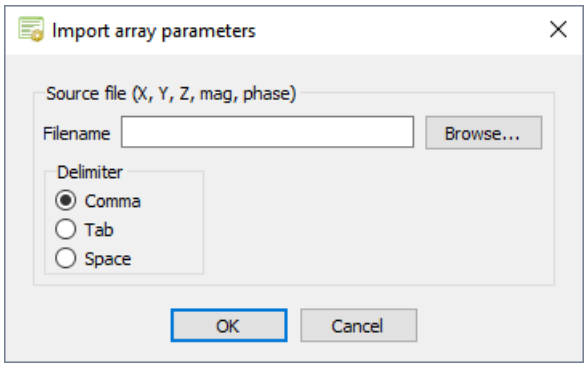


*Figure 607: The **Import array parameters** dialog.*

1. In the **Filename** field, browse for the file you want to import. The number of antenna array elements is determined from the specified file.

93. The highest-level items in the model tree are referred to as "parts".

Note:

- If you have selected **Custom array** in **Step 5**, the file that you import must contain the X, Y and Z coordinates, amplitude, and phase (in degrees).

  For example (if you use comma-separated values):

  ```
  20, 30, 40, 1, 0
  25, 30, 40, 2, 20
  28, 30, 40, 5, 30
  ```

- If you have selected **Custom (import from file)** in **Step 1**, the file that you import must contain the magnitude, and phase (in degrees), where the base element is the last element in the file.

  For example (if you use comma-separated values):

  ```
  1, 0
  2, 20
  5, 30
  ```

  Figure 608 illustrates the element order when importing the magnitude and phase from a file.

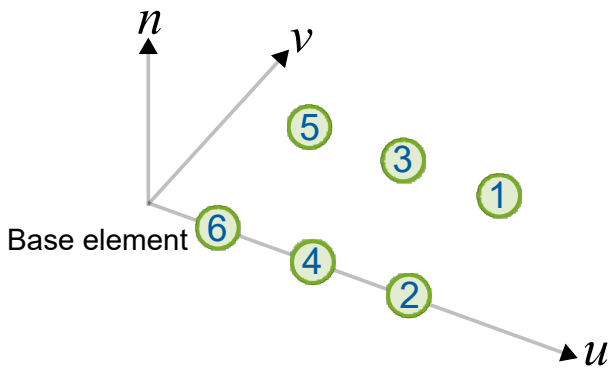


Figure 608: An example showing the element order for a 3x2 planar antenna array where the magnitude and phase are imported from a file. The base element is the last element specified in the file.

2. Under **Delimiter**, select the delimiter type used in the file to be imported from the following list:
   - **Comma**
   - **Tab**
   - **Space**
3. Click **OK** to close the dialog.

The application macro creates an array as specified.

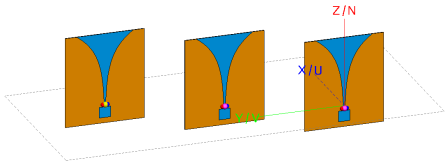


Figure 609: An example of a custom antenna array created from a single base element.

Limitations of Generate Antenna Array Macro

The **Generate antenna array** application macro has several limitations on how the base element is defined and the amplitude distribution.

The following limitations should be noted:

- The base element consists only of the antenna. Geometry that does not form part of the array can be added after generating the array.
- The base element must be either a geometry part or model mesh part (the highest-level item in the model tree).
- When a mathematical distribution is only applied to a single axis, a uniform distribution is applied to the second axis.
- When defining an irregular-spaced array by importing from file, the number of elements is determined by the imported file.

9.3.3 Compare CADFEKO Models

The application macro allows you to compare two CADFEKO models to find the differences between the two models.

The results of the comparison are displayed on the dialog or can be exported to file.

Using the Application Macro

Use the **Compare CADFEKO models** application macro to compare two CADFEKO models.

> **Restriction:** Keep the order of collections the same between models as indexes are used to compare objects.

1. Run the **Compare CADFEKO models** from the application macro library in CADFEKO
   The **Compare CADFEKO Models** dialog is displayed.

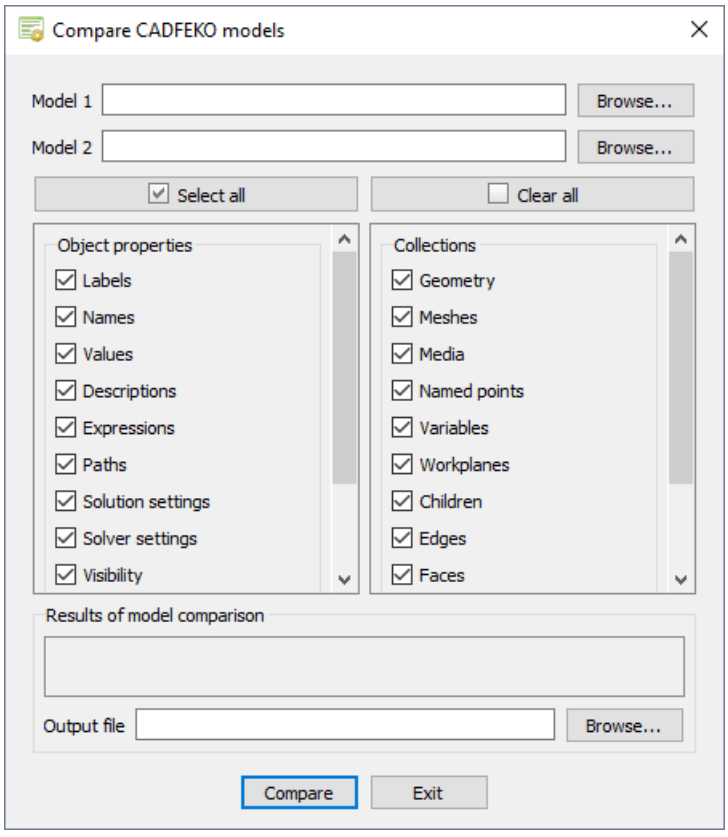


*Figure 610: The **Compare CADFEKO Models** dialog.*

2. In the **Model 1** field, browse to the first CADFEKO model.
3. In the **Model 2** field, browse to the second CADFEKO model.

4. Under **Object properties**, select the object[94] properties to compare:

- *Labels*
  Compare the labels of objects.
- *Names*
  Compare the names of objects (variables and objects).
- *Values*
  Compare the values of objects (variables).
- *Descriptions*
  Compare the description of objects (variables).
- *Expressions*
  Compare the expressions of objects (variables).
- *Paths*
  Compare file paths of objects.
- *Solution settings*
  Compare the model solution settings of objects.
- *Solver settings*
  Compare the solution solver settings of objects.
- *Visibility*
  Compare the visibility of objects.
- *Locked*
  Compare the locked[95] state of objects.
- *Included*
  Compare the included[96] state of objects.
- *Ground plane*
  Compare ground planes (objects).

94. An object is an entity within an object oriented programming language with two main characteristics: a state and a behaviour. The settings of the object are stored in its properties and its abilities are accessed through methods.
95. Lock a part to prevent modification to the simulation mesh (and prevent the part from being edited).
96. A geometry part (or mesh part) that does not contain any ports, sources or loads can be temporarily excluded from the model without having to delete the part.

5. Under **Collections**, select the collections[97] to compare:
   - *Geometry*
     Compare the collections of geometry.
   - *Meshes*
     Compare the collections of editable meshes.
   - *Media*
     Compare the collections of media.
   - *Named points*
     Compare the collections of named points in the model.
   - *Variables*
     Compare the collections of variables in the model.
   - *Workplanes*
     Compare the collections of workplanes in the model.
   - *Children*
     Compare the operator's collection of child operators.
   - *Edges*
     Compare the operator's collection of edges.
   - *Faces*
     Compare the operator's collection of faces.
   - *Regions*
     Compare the operator's collection of regions.
   - *Transforms*
     Compare the operator's collection of transforms.
   - *Wires*
     Compare the operator's collection of wires.
6. Click **Compare** to evaluate the models and start the comparison between model 1 and model 2. Under **Results of model comparison**, the results of the comparison are displayed.
7. [Optional] In the **Output file** field, specify the text file to export the results for the model comparison.

97. A `collection` is a special object that contains objects of which there can be more than one. For example, there can be multiple sources, far fields, geometry parts and so on. When referencing an item in a collection, an index must always be specified, for example `farfield[1]` or `farfield["FarField"]`.

9.3.4 Create Edge Port on Finite Substrate Macro

The **Create Edge Port** application macro creates an edge port on a finite substrate.

This application macro uses three defined named points in the model to define where the edge port will be located. It removes a small cuboidal section[98] and creates a PEC "bow-tie" structure. An edge port is then added to the "centre" edge of the "bow-tie" structure. The option is available to add a voltage source to the edge port.

Related information

How to feed a microstrip annular-ring slot antenna on a finite ground using an edge-feed

Using the Application Macro

Run the **Create Edge Port** application macro to create an edge port on a finite substrate. It creates a "bow-tie" structure given the specified parameters and adds an edge port.

1. Open a CADFEKO model containing a finite substrate.
2. Run the **Create Edge Port** application macro from the application macro library in CADFEKO.

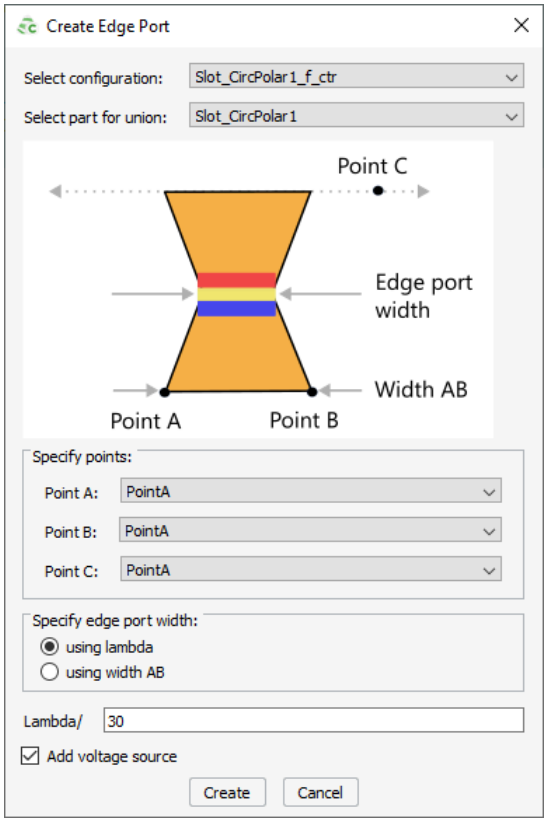


*Figure 611: The **Create Edge Port** dialog.*

98. A port cannot be placed on the boundary of a finite dielectric substrate when using the method of moments (MoM).

3. In the **Select configuration** drop-down list, select a configuration. The frequency of the configuration is used to determine the edge length of the edge port.
4. In the **Select part for union** drop-down list, select the part that will be unioned with the "bow-tie" structure.
5. In the **Specify points** group, specify the named points to define the location of the edge port.

   Points A and B is used to determine the maximum width of the "bow-tie" structure.

   Point C determines the "height" of the "bow-tie" structure.
6. In the **Specify edge port width** group, select one of the following options:
   - **using lambda**
     - The edge port width is determined using lambda. By default, a value of $\frac{\lambda}{30}$ is used, but the value can be modified.
   - **using width AB**
     - The edge port width is determined using the physical width AB. By default, a value of (width AB)/7 is used, but the value can be modified.
7. Select the **Add voltage source** check box to add a voltage source after the port is created.
8. Click **Create** to run the script and to create the edge port.

9.3.5 Create Coaxial Feed Macro

The **Create Coaxial Connector Feed** application macro creates a coaxial connector on a finite substrate.

This application macro uses three defined named points in the model to define where the coaxial connector will be located. The coaxial connector is defined using an outer diameter, inner diameter, coaxial length and insulating dielectric.

Using the Application Macro

Run the **Create Coaxial Connector Feed** application macro to create a coaxial connector on a finite substrate.

1. Open a CADFEKO model where you would like to add a coaxial connector.
2. Run the **Create Coaxial Connector Feed** application macro from the application macro library in CADFEKO.

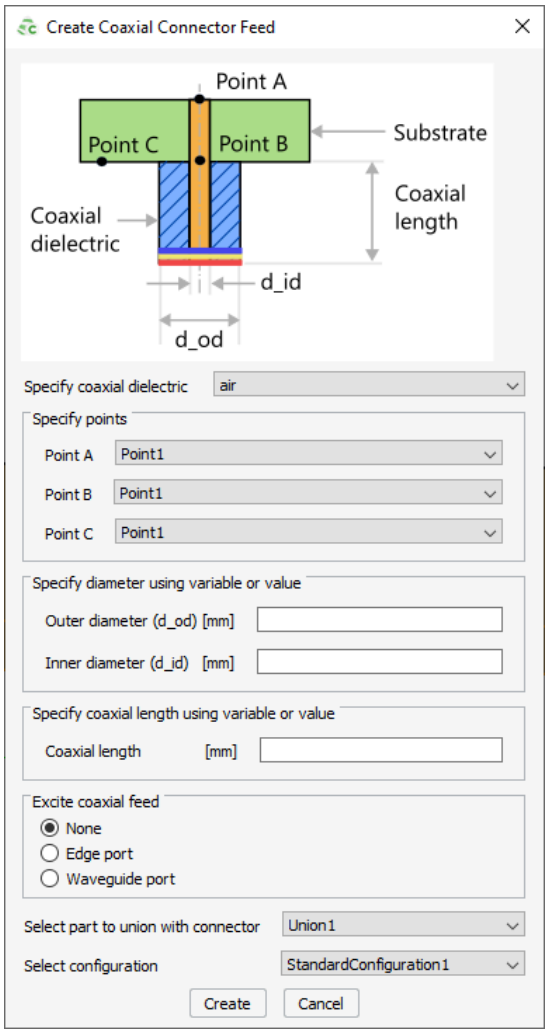


*Figure 612: The **Create Coaxial Connector Feed** dialog.*

3. In the **Specify coaxial dielectric** drop-down list, select the insulating connector dielectric.
4. In the **Specify points** group, specify the named points to define the location of the coaxial connector.

   Note: Point C can be any point on the surface of the substrate (it is not required to be on the edge of the substrate).

5. In the **Specify diameter using variable or value** group, specify the following:
   a) In the **Outer diameter (d_od)** field, specify either a defined variable or a value for the outer diameter of the connector.
   b) In the **Inner diameter (d_id)** field, specify either a defined variable or a value for the inner diameter of the connector.

      Note: The unit for the diameter (outer and inner) and the length is taken as the model unit and indicated in brackets next to the relevant input fields.

6. In the **Specify coaxial length using variable or value** group, specify the coaxial length of the connector.
7. In the **Excite coaxial feed** group, select one of the following options:
   - **None**: No port or excitation is added to the coaxial connector.
   - **Edge port**: The coaxial connector is fed using an edge port with a voltage source. The construction takes into account the limitation that a port cannot be placed on the boundary of a finite dielectric substrate when using the method of moments (MoM).
   - **Waveguide port**: The coaxial connector is fed using a waveguide port with a waveguide source.
8. In the **Select part to union with connector** drop-down list, select the model part that will be unioned with the coaxial connector.
9. In the **Select configuration** drop-down list, select a configuration. The specified configuration determines if a global or local source should be added. The frequency of the configuration is also used to determine the edge mesh refinement.
10. Click **Create** to run the script and to create the coaxial connector.

9.3.6 Other CADFEKO Application Macros

A collection of smaller CADFEKO application macros are available, but these macros do not include step-by-step instructions.

Pre-Process PollEx REI File

This application macro imports a `.rei` file from Altair PollEx and creates the associated geometry in CADFEKO.

Automatic Mesh Refinement

This application macro performs an automatic mesh refinement process based on error estimates using the **Adaptive refinement** tool available in CADFEKO in an itterative fashion. The progress and feedback is shown in the CADFEKO message window.

During the mesh refinement process, request, configurations, frequency and other settings may be adjusted.

Once the mesh refinement completes, a full simulation (including all configurations. requests, frequencies and setting in the original model) will be run. This simulated model named `<filename>_refined` (where <filename> is the name of the original model) will remain open in CADFEKO at the same folder location as the original model. The original model will remain unmodified.

The **Meshing rules** added during automatic mesh refinement can be viewed in the model tree.

Variables are added in the refined model that may be used to adjust the automatic refinement thresholds and refine the meshing further if needed.

Automated mesh refinement may also be launched from a Feko terminal using the following command syntax:

```
cadfeko --non-interactive '<filepath>' --run-script %FEKO_SHARED_HOME%
\installedapplicationmacrolibrary\CADFEKO\AutoAdaptiveMeshing\AutoAdaptiveMeshing.lua
 | more
```

Create Frequency Ranges

This application macro creates a separate model for each frequency range specified. The following solvers MoM, MLFMM, and ACA, can be specified for each range to improve simulation time.

> **Note:** Run the **Combine Results** application macro in POSTFEKO to combine the result for each frequency range.

Create Rough Sea Surface

This application macro creates a CADFEKO model of a rough sea surface.

Create Inductive Charging Coils

This application macro creates a CADFEKO model of two inductive charging coils.

Ideal Power Divider

This application macro generates a network model of an ideal n-port power divider with unequal division, a 2-port Wilkinson power divider with unequal division, and an n-port Wilkinson power divider with equal division.

Create Far Field Equivalent Sources Split Over Frequency

This application macro creates a far field equivalent source for each frequency split over multiple configurations. The application macro also adds a receiving antenna request with each configuration.

> **Note:** Run the Combine Far Field Equivalent Sources application macro in POSTFEKO to combine the result for each configuration.

Optenni Lab: Port Matching

This application macro uses Optenni Lab to generate matching networks for all desired ports.

Parameter Sweep: Create Models

This application macro generates different permutations of a parametric model based on varying the value of the model variables.

Create Impedance Sheet for Layered Metals

This application macro creates an effective surface impedance from a stacked metal definition. The application macro requires a metal to be defined in the model.

Create Wireless Communication Measurement Configuration

This application macro creates a standard configuration with a far field request with a pre-defined frequency and angular definition. The application macro can also suggest the required increment to use for the far field request given the frequency range and largest dimension of the device under test.

> **Note:** Use the **Calculate Wireless Communication Performance** application macro to calculate EIRP,EIS,TRP and TIS quantities with the far field data from this macro.

9.4 POSTFEKO Application Macros

A collection of Lua application macros are available to automate repetitive tasks in POSTFEKO.

9.4.1 Characteristic Mode Plotter

This POSTFEKO application macro can be used to plot all the standard parameters that are available after a characteristic mode analysis simulation was performed.

Characteristic Mode Analysis

Characteristic mode analysis (CMA) is the numerical calculation of a weighted set of orthogonal current modes that are supported on a conducting surface. The sets of characteristic near fields and far fields associated with these characteristic currents can provide insight into the radiating properties of structures, allowing for a systematic approach to antenna design and placement.

Characteristic modes are obtained by solving a particular weighted eigenvalue equation that is derived from the method of moments impedance matrix. Feko has a built-in solver that calculates these modes, with no need for post-processing by the user. The eigen values, modal significance, characteristic angles, currents, near fields, and far fields can be visualised in POSTFEKO.

Characteristic Mode Plotter

Characteristic mode analysis calculates various parameters of interest. The characteristic mode plotter was developed to plot these parameters, since it can be a tedious task to do this manually for multiple modes.

Select the data to analyse, and which quantities to plot: eigen value, modal significance, characteristic angle and, if available, modal excitation and weighting coefficients. Each selected quantity is plotted on a new graph.

Either frequency or mode index have to be selected to define the independent axis. If both are selected, two graphs will be created for each quantity. The number of modes to be plotted is specified. By default, both tracked and untracked modes are plotted on each graph.

Example for Using the Characteristic Mode Plotter

The characteristic mode analysis plotter application macro is used to plot the various CMA parameters for a simple dipole antenna.

The example model is a half-wavelength wire dipole at 74.9 MHz. The dipole length is 2 m, and it has a wire radius of 2 mm. It is excited by a voltage source at the centre of the wire.

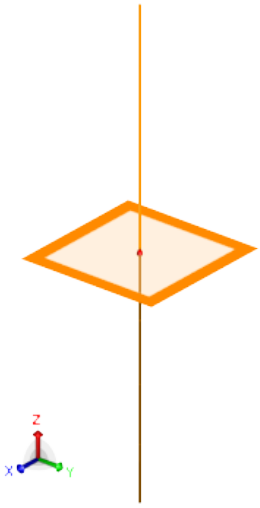


Figure 613: Simple dipole model.

Tip: Find the example in the <FEKO_SHARED_HOME> directory:
`<FEKO_SHARED_HOME>/installedapplicationmacrolibrary/POSTFEKO/CharacteristicModeAnalysis/CMAPlotter/examples`.

Using the Application Macro

Execute the application macro in POSTFEKO to plot characteristic mode quantities on Cartesian graphs.

1. Start with a POSTFEKO session containing at least one model with characteristic mode analysis results.

The results from a single characteristic mode analysis request will be used as input to the macro.

2. Execute the **Plot characteristic modes** application macro in POSTFEKO to plot the characteristic modes.

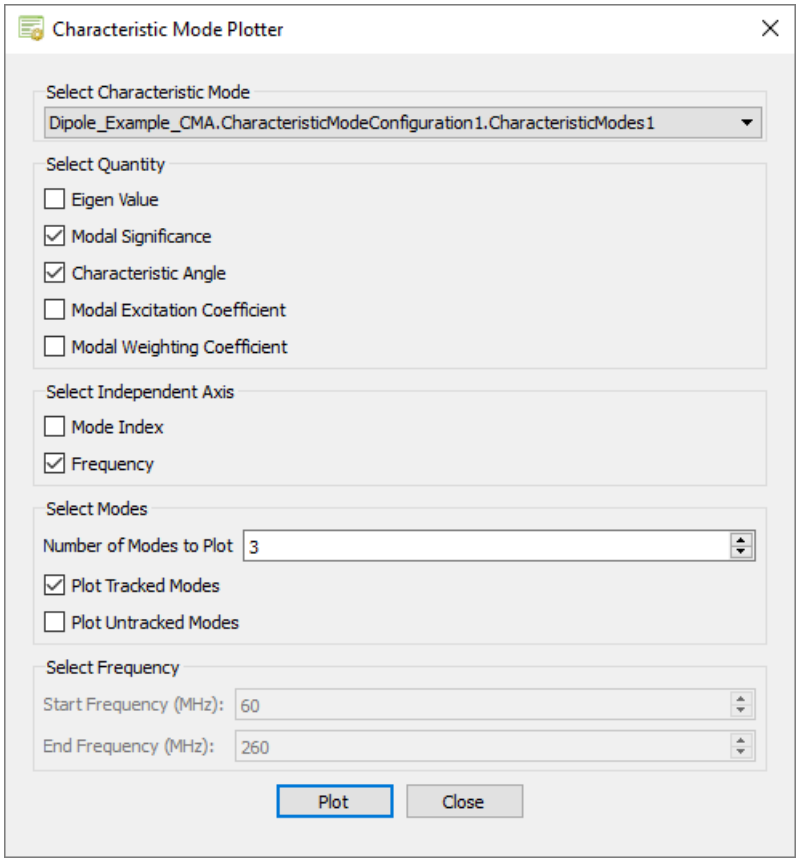


*Figure 614: The **Characteristic Mode Plotter** dialog.*

A dialog shows the available characteristic mode results.

3. Select the result and quantities of interest.

> **Restriction:** The **Modal Excitation Coefficient** and **Modal Weighting Coefficient** can only be plotted when the model contains a source and the modal excitation coefficient calculation is enabled in the request.

4. Select **Frequency**, **Mode Index**, or both as the independent axis.
   A new graph for each selected independent axis will be created for each quantity of interest.
5. Enter the highest mode index to be considered. This determines the number of modes that are plotted on each graph.
   A trace is added to each graph for each calculated mode with an index lower or equal than the entered value.
6. Select to plot either the **Tracked Modes**, the **Untracked Modes**, or both.
7. Select **Plot** to start the plotting on a Cartesian graph.
8. View the graphs generated by the macro.

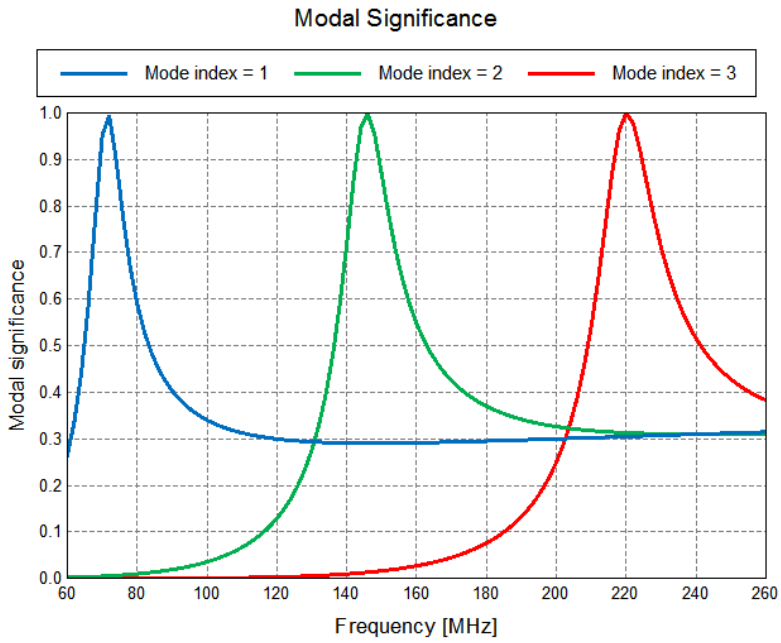


Figure 615: Modal significance graph for a simple dipole antenna.

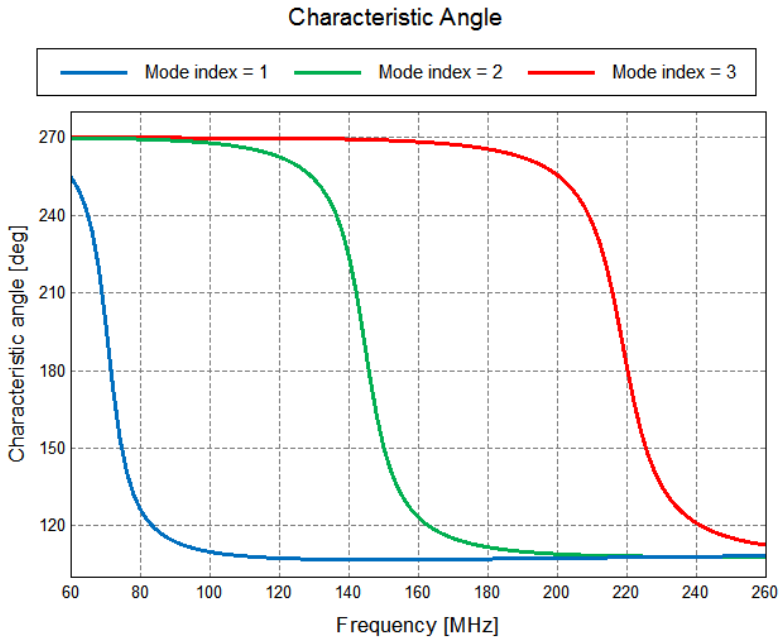


Figure 616: Characteristic angle graph for a simple dipole antenna.

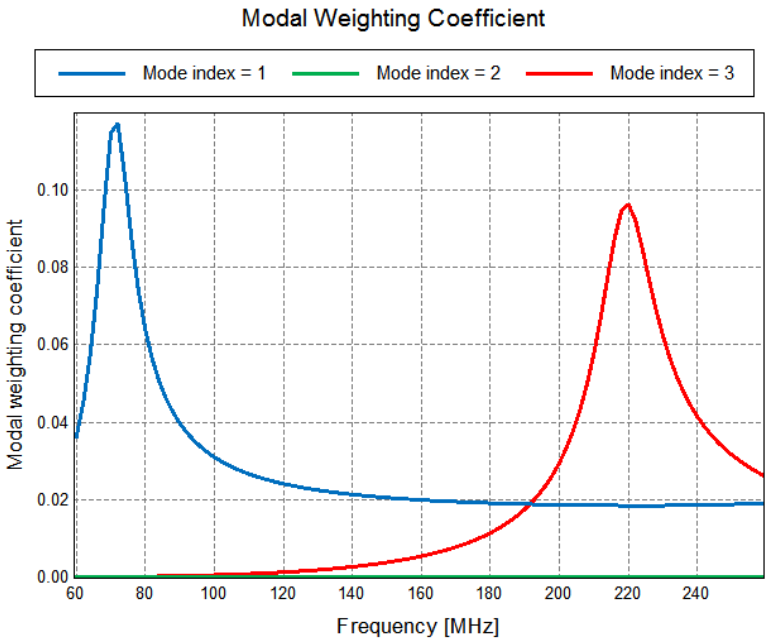


Figure 617: Modal weighting coefficient graph for a simple dipole antenna.

9.4.2 MIMO Performance Evaluation

This application macro is used for calculating mean effective gain (MEG) and envelope correlation coefficient (ECC) for a MIMO antenna configuration. The MEG ratio can also be plotted.

The application macro is used for calculating mean effective gain (MEG), envelope correlation coefficient (ECC)[99]. Using the Maths option for the 2D graph, the MEG ratio can also be plotted.

Currently, the application macro only supports two channel MIMO: the two channels are simulated as two separate configurations, each with its far field request. You can choose to sweep the cross polarization ratio (XPR), sweep the frequency or both. The propagation environment can also be defined. The default is uniform, with $P_\varphi = P_\theta = \frac{1}{4\pi}$.

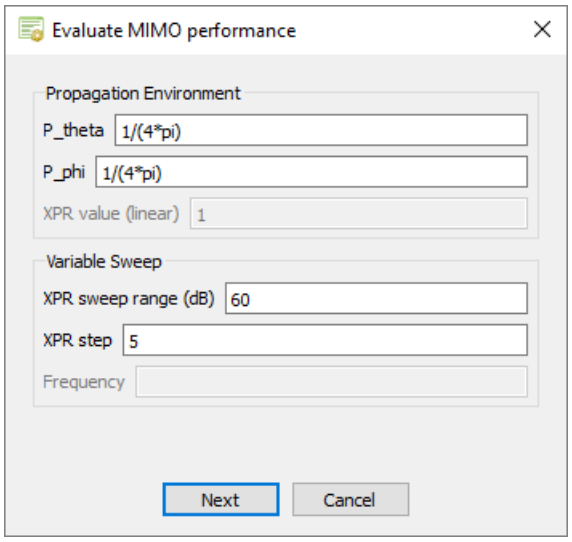


*Figure 618: The **Evaluate MIMO performance** dialog.*

Plots of ECC against XPR or MEG against frequency are plotted depending on the selected options.

99. M.P. Karaboikis, V.C. Papamichael, G.F. Tsachtsiris., C.F. Soras and V.T. Makios, Integrating Compact Printed Antennas Onto Small Diversity/MIMO Terminals, IEEE Transactions on Antennas & Propagation, Vol. 56, No. 7, July 2008.

9.4.3 Multiport Post-Processing

The *Multiport post-processing* application macro allows you to calculate results for changes in the port loading without rerunning the Solver. Results that are supported are far fields, near fields, currents and specific port parameters, for example, the voltage, current and S-parameters of each port.

Overview

The *Multiport post-processing* application macro calculates the port reflections and field values for changes in port loading, without rerunning the Solver. Through scripting in POSTFEKO, loads can be modified as a post-processing step.

Requirements for performing the post-processing for a given model, are the availability of the scattering matrix for the model and the field requests for each configuration in the scattering parameter solution. The application macro uses the extracted S-parameters and the field values for all configurations to calculate (and export) near field data, far field data (including gain), and the new S-parameter matrix (taking the loading into account) without requiring further Feko simulations.

The following application macros are provided:

Generate multiport configurations
: A CADFEKO application macro that creates the model for post-processing with the *Multiport post-processing* macro.

Multiport post-processing
: A POSTFEKO application macro that performs multiport post-processing.

Base Multiport Feko Simulation

A base multiport Feko simulation is required to generate the required input data for the *Multiport post-processing* application macro.

S-parameters are calculated in Feko by loading all ports with the port reference impedance and then, in turn, exciting each port in the model. The voltages and currents calculated at all the ports can be combined to determine the S-parameter matrix. For multiport post-processing, the field values (near and far fields) also need to be determined for each configuration in the S-parameter calculation. This requires that $m+1$ simulation solutions need to be performed by default, where m is the number of ports in the model.

Reduce the calculations to m simulations for a model consisting of m configurations:

- The loads for all configurations must be identical and set to the reference impedance for the multiport system.
- Each configuration has a single source and for all the configurations, each port is excited once.
- Requests in all configurations must be identical.
- All other configurations settings (except for the excitations) are identical.

Tip: Use **Generate multiport configurations** application macro to simplify the model creation process.

Since the loads are identical for all configurations and only the source is modified between configurations, the simulation is performed efficiently when using the method of moments (MoM) since the expensive matrix fill and LU-decomposition is only performed once at each frequency.

For all subsequent configurations, only the right-hand side vector (sources) is updated and a backward substitution is performed before calculating the output for the requests.

Multiport Post-Processing Workflow

The basic workflow of the *Multiport post-processing* application macro (`MultiportPostProcessing.lua`) is described.

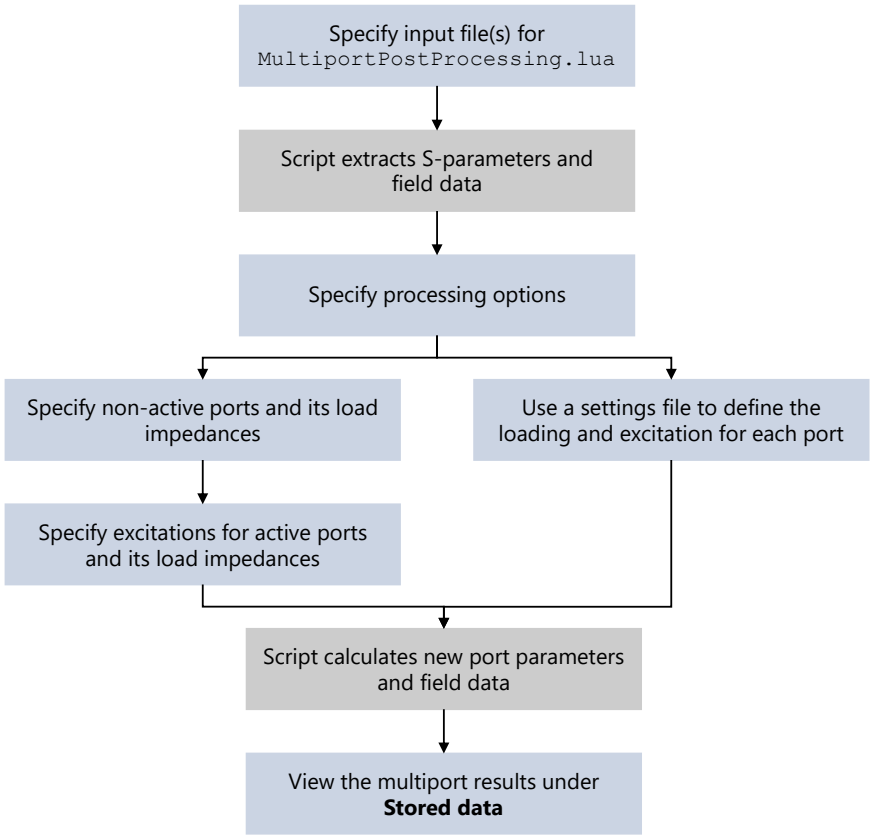


Figure 619: Post-processing workflow for the Multiport post-processing macro.

1. Specify the input file (or files) that will be used to extract the port information[100] using one of the following options:
   - Feko model (`.fek` file)
   - Measurement data
2. The application macro extracts the S-parameters and the field data (far field and near field).
3. Specify the processing options.

100. For example, voltage, currents and field data for each port.

4. Specify the non-active ports as well as what the load impedance is at these ports. The following load types are available:
   - Point loads that do not connect to other ports. These ports can be loaded with any of the following:
     - Complex loads
     - Series RLC circuit
     - Parallel RLC circuit
     - Short circuit
     - Open circuit
     - One-port Touchstone (`.s1p`) file
   - A single multiport Touchstone file that contains all the port terminations (ports can be connected through the S-parameter matrix).
   - Multiple one-port Touchstone files (one at each port). This option is equivalent to selecting the first option above (point loads) and specifying all point loads via one-port Touchstone files.
   - Specify the excitations for the active ports as well as the load impedance at these ports.

     The ports can be loaded with any of the following:
     - Direct connection
     - Complex load
     - One-port Touchstone (`.s1p`) file
     - Two-port Touchstone (`.s2p`) file

     > **Note:** Ports not specified as non-active are assumed to be sources.
5. The application macro calculates the new port parameters and field data.
6. View the multiport results in POSTFEKO, under **Stored data**.

Alternatively, you can specify the loading and excitations for each port using a settings file.

Create a Multiport Lua Settings File

The Multiport post-processing application macro can be configured to use a Lua settings file to define the loading and excitation for each port. This simplifies the procedure for frequent calculations or large multiport setups with many ports.

The *Multiport post-processing* application macro is dependent on the order in which the configurations are returned from the Lua settings file. It is expected that the configuration tables are in the following order:

1. Active ports configuration
2. Non-active ports configuration
3. Field data configuration
4. Processing options

The following command should be defined at the end of the Lua settings file.

```
return {activePortsConfiguration,nonActivePortsConfiguration,fieldDataConfiguration,processingOptions}
```

Note:

- The *fieldDataConfiguration* table is only required for the measurement method of the *Multiport post-processing* application macro to map the field data to the correct ports.

Active Port Configuration

Set up the Lua settings file for the active ports in a multiport active ports configuration. Each active port in the configuration requires an excitation and a load specification.

Port Excitation Specification

The magnitude and phase for each active ports in the active port configuration are specified as follows:

```
-- create table to store the source data in
activePortsConfiguration = {}
-- Source Port1
source = {}
source.Label = "Port1"
source.Index = 1
source.Value = pf.Complex(1,0)
table.insert(activePortsConfiguration,source)
```

Port Loading Specification

A load table that groups the loading for each port, is added to the active ports configuration. The four loading types for an active port are:

- Direct connection (no additional loading)
- Complex load
- One-port Touchstone network (`.s1p`)
- Two-port Touchstone network (`.s2p`)

```
-- Load options to attach to active port.
-- Create load table and load data table
activePortsConfiguration.Load ={}
activePortsConfiguration.Load.Data ={}
-- Type = 1 (direct connection or no loading),
load = {}
load.Type = 1
table.insert(activePortsConfiguration.Load.Data,load)
-- Type = 2 (complex)
load ={}
load.Type = 2
load.Value = pf.Complex(50,0)
table.insert(activePortsConfiguration.Load.Data,load)
-- Type = 3 (One port Touchstone network)
load = {}
load.Type = 3
-- Relative path from settings file location on drive
```

```
load.Filename = "test.s1p"
table.insert(activePortsConfiguration.Load.Data,load)
-- Type = 4 (Two port Touchstone network)
load = {}
load.Type = 4
-- Relative path from settings file location on drive
load.Filename = "example.s2p"
table.insert(activePortsConfiguration.Load.Data,load)
```

Note: Measurement data includes the reference impedance when determining the port loading.

Non-Active Port Configuration

Set up the load configuration to modify the load attached to a non-active port. There are three load configuration types: individual (single port definition), multiport single Touchstone (one Touchstone file for all the ports), and multiple single port Touchstone files (one file for each port). Only one configuration type can be used per calculation.

Specifying Individual Loading for Non-Active Ports

A load configuration table is used to store the individual port loading in for the non-active ports and the type property of the table is set to *Individual*. Each load is inserted in the *Data* table of the load configuration. An individual port load is defined as one of the following types:

- Complex
- Series RLC
- Parallel RLC
- Short or Open circuit
- One-port Touchstone (`.s1p`) file

Below is an example of how the load types can be defined

```
nonActivePortsConfiguration = {}
nonActivePortsConfiguration.Type = "Individual"
nonActivePortsConfiguration.Data = {}
-- Load specification for complex series load
load = {}
load.Label = "Port_1"
load.Index = 1
load.Type = "Complex"
load.Value = pf.Complex(150,20)
table.insert(nonActivePortsConfiguration.Data,load)
-- Load specification for series RLC load
load = {}
load.Label = "Port_2"
load.Index = 2
load.Type = "SeriesRLC" -- or "ParallelRLC"
load.R = 30
load.L = 20e-9
load.C = 5e-12
table.insert(nonActivePortsConfiguration.Data,load)
-- Load specification for short or open circuit
load = {}
```

```
load.Label = "Port_3"
load.Index = 3
load.Type = "Open" -- or "Short"
table.insert(nonActivePortsConfiguration.Data,load)
load = {}
load.Label = "Port_3"
load.Index = 4 -- index should match port number
load.Type = "Touchstone1port"
-- Relative path from Lua settings file location on drive
load.Filename = "test_file.s1p"
table.insert(nonActivePortsConfiguration.Load.Data,load)
```

Note: `Filename` (property) is the relative path from the settings file to the Touchstone file.

Specifying Loading with Individual Touchstone Files

A *nonActivePortsConfiguration* table is used to store the individual port loading in for the non-active ports and the type property of the table is set to *Individual_Touchstone_files*. Each load is inserted in the *Data* table of the load configuration. For the individual Touchstone files type configuration, it is assumed that each load is a single `.s1p` file.

```
nonActivePortsConfiguration = {}
nonActivePortsConfiguration.Type = "Individual_Touchstone_files"
nonActivePortsConfiguration.Data = {}
load = {}
load.Label = "Port_1"
load.Index = 1
-- Relative path from Lua settings file location on drive
load.FileName = "test_file.s1p"
table.insert(nonActivePortsConfiguration.Data,load)

load = {}
load.Label = "Port_2"
load.Index = 2
-- Relative path from Lua settings file location on drive
load.FileName = "test_file.s1p"
table.insert(nonActivePortsConfiguration.Data,load)
```

Note: `Filename` (property) is the relative path from the settings file to the Touchstone file.

Specifying Loading with a Single Touchstone File

A *nonActivePortsConfiguration* table is used to store the port loading for the non-active ports and the type property of the table is set to *Touchstone*. For the Touchstone configuration type, it is assumed that all the loads are defined in a single Touchstone file.

```
nonActivePortsConfiguration = {}
nonActivePortsConfiguration.Type = "Touchstone"
-- Relative path from Lua settings file location on drive
nonActivePortsConfiguration.FileName = "4_port_load.s4p"
```

Field Data Configuration

Set up the field data configuration table to be used with the measurement method to map to the correct near field or far field data (available under **Stored data** in the `.pfs` file).

A *fieldDataConfiguration* table is used with a *FarFields* or *NearFields* attribute. It is only required to specify the label of the stored field and the index of the item in the table maps to the port number in the multiport configuration.

```
fieldDataConfiguration = {}
-- Specify the Far field data
-- Note the label should match the stored data in the .pfs file.
fieldDataConfiguration["FarFields"] = {}
fieldDataConfiguration["FarFields"][1].Label = "FarField_1"
fieldDataConfiguration["FarFields"][2].Label = "FarField_2"
fieldDataConfiguration["FarFields"][3].Label = "FarField_3"
fieldDataConfiguration["FarFields"][4].Label = "FarField_4"

-- Specify the Near field data
-- Note the label should match the stored data in the .pfs file.
fieldDataConfiguration["NearFields"] = {}
fieldDataConfiguration["NearFields"][1].Label = "NearField_1"
fieldDataConfiguration["NearFields"][2].Label = "NearField_2"
fieldDataConfiguration["NearFields"][3].Label = "NearField_3"
fieldDataConfiguration["NearFields"][4].Label = "NearField_4"
```

Processing Options

Set up the *processingOptions* table to be used with the command-line interface. The *processingOptions* table defines which data is exported as well as setting non-default values.

A *processingOptions* table is used with the following attributes to set the processing options for the multiport script.

deembedToAntenna
: Set this option to move the reference plane to where the port parameters were calculated before the source loading definition (if the sources are loaded). This variable maps to the **Subtract source loading** option on the GUI.

referenceImpedance
: Set the system reference impedance for the measurement method.

storeDataSets
: Set this option to store the data sets in the `.pfs` session.

includeSourceReferenceImpedance
: Set this option to include the source reference impedance in the multiport calculations.

mergeWithExistingStoredData
: Set this option to merge the new multiport calculation with existing stored data.

calculateScaledRequests
: Set this option to calculate the scaled far fields and near fields, and currents if available.

exportDataSets
: Set this option to export the far fields and near fields and save a POSTFEKO session with the port parameters.

prefix
: Add a result prefix for the stored data items.

exportScallingCoefficients
: Set this option to export the scaling coefficients to a `.xml` and `.mat` files.

validateModel
: Set this option when using the Feko model method to validate the model setup.

Note:

- The *processingOptions* table is only required if the Lua settings file is used with the command-line interface.
- Select the **Save settings to file (*.lua)** check box on the **Processing options** dialog to save the *processingOptions* table.

```
local processingOptions={
  ["deembedToAntenna"] = "false",
  ["referenceImpedance"] = "50",
  ["storeDataSets"] = "true",
  ["includeSourceReferenceImpedance"] = "false",
  ["mergeWithExistingStoredData"] = "false",
  ["calculateScaledRequests"] = "false",
  ["exportDataSets"] = "false",
  ["prefix"] = "SNP",
  ["exportScallingCoefficients"] = "false",
  ["validateModel"] = "false"
}
```

Command Line Arguments for Launching the Multiport Post-Processing Macro

The Multiport application macro can be called via the command line through POSTFEKO. Use the `--configure-script` argument to pass configuration information to the multiport post-processing script.

Use the following command-line parameters to launch the multiport script:

```
postfeko [SESSION] --non-interactive --run-script=[SCRIPT] --configure-script="[OPTIONS]"
```

Note: The *--configure-script* parameter requires that the input variables are wrapped in quotes with an empty space separating each variable.

SESSION
: A single session (`.pfs`) may be specified that may or may not exist.

SCRIPT
Specify the path to the `MultiportPostProcessing.lua` file.

OPTIONS

mppSettings
The path to a multiport settings file.

outputDirectory
Specify the path for the exported result files.

fekoModel
Specify the path to the `.fek` file, which contains the pre-processed multiport configuration data.

snpFile
Specify the path to the `.snp` file for the multiport system using measurements.

outputDirectory
Specify the path for the exported result files.

referenceImpedance
[Optional] The real part of the system reference impedance used in the measurements.

prefix
[Optional] Specify a result prefix.

pfsFile
[Optional] Specify the path to `.pfs` file containing additional field measurements.

> **Note:** The mapping from the stored data to each port is required in the multiport settings file.

Output File Format for Scaling Coefficients

The file format and data structure for the output files (`.xml` and `.mat`) are described. These files are generated for storing the scaling coefficients for a multiport calculation.

XML File Format

The `.xml` file has the following structure. The scaling coefficient, voltage, impedance, current and the reference impedance data for each port are grouped in the *result* element, and the frequency data is grouped in the *frequencies* element.

```
<multiport xmlns:xsi='http://www.w3.org/2001/XMLSchema-
instance' version='1' xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
  <modelname date='2020-11-03'>modelname.fek</modelname>
  <units voltage='V' current='A' impedance='Ohm' frequency='Hz'/>
  <frequencies>
    <frequency id='1' value='1000000000'/>
    <frequency id='2' value='1100000000'/>
  </frequencies>
  <result>
    <port id='1' name='Port1'>
```

```
      <data freqid='1'>
        <voltage re='-0.037831747128302' im='-0.043140582569938'/>
        <current re='-0.00054045353040432' im='-0.00061629403671339'/>
        <impedance re='70' im='0'/>
        <referenceimpedance re='50' im='0'/>
        <scalingcoefficient re='0.010809070608086' im='0.012325880734268'/>
      </data>
      <data freqid='2'>
        <voltage re='-0.02772197810429' im='0.011073202066871'/>
        <current re='-0.00039602825863272' im='0.00015818860095529'/>
        <impedance re='70' im='0'/>
        <referenceimpedance re='50' im='0'/>
        <scalingcoefficient re='0.0079205651726544' im='-0.0031637720191059'/>
      </data>
    </port>
    <port id='2' name='Port2'>
      <data freqid='1'>
        <voltage re='1' im='0'/>
        <current re='0.0046363683836995' im='-0.0023538558703642'/>
        <impedance re='171.48521215495' im='87.06199333313'/>
        <referenceimpedance re='50' im='0'/>
        <scalingcoefficient re='0.73337361161563' im='-0.11883121546767'/>
      </data>
      <data freqid='2'>
        <voltage re='1' im='0'/>
        <current re='0.0022281204133835' im='-0.0021158448496953'/>
        <impedance re='235.99670515533' im='224.10477016802'/>
        <referenceimpedance re='50' im='0'/>
        <scalingcoefficient re='0.60952650039195' im='-0.10852699197221'/>
      </data>
    </port>
  </result>
</multiport>
```

Result Data Format in `.mat` File

The scaling coefficient data is stored in a `.mat` file in the following result structure *ResultData_<modelname or snp file>*. The frequency data can be accessed as follows using GNU Octave or Altair Compose:

```
ResultData_<modelname>.Frequencies
```

The scaling coefficients and some additional port results voltage, current, impedance and the reference impedance can be accessed as follows using GNU Octave or Altair Compose:

```
ResultData_<modelname>.<portname>.scalingcoefficient
ResultData_<modelname>.<portname>.voltage
ResultData_<modelname>.<portname>.current
ResultData_<modelname>.<portname>.impedance
ResultData_<modelname>.<portname>.referenceimpedance
```

Multiport Post-Processing Limitations

The *Multiport post-processing application macro* has several limitations.

The following options are not supported by the *Multiport post-processing* application macro:

- The application macro requires a specific label for the ports and loads when using the Feko model as input. The label mapping should be consistent between ports and loads. The recommended naming convention for the loads and ports are <label>_x, where x is the port number.
- The directivity factor for far fields is not calculated (the gain factor is calculated). Any directivity that would be calculated would have to make several assumptions, and you could easily obtain incorrect results. If directivity had to be calculated, it could be done in one of the following ways:
  - The power lost in the loads, added by the post-processing can be taken into account to approximate the directivity. The approximation is only valid as long as the model contains no other loads, and the model consists of lossless materials.
  - The far field can be integrated, but this requires a sufficiently fine sampled far field for the results to be accurate.
- The application macro assumes that the power settings were not changed from the default (PW 0).
- Current sources are not supported for field calculations. S-parameters are only calculated where there were changes in the load values.
- Exporting current data are not supported since the data is associated with a mesh.

Example 1: Feko Model as Input

Example 1 uses a CADFEKO model (`plate4prt.cfx`) as input for the Multiport post-processing.

The results between the Feko simulation and the *Multiport post-processing* application macro calculation are compared.

Creating the CADFEKO Model for Post-Processing

Use the `GenerateMultiportConfigurations.lua` application macro in CADFEKO to create the model.

> **Note:** Requirements for the **Generate multiport configurations** application macro:
> - The model should contain a single standard configuration.
> - The model should have no sources defined.
> - The model should contain more than one port.
> - All ports should have loads terminated by the reference impedance.
> - The naming convention for loads and ports are *<label>_x*, where x is the port number.

1. Open `plate4prt.cfx` in CADFEKO.

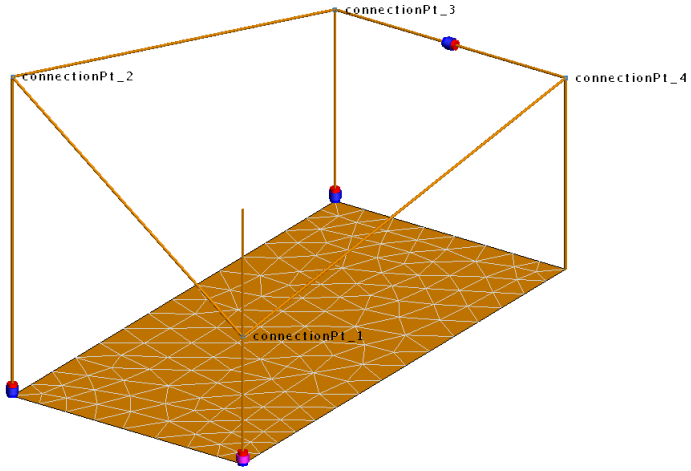


Figure 620: Example plate with four wire ports in CADFEKO.

> **Tip:** Find the examples in the <FEKO_SHARED_HOME> directory:
> `<FEKO_SHARED_HOME>/installedapplicationmacrolibrary/POSTFEKO/MultiportCalculation/examples`.

2. Run **Generate multiport configurations** application macro in CADFEKO.
The **Modify CADFEKO model** dialog is displayed.

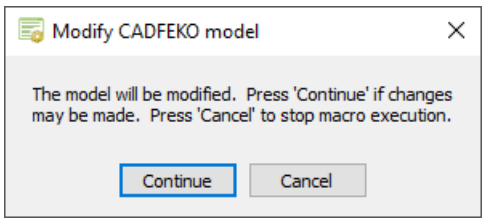


*Figure 621: The **Modify CADFEKO model** dialog.*

3. Click **Continue** to create the multiport configurations.
A standard configuration is created for each port. In each configuration, a single port is excited with the reference impedance from the assigned load. The requests from standard configuration **config** are transferred to each configuration to calculate the fields for each excitation.

*Figure 622: An example of the multiport configurations generated by the **Generate multiport configurations** application macro.*

4. Run the Feko Solver.

Calculating the Port Parameters and Field Data

Use the **Multiport post-processing** application macro in POSTFEKO to calculate the port reflections and field data for a model (`plate4prt.fek`) with different load configurations.

1. Open POSTFEKO and run the **Multiport post-processing** application macro from the application macro library.
   The **Multiport post-processing** dialog is displayed.
2. Specify the input method for the **Multiport post-processing** application macro.
   For this example, a Feko model (`plate4prt.fek`) is used as input.

   > **Tip:** Find the examples in the <FEKO_SHARED_HOME> directory:
   > `%FEKO_SHARED_HOME%/installedapplicationmacrolibrary/POSTFEKO/MultiportCalculation/examples`.

   a) Under **Definition method**, select **Feko model**.
   b) In the **Model** field, browse to the file location of `plate4prt.fek`.

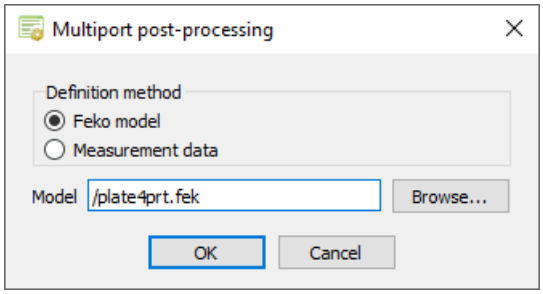


*Figure 623: The **Multiport post-processing** dialog.*

   c) Click **OK** to close the **Multiport post-processing** dialog.

   The **Processing options** dialog is displayed.
3. Specify the processing options and data handling.
   a) Under **Port (source and load) information**, select **Specify using dialogs**.
   b) Under **Data handling**, select **Replace stored data (if they exist)**.
   c) Under **Export options** , clear the **Export generated results (`*.ffe`,`*.hfe`,`*.efe`,`*.pfs`)** check box.
   d) [Optional] Under **Export options**, select the **Export scaling coefficients (*.xml, *.mat)** to export the scaling coefficients to file.
   e) [Optional] Under **Additional options**, select **Add reference impedance** to verify the model setup.
   f) [Optional] Under **Additional options**, select **Add reference impedance (Z0) in calculations** to
   g) [Optional] Under **Additional options**, select **Subtract source loading (move the reference plane before the source load)** to verify the model setup.
   h) Under **Additional options**, select **Calculate scaled fields and currents** to calculate the scaled fields and currents.

i) [Optional] Under **Additional options**, select **Save settings to file (*.lua)** to save the multiport settings to file.

j) [Optional] In the **Results prefix (optional)** field, specify a results prefix to group the calculated results in POSTFEKO.

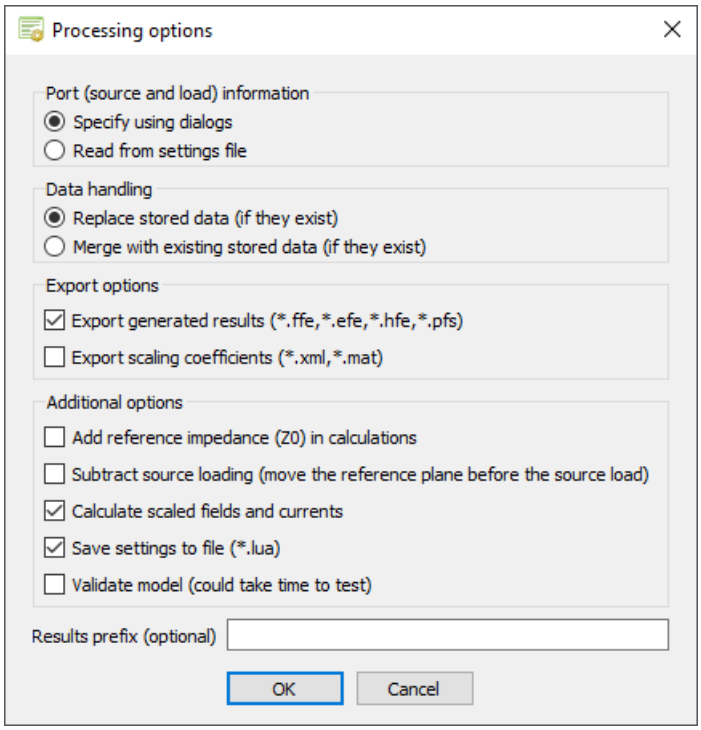


*Figure 624: The **Processing options** dialog.*

k) Click **OK** to close the **Processing options** dialog.

The **Select ports to load** dialog is displayed.

4. Specify the non-active (terminated) ports.

For this example, only *Port_3* and *Port_4* are non-active (terminated) ports.

a) Clear the **Port_1** check box.

b) Clear the **Port_2** check box.

c) Under **Load specification type**, select **Point loads (multiple 1-port loads of varying types)** to specify the individual loads.

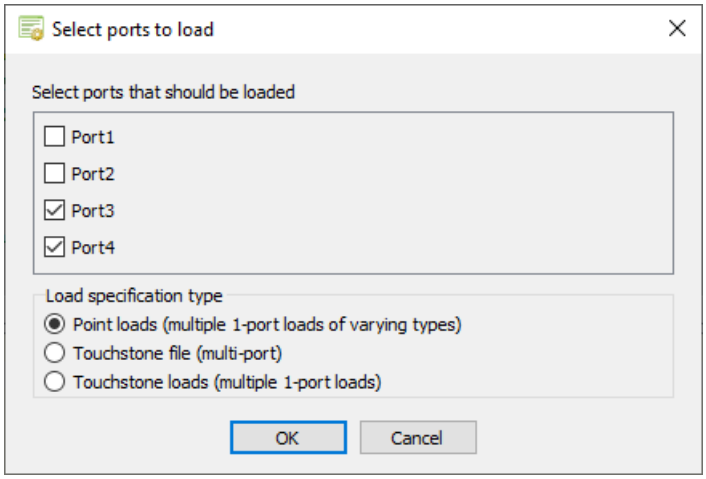


*Figure 625: The **Select ports to load** dialog.*

d) Click **OK** to close the **Select ports to load** dialog.

The **Load type for terminated ports** dialog is displayed.

5. Specify the load types for the non-active (terminated) ports.

a) In the **Load type for port Port_3** field, select **Complex load.**

b) In the **Load type for port Port_4** field, select **Complex load.**

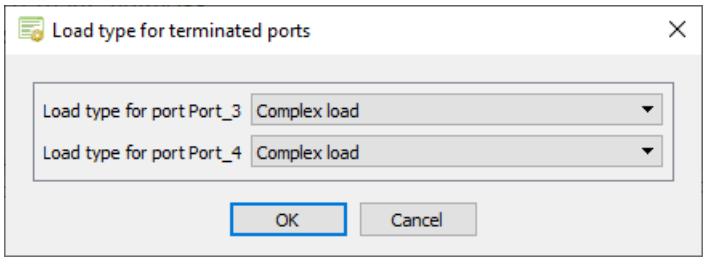


*Figure 626: The **Load type for terminated ports** dialog.*

c) Click **OK** to close the **Load type for terminated ports** dialog.

The **Select load parameters** dialog is displayed.

6. Specify the load values for the non-active (terminated) ports.

a) Under **Port_3 (Complex impedance)**, specify the following values in Ohm:

- Real component: 25
- Imaginary component: 30

b) Under **Port_4 (Complex impedance)**, specify the following values in Ohm:

- Real component: 75
- Imaginary component: 0

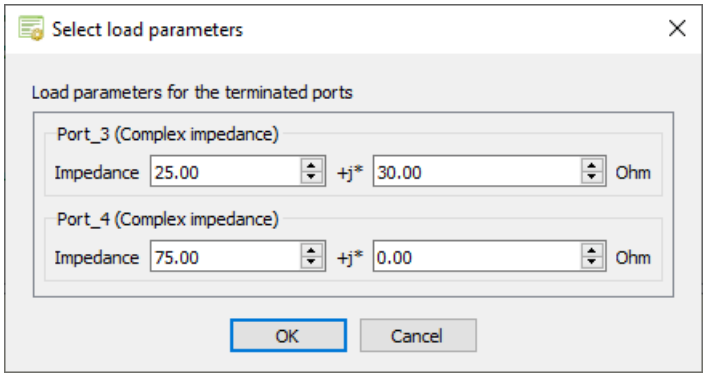


*Figure 627: The **Select load parameters** dialog.*

c) Click **OK** to close the **Select load parameters** dialog.

The **Select source parameters** dialog is displayed.

7. Specify the excitations for the active ports and the load impedances.

a) Specify the excitation for **Port_1**.

- Under **Port_1**, in the **Definition method** drop-down list, select **Direct connection**.
- Specify the following values:
  - **Amplitude**: 1 V
  - **Phase**: 0°

b) Specify the excitation for **Port_2**.

- Under **Port_2**, in the **Definition method** drop-down list, select **Direct connection**.
- Specify the following values:
  - **Amplitude**: 2 V
  - **Phase**: 0°

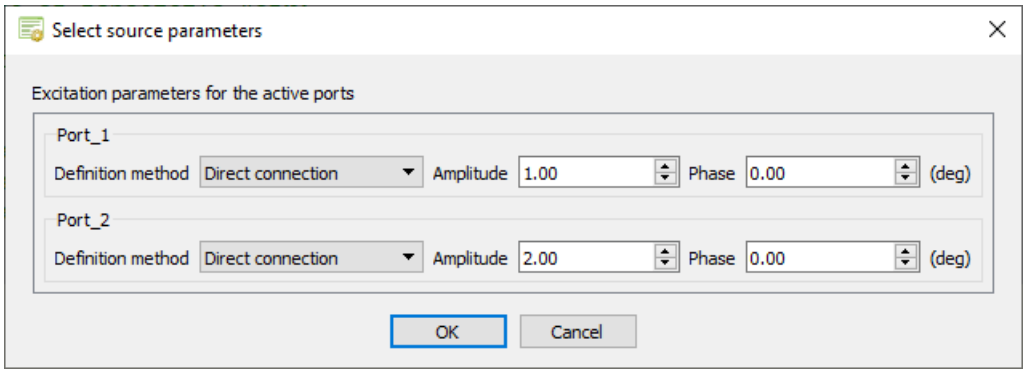


*Figure 628: The **Select source parameters** dialog*

c) Click **OK** to close the **Select source parameters** dialog.

The new results are calculated and available in POSTFEKO in the **Project Browser** under **Stored data**.

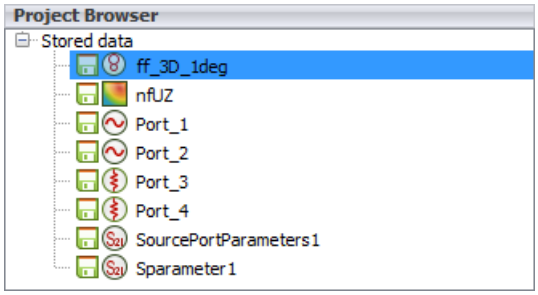


*Figure 629: The multiport results under **Stored data**.*

8. [Optional] Run the `plate4prt_example1_reference.cfx` in CADFEKO, load the `plate4prt_example1_reference.fek` in the `plate4prt.pfs` session and compare the results to the Solver.
The field data is compared as calculated by the *Multiport post-processing* application macro to the solution obtained by the Solver, see Figure 630 and Figure 631.

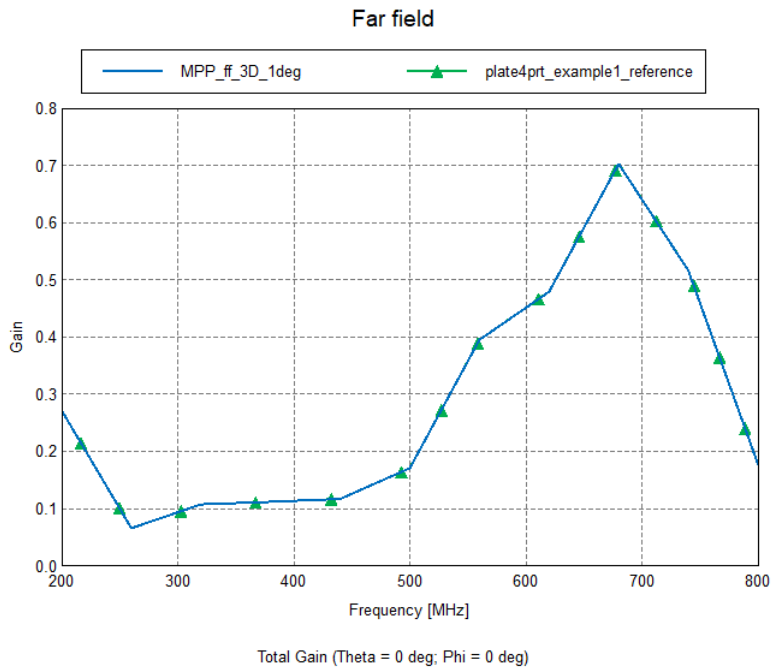


Figure 630: Comparison of near field values as a function of frequency at an arbitrary position.

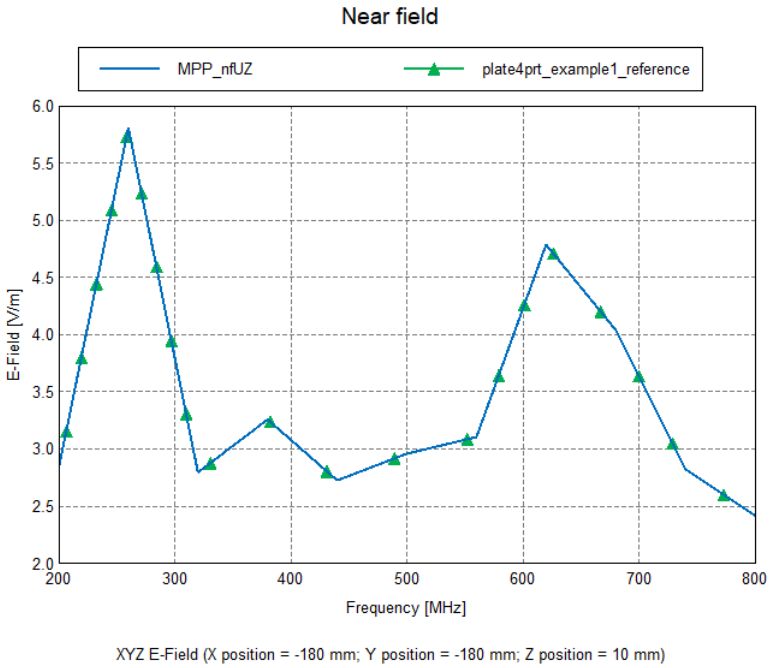


Figure 631: Comparison of far field gain over frequency in an arbitrary direction.

Example 2: Measurements (Stored Data) from a POSTFEKO Session as Input

Example 2 shows how to use stored far field and near field data, in a POSTFEKO session with a multiport S-parameter Touchstone (`.snp`) file in the *Multiport post-processing* application macro.

Calculating the Port Parameters and Field Data

For this example, a `plate4prt.pfs` file containing the far fields and near fields for each port is provided with the multiport S-parameter Touchstone file.

1. Open POSTFEKO and run the **Multiport post-processing** application macro.
   The **Multiport post-processing** dialog is displayed.
2. Specify the input method for the **Multiport post-processing**application macro.

   For this example, measurement data is used as input.

   > **Tip:** Find the examples in the <FEKO_SHARED_HOME> directory:
   > `<FEKO_SHARED_HOME>/installedapplicationmacrolibrary/POSTFEKO/MultiportCalculation/examples`.

   a) Under **Definition method**, select **Measurement data**.
   b) In the **S-parameters (DUT)** field, browse to the file location of `plate4prt_SP.s4p`.
   c) Under **Additional data**, select the **Include transmission measurements** check box.

d) Under **Additional data** in the **Transmission measurements** field, browse to the file location of `plate4prt_field_data.pfs`.

e) In the **Reference impedance (Z0)** field, enter 50 Ohm.

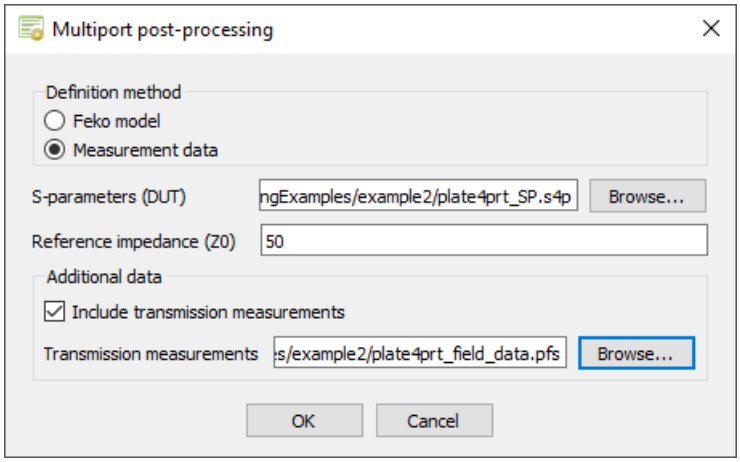


*Figure 632: The **Multiport post-processing** dialog.*

f) Click **OK** to close the **Multiport post-processing** dialog.

The **Processing options** dialog is displayed.

3. Specify the processing options and data handling.

a) Under **Port (source and load) information**, select **Specify using dialogs**.

b) Under **Data handling**, select **Replace stored data (if they exist)**.

c) Under **Export options** , clear the **Export generated results (`*.ffe,*.hfe,*.efe,*.pfs`)** check box.

d) [Optional] Under **Additional options**, select **Add reference impedance (Z0) in calculations** to include the reference impedance losses in the calculation.

e) [Optional] Under **Additional options**, select **Subtract source loading (move the reference plane before the source load)** to verify the model setup.

f) [Optional] Under **Additional options**, select **Save settings to file (*.lua)** to save the multiport settings to file.

g) [Optional] In the **Results prefix (optional)** field, specify a result prefix.

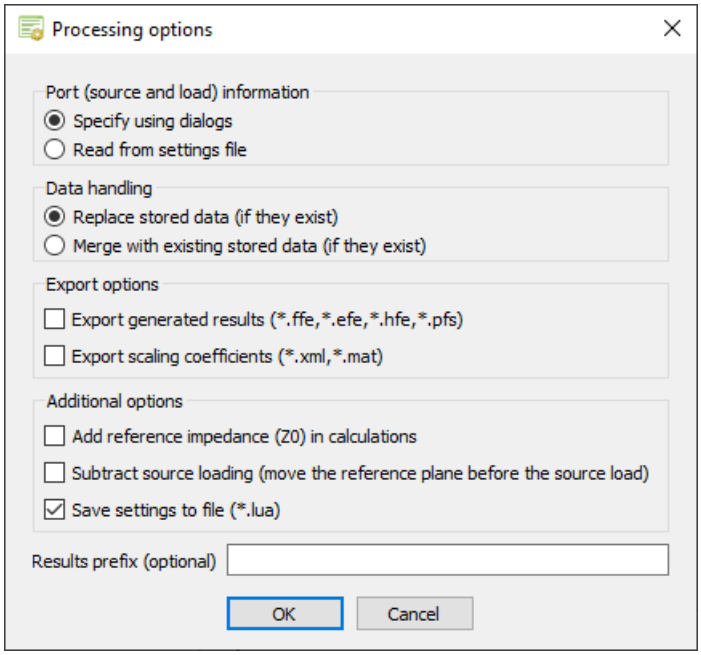


*Figure 633: The **Processing options** dialog.*

h) Click **OK** to close the **Processing options** dialog.

The **Select transmission measurement data** dialog is displayed.

4. Specify the transmission measurement data.

a) Under **Far fields**, map the far field data to the correct port.

- **Port1**: **FarField_Port_1**
- **Port2**: **FarField_Port_2**
- **Port3**: **FarField_Port_3**
- **Port4**: **FarField_Port_4**

b) Under **Near fields**, map the near field data to the correct port.

- **Port1**: **NearField_Port_1**
- **Port2**: **NearField_Port_2**
- **Port3**: **NearField_Port_3**
- **Port4**: **NearField_Port_4**

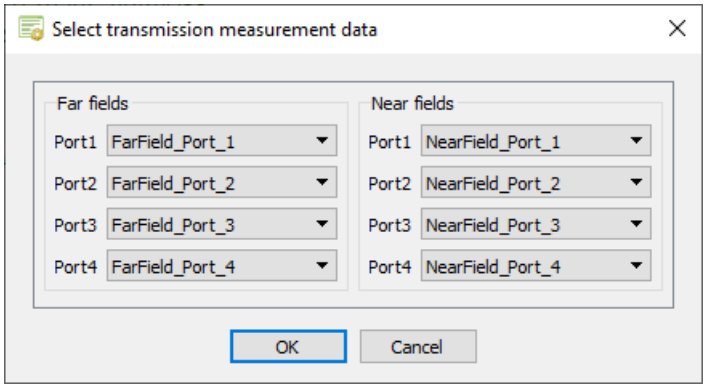


*Figure 634: The **Select transmission measurement data** dialog.*

c) Click **OK** to close the **Select transmission measurement data** dialog.

The **Select ports to load** dialog is displayed.

5. Specify the non-active (terminated) ports.

For this example, only *Port_3* and *Port_4* are non-active (terminated) ports.

a) Clear the **Port_1** check box.

b) Clear the **Port_2** check box.

c) Under **Load specification type**, select **Point loads (multiple 1-port loads of varying types)** to specify the individual loads.

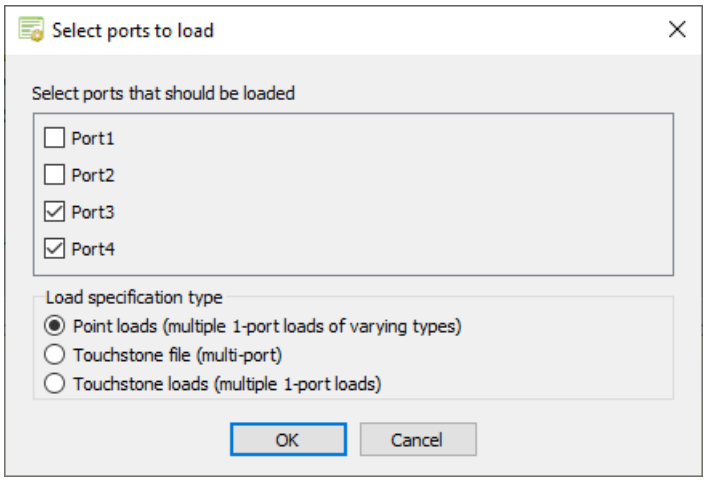


*Figure 635: The **Select ports to load** dialog.*

d) Click **OK** to close the **Select ports to load** dialog.

The **Load type for terminated ports** dialog is displayed.

6. Specify the load types for the non-active (terminated) ports.

a) In the **Load type for port Port_3** field, select **Complex load.**

b) In the **Load type for port Port_4** field, select **Complex load.**

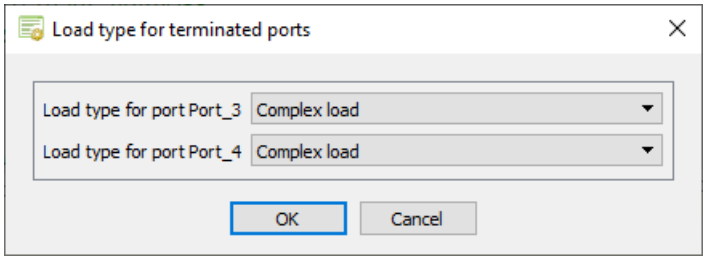


*Figure 636: The **Load type for terminated ports** dialog.*

c) Click **OK** to close the **Load type for terminated ports** dialog.

The **Select load parameters** dialog is displayed.

7. Specify the load values for the non-active (terminated) ports.

a) Under **Port_3 (Complex impedance)**, specify the following values in Ohm:

- Real component: 25
- Imaginary component: 30

b) Under **Port_4 (Complex impedance)**, specify the following values in Ohm:

- Real component: 75
- Imaginary component: 0

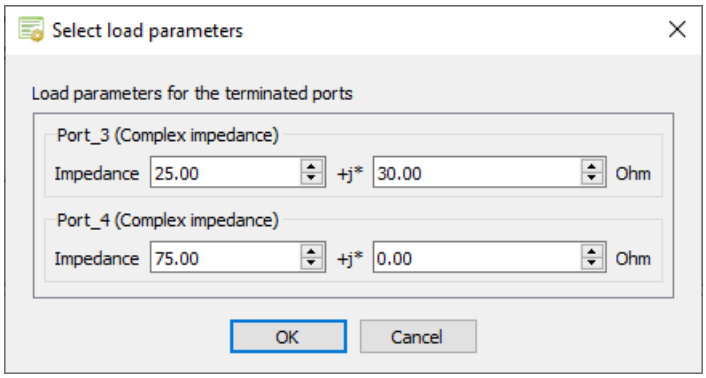


*Figure 637: The **Select load parameters** dialog.*

c) Click **OK** to close the **Select load parameters** dialog.

The **Select source parameters** dialog is displayed.

8. Specify the excitations for the active ports and the load impedances.

a) Specify the excitation for **Port_1**.

- Under **Port_1**, in the **Definition method** drop-down list, select **Direct connection**.
- Specify the following values:
  - **Amplitude**: 1 V
  - **Phase**: 0°

b) Specify the excitation for **Port_2**.

- Under **Port_2**, in the **Definition method** drop-down list, select **Direct connection**.

- Specify the following values:
  - **Amplitude**: 2 V
  - **Phase**: 0°

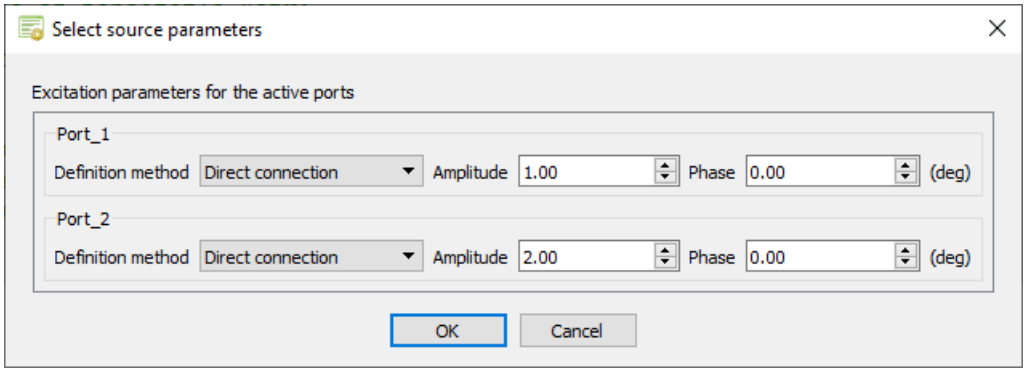


*Figure 638: The **Select source parameters** dialog*

c) Click **OK** to close the **Select source parameters** dialog.

The new results are calculated and available in POSTFEKO in the **Project Browser** under **Stored data**.

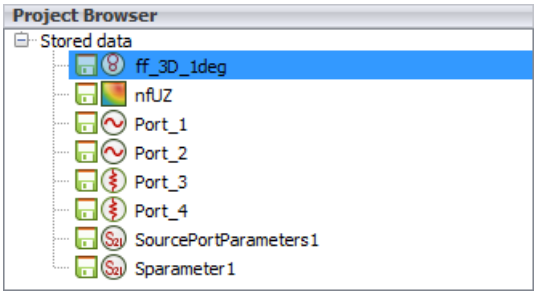


*Figure 639: The multiport results under **Stored data**.*

Example 3: Lua Settings File to Define Loading and Excitation for Each Port

Example 3 shows how to use a Lua settings file to set up the multiport active and non-active port configurations.

The Lua settings file applies to both the Feko model or measurement data as the input file for the *Multiport post-processing* script.

Creating the CADFEKO Model for Post-Processing

Use the **Generate multiport configurations** application macro in CADFEKO to create the model.

Note: Requirements for the **Generate multiport configurations** application macro:

- The model should contain a single standard configuration.
- The model should have no sources defined.
- The model should contain more than one port.
- All ports should have loads terminated by the reference impedance.
- The naming convention for loads and ports are *<label>_x*, where x is the port number.

1. Open `plate4prt.cfx` in CADFEKO.

   Tip: Find the examples in the <FEKO_SHARED_HOME> directory:
   `<FEKO_SHARED_HOME>/installedapplicationmacrolibrary/POSTFEKO/MultiportCalculation/examples`.

2. Run **Generate multiport configurations** application macro in CADFEKO.
   The **Modify CADFEKO model** dialog is displayed.

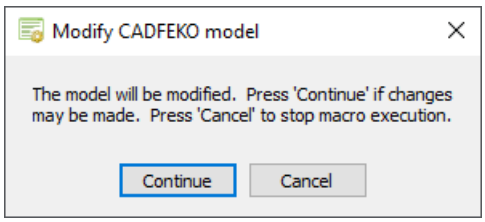


*Figure 640: The **Modify CADFEKO model** dialog.*

3. Click **Continue** to create the multiport configurations.
   A standard configuration is created for each port. In each configuration, a single port is excited with the reference impedance from the assigned load. The requests from standard configuration **config** are transferred to each configuration to calculate the fields for each excitation.

*Figure 641: An example of the multiport configurations generated by the **Generate multiport configurations** application macro.*

Creating the Lua Settings File

Specify the loading and excitations for each port using a Lua settings file.

1. Open a text editor and create the following settings file:

```
-- Non-active port configuration
nonActivePortsConfiguration = {}
nonActivePortsConfiguration.Data = {}
-- Empty non-Active port configuration since all ports are active.

-- Active port configuration
activePortsConfiguration = {}
activePortsConfiguration.Load ={}
activePortsConfiguration.Load.Data ={}
-- Source Port1 source = {}
source.Label = "Port1"
source.Index = 1
source.Value = pf.Complex(1,0)
table.insert(activePortsConfiguration,source)
-- Load attached to active port 1
-- Type = 1 (direct connection),
-- Type = 2 (complex),
-- Type = 3 (one port Touchstone network),
-- Type = 4 (two port Touchstone network)
load ={}
load.Type = 2
load.Value = pf.Complex(10,-30)
table.insert(activePortsConfiguration.Load.Data,load)
-- Source for Port2
source = {}
source.Label = "Port2"
source.Index = 2
source.Value = pf.Complex(1,0)
table.insert(activePortsConfiguration,source)
-- Load attached to active port 2
load ={}
load.Type = 2
load.Value = pf.Complex(70,-30)
table.insert(activePortsConfiguration.Load.Data,load)
-- Source Port3
source = {}
source.Label = "Port3"
source.Index = 3
source.Value = pf.Complex(1,0)
table.insert(activePortsConfiguration,source)
-- Load attached to active port 3
load ={}
load.Type = 2
load.Value = pf.Complex(105,-30)
table.insert(activePortsConfiguration.Load.Data,load)
-- Source Port4
source = {}
source.Label = "Port4"
source.Index = 4
source.Value = pf.Complex(1,0)
table.insert(activePortsConfiguration,source)
-- Load attached to active port 4
load ={}
load.Type = 2
load.Value = pf.Complex(50,0)
table.insert(activePortsConfiguration.Load.Data,load)
```

```
-- Return the configurations in a settings table
return {activePortsConfiguration, nonactivePortsConfiguration}
```

2. Save the `example3_settings.lua` file.

Calculating the Port Reflections and Field Data

Use the **Multiport post-processing** application macro in POSTFEKO to calculate the port reflections and field data for a model (`plate4prt.fek`) with different load configurations.

1. Open POSTFEKO and run the **Multiport post-processing** application macro from the application macro library.
   The **Multiport post-processing** dialog is displayed.
2. Specify the input method for the **Multiport post-processing** application macro.
   For this example, a Feko model (`plate4prt.fek`) is used as input.

   Tip: Find the examples in the <FEKO_SHARED_HOME> directory: `%FEKO_SHARED_HOME%/installedapplicationmacrolibrary/POSTFEKO/MultiportCalculation/examples`.

   a) Under **Definition method**, select **Feko model**.
   b) In the **Model** field, browse to the file location of `plate4prt.fek`.

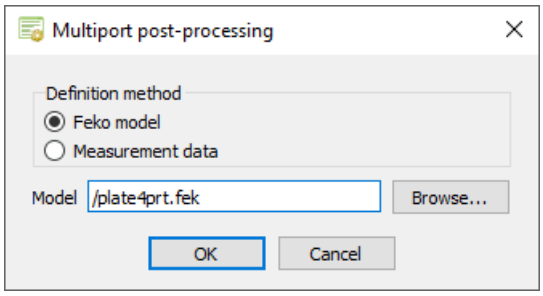


*Figure 642: The **Multiport post-processing** dialog.*

   c) Click **OK** to close the **Multiport post-processing** dialog.

   The **Processing options** dialog is displayed.
3. Specify the processing options and data handling.
   a) Under **Port (source and load) information**, select **Read from settings file**.
   b) Under **Data handling**, select **Replace stored data (if they exist)**.
   c) Clear the **Export generated results** check box.
   d) [Optional] Select **Validate model (could take time to test)** to verify that the model was set up correctly.
   e) [Optional] In the **Results prefix (optional)** field, specify a result prefix.

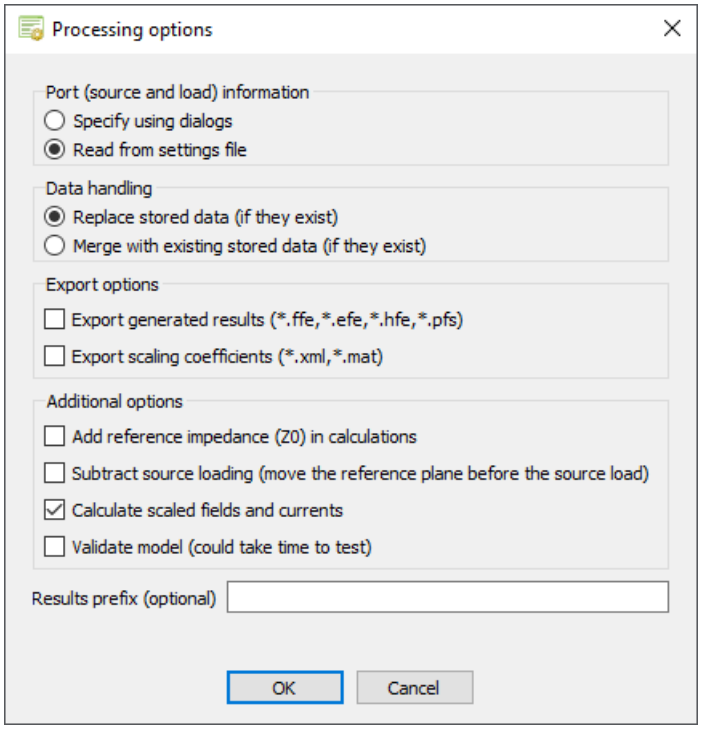


*Figure 643: The **Processing options** dialog.*

f) Click **OK** to close the **Processing options** dialog.

The **Select file containing port settings** dialog is displayed.

4. Specify the Lua settings file to define the loading and excitation for each port.

a) In the **File name** field, browse to the file location of `example3_settings.lua`.

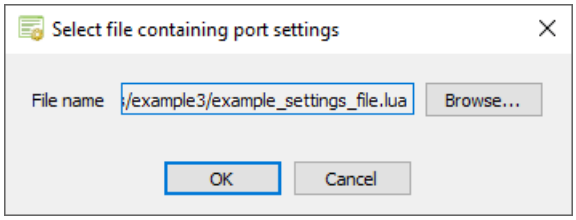


*Figure 644: The **Select file containing port settings** dialog.*

b) Click **OK** to close the **Select file containing port settings** dialog.

The new results are calculated and available in POSTFEKO in the **Project Browser** under **Stored data**.

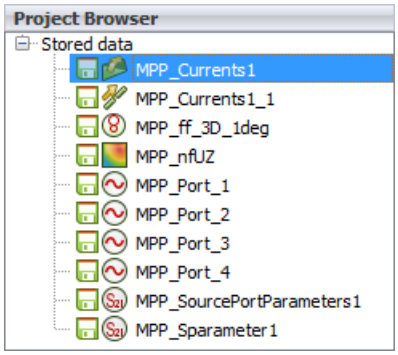


*Figure 645: The multiport results under **Stored data**.*

Example 4: Command-Line Interface

The multiport application macro supports calculation from the command-line by calling the application macro via POSTFEKO.

The first example shows how to use a Feko model with the interface. The second example uses a Touchstone file as input.

A pre-configured Lua settings file is used from Example 1: Feko Model as Input. The settings file contains a *processingOptions* table which is used to set up the settings required for a multiport run from the command-line. Both examples use the same Lua settings file. The Lua settings file is configured to export the scaling coefficients to file which can be used in other post-processing applications.

> **Note:** The settings file can be automatically generated by using the **Save settings to file (*.lua)** check box on the **Processing options** dialog from the GUI.

```
local processingOptions={
  ["deembedToAntenna"] = "false",
  ["referenceImpedance"] = "50",
  ["storeDataSets"] = "true",
  ["includeSourceReferenceImpedance"] = "false",
  ["mergeWithExistingStoredData"] = "false",
  ["calculateScaledRequests"] = "false",
  ["containsCurrentSource"] = "false",
  ["exportDataSets"] = "false",
  ["prefix"] = "",
  ["storeAdditionalData"] = "false",
  ["exportScallingCoefficients"] = "true",
  ["idealSource"] = "true",
  ["printSummary"] = "true",
  ["validateModel"] = "false"
}
local sourceConfiguration={
  [1] = {
    ["Value"] = "1 + 0i",
    ["Index"] = 1,
    ["Label"] = "Port_1"
  },
  [2] = {
    ["Value"] = "2 + 0i",
    ["Index"] = 2,
    ["Label"] = "Port_2"
  },
  ["Load"] = {
    ["Data"] = {
      [1] = {
        ["Label"] = "Port_1",
        ["Type"] = 1
      },
      [2] = {
        ["Label"] = "Port_2",
        ["Type"] = 1
      }
    }
  }
}
local loadConfiguration={
  ["Data"] = {
```

```
    [1] = {
      ["Value"] = "25 + 30i",
      ["Type"] = "Complex",
      ["Index"] = 3,
      ["Label"] = "Port_3"
    },
    [2] = {
      ["Value"] = "75 + 0i",
      ["Type"] = "Complex",
      ["Index"] = 4,
      ["Label"] = "Port_4"
    }
  },
  ["Type"] = "Individual"
}
return{sourceConfiguration,loadConfiguration,{},processingOptions}
```

Using a Feko Model

Execute a multiport calculation from the command-line using a Feko model file as input.

1. Open a Feko terminal.
2. Change to the directory where Example 4 is located using the following command

   ```
   cd <insert path>/example4
   ```

3. Use the following command to execute the multiport application macro from the command-line.

   ```
   postfeko plate4prt.pfs --non-interactive --run-script="<insert path>/
   MultiportPostProcessing.lua" --configure-script="mppSettings=[[<insertpath>/
   example4/plate4prt_multiport_settings.lua]] fekoModel=[[<insert path>/example4/
   plate4prt.fek]] outputDirectory=[[<insert path>/example4]] prefix='cmd1_'"
   ```

   Note: To specify paths for the *--configure-script* variable it is recommended to use literal strings, for example, [[<insert path>]].

   The results is available in `plate4prt.pfs`. The following additional exported files `cmd_plate4prt_scaling_coefficients.mat` and `cmd_plate4prt_scaling_coefficients.xml` are available for further post-processing in the specified *outputDirectory*.

Using a Touchstone File

Execute a multiport calculation from the command-line using a Touchstone file as input.

1. Open a Feko terminal.
2. Change to the directory where Example 4 is located using the following command

   ```
   cd <insert path>/example4
   ```

3. Use the following command to execute the multiport application macro from the command-line.

   ```
   postfeko plate4prt.pfs --non-interactive --run-script="<insert path>/
   MultiportPostProcessing.lua" --configure-script="mppSettings=[[<insertpath>/
   ```

```
example4/plate4prt_multiport_settings.lua]] snpFile=[[<insert path>/example4/
plate4prt.s4p]] outputDirectory=[[<insert path>/example4]] prefix='cmd2_'"
```

Note: To specify paths for the *--configure-script* variable it is recommended to use literal strings, for example, [[<insert path>]].

The results is available in `plate4prt.pfs`. The following additional exported files `cmd2_plate4prt_scaling_coefficients.mat` and `cmd2_plate4prt_scaling_coefficients.xml` are available for further post-processing in the specified *outputDirectory*.

9.4.4 Plot Multiple S-Parameter Traces Macro

Plot multiport S-parameter results.

This POSTFEKO application macro is ideal for plotting multiport S-parameter results, especially for simulations with a large number of ports. The application macro provides the user with the following options:

- Select which S-parameter configuration to analyse.
- Select whether to plot the reflection coefficients and/or transmission coefficients for all ports.
- Set the Y axis to dB.
- Select to create a new graph or to use the most recently created existing graph.

Example Model

An MRI birdcage model is used as an example to demonstrate the application macro that plots multiple S-parameter traces.

The birdcage MRI coil has two ports at the bottom rung. They are excited with equal magnitudes and 90° phase difference. Both rungs have sixteen equal-value capacitors, which tune the resonance of the system. The diameter of the coil is 30 cm. Around the coil is a cylindrical shield. The birdcage coil is aligned with the Z axis.

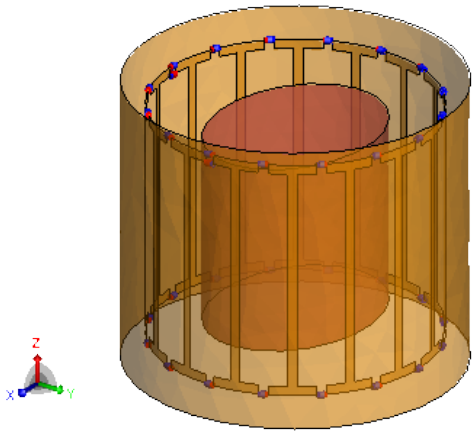


Figure 646: Generic birdcage MRI coil.

Tip: Find the examples in the <FEKO_SHARED_HOME> directory:

`<FEKO_SHARED_HOME>/installedapplicationmacrolibrary/POSTFEKO/PlotMultipleSParameterTraces/examples`.

Using the Application Macro

Execute the application macro in POSTFEKO to plot S-parameter traces on a Cartesian graph.

1. Start with a POSTFEKO session containing at least one model with S-parameter configuration results.

The results from a single S-parameter configuration request will be provided as input to the macro.

2. Execute the application macro to plot the S-parameters.
   A dialog prompts the user to select the configuration and plot settings.
3. Carry out the choices to set up the S-parameter graph.
   a) Select the S-parameter result.
   b) Indicate which coefficients to plot by selecting at least one of the options to **Plot the reflection coefficients** or **Plot the transmission coefficients**.
   c) Select **Plot in dB** if the Y axis should be in dB.
   d) Select **Plot on a new graph** to add the selected results to a new graph.
   e) Select **Finish**.

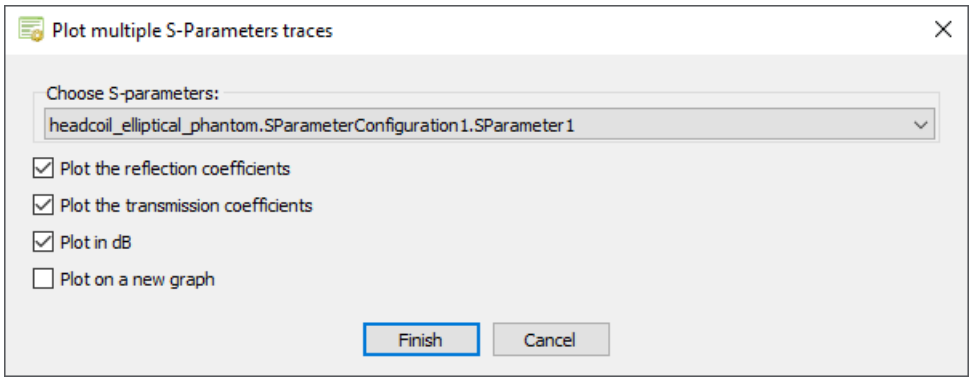


Figure 647: Dialog for selecting the result, quantities of interest and graph settings.

4. If a new graph is selected, enter a name for the graph on the next dialog and select **Finish**.

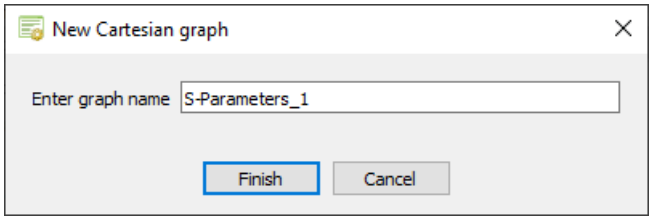


Figure 648: Dialog for entering the graph name.

5. View the graph generated by the macro.

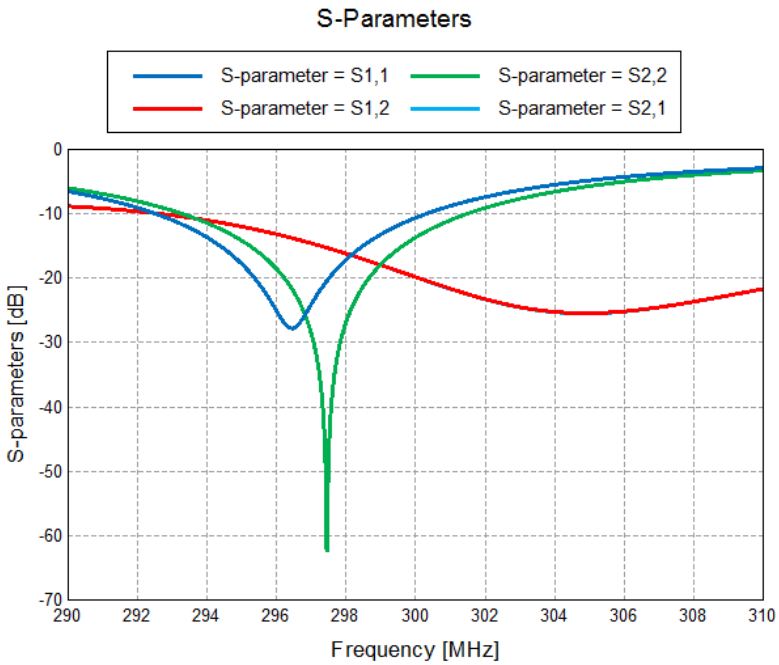


Figure 649: S-parameter graph with transmission and reflection coefficient magnitudes in dB for an MRI birdcage model.

9.4.5 Tile Windows

An application macro to tile any POSTFEKO views side-by-side.

It is often required to tile views side-by-side in POSTFEKO to visually compare the data in them. This application macro sets up such a view as follows:

1. Start with a POSTFEKO session containing more than one view.
2. Run the application macro.
3. Select **Vertical** or **Horizontal** tile **Orientation**.
4. Select the **Windows** to tile.
5. Select **OK**.

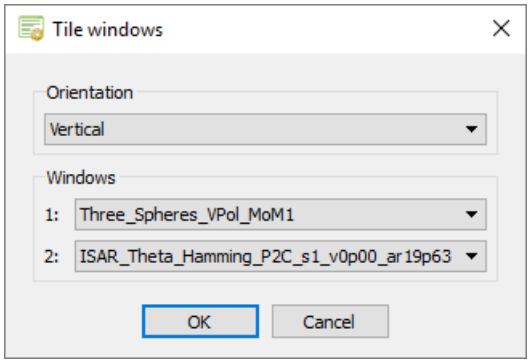


*Figure 650: The **Tile windows** dialog.*

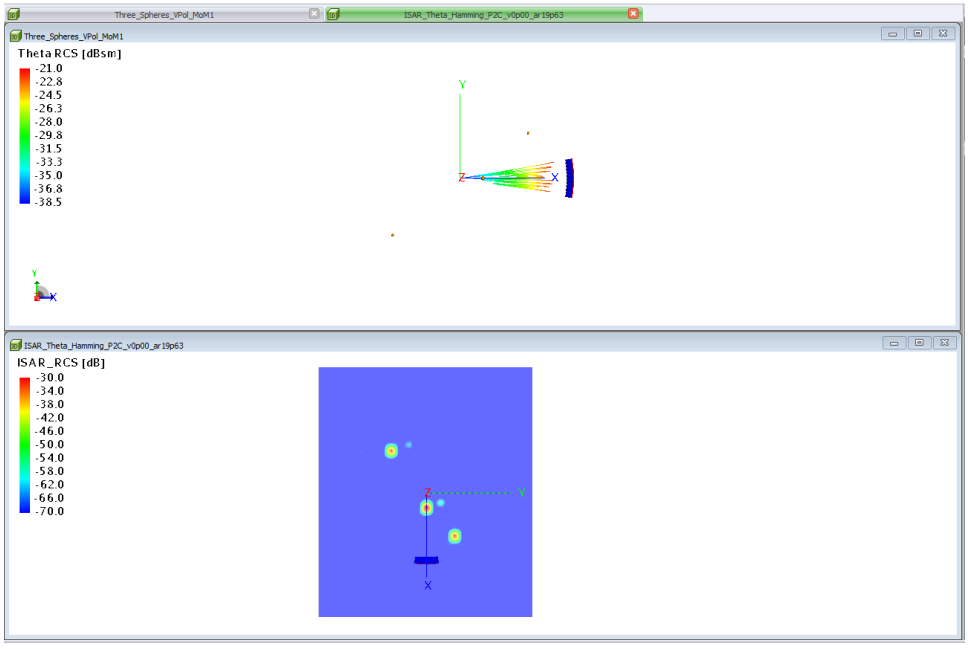


Figure 651: Two views tiled vertically

9.4.6 Plot Interference Matrix

This application macro uses the S-parameter data from a Feko simulation to display a coupling matrix for a multi-port scenario.

Interference Matrix Definition

An interference matrix is a way to illustrate the cross-coupling between ports of a multiport system. Example applications are antenna arrays, feed networks or power dividers.

The off-diagonal components of a multiport S-parameter matrix contain all the required information to create an interference matrix. The user sets the maximum and minimum thresholds and the colours indicate if the cross-coupling between the ports are within the acceptable range. The rows represent the transmitting ports and the columns the receiving ports. Green boxes indicate that interference between the ports is below the threshold minimum. The diagonal elements contain no cross-coupling information since it is the power measured at the port from exciting the port itself.

Interference matrix @ 6E+07 Hz

	Port 1	Port 2	Port 3
Port 1		-25.2	-25
Port 2	-25.2		-25
Port 3	-25	-25	

Legend

Below marginal threshold	x < -20
Within marginal range	-20 ≤ x ≤ -15
Above marginal threshold	x > -15
No coupling	

Figure 652: Example of an interference matrix with a threshold range between -15 and -20 dB.

Example of Interference Between Three Dipoles

An antenna array is used to show the workflow of the application macro and calculate the interference matrix for the multiport scenario.

In `three_dipole_example.cfx`, three dipole antennas are created and positioned a distance from each other. Electric symmetry is used to reduce runtime but is not required.

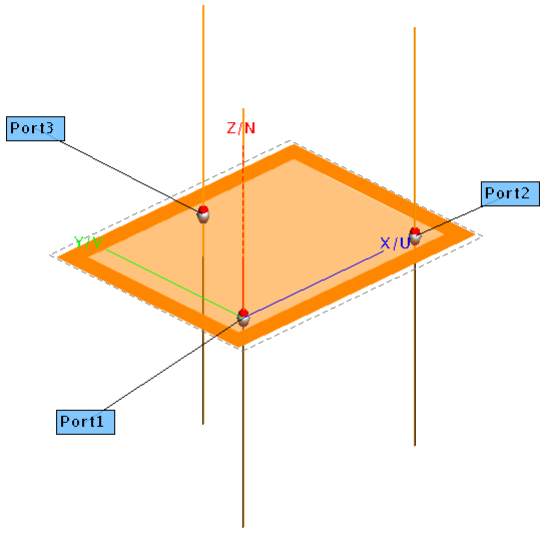


Figure 653: Example of model setup for the three dipole scenario.

Tip: Find the examples in the <FEKO_SHARED_HOME> directory:

`<FEKO_SHARED_HOME>/installedapplicationmacrolibrary/POSTFEKO/PlotInterferenceMatrix/examples`.

In the S-parameter solution request, all the ports are added, and the characteristic impedance for each port is set to 50 ohms.

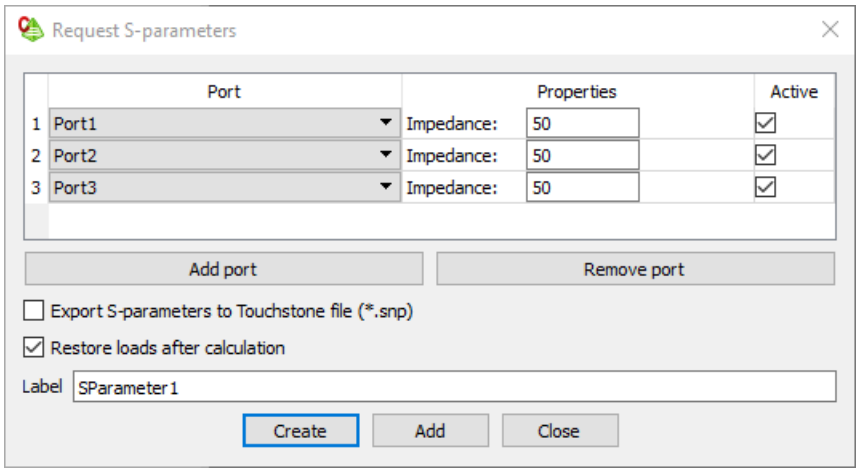


*Figure 654: The **Request** S-parameters dialog.*

Restriction: All ports of interest need to be added to the S-parameter request, and all ports in the request must be set active. The application macro calculates the complete interference matrix for all ports in the request.

Using the Application Macro

Execute the application macro in POSTFEKO to determine the interference matrix for the multiport S-parameter calculation.

1. Open POSTFEKO, add a model and run the **Plot interference matrix** application macro.

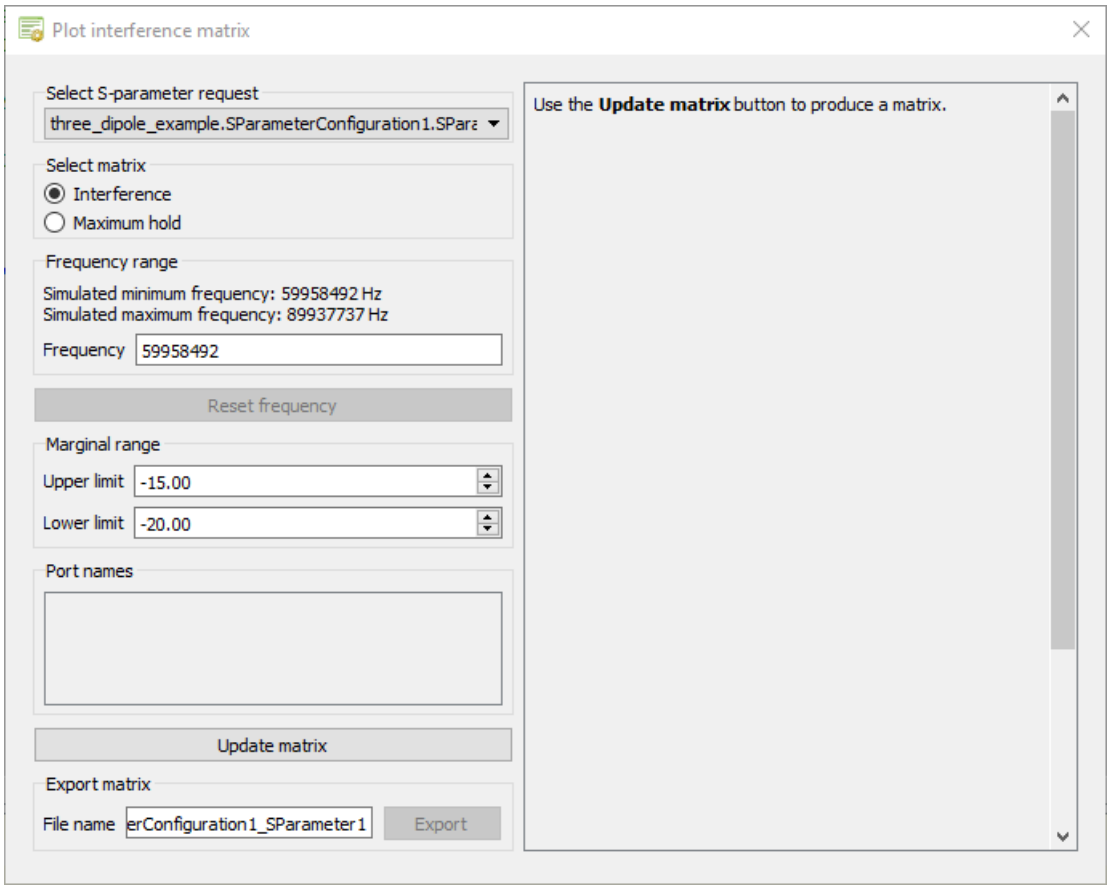


*Figure 655: The **Plot interference matrix** dialog.*

2. Select the **S-parameter request** from the drop-down list.

> **Note:** All ports in the request must be active for the application macro to successfully calculate the interference matrix.

3. Select the **Interference** or **Maximum hold** matrix to be calculated.
4. If **Interference** is selected, enter the **Frequency** for calculation in Hz.
5. If **Maximum hold** is selected, enter the **Upper frequency** and **Lower frequency** in Hz and the **Number of samples**.
6. Set the **Upper limit** for the marginal range in dB.
7. Set the **Lower limit** for the marginal range in dB.
8. Click **Update matrix**.
   The matrix is displayed on the dialog.

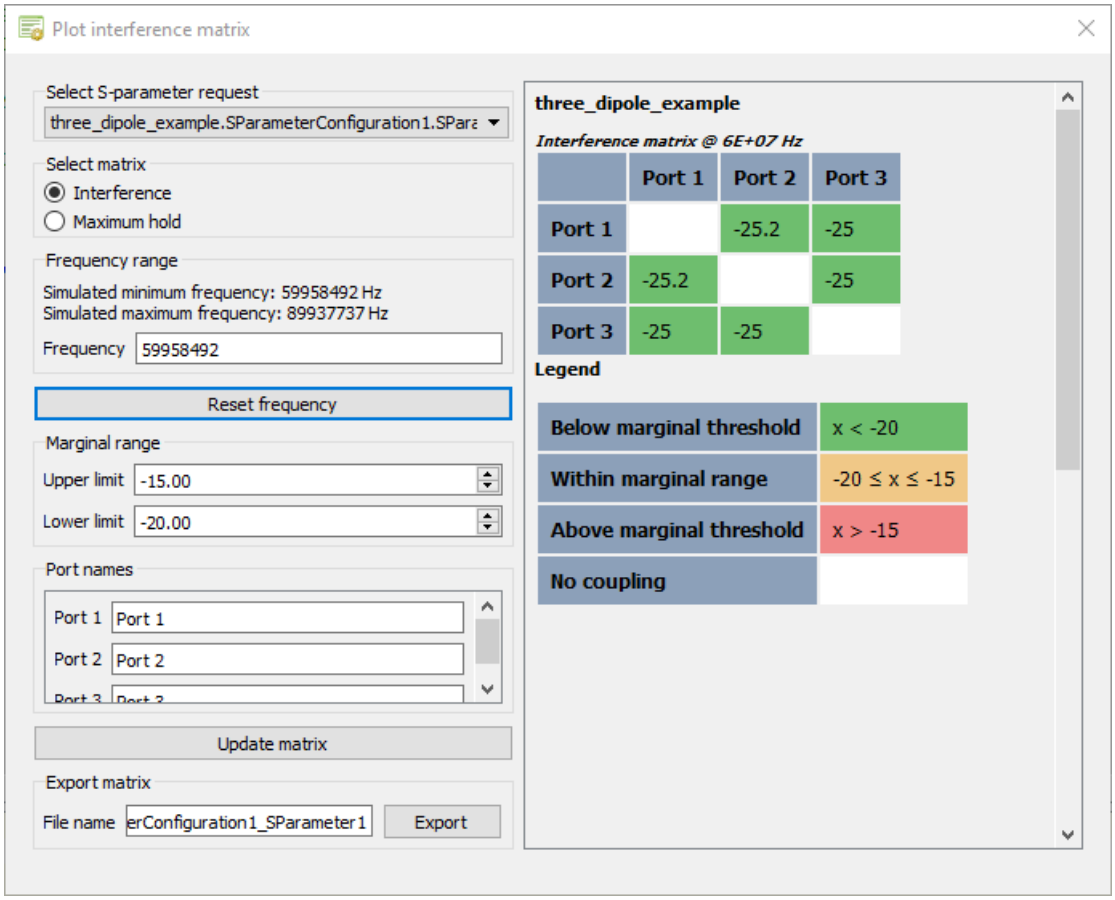


*Figure 656: The **Plot interference matrix** dialog after updating the matrix.*

9. [Optional] Modify the port names and click **Update matrix**.
10. [Optional] Enter a **File name** and click **Export**.
    The interference matrix is exported to a `.html` file.
11. Press Escape or click **X** in the top right corner to close the **Plot interference matrix** dialog.

9.4.7 Characteristic Mode Synthesis and Design

The characteristic mode synthesis and design application macro is a post-processing application macro that can be used to calculate a weighted sum for the currents, near fields, and far fields requests for specific characteristic modes of interest. The application macro uses a modified version of the modal weighting coefficient (MWC) to use the radiating phase when synthesising the results with the macro.

Characteristic Mode Analysis

Characteristic mode analysis (CMA) is the numerical calculation of a weighted set of orthogonal current modes that are supported on a conducting surface. The sets of characteristic near fields and far fields associated with these characteristic currents can provide insight into the radiating properties of structures, allowing for a systematic approach to antenna design and placement.

Characteristic modes are obtained by solving a particular weighted eigenvalue equation that is derived from the method of moments impedance matrix. Feko has a built-in solver that calculates these modes, with no need for post-processing by the user. The eigen values, modal significance, characteristic angles, currents, near fields, and far fields can be visualised in POSTFEKO.

Example of an Antenna Attached to a Plate

The example illustrates the synthesis method of the application macro. Optionally, the design method can be used to specify a custom weighting coefficient.

An example of the antenna and plate model is shown below. The antenna is excited at the corner of the plate with a voltage source; for this example, the frequency range is 700 MHz to 960 MHz.

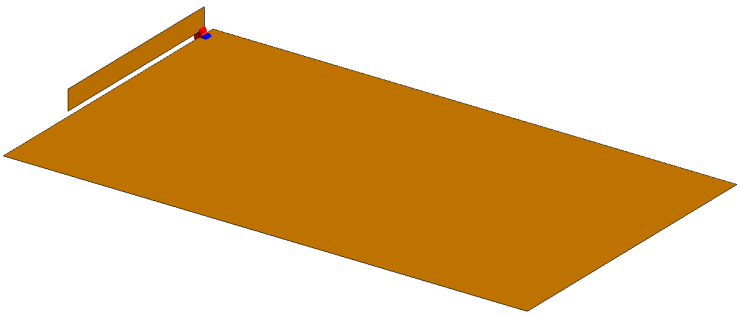


Figure 657: Example of the antenna attached to a plate.

Tip: Find the examples in the <FEKO_SHARED_HOME> directory:

`<FEKO_SHARED_HOME>/installedapplicationmacrolibrary/POSTFEKO/CharacteristicModeAnalysis/SynthesisAndDesign/examples`.

The model is set up with a CMA configuration and the following requests:

- A CMA request with 20 modes and the **Compute modal excitation coefficients** check box is selected.
- A far field request with the default 3D pattern.

- A current request calculating all the currents on the structure.

An additional standard configuration with the same far field and current requests is added to compare the weighted sum results with the full method of moments (MoM) solution.

Note: Use the CMA plotter application macro to plot CMA quantities and to determine the dominant modes at the frequency of interest.

Using the Macro

Execute the `CM_synthesis_and_design.lua` application macro in POSTFEKO to calculate the weighted sum for the characteristic modes of interest.

1. Open `antenna_plate.fek` in POSTFEKO.
2. Execute the `CMA_synthesis_and_design.lua` application macro in POSTFEKO.
   The **Characteristic mode synthesis and design** dialog is displayed.

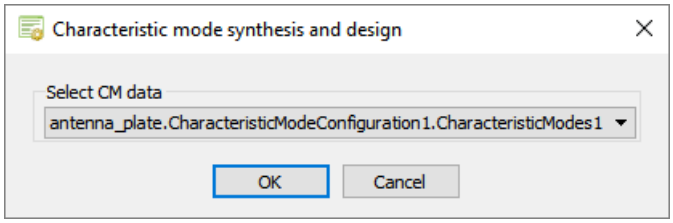


*Figure 658: The **Characteristic mode synthesis and design** dialog.*

3. In the **Select CM data** drop-down list, select **antenna_plate.CharacteristicModeConfiguration1.CharacteristicModes1** and click **OK**.
   The second **Characteristic mode synthesis and design** dialog is displayed.

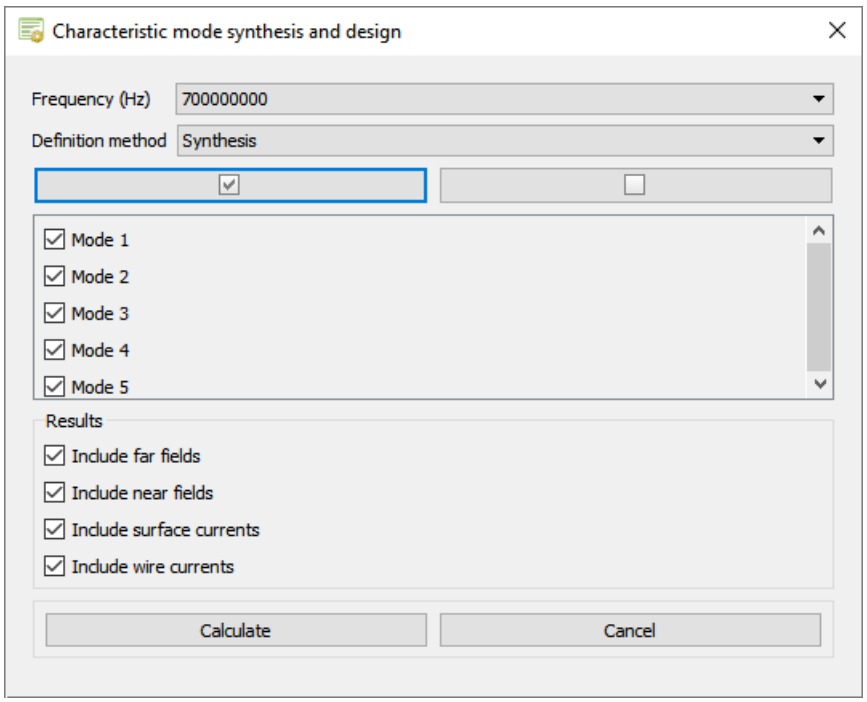


*Figure 659: The **Characteristic mode synthesis and design** dialog.*

4. In the **Frequency (Hz)** drop-down list, select a frequency.
5. In the **Definition method** drop-down list, select one of the following.
   - Select **Synthesis**, to use the modal weighting coefficient as the weight for each mode interest.

     Note: The synthesis mode requires the modal excitation coefficients to be calculated.

   - Select **Design**, to specify the weighting coefficient manually for each mode of interest.
6. Select the modes of interest to calculate the weighted sum.

   Tip: Use the ☑ button to select all modes and the ☐ button to clear all modes.

7. If **Design** is selected, enter the weights for the selected modes, for each mode.
   - Enter a value for the real part of the complex weight **Weight x:Re**.
   - Enter a value for the imaginary part of the complex weight **Im**.
8. Specify the results to calculate a weighted sum for the characteristic modes.
   - Select the **Include far fields** check box, to calculate a weighted sum for the far field data.
   - Select the **Include near fields** check box, to calculate a weighted sum for the near field data.
   - Select the **Include surface currents** check box, to calculate a weighted sum for the surface current data.

- Select the **Include wire currents** check box, to calculate a weighted sum for the wire current data.

9. Click **Calculate** to start the calculation process.
The following dialogs are displayed.

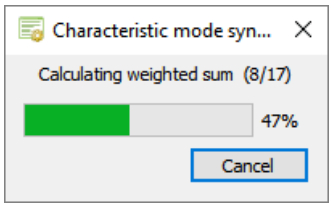


*Figure 660: The **Characteristic mode synthesis and design** (progress) dialog.*

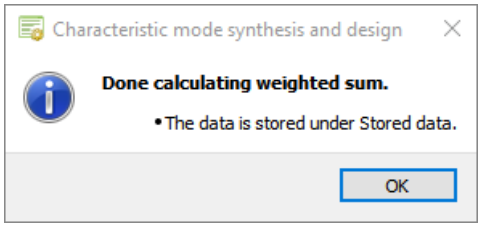


*Figure 661: The **Characteristic mode synthesis and design** (Done) dialog.*

10. View the results under stored data.

> **Note:** The label convention for the synthesis definition methods are:
> - Synthesis: <requestname>_SM1__M20.
> - Design: <requestname>_DM1__M20.

An example using the synthesis definition method, summing modes 1 to 20 at 830 MHz for the antenna plate example. The results for the far field and currents are shown. The full solution (MoM) is compared with the synthesis method.

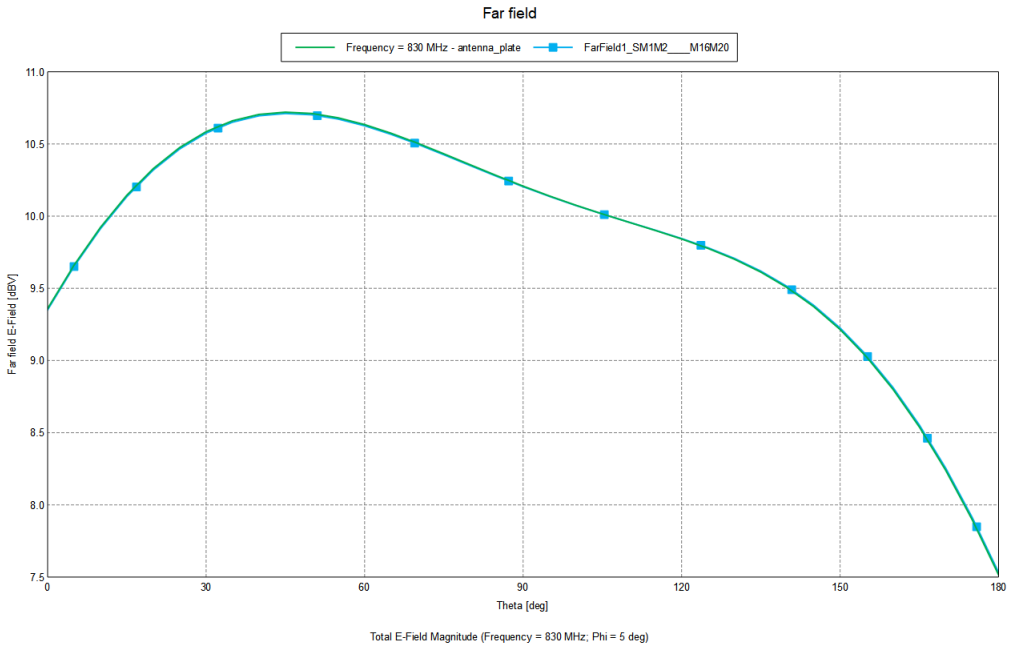


Figure 662: Example comparing the full solution to the synthesis definition method (using modes 1 to 20) for the far field.

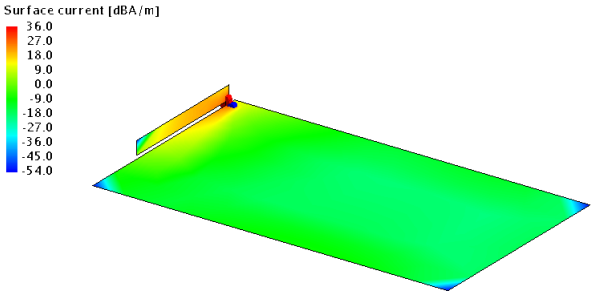


Figure 663: Example of the surface current for the synthesis definition method (using modes 1 to 20).

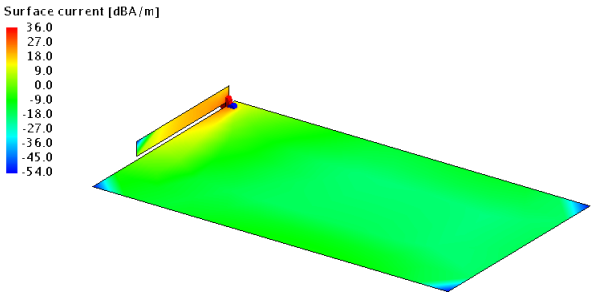


Figure 664: Example of the full solution for the surface current.

9.4.8 Other POSTFEKO Application Macros

A collection of smaller POSTFEKO application macro are available, but these macros do not include step-by-step instructions.

Copy Graph Formatting

This application macro copies selected properties from the reference graph to the selected target graphs. It allows you to change the window caption and graph title with ease. Supported graph types are Cartesian graph, polar graph and Smith chart.

Advanced Field Processing

This application macro calculates simple statistical parameters, for example, median, minimum, maximum, average, cumulative distribution function (CDF) and histogram for far fields and near fields for any given configuration.

Average Gain

This application macro calculates the average gain or average realised gain over frequency or angular range.

Calculate Mixed Mode S-Parameters

This application macro converts 4-port single-ended S-parameters to mixed-mode (differential and common mode) S-parameters and plots the single-ended and mixed-mode S-parameters on a Cartesian graph and Smith chart.

Calculate Z_{in} from Rho

This application macro converts the co-polarised reflection coefficient to an equivalent input impedance relative to the free-space impedance.

Calculate Percentage Gain above Threshold

This application macro calculates the percentage far field samples that are above a specified threshold and displays the total gain that is above or equal to the threshold on a 3D view.

Maximum Coupling from Two Port S-Parameters

This application macro calculates the maximum coupling between two ports from S-parameter data.

Convert S-Parameters to Y and Z-Parameters

This application macro converts the selected S-parameters to Z- and Y-parameters.

RCS Upsampling

This application macro resamples a monostatic radar cross-section (RCS) through half-period interpolation via the fast Fourier transform (FFT).

Filter Near Fields

This application macro filters a near field result based on a specified threshold or range.

Parameter Sweep - Resource Usage

This application macro plots the memory and runtime of permutations of a parameter sweep.

> **Note:** Run Parameter Sweep: Create Models to set up the CADFEKO model.

Energy Density Calculator

This application macro calculates the equivalent energy densities from a near field data set.

Group Delay Calculator

This application macro calculates the group delay from S-parameters for an arbitrary N-port system.

Calculate Skin Depth

This application macro calculates the skin depth from either resistivity or conductivity.

Plot Smith Chart Reference Circle

This application macro to adds a reference trace (circle) on a Smith chart. The resulting trace can also be copied to a Cartesian graph resulting in a straight line on the Cartesian graph.

3D View Synchroniser

This application macro synchronises all 3D views model orientation in the POSTFEKO session to the reference view.

DRE File Import

This application macro imports a data set from a DRE file into POSTFEKO.

DRE File Export

This application macro exports a data set to DRE file format from POSTFEKO.

Import Altair WinProp Trajectory Results

This application macro imports Altair WinProp trajectory results into POSTFEKO. The results are normally from a virtual drive test where one evaluates wireless connectivity along a trajectory, for example, a trajectory of several kilometres length near an LTE base station in a (sub)urban scenario.

Import Altair WinProp Trajectory Results from Measurements

This application macro imports trajectory results from measurements (`.asc`) into POSTFEKO. The results are normally from a virtual drive test where one evaluates wireless connectivity along a trajectory, for example, a trajectory of several kilometres length near an LTE base station in a (sub)urban scenario.

Combine Far Field Equivalent Sources

This application macro combines the receiving antenna results of the far field equivalent sources split over frequency.

Note: Run Create Far Field Equivalent Sources Split Over Frequency to set up the CADFEKO model.

Parameter Sweep: Combine Results

This application macro merges different permutations of a parametric model generated by the associated CADFEKO parameter sweep macro.

Note: Run Parameter Sweep: Create Models to set up the CADFEKO model.

Create an Altair HyperStudy Extraction Script

This application macro generates an Altair HyperStudy extraction script that reads the trace data from a Cartesian graph or polar graph. It requires that the name of the `.pfs` file and Feko model match, and are stored at the same location.

Calculate Wireless Communication Performance

This application macro calculates the effective isotropic radiated power (EIRP), effective isotropic sensitivity (EIS), total radiated power (TRP) and total isotropic sensitivity (TIS) for a given far field request.

Note: Use the **Create Wireless Communication Measurement Configuration** application macro in CADFEKO to create the far field data with.

9.5 Shared Application Macros

A collection of Lua macros are available to automate repetitive tasks where the workflow span both CADFEKO and POSTFEKO.

9.5.1 Convert Between Loss Tangent and Conductivity

It is possible to convert between the loss tangent and conductivity description of the material losses at a single frequency. This application macro performs the calculation and can be used in CADFEKO and POSTFEKO.

Given the frequency, relative permittivity and either the loss tangent or the conductivity of a material, the remaining parameter can be calculated through the relationship

$$\tan\delta = \frac{\sigma}{\omega\varepsilon_0\varepsilon_r} \tag{132}$$

where $\tan\delta$ is the loss tangent, σ is the conductivity, $\omega = 2\pi f$ is the angular frequency, ε_0 is the free space permittivity and, ε_r is the real part of the relative complex permittivity of the medium.

It is often required to convert between these parameters when using the FDTD solver, since frequency-dependent materials are not supported for FDTD.

Using the Application Macro

Execute the application macro in CADFEKO or POSTFEKO to launch the calculator for converting between loss tangent and conductivity.

The application macro is easy to use, and the interface is self-explanatory.

1. Run the application macro from the script editor or add and run it from the macro library in CADFEKO or POSTFEKO.

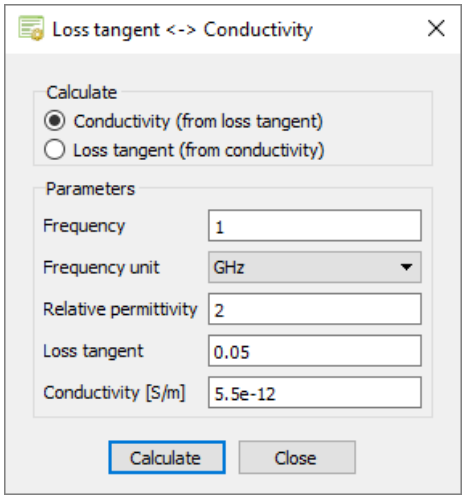


*Figure 665: The **Loss tangent** ↔ **Conductivity** dialog.*

2. Select the conversion of interest. Either calculate the loss tangent from the conductivity or calculate the conductivity from the loss tangent.
3. Enter the frequency and relative permittivity.
4. Enter the loss tangent or conductivity.
5. Click **Calculate** to see the result.
6. Click **Close** to close the dialog.

9.5.2 Farm Model to a Cluster Macro

This application macro is used to divide a larger model with many frequency points, and plane wave sources requested in multiple directions into sub-problems with smaller chunked frequencies and incident directions. The sub-problems are executed concurrently on a cluster reducing the run time to solve the larger problem.

Overview

The application macro is divided into two parts. The first part in CADFEKO creates the models to submit to a cluster. The second part in POSTFEKO merges the results.

1. **Farm model to cluster**: The application macro splits the original CADFEKO model into individual models using the frequency and plane wave source setup per configuration.
2. **Combine results from cluster**: The application macro uses a `.xml` summary file as input to merge the individual runs from the cluster and store the data in POSTFEKO.

Setting up an RCS Sweep Configuration

Set up an RCS sweep configuration. The `farm_model_to_cluster.lua` uses the frequency and incident angle information from plane wave sources in the model to create the sweep.

1.
> **Tip:** Find the examples in the <FEKO_SHARED_HOME> directory:
> `<FEKO_SHARED_HOME>/installedapplicationmacrolibrary/Shared/FarmingMacro/examples`.

Open `RCS_base.cfx` in CADFEKO.
The `RCS_base.cfx` model is displayed.

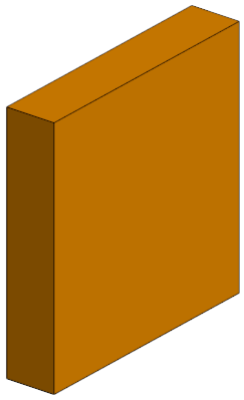


Figure 666: The `RCS_base.cfx` model.

2. In the model tree (Configuration tab), view the frequency options for StandardConfiguration1. The **Solution frequency** dialog is displayed.

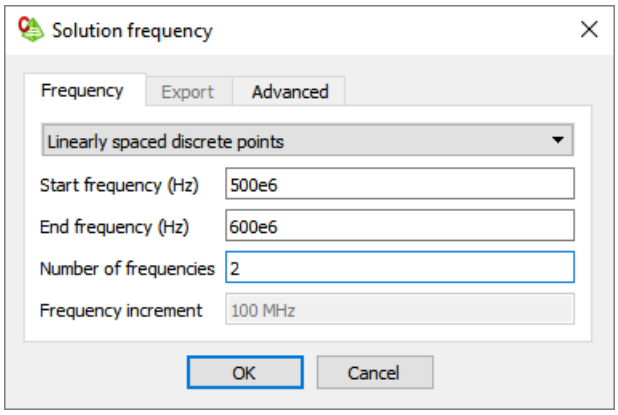
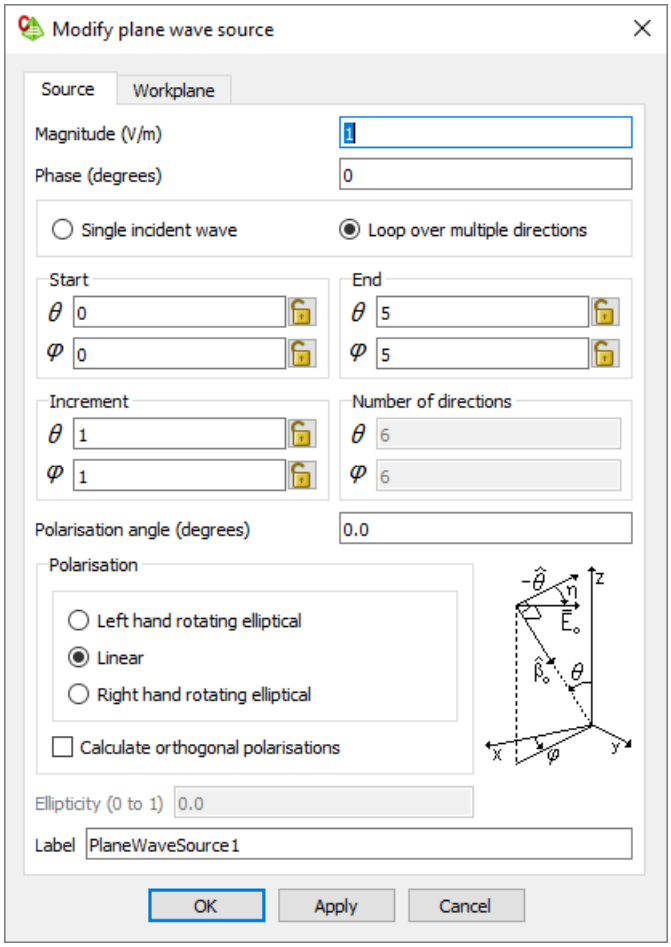


*Figure 667: The **Solution frequency** dialog.*

> **Note:** If the frequency is defined globally, the same sweep setup is used for all the configurations when creating the individual models.

3. On the **Frequency** tab, from the drop-down list, select how the frequency is swept:
   - Select **Single frequency**, if only one frequency point is required.
   - Select **Linearly spaced discrete points**, for a linear sweep between the start and end frequency points.
   - Select **Logarithmically spaced discrete points**, for a logarithmically spaced frequency sweep.
   - Select **List of discrete points**, for a specific set of frequency points for the sweep.
4. Enter the required frequency parameters.
5. Modify the plane wave source in the model.

- Select **Single incident**, if only one angle of incidence is required.
- Select **Loop over multiple directions**, if multiple angles of incidence are required in the sweep.



6. Enter the **Start**, **End** and **Increment** for each angle of interest.
7. [Optional] Specify the **Polarisation angle** or select **Calculate both orthogonal polarisations** check box.
8. Add requests as required, for example, far fields, near fields and currents.
9. [Optional] Add additional configurations, if multiple sweeps for different frequencies and angle if incidents are required. Follow Step 2 to Step 8 for each sweep configuration.
   An additional sweep configuration is added to the model.

> **Note:** For each sweep configuration a subfolder is created in the output folder for the individual models with the configuration name as label.

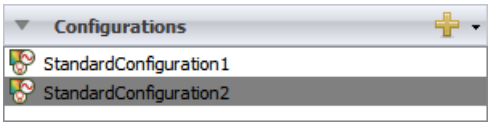


Figure 668: An example of an additional sweep configuration added to the model.

Setting up the PBS Job Template File

The job template file contains a set of instructions for the PBS cluster. This list of commands is used for each sweep configuration when submitting the runs to the cluster.

Note: Before submitting jobs to the cluster, verify the *pbs_job_template.sh* settings with the PBS administrator.

1. Open the `pbs_job_tempalte.sh` file in a text editor (Notepad++ recommended).

   Note: For Windows workstations, set the end of line character (EOL) to **Unix(LF)**.

   The following template is displayed.

```
#!/bin/bash
# Simple example for a job script for using Altair Feko
# with a queuing system (PBS, ...)
# ------------------------------------------------------------
# General settings for PBS
#PBS -l select=2:ncpus=4
#PBS -l walltime=1:30:00
#PBS -k oe
#PBS -V

# General settings for the Altair Feko job

# Point to the Altair Feko installation from the PBS Home directory
cd $PBS_HOME
. ../../opt/altair/2018.2/altair/feko/bin/initfeko
```

2. Change the *#PBS -l select* instruction to the required number of compute nodes and cores to be used per run, for example, *#PBS -l select=2:ncpus=4* indicates that two compute nodes and four cores are used per run.
3. Change the *#PBS -l walltime = Hours:Minutes:Seconds* instruction to the expected runtime of a job.
4. [Optional] Add additional *#PBS* instructions as required, for example, to direct output of error feedback to a custom directory.
5. Change the location from the *$PBS_HOME* directory to point to the Feko installation.
6. Save the file `pbs_job_tempalte.sh` file.

Submitting Jobs to the Cluster

Execute the **Farm model to cluster** application macro in CADFEKO to split a model into smaller runs with one frequency and angles of incidence (phi and theta) to a cluster.

1. Open CADFEKO and run the **Farm model to cluster** macro.
   The **Farm model to cluster** dialog is shown.

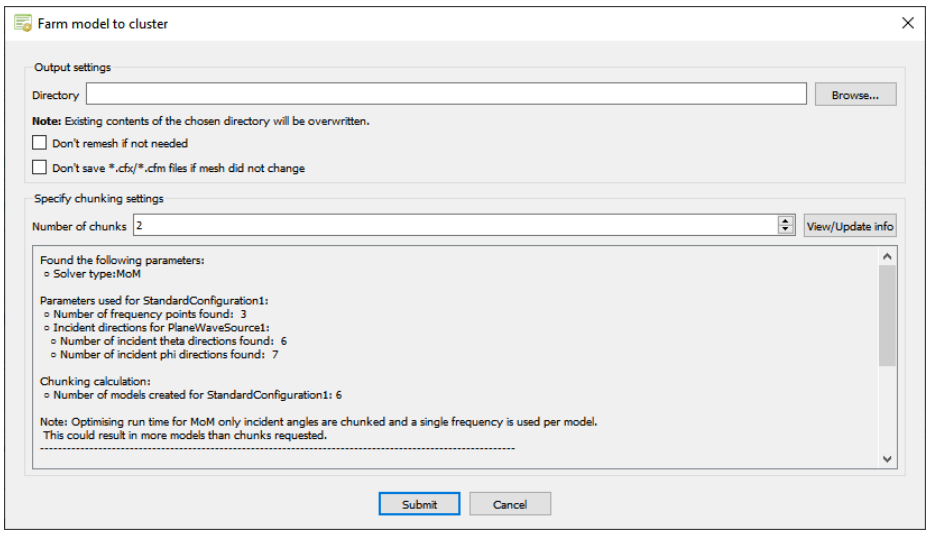


*Figure 669: The **Farm model to cluster** dialog.*

2. Click **Browse** to select the output directory.
3. [Optional] Select the **Don't remesh if not needed** check box to only mesh the model when required.
4. [Optional] Select the **Don't save *.cfx/*.cfm files if mesh did not change** check box to preserve disk space and only create the files that are required.
5. In the **Number of chunks** field, enter the number of chunks to farm to a cluster.
6. Click **View/Update info** to display additional information about the sub-models.

> **Note:** If a plane wave source is included in the configuration only the incident directions are chunked and a single frequency is used per sub-model to optimise the run time.

7. Click **Submit** to start splitting the models to submit the to the cluster.
   Once the model creation has started, a progress dialog is shown.

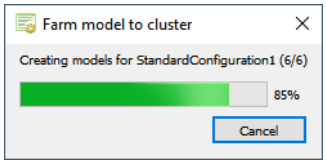


*Figure 670: The **Farm model to cluster** progress dialog.*

8. Complete the process by clicking **OK** to acknowledge the messages.

- The process is finished. The individual models are available in the configuration subfolder in the output directory. A submission file for each sweep configuration is available in the output directory.

Once the models are submitted to the cluster, the following dialog is shown.

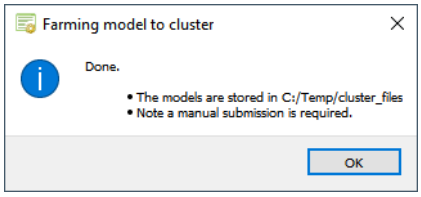


*Figure 671: The **Farming model to cluster** completion dialog.*

Merging the Results Using the GUI

Execute the `combine_results.lua` in POSTFEKO to merge the results completed from the cluster run.

1. Open POSTFEKO and run the `combine_results.lua` macro.
   The **Combine results** dialog is shown.

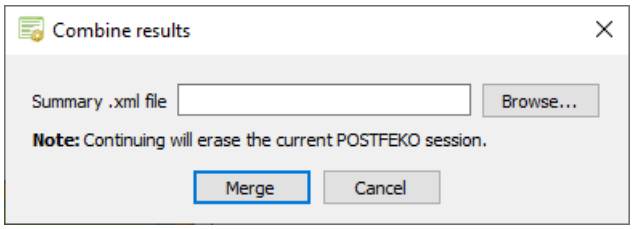


Figure 672: The Combine results dialog.

2. Click **Browse** to navigate to the `.xml` summary file for the configuration(s).
3. Click **Merge** to start a new POSTFEKO session to merge the data.
   The **Combine data** dialog is shown, indicating the merging progress.

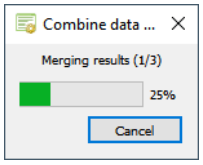


*Figure 673: The **Combine data** dialog.*

4. Complete the process by clicking **OK** to acknowledge the messages.
   - Once the merged data is stored, the following dialog is displayed. Click **OK** to finish the process.

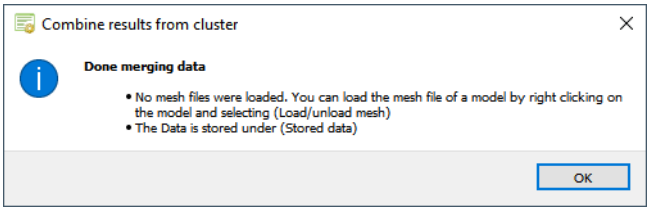


*Figure 674: The **Combine results from cluster** dialog.*

5. View the combined data in POSTFEKO.
   The combined results are saved under **Stored data**.

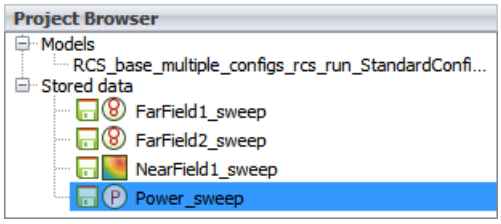


Figure 675: The combined results are available under stored data in the project browser.

Merging the Results on the Cluster

Execute the `combine_results.lua` from the command line using POSTFEKO to merge the results on a cluster. Use the `--configure-script` argument to pass configuration information to the `combine_results.lua` script.

Use the following command-line parameters to launch the `combine_results.lua` script in non-interactive mode:

```
QT_QPA_PLATFORM=offscreen $FEKO_HOME/bin/postfeko [SESSION] --non-interactive
--run-script=[SCRIPT] --configure-script="[OPTIONS]"
```

Note: The *--configure-script* parameter requires that the input variables are wrapped in quotes with an empty space separating each variable.

SESSION
A single session (`.pfs`) may be specified that may or may not exist.

SCRIPT
Specify the absolute path to the `combine_results.lua` file.

Tip: Find the combine script in the <FEKO_SHARED_HOME> directory:
`$FEKO_SHARED_HOME/installedapplicationmacrolibrary/Shared/FarmingMacro`.

OPTIONS

xmlSweepFile
The absolute path to the Cluster sweep summary `*.xml` file on the cluster.

> **Note:** To specify paths for the *--configure-script* variable it is recommended to use literal strings, for example, [[<insert path>]].

Macro Limitations

The farming application macro has limitations on sweep combinations that can be requested for frequency and angle sweeps.

Farm Model to Cluster Macro

- Continuous frequency is not supported.

Combine Results from Cluster Macro

- The application macro only merges results per sweep configuration.

9.5.3 Other Shared Application Macros

A collection of smaller shared application macros are available, but these macros do not include step-by-step instructions.

Transmission Line Calculator

This application macro calculates the physical or electrical properties for a microstrip, stripline or coplaner waveguide transmission line.

Appendices

This chapter covers the following:

EDITFEKO Cards

Each geometry and calculation request are entered on a separate line in the `.pre` and are referred to as cards.

A-1.1 Geometry Cards

Geometry cards are used to create geometry and are placed before the EG card in the `.pre` file.

Table 65: Geometry Cards.

Card	Description
BC	Specifies the outer boundaries of the simulation region.
BL	Creates a line.
BP	Creates a parallelogram.
BQ	Creates a quadrangle.
BT	Creates a triangle.
CB	Changes already assigned labels.
CL	Creates a circular line using segments.
CN	Changes the direction of the normal vector.
DC	Connects a discontinuous mesh and geometry where one is a static mesh and the second is a dynamic parameterised geometry.
DD	Utilises domain decomposition to solve a MoM model more efficiently.
DK	Creates a dielectric or magnetic eighth of a sphere.
DP	Defines a node point.
DZ	Creates a cylindrical dielectric shell.
EG	Defines the end of the geometry.
EL	Creates a segment of an ellipsoid
FA	Defines an antenna array analysis.
FM	Sets parameters related to the MLFMM.
FO	Defines a Fock region.
FP	Sets parameters related to the FEM.
HC	Creates a cylinder with a hyperbolic border.
HE	Creates a coil from wire segments.
HP	Creates a plate with a hyperbolic border.

Card	Description
HY	Creates a hyperboloid section.
IN	Reads an external include file containing mesh information.
IP	Sets the parameter that defines the degree of meshing.
KA	Defines the border of the PO area.
KK	Creates a elliptical conical segment.
KL	Sets the wedges in the PO area.
KR	Creates a planar elliptical element.
KU	Creates a spherical element.
LA	Specifies the label, for example, for segments, triangles and polygonal plates.
ME	Defines the medium.
MB	Defines a modal port boundary condition.
NC	Defines the name for the next configuration.
NU	Creates a NURBS surface from specified control points.
PB	Creates a paraboloid.
PE	Specifies the unit cell that will be used in periodic boundary condition calculations.
PH	Creates a flat plate with an elliptic hole.
PM	Creates a polygonal shape that is meshed into triangles.
PO	Applies the physical optics approximation.
PY	Creates a polygonal plate for use with UTD.
QT	Creates a dielectric or magnetic cuboid (meshed into tetrahedral elements).
QU	Creates a dielectric or magnetic cuboid (meshed into cuboidal elements).
RM	Specified remeshing and adaptive mesh refinement.
SF	Enters a scaling factor, with which all dimensions are multiplied.
SL	Defines for the combined MoM/MTL, the transitioning point from circuit elements (defined in a cable schematic) to the full wave model (defined using DP cards).
SY	Utilises symmetry in the construction of the geometry.

Card	Description
TG	Transformation (for example, translation and rotation) of the geometric structures
TO	Creates a toroid.
TP	Transforms a point.
UT	Parameters for the uniform theory of diffraction (UTD).
UZ	Creates a cylinder for use in the UTD region.
VS	Specifies known visibility information (required when using physical optics with multiple reflections).
WA	Define all active windscreen antenna elements.
WG	Creates a parallelogram consisting of a wire grid.
WR	Defines the dielectric windscreen reference plane.
ZY	Creates a cylindrical element.

** Card

With this card a comment line can be defined whereby all text in this line is ignored.

On the **Home** tab, in the **Edit** group, click the  **Comment** icon.

The following lines of code contains a full comment line as well as comments at the end of lines that define variables.

```
** Definition of parameters
#lambda=1.0  ** Wavelength
#radius=#lambda/2  ** Cylinder radius
#height=2*#lambda  ** Cylinder height
```

BC Card

The BC card is used to define the boundary layers for an FDTD voxel mesh.

On the **Solve/Run** tab, in the **Solution settings** group, click the **Boundary conditions (BC)** icon.

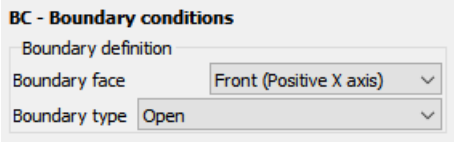


*Figure 676: The **BC - Boundary conditions** dialog.*

Parameters:

Boundary face	This option specifies the bounding box face to be modified, **Top (Positive Z axis)**, **Bottom (Negative Z axis)**, **Right (Positive Y axis)**, **Left (Negative Y axis)**, **Front (Positive X axis)** and **Back (Negative X axis)**.
Boundary type	This option specifies the boundary condition type, **Open**, **Perfect electric conductor (PEC)** and **Perfect magnetic conductor (PMC)**. The open radiating boundary is implemented as a convolutional perfectly matched layer (CPML). The PEC and PMC boundaries allow efficient simulation of infinitely large electrically and magnetically conducting planes.

Related tasks
Specifying the FDTD Boundary Conditions (CADFEKO)

BL Card

The BL card is used to connect two points to form a line, which is then subdivided into wire segments.

On the **Construct** tab, in the **Wires** group, click the **Line (BL)** icon.

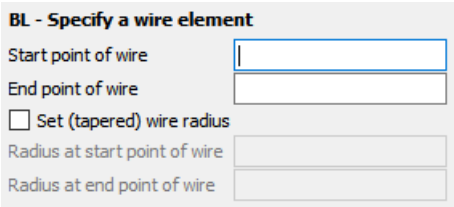


*Figure 677: The **BL - Specify a wire element** dialog.*

Parameters:

Start point of wire	The start point of the wire (previously defined with the DP card).
End point of wire	The end point of the wire (previously defined with the DP card).
Set wire radius	Select the **Set wire radius** check box to override the radius set at the previous IP card for the current wire. This setting does not affect segments created after this card. Both radii values are in metre and are affected by the SF card scaling factor. If only the start radius is specified, the wire will have a constant radius.
Radius at start point of wire	The radius of the wire at the start point.
Radius of end point of wire	The radius of the wire at the end point.

The points have to be defined by a DP card, prior to using this card. The wire radius is set by an IP card preceding the BL card, but can be set locally.

> **Tip:** Create a tapered wire by using different radii values for the start point and end point.

Figure 678: Sketch illustrating the use of the BL card.

Examples of BL Card Usage

- The BL card can be used to create segmented wires. The radius is specified with an IP card.

Figure 679: Example of a segmented wire created with the BL card.

- The BL card can be used to create a tapered and segmented wire:

Figure 680: Example of a tapered radius created with the BL card.

Related reference
Line (CADFEKO)

BP Card

A mesh of surface triangles in the shape of a flat parallelogram can be created with this card. In general, this card is replaced by the PM card. This card should only be used when the user wants to force the very regular meshing that this card produces.

On the **Construct** tab, in the **Surfaces** group, click the **Plate (BP)** icon.

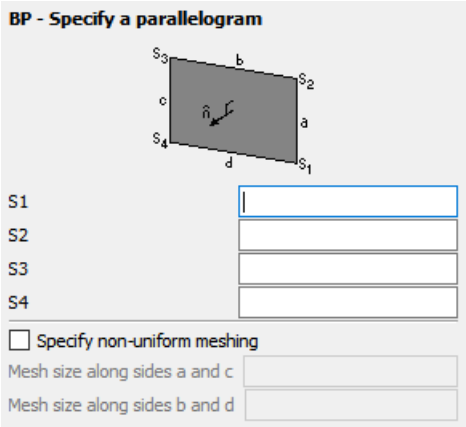


*Figure 681: The **BP - Specify a parallelogram** dialog.*

Parameters:

S1, S2, S3, S4	The points S1 to S4 are the four corner points of the parallelogram. These points should have been defined previously with DP cards.
Specify non-uniform meshing	Normally, a parallelogram is segmented according to the edge length specified with the IP card. When creating small microstrip lines, it may be desirable to use a finer segmentation in one direction. Check this item if a finer segmentation is required in one direction. The mesh sizes are in m and are scaled by the SF card.
Mesh size along sides a and c	Edges S1-S2 and S3-S4
Mesh size along sides b and d	Edges S2-S3 and S4-S1

The points are connected in the order that they appear in the BP card. Thus the user has to ensure that the points describe a parallelogram. If this is not the case, then PREFEKO will abort with the appropriate error message.

The direction of the normal vector of the subdivided triangles is determined by the right hand rule through all corners. This direction is only important when used with physical optics (PO card), dielectrics (ME card), or with the combined field integral equation (CF card).

Example of BP card usage

- The BP card can be used to create a plate with uniform meshing:

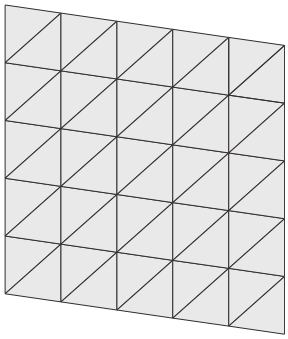


Figure 682: Example a uniformly meshed parallelogram created with the BP card .

- The BP card can be used to create a plane with non-uniform meshing:

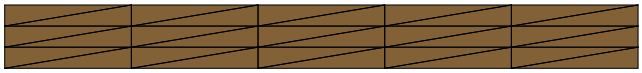


Figure 683: Example a non-uniformly meshed parallelogram created with the BP card.

Related reference
Rectangle (CADFEKO)

BQ Card

This card defines a mesh of surface triangles in the shape of a flat quadrangle.

On the **Construct** tab, in the **Surfaces** group, click the **Quadrangle (BQ)** icon.

> **Note:** For creating planar surface meshes, in general the PM card would be preferred over the BQ card.

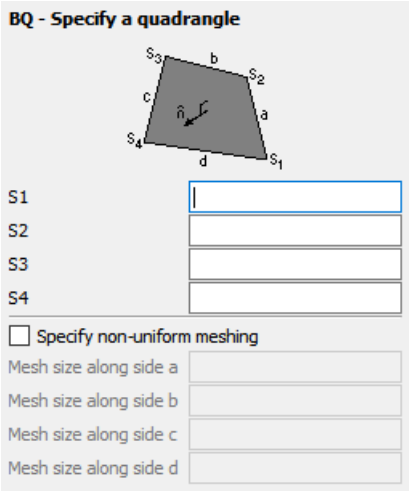


*Figure 684: The **BQ - Specify a quadrangle** dialog.*

Parameters:

S1, S2, S3, S4	The points S1 to S4 are the four corner points of the quadrangle. These points should have been defined previously with the DP card.
Specify non-uniform meshing	Usually a quadrangle is meshed according to the edge length specified with the IP card. When creating, for example, small microstrip lines it may be required to use a finer mesh size in a particular direction. Check this item if finer meshing is required along any edge. The mesh sizes are in metres and are scaled by the SF card.
Mesh size along side a:	The mesh size along edge S1–S2.
Mesh size along side b:	The mesh size along edge S2–S3.
Mesh size along side c:	The mesh size along edge S3–S4.
Mesh size along side d:	The mesh size along edge S4–S1.

The points have to be defined with DP cards before the BQ card and the points are connected in the order that they appear in the BQ card.

In principal the BQ card can create all types of quadrangles, including parallelograms. The difference is that the BP card creates a regular subdivision.

The direction of the normal vector ($\hat{n}$) of the subdivided triangles is determined by the right hand rule through all the corners. This direction is only important when used with the physical optics (PO card) or with dielectrics (ME card) or for the CFIE (CF card).

Examples of BQ card usage

The mesh shown below was created with a BQ card using uniform meshing.

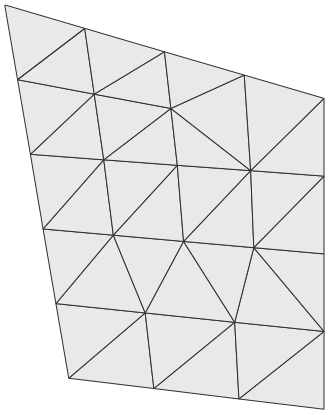


Figure 685: Example of a uniform mesh in the shape of a quadrangle created with the BQ card.

The mesh shown below uses two BQ cards to create a plate with a finely meshed slot.

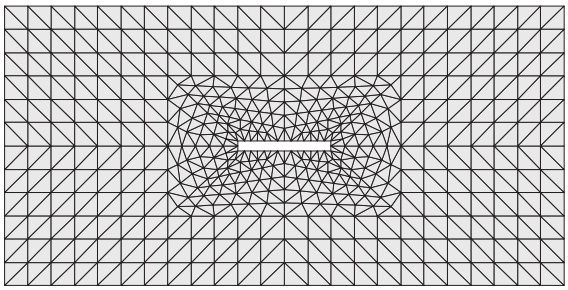


Figure 686: Example of a inhomogeneous mesh created with the BQ card.

Related reference

Polygon (CADFEKO)

BP Card

DP Card

PM Card

QU Card

BT Card

This card defines a mesh of surface triangles in the shape of a flat triangle.

On the **Construct** tab, in the **Surfaces** group, click the **Triangle (BT)** icon.

> **Note:** For creating planar surface meshes, in general the PM card would be preferred over the BT card.

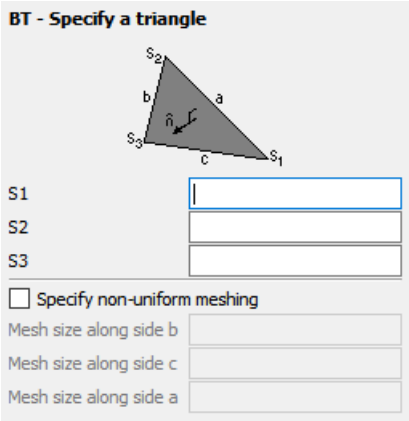


*Figure 687: The **BT - Specify a triangle** dialog.*

Parameters:

S1, S2, S3	The points S1 to S3 are the three corner points of the triangle. These points should have been defined previously with the DP card.
Specify non-uniform meshing	Usually a triangle is meshed according to the edge length specified with the IP card. It may be required to use a finer mesh size in a particular direction. Check this item if finer meshing is required along any edge. The mesh sizes are in metres and are scaled by the SF card.
Mesh size along side b:	The mesh size along edge S2–S3.
Mesh size along side c:	The mesh size along edge S3–S1.
Mesh size along side a:	The mesh size along edge S1–S2.

The direction of the normal vector ($\hat{n}$) of the subdivided triangles is determined by the right hand rule through all the corners. This direction is only important when used with the physical optics (PO card) or with dielectrics (ME card) or for the CFIE (CF card).

Examples of BT card usage

The mesh shown below was created with a BT card using uniform meshing.

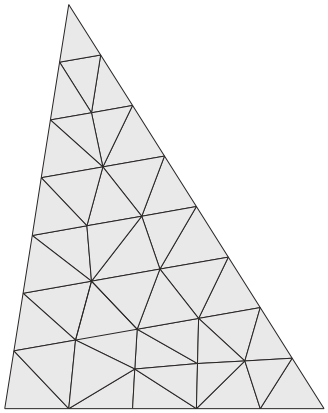


Figure 688: Example of a uniform mesh in the shape of a triangle created with the BT card.

The mesh shown below was created with a BT card using non-uniform meshing.

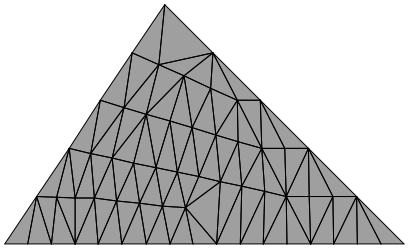


Figure 689: Example of a non-uniform mesh in the shape of a triangle created with the BT card.

Related reference

DP Card
CF Card
IP Card
ME Card
PM Card
PO Card

CB Card

The CB card can be used to change geometry element labels that have been previously defined. Labels that are associated with points, segments, triangles, cuboids, polygonal plates and tetrahedral elements can be updated.

On the **Construct** tab, in the **Modify** group, click the **Change label (CB)** icon.

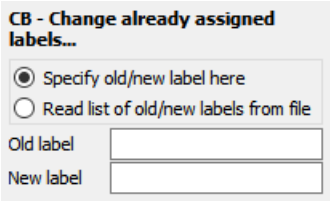


*Figure 690: The **CB - Change already assigned labels** dialog.*

Parameters:

Specify old/new label here	This selection allows to specify an old label and a new label in the corresponding input fields.
Read list of old/new labels from file	This selection allows to read a list of old/new label pairs from an external data file. See further details below.
Old label	All the structures with this label are relabelled.
New label	The new label for all the structures with the old label.
File name	The name of the file used when reading the label list from an external data file.

Renaming labels is especially useful when more labels are created by using symmetry (SY card), transformations (TG card), when importing geometry from files (IN card) or any other properties that would be set by label. Structures created after the CB card are not affected.

In order to make the renaming of a whole set of different labels simpler, the **Old label** field in the CB card is also supporting wild cards "*" (an arbitrary sequence of characters) and "?" (a single arbitrary character). For instance, to rename all these labels

```
Cube.Face1
Cube.Face2
Cube.Face3
Cube.Face4
Cube.Face5
Cube.Face6
```

to a new label `CubeSurface` one could use six CB cards, but with the wild cards this is much simpler to use just one CB card and specify the old label as

```
Cube.Face?
```

or also as

```
Cube.*
```

depending on what other labels are also in the model.

Note that such wild cards are only supported in the **Old label** field of the CB card. The **New label** must be unique.

Another possibility to do a bulk renaming of labels is to read a label mapping table from an external file, which follows the syntax of the ANSA package. In ANSA version 11, this file consists of an arbitrary number of lines

```
old_label | new_label
```

where the old and new label entries are separated by the "|" character. Alternatively, the ANSA version 12 format is also supported, where there is no "|" character to separate the old and new label, but just white space such as a space or tab character. Comment lines are allowed in these files and are indicated using "**" as the first characters of the line.

Some external meshing programs can for instance export a NASTRAN file along with such a mapping table, and then by using the two commands

```
IN   3  3  "geometry.nas"
CB: :  :  :  :  :  :  :  :  :  :  :  :  : "geometry.txt"
```

one can get the model into Feko with the proper names of the parts. In this case, the file `geometry.pre` would then do a proper mapping of the NASTRAN property to the part name in the original CAD program.

Related reference

IN Card

PO Card

SK Card

SY Card

TG Card

CL Card

This card defines an arc consisting of wire segments.

On the **Construct** tab, in the **Wires** group, click the **Elliptic arc (CL)** icon.

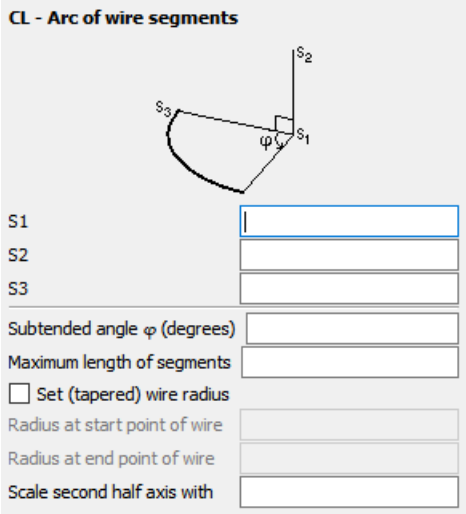


*Figure 691: The **CL - Arc of wire segments** dialog.*

Parameters:

S1	The centre point of the circle.
S2	A point perpendicular to the plane in which the circle lies and above its centre.
S3	The start point of the arc.
Subtended angle φ	The direction of the subtended angle, φ, is in the positive sense around the axis S1–S3. A negative value will create the arc in the opposite direction.
Maximum length of segments	The maximum length of the segments that make up the arc. If this field is left empty, the value specified with the IP card is used. This length is in m and is scaled by the SF card.
Set (tapered) wire radius	If checked, the radius set at the previous IP card is overridden for the current arc. This setting does not affect segments created after this card. Both radius values are in metres and are affected by the SF card scaling factor. If only the start radius is specified, the arc will have a constant wire radius.
Radius at start point of wire	The radius of the wire at the start point.
Radius at end point of wire	The radius of the wire at the end point.

Scale second half axis with	If this parameter is empty or is set to 1, a circular arc is created. If set to $\frac{b}{a}$, an elliptical arc is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where *a* is the distance S1–S3. It is not recommended to generate elliptical arcs with a CAD system if the elliptical arc has an extremely small or extremely large axial ratio as the distortion formulation used in PREFEKO may fail for such cases.

Often modelling the geometry of an arc requires shorter segments than those used for straight wires. Thus the maximum segment length specified with the IP card can be overridden along the arc by specifying a value in the **Maximum length of segments** text box.

The radius of the arc is given by the distance between the points S1 and S3.

Examples of CL card usage:

The meshes depicted in the two figures below are created with the CL card. The first figure shows the result of a wire arc with a uniform radius, and the second figure shows the result with an exaggerated taper specified.

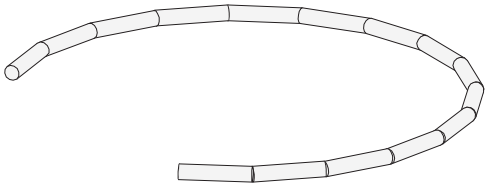


Figure 692: Example of an arc with uniform wire radius created with a CL card.

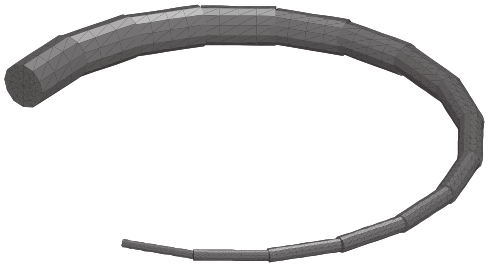


Figure 693: Example of an arc with tapered wire radius created with the CL card.

Related reference

IP Card

NW Card

PR Card

SF Card

CN Card

This card is used to reverse the normal direction of previously created triangles or polygonal plates, for example after importing CAD data.

On the **Construct** tab, in the **Modify** group, click the **Reverse normals (CN)** icon.

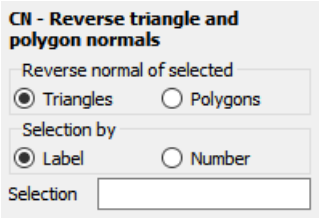


*Figure 694: The **CN - Reverse triangle and polygon normals** dialog.*

Parameters:

Reverse normal of selected	In this group, the user selects to reverse the normals of either **Triangles** or **Polygons**.
Selection by	Here, the user specifies whether the triangles or polygonal plates are identified by their **Label** or absolute element **Number**.
Selection	The label or element number of the triangles or polygonal plates that are to have their normals reversed.

The normal direction can be important, such as when defining dielectric surfaces. For triangles, the normal vector is reversed by interchanging corners 1 and 3. For polygonal plates, the first point remains as is but the corner points are listed in the opposite direction.

The CN card changes the normal of the affected triangles but it does not change the settings of the ME card (which medium is on which side of the triangle is determined by the normal vector). For example, if the ME card is used to specify that the normal vectors of the triangles point from medium 5 to medium 2 then the application of the CN card will effectively change which medium lies on which physical side of the triangle.

Related tasks
Reversing Face Normals (CADFEKO)
Related reference
ME Card

DC Card

Domain connectivity (discontinuous Galerkin domain decomposition method) can be used to connect a discontinuous mesh and geometry where one is a static mesh (for example, a large imported car model) and the second is dynamic parameterized geometry (for example, a small antenna).

On the **Solve/Run** tab, in the **Solution settings** group, click the **Domain connectivity (DC)** icon.

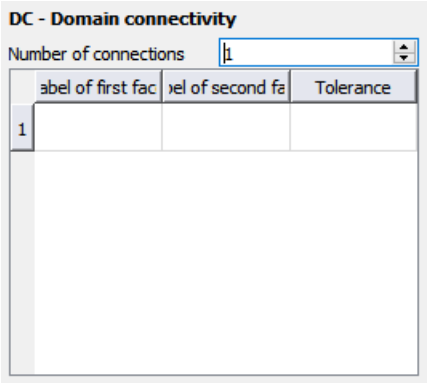


*Figure 695: The **DC - Domain connectivity** dialog.*

Parameters:

Number of connections	Specify the number of connections for the two meshes.
Label of first/second face	Define for each connection, the labels of the corresponding faces.
Tolerance	Define for each connection point, a tolerance distance to distinguish between regions, where a gap in the model is desired (or not desired).

Requirements for Domain Connectivity

The following boundary conditions for the application of the method have to be considered:

- Gap distance *g* should be about λ/1000 and must not be greater than λ/100.
- The meshes may not penetrate each other.
- The domain connectivity edges may not be part of a dielectric region. However, dielectrics can be defined elsewhere in the model.
- For T-junctions the domain connectivity-cut may not be through the junction.
- Domain connectivity can be applied for MoM and MLFMM, but not for asymptotic methods like PO, LE-PO, RL-GO and UTD.

Related concepts

Discontinuous Mesh and Geometry Parts

Workflow for Connecting Discontinuous Mesh and Geometry Parts

Domain Connectivity

Related tasks

Defining Domain Connectivity

DD Card

Domain decomposition can be used to store parts of a method of moments solution in a file to be reused in future simulations. The stored part must remain static but the rest of the model can change.

On the **Solve/Run** tab, in the **Solution settings** group, click the **Domains (DD)** icon.

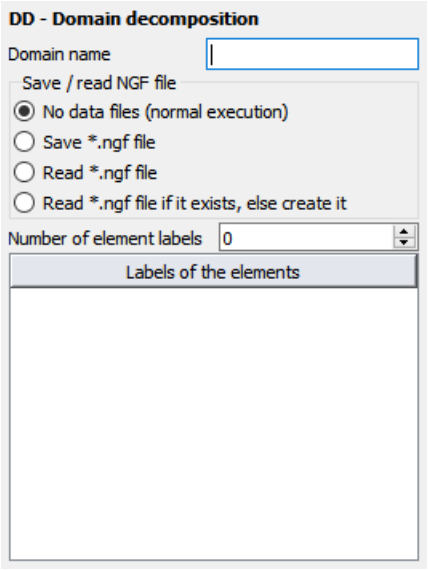


*Figure 696: The **DD - Domain decomposition** dialog.*

Parameters:

No data files (normal execution)	When this option is selected, no data files are read or saved to a `.ngf` file.
Save *.ngf file	The results are saved to a `.ngf` file.
Read the *.ngf file	The results are read from a `.ngf` file.
Read *.ngf file if it exists, else create it	The `.ngf` file is read if it exists, else a `.ngf` file is created containing the results.
Number of element labels	The number of element labels to be included in the fixed static part.
Labels of the elements	The labels of the elements which are to be defined as the fixed static part in the model.

Note: Only a single DD card is permitted in a model.

Related tasks
Using NGF to Reduce CPU Time (CADFEKO)
Related reference
PS Card

DK Card

This card defines an eighth of a sphere, meshed into cuboidal elements, for solving with the volume equivalence principle in the MoM.

On the **Construct** tab, in the **Volumes** group, click the **Sphere (DK)** icon.

The meshing parameters are set with the IP card, and the medium, as set at the ME card, is assigned to all created cuboidal elements.

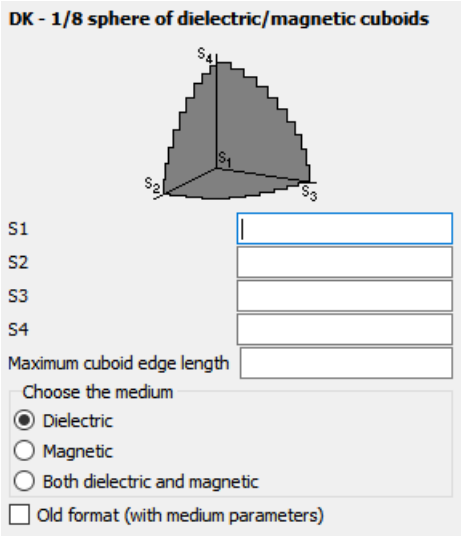


*Figure 697: The **DK - 1/8 sphere of dielectric/magnetic cuboids** dialog.*

Parameters:

S1	The centre of the sphere.
S2, S3, S4	Specify the three directions S1–S2, S1–S3 and S1–S4, that form the border of the eighth of the sphere. They must be perpendicular to each other and all three must have the same length (the sphere's radius).
Maximum cuboid edge length	The maximum side length of cuboids along the curved edge (in metres) can be specified. This value is scaled by the SF card. If left empty, the value specified with the IP card is used.
Choose the medium	Select whether the sphere is **Dielectric** or **Magnetic** or **Both dielectric and magnetic**. This is always with respect to the environment. For example if the relative permittivity ε_r of the cuboid material differs from the environment then this is a dielectric sphere.
Old format (with medium parameters)	For a detailed description of this parameter please see the QU card.

Dielectric bodies treated with the volume equivalence principle (using cuboids) cannot be used simultaneously with dielectric bodies treated with the surface equivalence principle or the FEM or with special Green's functions.

Example of DK card usage

The DK card can be used to create a mesh of an eighth of a sphere as shown below.

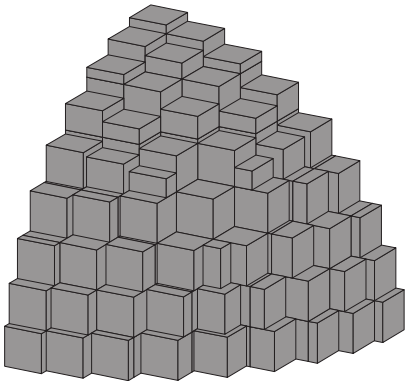


Figure 698: Example of an eighth of a sphere created with the DK card.

Related reference

IP Card

ME Card

QU Card

SF Card

DP Card

The DP card is used to define points in space. These points are used to define the extent and orientation of other geometric entities and to locate excitations.

On the **Home** tab, in the **Define** group, click the **Point (DP)** icon.

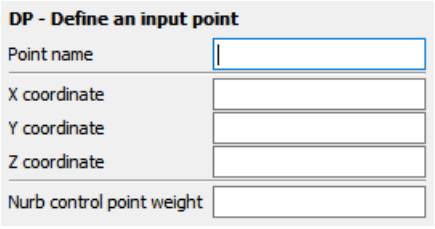


*Figure 699: The **DP - Define an input point** dialog*

To avoid ambiguity each point is assigned a name (a 5 character string if column based format is used.) In other cards (for example the BL card) the points are referred to by their names.

Parameters:

Point name	Name of the point.
X coordinate	X coordinate of the point in m (is scaled by the SF card).
Y coordinate	Y coordinate of the point in m (is scaled by the SF card).
Z coordinate	Z coordinate of the point in m (is scaled by the SF card).
Nurb control point weight	The weight of the control point when this point is used with the NU card (NURBS surfaces). If the field is empty, it defaults to 1.

Examples of DP Card Usage

In addition to its coordinates, each point is also assigned the current label (see LA card), so that a group of points can be selected by label, for example when moving points with the TP card. Point names may use the characters a-z, A-Z, 0-9 and the special character _ and no distinction is made between upper and lower case characters. Thus *P1a* and *p1A* refers to the same point. In addition, when defining or using node names, simple variable names of the form A#i are allowed. The algorithm is that if a hash sign is found in a node point name, this hash sign and everything that follows is interpreted as a variable string, evaluated and rounded to the nearest integer. Thus if we have *#k=15* and use or define a point *P#k* then this is equivalent to using *P15* as point name.

```
[node!variable name]
```

For instance, it would now be possible to define the points P1 to P20 inside a loop.

```
!!for #k = 1 to 20
DP: P#k
!!next
```

Unlike most other geometry cards, the DP card (as well as the TP card) may also be used in the control section (after the EG card) of the `.pre` file. This allows defining the points required by the AP card in this part of the file.

Related tasks

Defining a Named Point (CADFEKO)

Related reference

BL Card

EG Card

LA Card

NU Card

SF Card

TP Card

DZ Card

The DZ card is used to create a cylindrical shell, meshed into cuboidal elements for using the volume equivalence principle in the method of moments. The meshing parameters as set at the IP card are used, and the medium as set at the ME card is assigned to all created cuboidal elements.

On the **Construct** tab, in the **Volumes** group, click the **Cylinder (DZ)** icon.

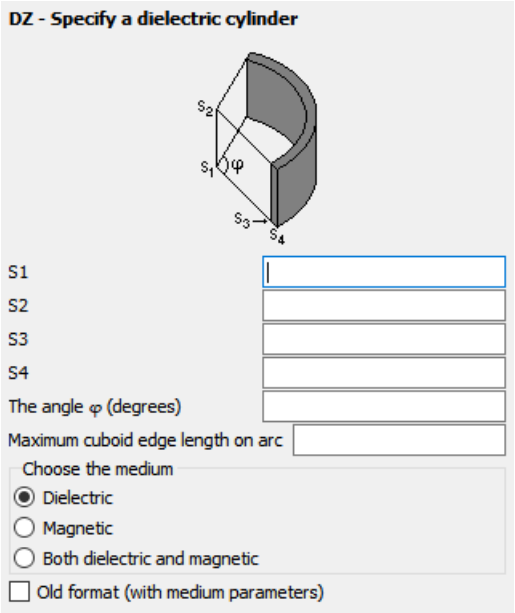


*Figure 700: The **DZ - Specify a dielectric cylinder** dialog*

Parameters:

S_1	The start point of the cylinder axis.
S_2	The end point of the cylinder axis.
S_3	Point on the inside of the shell.
S_4	Point on the outside of the shell.
The angle φ	The angle of the cylindrical segment in degrees.
Maximum cuboid edge length on arc	Maximum edge length of the cuboids along the arc in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.
Choose the medium	Select here whether the cylindrical shell is dielectric or magnetic or both (this is always with respect to the environment, for example if the relative permittivity ε_r of the cuboid material differs from the environment, then this is a dielectric cylinder).
Old format (with medium parameters)	For a detailed description of this parameter please see the QU card.

Dielectric bodies treated with the volume equivalence principle (using cuboids) cannot be used simultaneously with dielectric bodies treated with the surface equivalence principle or the FEM or with special Green's functions.

Example of DZ card usage

Below is an example of a cylindrical shell created with the DZ card.

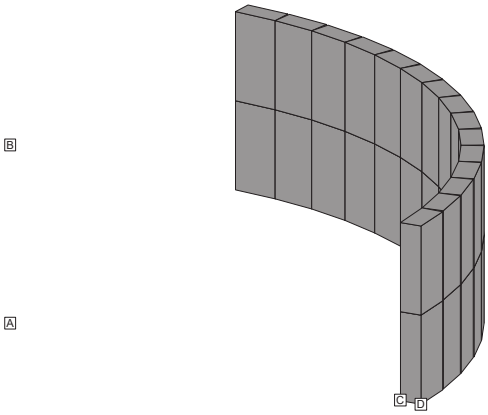


Figure 701: Example of an cylindrical shell created with the DZ card.

Related reference

IP Card

ME Card

QU Card

SF Card

EG Card

The EG card indicates the end of the geometrical input. It is essential that the EG card is used.

On the **Home** tab, in the **Structure** group, click the **End geometry (EG)** icon.

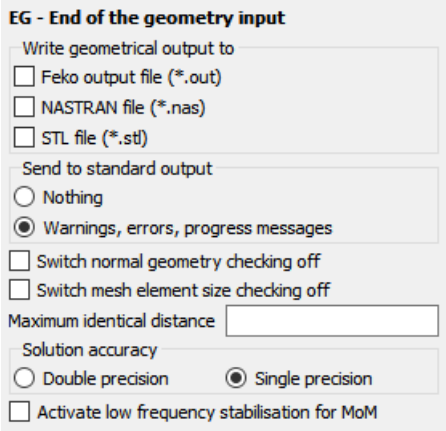


*Figure 702: The **EG - End of the geometry input** dialog*

Parameters:

Write geometrical output to
The geometry data of the segments and surface elements can be written to the Feko output file, a NASTRAN format file, an STL file, or any combination thereof. The name of the NASTRAN or STL file will be the same as the Feko model, but with a `.nas` or `.stl` extension. Writing the geometry data to the output file may lead to huge output files.

Send to standard output
If the field **Nothing** is selected, no messages are sent to the standard output device (usually the screen). If the item **Warnings, errors, progress messages** is selected, warnings, errors and messages that indicate the program's progress are sent to the standard output device.

Switch normal geometry checking off
If this item is checked, the verification of the geometry elements Feko will be switched off (see the discussion below this table).

Switch mesh element size checking off
If this item is checked, the verification of the mesh elements size in relation to the frequency will be switched off

Solution accuracy
This parameter can be set to force Feko to use single precision for the storage of the memory critical arrays. Single precision storage is the default behaviour, and as compared to double precision the memory requirement is then half. Using double precision is recommended when the Feko kernel gives a warning to switch to double precision (this might happen for instance at low frequencies where an increased accuracy is required).

Activate low frequency stabilisation for MoM

At very low frequencies (frequency range where the largest dimension of the model is much smaller than a wavelength), the method of moments (MoM) solution can become numerically unstable and singular.

The default MoM solution uses single precision. When using double precision, the MoM solution is valid for lower frequencies than for single precision. If the solver gives a warning about the matrix stability when using double precision, then it is recommended to use low-frequency stabilisation.

If this item is selected, low frequency stabilisation for MoM is activated.

> **Note:** Low-frequency stabilisation is not required at higher frequencies and increases the runtime. Double precision uses double the memory of single precision.

Auto	The Solver determines automatically if low frequency stabilisation should be used for the model.
Always on	Enable low frequency stabilisation for the model.

The following should be noted regarding the export of the Feko geometry to NASTRAN or STL:

- The STL export just dumps the data of all triangular patches of the Feko model to an ASCII formatted STL file. Any other geometry (for example wires, tetrahedra) is not exported since the STL format does not make provision for this. It should also be noted that the Feko mesh does not contain any information about the geometry that the triangle mesh elements were created from. Thus there is no special grouping of elements based on regions or solid parts in the STL file. This implies that the exported STL file does not represent a valid STL file in the strict sense. However, the exported information is still useful in most cases.
- For the NASTRAN export, the wide column format is used to ensure that all significant digits are exported. Unlike the STL export, in NASTRAN all the various mesh elements used in Feko are present. However, information is lost, for instance for wire elements the thickness (wire radius) can not be exported simply because the NASTRAN file format does not make provision for this. Also the NASTRAN property is used to represent the Feko label. But since NASTRAN properties are just integer values and the Feko label can be an arbitrary string, a mapping is done so that from each Feko label just the associated number is used and exported as the NASTRAN property.

The **Maximum identical distance** is used to set the tolerance in the mesh. The mesh information is created by the program PREFEKO, and stored in a `.pre` file, in which all the triangles and segments are described by their corner points. Due to rounding errors it is possible that, for example, end points of connecting segments do not coincide. When searching for nodes, an ohmic connection is made when the difference is smaller than the **Maximum identical distance**.

Feko automatically checks for typical user errors that have been observed in the past. Examples of errors are connecting a wire segment to the middle of another wire, where the connection points do not coincide, or connecting surfaces that have different segmentation along the common edge. Such errors are detected if the parameter Switch normal geometry checking off is unchecked. The error detection routine should always be used. However, if the same geometry is to be used a number of times, the error detection can be disabled by checking this item.

If the surrounding medium is not vacuum, one can set the material parameters with the EG card as shown above. Alternatively the parameters of the surrounding medium can be set with the GF card which offers greater flexibility. For example, the GF card can be used to set the material parameters (as an arbitrary function of frequency) inside a frequency loop which is not possible with the EG card.

Related reference
GF Card

EL Card

A mesh of surface triangles in the shape of an ellipsoidal section is created with the EL card.

On the **Construct** tab, in the **Surfaces** group, click the **Ellipsoid (EL)** icon.

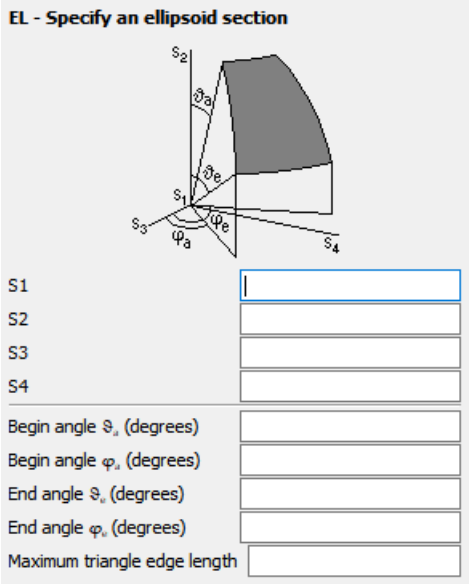


*Figure 703: The **EL - Specify an ellipsoid section** dialog.*

Parameters:

S1 — The centre point of the ellipsoid.

S2 — A point, in the direction $\vartheta = 0^\circ$ in elliptical coordinates. The distance of the two points S1 and S2 determines the half-axis of the ellipsoid in this direction.

S3 — A point in the direction $\vartheta = 90^\circ$, $\varphi = 0^\circ$ in elliptical coordinates. The distance of the two points S1 and S3 determines half of the axis of the ellipsoid in this direction.

S4 — A point in the direction of the third coordinate, for example, the axes S4–S1, S3–S1 and S2–S1 must be perpendicular. The distance of the two points S1 and S4 determines half of the axis of the ellipsoid in this direction.

Begin angle ϑ_a — Start angle of the ellipsoid in degrees.

Begin angle φ_a — Start angle of the ellipsoid in degrees.

End angle ϑ_e — End angle of the ellipsoid in degrees.

End angle φ_e — End angle of the ellipsoid in degrees.

Maximum triangle edge length	Maximum length of the triangles along the curved edge in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Note that the angles ϑ and φ are defined in an elliptical, rather than a spherical coordinate system. For a Cartesian coordinate system with origin S1, x axis in direction of S3, y axis in the direction of S4 and z axis in the direction of S2, a point *r* on the surface of the ellipsoid is given as

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a\sin(\vartheta)\cos(\varphi) \\ b\sin(\vartheta)\sin(\varphi) \\ c\cos(\vartheta) \end{bmatrix} \tag{133}$$

where the lengths *a*, *b* and *c* are the lengths of the ellipsoid's three half-axes. (For example the length *a* is the distance between the points S3 and S1).

The normal vector of the generated triangles always points outwards. The algorithm used for the segmentation can fail if the ratio of the half-axis is too extreme, for example if the longest half-axis is a factor 100 longer than the shortest. It is strongly advised to check the geometry with POSTFEKO.

Example of EL card usage

The mesh shown in Figure 704 is generated by using the EL card.

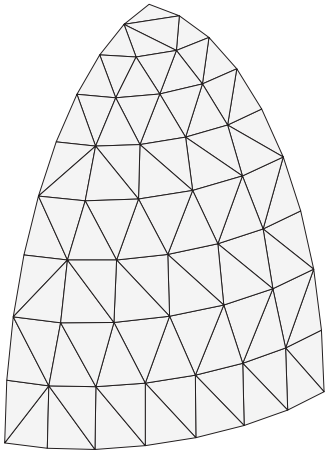


Figure 704: Example of an eighth of an ellipsoid created with the EL card.

Related reference

IP Card

SF Card

FA Card

This card is used to define a finite antenna array which includes mutual coupling and edge-effects.

On the **Home** tab, in the **Planes / arrays** group, click the **Array (FA)** icon.

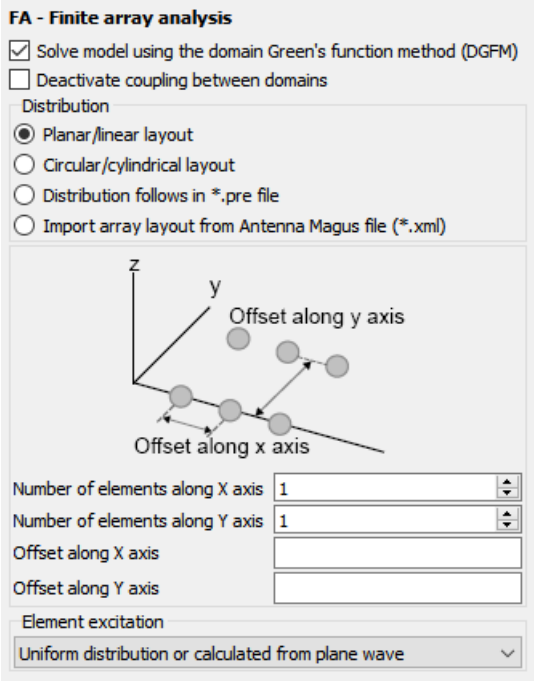


*Figure 705: The **FA - Finite array analysis** dialog, with **Planar/linear layout** selected.*

Parameters:

Solve model using the domain Green's function method (DGFM)	If this option is selected, the DGFM will be used in the analysis of the finite antenna array. Not checking this option will result in the creation of the geometry and excitations only.
Deactivate coupling between domains	If this option is checked, mutual coupling will not be considered in the DGFM.

Planar / Linear Layout

This option defines the finite antenna array with a planar or linear distribution.

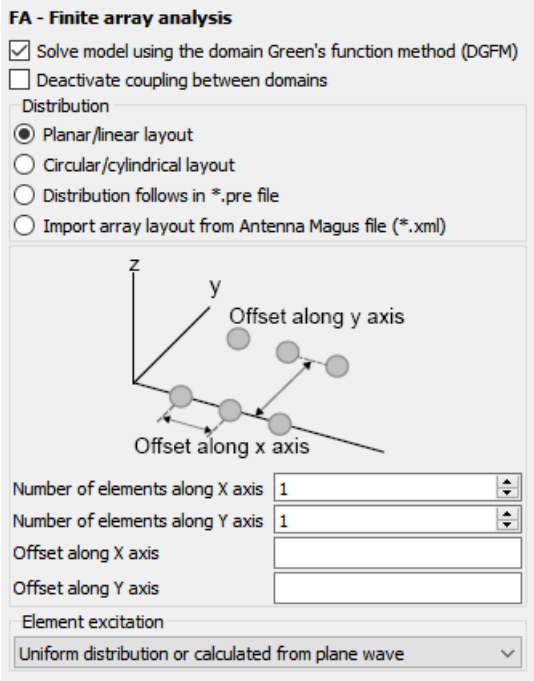


*Figure 706: The **FA - Finite array analysis** dialog, with **Planar/linear layout** selected.*

Parameters:

Number of elements along X axis	The number of finite antenna array elements along the X axis.
Number of elements along Y axis	The number of finite antenna array elements along the Y axis.
Offset along X axis	The distance between the finite antenna array elements along the X axis.
Offset along Y axis	The distance between the finite antenna array elements along the Y axis.
Element excitation	If the **Uniform distribution or calculated from plane wave** option is selected, the array elements will either have a uniform distribution or the distribution will be calculated from the plane wave if a plane wave is present in the model. If the **Specify amplitude and phase for each element** option is selected, the user can specify the excitation for each element.
Magnitude scaling	The excitation magnitude for the respective element is scaled relative to the base element.

Phase offset (degrees)	The phase offset in degrees for the respective element relative to the base element.

Related tasks
Creating a Linear / Planar Antenna Array (CADFEKO)

Circular / Cylindrical Layout

This option defines the finite antenna array with a circular or cylindrical distribution.

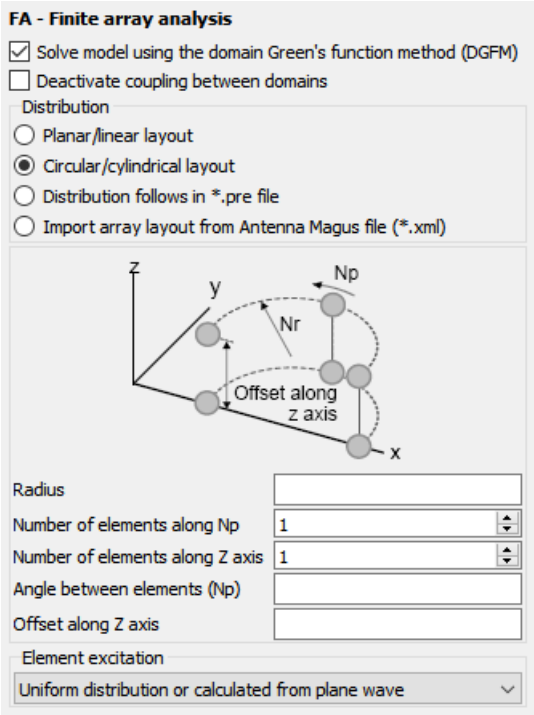


*Figure 707: The **FA - Finite array analysis** dialog, with **Circular/cylindrical layout** selected.*

Parameters:

Radius	The radius of the circular/cylindrical antenna array.
Number of elements along Np	The number of finite antenna array elements along the Np/ϕ direction.
Number of elements along Z axis	The number of finite antenna array elements along the Z axis.
Angle between elements (Np)	The angle between the respective finite antenna array elements.
Offset along Z axis	The distance between the finite antenna array elements along the Z axis.
Element excitation	If the **Uniform distribution or calculated from plane wave** option is selected, the array elements will either have a uniform

	distribution or the distribution will be calculated from the plane wave if a plane wave is present in the model. If the **Specify amplitude and phase for each element** option is selected, the user can specify the excitation for each element.
Magnitude scaling	The excitation magnitude for the respective element is scaled relative to the base element.
Phase offset (degrees)	The phase offset in degrees for the respective element relative to the base element.

Related tasks
Creating a Cylindrical / Circular Antenna Array (CADFEKO)

Distribution Follows in *.pre File

This option defines the finite antenna array by specifying the distribution and excitation for each individual element directly in the .pre file.

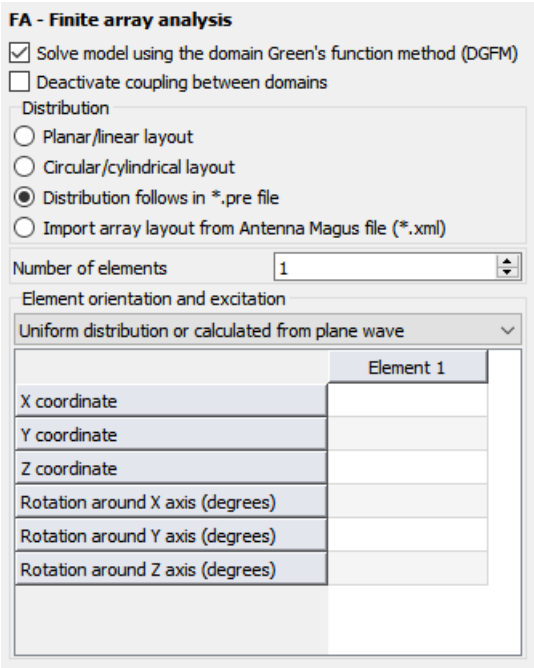


*Figure 708: The **FA - Finite array analysis** dialog, with **Distribution follows in *.pre file** selected.*

Parameters:

Number of elements	The number of antenna array elements specified in the antenna array.
X, Y, Z coordinate	Cartesian coordinates for the location of the respective antenna array elements.

Rotation around X axis (degrees)	Angle of rotation around the X axis in degrees.
Rotation around Y axis (degrees)	Angle of rotation around the Y axis in degrees.
Rotation around Z axis (degrees)	Angle of rotation around the Z axis in degrees.
Element excitation	If the **Uniform distribution or calculated from plane wave** option is selected, the array elements will either have a uniform distribution or the distribution will be calculated from the plane wave if a plane wave is present in the model. If the **Specify amplitude and phase for each element** option is selected, the user can specify the excitation for each element.
Magnitude scaling	The excitation magnitude for the respective element is scaled relative to the base element.
Phase offset (degrees)	The phase offset in degrees for the respective element relative to the base element.

Import Array Layout from Antenna Magus File (*.xml)

This option imports the finite antenna array from an Antenna Magus file (.xml).

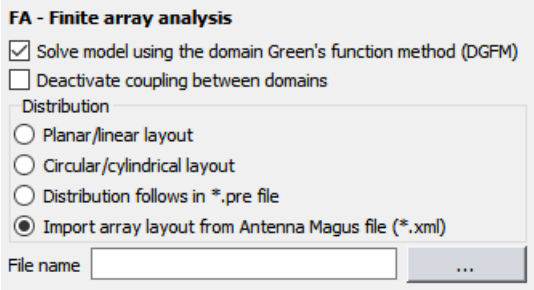


*Figure 709: The **FA - Finite array analysis** dialog, with **Import array layout from Antenna Magus (*.xml)** selected.*

Parameters:

File name	The file name of the Antenna Magus file from which the antenna array is to be imported.

Finite Antenna Array Limitations

A finite antenna array or DGFM is not applicable to all models, but can significantly improve simulation speeds when it can be applied.

- The DGFM is supported for MoM examples containing metallic triangles, wires and connections between them. Skin effect and dielectric / magnetic coatings are supported for metallic surfaces. Coatings and discrete loads are supported for wires.
- Only a single FA card is allowed in the model.
- The following is not allowed in conjunction with the DGFM:
  - General network, transmission line, network load or a cable load
  - VEP, FEM or FDTD solution method

Note: Interconnected or very closely coupled domains with a spacing less than $\frac{\lambda}{10}$ between array elements are not allowed.

FM Card

The FM card is used to instruct the Feko solver to calculate the solution using accelerated methods, for example, using the multilevel fast multipole method (MLFMM) or adaptive cross-approximation (ACA). An option is available to apply compression to looped plane wave sources.

On the **Solve/Run** tab, in the **Solution settings** group, click the **MLFMM (FM)** icon.

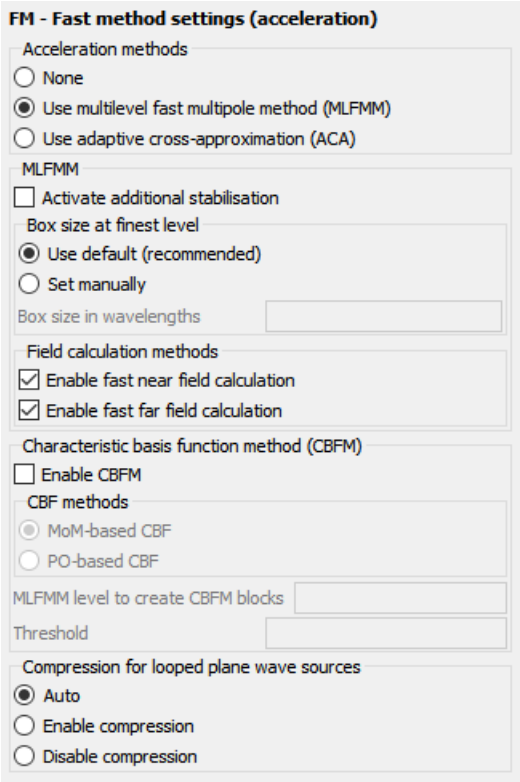


*Figure 710: The **FM- Fast method settings (acceleration)** dialog.*

Parameters:

None	Neither the MLFMM nor the ACA is activated.
Use multilevel fast multipole method (MLFMM)	The MLFMM is used instead of the standard method of moments (MoM) to calculate the solution on all structures.
Use adaptive cross-approximation (ACA)	The ACA method is used to calculate the solution. This method approximates the MoM impedance matrix by constructing a sparse H-matrix (only a few selected elements are computed). This method is applicable to low frequency problems or when using a special Green's function.
Activate additional stabilisation	Activate additional stabilisation for the MLFMM to address models with severe convergence problems.
Box size at finest level	The MLFMM is based on a hierarchical tree-based grouping algorithm, and depending on the frequency and the model

dimensions Feko automatically determines the number of levels in this tree and the size of the boxes at the finest level. It is also recommended that this default box size of 0.23λ is kept. When there is no convergence in the MLFMM, then advanced users might try to slightly increase or decrease this box size by setting it manually (the input is in terms of the wavelength).

Field calculation methods

The Solver determines automatically which calculation method to use for the fastest field calculation.

Enable fast far field calculation
If selected, the MLFMM method is included in the available options for calculating the far field (default). If not selected, the Solver uses either traditional integration or other fast methods to calculate the far field.

Enable fast near field calculation
If selected, the MLFMM method is included in the available options for calculating the near field (default). If not selected, the Solver uses either traditional integration or other fast methods to calculate the near field.

Characteristic basis function method (CBFM)

Enable CBFM
Enable characteristic basis function methods (CBFM) to reduce the number of unknowns (which reduces the memory and runtime).

MoM-based CBFM
Use the MoM solution method to generate the CBFMs. Solution is more accurate, but at the cost of an increase in runtime and memory.

PO-based CBF
Use the PO solution method to generate the CBFMs. Solution is less accurate, but with a shortened runtime and a decrease in memory.

Block size in wavelegths
Size of the blocks in which the characteristic basis functions (CBFs) are defined. A larger size requires more processing

		time, but yields a higher reduction of the number of unknowns. The box size can be a value between 0.5 and 2, where 1 is the recommended size.
	MLFMM level to create CBFM blocks	It defines the MLFMM level that will be used to create the boxing. These boxes will determine the size of the blocks in which the characteristic basis functions (CBFs) are defined where the finest level is 1.
	Threshold	Parameter used to discard less significant CBFMs, between 0 and 1. A higher value discards more CBFMs, with a better reduction of the number of unknowns. The accuracy may be affected for high threshold values. The typical threshold value is about 5e-4.

Compression for looped plane wave sources

Compression of a plane wave loop over direction of incidence can be used to speed up calculations over multiple directions of incidence (for example, when calculating RCS).

Auto	The Solver determines automatically if compression should be used for the model (if the method is likely to speed up the solution).
Enable compression	Enable compression for looped plane wave sources in the model.
Disable compression	Do not use compression for looped plane wave sources in the model.

MLFMM

> **Note:** The MLFMM is not supported in conjunction with the multilayer Green's function.

Advantages of adaptive cross-approximation (ACA)

- ACA is done on the matrix level.
- No frequency breakdown like MLFMM.
- Can also be used with the multilayer Green's function.
- Direct solution.

Related concepts
MLFMM Settings (CADFEKO)
Related tasks
Solving a Model with MLFMM (CADFEKO)

FO Card

The FO card is used to define an area in which the surface current density is an approximation.

On the **Solve/Run** tab, in the **Rays** group, click the **Physical optics** icon. From the drop-down list, select the **Fock current (FO)** icon.

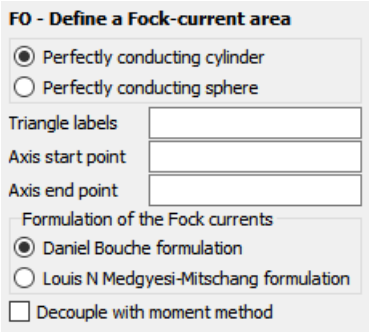


*Figure 711: The **FO - Define a Fock-current area** dialog.*

Parameters:

Perfectly conducting cylinder	Select this option if the Fock area is/ resembles a cylinder.
Perfectly conducting sphere	Select this option if the Fock area is/resembles a sphere.
Triangle labels	The label of the metallic triangles that form the surface of the Fock area (for example, the surface of the cylinder).
Axis start point	This dialog is only visible for Fock cylinders and correspond to the start of the cylinder axis.
Axis end point	This dialog is only visible for Fock cylinders and correspond to the end of the cylinder axis.
Sphere centre point	This dialog is only visible for Fock spheres and should correspond to the centre of the sphere.
Formulation of the Fock currents	The type of process for the Fock currents, either the Daniel Bouche method or the Louis N. Medgyesi–Mitschang method.
Decouple with moment method	Select this check box to force Feko to neglect the coupling between the MoM and Fock regions, so that there is no feedback by which the Fock currents may influence the current distribution in the MoM region. This option, which is particularly applicable when the MoM and Fock regions are not in close proximity, should result in a considerable reduction in computational effort and storage space.

The radius of the cylinder or sphere does not have to be defined. It is determined by the distance to the metallic triangles, with the label specified in **Triangle labels**.

The cylinder Fock currents can also be applied to cones (KK card, approximated by a staircase construction of cylinders) and sections of a torus that resembles a cylinder (TO card). Although the FO card is strictly only applicable to spherical and cylindrical surfaces, it is often a good approximation on conical and toroidal surfaces.

It must be noted that the search for creeping rays on the Fock surface does not take into account multiple Fock regions, for example, one creeping ray can only exist in one Fock region. Therefore, when for instance modelling a sphere and using symmetry, it is highly advisable to create part of the sphere, then use the SY card to mirror it, and only then use one FO card which applies to the whole sphere. When using SY or TG cards, they do operate on already existing Fock regions defined above these cards and thus mirror or move them, but with the SY card the problem is that after applying symmetry there exist multiple Fock regions. The creeping rays along geodesic lines stop at the boundary of each Fock region.

Related reference

KK Card

SY Card

TG Card

TO Card

FP Card

Options related to the Feko solution parameters is set using the FP card. The basis functions used when using FEM or MoM is set globally or on specific labels.

On the **Solve/Run** tab, in the **Solution settings** group, click the **Basis function (FP)** icon.

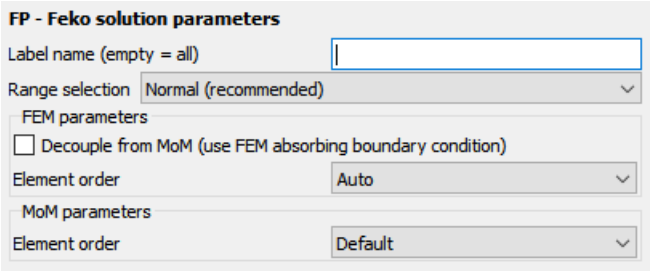


*Figure 712: The **FP - Feko solution parameters** dialog.*

Parameters:

Label name (empty = all)
Label of elements on which the basis function settings are applicable. When this field is empty, the setting is applied globally to all elements.

Range selection
Specify the preferred range that should be used for the higher order basis function selection when the order is set to **Auto**. This setting is used by the Feko kernel to influence the order to be used for a particular triangle size. Selecting higher orders result in computation time and memory usage to increase if the mesh remains unchanged. Lower orders reduce computation time and memory usage.

FEM parameters

Decouple from MoM (use FEM absorbing boundary condition)
The MoM/FEM hybrid method fully couples the FEM and MoM techniques and also the surface of the FEM region is treated by the method of moments. For certain applications when there is a larger separation between the MoM and the FEM regions (for example, human body with a GSM base station antenna) this decoupling check box can be checked. When the FEM and MoM are decoupled, similar to switching off the coupling for the MoM/PO or MoM/UTD hybrid methods, first the MoM region is solved for while neglecting the FEM domain, and the MoM solution is used as an impressed excitation for the FEM solution. The MoM is also not used on the FEM surface when they are decoupled, but rather an absorbing boundary condition is applied on the FEM surface. The advantage of this decoupling is a saving of memory and computation time.

Element order

Three options are available for the order of the FEM solution. The **Auto** option allows the Solver to select the order automatically. Second order will be used by default. However, when having a fine mesh (like modelling details of a biological structure), then one might consider switching to first order only to reduce the number of unknowns.

When first order is selected, then CT/LN basis functions are used everywhere, on the boundary and inside the FEM region. When switching to first order, normally a finer mesh is required to get the same solution accuracy compared to second order basis functions.

When second order is selected, Feko uses hierarchical tetrahedral elements with LT/QN (linear tangential/quadratic normal) vector basis functions for the electric field inside the FEM region. On the boundary surface of the FEM region CT/LN (constant tangential/ linear normal) vector basis functions are used for the equivalent electric and magnetic surface currents.

MoM parameters

Element order

The order of the basis functions used for the MoM is determined by this setting.

The **Default** option selects the default order used in Feko. The default is currently set to use RWG (Rao-Wilton-Glisson) [Rao-Wilton-Glisson] basis functions.

The **Auto** option allows Feko to determine the best order to use. The order of the basis function is determined by the size of the triangle and influenced by the Range selection setting.

Hierarchical basis functions can be used by selecting any of the orders in the list (0.5, 1.5, 2.5 or 3.5). Higher order basis functions have more unknowns,but they allow the triangle size to be increased considerably

Related concepts
FEM Parameters (CADFEKO)
Related tasks
Solving a Region with FEM (CADFEKO)

HC Card

The HC card creates a cylinder with a hyperbolic border.

On the **Construct** tab, in the **Surfaces** group, click the **Hyperbolic cylinder (HC)** icon.

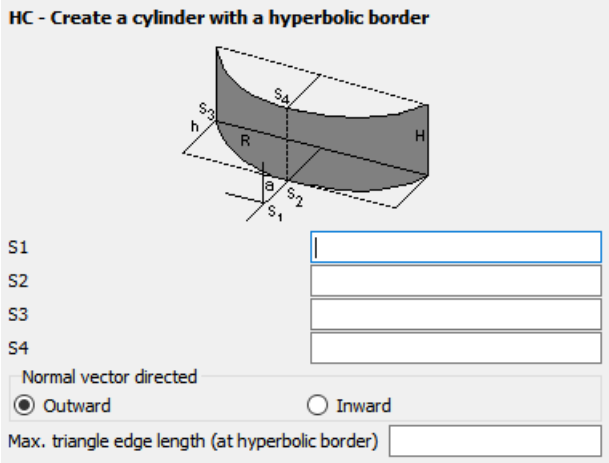


*Figure 713: The **HC - Create a cylinder with a hyperbolic border** dialog.*

Parameters:

S1	The origin (where the asymptotes intersect) of the hyperbolic border.
S2	The pole of the hyperbolic border.
S3	The point where the hyperbolic arc and the straight edge intersect.
S4	The pole of the hyperbolic border at the opposite edge of the cylinder.
Normal vector directed	The normal vectors is directed inward or outward.
Max. triangle edge length (at hyperbolic border)	The maximum edge length on the hyperbolic border.

The hyperbolic border may require shorter mesh edges than those used for straight edges. Thus the maximum segment length specified in the IP card may be overridden along the arc by setting **Max. triangle edge length (at hyperbolic border)**.

Example of HC card usage

The cylinder with a hyperbolic border shown in Figure 714 is created using the HC card.

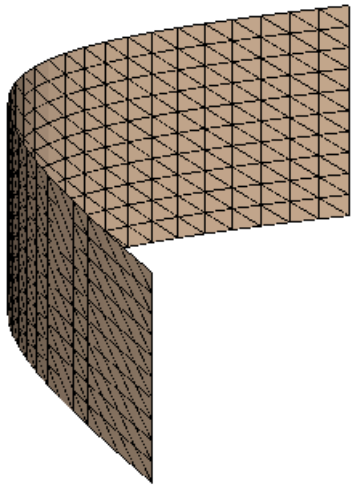


Figure 714: Example of a cylinder with a hyperbolic border created with the HC card.

Related reference
IP Card

HE Card

The HE card creates a helical coil, consisting of wire segments.

On the **Construct** tab, in the **Wires** group, click the **Helix (HE)** icon.

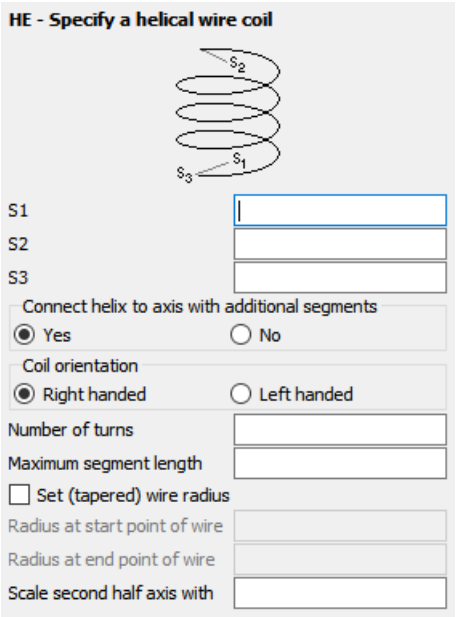


*Figure 715: The **HE - Specify a helical wire coil** dialog.*

Parameters:

S1 — The start point of the coil's axis.

S2 — The end point of the coil's axis.

S3 — The start point of the windings.

Connect helix to axis with additional segments — Create connections from the two ends of the coil to the axis (at points S1 and S2). See also left side of Figure 716. If the connections are not generated, point S3 is a connection point. See also the right side of Figure 716.

Coil orientation — Indicate whether a right- or left handed coil should be created.

Number of turns — In this field the number of turns for the helix is entered. It need not be an integer number.

Maximum segment length — Maximum length of the segments, that are used for the windings in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Set (tapered) wire radius — If this item is checked, a tapered wire radius can be set. Normally the wire radius is set with the IP card. Checking this item overrides this radius for the current helix without affecting the

	default for later segments (The radius is in m and is affected by the SF card scaling factor.) The segments connecting to the axis are not tapered and have radii corresponding to the start point and end point respectively.
Radius at start point of wire	The radius of the wire at the start of the coil.
Radius at end point of wire	The radius of the wire at the end point of the coil.
Scale second half axis with	If this parameter is empty or is set to 1, a helix with a circular cross section is created. If set to $\frac{b}{a}$, a helix with an elliptical cross section is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S1–S3. It is not recommended to generate elliptical helices with extremely small or extremely large axial ratios with a CAD system as the distortion formulation used in PREFEKO may fail in these cases.

Quite often modelling the geometry of the coil requires shorter segments than those used for straight wires. Thus the maximum segment length specified by the IP card can be overridden along the arc by setting Maximum segment length.

The windings are generated between the two points S1 and S2, that lie on the axis. The radius of the coil is defined by the distance between the points S1 and S3. For elliptical cross sections this is the length of one half axis and the other one is Scale second half axis times this length.

Example of HE card usage

The two coils in Figure 716 are created using the HE card.

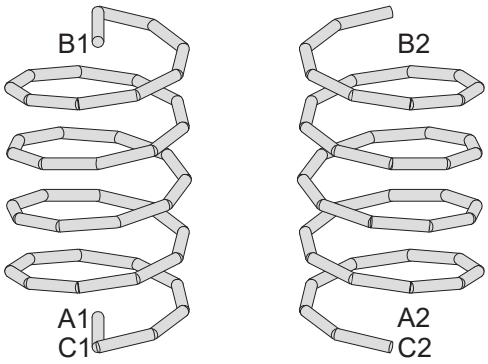


Figure 716: Example of two coils created with the HE card. The coil on the right is coiled in the left handed direction.

Related reference

Helix (CADFEKO)

IP Card

SF Card

HP Card

The HP card creates a plate with a hyperbolic border.

On the **Construct** tab, in the **Surfaces** group, click the **Hyperbolic plate (HP)** icon.

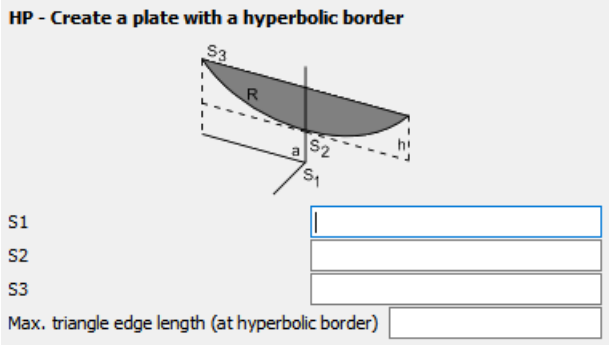


*Figure 717: The **HP - Create a plate with a hyperbolic border** dialog.*

Parameters:

S1	The origin (point at which the asymptotes intersect) of the hyperbolic border.
S2	The pole of the hyperbolic border.
S3	The point where the hyperbolic arc and straight edge intersect.
Max. triangle edge length (at hyperbolic border)	The maximum edge length on the hyperbolic border.

The hyperbolic border may require shorter mesh edges than those used for straight edges. The maximum edge length specified in the IP card can be overridden on the hyperbolic arc by setting **Max. triangle edge length (at hyperbolic border)**.

Example of HP card usage

The plate with a hyperbolic border shown in Figure 718 is created using the HP card.

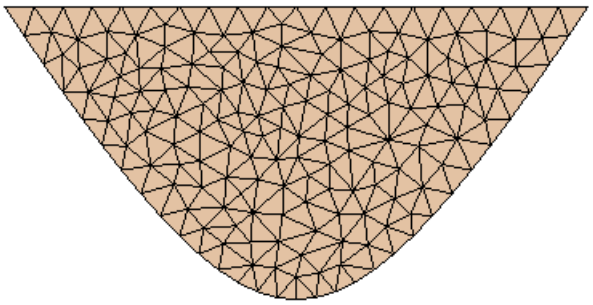


Figure 718: Example of a plate with a hyperbolic border created with the HP card.

Related reference
IP Card

HY Card

With this card a hyperboloid section can be created.

On the **Construct** tab, in the **Surfaces** group, click the **Hyperboloid (HY)** icon.

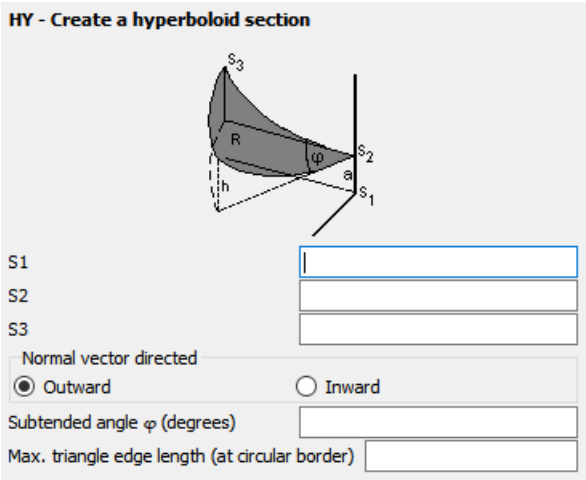


*Figure 719: The **HY - Create a hyperboloid section** dialog.*

Parameters:

S1	The origin (point at which the asymptotes intersect) of the hyperboloid section.
S2	The pole of the hyperbolic border.
S3	A point that defines the outer border of the hyperboloid section.
Normal vector directed	The normal vectors can be directed **Inward** or **Outward**.
Subtended angle φ (in degrees):	The subtended angle measured from S3.
Max. triangle edge length (at circular border):	The maximum edge length on the circular border of the hyperboloid section.

The hyperbolic border may require shorter mesh edges than those used for straight edges. Therefore the maximum segment length specified in the IP card may be overridden along the circular arc by setting **Max. triangle edge length (at circular border)**.

Example of HY card usage:

The hyperboloid section shown is created using the HY card.

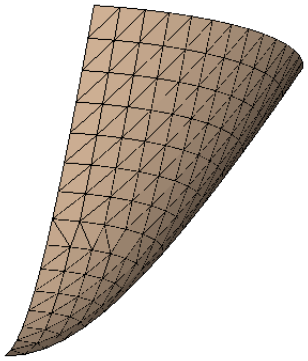


Figure 720: Example of a hyperboloid section created with the HY card.

IN Card

The IN card is used to include external files. These files may be other `.pre` files (which are included as if they were part of the master file) or mesh data files containing wire segments, triangles, quadrangles, tetrahedral volume elements and/or polygonal plates (in FEMAP neutral, ASCII format, NASTRAN, AutoCAD DXF, NEC model, CONCEPT geometry, STL, PATRAN neutral, ANSYS CDB, ABAQUS, GiD or I-DEAS UNV mesh files).

On the **Construct** tab, in the **Import** group, click the **Import (IN)** icon.

Parameters:

These fields are common to more than one option:

File name	The name of the file. This parameter is required for all import options. The file name may contain directory names as well, for example, `../myfiles/include.inc` and will have different extensions for the various import options. Both \ and / are allowed on Windows and UNIX systems.
Include segments	Check this item to include all wire segments that match the label selection.
Include triangles	Check this item to include surface triangles.
Include quadrangles	Check this item to include quadrangles. The quadrangles are subdivided into triangles (along the shortest diagonal) during importation.
Include tetrahedral elements	Check this item to include tetrahedral elements (for FEM).
Include polygons	Check this item to include polygonal plates (for UTD).
Include node points	Check this item to include node points.
Include only node points for imported triangles and/or wires	If this item is checked, only the node points which are used by the imported elements, are imported. This is useful if one imports, for example, a few segments from a file containing a large number of triangles. With this option one may then only import the points associated with the segments — even if they have the same label as the ones associated with triangles only.
Label selection	Most options allow label selective importing. (How the various layers/properties / names are converted to Feko labels is discussed separately for each import option.) One may **Include all items**, **Include items with only a single label** or **Include items with range of labels**. If the first option is selected all elements are imported, irrespective of label. If the single label option is selected, the **Include structures with.. .** field becomes active. Specify the label (as it will be after conversion) in this

field. If the range option is selected, the **Up to.. .** field also becomes active. All elements with the label larger or equal to the first and smaller or equal to the second field, are included. If the import option does not support label selection, all elements are imported.

Scale factor

An optional constant scaling factor can be applied to the imported geometry. This is necessary, for example, if separate CAD files with different units must be imported, or if the `.pre` is, for example, created using mm while the CAD file is constructed using inches as unit. It should be noted that the scaling factor specified here is applied in addition to any scaling factor that may be set with the SF or TG cards.

Related reference

ME Card

SF Card

TG Card

PREFEKO File

This option is used to include cards and commands (such as variable definitions) from a separate file. You may, for example, create a single file with an antenna which is then imported into different environment models.

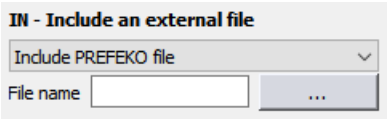


*Figure 721: The **IN - Include an external file (**PREFEKO**)** dialog.*

For this option, the file name is the only parameter. This file is included as if it was part of the master `.pre` file. These include files usually have the extension `.inc`, but can have any extension. The cards and instructions in the included files are processed as if they were part of the main file. Therefore, points and labels defined in the included file remain valid in the remainder of the main file.

> **Note:** That it is also possible to use such an IN card in the control section of the `.pre` file (for instance to import a feed model).

When reading a PREFEKO file it is not possible to add a scaling factor to the IN card. In this case the TG card must be used if the global (SF card) scaling option is not sufficient.

It is possible to use multiple nested levels of include files (for example, one include file can include another one and so on). It is also possible to specify together with the file name an absolute or relative path like in

```
IN 0 "..\subdir\file.inc"
```

In such a case — if multiple levels of include files are used — it is first tried to find the include file using the path relative to the location of the file where the IN card is used. If the include file is not found there, then PREFEKO also tries to find the include file using the path relative to the location of the main `.pre` file which is processed.

Related reference

SF Card

TG Card

FEMAP Neutral File

This option is used to import models generated by the commercial CAD meshing program FEMAP. The models must be exported from FEMAP in the FEMAP neutral file format (`.neu`).

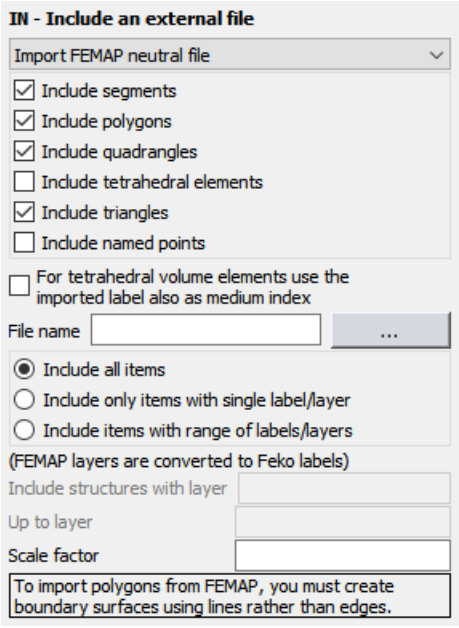


*Figure 722: The **IN - Include an external file (FEMAP neutral)** dialog*

This card supports all the parameters described in the general section of the IN card above.

The label selection uses the FEMAP layer numbers which are converted to Feko labels. Wires must be meshed into elements which are imported as segments, surfaces into triangles or quadrangles which are imported as Feko triangles, and boundary surfaces are imported as polygonal plates. The boundary surface must be bordered with line curves rather than edge curves.

The user can also elect to import points from the `.neu` file. All points defined as such in FEMAP are then available in PREFEKO as points (as if they were defined by DP cards) of the form *Pxxx* where *xxx* is the point ID in FEMAP. This may be used, for example, when attaching additional structures to a geometry partly created in FEMAP. In addition, the coordinate values of the point are available as variables in PREFEKO. For example, the variables *#p1234x*, *#p1234y* and *#p1234z* give the coordinates of the FEMAP point with ID 1234.

> **Note:** That points are not included by default.

It should be remembered that it is not possible to specify a wire radius in FEMAP. Thus the wire radius must be specified by an IP card preceding the IN card. Similarly, when specifying the surface of a dielectric, the IN card must be preceded with the correct ME card (completely analogous to the case without FEMAP).

POSTFEKO should be used to verify the included geometry.

Related reference
DP Card

Import Special ASCII Data File

This option is used to import meshes stored in the geometry data file as specified below.

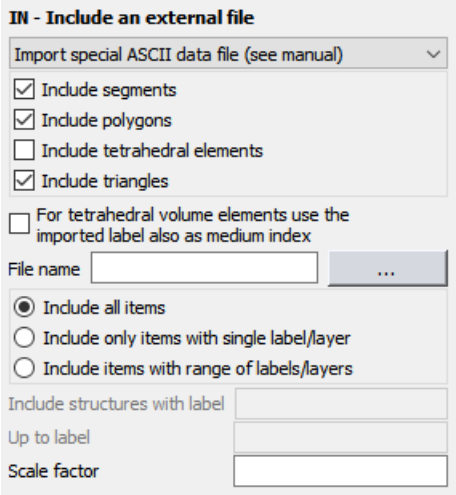


*Figure 723: The **IN - Include an external file (Special ASCII data file)** dialog.*

The ASCII format supports segments, triangles, tetrahedra and polygonal plates, but all other (non-selection) parameters discussed in the general section of the IN card above apply. In this case the label is specified directly in the file and no conversion is required.

Dielectric triangles or metallic triangles which form the surface of a dielectric, are created by preceding the IN card with the appropriate ME card. (In exactly the same way as is the case without the IN card.)

The data of the segments, triangles and polygonal plates are given in an ASCII file, formatted as shown below. There is no need to adhere to specific columns, the data fields merely have to be separated by one or more spaces.

```
nk   nd   ns   np   nt
x(1)   y(1)   z(1)   (String_name)
x(2)   y(2)   z(2)   (String_name)
...
x(nk)  y(nk)  z(nk)   (String_name)

d1(1)  d2(1)  d3(1)  0  (Label)
d1(2)  d2(2)  d3(2)  0  (Label)
...
d1(nd) d2(nd) d3(nd) 0  (Label)

s1(1)  s2(1)  0      0  (Label)
s1(2)  s2(2)  0      0  (Label)
...
s1(ns) s2(ns) 0      0  (Label)

nnp(1)   p1(1)  p2(1)  p3(1)  ...   (Label)
nnp(2)   p1(2)  p2(2)  p3(2)  ...   (Label)
...
nnp(np)  p1(np) p2(np) p3(np) ...   (Label)

t1(1)  t2(1)  t3(1)  t4(1)  (Label)
t1(2)  t2(2)  t3(2)  t4(2)  (Label)
...
```

```
t1(nt)  t2(nt)  t3(nt)  t4(nt)  (Label)
```

The variables are described as follows

nk	Number of nodes.
nd	Number of triangles.
ns	Number of segments.
np	Number of polygonal plates.
nt	Number of tetrahedral volume elements (defaults to 0 if not specified).
x(i)	X coordinates of node *i* in metre (is scaled by the SF card).
y(i)	Y coordinates of node *i* in metre (is scaled by the SF card).
z(i)	Z coordinates of node *i* in metre (is scaled by the SF card).
d1(j)	Number (index) of the first vertex of triangle *j*.
d2(j)	Number (index) of the second vertex of triangle *j*.
d3(j)	Number (index) of the third vertex of triangle *j*.
s1(k)	Number (index) of the starting point of segment *k*
s2(k)	Number (index) of the end point of segment *k*
nnp(m)	Number of corner points in polygon metre.
p1(m)	Number (index) of the first corner of polygon metre.
p2(m)	Number (index) of the second corner of polygon metre.
p3(m)	Number (index) of the third corner of polygon metre.
t1(m)	Number (index) of the first corner of tetrahedron metre.
t2(m)	Number (index) of the second corner of tetrahedron metre.
t3(m)	Number (index) of the third corner of tetrahedron metre.
t4(m)	Number (index) of the fourth corner of tetrahedron metre.
...	Number (index) of the additional corners of polygon metre.
String_name	Optional string name of the point. It must be a string of up to five characters, similar to the point name of the DP card. If a point is named, it can be used in any card following the IN card.

Label

Specifying the label as the last parameter of any structure is optional. If no label is specified, the value defined at the last LA card will be used.

> **Note:** That if a label or range of labels is specified (with parameters after the file name), this LA card label will be used to determine if a structure is included or not.

The radius of segments must be specified by an IP card before the IN card. It is recommended to check the geometry with POSTFEKO.

Example

The structure in Figure 724, consisting of 5 node points and 3 triangles with label 7 (no segments or polygonal plates), may be imported from the following data file

```
5  3  0  0
3.0  0.0  1.0
4.0  2.0  1.0
2.5  3.0  2.5
0.0  3.0  4.0
1.0  0.0  3.0
1  2  3  0  7
1  3  5  0  7
3  4  5  0  7
```

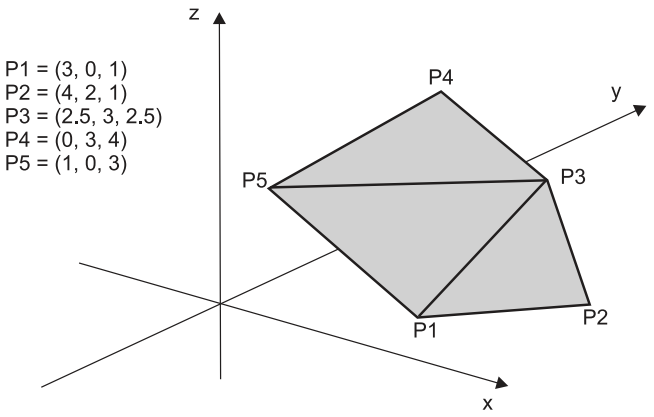


Figure 724: Example of a structure with 5 node points and 3 triangles created with the IN card.

Related reference

DP Card
IP Card
LA Card
ME Card
SF Card

Import NASTRAN File

With this option, PREFEKO can import a model from a NASTRAN file. It supports both 8-character and 16-character wide column based files as well as comma separated files. NASTRAN files in cylindrical, spherical coordinate systems, as well as files in Cartesian coordinate systems are supported. Only the keywords GRID, CTRIA3, CQUAD4, CTETRA, CBAR and CROD for nodes, triangles, quadrangles (divided into two triangles along the shortest diagonal), tetrahedral elements, and segments are processed.

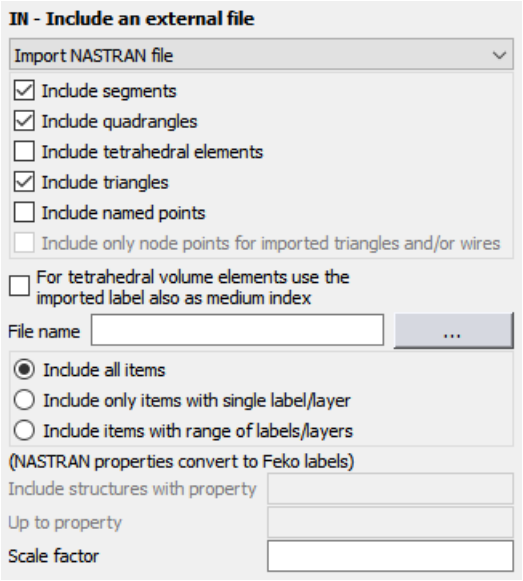


*Figure 725: The **IN - Include an external file (NASTRAN file)** dialog.*

NASTRAN does not support polygonal plates, but all other parameters in the general section of the IN card above apply. The label selection uses the NASTRAN properties which are converted to Feko labels.

As when importing FEMAP neutral files, the wire radius must be set with the IP card preceding the IN card, and an ME card must be used when specifying dielectric surfaces in the same way as when the IN card is not present.

The user can also import points from the NASTRAN file. The points defined in the NASTRAN file will then be available in PREFEKO as points (as if they were defined by DP cards) of the form *Nxxx* where *xxx* is the index of the grid point. This may be used, for example, to attach additional structures to the geometry. In addition, the coordinate values of the point are available as variables in PREFEKO. For example, the variables *#n1234x*, *#n1234y* and *#n1234z* give the coordinates of the NASTRAN grid point with index 1234.

> **Note:** That points are not included by default.

Since grid points do not have an associated property, points are imported irrespective of their label, but they may be limited to those used for the imported geometry.

Each line in the 8-character column based format consists of one keyword such as **GRID** starting in column 1. From column 9 onwards follow 9 input fields with widths of 8 characters each. Thus input field 1 uses columns 9 to 16, input field 2 uses columns 17 to 24 and the rest. The ninth (and last)

input field ends at column 80. Below is a very simple NASTRAN example file consisting of a plate (property 1; subdivided into eight triangles) and a rod (property 2; subdivided into two segments).

```
|1             |9         |17      |25       |33       |41       |49
```

```
ID XXXXXXXX,YYYYYYYY
CEND
BEGIN BULK
GRID            1              0.0      0.0      0.0
GRID            2          0.50000      0.0      0.0
GRID            3          1.00000      0.0      0.0
GRID            4              0.0  0.50000      0.0
GRID            5          0.50000  0.50000      0.0
GRID            6          1.00000  0.50000      0.0
GRID            7              0.0  1.00000      0.0
GRID            8          0.50000  1.00000      0.0
GRID            9          1.00000  1.00000      0.0
GRID           10          0.50000  0.50000  2.00000
GRID           11          0.50000  0.50000  1.00000
CROD            9        2        5       11
CROD           10        2       11       10
CTRIA3          1        1        4        5        8
CTRIA3          2        1        4        8        7
CTRIA3          3        1        5        6        9
CTRIA3          4        1        5        9        8
CTRIA3          5        1        1        2        5
CTRIA3          6        1        1        5        4
CTRIA3          7        1        2        3        6
CTRIA3          8        1        2        6        5
ENDDATA
```

For the node points Feko also supports 16 character wide input fields. The keyword GRID in columns 1 to 4 is followed by a star and three spaces. The node ID is then in columns 9 to 24, the x coordinate in columns 43 to 56, y in columns 57 to 72 and z in columns 9 to 24 of the next line.

For example

```
|1       |9               |25...      |43            |57              |73      |81
```

```
GRID*                   1               50.000000000   -18.480176926        1
*       1    23.222875595
GRID*                   2               50.000000000   -18.480176926        2
*       2   -13.410394669
```

For the comma separated format, the individual entries are separated by comas

```
GRID,1,0,-238.533,186.7983,0.000000,0
GRID,2,0,-244.777,214.3057,172.9991,0
GRID,3,0,288.0060,115.1831,339.8281,0
GRID,4,0,356.2201,50.15516,0.000000,0
CTRIA3,1,1,1,2,3,,0.0,,
CTRIA3,2,1,1,2,4,,0.0,,
```

Related reference

DP Card

IP Card

ME Card

Import AutoCAD DXF File

This card allows the import of `.dxf` models. The `.dxf` file must comply with the release 12 DXF format specifications. It should contain meshed- or closed- polyline surfaces (see the discussion below) and lines (that will be segmented by PREFEKO as discussed below).

In addition to the file name, label selection and scale factor discussed in the general section of the IN card above, the DXF import option supports the following element selection:

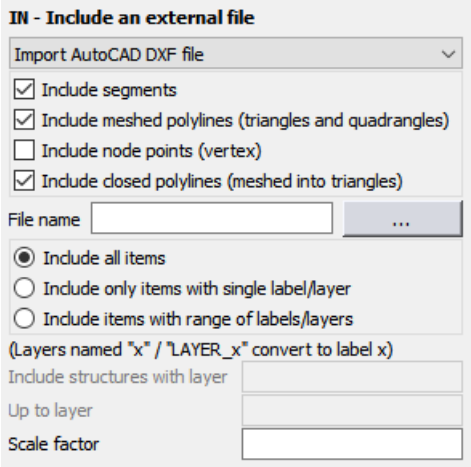


*Figure 726: The **IN - Include an external file (AutoCAD DXF)** dialog.*

Parameters:

Include segments	Check this item to include all wire segments that match the label selection.
Include meshed polylines (triangles and quadrangles)	Check this item to include meshed polylines (triangles and quadrangles) from the DXF file into the model.
Include node points (vertex)	Check this item to include node points from the DXF file into the model.
Include closed polylines (meshed into triangles)	Check this item to include closed polylines from the DXF file into the model. These structures will be meshed by PREFEKO.

Layers named n or `LAYER_n` (where n is an integer number) in the `.dxf` model are converted to label n in Feko. For all structures for which no label is defined in this format, the label specified with the last LA card preceding the IN card is used. (If no such LA card is in effect, the default is label 0.) This label is used in the label selection.

As for the other meshed CAD formats, dielectric triangles or metallic triangles which form the surface of a dielectric, are created by preceding the IN card with the appropriate ME card.

PREFEKO only processes the geometry information in the section of the file between the keyword `ENTITIES` and `ENDSEC`.

The user can import points (for example, vertices of polylines and start/end points of lines) from the DXF file. The points defined in the DXF file will then be available in PREFEKO as points of the form *Qxxx* where *xxx* is a consecutive point index. In addition, the coordinate values of the point are available as

variables in PREFEKO. For example, the variables *#q1234x*, *#q1234y* and *#q1234z* give the coordinates of point 1234.

> **Note:** That points are not included by default. The imported node points will have the label (for example, converted layer) of the corresponding LINE or POLYLINE structure (the layer of the VERTEX block for polylines is not used).

A label range selection at the IN card may be applied so that only the points with a correct layer will be imported.

Segments are imported from blocks defined by the keyword LINE.

```
  0
LINE
  8
LAYER_01
  .
  .
 10
-0.0538
 20
0.0
 30
8.134
 11
5.110
 21
2.857
 31
0.0
  0
 ... (next keyword)
```

The group code 8 at a point below LINE indicates that the next line contains the layer name. In this case, the layer will be converted to label 1. The line will be imported and segmented if this label lies in the required range. (If not, PREFEKO will search for the next occurrence of LINE.) Next the x, y and z components of the start point follow the group codes 10, 20 and 30; and those of the end point follow the codes 11, 21 and 31.

Here the start and end points are (x, y, z) = (-0.0538, 0.0, 8.134) and (5.110, 2.857, 0.0) respectively. If any of the coordinate group codes are absent (such as in a 2D model), the related coordinate is set to zero. The block is terminated by the group code 0. The wire is segmented according to the maximum segment length specified by the IP card, and the segments also have the radius specified by this card.

Meshed surfaces are imported from blocks denoted with the keyword POLYLINE. This block contains the layer name (following the group code 8 as before; if there is no group code 8 before the first VERTEX, the label specified with the last LA card will be used) and a number of VERTEX structures. There can be an arbitrary number of VERTEX structures, but there should be at least four.

The POLYLINE structure is terminated by the keyword SEQEND.

```
  0
POLYLINE
  8
LAYER_02
  .
```

```
  .
VERTEX
  .
  .
VERTEX
  .
  .
VERTEX
  .
  .
  0
SEQEND
```

There are two types of vertices. The first type defines points in space

```
  0
VERTEX
  8
LAYER_02
  .
  .
 10
7.919192
 20
3.393939
 30
0.0
  .
  .
  0
 ... (next keyword)
```

where the x, y and z components of the point follow the group codes 10, 20 and 30. The layer information is ignored. The second type of vertex is a “linker”.

```
 0
VERTEX
  8
LAYER_02
  .
  .
 70
   128
 71
     4
 72
     2
 73
     1
 74
     3
  .
  .
  0
 ... (next keyword)
```

which defines a triangle or quadrangle by specifying the indices (starting from 1 in the order the non-linker vertices are specified) of the vertices which form its corner points. Vertices are defined as linkers by setting a value of 128 in the group code 70 field. For linker vertices the coordinates are ignored.

> **Note:** That old `.dxf` versions do not contain linker vertices — they cannot be imported. (Usually they do not contain mesh information.)

The four integer numbers after the group codes 71, 72, 73 and 74 give the indices of corners of the triangle or quadrangle. (In the case of a triangle one of these is absent.) PREFEKO divides each quadrangle into two triangles along the shortest diagonal.

In addition to being able to import meshed polylines, closed polylines can also be imported. These will be meshed into triangular patches during the import according to the meshing parameters set at the IP card.

Related reference

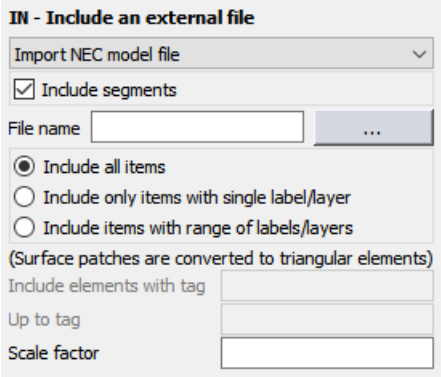
IP Card

LA Card

ME Card

Import NEC Model File

PREFEKO supports the import of wire geometry from NEC[38] models.



> **Note:** That NEC models usually consist of wire grid surfaces and it would be more efficient to convert the models to Feko surfaces, but this cannot be done automatically.

For this option only the file name, label selection and Include segments are supported.

The label selection uses the NEC tags which are converted to Feko labels. This applies to the tag when the element is defined. If the tag is modified after the inclusion (for example with the GM card) the elements with the modified tag are also included. The type selection parameter x is also supported, but it may only have the value 1 for wire segments.

The NEC import filter considers only the geometry cards CM, CE, GA, GW, GM, GR, GS, GX and GE. A warning is given if other cards are encountered. If the model contains multiple geometries only the first one is read.

Related reference

CM Card

Import CONCEPT Geometry File

With this option one may import CONCEPT[39] geometry files

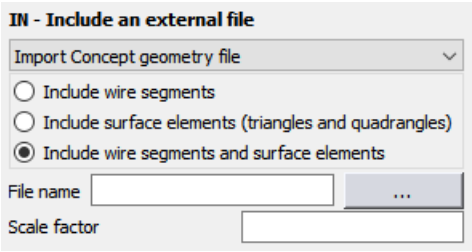


*Figure 727: The **IN - Include an external file (Concept geometry file)** dialog.*

Since CONCEPT uses two different files for wires and surface elements, the type of element selection is obligatory and determines the type of geometry file to be read. The only other parameters supported here are the file name and scale factor. (The CONCEPT file does not contain labels.) If the CONCEPT model contains quadrangles, they are divided into triangles.

Since wires don't have a radius in the model files, the radius is specified with a preceding IP card. Likewise, the elements don't have labels, and the label as specified at the last LA card before the IN card is used. If there is no LA card, the label defaults to zero.

As for the CAD models, dielectric triangles or metallic triangles which form the surface of a dielectric, are created by preceding the IN card with the appropriate ME card.

The CONCEPT files for wires are as follows

```
number_of_wires
x_start y_start z_start x_end y_end z_end   [number_of_wires times]
```

where the first integer specifies the number of wires followed by the coordinates of the start and end point of each wire. The file is completely free format — the values are just separated by white space. The surface file is

```
number_of_nodes   number_of_patches
x y z                                   [number_of_nodes times]
p1 p2 p3 p4                             [number_of_patches times]
```

again using free format. The values `x`, `y` and `z` specify the node coordinates and `p1`, `p2`, `p3` and `p4` specify the corner nodes of the triangles (in this case `p4` is 0) and quadrangles.

Related reference

IP Card

LA Card

ME Card

Import STL File

PREFEKO can also import STL — both ASCII and binary — files. STL files supports only triangular patches and these are all imported. Also, since the STL file makes no provision for any labels, label selection is not supported. The scale factor is supported.

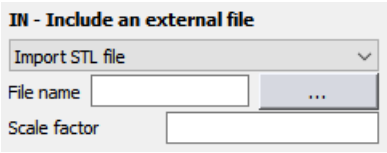


*Figure 728: The **IN - Include an external file (STL)** dialog.*

An example of an ASCII file is.

```
SOLID CATIA STL PRODUCT
  FACET NORMAL -4.602166E-01 -1.858978E-01 -8.681260E-01
    OUTER LOOP
      VERTEX    4.789964E-01 -8.440244E-01  2.878882E-01
      VERTEX    4.764872E-01 -8.439470E-01  2.892018E-01
      VERTEX    4.783065E-01 -8.414296E-01  2.876983E-01
    ENDLOOP
  ENDFACET
  FACET NORMAL -4.601843E-01 -1.859276E-01 -8.681367E-01
    OUTER LOOP
      VERTEX    4.764872E-01 -8.439470E-01  2.892018E-01
      VERTEX    4.761175E-01 -8.425569E-01  2.891001E-01
      VERTEX    4.783065E-01 -8.414296E-01  2.876983E-01
    ENDLOOP
  ENDFACET
ENDSOLID
```

For the description of binary STL files, please see: www.ennex.com

Import CADFEKO Mesh File

CADFEKO exports the mesh to a `.cfm` file which is imported by an IN card in the default PREFEKO file created by CADFEKO. The options are similar to those of the other formats that PREFEKO can import. For the import of `.cfm` files (in addition to the mesh element inclusion/exclusion options provided) provision is made for the inclusion/exclusion of the variables that are defined in the `.cfm` file during the import process. Imported variables can then be referred to in other PREFEKO cards.

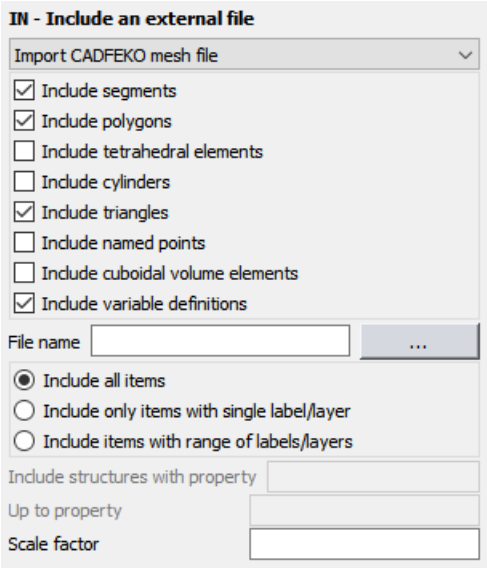


*Figure 729: The **IN - Include and external file (CADFEKO mesh)** dialog.*

Related reference

DP Card

Import Feko HyperMesh File

Import a mesh from a Feko HyperMesh (`.fhm`) file.

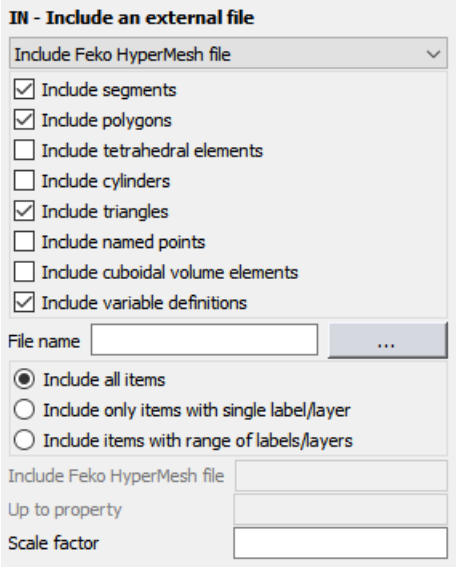


*Figure 730: The **IN - Include and external file (Feko HyperMesh file)** dialog.*

Import media definitions from a `.inc` file by using the **Include PREFEKO file** option.

Import PATRAN Neutral File

PREFEKO also supports importing PATRAN files.

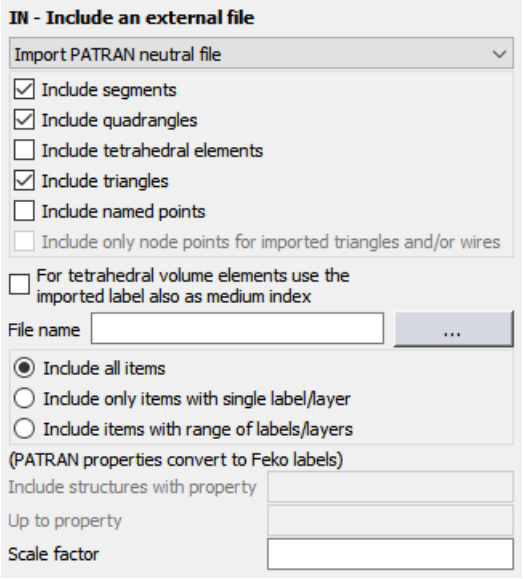


*Figure 731: The **IN - Include an external file (PATRAN)** dialog.*

PATRAN does not support polygonal plates, but all other parameters in the general section of the IN card above apply. The label selection uses the PATRAN material ID's which are converted to Feko labels. Only the following PATRAN neutral packet types are imported:

01	Node data (all coordinates are interpreted in the global rectangular frame, local coordinate frames are not supported).
02	Element data. The shapes 2(bar), 3(tri), 4(quad) and 5(tet) are allowed. Quadrangles are automatically subdivided into triangles along the shortest diagonal.
99	End of file flag.

Other packet types are ignored.

As when importing `.neu` files, the wire radius must be set with the IP card preceding the IN card, and an ME card must be used when specifying dielectric surfaces in the same way as when the IN card is not present.

The user can also import points from the PATRAN file similar to importing points from FEMAP or NASTRAN files. The points defined in the PATRAN file will then be available in PREFEKO as points (as if they were defined by DP cards) of the form *Txxx* where *xxx* is the index of the grid point. This may be used, for example, to attach additional structures to the geometry. In addition, the coordinate values of the point are available as variables in PREFEKO. For example, the variables *#t1234x*, *#t1234y* and *#t1234z* are set to the coordinates of the point with index 1234. Note that points are not included by default. Since points do not have an associated property ID, points are imported irrespective of their label.

Related reference

DP Card

IP Card

ME Card

Import ANSYS CDB File

PREFEKO supports the import of geometry from ANSYS `.cdb` files. By default, when exporting such files from ANSYS, the BLOCKED option is used. PREFEKO only understands this BLOCKED syntax, the UNBLOCKED version is not supported. Also regarding the element type, only the ANSYS element types 200 (filaments, triangles, tetrahedral elements, and brick elements) as well as element type 16 (pipe16, wire with a finite radius) are supported.

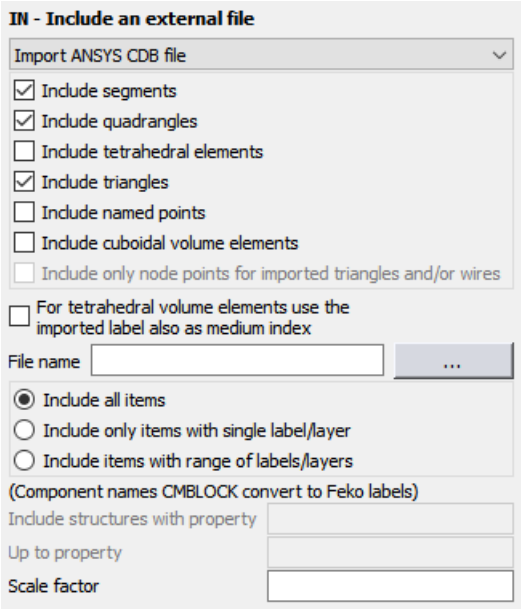


*Figure 732: The **IN - Include an external file (ANSYS CDB)** dialog.*

The selection of polygonal plates and quadrangles are not supported, but all other (non-selection) options discussed in the general section of the IN card is supported. It also supports an additional selection option:

Include cuboidal volume elements	Check this item to include cuboidal elements to be used with the volume equivalence principle in Feko.

The component name from the CMBLOCK is converted to the Feko label. Since the Feko label must be an integer value, only component names which are integer strings (for example, 15) or end with an underscore followed by an integer string (for example, FEED_7) will be converted to Feko labels (15 and 7 in the examples above). In all other cases (for example, for a component name PATCH) the Feko label will be set to zero.

Note: Unlike most of the other CAD import formats supported by the IN card, the ANSYS CDB file makes provision for a wire radius of the segments of type pipe16 (real constant from the associated RLBLOCK)

This is then used during the import and any setting at the IP card is ignored (the IP card radius is still used for filaments of element type 200). For dielectric bodies, one must use an ME card to specify the element type and medium indices. The ANSYS field for the material number cannot be used, since for triangles Feko requires two such material indices (medium on each side).

The user can also import points from the ANSYS CDB file similar to importing points from FEMAP or NASTRAN files. The points defined in the ANSYS CDB file will then be available in PREFEKO as points (as if they were defined by DP cards) of the form *Cxxx* where *xxx* is the index of the grid point. This may be used, for example, to attach additional structures to the geometry. In addition, the coordinate values of the points are available as variables in PREFEKO. For example, the variables *#c1234x*, *#c1234y* and *#c1234z* are set to the coordinates of the point with index 1234.

Note: Points are not included by default.

Related reference

DP Card

IP Card

ME Card

Import ABAQUS Mesh File

PREFEKO supports the import of mesh files from ABAQUS `.inp` files.

The various options for the card panel are as described globally or at the other import options:

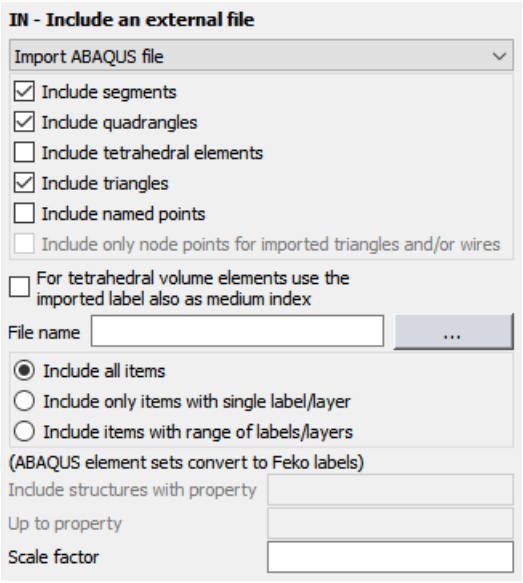


*Figure 733: The **IN - Include an external file (ABAQUS)** dialog.*

Here for ABAQUS mesh files, the ABAQUS element set (`ELSET` command) is converted to the Feko label.

Points are imported similar to the case for NASTRAN files, for example, they are available in PREFEKO as points (as if defined by DP cards) of the form *Nxxx* where *xxx* is the index of the grid point. The coordinate values of the point are available as variables of the form *#n1234x*, *#n1234y* and *#n1234z*.

> **Note:** That points are not included by default.

Related reference

DP Card

Import GiD Mesh

PREFEKO supports the import of mesh files from GiD `.msh` files.

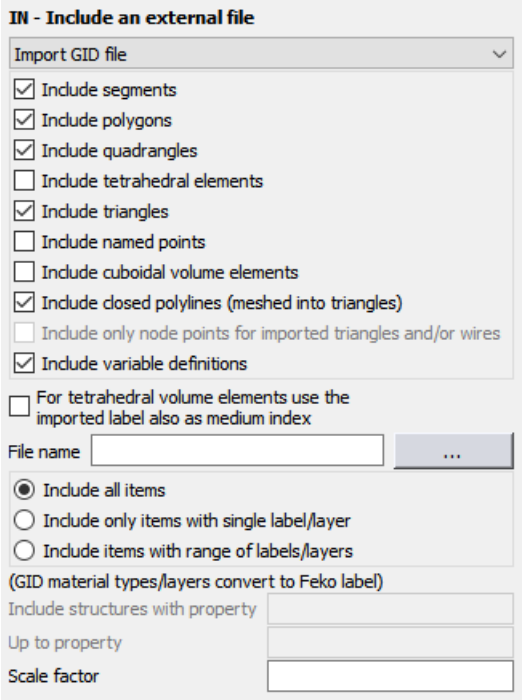


*Figure 734: The **IN - Include an external file (GiD)** dialog.*

For GiD mesh files, the material type or layer property is converted to the Feko label during the import.

The import options are similar to the options for NASTRAN and ABAQUS mesh imports. The following should be noted regarding the support of special elements in the GiD mesh.

- Hexahedral elements are not supported in the Feko import of GiD meshes.
- Nodes that only contain 2 coordinates are interpreted as X and Y coordinates on the z = 0 plane.
- Quadrilateral elements are divided into triangle elements along the shortest diagonal during import.

Import I-DEAS UNV Mesh

PREFEKO supports the import of mesh files containing segments and triangles from I-DEAS UNV `.unv` files.

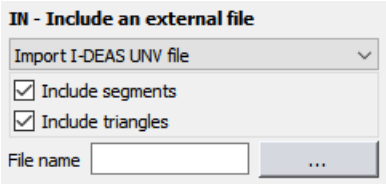


*Figure 735: The **IN - Include an external file (I-DEAS)** dialog.*

IP Card

The IP card defines a number of meshing parameters as well as the wire radius.

On the **Home** tab, in the **Define** group, click the **Meshing (IP)** icon.

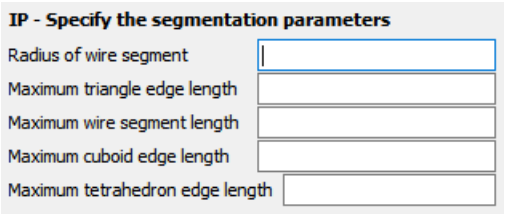


*Figure 736: The **IP - Specify the segmentation parameters** dialog.*

Parameters:

Radius of wire segment	Segment radius in m (it is scaled by the SF card).
Maximum triangle edge length	Maximum edge length of triangular elements in m (it is scaled by the SF card).
Maximum wire segment length	Maximum edge length of triangular elements in m (it is scaled by the SF card).
Maximum cuboid edge length	Maximum edge length of cuboidal volume elements for dielectrics (volume equivalence principle of the method of moments) in m (it is scaled by the SF card).
Maximum tetrahedron edge length	Maximum edge length of tetrahedral volume elements (finite element method) in m (it is scaled by the SF card).

The IP card only affects the commands or cards subsequent to itself. The implication is that the IP card has to be declared prior to the cards that define segments, triangles or cuboids.

It is possible to use more than one IP card in a file. This is necessary when a finer mesh is required in certain parts or where different radii are used in the geometry. For any command, such as the BL card, the previous IP card is applicable.

Specific rules apply relating the element size for meshing to the wavelength.

Related concepts
Meshing Guidelines
Related tasks
Modifying the Auto-Generated Mesh
Related reference
BL Card
SF Card

KA Card

The KA card defines an edge between two points that forms the border of the PO area. On this edge the fringe wave currents are taken into account.

On the **Solve/Run** tab, in the **Rays** group, click the **Physical optics** icon. From the drop-down list, select the **PO edge (KA)** icon.

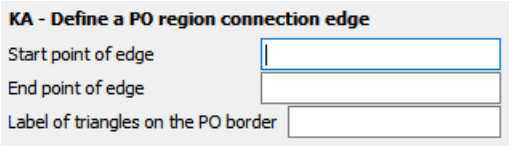


*Figure 737: The **KA - Define a PO region connection edge** dialog.*

Start point of edge	The start point of the edge.
End point of edge	The end point of the edge. The start/end point can be arbitrary and the direction of the edge is irrelevant.
Label of triangles on the PO border	The label of the PO triangles next to the PO border. The edge correction current from this edge is applied to all triangles with this label.

Note: The surface must be flat, which implies that all triangles with the label specified here must lie in the same plane.

KK Card

The KK card defines a mesh of surface triangles in the shape of a conical section.

On the **Construct** tab, in the **Surfaces** group, click the **Cone (KK)** icon.

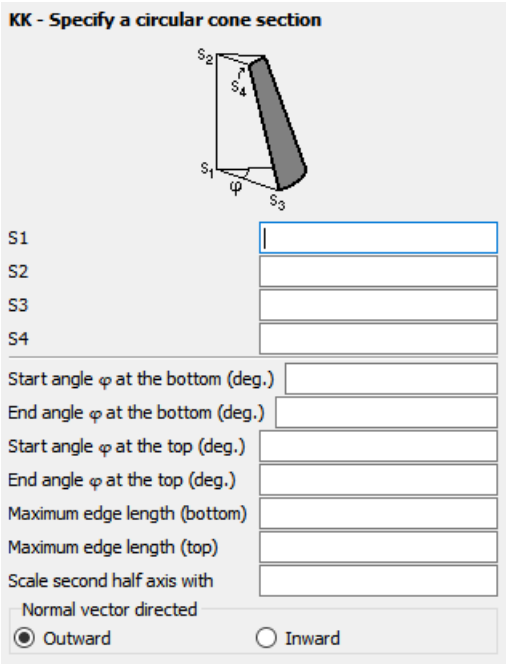


*Figure 738: The **KK card - Specify a circular cone section** dialog.*

Parameters:

S1	The start point of the axis of the cone (the centre of the base).
S2	The end point of the axis (the tip of the cone, or the centre point of the circle when creating a conical section).
S3	A point on the radius of the base.
S4	If this parameter is defined, the cone is cut off at the top. If not defined the cone will have a sharp tip. This point must be in the plane given by S1, S2 and S3.
Start angle at the bottom	The angle φ from the plane S2-S1-S3 at which the bottom of the cone should start.
End angle at the bottom	The angle φ from the plane S2-S1-S3 at which the bottom of the cone should end.
Start angle at the top	The angle φ from the plane S2-S1-S3 at which the top of the cone should start.
End angle at the top	The angle φ from the plane S2-S1-S3 at which the top of the cone should end.

Maximum edge length (bottom)	The maximum edge length of the triangles along the base arc (in the plane containing S1) of the cone. This value is in m and is scaled by the SF card. If this parameter is left empty, the value specified with the IP card is used.
Maximum edge length (top)	This value only applies if S4 is specified and gives the maximum edge length of the triangles along the top arc (in the plane containing S2) of the cone. This value is in m and is scaled by the SF card. If this parameter is left empty, the value specified with the IP card is used.
Normal vector directed	The triangles can be created so that the normal vector points **Outward** (away from the axis) or **Inward** (to the inside of the cone).
Scale second half axis	If this parameter is empty or is set to 1, a cone with a circular cross section is created. If set to $\frac{b}{a}$, a cone with an elliptical cross section is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S1–S3. It is recommended to generate elliptical cones with extremely small or extremely large axial ratios with a CAD system as the distortion formulation used in PREFEKO may fail in these cases.

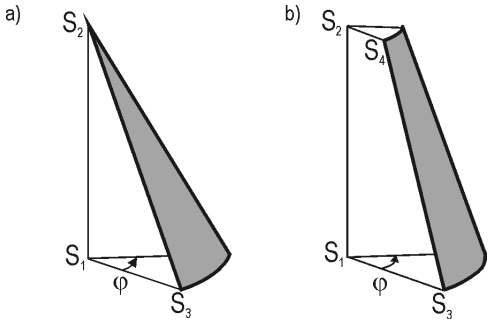


Figure 739: Sketch of using the KK card: (a) cone and (b) conical section.

The fineness of the mesh on the shell's surface is determined by the maximum edge length specified by the last IP card prior to the KK card. Along the arcs, accurate modelling of the geometry may require finer segmentation and the values **Maximum edge length (bottom)** and **Maximum edge length (top)** specify the maximum edge length along the corresponding arcs. **Maximum edge length (top)** is only used when a truncated cone is created. If either of these values is not specified the length specified with the IP card will be used on the corresponding arc.

Examples of KK card usage

All of the following meshes are created using the KK card. These examples show a sharp cone, an oblique elliptical cone, a conical section with different angles at the top and bottom as well as a conical section where the start angle is not in the plane S2–S1–S3.

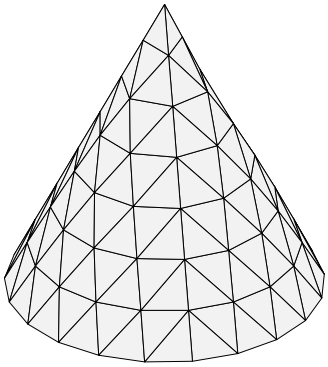


Figure 740: Example of a conical shell created with the KK card.

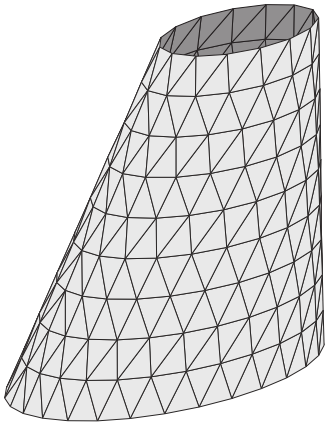


Figure 741: Example of a cone with elliptical cross section created with the KK card.

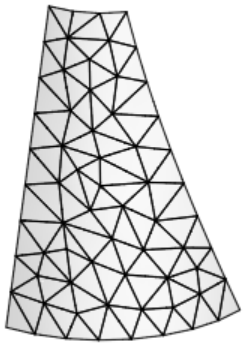


Figure 742: Example of cone with different subtended angles at the top and at the bottom created with the KK card.

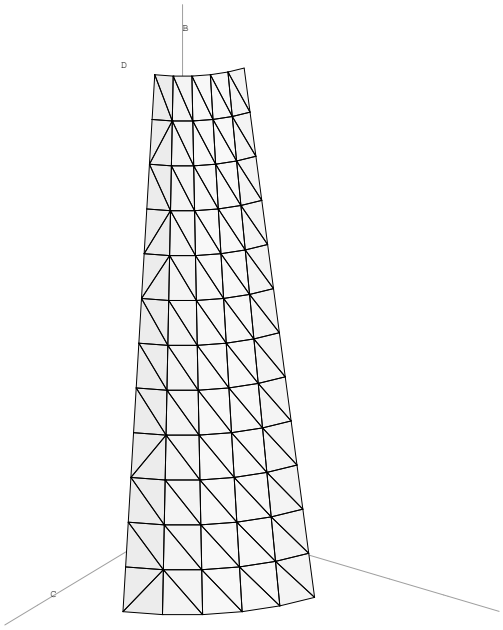


Figure 743: Example of a conical section where the start angle is not in the plane defined by S1, S1 and S3 created with the KK card.

Related reference

Cone (CADFEKO)

IP Card

SF Card

KL Card

The KL card defines a wedge for which correction terms are added to the PO currents on two surfaces connected to it.

On the **Solve/Run** tab, in the **Rays** group, click the **Physical optics** icon. From the drop-down list, select the **PO wedge (KL)** icon.

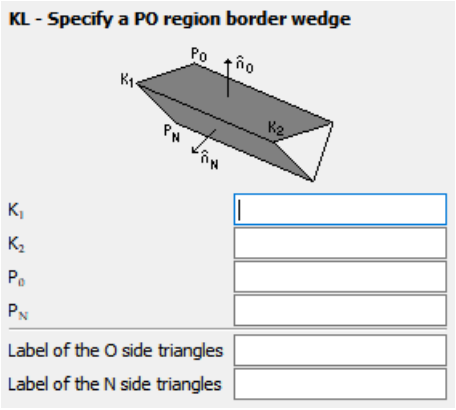


*Figure 744: The **KL card - Specify a PO region border wedge** dialog.*

Parameters:

K_1	The start point of the axis of the cone (the centre of the base).
K_2	The end point of the axis (the tip of the cone, or the centre point of the circle when creating a conical section).
P_0	A point on the *O* side of the wedge.
P_N	A point on the *N* side of the wedge.
Label of the O side triangles	The label of the PO triangles that are adjacent to the wedge on the *O* side. This means that the corresponding correction term for the *O* side is assigned to the PO triangles that have this label.
Label of the N side triangles	The label of the PO triangles that are adjacent to the wedge on the *N* side. This means that the corresponding correction term for the *N* side is assigned to the PO triangles that have this label.

Note: The wedge must be between flat surfaces, and that all triangles with the label specified here must lie in the same plane.

KR Card

This card creates a mesh of surface triangles in the shape of circular region with or without a hole. It is also possible to create an elliptical region.

On the **Construct** tab, in the **Surfaces** group, click the **Ellipse (KR)** icon.

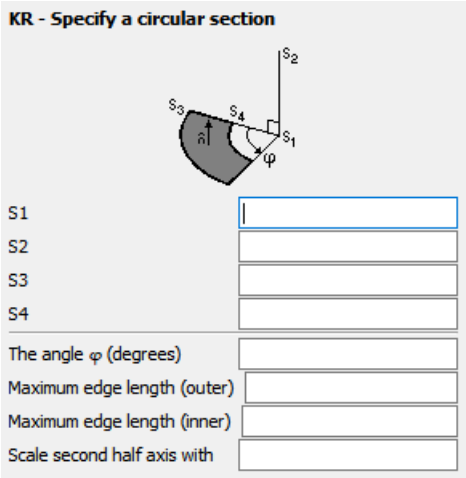


*Figure 745: The **KR - specify a circular section** dialog.*

Parameters:

S1 — The centre point of the circle.

S2 — A point that is located at any distance perpendicular to and above the plane of the circle.

S3 — A point on the outer arc.

S4 — If there is a value present for this parameter, then a circular ring is created. S4 must be located between S1 and S3.

The angle φ (degrees) — The angle subtended by the arc in degrees.

Maximum edge length (outer) — The maximum edge length of the triangles along the outer edge of the arc in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Maximum edge length (inner) — When a disk with a hole is created, this parameter defines the maximum edge length for the triangles along the inner edge of the arc in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Scale second half axis with — If this parameter is empty or is set to 1, a circular disk is created. If set to $\frac{b}{a}$, an elliptical disk is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S1–S3. It is recommended to generate elliptical disks with extremely small or

extremely large axial ratios with a CAD system as the distortion formulation used in PREFEKO may fail in these cases

The circle's plane is perpendicular to the line S1–S2. This length is arbitrary. The radius of the disc is given by the length between the points S3 and S1. The area that is to be subdivided (the shaded region in the figure for the KR card dialog) is generated by sweeping the edge S3–S1 around the axis S1–S2 through φ degrees in the mathematically positive sense. For φ = 360° a circle is obtained.

The fineness of the mesh is determined by the maximum edge length specified by the last IP card prior to the KR card. Along the arcs, accurate modelling of the geometry may require finer segmentation and the maximum edge length values specify the maximum edge length along the outer and inner (if applicable) arcs respectively. If any of these values are not specified the length specified with the IP card will be used on the corresponding arc.

The normal vectors of the triangles on the disk all point in the direction from S1 to S2.

Examples of the KR card usage

The following example meshes of a circular plate, a flat circular ring and a flat elliptical ring are all created using the KR card.

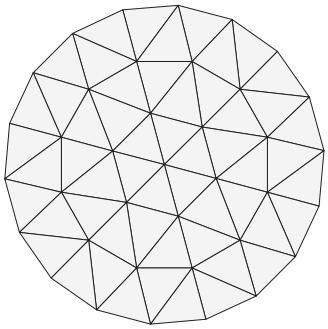


Figure 746: Example of a circular plate created with the KR card.

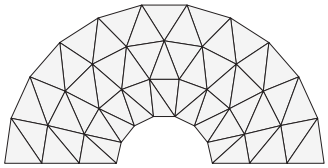


Figure 747: Example of a flat circular ring created with the KR card.

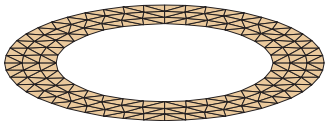


Figure 748: Example of an elliptical disk with a hole created with the KR card.

Related reference

IP Card

SF Card

KU Card

This card creates a mesh of surface triangles in the shape of a spherical section.

On the **Construct** tab, in the **Surfaces** group, click the **Sphere (KU)** icon.

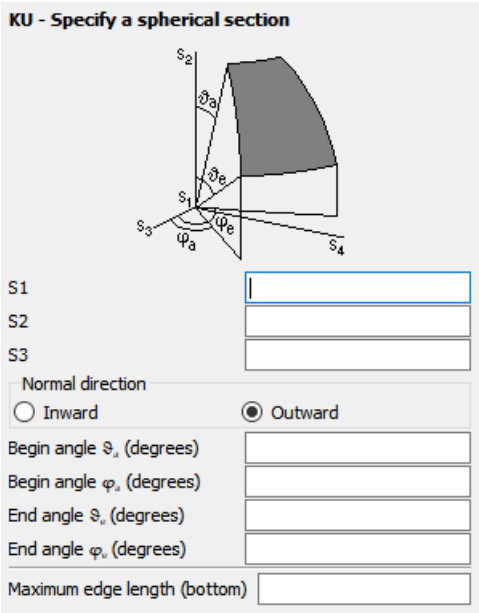


*Figure 749: The **KU - Specify a spherical section** dialog.*

Parameters:

S1	The centre of the sphere.
S2	A point that indicates the $\vartheta = 0°$ direction in a spherical coordinate system. The distance between S1 and S2 is the radius of the sphere.
S3	A point that indicates the $\vartheta = 90°$, $\varphi = 0°$ direction in a spherical coordinate system. The distance S1–S3 must be equal to the distance S1–S2.
Normal direction	The triangles can be created so that the normal vectors point **Outward** (away from the centre of the sphere) or **Inward** (towards it).
Begin angle ϑ_a (degrees)	The start angle ϑ_a in degrees of the spherical segment.
Begin angle φ_a (degrees)	The start angle φ_a in degrees of the spherical segment.
End angle ϑ_e (degrees)	The end angle ϑ_e in degrees of the spherical segment.
End angle φ_e (degrees)	The end angle φ_e in degrees of the spherical segment.

Maximum edge length (bottom)	The maximum length of the triangles along the curved edges in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

A complete sphere may be created with $\vartheta_a = \varphi_a = 0$, $\vartheta_e = 180°$ and $\varphi_e = 360°$.

An example of KU card usage:

The spherical segment is generated using the KU card.

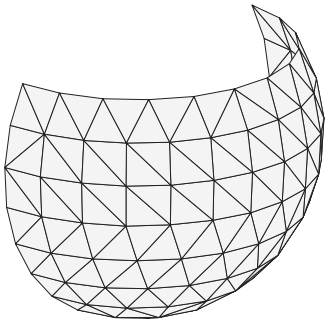


Figure 750: Example of a spherical segment created with the KU card.

Related reference

IP Card

SF Card

LA Card

With this card, labels are assigned to segments, triangles, polygonal plates, cuboids, uniform theory of diffraction cylinders and points.

On the **Home** tab, in the **Define** group, click the **Label (LA)** icon.

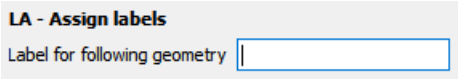


*Figure 751: The **LA - Assign labels** dialog.*

Parameters:

Label for following geometry	The label assigned to all geometry, for example, segments, triangles and tetrahedra defined in cards following this card.

In order to select the position of the feed (Ax cards)[101], the location of impedance loading (LD, LS, LP and LZ cards) or structures on which to apply the skin effect (SK cards), each segment, triangle, and so forth is assigned a label. Other applications include the selective transformation or copying of parts of geometry (TG card), and the output of currents on selected elements (OS card). Labels are also used to define triangles on which to apply physical optics (PO card).

All elements created after the LA card (by for example the BP card), are assigned the value or string specified label of the dialog. A different label is only assigned by a new LA card. All structures created before the first LA card (or if no LA card is present), is assigned the default label which is 0 (number zero).

Label names can be an arbitrary string using the characters A-Z, the digits 0-9 or also the underscore _. Labels may be manipulated using label increments and referenced using label ranges.

Related reference
PO Card

101. The definition Ax stands for any of the control cards, such as A0, A1, A2 and so forth.

MB Card

With this card, a modal port boundary condition may be applied on the boundary of a finite element method (FEM) region. A modal port essentially represents an infinitely long guided wave structure (transmission line) connected to a dielectric volume modelled with FEM.

In the **Source/load** tab, in the **Ports** group, click the **Modal boundary (MB)** icon.

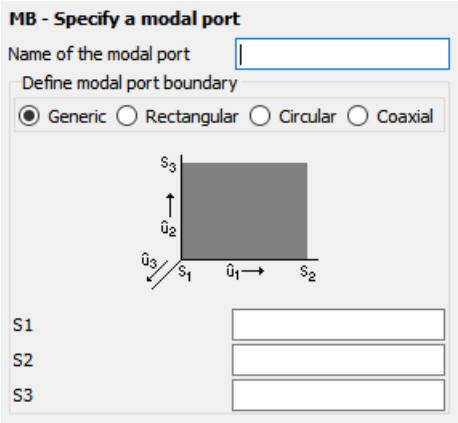


*Figure 752: The **MB - Specify a modal port** dialog.*

Parameters:

Name of the modal port — The label of the modal port.

Define modal port boundary

Rectangular — A rectangular waveguide cross section is used, which is defined by three points S1, S2, and S3 as follows: S1 is an arbitrary corner point, and S2 and S3 are two corner points which define the waveguide sides $\mathbf{u}_1$ (from S1 to S2) and $\mathbf{u}_2$ (from S1 to S3). The direction in which the mode is launched is given by $\mathbf{u}_3 = \mathbf{u}_1 \times \mathbf{u}_2$.

Circular — A circular waveguide cross section is used. The point S1 denotes the centre of the circular port, and the point S2 specifies the radius and start point for the angular dependency. A further point S3 must be perpendicular above the centre of the circular plate, so that the direction from S1 to S3 indicates the direction in which the waveguide modes are launched.

Coaxial — Here a feed of a coaxial waveguide with circular cross sections of both the inner and outer conductor can be specified.

The point definitions are the same as for the circular waveguide, except that an additional point S4 must be defined between S1 and S2 which specifies the radius of the inner conductor.

S1	Point S1 situated on the FEM modal boundary.
S2	Point S2 situated on the FEM modal boundary.
S3	Point S3 situated on the FEM modal boundary.

Note that the modal port is only available in conjunction with FEM applied to dielectrics. A FEM modal port can only be applied on the boundary of a FEM dielectric region, situated on a planar surface.

The technology behind a modal port is a two dimensional FEM eigensolver that computes the eigenvalues (modal propagation constants) and eigenvectors (modal electric field distribution) for the associated infinitely long guided wave structure.

The memory requirement can, for a modal port, be estimated from the number of tetrahedral faces on the modal port and the order of the solution. The eigensolver by default uses second order basis functions. Changing the solver settings to use first order basis functions for the FEM (FP card) will also apply to the modal port analysis. The number of unknowns for a first order solution is roughly double the number of modal port triangles, and for a second order solution, 7 times the number of modal port triangles. The memory requirement scales with N^2, where N is the number of unknowns.

The user should take note that memory and runtime scaling could become an issue with finely meshed modal port geometries. Note that when meshing modal ports, the default is to use second order basis functions on modal ports. Hence, a coarser mesh can be used than on the FEM/MoM boundary (where first order basis functions are always used).

Related tasks
Creating a FEM Modal Port (CADFEKO)
Related reference
FP Card

ME Card

This card must be used to distinguish the different media and to create segments and triangles (metallic or dielectric) within or on the surface of dielectrics solved with FEM or VEP as well as MoM/MLFMM.

In the **Home** tab, in the **Define** group, click the **Media** icon. From the drop-down list select the **Set medium (ME)** icon.

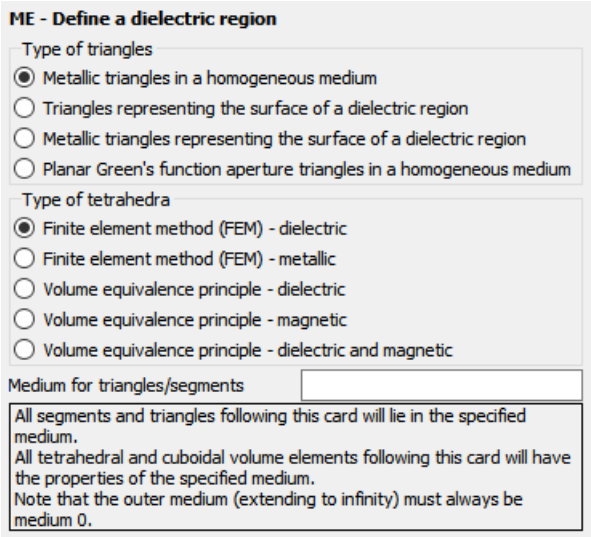


*Figure 753: The **ME - Define a dielectric region** dialog.*

Parameters:

Metallic triangles in a homogeneous medium

If this option is selected all the surface structures between this card and the next ME card are assumed to be fully contained inside the medium specified in **Medium for triangles/ segments**.

Triangles representing the surface of a dielectric region

If this option is selected all the surface structures created between this card and the next ME card are assumed to define the boundary between two media. Note that the user needs to provide the names of the media on both sides of the triangles. The normal vector points from **Medium A** to **Medium B**. For example consider a dielectric body of medium, DIELECTRIC constructed so that all the triangle normals point outward. Then **Medium A** is to be set to DIELECTRIC and **Medium B** to 0 (the number zero always represents the outer free space region).

Note: The dialog layout changes according to the selected options. For example the **Medium A** and **Medium B** text boxes are visible only when selecting the 2nd or 3rd options in the **Type of triangles** group.

Metallic triangles representing the surface of a dielectric region	If this option is selected all the surface structures created between this card and the next ME card are assumed to define a metallic boundary between two media. The selection of the sides is the same as for the non-metallic case, **Triangles representing the surface of a dielectric region**.
Planar Green's function aperture triangles in a homogeneous medium	All triangles generated after this card are planar Green's function aperture triangles in a homogeneous medium.
Finite element method (FEM) - dielectric	If this option is selected dielectric tetrahedral mesh elements will be solved with the FEM.
Finite element method (FEM) - metallic	If this option is selected metallic tetrahedral mesh elements are considered as part of the FEM solution. The medium label "Perfect electric conductor FEM" should be used to specify PEC metallic tetrahedra.
Volume equivalence principle - dielectric	If this option is selected dielectric tetrahedral mesh elements will be solved with the VEP.
Volume equivalence principle - magnetic	If this option is selected magnetic tetrahedral mesh elements will be solved with the VEP.
Volume equivalence principle - dielectric and magnetic	If this option is selected dielectric and magnetic mesh elements will be solved with the VEP.

All the wire segments that follow this card are assigned the properties of the medium in which they are located. Triangles are treated according to whether they are metallic triangles or triangles on the boundary of a dielectric object. Here the properties of the media are assigned to the respective sides.

All triangles and segments before an ME card represent metallic structures in free space. This is also the case when an input file does not have an ME card.

When using the FEM and meshing structures into tetrahedral elements or when using the VEP in connection with the MoM/MLFMM and meshing into cuboidal volume elements, then the selection in the **Type of triangles** group is not relevant. The specified medium will be used (**Medium A** if there are multiple media input fields).

The medium name can be an arbitrary string using the characters A-Z, the digits 0-9 or also the underscore '_'. Note that the outer medium must always be medium 0 (the number zero). The use of the ME card to create a simple dielectric sphere is shown in the `Kernel_scripting_examples` folder, `example_04.pre`. Note that the normal vectors of the sphere point outwards from medium 1 (the dielectric) to medium 0 (free space). The same folder also contains a more complex example (`example_23.pre`) of a dielectric cone (Medium 1) mounted on top of a metallic cylinder. There are three types of triangles involved:

- Metallic triangles in free space (Medium 0) on the bottom and side of the cylinder
- Metallic triangles also forming the border surface of the dielectric body on the lid of the cylinder (which is also the base of the dielectric cone)

- Dielectric triangles forming the surface of the dielectric body (the boundary between Medium 1 — the inner dielectric region — and Medium 0 — the free space outer region) on the top surface of the cone

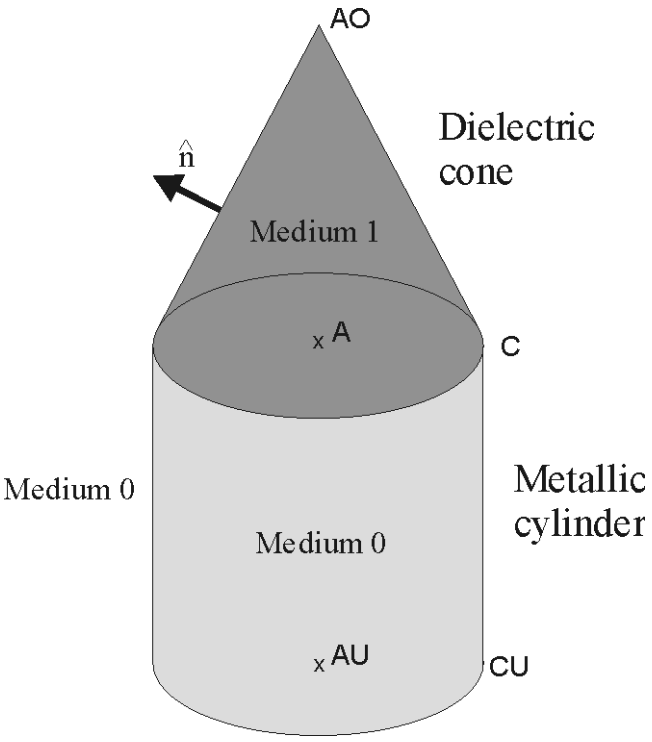


Figure 754: Example of a dielectric cone on top of a metallic cylinder to created with the ME card.

Related tasks
Applying a Dielectric to a Region (CADFEKO)
Related reference
KU Card

NC Card

This card defines the name to be used for the next configuration.

On the **Home** tab, in the **Structure** group, click the **Configuration name (NC)** icon.

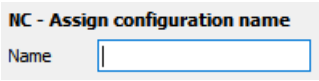


*Figure 755: The **NC - Assign a configuration name** dialog.*

Parameters:

Name	The name assigned to the configuration following the existing configuration.

NU Card

This card defines surface triangles representing a NURBS surface.

On the **Construct** tab, in the **Surfaces** group, click the **NURBS (NU)** icon.

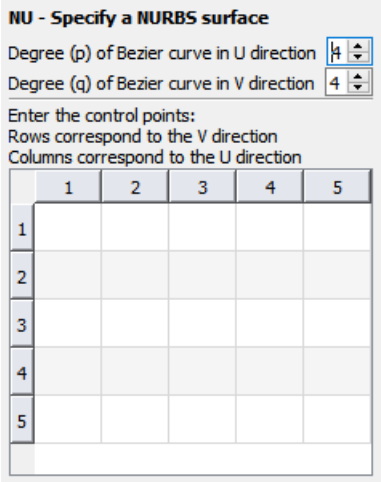


*Figure 756: The **NU - Specify a NURBS surface** dialog.*

Parameters:

Degree (p) of Bezier curve in U direction The degree of the Bézier curve in the U direction.

Degree (q) of Bezier curve in V direction The degree of the Bézier curve in the V direction.

Both p and q must be in the range from 1 to 4 where 1 is linear, 2 quadratic, and so forth. The control points are entered in the table, more or less representing their physical relation. There are p+1 rows and q+1 columns.

It is possible to create a triangular NURBS surface. In this case all control points on one side must be identical (use the same point). The weights of the control points are specified at the DP card. Note that for higher order Bézier curves, the surface does not pass through the control points except those on the corners.

Examples of NU card usage:

The “saddle” point shape is generated with the NU card using 4 and 6 control points respectively.

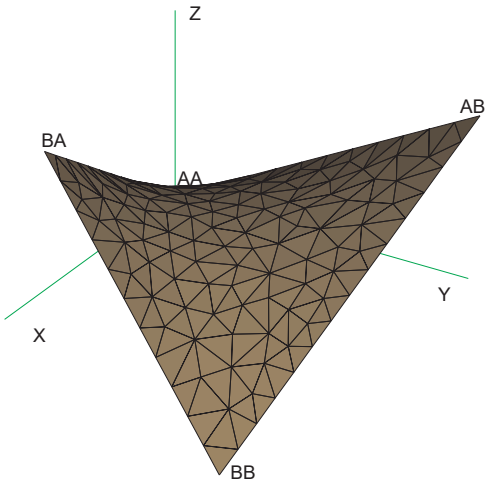


Figure 757: "Saddle" point example created with the NU card.

The linear-quadratic shape is also generated with the NU card using 4 and 6 control points respectively.

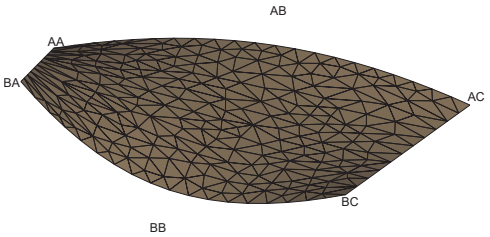


Figure 758: Linear-quadratic example created with the NU card.

NURBS may also be used to generate flat surfaces with curved edges. The section of a circular plate with a square hole is generated using the NU card.

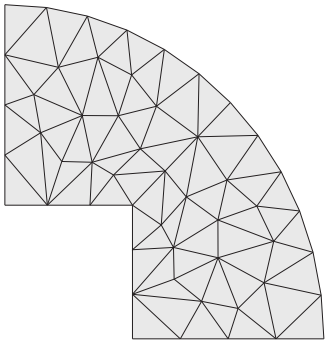


Figure 759: Flat surface with curved edges created with the NU card.

Related reference
NURBS (CADFEKO)

PB Card

This card can be used to generate a section of a parabolic reflector as shown in the figure on the card.

On the **Construct** tab, in the **Surfaces** group, click the **Paraboloid (PB)** icon.

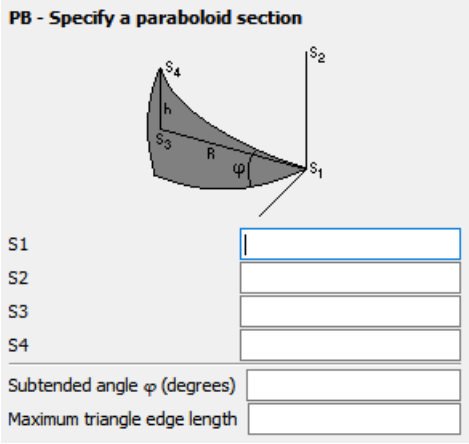


*Figure 760: The **PB - Specify a paraboloid section** dialog.*

Parameters:

S1	The centre of the paraboloid.
S2	A point perpendicular to the base plane and at any distance above the centre point.
S3	A point on the outer edge of the paraboloid, but on the base plane.
S4	A point in the plane S2–S1–S3, directly above S3 on the edge of the paraboloid.
Subtended angle φ (degrees)	The angle subtended by the arc of the parabolic reflector in degrees.
Maximum triangle edge length	Maximal edge length of the triangles along the outer edge of the arc in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

The radius R of the paraboloid is derived from the distance between the points S1 and S3, as can be seen in the figure in the card. The height is determined by the distance between points S3 and S4. The focal point f is determined by:

$$f = \frac{R^2}{4h} \tag{134}$$

Example of PB card usage:

The parabolic reflector as shown can be generated with the PB card.

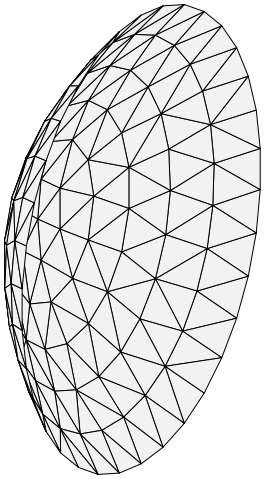


Figure 761: Example of parabolic reflector created with the PB card.

PE Card

This card defines the unit cell for a periodic boundary condition (PBC) calculation. The phase change between cells is specified with the PP card.

On the **Home** tab, in the **Planes / arrays** group, click the **Periodic boundary** icon. From the drop-down list, click the **Periodic boundary (PE)** icon.

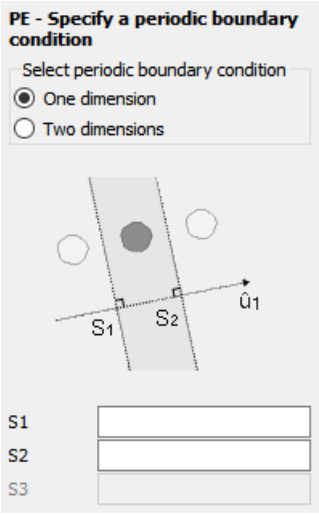


*Figure 762: The **PE - Specify a periodic boundary condition** dialog - **One dimension***

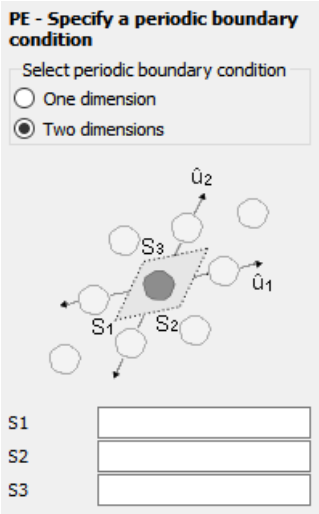


*Figure 763: The **PE - Specify a periodic boundary condition** dialog - **Two dimensions***

Parameters:

One dimension — One dimensional periodicity is used when the unit cell is repeated along a line. The unit cell definition is displayed in the dialog above depicting the **One dimension** option.

Two dimensions	Two dimensional periodicity is used when the unit cell is repeated to form a surface. The unit cell definition is displayed in the dialog above depicting the **Two dimensions** option.
S1	Name of the point S1.
S2	Name of the point S2.
S3	Name of the point S3. (Only required for two dimensional periodicity.)

The lattice vectors ($\hat{u}_1$ and $\hat{u}_2$) do not have to be orthogonal. Geometry may also cross the periodic boundary as long as it lines up with the geometry edge on the opposite side of the unit cell. These features allow a large number of problems to be solved.

The following restrictions apply for the current implementation of periodic boundary conditions:

- PBC is not available in conjunction with volume equivalence principle (VEP).
- Triangles and wires are not allowed to cross, but are allowed to touch the cell boundary.
- PBC can be used with the free space Green's function.
- PBC can only be used with MoM (both sequential and parallel), but not with the MLFMM, PO or UTD techniques.

Related tasks
Defining a PBC (CADFEKO)
Related reference
PP Card

PH Card

The PH card creates a triangular or quadrangular plate with a circular or elliptical hole as shown in the card. The hole can be used, for example, to attach a cylinder (ZY card) to the plate and it can be filled with the KR card.

On the **Construct** tab, in the **Surfaces** group, click the **Plate with hole (PH)** icon.

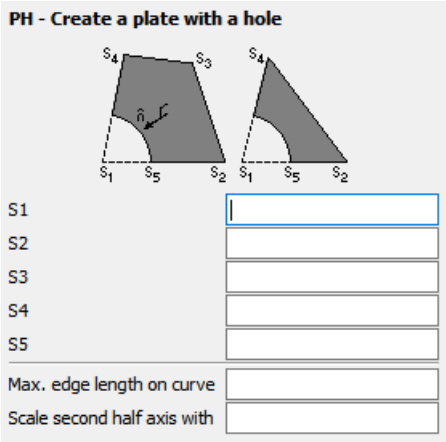


*Figure 764: The **PH - Create a plate with a hole** dialog.*

Parameters:

S1 — The corner where the hole is located (also the hole's centre).

S2 — The second corner of the plate.

S3 — The third corner of the plate. If this field is left empty, a triangular plate is created.

S4 — The fourth corner of the plate.

S5 — A point on the line S1–S2 that forms the starting point of the circle or ellipse bordering the hole. The special case where S5 is identical to S2 is supported, but due to the resulting geometry yields triangles with very small angles.

Max. edge length on curve — The maximum edge length of the triangles along the edge of the hole in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Scale second half axis with — If this parameter is empty or is set to 1, a circular hole is created. If set to $\frac{b}{a}$, an elliptical hole is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where *a* is the distance S1–S3. It is not recommended to generate the plate with a CAD system if the elliptical hole has an extremely small or extremely large axial ratio as the distortion formulation used in PREFEKO may fail for such cases.

Examples of PH card usage:

The PH card can be used to create the rectangular plate shown in Figure 765.

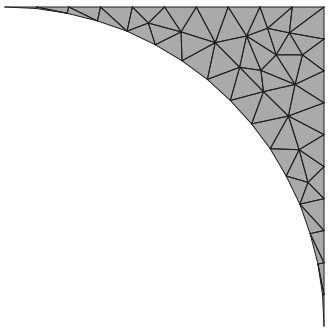


Figure 765: Example of a rectangular plate created with the PH card.

Note the extremely narrow triangles at the corners, as shown in Figure 766 and Figure 767, for a quadrangular and rectangular plate with the same elliptical hole. The triangular plate is obtained by leaving the field S3 empty.

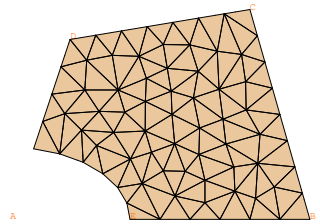


Figure 766: Example of a quadrangular plate with an elliptical hole created with the PH card.

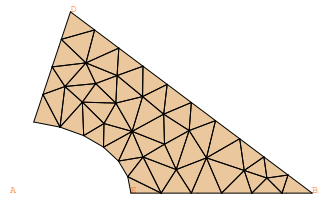


Figure 767: Example of a triangular plate with an elliptical hole created with the PH card.

Related reference

IP Card

KR Card

SF Card

ZY Card

PM Card

A surface mesh of triangles in the shape of a polygon is created by using the PM card. The PM card also allows the specification of interior mesh points. The PM card should generally be used in favour of other cards that create flat surface meshes with straight edges.

On the **Construct** tab, in the **Surfaces** group, click the **Polygon** icon. From the drop-down list, click the **Polygon (PM)** icon.

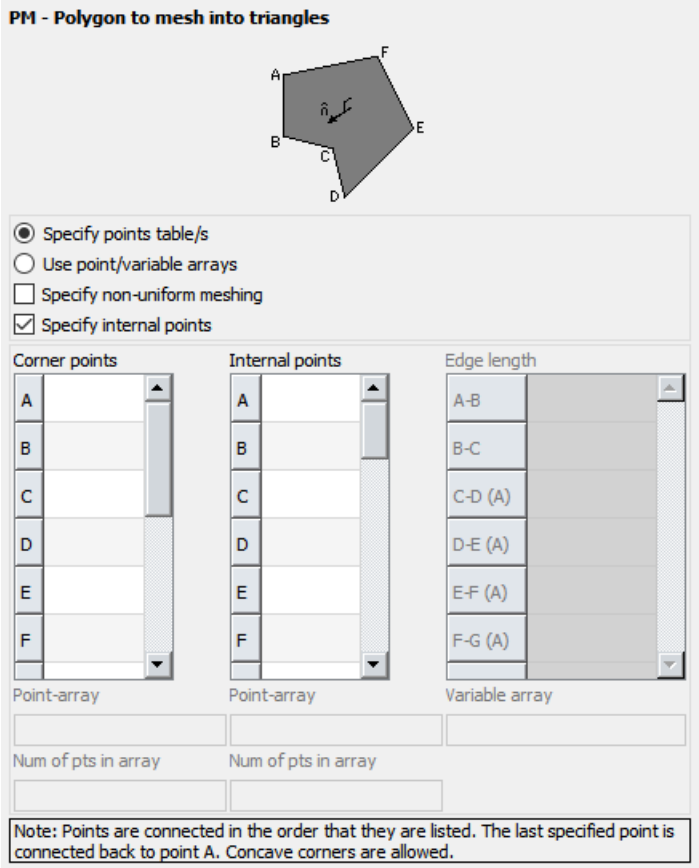


*Figure 768: The **PM - Polygon to mesh into triangles** dialog.*

Parameters:

Specify points table/s	This option allows data entry in a table with a maximum of 13 points.
Use point/variable arrays	This option allows data entry using arrays without a limit on the number of corner points.
Specify non-uniform meshing	Check this item to enable the table in which edge lengths for each edge can be entered.
Specify internal points	Check this item to enable the table in which internal points can be specified.

Corner points	Enter the points, previously defined with the DP card. The points can be entered in the rows or as point (node name) arrays.
Internal points	Any points internal to the structure where mesh points are required can be entered here. This is particularly useful for the connection points between surfaces and wires. The points can be entered in the rows or as point (node name) arrays.
Edge length	The mesh length on each edge can be set separately in this table. The edge lengths can be entered in the rows or as variable arrays. When using the table entry, any blank entries in this table will be meshed with the parameters set in the IP card. There may not be more entries in this table than in the first one. When using the array entry method, the variable array has to be the same length as the corner point array.

The polygon is defined by entering the points (previously defined with the DP card) on the corners of the polygon. A maximum of 13 corner points are allowed using the table entry method, but there is no restriction on the number of points when using the array entry method. The points are connected in the order that they are entered in the PM card. The array entry method allows specifying the number of elements, say n, in the array that should be used. Only the first n elements of the array is then used. Concave corners are allowed. The user can also specify a smaller or larger mesh size along certain edges.

Examples of PM card usage

Shown in Figure 769 is a plate with a concave corner created with the PM card — note the finer mesh along the edges from B to C and C to D.

A plate with three internal points, generated using a PM card is shown in Figure 770. Note that there are node points at Q1, Q2 and Q3.

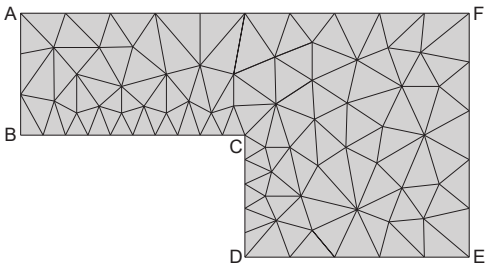


Figure 769: Example of a plate with a concave corner created with the PM card .

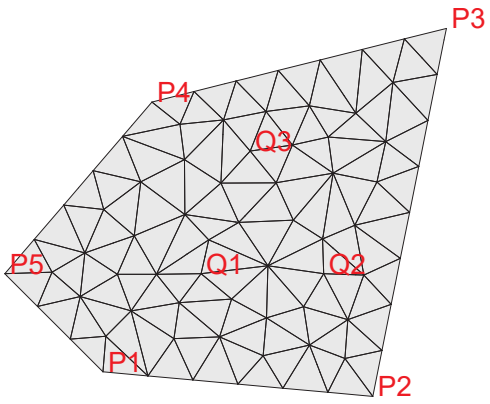


Figure 770: Example of a plate with internal mesh points created with the PM card.

Related reference

Polygon (CADFEKO)

BQ Card

DP Card

IP Card

PO Card

The PO card the application of the physical optics approximation is possible.

On the **Solve/Run** tab, in the **Rays** group, click the **Physical optics** icon. From the drop-down list, select the **Physical optics (PO)** icon.

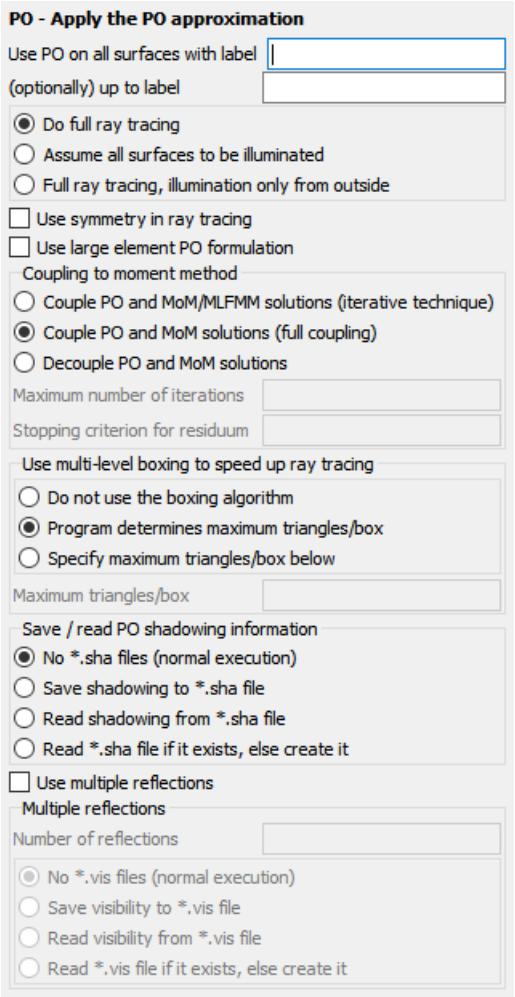


*Figure 771: The **PO - Apply the PO approximation** dialog.*

Parameters:

Use PO on all surfaces with label	Together with **(optionally) up to label** are used to specify the label or range of labels of all metallic/dielectric triangles that are treated with the physical optics approximation. If the second field is left blank, only the label specified in the first field is considered. See also LA and CB cards and also general discussion of label ranges.
Do full ray tracing	Normal, complete ray tracing is carried out.

Assume all surfaces to be illuminated	The ray tracing is switched off to save computational time. The assumption is made that all triangles on which the PO approximation is made are illuminated by the source and the moment method area. The side in relation to the normal vector is automatically determined.
Full ray tracing, illumination only from outside	Full ray tracing is done, but metallic triangles can only be lit from the side to which the normal vector is pointing. See note below.
Use symmetry in ray tracing	If full ray tracing is done, then symmetry can be used to reduce the computational time required to determine the shading. For electric and magnetic symmetry, this speed up is always used. If geometrical symmetry is used, then this item should be checked to utilise symmetry. It is possible to, for example, define half a plate and create the other half through geometric symmetry. An asymmetric object may then be placed in front of the plate. In this case symmetry should not be used in the ray tracing.
Use large element PO formulation	When this item is checked, electrically large triangular patches for PO are allowed (PO).
Couple PO and MoM/MLFMM solutions (iterative technique)	A hybrid iterative technique is used to determine the coupling between the MoM or MLFMM region, and the PO region.
Couple PO and MoM solutions (full coupling)	When this item is checked, the full coupling between the MoM region and the PO region is taken into account in the solution. The implication is that the currents in the PO region will have an effect on the current distribution in the MoM region.
Decouple PO and MoM solutions	When this item is checked, the coupling between the MoM region and the PO region is neglected. The implication is that the currents in the PO region has no effect on the current distribution in the MoM region. This option should lead to a saving in computational time and storage space and is especially useful when the PO region and the MoM is not directly adjacent.
Maximum number of iterations	The number of iterations limit for the iterative technique. This parameter is optional.
Stopping criterion for residuum	Termination criterion for the normalised residue when using the iterative method. Terminate with convergence when the normalised residue is smaller than this value. This parameter is optional.
Use multi-level boxing to speed up ray tracing	The ray tracing required for the physical optics is accelerated by recursively subdividing the problem domain, a so called “multilevel tree”. It must balance memory requirement against speed-up — both increase as the number of levels increases. The number of levels is determined by specifying the number of

triangles at the lowest level. The user can specify not to use this algorithm, for the program to determine maximum triangles/box or to specify this number manually. When a number is specified manually, it should be greater than 2 and at least a factor 10 less than the number of triangles in the problem. In general these options should only be set by advanced users.

Save/read PO shadowing information

For the PO formulation the information which triangles are illuminated and which are shadowed from the sources is required. Normally Solver computes this each time, but there is an option to store this to a file and load again to save time when the geometry stays constant (say for multiple runs using different frequencies). The options are:

No `.sha` files (normal execution)	Shadowing information is not stored or read — the default behaviour.
Save shadowing to a `.sha` file	The PO shadowing information is written to a `.sha` file for later reuse.
Read shadowing from a `.sha` file	The PO shadowing information is read from the `.sha` file, for example, the ray-tracing part is skipped. For large models this can result in considerable time saving.
Read `.sha` file if it exists, else create it	If a `.sha` file exists, the PO shadowing information is read from this file. Otherwise the information is calculated and saved in a `.sha` file for later use.

Use multiple reflections

When this item is checked, multiple reflections are considered for the ray tracing. The number of reflections that must be considered is set in the **Number of reflections** dialog. This parameter determines the number of reflections to be taken into account for triangles with labels in the specified range. (For example, the **Number of reflections** must be at least 2 to calculate the scattering from a dihedral and at least 3 for a trihedral.) Increasing the number of reflections that must be considered significantly increases computation time, and this should only be done based on physical considerations.

Visibility information

The visibility information related to multiple reflections can be saved to reduce the computation time for future runs. There are four options that can be selected with respect to saving the multiple reflection visibility information:

No `.vis` files (normal execution)	Visibility information is not used or stored — the default behaviour.

Save visibility to a *.vis file	The PO visibility information is stored in a `.vis` file for later reuse.
Read visibility from a `.vis` file	The PO visibility information is read from the `.vis` file, for example, the calculation of the visibility information is skipped. For large models this can result in considerable time saving.
Read `.vis` file if it exists, else create it	If a `.vis` file exists, the PO visibility information is read from this file. Otherwise the information is calculated and saved in a `.vis` file for later use.

The physical optics (PO) approximation can only be used for certain structures. Structures where the antenna is situated in front of a reflector are well suited. Then PO can be used for the triangles that form the reflector. This results in a large reduction in computational time and memory for electrically large objects.

Note that the ray tracing options and the number of reflections can be specified on a per label basis, by using multiple PO cards. All other parameters can only be specified once. For the global parameters, the values of the last PO card will be used.

Using **Full ray tracing, illumination only from outside** has two applications:

- Acceleration of the PO ray tracing with closed bodies (the normal vector must then point outward), since the dot product of the normal and propagation vectors can be used to quickly determine if a triangle is to be used in the ray tracing. In this case the closed model must be constructed with the normals pointing outward.
- In, for example the MoM/PO hybrid method on a closed body, the MoM region (such as an antenna) can be prevented from illuminating the PO region from inside.

A basis function that has been assigned to an edge between two triangles will only be solved with the PO, if the PO approximation has been declared for the labels of both triangles.

The metallic PO region must be perfectly conducting, for example, no losses are allowed. Dielectric coatings (see CO card) and thin dielectric sheets (see SK card) can, however, be treated with the PO approximation.

Large element PO

The following needs to be considered when regarding mesh size for large element PO:

- The triangular patch must be a large smooth area with no discontinuities of incident field (for example, close to a point source). The surface containing the triangular patch should also not contain any sharp corners.
- The shadowing is done on the triangular patch level, for example, smaller mesh elements are required close to the shadow boundaries.
- For near field calculations (electric or magnetic) or potentials, the maximum triangle mesh size is limited to 2λ.
- If only far field calculations are requested, the size of the mesh elements is not limited.

- If near fields calculations are requested, a warning will be given if the mesh size is set equal to or greater than $2\lambda_2$. An error will be given if the mesh size is set equal or greater than $6\lambda^2$.

When using large element PO, the following limitations apply:

- As in the case of normal PO, large element PO may not be used in conjunction with UTD.
- The PO solution only or the MoM/PO hybrid solution is supported, but it is then required that the MoM and PO regions are either coupled through the iterative method or decoupled (full coupling is not supported).
- In order to get one wave propagation factor k_n in the PO region, sources (for example, MoM basis functions) must be "close" together.
- Large element PO triangles must be PEC. Note that large element PO may be used in conjunction with normal PO. Connections between normal PO and large element PO are allowed.
- No edge/wedge correction terms are allowed for labels where large element PO is used.
- No extraction of singular terms for near field computations, for example, near fields are inaccurate closer than $\frac{\lambda}{20}$ .. $\frac{\lambda}{10}$ from surfaces.
- The export of surface currents and visualisation in POSTFEKO are not supported for large element PO regions.
- Multiple reflections are not allowed in the model if a large element PO region is present. When multiple reflections are present, RL-GO and faceted UTD should be the preferred method.
- Periodic boundary conditions are not allowed to be used in conjunction with large element PO.

The following are supported for large element PO:

- A BO ground for example a PEC plate or a real ground.
- Symmetry may be used (also for ray-tracing).
- Shadowing effects but not when located inside of individual large triangles.
- Parallel processing (for example, parallel ray tracing, parallel field computations and others.)
- Near field (E, H and potentials) computations as well as far field computations including RCS are allowed.
- All impressed sources my be used in conjunction with the large element PO.

Related concepts
PO and LE-PO Parameters (CADFEKO)
Related tasks
Solving Faces with PO (CADFEKO)
Related reference
CB Card
CO Card
LA Card
SK Card

PT Card

This card defines a port connected to the full wave model. Reference this port when defining sources, loads, networks and transmission lines.

In the **Source/load** tab, in the **Ports** group, click the PORT **Port (PT)** icon.

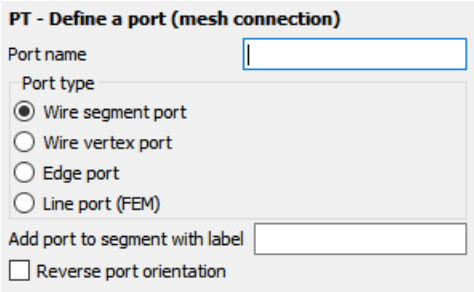


*Figure 772: The **PT - Define a port (mesh connection)** dialog.*

Parameters

Wire segment port	**Add port to segment with label**	Specifies the segment label where the port is placed.
	Reverse port orientation	Select this option if the polarity of the port is to be reversed.
Wire vertex port	**Add port to segment with label**	Specifies the segment label where the port is placed. The port has to be located at a node, either between two segments, or between a segment and a triangle, ground plane or polygonal plate.
	Port at start of segment	Select this option if the port location is at the start of the wire segment with a matching label.
	Point at end of segment	Select this option if the port location is at the end of the wire segment with a matching label.
	Use default feed direction (like basis functions)	If this option is selected, the positive feed direction is consistent with the basis function setup. For wire/surface junctions (UTD plates, infinite planes with the BO ground, or meshed triangle surfaces), this direction is away from the wire onto the surface. For wire connections between two segments, this direction is from the

		segment with the lower index to the segment with the higher index.
	Positive feed direction like wire segment orientation	If this option is set, then the positive feed direction is like the orientation of the wire segment with the specified label
	Negative feed direction like wire segment orientation	If this option is set, the negative feed direction is opposite to the orientation of the wire segment with the specified label.
Edge port	**Edge port between regions with multiple labels**	The edge port is placed on the edge between the regions with labels specified in **Negative side** and **Positive side**. The positive source direction is from the **Negative side** towards the **Positive side**. Use the field **Maximum number of labels on a side** to increase the number of rows available in the table.
	Edge port connected to ground/UTD	The edge port is placed on the edges of metallic triangles with the labels specified in the **Negative side** or **Positive side**. The edges of these triangles are connected to UTD surfaces or to a PEC ground plane (as specified with a BO or GF card).
	Edge port of microstrip between two points	The edge port is placed on all edges on a line between points (previously specified with DP cards). The points are specified in the **Start point of edge** and **End point of edge** dialogs. A GF card with a conducting ground must be present.
	Meshed surface represents positive feed side	By default, the feed direction is such that the meshed surface represents the negative feed side. The vector direction of the current is then towards the UTD or ground. When this option is checked, the feed orientation is reversed.
Line port (FEM)	Coordinates of the start and end points in the model unit.	

Note: Scale coordinate values using the SF card.

PY Card

The PY card defines (by specifying the corner points) a polygonal plate surface to which the UTD formulation is applied.

On the **Construct** tab, in the **Surfaces** group, click the **Polygon** icon. From the drop-down list, click the **Unmeshed polygon (PY)** icon.

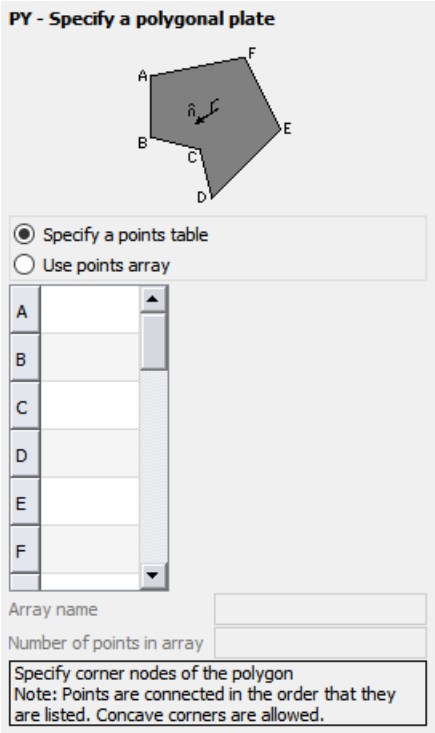


*Figure 773: The **PY - Specify a polygon plate** dialog.*

Parameters:

Specify a points table

This option allows data entry in a table with maximum of 26 points.

A	The first corner point of the polygon.
B	The second corner point of the polygon.

Use points array

This option allows data entry using arrays without any limit on the number of corner points.

Array name	The name of the point (node name) array.
Number of points in array	The number of elements, for example, n of the point (node name) array to use. Only the first n elements of the array will be used.

A maximum of 26 corner points are allowed using the table entry method, but there is no restriction on the number of points using the array entry method. The points are connected in the order that they are entered in the PY card. The corner points have to be defined prior to the PY card by a DP card.

Example of PY card usage

This card can be used to generate the polygon (in this case a triangle) shown in Figure 774. Note that this triangle is not meshed, as the result would be if the BT or PM cards had been used.

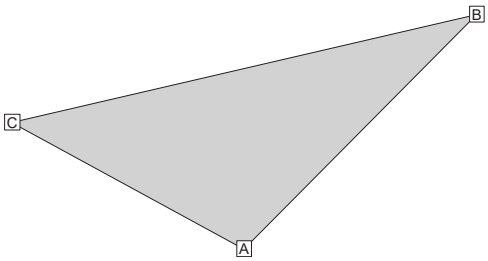


Figure 774: Example of a triangle created with the PY card.

Related reference

BT Card

DP Card

PM Card

QT Card

This card is used to create a dielectric or magnetic cuboid, meshed into smaller tetrahedral volume elements solved with the VEP or FEM.

On the **Construct** tab, in the **Volumes** group, click the **Cuboid** icon. From the drop-down list, click the **Cuboid (QT)** icon.

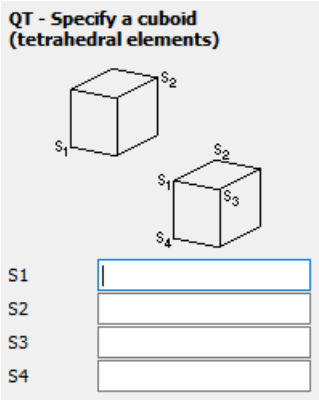


*Figure 775: The **QT - Specify a cuboid (tetrahedral elements)** dialog.*

The meshing parameters as set with the IP card are used, and the medium as set with the ME card is assigned to all meshed tetrahedral elements.

Parameters:

S1	First corner of the cuboid.
S2	Opposite corner of the cuboid if aligned with the principal planes, otherwise one of the corners adjacent to the first corner.
S3	Optional third corner of the cuboid, adjacent to the first.
S4	Optional fourth corner of the cuboid, adjacent to the first.

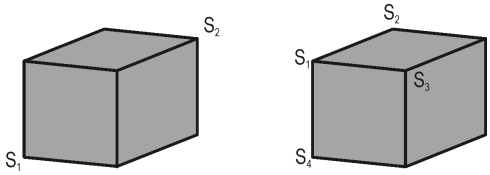


Figure 776: Sketch illustrating the use of the QT card.

Note that metallic structures are allowed with the FEM (metallic surfaces also inside the FEM region or on the FEM boundary, but wires only outside). But using dielectric bodies inside the MoM region (VEP or SEP) at the same time is not supported.

Example of QT Card Usage:

The dielectric cuboid shown in the figure below is generated using a QT card.

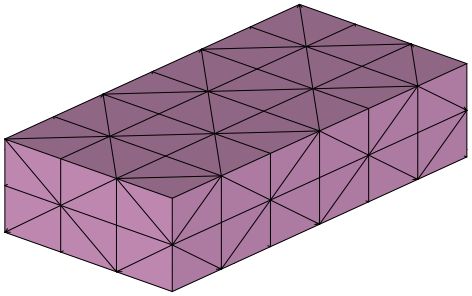


Figure 777: Example of a dielectric cuboid created with the QT card.

QU Card

This card creates a dielectric or magnetic cuboid, meshed into smaller cuboidal volume elements, for solving with the volume equivalence principle in the MoM.

On the **Construct** tab, in the **Volumes** group, click the **Cuboid** icon. From the drop-down list, click the **Cuboid (QU)** icon.

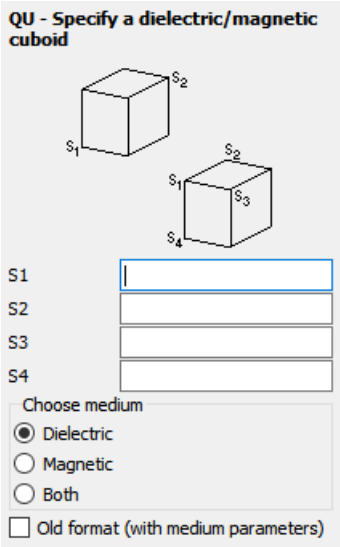


*Figure 778: The **QU - Specify a dielectric/magnetic cuboid** dialog.*

The mesh size is set with the IP card and the medium, specified with the ME card, is used for all meshed cuboidal elements.

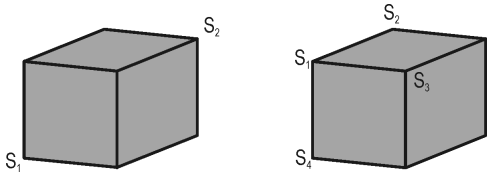


Figure 779: Sketch illustrating the use of the QU card.

Parameters:

S1	First corner of the cuboid.
S2	Opposite corner of the cuboid if aligned with the principal planes, otherwise one of the corners adjacent to the first corner.
S3	Optional third corner of the cuboid, adjacent to the first.
S4	Optional fourth corner of the cuboid, adjacent to the first.
Choose the medium	Select here whether the cuboid is **Dielectric** or **Magnetic** or **Both** (this is always with respect to the environment, if the

relative permittivity ε_r of the cuboid material differs from the environment, then this is a dielectric cuboid).

Old format (with medium parameters)

The old card format specified the material parameters directly at the QU card. This format is retained for backwards compatibility purposes.

> **Note:** It is not recommended to use the old card format for new models.

The current format requires that, prior to the QU card, the DI card is used to define the material parameters and the ME card is used to specify the type of tetrahedra. When checking this option, the panel layout will change to the old format (depending on the selection whether dielectric or magnetic cuboid) so that the material parameters can be entered. Feko then uses a compatibility mode and creates artificial media with names QU_MED_xx where "xx" is an index. When working on old models and pressing F1 in EDITFEKO on an existing QU card, the old format dialog will be opened automatically.

> **Note:** Dielectric bodies treated with the volume equivalence principle (using cuboids) cannot be used simultaneously with dielectric bodies treated with the surface equivalence principle or the finite element method or with special Green's functions.

Example of QU card usage

The dielectric cuboid (mesh of cuboids) shown below is generated with a QU card.

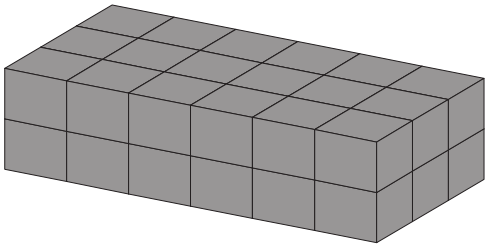


Figure 780: Example of a dielectric cuboid (mesh of cuboids) created with the QU card.

Related reference

DI Card

IP Card

ME Card

RM Card

The RM card provides a sophisticated remeshing and adaptive mesh refinement facility. Most types of meshes (surface mesh with triangular patches, wire segment mesh, cuboidal volume elements) created by any option supported in Feko (for example, direct creation in PREFEKO with cards, but also import from NASTRAN, FEMAP, PATRAN and the rest) can be used as a basis, and one can then apply either a local or a global mesh refinement. Unfortunately in Feko Suite 5.4 there is still a restriction that tetrahedral volume elements as used for the FEM cannot be refined with the RM card.

On the **Construct** tab, in the **Modify** group, click the **Refine mesh (RM)** icon.

A local mesh refinement refers to a point or a line as reference, or also to a complex cable harness geometry, which is defined directly by importing the corresponding `.rsd` file from CableMod or CRIPTE.

Note that similar to other Feko cards, the RM card applies only to what follows in the `.pre` file after the line where the RM card has been read. So for instance if one wants to import a mesh from a NASTRAN file via the IN card and do a mesh refinement during the import, then one first has to use the RM card, then followed by the IN card.

Multiple RM cards can be used, for instance if there are multiple areas in a model where the mesh shall be refined locally. Or also if we use a mesh refinement with respect to one point, the mesh size increases linearly with distance, and by adding another RM card with a global mesh refinement option, a threshold can be set.

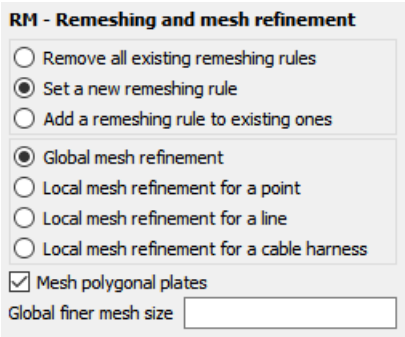


*Figure 781: The **RM - Remeshing and mesh refinement** dialog.*

Parameters:

Remove all existing remeshing rules	Clear all previously defined remeshing rules (for example, the behaviour is as if no RM card was read). This option is useful if after having imported a structure using mesh refinement, one wants to import another structure or create objects directly in PREFEKO, and for these new structures no mesh refinement shall be used. If this option is checked, all the other parameters are ignored.
Set a new remeshing rule	Set a new remeshing option (previously read RM cards will be discarded).

Add a remeshing rule to existing ones	Add a remeshing rule to the already defined ones (for example, existing RM card rules will be kept, the new rule will be added to these).
Global mesh refinement	Global mesh refinement using the specified finer mesh size.
Local mesh refinement for a point	Here an adaptive mesh refinement is performed to obtain a finer mesh close to a point. The point must have been defined before with a DP card, and its name is passed in the input field for the reference point. Note that this point can be located arbitrarily in space, there is no need for this to be on the meshed structure.
Local mesh refinement for a line	Here an adaptive mesh refinement is performed to obtain a finer mesh close to a line. The line is defined by its start and end point. These two points must have been created before with DP cards. Multiple simultaneously active RM cards can be used to perform a mesh refinement with respect to an arbitrary polygonal shaped line, composed by multiple straight line segments.
Local mesh refinement for a cable harness	With this option one can perform a local mesh refinement close to a cable harness. The cable harness geometry is specified by a CableMod/CRIPTE `.rsd` file. The file name of this must be entered into the respective input field (visible only when this option has been selected).
Mesh polygon plates	As a special feature, the RM card also allows to mesh unmeshed polygonal plates (which are used in Feko for the UTD) during the import. This can be very useful if, for example, a UTD model is imported from FEMAP using then boundary surfaces, and instead of the UTD a MoM or MLFMM or PO solution shall be conducted (where a mesh is required).
Reference point	When using local mesh refinement with respect to a point, then here the name of this point is entered (the point must have been specified before at a DP card).
Start point of line	When using local mesh refinement with respect to a line, then here the name of the start point of the line is entered (the point must have been specified before at a DP card).
End point of line	When using local mesh refinement with respect to a line, then here the name of the end point of the line is entered (the point must have been specified before at a DP card).
CableMod/CRIPTE `.rsd` file	When using local mesh refinement with respect to a cable harness, then here the file name of the CableMod/CRIPTE `.rsd` file is specified.

Global finer mesh size	When a global mesh refinement is used, then this is the new mesh size which shall be applied. Mesh coarsening is not supported, only mesh refinement. So when the new mesh size is larger than the existing mesh size, simply no mesh refinement will be done.
Distance D1	Reference distance d_1 for the mesh refinement, discussed below.
Mesh size at D1	Mesh size s_1 at the reference distance d_1, discussed below.
Distance D2	Reference distance d_2 for the mesh refinement, discussed below.
Mesh size at D2	Mesh size s_2 at the reference distance d_2, discussed below.

The mesh sizes specified for the global or local mesh refinement apply to all types of geometry (for example, triangles, wires, cuboidal volume elements) in the same manner. This is not a principal restriction. If different refinement options are desired say for wires and triangles, one can use one RM card, create or import say just triangles, and then use another RM card and after this create or import just wires etc.

If one RM card specifies a global mesh refinement, then the local mesh size is readily given by the global finer mesh size. If one does local mesh refinement with respect to a point, then first the distance of the mesh element to this point is determined. Similarly for a line or a cable harness, the shortest distance from the mesh element to this line or cable is determined. If we assume that this distance is d, then the local mesh size s is given by the equation

$$s = s_1 + \frac{s_2 - s_1}{d_2 - d_1}(d - d_1) \tag{135}$$

This means that for a distance $d = d_1$ we get the mesh size $s = s_1$, and for the distance $d = d_2$ the mesh size is $s = s_2$. For any other distances (smaller than d_1, in between d_1 and d_2, or also larger than d_2) a linear interpolation is used by means of the formula above. Thus the linear increase of the mesh size also continues for larger distances, but one should keep in mind that the RM card can only do a mesh refinement and no mesh coarsening, for example, as soon as for larger distances the remeshing option exceeds the already used mesh size of the original model, simply nothing will happen. Although not required, it is often useful to set the mesh size s_2 identical to the global already existing mesh size, then the parameter d_2 readily controls the region where a local mesh refinement is desired (for example, for distances d larger than d_2 the original mesh will be kept).

It shall also be mentioned here that if a CableMod or CRIPTE `.rsd` file is imported, in order to evaluate distance d between each mesh element and the cable harness in the right base unit (if an SF card scaling factor is set this can be for instance mm), the cable harness coordinates have to be scaled accordingly. Thus the SF scaling factor must be known before the RM card can be used. PREFEKO will give an error if an SF card is read and a RM card was processed before. The user must then just move the SF card in front of the RM card in the `.pre` file.

Examples of RM card usage

A first example is shown in Figure 782 with the original mesh on the left hand side and on the right hand side the result of a local mesh refinement with respect to a point is given. For the example in

Figure 783 a local mesh refinement with respect to two lines is used (for example, two simultaneously active RM cards).

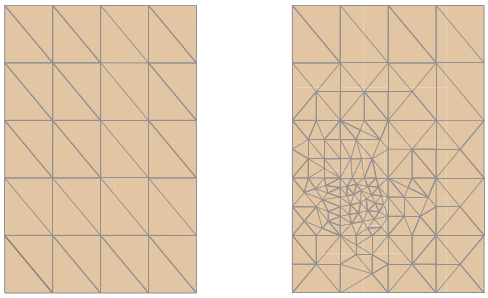


Figure 782: Example of local mesh refinement with respect to a point created with the RM card.

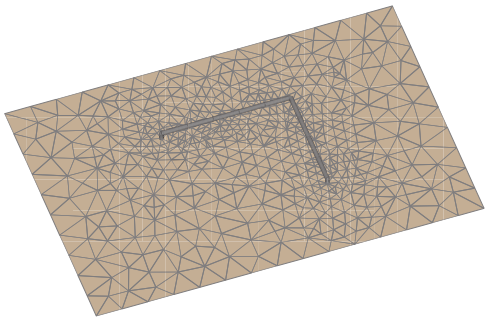


Figure 783: Example of local mesh refinement with respect to lines created with the RM card.

Related tasks

Refining Mesh Around a Point (CADFEKO)

Refining Mesh Along a Polyline (CADFEKO)

Related reference

DP Card

IN Card

SF Card

SF Card

This card scales the geometric data.

On the **Construct** tab, in the **Modify** group, click the **Scale (SF)** icon.

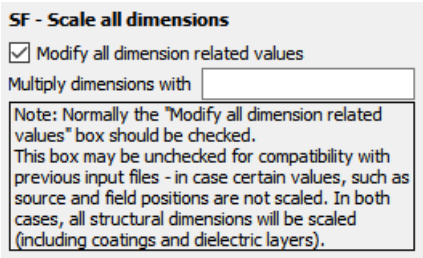


*Figure 784: The **SF - Scale all dimensions** dialog.*

This card is useful for specifying models in a convenient unit (for example cm) and specifying a scaling factor once.

Note: By default Feko reads all dimension related values as if specified in metres.

Parameters:

Modify all dimension related values	If this item is checked all geometrical dimensions are scaled. If unchecked, not all coordinate values are scaled (for example, the positions of near field calculations, see the list below). This should only be unchecked for backwards compatibility with old input files.
Multiply dimensions with	The scale factor. For example, if this is set to 0.001, all dimensions are entered in mm.

Only one SF card is allowed in the input file. This is global and can be positioned anywhere. However, since it is a geometry card it must be before the EG card.

If **Modify all dimension related values** is unchecked, the following is scaled:

- Coordinates of the corner points of the triangular surface elements.
- Coordinates of the corner points of the segments.
- Radii of the segments.
- Coordinates of the corner points of the cuboids.
- Named points defined by the DP card.
- Radii of the all the layers when the Green's function for a homogeneous or layered dielectric sphere is used.
- Thickness of the layers when the Green's function for a planar, multilayered substrate is used.
- Coordinates of the corner points of the polygonal plates.
- Coordinates, radii and dimensions of UTD cylinders.
- Coordinates of the corner points of tetrahedral volume elements.
- Thickness of dielectric surface elements.

- Radius and thickness of a wire coating.
- Coordinates of wedges and edges in the PO region.
- Coordinates of the Fock region.
- Transmission line length and end point coordinates.
- The dimensions of the aperture used in the AP card, as well as the specified gridlines, and the amplitudes of the A5 and A6 dipoles which depend on the incremental areas.
- The three coordinates defining a near field aperture receiving antenna, as well as the specified gridlines.
- The three coordinates defining the position of a PCB source (AJ card).
- The three coordinates defining the position of an impressed current source defined using model solution coefficients (AM card).
- The three coordinates defining the origin in the local coordinate system (MD card).
- The variable **Maximum identical distance**, specified with the EG card, which controls whether two points are considered to be coincidental in space or separate.
- FDTD geometrical parameters, for example, coordinates of the voxel corner points.
- General non-radiating network end point coordinates.
- Cables – all geometrical parameters, for example, paths coordinates, shield / cross-section dimensions and thickness and probe position along the path length.
- Windscreens, for example, the top offset of a windscreen.

If **Modify all dimension related values** is checked all geometrical dimensions and coordinates are scaled. This includes, in addition to the parameters listed above, the following:

- The coordinates of the source point specified in the excitation cards A1, A2, A3, A5, A6, A7 (if the selection is not made by label).
- Coordinates of the origin of the radiation pattern specified with the AR card.
- Coordinates of the start and end points of the impressed currents for the AI and AV source cards, as well as the wire radius specified with these cards.
- Radii of the coaxial feed in the A3 card.
- Positions where the near field is calculated with the FE card.
- Offset in the near field calculation.
- Coordinates of the plane position for a transmission / reflection coefficients request.
- Coordinates of the L2 and L4 loads.
- Coordinates of the start and end points of the impressed electric current filament (AF and LF cards).
- Coordinates of the origin of the spherical modes source specified with the AS card.
- Coordinates defining the waveguide aperture (AW card).
- The “middle” coordinate for AI and AV sources.
- SAR (when the position is specified).
- Offset in the far field calculation (only near field– OF card).
- Receiving antenna coordinates of the origin for:
    - Far field pattern

- Spherical mode pattern

> **Note:** If all data is specified in the unit of mm, with the scaling set to 0.001, all input values are interpreted as mm.

This also applies to the segmentation parameters (IP card) and possible translations (TG card).

Related tasks
Scaling Geometry (CADFEKO)
Related reference
CO Card
EG Card
GF Card
IP Card
OF Card
SK Card
TG Card
TL Card

SL Card

This card defines, for the combined MoM/MTL, the transitioning point from circuit elements (defined in a cable schematic) to the full wave model (defined using DP cards).

In the **Solve/Run** tab, in the **Cables** group, click the **Schematic link (SL)** icon.

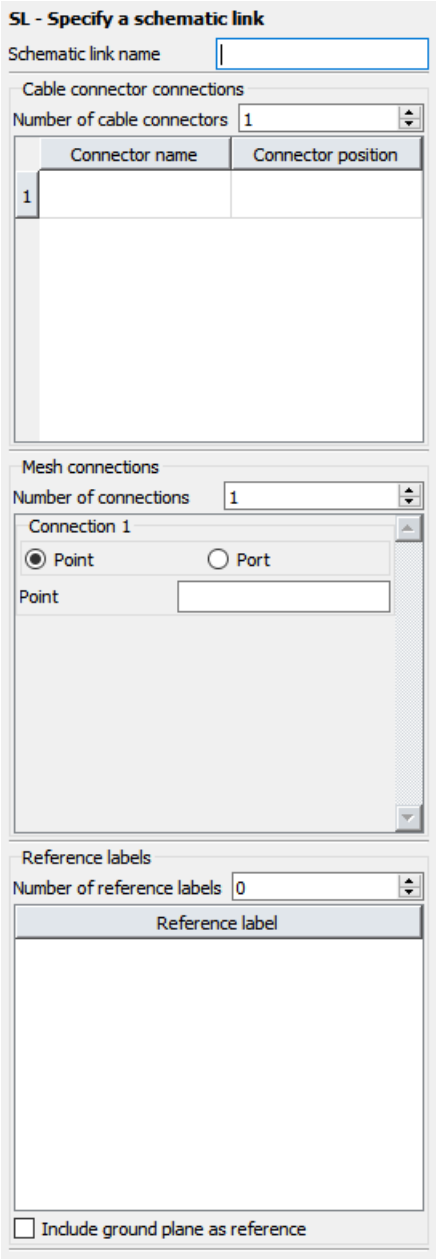


*Figure 785: The **SL - Specify a schematic link** dialog.*

Parameters:

Schematic link name The name of the interface definition between the schematic and the 3D full wave model.

Number of cable connectors

The number of cable paths terminating in the schematic link.

Connector name	The path start/end connector name. Start and end points of a cable path section are uniquely identified using the **Connector at start** and **Connector at end** labels at the CS card.
Connector position	The name of the data point coinciding with the path start/end connector name (defined at the CS card).

Number of connections

The combined number of geometric connections between the circuit elements and the full wave model. A connection can be either to a mesh vertex or a port.

Point

The name of the point connected to the full wave model (mesh vertex defined using the DP card). The mesh vertex point can be either a metallic triangle vertex, segment vertex or PEC infinite ground plane (defined using the BO card).

Port

The name of the port connected to the full wave model (defined using the PT card). The port negative terminal should coincide with the reference surface.

Number of reference labels

The number of labels defining the common reference surface, translating to a single node of reference on the cable schematic.

Reference label

The name of the label(s) defining the common reference.

Note:

- If one of the connections references a port, a reference label is required.
- If none of the connections references a port, a reference label is optional.

Include ground plane as reference

Select this check box to include the ground plane as part of the common reference surface, translating to a single node of reference on the schematic.

Note: The triangles (labels) and PEC ground plane that define the cable schematic link common reference surface should not be disjoint.

Related reference

BO Card

CS Card

DP Card
PT Card

SY Card

This card defines symmetry planes to reduce computation time and to reduce the number of elements to be meshed.

On the **Solve/Run** tab, in the **Solution settings** group, click the **Symmetry (SY)** icon.

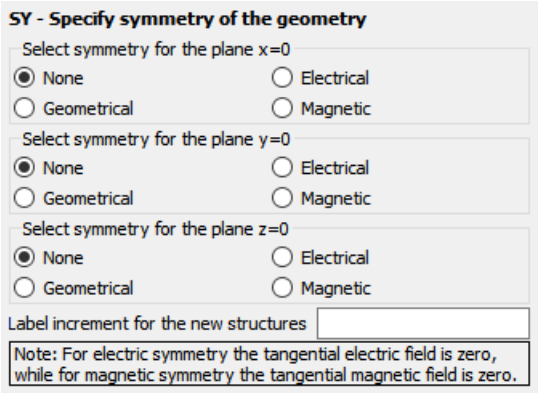


*Figure 786: The **SY - Specify symmetry of the geometry** dialog.*

Parameters:

Select symmetry for the plane x = 0	The type of symmetry, if any, in the YZ plane plane.
Select symmetry for the plane y = 0	The type of symmetry, if any, in the XZ plane plane.
Select symmetry for the plane z = 0	The type of symmetry, if any, in the XY plane plane.
Label increment for the new structures	After the labels are mirrored, the labels of the new elements are incremented with the value specified in this text box. Label 0 is, however, not incremented. The corresponding new elements will also have label 0. If this text box is empty or set to 0, the labels are not incremented. This means that the new elements will have the same label as that which they were created from.

All the conducting and/or dielectric triangles, segments, cuboids, tetrahedral volume elements, wedges, edges, Fock regions and polygonal plates that have been declared before the SY card, are mirrored. In addition, the second and third corners of the triangles are swapped, so that the direction of the normal vector is retained. Likewise the corners of image polygonal plates are rearranged to retain the normal direction. (The first corner point of the original polygon becomes the last corner of the mirror image.)

Sources are not mirrored. If, for example, a Hertzian dipole is placed on one side of the symmetry plane, the user must also place the correct image on the opposite side of the symmetry plane.

Multiple SY cards can be used and it is possible to mirror around more than one plane at the same time.

Related tasks
Defining Symmetry in the Model

Related information
Defining Symmetry in the Model (CADFEKO)

TG Card

With the TG card, the already entered geometric elements (triangles, segments and the rest) can be translated, rotated, mirrored and/or scaled. It is also possible to duplicate structures.

On the **Construct** tab, in the **Modify** group, click the **Translate geometry (TG)** icon.

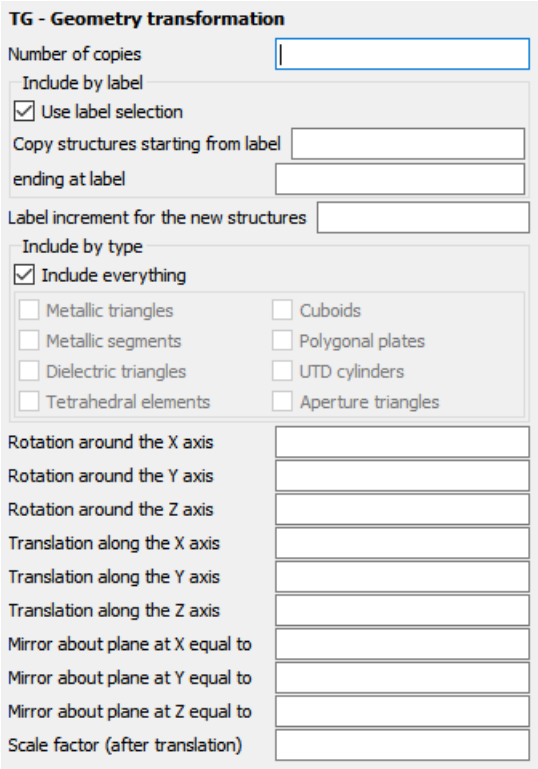


*Figure 787: The **TG - Geometry transformation** dialog.*

Parameters:

Number of copies
The number of copies to make, for example if set to 3 the selected elements will be rotated, translated, mirrored and scaled 3 times such that there will be a total of 4 structures. If set to 0, the existing elements, are rotated, translated, mirrored, scaled and the number of elements remains the same.

Use label selection
If this option is not checked, then the TG card applies to all the previously defined geometry. If this option is checked, then a label selective processing is possible.

Copy structures starting from label
Together with ending at label can be used to apply the TG card only to a selected part of the structure. The TG card is applied only to those elements whose label lies within the range set here (see also LA and CB cards and also the general discussion of label ranges). If the second field is left empty, only structures with the label set in the first field are considered.

> **Note:** That certain element types on the specified label(s) can be excluded from the selection lower in the card.

Label increment for the new structures

Each newly generated structure will be assigned a label that is incremented by this value from that of the original structure. An exception is the label 0 which is retained.

Include

This group can be used to specify which element types (provided they satisfy the label criterion) are rotated/translated.

Rotation around the X axis

Angle of rotation a_x around the X axis in degrees.

Rotation around the Y axis

Angle of rotation a_y around the Y axis in degrees.

Rotation around the Z axis

Angle of rotation a_z around the Z axis in degrees.

Translation along the X axis

Translation $\triangle_x$ in the X direction in metre (scaled by the SF card).

Translation along the Y axis

Translation $\triangle_y$ in the Y direction in metre (scaled by the SF card).

Translation along the Z axis

Translation $\triangle_z$ in the Z direction in metre (scaled by the SF card).

Mirror about the plane at X equal to

The geometry is mirrored around a plane at X equal to a constant specified. If no value is specified, no mirroring around the plane is performed.

Mirror about the plane at Y equal to

The geometry is mirrored around a plane at Y equal to a constant specified. If no value is specified, no mirroring around the plane is performed.

Mirror about the plane at Z equal to

The geometry is mirrored around a plane at Z equal to a constant specified. If no value is specified, no mirroring around the plane is performed.

Scale factor

The scaling factor γ, with which the structures must be scaled. (If left empty, it defaults to 1):

- For wire segments the wire radius is scaled as well as the coordinates of the start and end points.
- The scaling factor γ is applied after the translations/rotations have been conducted, for example, the new coordinates after the translation/rotation will be scaled. This means that the effective translation is the value specified at the TG card multiplied by the scaling factor. (If this is not desired, then two different TG cards may be used - the first applying only a scaling and the second performing the translation only).

When an SY card (symmetry) is used before the TG card, the TG card resets the symmetry if the new structures invalidates the symmetry. Cases where the symmetry is not reset is when, for example, the plane $z=0$ is a symmetry plane and the TG card specifies rotation about the Z axis for a symmetrical selection of elements. In this case the symmetry is retained.

Translation, rotation, mirroring and scaling are performed as a single transformation. The order is rotate, translate, scale and then mirror.

If more than one copy is made, successive points are generated from the previous point using the same relation.

With a TG card the simultaneous rotation around multiple axes as well as translation in multiple directions is possible. A point (x, y, z), for example the corner point of a triangle, is transformed to a new point

$$\begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix} = \gamma M \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \gamma \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \end{bmatrix} \tag{136}$$

with the rotation matrix

$$M = \begin{bmatrix} \cos a_y \cos a_z & -\cos a_y \sin a_z & \sin a_y \\ \cos a_x \sin a_z + \sin a_x \sin a_y \cos a_z & \cos a_x \cos a_z - \sin a_x \sin a_y \sin a_z & -\sin a_x \cos a_y \\ \sin a_x \sin a_z - \cos a_x \sin a_y \cos a_z & \sin a_x \cos a_z + \cos a_x \sin a_y \sin a_z & \cos a_x \cos a_y \end{bmatrix} \tag{137}$$

Multiplication by the rotation matrix M effectively rotates a point first by an angle a_z around the Z axis, then by an angle a_y around the Y axis and finally by an angle a_x around X axis. It is important to note that the second rotation around the Y axis represents the global Y axis. This is also equivalent to rotating a_x around the X axis, then rotating a_y around the new y' axis and finally rotating a_z around the new z'' axis.

The transformation angles as used by Feko in this order are generally referred to as Kardan angles as opposed to the also commonly used Euler angles. If the rotation shall be performed in the other order (for example, first around the X axis, then around the Y axis and finally around the Z axis), then one can simply use multiple consecutive TG cards. But since the same rotation algorithm is also used at other Feko cards (for instance AC or AR) where one cannot use multiple cards, a short PREFEKO code segment shall be given here which illustrates how the angles can be converted:

```
** Desired rotation angles so that we rotate first around x, then y, and
** then around z
#a1=30  ** Angle in deg. around X axis
#b1=60  ** Angle in deg. around Y axis
#c1=90  ** Angle in deg. around Z axis

** Precompute sin() and cos() terms
#ca1=cos(rad(#a1))
#cb1=cos(rad(#b1))
#cc1=cos(rad(#c1))
#sa1=sin(rad(#a1))
#sb1=sin(rad(#b1))
#sc1=sin(rad(#c1))

** Auxiliary terms resulting from equating the transformation matrices
#cc2=#cb1*#cc1/(sqrt((#cb1*#cc1)^2+(#ca1*#sc1-#sa1*#sb1*#cc1)^2))
```

```
#cb2=#cb1*#cc1/#cc2
#ca2=#ca1*#cc2/#cc1
#sa2=#cc2*(#sa1*#cc1-#ca1*#sb1*#sc1)/(#cb1*#cc1)
#sb2=#sa1*#sc1+#ca1*#sb1*#cc1
#sc2=#cc2*(#ca1*#sc1-#sa1*#sb1*#cc1)/(#cb1*#cc1)

** Finally compute the angles which must be used in Feko in the TG card
** for the rotation order first around z, then around y, and then around x
#a2=deg(atan2(#sa2,#ca2))
#b2=deg(atan2(#sb2,#cb2))
#c2=deg(atan2(#sc2,#cc2))
```

The file card based version of `example_18.pre` (see the Script Examples) gives an example of an application of the TG card.

Related tasks

Translating Geometry (CADFEKO)

Mirroring Geometry (CADFEKO)

Rotating Geometry (CADFEKO)

Related reference

AR Card

AC Card

CB Card

LA Card

SF Card

SY Card

TO Card

Using the TO card a surface mesh in the form of a toroidal segment can be generated.

On the **Construct** tab, in the **Surfaces** group, click the **Torus (TO)** icon.

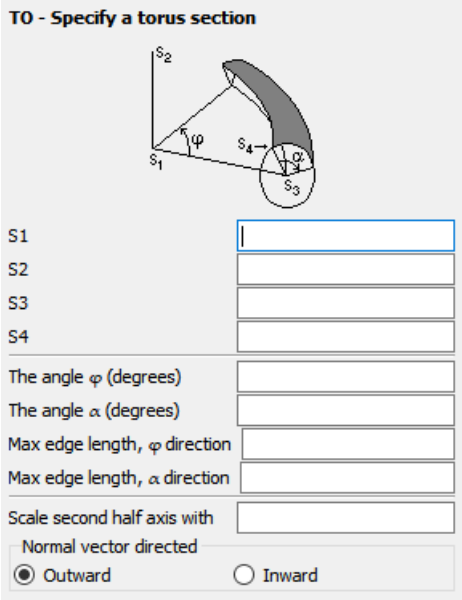


*Figure 788: The **Specify a torus section** dialog.*

Parameters:

S1 — The centre of the toroid.

S2 — A point that is perpendicular and is situated an arbitrary distance above the plane of the toroid.

S3 — The start point of the axis of the toroid.

S4 — A point on the surface of the toroidal segment. It must be in the plane S2–S1– S3.

The angle φ (degrees) — The angle of rotation around the axis S1–S2.

The angle α (degrees) — The angle of rotation around the axis of the toroid, see the figure displayed in the card.

Max edge length, φ direction — The maximum edge length along the curved edge in the φ direction in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Max edge length, α direction — The maximum edge length along the curved edge in the α direction in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.

Normal vector directed	The triangles can be created so that the normal vectors point **Outward** (outward, away the ring axis of the toroid) or **Inward**.
Scale second half axis with	If this parameter is empty or is set to 1, a toroid with a circular cross section is created. If set to $\frac{b}{a}$, an elliptical toroid is created by distorting the entire geometry along the second half axis (orthogonal to the axis S1–S3) with the factor $\frac{b}{a}$ where a is the distance S1–S3. It is not recommended to generate toroids where the elliptical cross section has extremely small or extremely large axial ratios with a CAD system (such as FEMAP) as the distortion formulation used in PREFEKO may fail in these cases.

A complete toroid is obtained by using the parameters $\varphi = 360^{\circ}$ and $\alpha = 360^{\circ}$.

Examples of TO usage

The toroidal segment, which is shown in Figure 790, is generated using a TO card. This card can also be used to generate the toroidal segment with an elliptical cross section as shown in Figure 791. Note that it is stretched in the direction of the Y axis, for example, it is elliptical in the φ-plane. It is also elliptical in the α-plane when $\varphi = 90^{\circ}$, but it is circular in the α-plane when $\varphi = 0^{\circ}$.

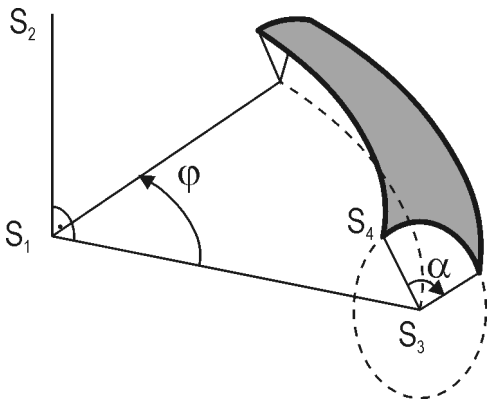


Figure 789: Sketch illustrating the use of the TO card.

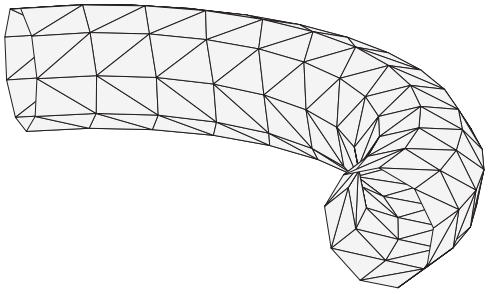


Figure 790: Example of an elliptical cross section created with the TO card.

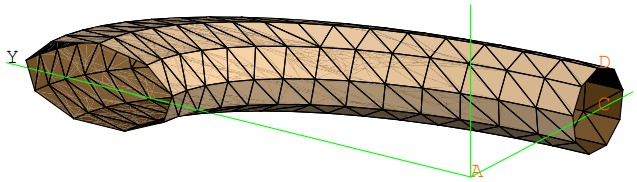


Figure 791: Example of a toroidal segment with an elliptical cross section created with the TO card.

Related reference

IP Card

SF Card

TP Card

With the TP card points (previously defined with the DP card) can be translated, rotated and/or scaled (relative to the origin).

On the **Construct** tab, in the **Modify** group, click the **Transform point (TP)** icon.

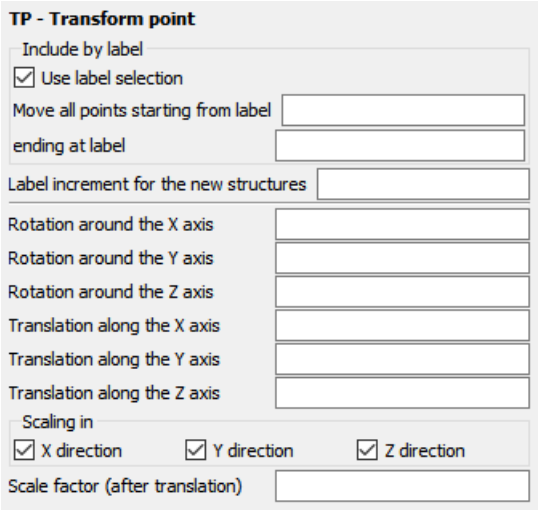


*Figure 792: The **Transform point** dialog.*

Parameters:

Use label selection
If this option is not checked, then the TP card applies to all the previously defined points. If this option is checked, then a label selective processing is possible.

Move all points starting from label
Together with **ending at label** specify the label range of points that must be translated, rotated or scaled. See the general discussion of label ranges. If the second field is left empty, only structures with the label set in the first field are considered.

ending at label
Together with **Move all points starting from label** specify the label range of points that must be translated, rotated or scaled. See the general discussion of label ranges. If the second field is left empty, only structures with the label set in the first field are considered.

Label increment for moved points
Each transformed point will be assigned a label that is the label of the original point incremented by this value. The exception are points with label 0 — their label is not incremented, it remains 0.

Rotation around the X axis
Angle of rotation a_x around the X axis in degrees.

Rotation around the Y axis
Angle of rotation a_y around the Y axis in degrees.

Rotation around the Z axis
Angle of rotation a_z around the Z axis in degrees.

Translation along the X axis	Translation $\triangle_X$ in the X direction in m (scaled by the SF card).
Translation along the Y axis	Translation $\triangle_Y$ in the Y direction in m (scaled by the SF card).
Translation along the Z axis	Translation $\triangle_Z$ in the Z direction in m (scaled by the SF card).
Scaling in	In this group the user may choose whether scaling is in the X direction, Y direction or Z direction or a combination of these.
Scale factor (after translation)	The scaling factor γ, with which the point is scaled after rotation and translation (if the parameter R_7 is not specified, it defaults to $\gamma = 1$).

If a point is rotated around more than one axis with a single card, it is rotated first by an angle a_Z around Z axis, then by a_Y around the Y axis and finally by a_X around the X axis. A more detailed description of the transformation can be found in the description of the TG card.

In an exception to the rule that all geometry cards must appear before the EG card, this card (as well as the DP card) can be used after the EG to define points for use in the AP card.

Related reference

AP Card
DP Card
EG Card
SF Card
TG Card

UT Card

This card defines the parameters for the uniform theory of diffraction (UTD) for polygonal plates and cylinders, faceted UTD for curved surfaces and ray launching geometrical optics (RL-GO).

On the **Solve/Run** tab, in the **Rays** group, click the **UTD + GO (UT)** icon.

Related concepts
UTD Ray Contributions (CADFEKO)
RL-GO Parameters (CADFEKO)
Related tasks
Solving Faces with UTD (CADFEKO)
Solving Faces with RL-GO (CADFEKO)

UTD (Polygonal and Cylindrical)

With this option the UTD parameters for UTD polygons and cylinders are specified.

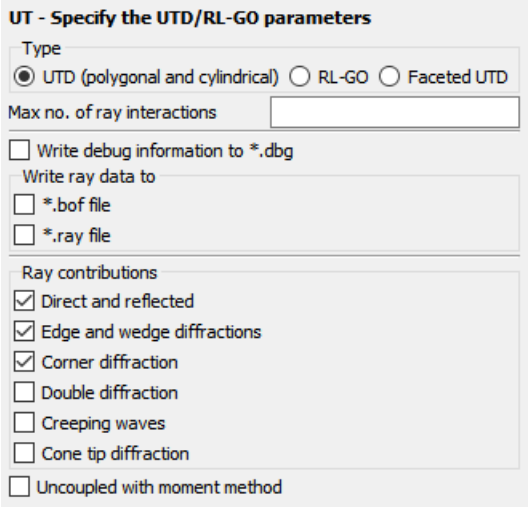


*Figure 793: The **UT - Specify the** UTD/RL-GO **parameters** dialog, set to **UTD (polygonal and cylindrical)**.*

Parameters:

Max no. of ray interactions	This parameter gives the maximal number of ray-interactions (that is reflection and diffraction combined). If for example, the parameter is set to 3, a ray can have 3 reflections, or 2 reflections and a diffraction. If set to 0, only direct rays are taken into account.
Write debug information to *.dbg	If this item is checked a debug file (extension `.dbg`) is generated. It contains large amounts of information and should only be used when debugging.

Write ray data to

**.bof file (default)*
This option exports the rays during the UTD solution process to the `.bof` file for visualisation in POSTFEKO.

**.ray file*
This option exports the rays during the UTD solution process to a `.ray` file. This text file can be used for custom post-processing.

> **Note:**
> - Large `.ray` files are possible when the MoM and UTD solution have not been decoupled and the MoM part contains a large number of mesh elements.
> - For parallel runs the run-time can also increase significantly when exporting ray data.

The following abbreviations are used in the .ray file and POSTFEKO:

- **·**: Creeping wave intermediate point on geometry surface
- **B**: Diffraction at an edge
- **D**: Diffraction at a corner or a tip
- **E**: Diffraction at a corner (of an edge)
- **K**: Diffraction at a wedge
- **Q**: Source point
- **R**: Reflection
- **S**: Observation point
- **C**: Creeping wave attaching and shedding point on geometry surface
- **V**: Reflection at the shadow boundary of a creeping wave

Ray contributions

Determines which ray contributions to take into account.

Direct and reflected	Direct and reflected rays are taken into account.
Edge and wedge diffractions	Diffraction on edges and wedges are taken into account. The ray may include an arbitrary number of reflections, but only one diffraction. The total number of interactions (the number of reflections

		plus one) for the diffraction may not be larger than specified in **Max no. of ray interactions**.
	Corner diffraction	Corner diffraction
	Double diffraction	Double diffraction on edges and wedges and combinations of reflections are taken into account. Single diffraction rays are not included in this item.
	Creeping waves	Creeping waves on curved surfaces.
	Cone tip diffraction	Tip diffraction at the tip of a cone.
Uncoupled with moment method	This item specifies whether the coupling from the UTD region to the MoM region should be considered. This option should only be used when the UTD and MoM regions do not couple (interact) strongly.	

Increasing the type and number of ray interactions increases accuracy and the computation time. The user should therefore make a compromise between the number of ray interactions and the ray contributions. Choices made in this card should be made on physical considerations to get optimal use from the UTD formulation.

The following restrictions apply for the hybrid MoM/UTD:

- No dielectric bodies or dielectric ground.
- Only flat polygonal plates or a single cylinder allowed in the UTD region.
- The structure types can be PEC or of lossy metals, and the PEC structures can have coatings and thin dielectric sheets.
- UTD and PO are not allowed to be used simultaneously in a model.

RL-GO

With this option the RL-GO parameters are specified.

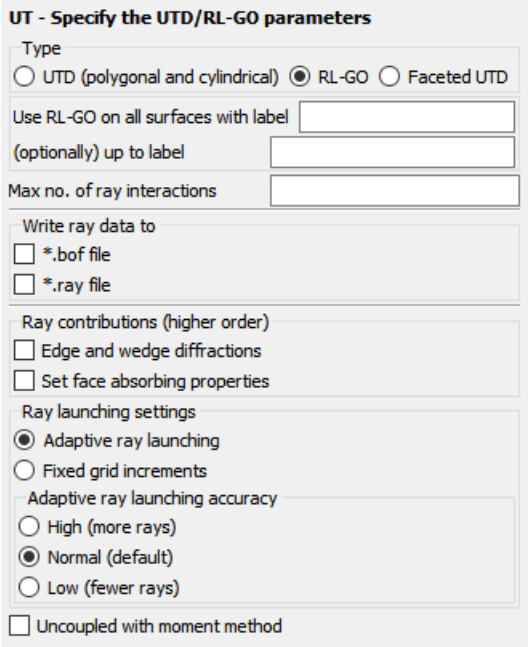


*Figure 794: The **UT - Specify the UTD/RL-GO parameters** dialog, set to **RL-GO**.*

Parameters:

Use RL-GO on all surfaces with label
The label to which the RL-GO should be applied

(optionally) up to label
The label up to which the RL-GO should be applied. RL-GO will not be applied to labels outside of the range between this field and the label in the previous field.

Max no. of ray interactions
This parameter gives the maximum number of ray-interactions (that is reflection and transmission combined). If for example, the parameter is set to 3, a ray can have 3 reflections, or 2 reflections and a transmission. If left empty, then the maximum number of ray interactions is determined automatically.

Write ray data to

**.bof file (default)*
This option exports the rays during the RL-GO solution process to the `.bof` file for visualisation in POSTFEKO.

**.ray file*
This option exports the rays during the RL-GO solution process to a `.ray` file. This text file can be used for custom post-processing.

Note:

- Large `.ray` files are possible when the MoM and UTD solution have not been decoupled and the MoM part contains a large number of mesh elements.
- For parallel runs the run-time can also increase significantly when exporting ray data.

The following abbreviations are used in the .ray file and POSTFEKO.

Q	Source point
R	Reflection
S	Observation point
T	Transmission on a dielectric

Ray contributions (higher order)

Determines which ray contributions to take into account.

Edge and wedge diffractions	Diffraction on edges and wedges are taken into account.
Set face absorbing properties	This option enables the specification of the absorbing and reflection/transmission properties of the faces with regards to rays. For both sides of a face (normal side / opposite to normal side), the following options are supported: • Consider transmission / reflection of rays from the face(s). • No transmission / reflection of rays from the face(s).

Ray launching settings

Specify the settings for the launching of rays.

Adaptive ray launching	The ray launching parameters are determined automatically during the solution process.
Fixed grid increment	The **Angular increment** (theta / phi) and the **Spatial increment** (distance between the individual rays in the parallel

	ray front) for the ray launching are specified by the user.
Adaptive ray launching accuracy	Controls the density of the rays launched as well as when to stop tracing a ray based on the ray's decay. Choose between **High (more rays)**, **Normal (default)** and **Low (fewer rays)**.

> **Tip:** Start with **Low (fewer rays)**, which uses the least computational resources, and when the model appears to be performing roughly as expected, use a higher setting.

Uncoupled with moment method	This item specifies whether the coupling from the UTD region to the MoM region should be considered. This option should only be used when the UTD and MoM regions do not couple (interact) strongly.

The main advantages of RL-GO are as follows:

- It is suitable for large arbitrarily shaped objects (also curved objects).
- Dielectrics are supported.
- A plane wave source required for the calculation of far field RCS is supported (the UTD has a problem with caustics).
- Multiple reflections can be solved more efficiently than UTD or PO.

When no UT card is used, the following default values apply:

- **Max no. of ray interactions**: 3
- **Write debug information to *.dbg**: unchecked
- **Export ray data for post-processing**: unchecked
- **Select ray contributions** includes:
  - **Direct and reflected**
  - **Edge and wedge diffracted rays**
  - **Corner diffraction**

Faceted UTD

With this option the UTD parameters for faceted UTD for planar and curved faces are specified.

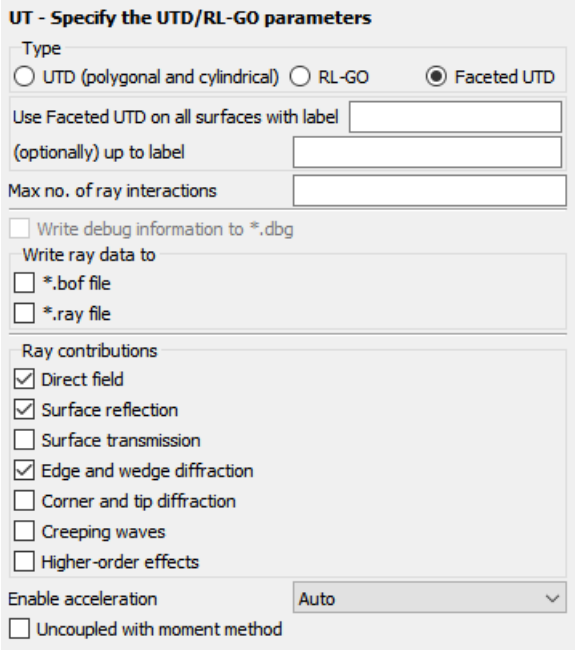


*Figure 795: The **UT - Specify the** UTD/RL-GO **parameters** dialog, set to **Faceted UTD**.*

Parameters:

Use faceted UTD on all surfaces with label
The label to which the faceted UTD should be applied

(optionally) up to label
The label up to which the faceted UTD should be applied. Faceted UTD will not be applied to labels outside of the range between this field and the label in the previous field.

Max no. of ray interactions
This parameter gives the maximal number of ray-interactions. If for example, the parameter is set to 3, a ray can have 3 reflections. If set to 0, only direct rays are taken into account.

Write ray data to

**.bof file (default)*
: This option exports the rays during the UTD solution process to the `.bof` file for visualisation in POSTFEKO.

**.ray file*
: This option exports the rays during the UTD solution process to a `.ray` file. This text file can be used for custom post-processing.

Note:

- Large `.ray` files are possible when the MoM and UTD solution have not been decoupled and the MoM part contains a large number of mesh elements.
- For parallel runs the run-time can also increase significantly when exporting ray data.

The following abbreviations are used in the .ray file and POSTFEKO:

- **·**: Creeping wave intermediate point on geometry surface
- **B**: Diffraction at an edge
- **D**: Diffraction at a corner or a tip
- **E**: Diffraction at a corner (of an edge)
- **K**: Diffraction at a wedge
- **Q**: Source point
- **R**: Reflection
- **S**: Observation point
- **C**: Creeping wave attaching and shedding point on geometry surface
- **V**: Reflection at the shadow boundary of a creeping wave

Ray contributions Determines which ray contributions to take into account.

Direct field	Direct rays are taken into account.
Surface reflection	Rays reflected by PEC and non-metallic planar and curved surfaces are taken into account.
Surface transmission	Rays transmitted (refracted) by PEC and non-metallic planar and curved surfaces are taken into account.
Edge and wedge diffraction	Diffraction on edges and wedges are taken into account.
Corner and tip diffraction	Diffraction at corners and tips are taken into account.
Creeping waves	Creeping waves on curved surfaces.

	Combination of different effects	Only multiple reflections plus one edge/wedge diffraction at any position along the ray path can be computed. This option is only active if **Surface reflection** and **Edge and Wedge diffraction** check boxes are selected and the **Max. no. of ray interactions** is larger than 1.

Uncoupled with moment method

This item specifies whether the coupling from the faceted UTD region to the MoM region should be considered. This option should only be used when the faceted UTD and MoM regions do not couple (interact) strongly.

Enable acceleration

An acceleration technique can be used to speed-up the search process for ray paths significantly but could result in some rays not being found in exceptional cases.

Auto	The Solver determines automatically if the acceleration technique should be used for the model (if the method is likely to speed up the solution).
On	This option enables the acceleration technique. Runtime decreases but technique could result in some rays not being found in exceptional cases.
Off	This option disables the acceleration technique at the expense of a runtime increase.

Increasing the type and number of ray interactions increases accuracy and the computation time. The user should therefore make a compromise between the number of ray interactions and the ray contributions. Choices made in this card should be made on physical considerations to get optimal use from the UTD formulation.

The following restrictions apply for the faceted UTD:

- Only planar triangles are allowed.

UZ Card

The UZ card is used to create a cylinder that will be solved with the uniform theory of diffraction (UTD).

On the **Construct** tab, in the **Surfaces** group, click the **Cylinder** icon. From the drop-down list, click the **Unmeshed cylinder (UZ)** icon.

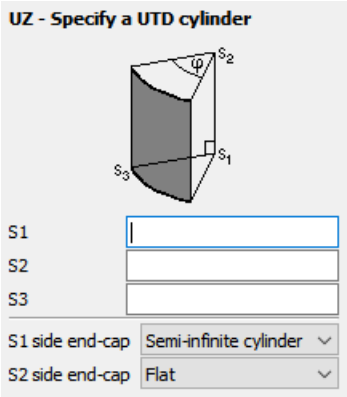


*Figure 796: The **UZ - Specify a UTD cylinder** dialog.*

Parameters:

S1	The start point of the cylinder axis (previously defined with a DP card).
S2	The end point of the cylinder axis (previously defined with a DP card).
S3	A point on the radius of the cylinder. The angle S2–S1–S3 must be 90°.
S1 side end-cap	Select a flat end cap or a semi-infinite end on the side of S1.
S2 side end-cap	Select a flat end cap or a semi-infinite end on the side of S2.

Example of UZ Card Usage

The UZ card is used to create a UTD cylinder. Note the absence of discretisation.

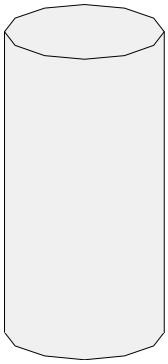


Figure 797: Example of cylinder created with the UZ card.

Related tasks
Creating a UTD Cylinder (CADFEKO)
Related reference
DP Card

VS Card

This card specifies known visibility information (required when using physical optics with multiple reflections) to reduce the calculation time.

On the **Solve/Run** tab, in the **Rays** group, click the **Physical optics** icon. From the drop-down list, select the **PO visibility (VS)** icon.

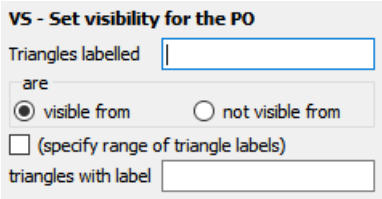


*Figure 798: The **VS - Set visibility for the PO** dialog.*

To accurately compute multiple reflections Feko needs to determine which basis functions are visible to each other. Since this applies to all the PO triangles it may be very time consuming for large problems. The time required to determine the visibility may be greatly reduced if the user can specify which triangles are visible or hidden from each other.

Parameters:

Triangles labelled	The label of the source triangles.
are visible from	All triangles with the label specified in the field **Triangles labelled** are visible from all triangles with label(s) indicated in the fields below.
are not visible from	All triangles with the label specified in the field **Triangles labelled** are not visible from all triangles with label(s) indicated in the fields below.
(specify range of labels)	If this item is unchecked, only a single label is specified (in **triangles with label**). If checked, the card applies to all triangles with labels in the range from the value specified in **triangles with label** to those in **to triangles with label**.

Note that visibility is reciprocal, meaning if all triangles with label n are visible from all triangles with label m, all triangles with label m are visible from all triangles with label n as well.

Basis functions cannot illuminate each other if all the triangles they are attached to lie in the same plane.

The VS card should only be used if the user can specify the visibility beyond any doubt and if it applies to all triangles of that label. If no information is specified for a specific combination of labels/triangles, full ray tracing will be executed.

The figure below depicts a structure consisting of four flat plates and a cylindrical section. The two plates orientated at 45 degrees to the coordinate system (labelled 1 and 3), are half as wide as the

plates with labels 0 and 2. Thus a subset of the triangles with label 2 are visible to triangles with label 0, but not all.

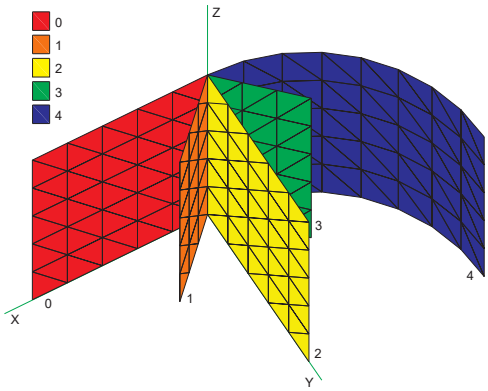


*Figure 799: The **VS - Set visibility for the PO** dialog.*

Visibility information should be specified starting at label 0, then label 1, and so forth. VS cards should be specified as follows:

- Triangles labelled 0 are not visible from triangles with label 0.
- Triangles labelled 0 are visible from triangles with label 1.
- Triangles labelled 0 are not visible from triangles with labels 3 to 4.
- Triangles labelled 1 are not visible from triangles with label 1.
- Triangles labelled 1 are visible from triangles with label 2.
- Triangles labelled 1 are not visible from triangles with labels 3 to 4.
- Triangles labelled 2 are not visible from triangles with label 2.
- Triangles labelled 2 are visible from triangles with label 3.
- Triangles labelled 3 are not visible from triangles with label 3.
- Triangles labelled 3 are visible from triangles with label 4.

The VS cards are realised with code section below.

```
VS: 0 : 3 :   : 0
VS: 0 : 1 :   : 1
VS: 0 : 4 :   : 3 : 4
VS: 1 : 3 :   : 1
VS: 1 : 1 :   : 2
VS: 1 : 4 :   : 3 : 4
VS: 2 : 3 :   : 2
VS: 2 : 1 :   : 3
VS: 3 : 3 :   : 3
VS: 3 : 1 :   : 4
```

Since all the triangles with label 0 lie in the same plane, they cannot illuminate each other. Thus the first card states that label 0 is hidden from label 0.

All triangles with label 1 are visible from all triangles with label 0. This is specified by the second VS card. Since a subset of triangles with label 2 are visible from triangles with label 0 while others are hidden, any information for this combination of layers cannot be specified. However, the plate with label

2 shadows all triangles with labels 3 and 4 and it can be specified that these are hidden. This is done with the third VS card. Note that this card specifies a range of hidden labels.

Next specify which triangles are visible (or hidden) from all triangles with label 1. As for label 0, triangles with label 1 are not visible to each other, specified by the fourth VS card. All triangles with labels 0 and 2 are visible from all triangles with label 1. Since the visibility between labels 0 and 1 has already been specified, it does not have to be specified again. The fifth VS card then specifies that label 2 is completely visible from label 1. As for label 0, both labels 3 and 4 are hidden completely which completes the first six VS cards.

Next consider label 2. As before labels lower than 2 need not be considered. Also the label is hidden from itself as indicated by VS card number seven. Next it can be stated that label 3 is visible, but nothing can be specified about label 4 as not all of these triangles will be visible.

Similarly VS cards 9 and 10 state that label 3 is not visible to itself and fully visible to label 4. Finally consider the case for triangles with label 4. All visibility with layers 0 to 3 has been specified and may not be specified again. Unlike the previous flat plates, layer 4 is curved and triangles may indeed illuminate other triangles with the same layer. However, not all other triangles will be illuminated (this is only possible for a doubly concave surface), so no information can be specified for label 4.

Related reference
PO Card

WA Card

The WA card is used to define all windscreen antenna solution elements. This would include all elements in close proximity to the finite glass structure and can consist of either segments or triangles (all defined by labels).

On the **Solve/Run** tab, in the **Windscreen** group, click the **Windscreen element (WA)** icon.

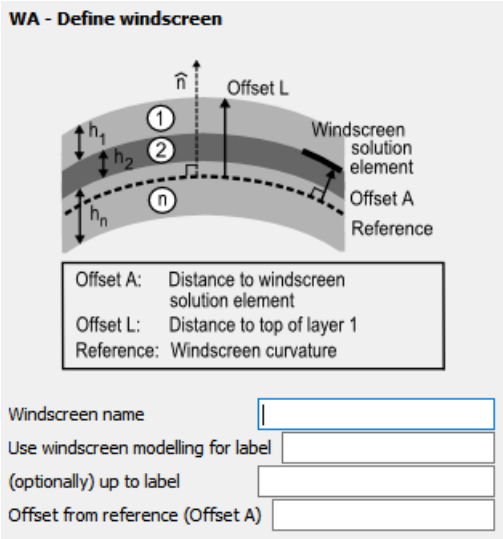


*Figure 800: The **WA - Define windscreen** dialog.*

> **Note:** The WR, WA and WD cards should be used together to create windscreen antenna models. These three cards respectively define the windscreen reference surface, the windscreen solution elements (antenna) and the windscreen layered media definition.

Parameters:

Windscreen name	Name of the windscreen.
Use windscreen modelling for label	Start of the label range.
(optionally) up to label	[Optional] End of the label range.
Offset from reference (Offset A)	Offset of the specified label geometry with respect to the reference windscreen triangles.

The antenna elements will be limited to lying tangentially with respect to the windscreen surface. Using a defined offset from the reference plane (in the direction of the reference plane normal), these elements can then be positioned at the exact required location. This offset is specifically needed because of the limitations of a finite mesh (compared to a smooth surface) in combination with curvature in the model.

Related tasks
Applying a Windscreen Layer to a Face (CADFEKO)

WG Card

The WG card is used to create a wire grid in the shape of a parallelogram.

On the **Construct** tab, in the **Wires** group, click the **Wire grid (WG)** icon.

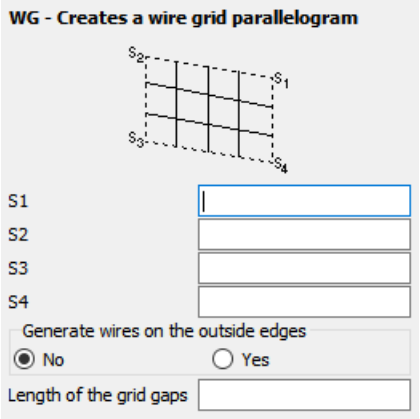


*Figure 801: The **WG - Creates a wire grid parallelogram** dialog.*

Parameters:

S1, S2, S3, S4	The four corners of the parallelogram in consecutive order.
Generate wires on the outside edges	Select **No** to create the wires inside the parallelogram only or **Yes** to generate all wires. This option is important when creating two adjacent parallelograms to ensure that the segments along the sides are not generated twice.
Length of the grid gaps	The maximum segment length is given by the IP card. This parameter is an integer number and specifies the density of the grid. If, for example, this is set to 2, the wires only cross at every second segment.

Examples of WG Card Usage

The following wire grids are created using the WG card. The grid spacing is specified in terms of the number of segment lengths as defined with an IP card.

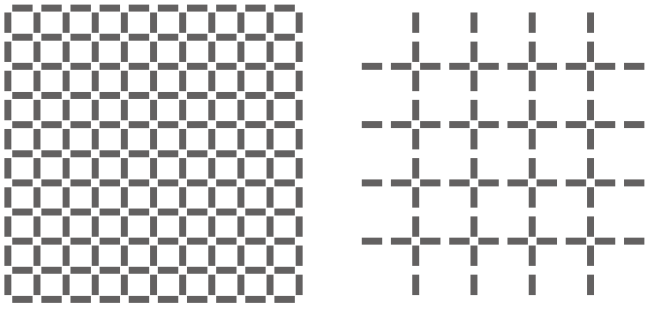


*Figure 802: Examples of wire grids created with the WG card . The example on the right has a value of 2 in **Length of the grid gaps**.*

Related reference
IP Card

WR Card

The WR card is used to define a dielectric windscreen reference plane. Geometrically this surface is not part of the electromagnetic model and is used simply to determine the curvature factor between the two elements on the windscreen.

On the **Solve/Run** tab, in the **Windscreen** group, click the **Windscreen reference (WR)** icon.

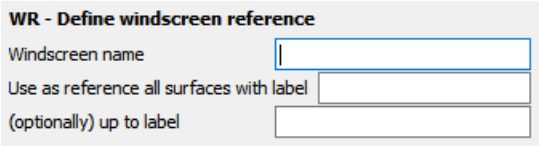


*Figure 803: The **WR - Define windscreen reference** dialog.*

> **Note:** The WR, WA and WD cards should be used together to create windscreen antenna models. These three cards respectively define the windscreen reference surface, the windscreen solution elements (antenna) and the windscreen layered media definition.

Parameters:

Windscreen name	The name of the windscreen.
Use as reference all surfaces with label	The start of the label range.
(optionally) up to label	[Optional] The end of the label range.

> **Note:** The windscreen reference triangles are defined by label and since this plane also forms the zero reference with respect to the defined windscreen antenna elements and glass layers, they should all have their normals pointing in the same direction.

Related tasks
Applying a Windscreen Layer to a Face (CADFEKO)
Related reference
WA Card
WD Card

ZY Card

This card defines a surface mesh in the form of a cylindrical segment.

On the **Construct** tab, in the **Surfaces** group, click the **Cylinder** icon. From the drop-down list, click the **Cylinder (ZY)** icon.

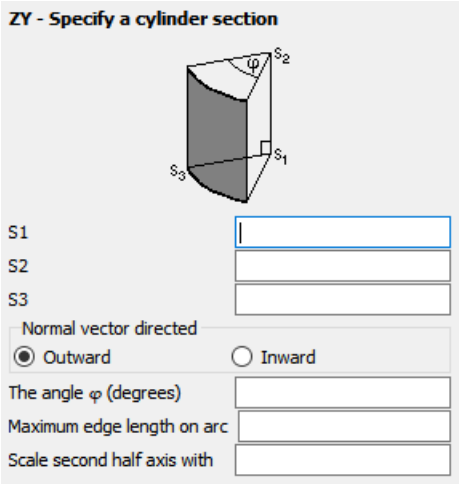


*Figure 804: The **ZY - Specify a cylinder section** dialog.*

Parameters:

S1	The start point of the axis.
S2	The end point of the axis.
S3	A point on the corner of the cylindrical segment.
Normal vector directed	The triangles can be created so that the normal vector is points **Outward** or **Inward**.
The angle φ (degrees)	The angle in degrees, which is subtended by the cylindrical arc.
Maximum edge length on arc	Maximum edge length of the triangles along the curved side in m (is scaled by the SF card). If this parameter is left empty, the value specified with the IP card is used.
Scale second half axis with	If this parameter is empty or is set to 1 , a circular cylinder is created. If set to $\frac{b}{a}$, an elliptical cylinder is created. Here $\frac{b}{a}$ gives the ratio of the two half axes, where a is the distance S_1–S_3. It is not recommended to generate elliptical cylinders with extremely small or extremely large axial ratios with a CAD system as the distortion formulation used in PREFEKO may fail in these cases.

For an orthogonal cylinder (that is the lines S_1–S_2 and S_1–S_3 are perpendicular), the segmented area (shaded in the figure of the cylinder) is obtained by rotating the point S_3 around the axis S_1–S_2 through φ. For $\varphi = 360°$ a full cylinder is created.

An oblique cylinder (that is the circular or elliptical rim is not perpendicular to the axis) can also be created. Then S_1–S_2 still represents the axis, but the top and bottom planes of the rims are defined by planes perpendicular to the plane defined by the three points S_1, S_2, S_3, and parallel to the line S_1–S_3.

Examples of ZY card usage:
The cylindrical section below was created with a single ZY card.

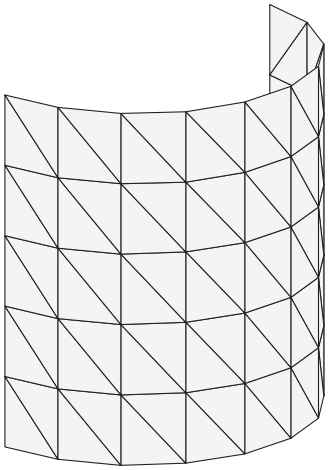


Figure 805: Example of a cylindrical section using the ZY card.

```
**
IP:  :  :  :  :  : 0.4
DP: A :  :  :  : 0.0 : 0.0 : 0.0
DP: B :  :  :  : 0.0 : 0.0 : 2.0
DP: C :  :  :  : 1.0 : 0.0 : 0.0
ZY: A : B : C :  :  : 180.0 : 0.35
EG: 1 : 0 : 0 :  :  :  :  :  :  :  :  : 1
EN
```

The elliptical cylinder below was created with a single ZY card.

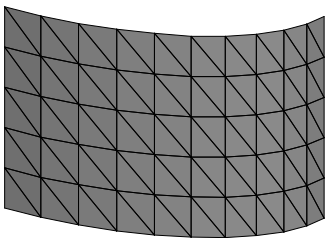


Figure 806: Example of an elliptical cylinder using the ZY card.

```
** Example for an elliptical cylinder

** General parameters
#a=1.5  ** first half axis of the elliptical cylinder
#b=2.5  ** second half axis of the elliptical cylinder
#h=4  ** height of the cylinder

** Segmentation
#kanl=0.4
IP:  :  :  :  :  : #kanl
```

```
** Define points
DP: A :  :  :  : 0 : 0 : 0
DP: B :  :  :  : 0 : 0 : #h/2
DP: C :  :  :  : #a : 0 : 0

** Define the geometry
ZY: A : B : C :  :  : 90 : #kanl : #b/#a

** End
EG: 1 : 0 : 0 :  :  :  :  :  :  :  :  : 1
EN
```

The non-orthogonal cylinder below was created with a single ZY card.

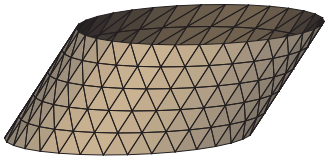


Figure 807: Example of an non-orthogonal cylinder created with the ZY card.

```
** Example for a non-orthogonal elliptical cylinder
IP:  :  :  :  :  : 0.3
DP: A :  :  :  :  : 0.0 : 0.0 : 0.0
DP: B :  :  :  :  : 0.7 : 0.0 : 1.0
DP: C :  :  :  :  : 1.0 : 0.0 : 0.0
ZY: A : B : C :  :  : 360.0 :  : 1.5
EG: 1 : 0 : 0 :  :  :  :  :  :  :  :  : 1
EN
```

Related reference
Cylinder (CADFEKO)

A-1.2 Control Cards

Control cards are used to specify requests and solver settings and are placed after the EG card in the `.pre` file.

Table 66: Control Cards.

Card	Description
A0	Defines a linear polarised plane wave incident on the structure.
A1	Defines an excitation by means of a voltage gap on a segment (impressed electric field strength along a segment).
A2	Defines an excitation by means of a voltage gap at a node (between two segments).
A3	Defines an excitation by means of a magnetic ring current (TEM-frill) on a segment to model a coaxial feed.
A5	Defines an excitation by means of an electric Hertzian dipole. The position and orientation in space are arbitrary.
A6	Defines an excitation by means of an magnetic Hertzian dipole. The position and orientation in space are arbitrary.
A7	Defines an excitation by means of a voltage gap on an edge between two triangles. This card has been generally replaced by the AE card.
AC	This card reads the geometry and current distribution (possibly for more than one frequency) from an `.rsd` file created by a transmission line simulation program (for example, CRIPTE or CableMod) or by a PCB simulation tool (PCBMod) or by export with the OS card. The excitation is due to the electromagnetic fields radiated by these line currents.
AE	Defines an excitation between triangle edges similar to the A7 card, however the AE card permits the simultaneous excitation of several edges.
AF	Define an excitation by an impressed line current in the FEM region.
AI	Define an excitation by an impressed line current.
AJ	Define an excitation by means of an impressed current source defined using current data calculated for a PCB.
AK	Define an excitation by means of a voltage source connected to a radiating cable.
AM	Define an excitation by means of an impressed current source defined using model solution coefficients.

Card	Description
AN	Define an excitation by means of a voltage source connected to a non-radiating network port.
AP	Define an excitation by means of equivalent sources in an aperture (array of electrical and magnetic Hertzian dipoles).
AR	Define an excitation by an antenna with a given radiation pattern.
AS	Define an excitation by means of impressed radiating spherical modes.
AT	Define an excitation by means of a voltage source applied to a voxel mesh.
AV	Define an excitation by an impressed line current similar to the AI card, but the endpoint of the current is electrically connected to a conducting surface.
AW	Excitation by an impressed mode on a waveguide port.
BO	Insert a reflective ground in the model.
CA	Defines a cable path section for the cable irradiation computation.
CD	Defines a specific cable cross section (for example, single, coaxial, ribbon and bundle).
CF	Sets the type of integral equation for perfectly conducting metallic surfaces.
CG	Select the algorithm used to solve the matrix equation.
CH	Groups cable harness specific properties.
CI	Defines a cable interconnect and termination.
CM	Field calculation for CableMod and CRIPTE (coupling into transmission lines) or PCBMod (coupling into a PCB).
CO	Inserts a dielectric and/or magnetic surface on the elements.
CR	Specifies the orientation for a 3D anisotropic medium.
CS	Defines a cable path section and the centre/reference location to which a cable cross section is applied.
DA	Exports data to additional ASCII files.
DI	Defines a dielectric medium.
DL	Defines a layered dielectric medium.
EE	Adds a request to calculate error estimates.

Card	Description
EN	Indicates the end of the input file.
FD	Defines the FDTD solver settings.
FE	Adds a request to calculate the near fields.
FF	Add a request to calculate the far fields.
FR	Defines the frequencies at which the calculations are to be carried out.
GF	Define a homogeneous medium, a layered dielectric sphere or a planar multilayer substrate.
KC	Transfer the signal names in CADFEKO to POSTFEKO
KS	Transfer the connector names in CADFEKO to POSTFEKO.
L2	Defines a complex load on a vertex.
LC	Defines a cable load.
LD	Defines a distributed load, consisting of resistance, inductance and capacitance.
LE	Defines a load on the edge between surface triangles.
LF	Impress a complex impedance between two points inside a FEM mesh.
LN	Defines a complex load to any non-radiating network port that is not connected to geometry.
LP	Defines a parallel circuit (resistance, inductance and capacitance load.
LS	Defines a series circuit (resistance, inductance and capacitance) load.
LT	Defines a series circuit (resistance, inductance or capacitance) load to a voxel mesh to be used in conjunction with the FDTD method.
LZ	Defines a complex load on a segment.
MD	Specify the options for model decomposition and write the solution coefficients to a `.sol` file.
NW	Defines a linear non-radiating network.
OF	Specify the offset / displacement of the origin when calculating near fields or far fields.
OM	Calculates the weighted set of orthogonal current-modes that are sup ported on a conducting surface.
OS	Saves the surface currents in a file.
PP	Defines the phase for periodic boundary condition calculation.

Card	Description
PR	Defines a current / voltage probe.
PS	Sets general control parameters.
PW	Defines the radiating power of a transmitting antenna.
RA	Defines an ideal receiving antenna.
SA	Defines a request to calculate SAR in dielectric media.
SB	Defines a magnetostatic bias field to be applied to a 3D anisotropic medium.
SC	Defines a SPICE circuit that can be used as a load when defining a load.
SD	Define shield layer definitions.
SH	Define solid or braided cable shields.
SK	Takes finite conductivity into account through the skin effect of ohmic losses; also for thin dielectric layers.
SP	Calculates the S-parameters for the active sources.
TL	Specifies a non-radiating transmission line.
TR	Calculates reflection and transmission coefficients for an incident plane wave on a planar structure.
WD	Defines the dielectric properties of the windscreen glass layers.

AX Cards

These cards define the type of excitation (source) as well as other relevant parameters.

Table 67: Types of excitation and other relevant parameters.

Card	Type of Excitation
A0	A linear polarised plane wave incident on the structure.
A1	Excitation by means of a voltage gap on a segment (that is an impressed electric field strength along a segment).
A2	Excitation by means of a voltage gap at a node (that is between two segments).
A3	Excitation by means of a magnetic ring current (TEM-frill) on a segment. Thus a coaxial feed can be modelled.
A5	An electric Hertzian dipole is used as excitation. The position and orientation can be arbitrary.
A6	A magnetic Hertzian dipole is used as excitation. The position and orientation can be arbitrary.
A7	Excitation by means of a voltage gap on an edge between two triangles. It is recommend to rather use the AE card.
AC	This card reads the geometry and current distribution (possibly for more than one frequency) from an `.rsd` file created by a transmission line simulation program (CRIPTE or CableMod) or by a PCB simulation tool (PCBMod) or by export with the OS card in Feko. The excitation is due to the electromagnetic fields radiated by these line currents.
AE	The AE card is an excitation between triangle edges similar to the A7 card, however the AE card permits the simultaneous excitation of several edges.
AF	Excitation by an impressed line current in the FEM region.
AI	Excitation by an impressed line current.
AJ	Excitation by means of an impressed current source defined using current data calculated for a PCB.
AK	Excitation by means of a voltage source connected to a radiating cable.
AM	Excitation by means of an impressed current source defined using model solution coefficients.
AN	Excitation by means of a voltage source connected to a non-radiating network port.
AP	Excitation by means of equivalent sources in an aperture (array of electrical and magnetic Hertzian dipoles).
AR	Excitation by an antenna with a given radiation pattern.

Card	Type of Excitation
AS	Excitation by means of impressed radiating spherical modes.
AT	Excitation by means of a voltage source applied to a voxel mesh.
AV	Excitation by an impressed line current similar to the AI card, but the end point of the current is electrically connected to a conducting surface.
AW	Excitation by an impressed mode on a waveguide port.

The different ways to realise a voltage source are summarised in Figure 808 and Figure 808. The impressed electric field strength is indicated by $\mathbf{E}_i$.

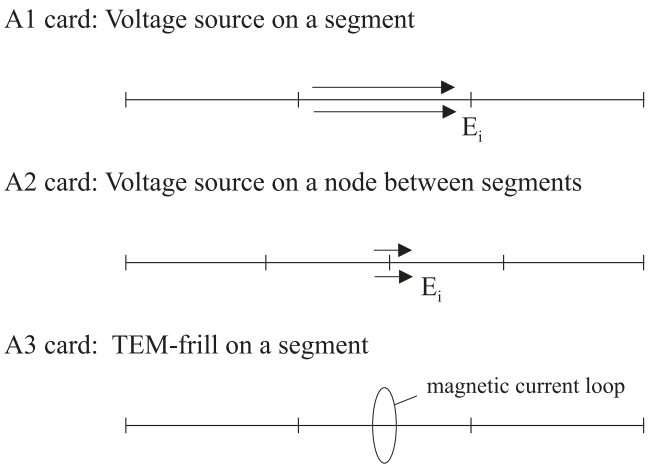
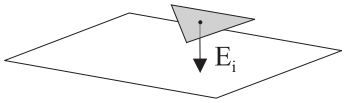
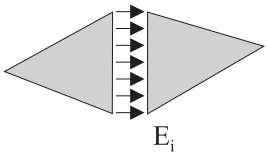


Figure 808: Possible ways to realise a voltage source on a wire segment.

A4 card: Vertical pin approximation (dielectric substrate)



A7 card: Voltage gap on an edge



AE card: Voltage gap along edges

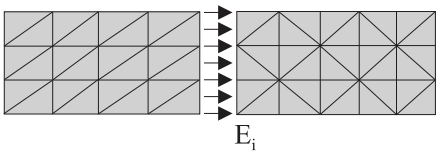


Figure 809: Possible ways to realise a voltage source in connection with triangles.

More than one excitation is also allowed. It is possible, for example, to generate an elliptically polarised plane wave by super-imposing two out-of-phase linearly polarised plane waves with different amplitudes. It is also possible to feed an antenna with two different voltage sources. For this purpose the parameters **New source** and **Add to sources** are available in each Ax card. This parameter indicates whether the current excitation is additional (**Add to sources**) or not (**New source**). When **New source** is selected, only the current excitation will be used and the excitations prior to the current one will be erased.

For the excitations A1, A2, A3 and A7 it is possible to select the feed element through the label. Alternatively the position of the feed is specified in Cartesian coordinates. Feko then searches for a segment or an element at this position. This comparison of the position uses the tolerance parameter **Maximum identical distance** (see the EG Card).

Related reference

EG Card

OS Card

A0 Card

The A0 card defines a linearly polarised incident plane wave.

On the **Source/Load** tab, in the **Ideal sources** group, click the **Plane wave** icon.

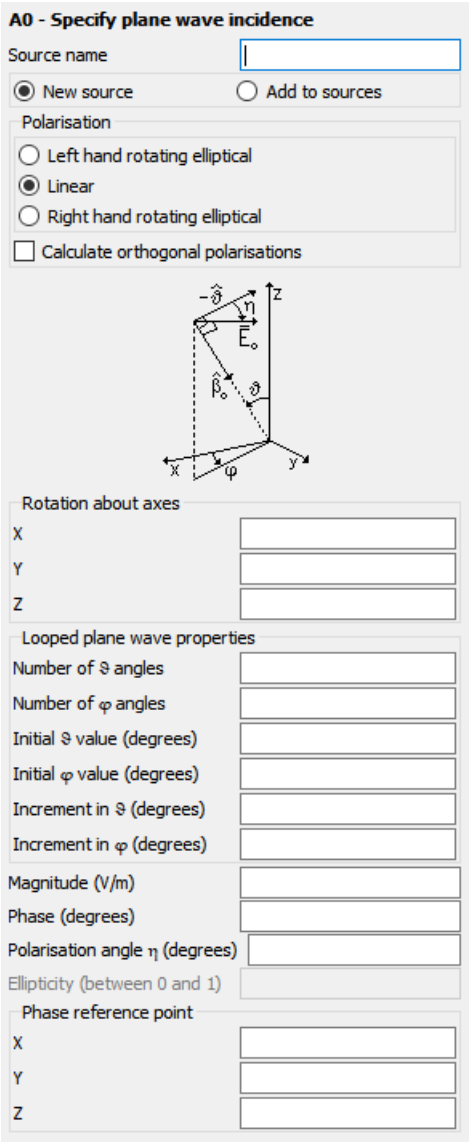


*Figure 810: The **A0 - Specify plane wave incidence** dialog.*

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.

Polarisation	This group sets the behaviour of the polarisation vector. If either the **Left hand rotating elliptical** or **Right hand rotating elliptical** is selected, the **Ellipticity** must be specified. For **Linear** polarisation, an option is available to **Calculate orthogonal polarisations**.
Rotation about axis	The rotation of the plane wave about the X axis, Y axis and Z axis.
Number of ϑ angles	If more than one direction of incidence is to be examined, then this parameter indicates the number of incident angles in the ϑ direction. If this field is left empty or set to 0, a value of 1 will be used.
Number of φ angles	If more than one direction of incidence is to be examined, then this parameter indicates the number of incident angles in the φ direction. If this field is left empty or set to 0, a value of 1 will be used.
Magnitude	Magnitude of the field strength $\mathbf{E}_0$ of the incident field in $\frac{\mathrm{V}}{\mathrm{m}}$.
Phase	Phase of the field strength $\mathbf{E}_0$ of the incident field in degrees.
Initial ϑ value:	Angle of incidence ϑ of the plane wave in degrees.
Initial φ value	Angle of incidence φ of the plane wave in degrees.
Polarisation angle η	It is the angle, in a right-handed sense when viewing in the incident direction from $-\hat{\vartheta}$ to $\mathbf{E}_0$. See the card image in Figure 810.
Increment in ϑ	If more than one direction of incidence is to be examined, ϑ is incremented by this value for each new angle of incidence.
Increment in φ	If more than one direction of incidence is to be examined, φ is incremented by this value for each new angle of incidence.
Ellipticity	This field is only applicable when either the **Left hand rotating elliptical** or **Right hand rotating elliptical** is selected under **Polarisation** and gives the ellipticity of the rotating polarisation. It must be larger than 0 (linear polarisation) and smaller or equal to 1 (circular polarisation).
Phase reference point	The phase reference point for plane waves is set to the global origin by default. By modifying the phase reference, the simulation will zero the incident plane wave phase to the specified position.

The direction of incidence $\hat{\beta}_0$ is specified by the incidence angles ϑ and φ. The polarisation angle η (measured from the negative of the spherical coordinate system unit vector $\hat{\vartheta}$) and the field strength vector $\mathbf{E}_0$ are defined as indicated in Figure 810.

The electric field strength of the incident field is then given by

$$\mathbf{E}_i(\mathbf{r}) = \left[\mathbf{E}_0 + jv\left(\mathbf{E}_0 \times \hat{\beta}_0\right)\right] \cdot e^{-j\boldsymbol{\beta}_0 \mathbf{r}} \tag{138}$$

where v is the ellipticity.

For linear polarisation, ellipticity is 0; for right hand rotating, ellipticity is equal to the value in the **Ellipticity** field; for left hand rotating, ellipticity is the negative of the value in the **Ellipticity** field. The incident magnetic field is given by

$$\mathbf{H}_i(\mathbf{r}) = \frac{1}{Z_F}\hat{\beta}_0 \times \mathbf{E}_i \tag{139}$$

with Z_F the wave impedance in the surrounding free space medium.

Note that the incident power density (which is required, for example, for RCS computations) is given by

$$\mathbf{S}_i = \frac{1}{2}\frac{|\mathbf{E}_0|^2}{Z_F}(1+v^2). \tag{140}$$

It is possible to vary the direction of incidence. This is particularly useful for example, when determining the monostatic radar cross section. The two parameters **Initial ϑ value** and **Initial φ value** indicate the direction of the first wave. The direction of incidence is varied in the ϑ direction by the increment **Increment in ϑ** and in the φ direction by **Increment in φ**. In each direction these two angles are examined and a total number of incident waves equal to the product of these two angles are examined.

If an A0 card with either **Number of ϑ angles** or **Number of φ angles** larger than 1 is read, all the following control cards up to, but excluding, the next Ax, FR or EN card will be read into a buffer. All these cards are then processed in a loop, over all the different angles of incidence.

If for example, the monostatic radar cross section is to be calculated for ϑ=90° and 0°≤φ≤180°, the following command is used:

```
A0: 0 : 0 : 181 : 1.0 : 0 :  :  : 0.0 : 0.0 : 0.1
FF: 2
EN
```

In this demonstration file, the FF card is read into the buffer and processed 181 times. Through the use of the parameter **Fields calculated only in incident direction** in the FF card the far field is calculated in the direction of the incident wave.

If more than one direction of incidence is to be examined, the right hand side of the linear equation system is changed, but the matrix remains unchanged. Thus it makes sense, by using the CG card, to use Gauss elimination (default if a CG card is not used) which performs a LU-decomposition of the matrix. When the direction of incidence is varied, then only the relatively fast backward substitution has to be performed.

Related tasks
Adding a Plane Wave Source (CADFEKO)

A1 Card

This card defines a voltage source that is placed on a wire segment.

On the **Source/Load** tab, in the **Sources on geometry** group, click the **Voltage source** icon.

From the drop-down list, click the **Voltage on segment (A1)** icon.

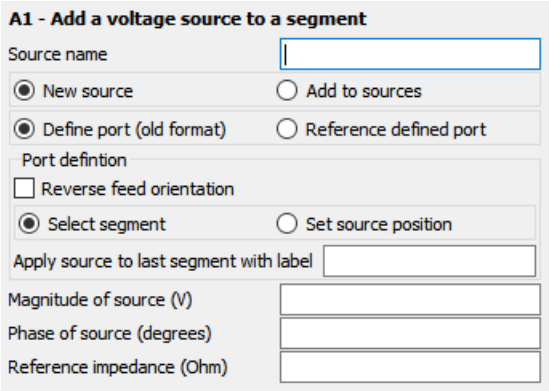


*Figure 811: The **A1 - Add a voltage source to a segment** dialog .*

Parameters:

New source — A new excitation is defined which replaces all previously defined excitations.

Add to sources — A new excitation is defined which is added to the previously defined excitations.

Define port (old format)

- **Reverse feed orientation** — By default, the vector of the voltage lies in the direction from the start of the segment to its end (the direction in which it was created). When this option is checked, the vector of the voltage lies in the opposite direction. This is the direction of the current flow through the segment. The internal EMF (electromagnetic force) of the impressed voltage source is in the opposite direction.
- **Select segment** — When this item is selected, the **Apply source to last segment with label** field becomes active. The label of the segment to which the source must be applied is specified with this text box. If more than one segment has this label, the source is applied to the last segment with this label.

Apply source to last segment with label	Label of the segment to which the source is applied. If more than one segment has the same label then source is applied to the last segment with this label.
Set source position	When this item is selected, the feed segment is determined by specifying the Cartesian coordinates in the **Segment centre** group. These values are in m and are scaled by the SF card if the SF card has **Modify all dimension related values** checked.

Reference defined port	The label of the referenced port defined using the PT card.
Magnitude of source (V)	Magnitude of the voltage source in V.
Phase of source (degrees)	Phase of the voltage source in degrees.
X, Y, Z position	The position of the centre of the feed segment in Cartesian coordinates (only available if **Set source position** is selected.)
Reference impedance (Ohm)	The reference impedance of the excitation is used for S-parameter calculations and is the reference impedance used for realised gain calculations. It is also the default reference impedance used to calculate and display the reflection coefficient in POSTFEKO. If this field is empty or 0 in an S-parameter calculation, the value specified at the SP card is used. For realised gain and reflection coefficient calculations, 50 Ohm will be assumed when the field is empty or 0.

Related tasks
Adding a Voltage Source (CADFEKO)
Related reference
A0 Card
AP Card
PT Card
SF Card

A2 Card

With this card a voltage source is placed at a node between two segments or between a segment and a triangle, ground plane or polygonal plate. It is mostly used to feed wires attached to plates.

On the **Source/Load** tab, in the **Sources on geometry** group, click the **Voltage source** icon.

From the drop-down list, click the **Voltage on vertex (A2)** icon.

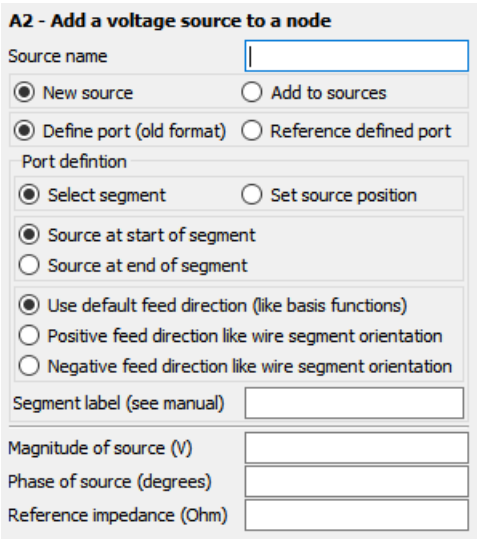


*Figure 812: The **A2 - Add a voltage source to a node** dialog.*

Parameters:

New source
A new source is defined that replaces all previously defined sources.

Add to sources
A new source is defined that is added to the previously defined sources.

Define port (old format)

Select segment
When this item is selected, then the **Segment label** field becomes active. Here one specifies the label of the segment that shall be fed (placed at either the start or end point as determined by the corresponding check box). The source has to be located at a node, either between two segments, or between a segment and a triangle, ground plane or polygonal plate. Only one segment with this label should be declared. If there is more than one segment with this label then only one node will be fed.

Set source position	If this check box is activated, the feed node is determined by specifying its Cartesian coordinates in the **Coordinates of node** group. These values are in m and may be scaled by the SF card. The source orientation is always consistent with the basis function defined over this node (see the information below under **Use default feed direction**).
Reference defined port	The label of the referenced port defined using the PT card.
Source at start of segment	This option is only available when selecting the feed segment by label. If set, it indicates that the feed location is at the start of the wire segment with a matching label.
Source at end of segment	This option is only available when selecting the feed segment by label. If set, it indicates that the feed location is at the end of the wire segment with a matching label.
Use default feed direction	This option is only available when selecting the feed segment by label. If set, it indicates that the positive feed direction is consistent with the basis function setup. For wire/surface junctions (UTD plates, infinite planes with the BO ground, or meshed triangle surfaces), this direction is away from the wire onto the surface. For wire connections between two segments, this direction is from the segment with the lower index to the segment with the higher index.
Positive feed direction like wire segment orientation	This option is only available when selecting the feed segment by label. If set, then the positive feed direction is like the orientation of the wire segment with the specified label.
Negative feed direction like wire segment orientation	This option is only available when selecting the feed segment by label. If set, the positive feed direction is opposite to the orientation of the wire segment with the specified label.
Magnitude of source	Magnitude of the voltage source in V.
Phase of source	Phase of the voltage source in degrees.
Reference impedance (Ohm)	The reference impedance of the excitation is used for S-parameter calculations and is the reference impedance used for realised gain calculations. It is also the default reference impedance used to calculate and display the reflection coefficient in POSTFEKO. If this field is empty or 0 in an S-parameter calculation, the value specified at the SP card is used. For realised gain and reflection

coefficient calculations, 50 Ohm will be assumed when the field is empty or 0.

There may not be more than two segments connected to the node for nodes between segments. For nodes between a segment and a triangle, UTD plate or an infinite plane only one segment is allowed to connect at the node.

Note: The vector direction of the feed is the direction of the current flow through the node. The internal electromagnetic force (EMF) of the impressed voltage is in the opposite direction.

Related tasks
Adding a Voltage Source (CADFEKO)
Related reference
SF Card
SP Card
PT Card

A3 Card

This card realises excitation by a magnetic ring current (TEM-frill) on a segment. It gives an an accurate model of a coaxial feed, but requires both the inner and outer radii.

On the **Source/Load** tab, in the **Sources on geometry** group, click the **Archaic sources** icon.

From the drop-down list, click the **Magnetic frill (A3)** icon.

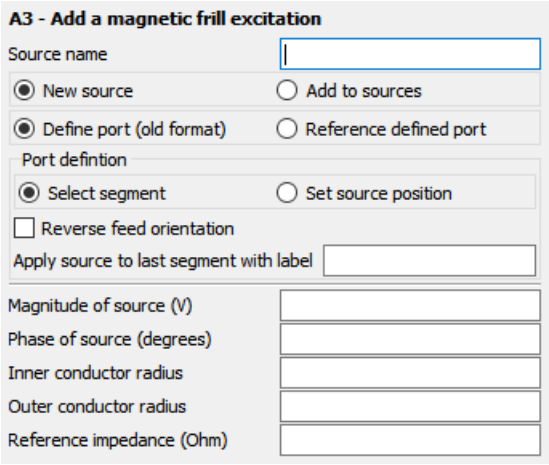


*Figure 813: The **A3 - Add a magnetic frill excitation** dialog.*

Parameters:

New source — A new source is defined that replaces all previously defined excitations.

Add to sources — A new source is defined which is added to the previously defined excitations.

Define port (old format)

- **Reverse feed orientation** — The vector of the excitation points in the direction of the segment by default - from the start point to the end point consistent with how the segment was created. When this option is checked, the orientation of the excitation is reversed.
- **Select segment** — When this item is selected, the **Apply source to last segment with label** text box becomes active. This text box is for specifying the label of the segment on which the TEM frill is placed. If more than one segment has this label, the source is applied to the last segment with this label.
- **Set source position** — When this item is selected, the feed segment is determined by specifying the

	Cartesian coordinates in the **Segment centre** group. These values are in m and are scaled by the SF card if the SF card has **Modify all dimension related values** checked.
Reference defined port	The label of the referenced port defined using the PT card.
Magnitude of source	Magnitude of the voltage source in V.
Phase of source	Phase of the voltage source in degrees.
Inner conductor radius	Radius of the inner conductor of the coaxial feed.
Outer conductor radius	Radius of the outer conductor of the coaxial feed.
Reference impedance (Ohm)	The reference impedance of the excitation is used for S-parameter calculations and is the reference impedance used for realised gain calculations. It is also the default reference impedance used to calculate and display the reflection coefficient in POSTFEKO. If this field is empty or 0 in an S-parameter calculation, the value specified at the SP card is used. For realised gain and reflection coefficient calculations, 50 Ohm will be assumed when the field is empty or 0.

The excitation is not, as in the A2 card, an impressed electric field strength, but is a magnetic ring current.

As a rule of thumb, the radius of the inner conductor must be the same as the radius of the segment. In addition the outer radius should be 2 to 3 times the size of the inner radius. If an impedance Z is desired, then the following relation can be used to determine the outer radius:

$$Z \approx 60\Omega \cdot \ln\left(\frac{\text{Outer conductor radius}}{\text{Inner conductor radius}}\right). \quad (141)$$

For a 50 Ω coaxial line the outer radius should be equal to 2.3 times the inner conductor radius.

Related tasks
Adding a Current Source (CADFEKO)
Related reference
A2 Card
SF Card
SP Card
PT Card

A5 Card

This card specifies excitation by an electric Hertzian dipole

On the **Source/Load** tab, in the **Ideal sources** group, click the **Electric dipole (A5)** icon.

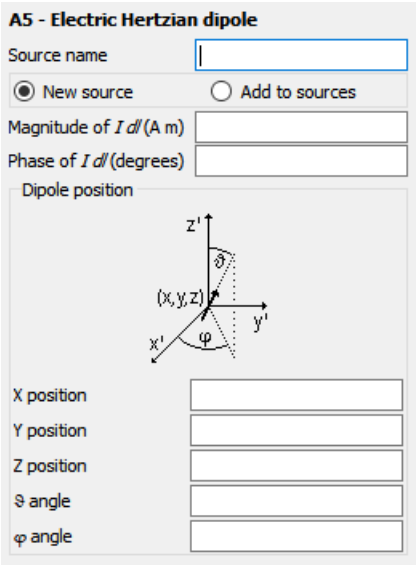


*Figure 814: The **A5 - Electric Hertzian dipole** dialog.*

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.
Magnitude of *I dl* (A m)	Absolute value of the complex amplitude $I \cdot l$ in Am.
Phase of *I dl* (degrees)	Phase of the complex amplitude $I \cdot l$ in degrees.
X, Y, Z position	Coordinates of the position of the dipole in m. These values are optionally scaled by the SF card.
ϑ **angle**	Orientation of the dipole in space. The angle ϑ (in degrees) is the angle between the dipole and the Z axis.
φ **angle**	Orientation of the dipole in space. The angle φ (in degrees) is the angle between the projection of the dipole onto the Z=0 and the X axis.

The dipole moment of the electric dipole is given by

$$p = \frac{Il}{j\omega} \tag{142}$$

The power radiated by the dipole in a free space environment is given by

$$p = \frac{\beta_0^2 Z_{F0} |I|^2}{12\pi} = \frac{\omega^2 \mu_0^2 |I|^2}{12\pi Z_{F0}} \quad (143)$$

Feko, however, considers the properties of the medium in which the dipole is located, as well as the coupling of the dipole with surrounding structures or other sources (for example other Hertzian dipoles in an array antenna), when calculating the power radiated by the Hertzian dipole.

Related tasks

Adding a Electric Dipole Source (CADFEKO)

Related reference

A0 Card

AC Card

AP Card

SF Card

A6 Card

This card specifies excitation by an elementary magnetic Hertzian dipole.

On the **Source/Load** tab, in the **Ideal sources** group, click the **Magnetic dipole (A6)** icon.

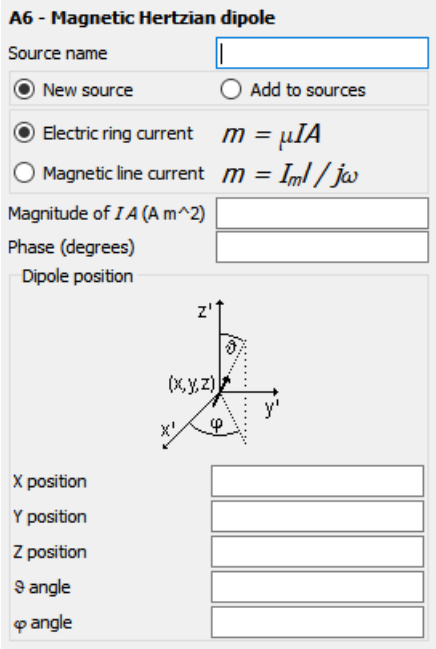


*Figure 815: The **A6 - Magnetic Hertzian dipole** dialog.*

Parameters:

New source
A new excitation is defined which replaces all previously defined excitations.

Add to sources
A new excitation is defined which is added to the previously defined excitations.

Electric ring current
Use the model of an electric ring current for the magnetic dipole (the moment $m = \mu IA$ where I is the loop current and A the enclosed surface area).

Magnetic line current
Use the model of an magnetic line current (the moment $m = \frac{I_m l}{j\omega}$ where I_m is the magnetic line current and l is the length of the dipole).

Magnitude of *I A* (A m^2)
The magnitude of the dipole. For the electric ring current it is $I \cdot A$ in Am^2. For the magnetic line current it is $I_m \cdot l$ in Vm.

Phase (degrees)
Phase of the complex amplitude in degrees.

X, Y, Z position	Coordinates of the position of the dipole in m. These values are optionally scaled by the SF card.
ϑ **angle**	The angle between the dipole and the Z axis in degrees.
φ **angle**	The angle between the projection of the dipole onto the plane Z=0 and the X axis in degrees.

The power radiated by the dipole in a free space environment is given by

$$p = \frac{\beta_0^4 Z_{F0} |IA|^2}{12\pi} \tag{144}$$

Feko, however, considers the properties of the medium in which the dipole is located, as well as the coupling of the dipole with surrounding structures or other sources (for example other magnetic dipoles in an array antenna), when calculating the power radiated by the Hertzian dipole.

Even though the two formulations, electric ring current and magnetic dipole, result in the same near and far fields, if the dipole moment *m* is the same, the radiated potentials are different.

The electric ring current model gives rise to a magnetic vector potential **A** while the magnetic dipole model results in an electric vector potential **F** as well as a magnetic scalar potential ψ.

Related tasks

Adding a Magnetic Dipole Source (CADFEKO)

Related reference

A0 Card

AC Card

AP Card

SF Card

A7 Card

This card specifies a voltage source on an edge between two triangles or at a connection between a single triangle and a PEC ground plane or UTD plate.

On the **Source/Load** tab, in the **Sources on geometry** group, click the **Archaic sources** icon.

From the drop-down list, click the **Voltage between triangles (A7)** icon.

> **Note:** The AE card is much simpler to use and is the recommended card to use for edge sources. (The A7 card is supported only for compatibility with Feko input files that were created before the AE card became available.)

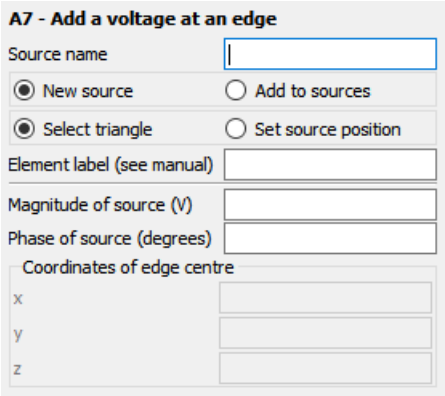


*Figure 816: The **A7 - Add a voltage at an edge** dialog.*

Parameters:

New source — A new excitation is defined which replaces all previously defined excitations.

Add to sources — A new excitation is defined which is added to the previously defined excitations.

Select triangle — If this item is selected, a triangle with label specified in the **Element label (see manual)** text box is searched for. The excitation is placed on the edge positioned opposite the first corner of the triangle. The label must be unique (if possible only one triangle must have this label). If there is more than one triangle with this label then only a single triangle will be fed. Alternatively, when selecting the option, **Set source position**, the feeding edge is determined by specifying its Cartesian coordinates in the **Coordinates of edge centre** group. These coordinates are in meters and optionally scaled by the SF card. The edge must be positioned between two triangles or between a triangle and a ground plane or UTD plate.

Magnitude of source (V) — Absolute value of the voltage source in V.

Phase of source (degrees) Phase of the voltage source in degrees.

If two triangles are connected to the edge, the basis function between these triangles will be excited. The vector direction of the voltage source will be in the same direction as the basis function associated with this edge. This is the direction of the current flow through the edge. The internal EMF (electromagnetic force) of the impressed voltage source is in the opposite direction.

In selected special cases there may be only one triangle connected to the edge. If the edge is positioned in the plane of a polygonal plate or a PEC ground plane (specified with a GF or BO card), the excitation is placed on the appropriate basis function connecting the triangle to the plate/plane. The positive feed direction is then towards the edge.

Related reference

AE Card

BO Card

GF Card

AB Card

This card is used to create a FEM modal excitation, which is the fundamental mode of the associated, infinitely long guided wave structure of the modal port.

On the **Source/Load** tab, in the **Sources on geometry** group, click the **Modal source (AB)** icon.

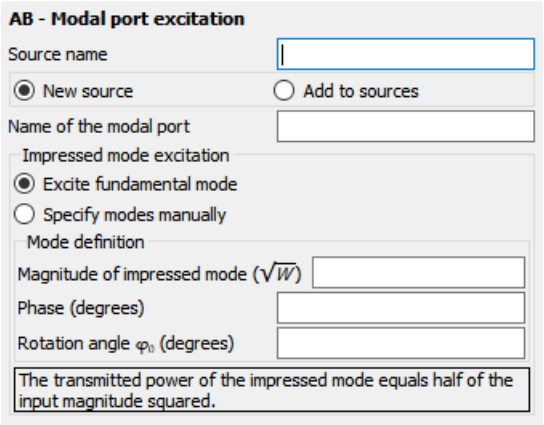


*Figure 817: The **AB - Modal port excitation** dialog.*

Parameters:

New source — A new excitation is defined which replaces all previously defined excitations.

Add to sources — A new excitation is defined which is added to the previously defined excitations.

Name of the modal port — Label of the modal port.

Excite fundamental mode — Select this option to automatically excite the fundamental mode of the waveguide. When this option is selected, the mode type and its indices (*m* and *n*) cannot be specified since they are determined automatically.

Specify modes manually

- **TE-mode** — If this option is checked, a $TE_{m,n}$ mode (also referred to as $H_{m,n}$) is used as excitation. This option is only available when **Excite fundamental mode** has not been selected.
- **TM-mode** — If this option is checked, a $TM_{m,n}$ mode (also referred to as $E_{m,n}$) is used as excitation. This option is only available when **Excite fundamental mode** has not been selected.

TEM-mode	If this option is checked (only available for the coaxial waveguide since TEM modes don't exist in rectangular/circular waveguides), a TEM mode is used as excitation. This option is only available when **Excite fundamental mode** has not been selected.
Mode index m	The index *m* of the $TE_{m,n}$ or $TM_{m,n}$ mode which is impressed at the port. Note that for a rectangular waveguide the index m is related to the $\mathbf{u}_1$ direction (for example from point S1 to S2). For a circular/coaxial waveguide, *m* denotes the angular dependency. This option is only available when **Excite fundamental mode** has not been selected.
Mode index n	The index *n* of the $TE_{m,n}$ or $TM_{m,n}$ mode which is impressed at the port. Note that for a rectangular waveguide the index *n* is related to the $\mathbf{u}_2$ direction (for example, from point S1 to S3). For a circular/coaxial waveguide, *n* denotes the radial dependency. This option is only available when **Excite fundamental mode** has not been selected.
Magnitude of impressed mode	The magnitude of the impressed mode (absolute value of the complex amplitude of the impressed mode). The transmitted power of the impressed mode equals half the input amplitude squared. **Note:** If left empty or set to 0 then the modal port will act as a passive (sink) port for fields incident on the port.
Phase (degrees)	Phase of the impressed mode in degrees.
Rotation angle φ_0 (degrees)	Rotation angle in degrees, applicable to circular and coaxial waveguide port shapes only. Specifies the rotation angle in degrees by which the impressed mode is rotated anti-clockwise with respect to the transverse reference axis.

AC Card

This card inputs data from a `.rsd` file containing the geometry of a transmission line or PCB structure and the current distribution along this line or on the PCB for one or more frequencies.

On the **Source/Load** tab, in the **Ideal sources** group, click the **Impressed current** icon. From the drop-down list, click the **Cable source (AC)** icon.

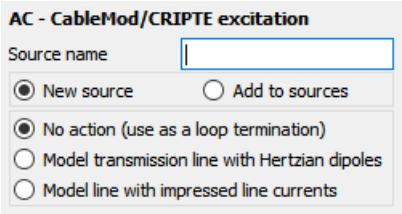


*Figure 818: The **AC - CableMod/CRIPTE excitation** dialog.*

The `.rsd` file is created by transmission line simulation programs CableMod or CRIPTE or by the PCB code, PCBMod or by the export of currents with the OS card. The excitation is due to the electromagnetic fields radiated by these line currents (the CM card allows the treatment of electromagnetic fields coupling into lines).

Parameters:

New source	This card inputs data from a `.rsd` file containing the geometry of a transmission line or PCB structure and the current distribution along this line or on the PCB for one or more frequencies. A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.
No action	No execution, do not read the `.rsd` file. This option is used to specify the end of the frequency loop, see comments below.
Model transmission line with Hertzian dipoles	The line geometry, frequency and currents are read from the `.rsd` file, and the line is modelled with an array of Hertzian dipoles (see the A5 card). The number of dipoles per line segment is specified with the parameter **Number of dipoles per transmission line**. Note that this model is only valid if the line segments do not make electrical contact with any conducting surfaces. (All the segments in the `.rsd` file must be of the type *intern* and not loaded.)
Model line with impressed line currents	The line geometry, frequency and currents are read from the `.rsd` file, and the line is modelled with a continuous current distribution using one AI card per line segment. (The AI cards are created automatically by PREFEKO when importing the `.rsd` file.) If a line segment has a loaded endpoint it is automatically modelled

	by an AV card to allow the electrical contact. The radius of the impressed current element can be set in the parameter **Radius of impressed current**. This parameter is optional and is passed on to the AI and AV cards. If the parameter is zero or empty a current filament approximation is used.
Source translation (directions)	When importing transmission line currents from CableMod or CRIPTE or PCB currents from PCBMod, then the coordinate system of these programmes as represented in the `.rsd` file is used and the source is imported at this position in Feko. Here an offset can be specified which translates the source in the X direction, Y axis and Z axis. Standard Feko units are used for these offsets (that is metres, but scaled accordingly if a factor is set at the SF card). Note that the units as specified in the `.rsd` file are not applicable here for the translation parameters (only to the import of the data).
Rotation about axes	Like the translation described above, an imported source can here be rotated and thus positioned arbitrarily. The rotation angles are in degrees, and uses the same definition as the AR and CG cards.
File name	The name of the `.rsd` file.
Use adaptive frequency sampling	Only read the minimum and maximum frequency from the `.rsd` file and obtain a continuous solution in this frequency band using adaptive frequency sampling. If this option is used only one AC card is permitted in the `.pre` file and no FR cards.
Maximum number of discrete frequency points	When using adaptive frequency sampling, the maximum number of sample points can be specified here. See also the discussion on adaptive frequency sampling for the FR card.
Minimum frequency stepping	When using adaptive frequency sampling it could be necessary to specify the minimum allowable frequency between samples. See also the discussion on adaptive frequency sampling for the FR card.

If the imported `.rsd` file contains currents for several frequencies, the option **New source** must be chosen as the AC card then results in a frequency loop and currents with different frequencies cannot be superimposed. (If it is not chosen, PREFEKO will give an error). The frequency is defined in the `.rsd` file, thus the preceding FR cards are ignored when processing an AC card.

All commands following the AC card in the Feko input file (for example FF, FE, OS, GF, BO and so forth) are processed within a frequency loop through all the frequencies in the `.rsd` file. The loop is terminated by any of the following three cards (these cards are not included in the loop they terminate).

- AC: importing a new `.rsd` file, or using the **No action (use as a loop termination)** option
- FR: manually setting a new frequency
- EN: end of the Feko input file

For example, if a `.rsd` file must be read and the near field calculated for each frequency, the input file could be as follows:

```
AC ... ** Read the *.rsd file
FE ... ** Calculate the near field
EN ** End
```

However, if , for example, a metal plate is to be modelled, which is excited first by an impressed line current and then also by a plane wave (in each case the near fields and the currents on the plate must be written to the output file), the input file would be as follows:

```
** Excitation by a line current
AC ... ** Read the *.rsd file
FE ... ** Calculate the near field
OS ... ** Output the currents
** Excitation by a plane wave
FR ... ** Set the frequency and terminate
AC loop
A0 ... ** Specify plane wave excitation
FE ... ** Calculate the near field
OS ... ** Output the currents
** End
EN
```

Related reference

A5 Card

AI Card

AV Card

AE Card

This card specifies an excitation at an edge between triangular surface elements.

On the **Source/Load** tab, in the **Sources** group, click the **Voltage source** icon. From the drop-down list, click the **Voltage on edge (AE)** icon.

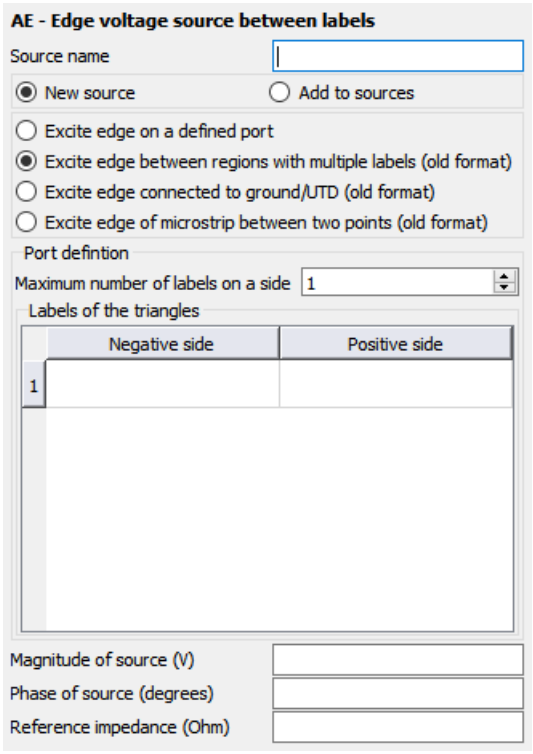


*Figure 819: The **AE - Edge voltage source between labels** dialog.*

The location of the feed point and the positive feed direction are substantially easier to specify than the A7 card. In addition it is possible to specify a feed edge that contains a number of triangle edges.

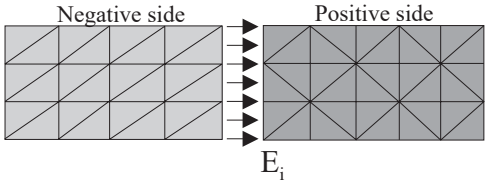


Figure 820: Example of the use of the AE card.

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.

Excite edge on a defined port	The name of the referenced port defined using a PT card.
Excite edge between regions with multiple labels (old format)	The excitation is placed on the edge between the regions with labels specified in **Negative side** and **Positive side**. Use the field **Maximum number of labels on a side** to increase the number of rows available in the table. The positive source direction is from the **Negative side** towards the **Positive side**.
Excite edge connected to ground/UTD (old format)	Excite the edges of metallic triangles with the labels specified in the **Negative side** or **Positive side**. The edges of these triangles are connected to UTD surfaces or to a PEC ground plane (as specified with a BO or GF card).
Excite edge of microstrip between two points (old format)	This is a special microstrip line feed. The excitation is placed on all edges on a line between points (previously specified with DP cards). The points are specified in the **Start point of edge** and **End point of edge** dialogs. A GF card with a conducting ground must be present. **Note:** The dialog layout and visibility of options will differ according to the selection in the **Excite edge** group.
Meshed surface represents positive feed side	By default, the feed direction is such that the meshed surface represents the negative feed side. The vector direction of the current is then towards the UTD or ground. When this option is checked, the feed orientation is reversed.
Magnitude of source (V)	Magnitude value of the voltage source in V.
Phase of source (degrees)	Phase of the voltage source in degrees.
Reference impedance (Ohm)	The reference impedance of the excitation is used for S-parameter calculations and is the reference impedance used for realised gain calculations. It is also the default reference impedance used to calculate and display the reflection coefficient in POSTFEKO. If this field is empty or 0 in an S-parameter calculation, the value specified at the SP card is used. For realised gain and reflection coefficient calculations, 50 Ohm will be assumed when the field is empty or 0.

The positive source direction as used above is in the direction of the current flow through the edge. The internal EMF (electromagnetic force) of the impressed voltage source is in the opposite direction.

It should be noted that the edge between the surfaces with labels **Negative side** and **Positive side** does not have to be straight. It is possible, for example, to excite two half cylinders against each other. If an impedance must also be applied to the edge, the AE card can be combined with the LE card.

For the case where the option, **between regions with multiple labels**, is selected more than two surfaces may be connected to a feed edge. Examples for this as seen in the figures are:

- one surface fed against three others
- two surfaces fed against two other surfaces
- and a feeding edge on a junction between three surfaces.

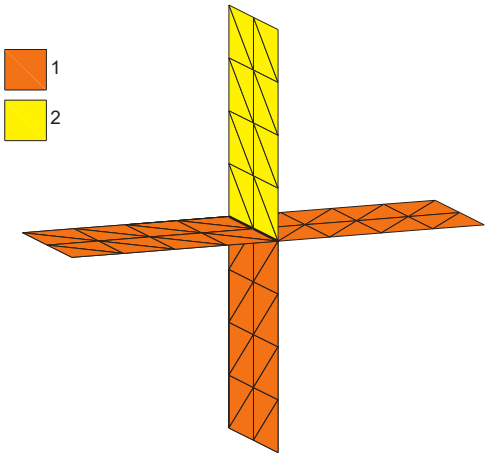


Figure 821: Example of a feed edge where more than one surface is connected on one side of the feed.

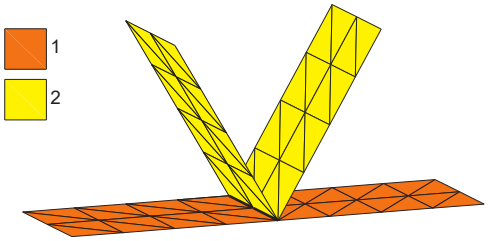


Figure 822: Example of a feed edge where more than one surface is connected on both sides (negative and positive) of the feed.

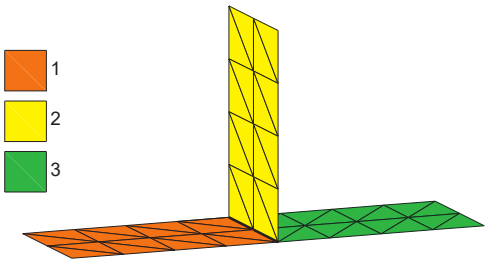


Figure 823: Example of a feed edge of three surfaces with different labels, where for instance label 2 could be fed against labels 1 and 3, but also label 1 could be fed against 2 and 3, or 3 could be fed against 1 and 2.

Related reference
PT Card

AF Card

This card defines a uniform electric current filament impressed between two arbitrary points inside of the FEM region (it does not have to coincide with the edges of tetrahedra). This can be used to excite for instance a patch antenna.

On the **Source/Load** tab, in the **Sources on geometry** group, click the **Current source (AF)** icon.

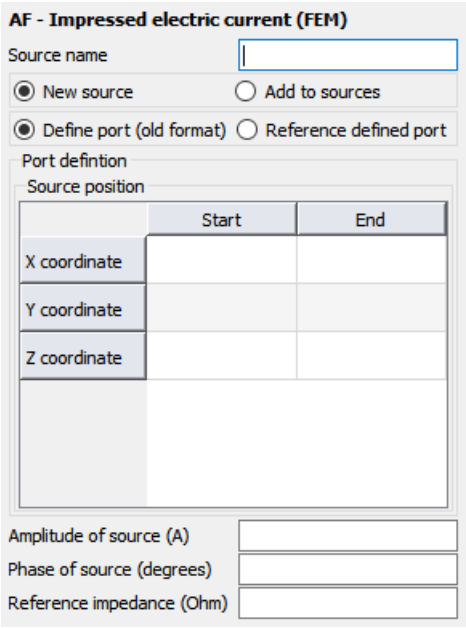


*Figure 824: The **AF - Impressed electric current (FEM)** dialog.*

Parameters:

New source — A new excitation is defined which replaces all previously defined excitations.

Add to sources — A new excitation is defined which is added to the previously defined excitations.

Define port (old format)

- **Source position** — When this item is selected, the feed segment is determined by specifying the Cartesian coordinates in the **Segment centre** group. These values are in m and are scaled by the SF card if the SF card has **Modify all dimension related values** checked.

Reference defined port — The label of the referenced port defined using the PT card.

Amplitude of source (A) — Amplitude in A of the impressed line current.

Phase of source (degrees)	Phase of the current in degrees.
X, Y, Z coordinate	Coordinates of the start and end points in m. (Note that all the coordinate values are optionally scaled by the SF card.)
Reference impedance (Ohm)	The reference impedance of the excitation is used for S-parameter calculations and is the reference impedance used for realised gain calculations. It is also the default reference impedance used to calculate and display the reflection coefficient in POSTFEKO. If this field is empty or 0 in an S-parameter calculation, the value specified at the SP card is used. For realised gain and reflection coefficient calculations, 50 Ohm will be assumed when the field is empty or 0.

The following restrictions apply when using the impressed current elements of the AF type inside a FEM region:

- The electric current source would usually be connected to metallic surfaces at its terminals, but this is neither enforced nor checked in Feko. If a physical connection to a metallic structure is required, then the user should ensure that the feed terminals are also attached in the discretised model.
- Input impedance is computed for this source from the line integral over the electric field solution between the terminals of the source. The length of the impressed current should therefore be small compared to the shortest wavelength in the band of interest to render a reasonable accuracy.
- An intrinsic limitation of this model is that no radius is taken into account, therefore the field is singular in the vicinity of the filament. This affects the accuracy of the computed input impedance of the source.

Related tasks
Creating a FEM Line Port (CADFEKO)
Related reference
SF Card
PT Card
SP Card

AI Card

This card defines an impressed current source that varies linearly between the values at the start and end points.

On the **Source/Load** tab, in the **Ideal sources** group, click the **Impressed current** icon. From the drop-down list, click the **Impressed current in space (AI)** icon.

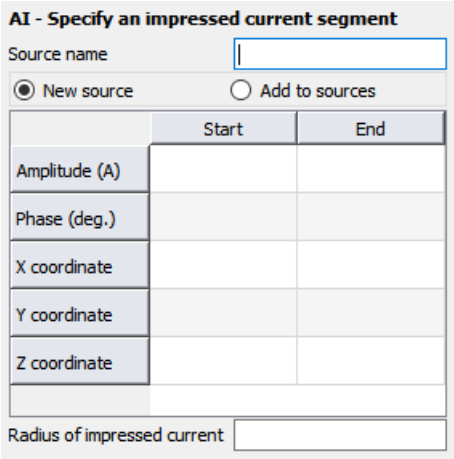


*Figure 825: The **AI - Specify an impressed current segment** dialog.*

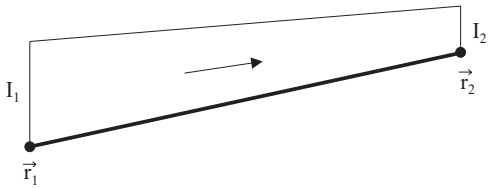


Figure 826: Impressed line current with a linear current distribution.

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.
Amplitude (A)	Amplitude $\|I_1\|$ in A of the current at the start point, $\mathbf{r}_1$, and end point, $\mathbf{r}_2$.
Phase (deg.)	Phase of the current at the start and end points in degrees.
X, Y, Z coordinate	Coordinates of the start and end points in m. (Note that all the coordinate values are optionally scaled by the SF card.)
Radius of impressed current	This parameter is optional. If specified, and different from zero, this value gives a finite wire radius for the impressed current element. Feko then assumes that the current is uniformly

distributed on the wire surface and uses the exact wire integral. If this parameter is not specified, the current filament approximation is used. (This value is optionally scaled by the SF card.)

The following restrictions apply when using the impressed current elements:

- It is not possible to attach the impressed current to a wire segment. (If the impressed current is making electrical contact with a triangular surface current element, the AV card should be used.)
- When modelling dielectric bodies with the surface equivalence principle, the current element must be in the free space medium, that is outside the dielectric bodies. (The material parameters of this medium can, however, be set with the EG and/or GF cards).
- When used with the spherical Green's function, the current element must be outside the dielectric spheres.
- The current segments may be joined with each other and with the AV card to form long paths and/or closed loops. The point charges that arise when the current does not go to zero at an end point or when there is a current discontinuity at a connection point, are not taken into consideration. This is required to model, for example, the case where radiating lines are terminated in a non-radiating structure. If these charges must be considered explicitly, the line current should be modelled by a row of Hertzian dipoles (see the A5 card). Note, however, that the constant line charge along the current segment is correctly taken into account.
- If several of these current elements are used, the total radiated power (required to calculate, for example, the far field gain/directivity) can only be calculated accurately if the mutual coupling between segments is taken into account. Due to neglecting the point charges at the ends of the segments, the coupling cannot be determined accurately. If exact values of the radiated power are required, it should be determined by integrating the far field (see the FF card). It should be noted that, for example, the computed near and far fields (the actual field strength values), the induced currents, coupling factors and losses are computed correctly.

Related tasks

Adding an Impressed Current Source (CADFEKO)

Related reference

A5 Card

AV Card

EG Card

FF Card

GF Card

SF Card

AJ Card

This card uses current data calculated for a PCB as an impressed current source. The data is read from an Altair PollEx `.rei` file.

On the **Source/Load** tab, in the **Equivalent sources** group, click the **PCB source (AJ)** icon.

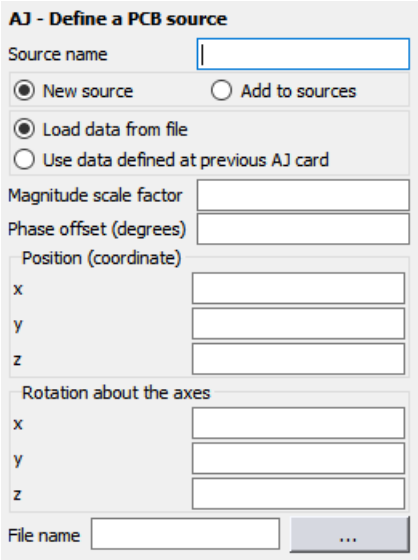


*Figure 827: The **AJ - Define a PCB source** dialog.*

This card is used for importing the trace/via current data calculated for a PCB and using this data as impressed line currents in Feko. This provides an interface between PollEx and Feko, where PollEx can be used to analyse complex PCB structures, and Feko to calculate the radiated emissions.

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.
Load data from file	Read the trace/via currents from a radiated emission interface (`.rei`) file created with PollEx.
Use data defined at previous AJ card	When using multiple AJ cards (different radiating PCBs in the same model) where the current data is identical, it is allowed to load the data just once and at subsequent AJ cards to check this option. The last defined PCB will be used and memory can be saved (no need to store it again). Note that it is still possible to set the PCB position, orientation as well as amplitude and phase individually.
Magnitude scale factor	This parameter is used to scale the magnitude of the current values by a constant value.

Phase offset (degrees)	This parameter specifies a constant additional phase for the current values in degrees.
Position (coordinate)	In this group the X, Y and Z coordinates of the source point (the position where the PCB is located) are entered. These values are affected by the scale factor of the SF card if used. In the XY plane, the source point is the mid-point of a rectangular or circular PCB or the first defined node of a polygon-shaped PCB. The Z coordinate of the source point defines the bottom of the PCB, with the layers added in the positive Z direction.
Rotation about the axes	In this group the angles with which the imported PCB is rotated around the X, Y and Z axes are entered in degrees.
File name	The name of the PollEx `.rei` input file.

Related tasks
Adding a PCB Source (CADFEKO)
Related reference
SF Card

AK Card

This card defines a voltage source to a radiating cable with or without irradiation considered.

On the **Source/Load** tab, in the **Sources on geometry** group, click the **Voltage source** icon.

From the drop-down list, click the **Voltage on cable (AK)** icon.

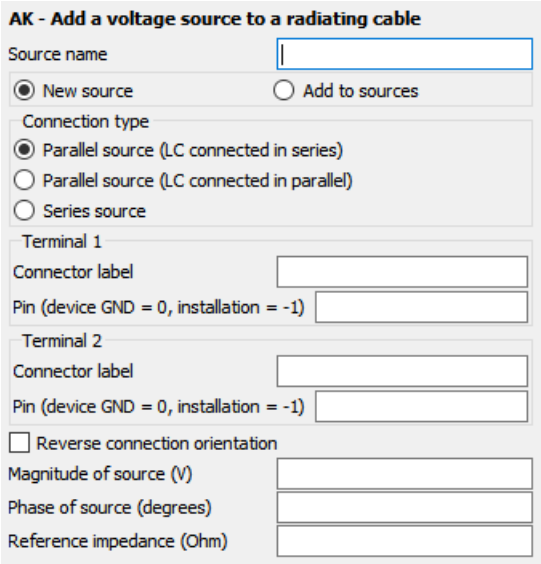


*Figure 828: The **AK - Add a voltage source to a radiating cable** dialog.*

Parameters:

New source
A new excitation is defined which replaces all previously defined excitations.

Add to sources
A new excitation is defined which is added to the previously defined excitations.

Connection type
The voltage source can be connected in more than one configuration. The selected configuration influences the placement of an LC load if it is connected between the same pins.

- **Parallel source (LC connected in series)** The source is connected between a pair of cable connector pins. All cable interconnections defined using a CI card will be placed in parallel with the source. An LC load between the same two pins would result in a load being connected in series with the source. During S-parameter calculations an LC load will be replaced with the system impedance to perform the S-parameter calculation. Also see Connection Types.

	Parallel source (LC connected in parallel)	The source is connected between a pair of cable connector pins. A LC load and all cable interconnections defined between the same pins will be connected in parallel with the source. Also see Connection Types.
	Series source	The source is added in series between a cable connector pin and any other defined connections such as LC loads and/or CI cable interconnections (or a parallel connected voltage source). Also see Connection Types.

> **Note:** Any cable interconnections (CI card) or loads (LC cards) will still be connected in parallel.

Terminal 1	**Connector label**	The connector label that uniquely identifies the connection of the cable path section.
	Pin (device GND = 0, installation = -1)	The pin number identifies the conductor associated with the cable path section as defined by **Connector label** in the CS card. If the pin is set to 0, the connector pin is connected to the local circuit device ground. If the pin is set to -1, the connector pin is connected to the global installation.
Terminal 2	**Connector label**	The connector label that uniquely identifies the connection of the cable path section.
	Pin (device GND = 0, installation = -1)	The pin number identifies the conductor associated with the cable path section as defined by **Connector label** in the CS card. If the pin is set to 0, the connector pin is connected to the local circuit device ground. If the pin is set to -1, the connector pin is connected to the global installation.

Reverse connection orientation	By default the positive terminal will be connected to the pin defined at **Terminal 1** and the negative terminal of the source will be connected to the pin defined at **Terminal 2**. When this option is selected the orientation of the source is reversed.
Magnitude of source	Magnitude of the voltage source in V.
Phase of source	Phase of the voltage source in degrees.
Reference impedance (Ohm)	The reference impedance of the excitation is used for S-parameter calculations and is the reference impedance used for realised gain calculations. It is also the default reference impedance used to calculate and display the reflection coefficient in POSTFEKO. If this field is empty or 0 in an S-parameter calculation, the value specified at the SP card is used. For realised gain and reflection coefficient calculations, 50 Ohm will be assumed when the field is empty or 0.

Related reference

Cable Schematic Elements (CADFEKO)

CI Card

CS Card

LC Card

Cable Pin Numbers

Adding a voltage source to a cable requires understanding of the conductor to cable conductor pin relation and rules.

Rules regarding pin numbering

- Pin 0 always corresponds to the geometry or meshed model.
- Use increasing pin numbers starting at the innermost conductor of a simple cable type and counting towards the outside.
- Use increasing pin numbers in the order of the cross section definition inside the cable bundle.

Pin numbering used for cable modelling can best be described by examples. The following set of examples illustrate how the pin numbering is set.

Example 1: An insulated single conductor above ground

The model consists of the following:

Sub-circuit 1	Single core (Pin 1) with geometry or meshed model as reference (Pin 0).

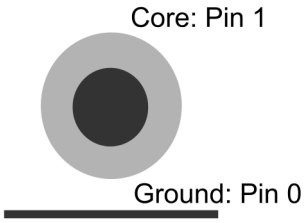


Figure 829: An insulated single conductor above ground.

Example 2: Coaxial cable above ground

The model consists of the following:

Sub-circuit 1	Outside of shield (Pin 2) with geometry or meshed model as reference (Pin 0).
Sub-circuit 2	Coaxial core (Pin 1) with the inside of shield (Pin 2) as reference.

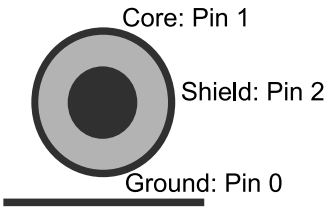


Figure 830: A coaxial cable above ground.

Example 3: Ribbon cable with n cores

The model consists of the following:

Sub-circuit 1	Cores 1..n (excluding Pin i) with Pin i as reference.

Figure 831: A ribbon cable with n cores.

Example 4: Complex bundle (mixed) cable

The model consists of the following:

Bundle1 (shielded)	Single1, Single2, Coax1, Ribbon1, Coax2, Bundle2 and Bundle3.
Ribbon1	3 cores.
Bundle2 (not shielded)	Single3, Single4 and Single5.

Bundle3 (shielded) Single6 and Coax3.

Regarding the loading or excitation for sub-circuit 2, in this example it is only allowed between 1, 2, the outside of the shield of coaxial cable 4, 5, 6, 7, the outside of the shield of coaxial cable 9, 10, 11, 12 and the outside of the shield of the coaxial cable 16 and the inside of the shield of the bundle 17.

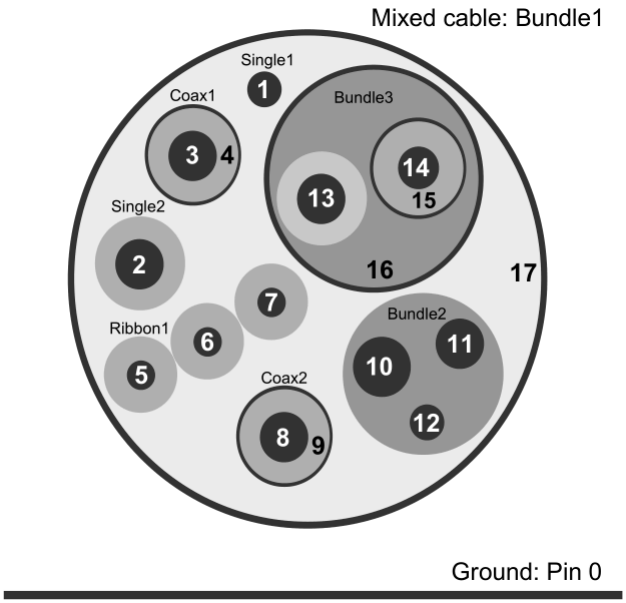


Figure 832: Complex (mixed) cable bundle with single conductors, ribbon cables and coax.

Table 68: Allowed loading between connector pins within the same sub-circuit and its reference pin.

Sub-circuit number	Pin numbers	Reference pin
1	$17_{outside}$	0 (geometry/meshed model).
2	1, 2, $4_{outside}$, 5, 6, 7, $9_{outside}$, 10, 11, 12 , $16_{outside}$	17_{inside}
3	3	4_{inside}
4	8	9_{inside}
5	13, $15_{outside}$	16_{inside}
6	14	15_{inside}

Connection Types

The connection types and orientation of the source and loading must be understood and applied correctly in order to have the desired excitation and loading of the cable bundle.

In all the examples a load (defined with the LC card) and two cable interconnections (defined with the CI card) are defined to illustrate the circuit configuration. These additional elements are not required for the parallel connection, but at least one additional element is required for the series connected source.

The examples illustrate the connections between two pins. An illustration of two pins is shown in the figure below, but note that the pins defined at each of the terminals could also be located on two different connectors (from different cable sections).

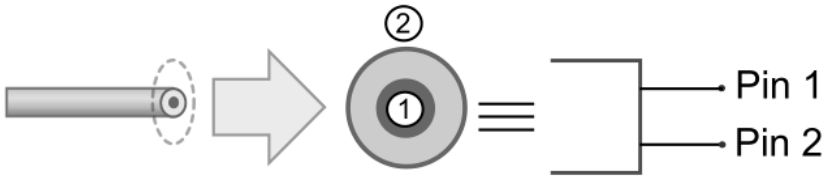


Figure 833: Illustration of a cable, its pin numbering and the schematic terminal pins it represents.

Parallel source (LC connected in series):

The example consists of the following set of elements:

- A parallel connected source where an LC load would be connected in series. The default connection orientation is used (positive terminal of the source is connected to the pin defined at the first terminal).
- An LC load is defined between the two pins.
- Two cable interconnections are defined between the two pins.

> **Note:** During S-parameter calculations the LC load will be replaced with a load with the value of the reference impedance for the port.

The figure below is an illustration of the connection between the two connector pins.

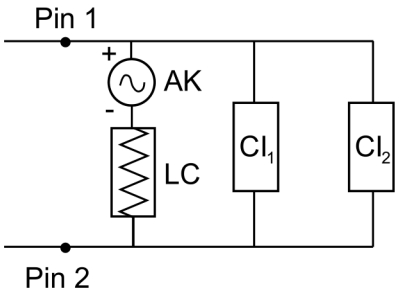


Figure 834: The Parallel source connection type where the LC load is connected in series with the AK source.

Parallel source (LC connected in parallel):

The example consists of the following set of elements:

- A parallel connected source where an LC load would be connected in parallel with the source. The connection orientation has been reversed so that the negative terminal of the source is connected to the pin defined at the first terminal.
- An LC load is defined between the two pins.
- Two cable interconnections are defined between the two pins.

Note: During an S-parameter calculation the LC load will not be replaced, but an additional load with the value of the source reference impedance will be added in series with the source. This load is indicated within a dashed line since it is only present during S-parameter calculations.

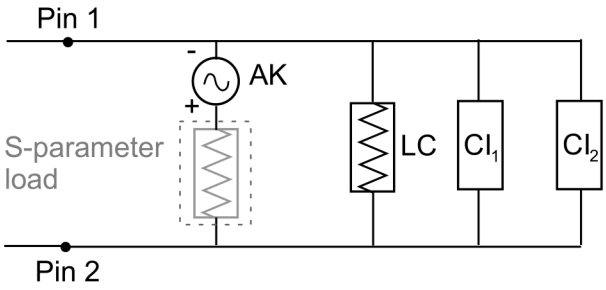


Figure 835: The Parallel source connection type where the LC load is connected in parallel with the AK source.

Series source

The example consists of the following set of elements:

- A series connected source. The connection orientation has been reversed so that the negative terminal of the source is connected to the pin defined at the first terminal.
- An LC load is defined between the two pins.
- Two cable interconnections are defined between the two pins.

Note: During an S-parameter calculation the LC load will not be replaced, but an additional load with the value of the source reference impedance will be added in series with the source. This load is indicated within a dashed line since it is only present during S-parameter calculations.

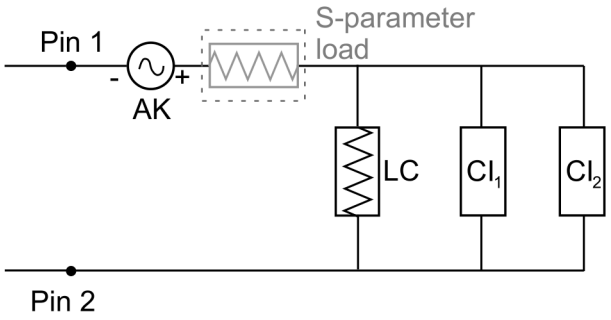


Figure 836: The source is placed in series with everything that is connected at Pin 1.

Series and parallel source

The last example is a more complicated example where two sources are added between the same pins. The example consists of the following set of elements:

- A parallel connected source where an LC load would be connected in parallel with the source. The connection orientation has been reversed so that the negative terminal of the source is connected to the pin defined at the first terminal.
- A series connected source connected to the pin defined by the second terminal of the source defined above. The default connection orientation is used (positive terminal of the source is connected to the pin defined at the first terminal).
- An LC load is defined between the two pins.
- Two cable interconnections are defined between the two pins.

Note: During an S-parameter calculation the LC load will not be replaced, but an additional load with the value of the source reference impedance will be added in series with the source. This load is indicated within a dashed line since it is only present during S-parameter calculations.

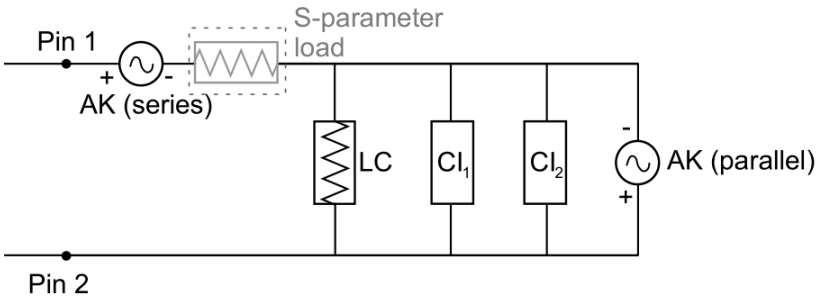


Figure 837: Two sources are placed between the same pins. The one source is connected in parallel and the other in series.

AM Card

This card uses model solution coefficients to define an impressed current source. The model solution coefficients are imported from a `.sol` file.

On the **Source/Load** tab, in the **Equivalent sources** group, click the **Solution coefficient source (AM)** icon.

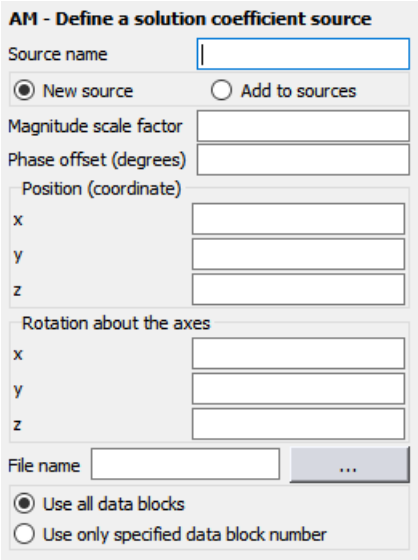


*Figure 838: The **AM- Define a solution coefficient source** dialog.*

A `.sol` file is created by the Solver when an MD card is requested.

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.
Magnitude scale factor	This parameter is used to scale the magnitude of the current values by a constant value.
Phase offset (degrees)	This parameter specifies a constant additional phase for current values in degrees.
Position (coordinate)	In this group, the X, Y and Z coordinates of the impressed current source are entered. These values are affected by the scale factor of the SF card if used.
Rotation about the axes	In this group, the angles with which the impressed current source is rotated around the X, Y and Z axes, are entered in degrees.
File name	The name of the `.sol` input file.

Use all data blocks	Import all data blocks from the specified `.sol` file. The data is interpolated for use at the operating frequency.
Use only specified data block number	Use the data from the nth frequency block in the `.sol` file.

Related reference

MD Card

SF Card

AN Card

This card defines a voltage source that can be added to any port of a general non-radiating network and that does not have a connection to geometry.

On the **Source/Load** tab, in the **Sources on geometry** group, click the **Voltage source** icon.

From the drop-down list, click the **Voltage on network (AN)** icon.

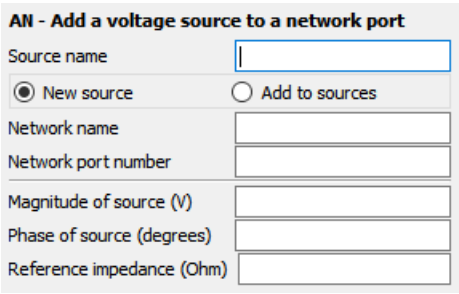


*Figure 839: The **AN - Add a voltage source to a network port** dialog.*

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to all previously defined excitations.
Network name	The network or transmission line name, which with the network port number, will uniquely identify the connection terminals.
Network port number	The network port number, which with the network or transmission line name, will uniquely identify the connection terminals.
Magnitude of source (V)	Magnitude of the voltage source in V.
Phase of source (degrees)	Phase of the voltage source in degrees.
Reference impedance (Ohm)	The reference impedance of the excitation is used for S-parameter calculations and is the reference impedance used for realised gain calculations. It is also the default reference impedance used to calculate and display the reflection coefficient in POSTFEKO. If this field is empty or 0 in an S-parameter calculation, the value specified at the SP card is used. For realised gain and reflection coefficient calculations, 50 Ohm will be assumed when the field is empty or 0.

Adding of a voltage source to an internal network port, which is not connected to geometry, is completely defined using the AN card. The connection is specified by the combination of the network name and port number that will be excited.

Note that for excitation of network ports connected to geometry, the A1 (voltage source on a segment), A2 (voltage source on a node), A3 (magnetic frill excitation) and AE (edge excitation) sources are supported.

Related concepts

Network Schematic View (CADFEKO)

Related reference

A1 Card

A2 Card

A3 Card

AE Card

AN Card

SP Card

AP Card

This card defines a planar, cylindrical or spherical aperture of measured or calculated field values that is converted by PREFEKO internally into an equivalent array of electric and magnetic dipoles (A5/A6 cards).

On the **Source/Load** tab, in the **Equivalent sources** group, click the **Aperture source (AP)** icon.

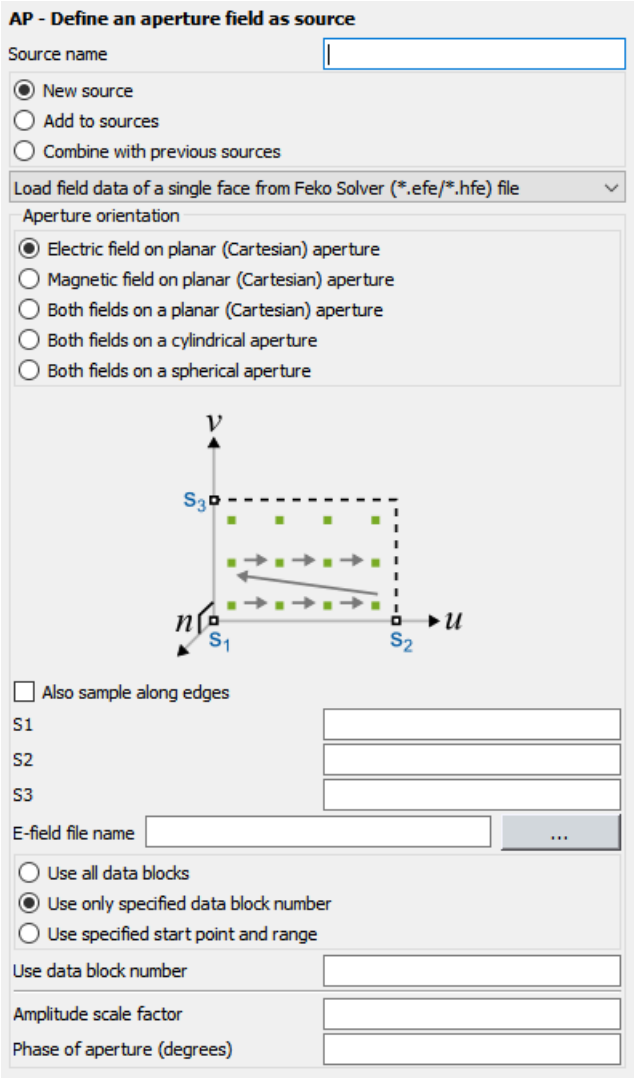


*Figure 840: The **AP - Define an aperture field source** dialog.*

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.

Combine with previous sources Combine multiple aperture field definitions to create a single near field source.

Load field data from

Feko Solver (*.efe/*.hfe) file Read the field values from `.efe`/`.hfe` format files calculated by Feko. The files should contain values describing a single face. The field data files are requested with the DA card.

Cartesian boundary from Feko Solver (*.efe/*.hfe) file Read the field values from `.efe`/`.hfe` format files calculated by Feko. The files should contain values describing a Cartesian boundary. The field data files are requested with the DA card.

CST near field scan (CST NFS) Read the field values from CST NFS format files.

Sigrity (*.nfd) input file Read the field values from a `.nfd` file.

MVG (*.mfxml) measurement file Read the field values from a `.mfxml` file.

ASCII text file Read the field values from an ASCII file.

The field data follows in the (*.pre) input file The field values are specified in the `.pre` file. The format of this file is described with the AR card.

Aperture orientation Here one may select a planar, cylindrical or spherical aperture. For a planar aperture one may elect to use only electric or magnetic fields (this radiates in both directions, see comments below).

Also sample along edges
This item is used to determine if dipoles are allowed to be positioned on the edges of the aperture or not (see figures below). When checked the outer dipoles will be positioned exactly on the edges. When unchecked the dipoles will be positioned half an increment away from the edges. Dipoles should not be positioned on the edges of two apertures that have a common side, otherwise two dipoles may have the same location and polarisation. If this is the case the power calculation in Feko will fail.

Swap source and field validity regions
This item is used for Cartesian boundary, spherical and cylindrical apertures to interchange the side of the aperture

where the fields are considered equivalent to the measured or calculated field values.

S1, S2 and S3
These text boxes are for input points (see the DP card) that define the orientation of the aperture. The figure on the dialog will depict the orientation. For a planar aperture the input points define the position of the origin and the direction of the $\hat{u}_2$ and $\hat{u}_3$ directions respectively. (The field data is assumed to vary first along the $\hat{u}_2$ direction.) For cylindrical and spherical apertures they define the origin and the direction of the local Z axis and X axis respectively. All angles are relative to these axes.

E-field file name The input file name from which the electric field data must be read. This may be either an `.efe` file or a text file (with units V/m).

H-field file name The input file name from which the magnetic field data must be read. This may be either an `.hfe` file or a text file (with units A/m).

File name The input file name from which the Sigrity (`.nfd`) or MVG (`.mfxml`) files are read.

Use all data blocks Import all data blocks from the specified `.efe`/`.hfe`, `.nfs`, `.nfd` or `.mfxml` file. The data is interpolated for use at the operating frequency.

Use only specified data block number Use the data from the nth frequency block in the specified `.efe`/`.hfe`, `.nfs`, `.nfd` or `.mfxml` file.

Use specified start point and range Select a specific near field pattern in a `.efe`/`.hfe` (a single face), `.pre` or ASCII text file.

Start from point number The number of the first field point to be used for the aperture. If set to 1, field values are read from the start of the file, for larger values the first point number-1 values (`.efe` and `.hfe` files) or lines (text files) are ignored. This may be used, for example, if the data file contains the field values for more than one frequency. This corresponds to the line number if all non-data lines are stripped from the file. The **Start from point number** field is not used if the field data is obtained from the .pre input file

	Number of points along...	These two text boxes specify the number of field points along each of the two axes of the aperture.
Include fields on boundary		When importing Cartesian boundary near field data from file, a face (or faces) of the near field boundary can be ignored. Clear the relevant check box to exclude the face.
Amplitude scale factor		A constant by which the amplitudes of all the dipoles in the aperture are scaled.
Phase of aperture		A constant phase added to all dipoles in the aperture.

The aperture is based on the equivalence principle. This states that the sources and scatterers inside a given volume can be removed, and modelled by placing the equivalent currents $\mathbf{J}_s = \hat{n} \times \mathbf{H}$ and $\mathbf{M}_s = -\hat{n} \times \mathbf{E}$ on the enclosing surface. The vector $\hat{n}$ is a unit vector, normal to the surface, and points towards the exterior region. The fields in this region are the same as the original fields, while those in interior region are zero.

Field values are read from the data files (with a possible offset specified with **Start from point number**) or the `.pre` input file and converted to equivalent electric (magnetic fields) and magnetic (electric field) dipoles at these points.

> **Note:** All angles are read from the data, but no distance values.

Therefore for planar apertures the positions are calculated entirely from the specified points (S1, S2 and S3). For cylindrical apertures the specified points (S1 and S2) define the extents of the aperture along the local $\hat{z}$ direction and S1–S3 specifies the direction of the X axis as well as the radius of the cylinder. The points are placed at the φ-values listed together with the field data. For spherical apertures, S1–S2 specifies the direction of the Z axis and S1–S3 the X axis. S2 and S3 must be positioned on the same radius which is also the radius of the field points. In this case both ϑ and φ are read with the data.

Figure 841 and Figure 842 show the application of the equivalence principle to a planar aperture. In both figures there are the same number of field points along the two orthogonal directions. If the **Also sample along edges** check box is selected, the first point is positioned at S1 with the following points in the direction of S2 as shown by the indices. When the **Also sample along edges** check box is cleared, the first point is positioned offset from S1. The normal vector is calculated from $\hat{n} = \hat{u}_2 \times \hat{u}_3$ with $\hat{u}_2$ and $\hat{u}_3$ as defined in the figure.

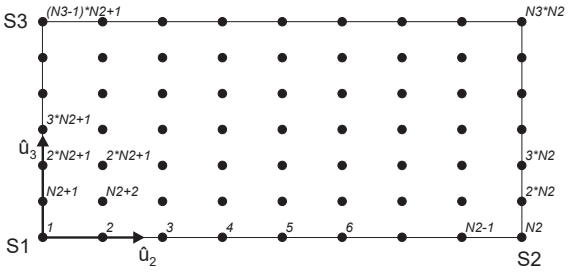


*Figure 841: Location of the equivalent dipoles on a planar aperture where the **Also sample along edges** check box is selected.*

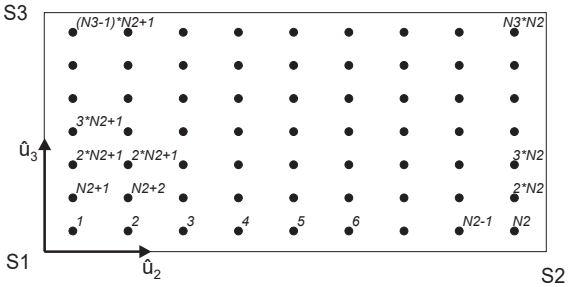


*Figure 842: Location of the equivalent dipoles on a planar aperture where the **Also sample along edges** check box is cleared.*

In Figure 843, dipole locations for a cylindrical aperture created from a data file containing field values for φ from 20° to 80° in 10° increments and 5 values in the z direction. When the **Also sample along edges** check box is selected, the samples extend up to the edges of the aperture. When the **Also sample along edges** check box is cleared, samples are not positioned on the edges.

> **Note:** With regards to selecting or clearing the **Also sample along edges** check box, with the option selected, the z positions of the samples will increase the height of the effective aperture, while in the φ direction the size of the effective aperture is increased by 5° on both sides.

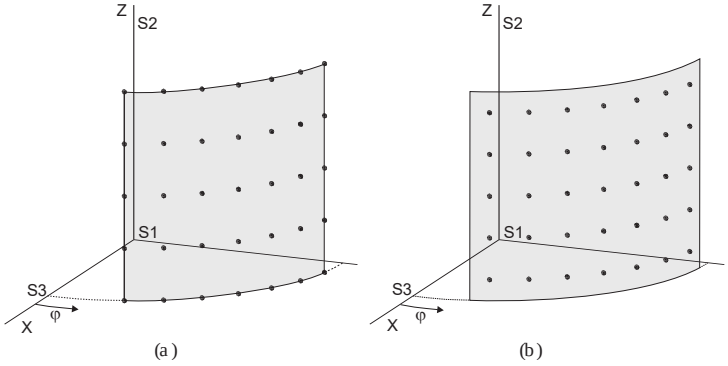


*Figure 843: Location of the equivalent dipoles on a cylindrical aperture: (a) **Also sample along edges** check box is selected (b) **Also sample along edges** check box is cleared.*

In Figure 844, the dipole locations for a spherical aperture created from field values for ϑ from 40° to 80° with 10° increments and φ from 20° to 80° also with 10° increments. In this case the aperture increases in size in both directions when selecting the **Also sample along edges** check box.

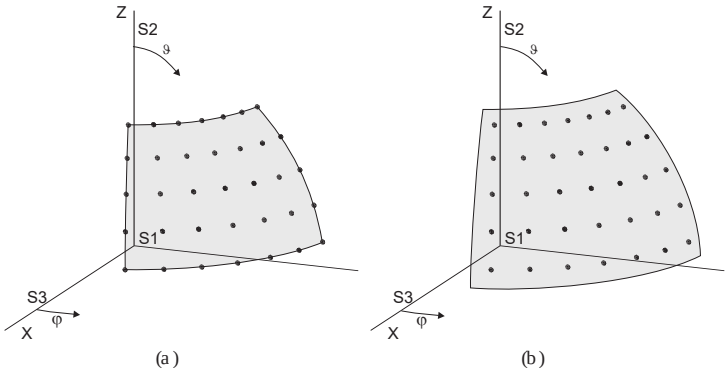


Figure 844: Location of the equivalent dipoles on a spherical aperture: (a) ***Also sample along edges*** *is checked (b)* ***Also sample along edges*** *is unchecked.*

For planar apertures, the data must vary first along the $\hat{u}_2$ direction. For cylindrical and spherical apertures PREFEKO will determine which coordinate is incremented first and write out the dipoles accordingly.

The dipole amplitude is the product of the surface current and the incremental area between samples. In addition, the amplitude of the dipoles on the sides (when checking **Also sample along edges**) are reduced by a factor of 2 and those on the corners by a factor of 4 to ensure that the effective aperture of integration has the same size as the specified aperture.

A fully closed surface can be created by specifying 6 planar apertures or a spherical one. The surface equivalence principle can be applied to this surface by reading both electric and magnetic fields for each plane. (For planar apertures the user should specify 6 AP cards, each using both electric and magnetic fields. If separate cards are used for the electric and magnetic fields the radiated power is not calculated correctly.) The normal vector must point towards the exterior region, which is normally outward. (For planar apertures created from `.efe` and `.hfe` files, the sample order determines the directions of $\hat{u}_2$ and $\hat{u}_3$ which, in turn, determines the normal vector $\hat{n}=\hat{u}_2\times\hat{u}_3$. If this is pointing into the cube, an additional 180° phase shift is obtained by setting **Phase of aperture (degrees)** to 180. This changes the sign of the field radiated by the aperture which, when interacting with the remaining sources, will result in the correct total fields in the desired region. All surfaces and scatterers inside the body must be removed but not those on the outside.

For planar apertures (for example, the opening of a horn antenna), one may use the mirror principle if the field at the edges can be neglected. This results in a duplication of the magnetic current and cancellation of the electric current. Therefore it is sufficient to read only the electric fields and scale by the factor **Amplitude scale factor**=2. In this case any sources or structures in the region towards which the normal is pointing should also be subjected to the mirroring (the structures should be electrically mirrored by using the SY card).

Note: The fields will only be correct in the direction that the normal vector points to. The symmetric fields in the other half-space will not be equal to the fields of the original problem.

Feko takes this application of the mirror principle into account and divides the total radiated power by two when calculating the power radiated by a planar aperture containing only electric or magnetic fields.

Related tasks
Adding a Near Field Source (CADFEKO)
Related reference
AR Card
DA Card
DP Card
FE Card
SY Card

Load Field Data From an ASCII Text File

When the data is read from an ASCII input file, the data must be given in a specific format.

Note: Each line in the ASCII file represents one point and the values are space delimited.

Planar Apertures

For planar apertures each line (point definition) must contain the following four parameters:

1. absolute value of the field component in the $\hat{u}_2$ direction
2. phase of the field component in the $\hat{u}_2$ direction
3. absolute value of the field component in the $\hat{u}_3$ direction
4. phase of the field component in the $\hat{u}_3$ direction

The data must be formatted in order for the position to first increment along the $\hat{u}_2$ direction.

Cylindrical Apertures

For cylindrical apertures each line (point definition) must contain the following five parameters:

1. angle φ in degrees
2. absolute value of the $\hat{\varphi}$ component
3. phase of the $\hat{\varphi}$ component
4. absolute value of the $\hat{z}$ component
5. phase of the $\hat{z}$ component.

Spherical Apertures

For spherical apertures it must contain the following six parameters:

1. ϑ angle
2. φ angle
3. absolute value of the $\hat{\vartheta}$ component

4. phase of the $\hat{\vartheta}$ component
5. absolute value of the $\hat{\varphi}$ component
6. phase of the $\hat{\varphi}$ component.

Related concepts
Example of AP Card Usage

Field Data Follows in the .PRE File

When the data is read from the `.pre` input file, the data must be formatted in a specific format.

The data must be formatted in one of the following formats:

- column based format
  - The four field components are the same as for when loading the field data from an ASCII file, and must be entered in columns of 10 characters wide from column 51 to column 90.

    The angle ϑ must be specified in column 30 to column 39 and ϕ must be specified in column 40 to column 49 (if applicable).
- colon separated format
  - The four field components are the same as for when loading the field data from an ASCII file, and must be entered in R_3 to R_6.

    The angle ϑ must be specified in R_1 and ϕ must be specified in R_2 (if applicable).

Note: If both electric and magnetic fields are required, all the electric fields are given first, followed by the same number of magnetic fields.

Example of a Spherical Aperture

An example of a spherical aperture where **Number of points along theta** = 3 and **Number of points along phi** = 3 (using colon separated format):

```
AP: 0 : 29 : S1 : S2 : S3 : 1 : 3 : 3 : 1 : 0   ** ApertureExcitation1
  :  :  :  :  :  : 0 : -90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 0 : 0 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 0 : 90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 90 : -90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 90 : 0 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 90 : 90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 180 : -90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 180 : 0 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 180 : 90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 0 : -90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 0 : 0 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 0 : 90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 90 : -90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 90 : 0 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 90 : 90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 180 : -90 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 180 : 0 : 1 : 0 : 1 : 0
  :  :  :  :  :  : 180 : 90 : 1 : 0 : 1 : 0
```

In the second row and onward, the field data is added to the `.pre`. Note that the electric fields are specified first, followed by the magnetic fields (therefore there are 9x2=18 field data lines).

Related concepts
Card Formats
Example of AP Card Usage

Example of AP Card Usage

An example of a typical AP usage is given.

Consider an open ended X-band waveguide radiating through a hole in a large ground plane as depicted in the figure. Away from the aperture the Z=0 plane is perfectly conducting (the tangential electric field is zero) while the magnetic field is not. Therefore electric symmetry is applied.

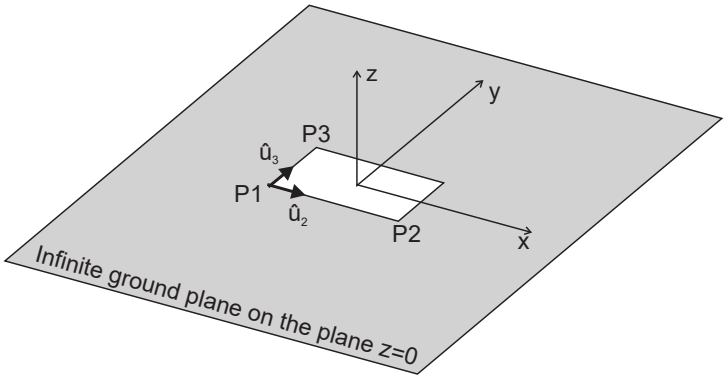


Figure 845: Example of an open waveguide as an implementation of the AP card.

For this example the field is considered to be purely y directed (it has only a $\hat{y}$, or $\hat{u}_3$, component). The field is assumed to be constant in the y direction and to have a cosine distribution in the x direction (the $\hat{u}_2$ axis).

With **Number of points along** $\hat{u}_2$ = 5 and **Number of points along** $\hat{u}_3$ = 3, the data file (`Guide.dat`) will be as follows:

```
0.0 0.0 0.0 0.0
0.0 0.0 0.707 0.0
0.0 0.0 1.0 0.0
0.0 0.0 0.707 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.707 0.0
0.0 0.0 1.0 0.0
0.0 0.0 0.707 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.707 0.0
0.0 0.0 1.0 0.0
0.0 0.0 0.707 0.0
0.0 0.0 0.0 0.0
```

The zero values will not result in any dipoles, but they must be in the data file to allow correct indexing. The `.pre` file will contain the following section:

```
...
** Only electric fields --- use electric symmetry in the z=0 plane
SY: 1 : 0 : 0 : 2

** Define the corner points of the aperture
#wx=0.02286
#wy = 0.01016
DP: P1 :  :  :  :  : -#wx/2 : -#wy/2 : 0
DP: P2 :  :  :  :  : #wx/2 : -#wy/2 : 0
DP: P3 :  :  :  :  : -#wx/2 : #wy/2 : 0

** The geometry ends after the corner nodes have been defined
EG: 1 : 0 : 0 :  :  :  :  :  :  :  :  : 1

** Specify the frequency
FR: 1 : 0 :  :  :  : 9.375e9

** Specify the AP card as a new source
** The amplitude factor of 2.0 is due to the use of the equivalence principle
AP: 0 : 3 : P1 : P2 : P3 : 1 : 5 : 3 : 2.0 : 0.0 :  :  :  : "Guide.dat"** nf_source
```

This will generate nine x directed magnetic dipoles of the correct magnitude in the `.fek` file.

An example of the AP card when including the field data directly in the `.pre` is as follows:

```
AP: 0 : 7 : P1 : P2 : P3 : : 5 : 3 : 2.0 : 0.0** nf_source
: : : : : : : : 0.0 : 0.0 : 0.0 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.707 : 0.0
: : : : : : : : 0.0 : 0.0 : 1.0 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.707 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.0 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.0 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.707 : 0.0
: : : : : : : : 0.0 : 0.0 : 1.0 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.707 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.0 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.0 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.707 : 0.0
: : : : : : : : 0.0 : 0.0 : 1.0 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.707 : 0.0
: : : : : : : : 0.0 : 0.0 : 0.0 : 0.0
```

AR Card

This card uses the radiation pattern of an antenna as an impressed source. The data is read from file or defined in the `.pre` file.

On the **Source/Load** tab, in the **Equivalent sources** group, click the **Radiation pattern (AR)** icon.

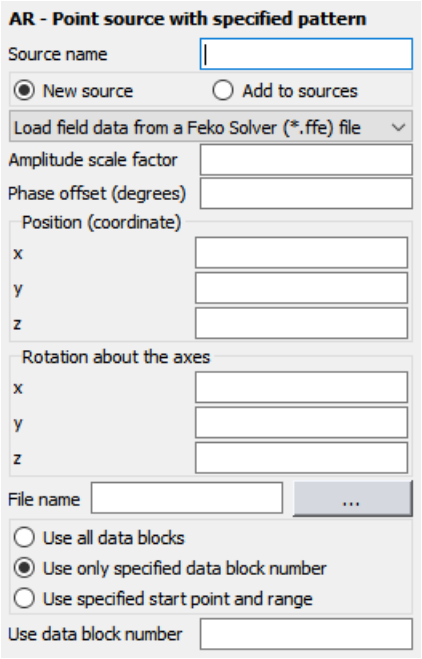


*Figure 846: The **AR - Point source with specified pattern** dialog.*

This card has a variety of uses, for example, importing measured radiation patterns, synthesising arrays from the individual patterns of the elements, realising radiation only within certain sectors, and so forth. In the MoM/UTD hybrid it is possible to simulate, for example, the antenna on its own and to save the far field in a `.ffe` file. This field is then imported and used as a source in the UTD part which will, compared to the full version of the antenna, greatly speed up the ray tracing computation as there is now only one source point.

Parameters:

New source — A new excitation is defined which replaces all previously defined excitations.

Add to sources — A new excitation is defined which is added to the previously defined excitations.

Load field data from

- **Feko Solver (*.ffe) file** — Read the radiation pattern from an `.ffe` format file, created with DA and FF cards.
- **CST far field scan (*.ffs)** — Read the radiation pattern from a CST far field scan.

external data file	Read the radiation pattern from an ASCII file (the format of this file is described below).
The file data follows in the (*.pre) input file	The radiation pattern is specified in the `.pre` file following the AR card (the format is described below). With this option FOR loops can be used to generate patterns from known functions.
Use last pattern defined at previous AR card	When using multiple AR cards (different radiating antennas in the same model) then it is quite common that the shape of the pattern is identical. If this is the case it is allowed to load the pattern just once and at subsequent AR cards to check this option. Then the last defined pattern will be used and memory can be saved (no need to store it again). Note that it is still possible to set the antenna position, orientation as well as amplitude and phase individually.

Amplitude scale factor The field strength values in each direction is determined from the data. This parameter is used to scale the amplitude of the entire pattern by a constant value.

Phase offset (degrees) This parameter specifies a constant additional phase for the entire pattern.

Position (coordinate) In this group the X, Y and Z coordinates of the source point (the position where the antenna is located) are entered in meters. This value is affected by the scale factor of the SF card if used.

Rotation about the axes In this group the angles with which the imported pattern is rotated around the x, y and z axes are entered in degrees. These fields are referred to as a_x, a_y and a_z in the rest of this discussion.

File name The name of the `.ffe`, `.ffs` or ASCII input file.

Use all data blocks Import all data blocks from the specified `.ffe` or `.ffs` file. The data is interpolated for use at the operating frequency.

Use only specified data block number Use the data from the nth frequency block in the specified `.ffe` or `.ffs` file.

Use specified start point and range Select a specific far field pattern in a `.ffe` or external file.

Start from point number	This parameter is only relevant when the data is read from a `.ffe` or an external file, and gives the line number of the first line to read from the input file. If the data must be read from the beginning of the file, the value in this field should be set equal to 1. This parameter is used when the `.ffe` file contains more than one pattern. For example, if the file contains the pattern at various frequencies, the correct pattern can be selected by setting this field to the appropriate value for each frequency. If the `.ffe` file is of a newer format and contains header data in addition to the data blocks, the point number refers to the actual point number excluding blank lines, comment lines and header lines.
Number of ϑ points	The number of ϑ angles in the pattern.
Number of φ points	The number of φ angles in the pattern.

The radiation pattern of the antenna must be specified in spherical coordinates (ϑ, φ) with the phase centre located at the origin of the local coordinates (as used in the pattern data). If this is not the case, the phase of the far fields will not be correct. (For example, if an `.ffe` file is exported with Feko to be used with the AR card, the origin should be shifted with the OF card to the phase centre of the antenna.) The vertical and horizontal components of the complex electric field E_{ϑ}^{FF} and/or E_{φ}^{FF} must be specified at discrete angles (ϑ_i, φ_j) with i and j larger or equal to 1 and smaller or equal to the number of points specified for the respective angles. The field values are entered in Volts and the actual far fields are calculated from

$$E_{\vartheta} = E_{\vartheta}^{FF} \cdot \frac{e^{-j\beta R}}{R} \tag{145}$$

or

$$E\varphi = E_{\varphi}^{FF} \cdot \frac{e^{-j\beta R}}{R} \tag{146}$$

with R the distance to the field point and β the complex propagation constant in the free space medium (see the EG and GF cards). These formulas are used for all distances R (also in the near field). However, Feko tests whether the far field conditions are met (by calculating the directivity and equivalent aperture) and gives an appropriate warning if this is not the case.

The permissible range of the angles ϑ_i is 0°...180° and they must be in ascending order, that is $\vartheta_{i+1} > \vartheta_i$. However, the angles do not have to be equidistantly spaced. (Therefore in the case of a highly directive antenna a denser grid can be used close to the main beam direction.) The same applies to the angles ϑ_j where the permissible range is 0°...360°. For field angles outside the start and end values defined in the data (for $\vartheta < \vartheta_1$, $\vartheta > \vartheta_{max}$, $\varphi < \varphi_1$ or $\varphi > \varphi_{max}$), the field strengths E_{ϑ}^{FF} and E_{φ}^{FF} are set to zero, so that a sector radiator can be realised. The values at field angles within the defined range are determined by bilinear interpolation. To realise a complete radiation pattern, rather than a sector radiator, the angles should be defined so that $\vartheta_1 = 0°$, $\vartheta_{max} = 180°$, $\varphi_1 = 0°$ and $\varphi_{max} = 360°$.

The radiation pattern, specified in the local spherical coordinate system (ϑ, φ) of the antenna, is read and initially placed at the origin of the global coordinate system in which the `.pre` file is constructed. The pattern is now rotated by an angle a_z around the Z axis, by a_y around the Y axis and by a_x around

the X axis. (The rotation is identical to the rotation executed by the TG card and the rotation matrix M is applicable to both the TG and AR cards.) Finally the pattern is shifted to the specified location.

If the AR card is used simultaneously with a ground plane (BO card), Feko includes the influence of the ground plane on the radiation pattern. The imported pattern must therefore be the free space radiation pattern of the antenna (in the absence of the ground plane). If this is not the case the influence of the ground plane is considered twice.

The use of the PW card to specify the radiated power is allowed. The field amplitudes $|E_{\varphi}^{FF}|$ and $|E_{\varphi}^{FF}|$ will be scaled accordingly. Multiple radiation patterns can be used simultaneously, and also with other sources such as an incident plane wave. In such a case, the coupling is not considered when the radiated power is determined.

The AR card cannot be used with special Green's functions for a layered sphere or for a layered substrate.

The format of the data depends on the type of file:

- an `.ffe` file

  With this option, the radiation pattern is read from an `.ffe` file created with Feko (using the DA and FF cards). All the data of the radiation pattern (angles and field values) are determined from the file. The user should ensure that the frequency is correct. If an antenna is analysed with Feko, the far field can be exported to the `.ffe` file using the commands (for 5° angle increments)

  ```
  DA: 0 : 0 : 1 : 0 : 0 :  :  :  : 0
  FF: 1 : 37 : 73 : 0 : 0 : 0 : 0 : 5 : 5 :  :  : 0
  ```

  Note 37 points are used for ϑ and 73 points for φ to ensure that the radiation pattern is closed (see also the comment above).

  This can then be imported as a source into another model with the command

  ```
  AR: 0 : 1 : 1 : 37 : 73 : 1.0 : 0.0 : 0.0 : 0.0 : 0.0 : 0.0 : 0.0 : "file.ffe"
  ```

- an external ASCII file

  With this option, the data is read from the specified external data file. Each line contains 6 space delimited data fields in the following order:

  1. The angle ϑ in degrees
  2. The angle φ in degrees
  3. Amplitude of the field strength E_{ϑ}^{FF} in V
  4. Phase of the field E_{ϑ}^{FF} in degrees
  5. Amplitude of the field strength E_{φ}^{FF} in V
  6. Phase of the field E_{φ}^{FF} in degrees

  The inner loop should be with respect to the angle ϑ so that the order of the lines is as follows (where N is the number of ϑ points and M is the number of φ points):

$$\begin{matrix} \vartheta_1 & \varphi_1 & \cdots \\ \vartheta_2 & \varphi_1 & \cdots \\ \vartheta_3 & \varphi_1 & \cdots \\ \vdots & \vdots & \cdots \\ \vartheta_N & \varphi_1 & \cdots \\ \vartheta_I & \varphi_2 & \cdots \\ \vartheta_2 & \varphi_2 & \cdots \\ \vdots & \vdots & \cdots \\ \vartheta_N & \varphi_M & \cdots \end{matrix} \quad (147)$$

- after this line in the `.pre` file

  This case is similar to reading an external ASCII file, except that the data is read directly from the `.pre` input file. The six data fields mentioned for the case of an ASCII file must appear in the 6 columns of 10 characters, starting at character 31 and ending at character 90 in the lines following the AR card. When this option is selected the card dialog shows additional input fields where the user can specify these values for each point. The data lines may be separated by comment lines (EDITFEKO, however, does not support this) and FOR–NEXT loops may be used. Even when using FOR loops the card dialog in EDITFEKO can be used to generate a typical line.

An ideal sector radiator:

The ideal sector radiator radiates 10 Watt of power in the horizontal polarisation in the angular region defined by $-70° \leq \varphi \leq 70°$ and $75° \leq \vartheta \leq 105°$. Since the angle range of the imported pattern must be positive, separate sources for the regions $0° \leq \varphi \leq 70°$ and $290° \leq \varphi \leq 360°$ should be defined. A more elegant solution is to define a single pattern in the range $0° \leq \varphi \leq 140°$ and rotate it by -70° around the Z axis. The complete radiation pattern is defined in the following input file (note that only horizontal polarisation, E_{φ}^{FF}, is required).

```
** Application example for the AR card: Sector radiator
** No other structures considered
EG: 0 : 0 : 0 : : : : : : : : : 1
** Set the frequency
FR: 1 : 0 : : : : 100e6
** Specified radiated power
PW: 1 : : : : : 10
** Define the sector radiator
AR: 0 : 3 :  : 2 : 2 : 1.0 : 0.0 : 0.0 : 0.0 : 0.0 : 0.0 : 0.0 : -70
  :  :  :  :  :  : 75 : 0 : 0 : 0 : 1 : 0
  :  :  :  :  :  : 105 : 0 : 0 : 0 : 1 : 0
  :  :  :  :  :  : 75 : 140 : 0 : 0 : 1 : 0
  :  :  :  :  :  : 105 : 140 : 0 : 0 : 1 : 0
** Check: Compute the full 3D radiation pattern with 5 deg stepping
FF: 1 : 37 : 73 : 0 : 0 : 0.0 : 5.0 : 5.0 : : : : 0
** End
EN
```

Feko determines a directivity of 10.1 dBi. The radiation pattern is easily validated by calculating the far field as shown with the FF card in the last step. The 3D far field is depicted in the figure below.

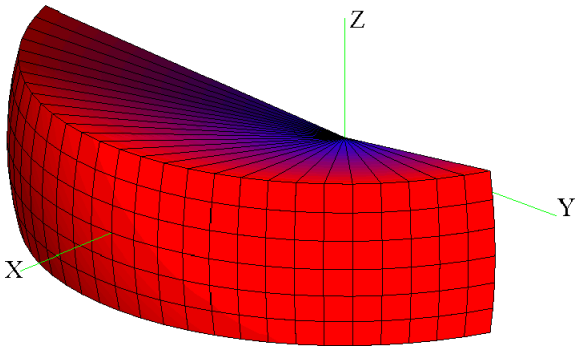


Figure 847: 3D radiation pattern of the sector radiator.

Related tasks

Adding a Far Field Point Source (CADFEKO)

Related reference

DA Card

FF Card

EG Card

GF Card

TG Card

AS Card

The AS card defines an excitation by means of impressed spherical modes which are either radiating (propagating in positive r direction to infinity, with r being the radius in a spherical coordinate system) or incident onto a structure (propagating towards the origin r = 0).

On the **Source/Load** tab, in the **Equivalent sources** group, click the **Spherical modes (AS)** icon.

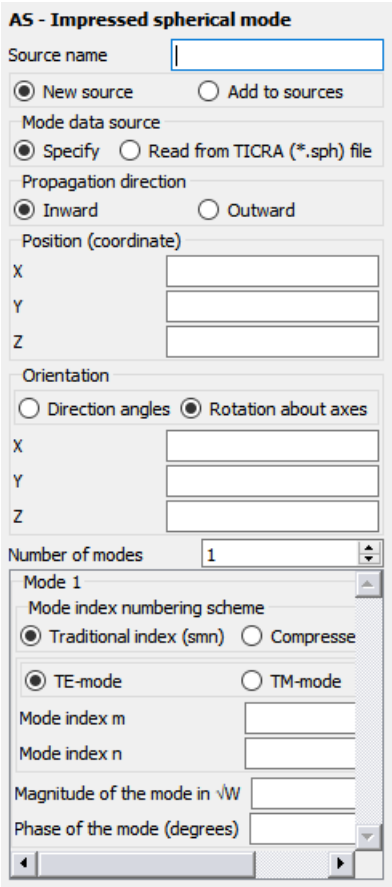


*Figure 848: The **AS card - Impressed spherical mode** dialog.*

This excitation option can thus be used for both the synthesis of an arbitrary electromagnetic field (sum of the modes weighted with complex mode coefficients), and also for the determination of the response (induced voltage or power at a load) of a receiving antenna due to the incident modes (leading to the so-called generalised scattering matrix).

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.

Propagation direction

Inward	The spherical waves propagate inward. The model is illuminated with modes propagating towards r = 0 (spherical Hankel function of the first kind $z_n^{(3)} = h_n^{(1)}$)
Outward	The spherical wave propagate outward. The modes radiate towards r = ∞ (spherical Hankel function of the second kind $z_n^{(4)} = h_n^{(2)}$).

This option is only available for when the modes are entered in the `.pre` file and not when the modes are imported from a TICRA file (`.sph`) in which case the outward propagating direction is used.

Mode data source

The spherical modes can be entered directly in the `.pre` file or it can be imported from a TICRA file (`.sph`) file.

> **Note:** Multiple spherical modes can be created as a single source.

When importing from a TICRA file, the imported spherical modes may be scaled and the phase given an offset by entering values for the **Magnitude scale factor** and **Offset phase (deg) by** fields.

Position (coordinate)

The coordinates of the origin r = 0 of the mode in m. These values are optionally scaled by the SF card.

ϑ angle

The ϑ angle between the spherical mode axis (N) and the Z axis in degrees.

φ angle

The φ angle between the projection of the spherical mode axis (N) onto the plane Z=0 and the X axis in degrees.

Rotation about axes

The rotation of the spherical mode source about the X axis, Y axis and Z axis.

Number of modes

The number of modes that are entered in the `.pre` file must be specified.

Traditional index (smn)

If this option is checked, you can specify TE-mode (s = 1) or TM-mode (s = 2) and the indices *m* and *n* in the group below. Here *n* is the mode index in radial direction and must be in the range 1,2,...∞ and *m* is the mode index in the azimuth direction φ. We do not distinguish between even and odd modes (with cos(mφ) and sin(mφ) angular dependencies), but rather use the angular

dependency $e^{jm\varphi}$. Thus the index *m* can also be negative, but it must be in the range $-n...n$.

Compressed index (j)

With this option, a compressed one-dimensional mode numbering scheme is used. The **Mode index j** is then specified in the field below. Here

$$j = 2\left[n(n+1)+m-1\right]+s \qquad (148)$$

where s = 1 for TE-modes and s = 2 for TM-modes. This unified mode numbering scheme allows the computation of an extended scattering matrix (with network and radiation ports). This index *j* then represents a unique port number in the scattering matrix.

Magnitude of the mode in $\sqrt{W}$

Absolute value of the complex amplitude of this specific spherical mode (due to the applied normalisation of the spherical modes, the unit of this amplitude is $\sqrt{W} = \sqrt{VA}$).

Phase of the mode (degrees)

The phase of the complex amplitude of this spherical mode in degrees.

Use all data blocks

Import all data blocks from the specified TICRA (`.sph`) file. The data is interpolated for use at the operating frequency.

Use only specified data block number

Use the data from the nth frequency block in the TICRA (`.sph`) file.

The implementation of the spherical modes at the AS card follows closely the references:

J. E. Hansen, *Spherical Near-field Antenna Measurements*, Peter Peregrinus Ltd., London, 1988 and B. C. Brock, *Using Vector Spherical Harmonics to Compute Antenna Mutual Impedance from Measured or Computed Fields*, Sandia National Laboratories, Report SAND2000-2217-Revised, April 2001. One must realise that Hansen assumes a complex time dependence of $e^{-i\omega t}$, while Feko always uses the positive sign $e^{j\omega t}$.

In Feko, using the modal coefficients Q^c_{smn} the electric and magnetic field strength is represented in a spherical coordinate system by

$$\mathbf{E}(r,\varphi,\vartheta) = \sum_{c=3}^{4}\sum_{s=1}^{2}\sum_{n=1}^{\infty}\sum_{m=-n}^{n} Q^{(c)}_{smn}\mathbf{F}^{(c)}_{smn}(r,\varphi,\vartheta) \qquad (149)$$

$$\mathbf{H}(r,\varphi,\vartheta) = \frac{j}{Z_F}\sum_{c=3}^{4}\sum_{s=1}^{2}\sum_{n=1}^{\infty}\sum_{m=-n}^{n} Q^{(c)}_{smn}\mathbf{F}^{(c)}_{3-s,m,n}(r,\varphi,\vartheta). \qquad (150)$$

Here *s*, *m* and *n* are the mode indices with *s* = 1 indicating the TE-mode and *s* = 2 the TM-mode, and *c* represents the propagation direction: *c* = 3 is inward and c = 4 is outward. The term Z_F denotes the wave impedance of the medium under consideration, β below is the corresponding wavenumber.

The spherical wave functions F^c_{smn} are given by

$$\begin{aligned}\mathbf{F}_{1mn}^{(c)}(r,\vartheta,\varphi) &= \mathbf{M}_{mn}^{(c)} \\ &= \beta\sqrt{\frac{Z_F}{2n}}\frac{1}{\sqrt{n(n+1)}}\left(-\frac{m}{|m|}\right)^m \\ &\quad \cdot(0\cdot\mathbf{e}_r \\ &\quad + z_n^{(c)}(\beta r)\frac{jm}{\sin\vartheta}\widehat{\mathbf{P}}_n^{|m|}(\cos\vartheta)e^{jm\varphi}\mathbf{e}_\vartheta \\ &\quad - z_n^{(c)}(\beta r)\frac{\partial}{\partial\vartheta}\left\{\widehat{\mathbf{P}}_n^{|m|}(\cos\vartheta)\right\}e^{jm\varphi}\mathbf{e}_\vartheta\Big)\end{aligned} \tag{151}$$

and

$$\begin{aligned}\mathbf{F}_{2mn}^{(c)}(r,\vartheta,\varphi) &= \mathbf{N}_{mn}^{(c)} \\ &= \beta\sqrt{\frac{Z_F}{2n}}\frac{1}{\sqrt{n(n+1)}}\left(-\frac{m}{|m|}\right)^m \\ &\quad \cdot\Big(\frac{n(n+1)}{\beta r}z_n^{(c)}(\beta r)\widehat{\mathbf{P}}_n^{|m|}(\cos\vartheta)e^{jm\varphi}\mathbf{e}_r \\ &\quad + \frac{1}{\beta r}\frac{\partial}{\partial(\beta r)}\left\{\beta r z_n^{(c)}(\beta r)\right\}\frac{\partial}{\partial\vartheta}\left\{\widehat{\mathbf{P}}_n^{|m|}(\cos\vartheta)\right\}e^{jm\varphi}\mathbf{e}_\vartheta \\ &\quad + \frac{1}{\beta r}\frac{\partial}{\partial(\beta r)}\left\{\beta r z_n^{(c)}(\beta r)\right\}\frac{jm}{\sin\vartheta}\widehat{\mathbf{P}}_n^{|m|}(\cos\vartheta)e^{jm\varphi}\mathbf{e}_\varphi\Big)\end{aligned} \tag{152}$$

with the associated Legendre function

$$\widehat{\mathbf{P}}_n^{|m|}(\cos\vartheta) = \sqrt{\frac{2n+1}{2}\frac{(n-|m|)!}{(n+|m|)!}}\,P_n^{|m|}(\cos\vartheta) \tag{153}$$

and the spherical Bessel functions

$$\begin{aligned}z_n^{(3)}(\beta r) &= h_n^{(1)}(\beta r) = j_n(\beta r) + j y_n(\beta r) \\ z_n^{(4)}(\beta r) &= h_n^{(2)}(\beta r) = j_n(\beta r) - j y_n(\beta r)\end{aligned}\ . \tag{154}$$

It should be noted that the Legendre polynomial $P_n^{|m|}(\cos\vartheta)$ as used in Feko follows the definitions of Abramowitz / Stegun (also used like this in Numerical Recipes) or also Harrington. The formulas used in other references (for example, Stratton or Hansen) have an extra factor $(-1)^m$ included. This is not considered in Feko, and thus the mode coefficients Q_{smn}^c might differ from those computed according to Hansen (there is also of course the other time dependency).

Theoretically the index n runs in the range 1,2,...,∞. For any practical application, one will have to consider a finite number of modes only, limit the range $n = 1...N$. A few rules of thumb exist for the selection of N. For instance when representing the pattern of an antenna by spherical modes one can use the upper limit

$$N \approx \beta r_0 = 2n\frac{r_0}{\lambda}, \tag{155}$$

where β is the wavenumber, λ the wavelength, and r_0 denotes the radius of the smallest sphere enclosing the antenna. In critical cases, one might also rather use

$$N \approx \beta r_0 + 10 \tag{156}$$

or

$$N \approx \beta r_0 + 3\sqrt[3]{\beta r_0} \tag{157}$$

When using the compressed numbering scheme with one index j, any upper limit N for n will with the largest values $m = N$ and $s = 2$ translate into an upper limit

$$J = 2\left[N(N+1)+N-1\right]+2 = 2N(N+2) \tag{158}$$

for j ($j = 1...J$ then). So for instance for an antenna with enclosing radius $r_0 = \frac{\lambda}{4}$ (then $\beta r_0 = 1.57 = 1.57$) when using the last of the three rules of thumb above, one would need roughly $N \approx 5$ or $J \approx 70$ modes, respectively. For $r_0 = \lambda$ these limits become already $N \approx 12$ and $J \approx 336$, and for $r_0 = 5\lambda$ one has to use $N \approx 41$ and $J \approx 3526$ modes The modes have been normalised such that each mode has a constant power flow through any spherical surface (either inwards or outwards). In principle one could use the PW card for this, but then power normalisation works only if there is not more than one mode active at the same time (when using the PW card, just the total radiated power of all the modes is determined, and then each mode is scaled with the same factor, so that the total radiated power is correct, but here we enforce a specified power for each individual mode). The power for each mode is independent of the mode indices $P = 0.5\left|\text{Magnitude of the model}\right|^2$

(unit is correctly Watt since the amplitude has a unit $\sqrt{W}$). Since the modes are orthogonal, if multiple AS cards are active at the same time, the powers of the individual modes can just be added. Any other power corrections (such as due to metallic elements being in the vicinity) are not taken into account in Feko.

If an AS excitation is used in connection with multiple different media, it should be noted that we assume outward propagating modes (when **Outward** is selected) to originate from the source position. The source is located in the medium where its position is specified, and its contribution will be zero in all other media.

For inward propagating modes (when **Inward** is selected) we assume the propagating modes to originate at infinity, in the free space medium 0, and such modes only contribute to this medium with index 0. In connection with the UTD, only outward propagating modes are allowed (they have a well defined source point), while inward propagating modes are not supported (neither a source point nor an incidence direction can be assigned to such modes).

When computing the far field with the FF card, then outward propagating modes are included normally, which is important when synthesising antenna patterns by means of spherical modes. However, for inward propagating modes the far field limit for $R \to \infty$ of the field strength with

$$\frac{e^{-j\beta R}}{R} \tag{159}$$

split off does not exist (similar to the non-existent far field for an incident plane wave). Thus such inward propagating modes are excluded from any far field computation. This is not a problem, since normally inward propagating modes are applied when computing the generalised antenna scattering matrix (the response of a receiving antenna to an incident mode). Then one looks at these quantities:

- Induced voltage or power at the antenna terminals for the network ports (no far field computation).
- Field scattered back, and decomposition of this field into spherical modes (far field ports). Here one needs the far field computation, but similar to an RCS computation with an incident plane wave, only the scattered far field is of interest, which can be obtained from the FF card without problems.

Note: All the modes (inwards and outwards propagating) are correctly included when doing a near field computation with the FE card (also for very large distances).

As an application example, we consider the TE-mode $n = 5$ and $m = 0$ and compute the far field pattern:

```
** Create the far-field radiation pattern of a spherical mode

** No geometry
EG: 1 : 0 : 0 :  :  :  :  :  :  :  :  : 1

** Set the frequency
FR: 1 : 0 :  :  :  : 100e6

** Spherical mode indices
#s=1  ** 1 = TE-mode, 2 = TM-mode
#n=5  ** mode order n (with n=1, 2, 3, ...)
#m = 0      ** mode order m (with m=-n ... n)

**   Excitation
AS: 0 : 4 : #s : #m : #n : 1 : 0

**   Compute the full far-field pattern
FF: 1 : 91 : 37 : 0 : 0 : 0 : 0 : 2 : 10 :  :  : 0

** End
EN
```

The resulting pattern is shown in Figure 849. From the Feko output file one can see the correct radiated power of 0.5 Watt as obtained from the far field integration:

```
   Integration of the normal component of the Poynting vector in the angular
   grid DTHETA =    2.00 deg. and DPHI =   10.00 deg. (    3367 sample points)
     angular range THETA        angular range PHI       radiated power
    -1.00 .. 181.00 deg.      -5.00 .. 365.00 deg.     5.13889E-01 Watt
     0.00 .. 180.00 deg.       0.00 .. 360.00 deg.     5.00001E-01 Watt
```

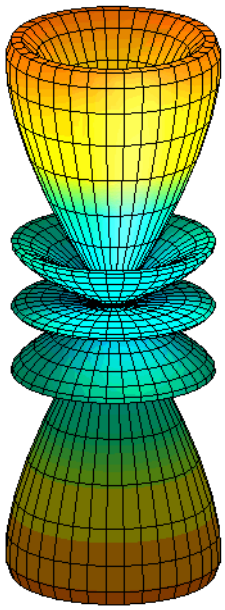


Figure 849: The 3D radiation pattern of a spherical TE-mode with n=5 and m=0.

Related tasks

Adding a Spherical Modes Source (CADFEKO)

Related reference

FE Card

FF Card

AT Card

This card defines a voltage source that is applied to a voxel mesh in connection with a finite difference time domain (FDTD) method.

On the **Source/Load** tab, in the **Sources on geometry** group, click the **Voltage source** icon.

From the drop-down list, click the **Geometry source (AT)** icon.

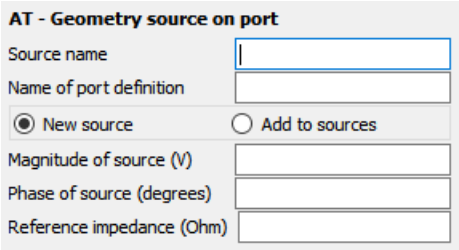


*Figure 850: The **AT - Geometry source on port** dialog.*

Parameters:

Source name — The name of the excitation to be defined.

Name of port definition — The name of the port.

New source — A new excitation is defined which replaces all previously defined excitations.

Add to sources — A new excitation is defined which is added to the previously defined excitations.

Magnitude of source (V) — Magnitude of the voltage source in V.

Phase of source (degrees) — Phase of the voltage source in degrees.

Reference impedance (Ohm) — The reference impedance of the excitation is used for S-parameter calculations and is the reference impedance used for realised gain calculations. It is also the default reference impedance used to calculate and display the reflection coefficient in POSTFEKO. If this field is empty or 0 in an S-parameter calculation, the value specified at the SP card is used. For realised gain and reflection coefficient calculations, 50 Ohm will be assumed when the field is empty or 0.

AV Card

This card defines an impressed current source similar to the AI card but that makes electrical contact with a conducting surface.

On the **Source/Load** tab, in the **Ideal sources** group, click the **Impressed current** icon. From the drop-down list, click the **Impressed current connected to mesh vertex (AV)** icon.

The current varies linearly between the value at the start point and that at the end point. At the connection point special singular functions are used for the surface current density on the triangles to allow continuous current flow.

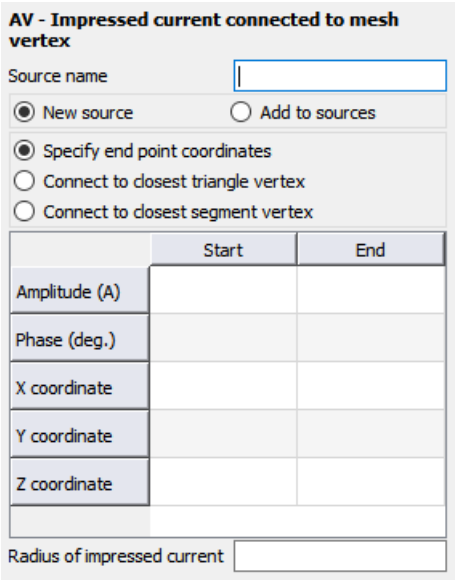


*Figure 851: The **AV - Impressed current connected to a triangle** dialog.*

Parameters:

New source	A new excitation is defined which replaces all previously defined excitations.
Add to sources	A new excitation is defined which is added to the previously defined excitations.
Specify end point coordinates	The coordinates of the end point $\mathbf{r}_2$ are specified with the X, Y, Z coordinate fields. This point must coincide with a corner point of one or more triangles.
Connect to closest triangle vertex	The coordinates of the end point $\mathbf{r}_2$ are not known. In this case the X, Y, Z coordinate fields of the end point are not used. Feko searches through all the metallic triangles for the corner point that is closest to the start point $\mathbf{r}_1$ of the current element. This is then the end point $\mathbf{r}_2$.

Connect to closest segment vertex	The coordinates of the end point $\mathbf{r}_2$ are not known. In this case the X, Y, Z coordinate fields of the end point are not used. Feko searches through all the metallic segments for the vertex that is closest to the start point $\mathbf{r}_1$ of the current element. This is then the end point $\mathbf{r}_2$.
Amplitude (A)	Current amplitude (in A) at the start, $\mathbf{r}_1$, and end, $\mathbf{r}_2$, points.
Phase (deg.)	Phase of the current at the start point in degrees.
X, Y, Z coordinate	Coordinates of the start and end points in m. (Note that all the coordinate values are optionally scaled by the SF card.)
Radius of impressed current	This parameter is optional. If specified, and different from zero, this value gives a finite wire radius for the impressed current element. Feko then assumes that the current is uniformly distributed on the wire surface and uses the exact wire integral. If the parameter is not specified, the current filament approximation is used. This value is optionally scaled by the SF card.

The following restrictions apply when using the impressed current elements making electrical contact with conducting surface:

- All the restrictions given in the discussion of the AI card also apply in this case.
- The start point of the impressed current segment may be connected with AI cards or further AV cards. If there is a current discontinuity at this point, the resulting point charge is not considered (see the discussion given with the AI card). Line charges along the current path and surface charges on the triangles are correctly taken into account. At the connection point $\mathbf{r}_2$ a continuous current model is used so that a point charge is not possible here.

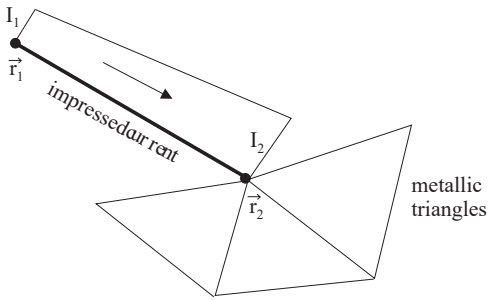


Figure 852: Impressed line current with a linear current distribution and electrical contact to conducting triangles.

Related reference

AI Card

SF Card

AW Card

The AW card is a two line card which is used to define a waveguide port excitation. With this card a waveguide port excitation by an impressed mode on a rectangular, circular, or coaxial waveguide, can be modelled or the impressed travelling modes in all waveguides of a multi-port network can be imported from a `.fim` file.

On the **Source/Load** tab, in the **Sources on geometry** group, click the  **Waveguide source (AW)** icon.

Related tasks

Creating a Waveguide Port (CADFEKO)

Creating a Waveguide Source(CADFEKO)

Specify the Source in the PRE File

With this option a waveguide port excitation by means of an impressed mode on a rectangular, circular, or coaxial waveguide, can be modelled.

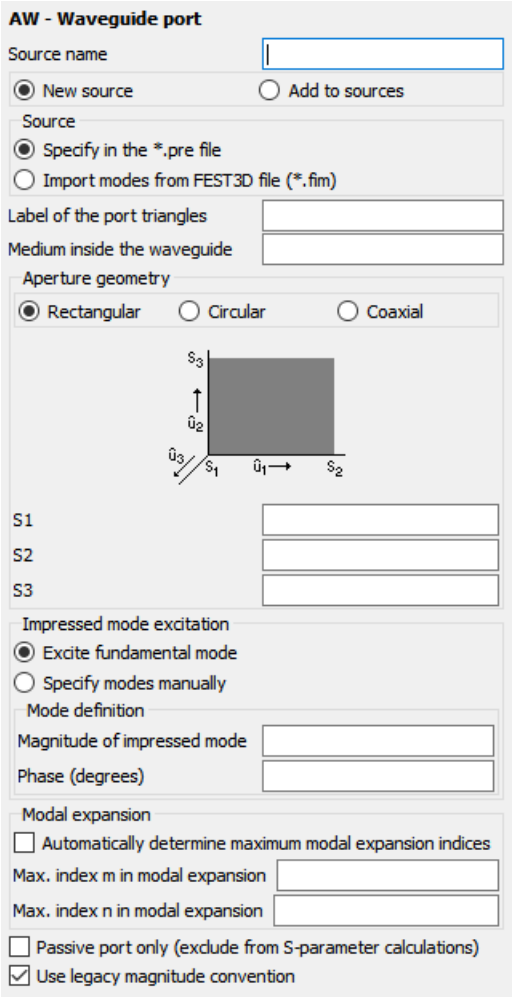


*Figure 853: The **AW - Waveguide port** dialog.*

Parameters:

New source
A new excitation is defined which replaces all previously defined excitations.

Add to sources
A new excitation is defined which is added to the previously defined excitations.

Label of the port triangles
Label of the triangular mesh elements in the mesh which represent the waveguide port. (If multiple solutions are defined in the same `.pre` file, then the usage of the waveguide ports with respect to the label(s) to which it/they are applied must be consistent for all solutions.)

Medium inside the triangles
The label of the medium inside the modelled waveguide.

Rectangular

A rectangular waveguide cross section is used, which is defined by three points S1, S2, and S3 as follows: S1 is an arbitrary corner point, and S2 and S3 are two corner points which define the waveguide sides $\mathbf{u}_1$ (from S1 to S2) and $\mathbf{u}_2$ (from S1 to S3). The direction in which the mode is launched is given by $\mathbf{u}_3 = \mathbf{u}_1 \times \mathbf{u}_2$.

Circular

A circular waveguide cross section is used. The point S1 denotes the centre of the circular port, and the point S2 specifies the radius and start point for the angular dependency. A further point S3 must be perpendicular above the centre of the circular plate, so that the direction from S1 to S3 indicates the direction in which the waveguide modes are launched.

Coaxial

Here a feed of a coaxial waveguide with circular cross sections of both the inner and outer conductor can be specified. The point definitions are the same as for the circular waveguide, except that an additional point S4 must be defined between S1 and S2 which specifies the radius of the inner conductor.

Excite fundamental mode

Select this option to automatically excite the fundamental mode of the waveguide. When this option is selected, the mode type and its indices (*m* and *n*) cannot be specified since they are determined automatically.

TE-mode

If this option is checked, a $TE_{m,n}$ mode (also referred to as $H_{m,n}$) is used as excitation. This option is only available when **Excite fundamental mode** has not been selected.

TM-mode

If this option is checked, a $\mathrm{TM}_{m,n}$ mode (also referred to as $E_{m,n}$) is used as excitation. This option is only available when **Excite fundamental mode** has not been selected.

TEM-mode

If this option is checked (only available for the coaxial waveguide since TEM modes don't exist in rectangular/circular waveguides), a TEM mode is used as excitation. This option is only available when **Excite fundamental mode** has not been selected.

Mode index m

The index *m* of the $TE_{m,n}$ or $\mathrm{TM}_{m,n}$ mode which is impressed at the port. Note that for a rectangular waveguide the index m is related to the $\mathbf{u}_1$ direction (for example from point S1 to S2). For a circular/coaxial waveguide, *m* denotes the angular dependency. This option is only available when **Excite fundamental mode** has not been selected.

Mode index n

The index *n* of the $TE_{m,n}$ or $\mathrm{TM}_{m,n}$ mode which is impressed at the port. Note that for a rectangular waveguide the index *n* is related to the $\mathbf{u}_2$ direction (for example, from point S1 to S3). For a circular/coaxial waveguide, *n* denotes the radial dependency.

	This option is only available when **Excite fundamental mode** has not been selected.
Magnitude of impressed mode	Absolute value of the complex amplitude of the impressed mode. For a TE-mode the unit is $\frac{A}{m}$, for a TM-mode the unit is $\frac{V}{m}$, for a TEM-mode the unit is V. Note that an amplitude of zero can also be specified. In this case a waveguide port is acting purely as a passive port (for example, as waveguide termination), and no wave is launched.
Phase (degrees)	The phase of the impressed mode in degrees.
Rotation angle φ_0 (degrees)	This option is available for circular and coaxial modes only and indicates the rotation angle in degrees by which a mode is rotated anti-clockwise with respect to the reference direction (point S2).
Modal expansion	At a waveguide port a specific mode is used as impressed excitation. However, due to discontinuities in the model, also higher order modes can result and will be propagating backwards through the port (applies to both active and passive ports). The maximum modal expansion indices taken into account during the calculation can be determined automatically by the kernel or specified manually. The included modes must be sufficient to capture the resulting field distribution of the problem. Note that the mesh across the waveguide port must be fine enough to represent the potential rapid field variation of included higher order modes. Also note that an increased number of higher order modes included in the model may have a significant impact on the run-time. If specified manually then the input values denote the maximum mode indices m and n which will be used to expand the backwards travelling waves. If determined automatically, then all propagating modes will be included, as well as evanescent modes that decay faster than $1/e$ at a tenth of a wavelength away from the waveguide port.
Passive port only (exclude from S-parameter calculations)	The waveguide port can be marked as passive only so that it will not be considered during S-parameter calculations. In this case the port is acting purely as a passive waveguide termination, and the coupling to and from this port will not be calculated.
Use legacy convention	Select the **Use legacy magnitude convention** check box to use the legacy definition of mode-dependent units (for example, for TE-mode it is A/m; for TM-mode it is V/m). The default is to use the power-based definition of magnitude. This definition is more intuitive and is common to all mode types.

Import Modes from FEST3D File

With this option,impressed forward travelling modes in all waveguides of a multi-port network can be imported from a `.fim` file.

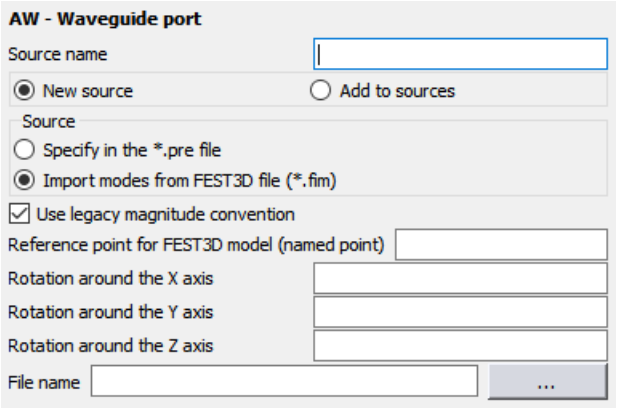


*Figure 854: The **AW - Waveguide port** dialog for importing from a FEST3D file.*

Parameters:

Reference point for FEST3D model (named point)	A named point indicating the translation of the imported model in the Feko coordinate system. Note that this point must have been previously defined with a DP card.
Rotation around the X axis	This specifies the rotation of the imported model around the X axis in degrees.
Rotation around the Y axis	This specifies the rotation of the imported model around the Y axis in degrees.
Rotation around the Z axis	This specifies the rotation of the imported model around the Z axis in degrees.
File name	The name of the `.fim` file.

In order to model a waveguide port excitation by an impressed mode, the cross section of the waveguide at the port location must be meshed into metallic triangles with a unique label. The propagation direction is given by the unit vector $\mathbf{u}_3$, see the small graphics in the AW card panel above.

In general, specific meshing rules exist in Feko relating the triangular patch size to the wavelength. When meshing the cross section of a waveguide to define a waveguide port, the mesh size must be small enough to capture the field distribution of the highest mode (m, n) which is included in the expansion. Feko checks this automatically and gives a warning for coarse meshes or an error if the mesh size is too large. One must then either refine the mesh just at the port or reduce the maximum mode indices used in the expansion.

The following restrictions apply when using a waveguide port excitation:

- Waveguide ports are available for models containing metallic objects (wires and surfaces and wire/surface junctions, including PO) and dielectrics (solved using the SEP, FEM) or dielectric coatings and thin dielectric sheets.

- Special Green's functions may not be used in conjunction with waveguide port feeds.
- When using waveguide ports, then UTD is not allowed in the same model. Faceted UTD supports waveguide ports if uncoupled to MoM. The same as for RL-GO. Note, however, that in Feko it is possible (using the AR or AP cards) to decompose a model (say a horn antenna in front of a reflector) into different sub-problems. See Example_35 in the Scripting Examples guide for an illustration of this decomposition technique.

The reflection coefficient at each waveguide port (S_{11}) is always calculated and available for display in POSTFEKO on an S-parameter graph, even when no S-parameter calculation has been requested. Requesting S-parameters with the SP card is supported for waveguide ports. Multiple ports (active and/or passive) can be present in the model. S-parameters are directly based on the waveguide impedance of the specific mode under consideration. The reference impedance as specified at the SP card is not used for waveguide ports.

Examples for the application of waveguide feeds can be found in the Examples Guide (E-2) and in the Script Examples, example_08 and example_34.

In order to rule out any possible doubts and ambiguities regarding the waveguide mode definitions, we give here the explicit expressions of the modes as used in Feko. This implementation follows closely the conventions in S. Ramo, J. R. Whinnery, and T. van Duzer, Fields and Waves in Communication Electronics, John Wiley & Sons, Inc., 3rd ed., 1994.

Note that prior to Feko 2022.2.2 the phase reference was tied to the axial field component of a TE or TM mode. The corresponding expressions can be obtained by replacing $\boldsymbol{A}$ with $j\,\boldsymbol{A}$ in Equation 161 to Equation 163, Equation 165 to Equation 167, Equation 169 to Equation 171, Equation 173 to Equation 175, Equation 179 to Equation 181 and Equation 183 to Equation 185.

Rectangular waveguide expressions

Local Cartesian coordinates (u_1, u_2, u_3) are assumed. The factor $e^{j\omega t \pm j\beta u_3}$ is omitted for brevity, where β is the complex modal propagation coefficient. In the expressions below, let a be the dimension of the waveguide port in $\mathbf{u}_1$ (distance between S1 and S2), and b the dimension of the waveguide port in $\mathbf{u}_2$ (distance between S1 and S3). The modal cutoff coefficient is the same for TM- and TE-modes, and is given by,

$$\beta_c = \left(\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2\right)^{1/2} \tag{160}$$

For transverse magnetic (TM) modes the axial magnetic field component vanishes, and the axial electric field component for the $TM_{m,n}$ mode is expressed as,

$$E_{u_3}^{\mathrm{TM}}(u_1, u_2) = -\mathrm{j}\mathbf{A}\sin\left(\frac{m\pi u_1}{a}\right)\sin\left(\frac{n\pi u_2}{b}\right) \tag{161}$$

for m = 1,2, ... , and n = 1,2, ... **A** is the complex amplitude of the impressed mode in $\frac{V}{m}$. The remaining field components are,

$$E_{u_1}^{\mathrm{TM}}(u_1, u_2) = Z_{\mathrm{TM}} H_{u_2}^{\mathrm{TM}}(u_1, u_2) = -\frac{\beta}{\beta_c^2}\left(\frac{m\pi}{a}\right)\mathbf{A}\cos\left(\frac{m\pi u_1}{a}\right)\sin\left(\frac{n\pi u_2}{b}\right) \tag{162}$$

$$E_{u_2}^{\mathrm{TM}}(u_1, u_2) = -Z_{\mathrm{TM}} H_{u_1}^{\mathrm{TM}}(u_1, u_2) = -\frac{\beta}{\beta_c^2}\left(\frac{n\pi}{b}\right)\mathbf{A}\sin\left(\frac{m\pi u_1}{a}\right)\cos\left(\frac{n\pi u_2}{b}\right) \tag{163}$$

$$Z_{\mathrm{TM}}=\frac{\beta}{\omega\varepsilon} \tag{164}$$

For transverse electric (TE) modes the axial electric field component vanishes, and the axial magnetic field component for the $TE_{m,n}$ mode is expressed as,

$$H_{u_3}^{\mathrm{TE}}(u_1,u_2)=-\mathrm{j}\mathbf{A}\cos\left(\frac{m\pi u_1}{a}\right)\cos\left(\frac{n\pi u_2}{b}\right) \tag{165}$$

for $m=0,1,2,\ldots$, and $n=0,1,2,\ldots$, (but not both m and n zero). **A** is the complex amplitude of the impressed mode $\frac{A}{m}$. The remaining field components are,

$$E_{u_1}^{\mathrm{TE}}(u_1,u_2)=Z_{\mathrm{TE}}H_{u_2}^{\mathrm{TE}}(u_1,u_2)=\frac{\beta}{\beta_c^2}\left(\frac{n\pi}{b}\right)Z_{\mathrm{TE}}\mathbf{A}\cos\left(\frac{m\pi u_1}{a}\right)\sin\left(\frac{n\pi u_2}{b}\right) \tag{166}$$

$$E_{u_2}^{\mathrm{TE}}(u_1,u_2)=-Z_{\mathrm{TE}}H_{u_1}^{\mathrm{TE}}(u_1,u_2)=-\frac{\beta}{\beta_c^2}\left(\frac{m\pi}{a}\right)Z_{\mathrm{TE}}\mathbf{A}\sin\left(\frac{m\pi u_1}{a}\right)\cos\left(\frac{n\pi u_2}{b}\right) \tag{167}$$

$$Z_{\mathrm{TE}}=\frac{\omega\mu}{\beta} \tag{168}$$

Circular waveguide expression

Local cylindrical coordinates (r,φ,z) are assumed with the Z axis on the waveguide axis (S1-S3). The factor $e^{j\omega t\pm j\beta z}$ is omitted for brevity, where β is the complex modal propagation coefficient. The expressions below are valid for the fields inside a circular waveguide, for example, $r\le a$, where a is the radius of the waveguide port. $J_m(x)$ is the m^{th} order Bessel function of the first kind, and $J'_m(x)$ denotes the derivative with respect to the argument.

For transverse magnetic (TM) modes the axial magnetic field component vanishes, and the axial electric field component for the $\mathrm{TM}_{m,n}$ mode is expressed as,

$$E_z^{\mathrm{TM}}(r,\varphi)=-\mathrm{j}\mathbf{A}J_m(\beta_c r)\cos\left[m(\varphi-\varphi_0)\right] \tag{169}$$

for $m=0,1,2,\ldots$, and n = 1,2, ... **A** is the complex amplitude of the impressed mode in $\frac{V}{m}$ and φ_0 is the rotation angle. The modal cutoff coefficient β_c is the n^{th} zero of $J_m(\beta_c a)=0$. The remaining field components are,

$$E_r^{\mathrm{TM}}(r,\varphi)=Z_{\mathrm{TM}}H_\varphi^{\mathrm{TM}}(r,\varphi)=-\frac{\beta}{\beta_c}\mathbf{A}J'_m(\beta_c r)\cos\left[m(\varphi-\varphi_0)\right] \tag{170}$$

$$E_\varphi^{\mathrm{TM}}(r,\varphi)=-Z_{\mathrm{TM}}H_r^{\mathrm{TM}}(r,\varphi)=\frac{\beta m}{\beta_c^2 r}\mathbf{A}J_m(\beta_c r)\sin\left[m(\varphi-\varphi_0)\right] \tag{171}$$

$$Z_{\mathrm{TM}}=\frac{\beta}{\omega\varepsilon} \tag{172}$$

For transverse electric (TE) modes the axial electric field omponent vanishes, and the axial magnetic field component for the $TE_{m,n}$ mode is expressed as,

$$H_z^{\mathrm{TE}}(r,\varphi) = -\mathrm{j}\mathbf{A}J_m(\beta_c r)\cos\left[m(\varphi-\varphi_0)\right] \tag{173}$$

for $m = 0,1,2,\ldots$, and n = 1,2, ... **A** is the complex amplitude of the impressed mode in $\frac{A}{m}$ and φ_0 is the rotation angle. The modal cutoff coefficient β_c is the n^{th} zero of $J'_m(\beta_c a) = 0$. The remaining field components are,

$$E_r^{\mathrm{TE}}(r,\varphi) = Z_{\mathrm{TE}}H_\varphi^{\mathrm{TE}}(r,\varphi) = \frac{\beta m}{\beta_c^2 r}Z_{\mathrm{TE}}\mathbf{A}J_m(\beta_c r)\sin\left[m(\varphi-\varphi_0)\right] \tag{174}$$

$$E_\varphi^{\mathrm{TE}}(r,\varphi) = -Z_{\mathrm{TE}}H_r^{\mathrm{TE}}(r,\varphi) = \frac{\beta}{\beta_c}Z_{\mathrm{TE}}\mathbf{A}J'_m(\beta_c r)\cos\left[m(\varphi-\varphi_0)\right] \tag{175}$$

$$Z_{\mathrm{TE}} = \frac{\omega\mu}{\beta} \tag{176}$$

Coaxial waveguide expressions

Local cylindrical coordinates (r, φ, z) are assumed with the Z axis on the waveguide axis (S_1-S_3). The factor $e^{j\omega t \pm j\beta z}$ is omitted for brevity, where β is the complex modal propagation coefficient. The expressions below are valid for the fields inside a coaxial waveguide, for example, for $r_i \le r \le r_o$, where r_o is the radius of the outer conductor and r_i is the radius of the inner conductor of the coaxial waveguide port. $J_m(x)$ and $N_m(x)$ are m^{th} order Bessel function of the first and second kind, respectively and $J'_m(x)$ and $N'_m(x)$ denote the derivatives with respect to the argument.

The fundamental mode in a coaxial waveguide is a TEM wave and propagates with $\beta = \omega\sqrt{\mu\varepsilon}$.The axial electric and magnetic field components are zero for a TEM-mode, and the transverse field components have a static field distribution,

$$E_r^{\mathrm{TEM}}(r,\varphi) = Z_{\mathrm{TEM}}H_\varphi^{\mathrm{TEM}}(r,\varphi) = -\frac{\mathbf{A}}{r} \tag{177}$$

$$Z_{\mathrm{TEM}} = \sqrt{\frac{\mu}{\varepsilon}} \tag{178}$$

A is the complex amplitude of the impressed mode in *V*

For transverse magnetic (TM) modes the axial magnetic field component vanishes, and the axial electric field component for the $\mathrm{TM}_{m,n}$ mode is expressed as,

$$E_z^{\mathrm{TM}}(r,\varphi) = -\mathrm{j}\mathbf{A}\left(-\frac{J_m(\beta_c r)}{J_m(\beta_c r_o)} + \frac{N_m(\beta_c r)}{N_m(\beta_c r_o)}\right)\cos\left[m(\varphi-\varphi_0)\right] \tag{179}$$

for $m = 0,1,2,\ldots$, and n = 1,2, ... **A** is the complex amplitude of the impressed mode in $\frac{V}{m}$ and φ_0 is the rotation angle. The modal cutoff coefficient β_c is the n^{th} root of the transcendental characteristic equation, $N_m(\beta_c r_i)J_m(\beta_c r_o) = N_m(\beta_c r_o)J_m(\beta_c r_i)$, enforcing the boundary condition that E_z must be zero at r_i and r_o. The remaining field components are,

$$E_r^{\mathrm{TM}}(r,\varphi) = Z_{\mathrm{TM}} H_\varphi^{\mathrm{TM}}(r,\varphi) = -\frac{\beta}{\beta_c}\mathbf{A}\left(-\frac{J'_m(\beta_c r)}{J'_m(\beta_c r_o)}+\frac{N'_m(\beta_c r)}{N'_m(\beta_c r_o)}\right)\cos\left[m(\varphi-\varphi_0)\right] \quad (180)$$

$$E_\varphi^{\mathrm{TM}}(r,\varphi) = -Z_{\mathrm{TM}} H_r^{\mathrm{TM}}(r,\varphi) = \frac{\beta m}{\beta_c^2 r}\mathbf{A}\left(-\frac{J_m(\beta_c r)}{J_m(\beta_c r_o)}+\frac{N_m(\beta_c r)}{N_m(\beta_c r_o)}\right)\sin\left[m(\varphi-\varphi_0)\right] \quad (181)$$

$$Z_{\mathrm{TM}} = \frac{\beta}{\omega\varepsilon} \quad (182)$$

For transverse electric (TE) modes the axial electric field component vanishes, and the axial magnetic field component for the $TE_{m,n}$ mode is expressed as,

$$H_z^{\mathrm{TE}}(r,\varphi) = -\mathrm{j}\mathbf{A}\left(-\frac{J_m(\beta_c r)}{J'_m(\beta_c r_o)}+\frac{N_m(\beta_c r)}{N'_m(\beta_c r_o)}\right)\cos\left[m(\varphi-\varphi_0)\right] \quad (183)$$

for $m = 0,1,2,\ldots$, and n = 1,2, ... **A** is the complex amplitude of the impressed mode in $\frac{A}{m}$ and φ_0 is the rotation angle. The modal cutoff coefficient β_c is the n^{th} root of the transcendental characteristic equation, $N'_m(\beta_c r_i)J'_m(\beta_c r_o) = N'_m(\beta_c r_o)J'_m(\beta_c r_i)$, enforcing the boundary condition that the derivative of H_z normal to the conductors must be zero at the inner and outer radii. The remaining field components are,

$$E_r^{\mathrm{TE}}(r,\varphi) = Z_{\mathrm{TE}} H_\varphi^{\mathrm{TE}}(r,\varphi) = \frac{\beta m}{\beta_c^2 r} Z_{\mathrm{TE}}\mathbf{A}\left(-\frac{J_m(\beta_c r)}{J'_m(\beta_c r_o)}+\frac{N_m(\beta_c r)}{N'_m(\beta_c r_o)}\right)\sin\left[m(\varphi-\varphi_0)\right] \quad (184)$$

$$E_\varphi^{\mathrm{TE}}(r,\varphi) = -Z_{\mathrm{TE}} H_r^{\mathrm{TE}}(r,\varphi) = \frac{\beta}{\beta_c} Z_{\mathrm{TE}}\mathbf{A}\left(-\frac{J'_m(\beta_c r)}{J'_m(\beta_c r_o)}+\frac{N'_m(\beta_c r)}{N'_m(\beta_c r_o)}\right)\cos\left[m(\varphi-\varphi_0)\right] \quad (185)$$

$$Z_{\mathrm{TE}} = \frac{\omega\mu}{\beta} \quad (186)$$

Related reference

AR Card
AP Card
DP Card
SP Card

BO Card

This card defines a ground plane with the reflection coefficient approximation (at z = 0). All computations that follow this card will include the ground plane.

On the **Home** tab, in the **Planes / arrays** group, click the **Plane / ground** icon. From the drop-down list, click the **Reflective ground (BO)** icon.

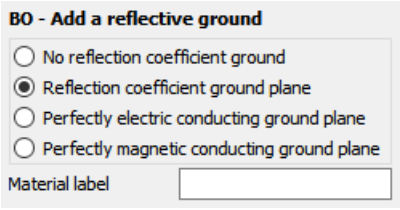


*Figure 855: The **BO - Add a reflective ground** dialog.*

Parameters:

No reflection coefficient ground	No ground plane. This option is used to switch off the reflection ground if the effect of different grounds are considered in a single input file, for example to consider a GF card.
Reflection coefficient ground plane	Use the reflection coefficient ground plane approximation with the material parameters specified in the remaining input fields.
Perfectly electric conducting ground	Use an ideal electric ground in the plane z = 0. In this case the remaining parameters are ignored.
Perfectly magnetic conducting ground	Use an ideal magnetic ground in the plane z = 0. In this case the remaining parameters are ignored.
Material label	The label of the medium to be used, as defined in the DI card.

It should be noted that it is not possible to calculate the fields below the ground plane (in the z < 0 half space). In addition all structures must be in the region z > 0. If calculations inside the ground are required, for example when there are structures below ground, the exact Sommerfeld integrals (GF card) must be used.

When using a perfect electric or magnetic reflection coefficient ground plane, structures can be arbitrarily close to the ground (while remaining above it). Segment end points and triangle edges lying in the plane of the ground plane will make electrical contact with a perfect electric ground plane. For a perfect magnetic ground plane the boundary condition forces the current to zero at this point.

If real ground parameters are used, the reflection coefficient approximation is more accurate for structures further from the ground plane. Typically structures should not be closer than about $\frac{\lambda}{10}$ will give a warning if this is the case).

A dielectric ground (real earth) can only be used with bodies treated with MoM, MLFMM, PO, FEM, or the hybrid MoM/PO.

Note: The hybrid MoM/UTD method cannot be used in the presence of a real ground.

Related tasks

Defining an Infinite Planar Multi Layer Substrate (CADFEKO)

CA Card

The CA card is used to define a section of a shielded cable which is used for irradiation (for example, computing induced currents and voltages at the cable terminals) due to external sources. Transmission line theory is applied, for example, no need to discretise the cable as with the MoM. A section is defined as a straight part of a cable (one cable can consist of multiple sections).

In the **Solve/Run** tab, in the **Cables** group, click the **Irradiating cable (CA)** icon.

Note: That when new models are created, it is recommended to use the SH card (shield definition), CD card (cable cross section definition), CS card (cable path section definition) and LC cards (cable loads).

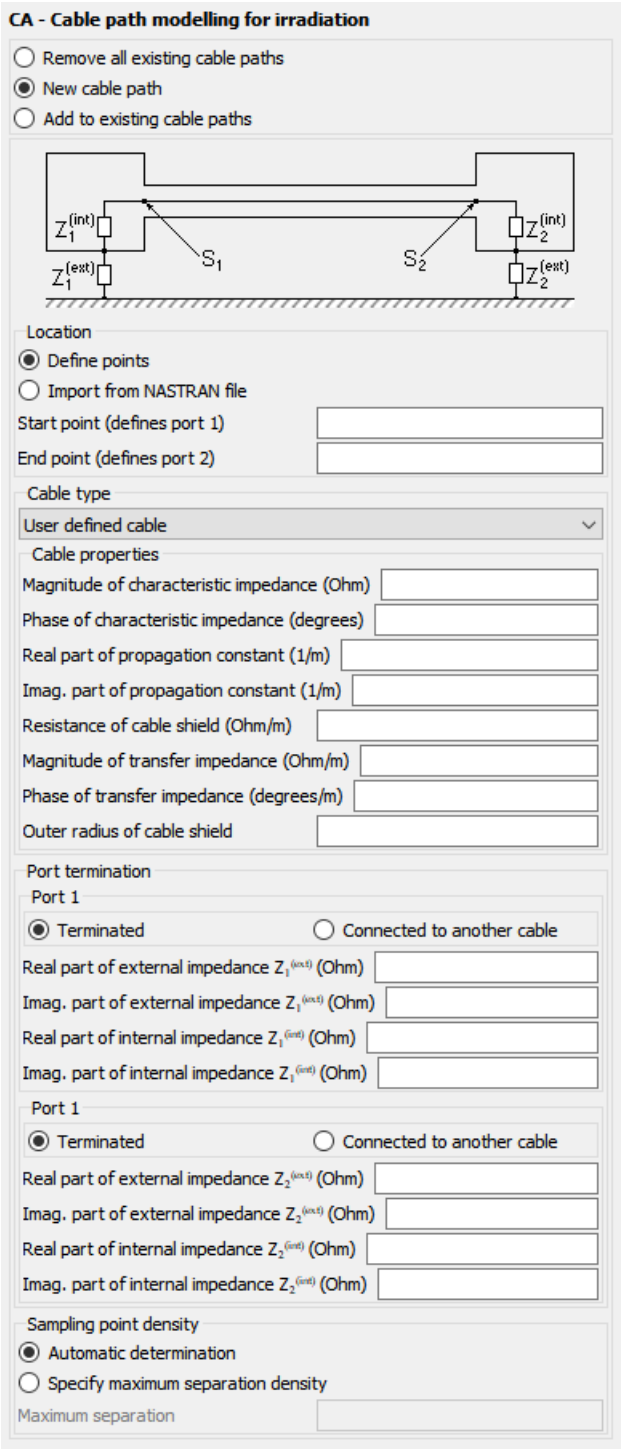


*Figure 856: The **CA - Cable path modelling for irradiation** dialog.*

Parameters:

Remove all existing cable paths If checked, all previously defined cable paths are removed. All the other input parameters are ignored.

New cable path Defines a new cable path, all previously defined cable paths are replaced.

Add to existing cable paths An additional cable path is defined (for example, the previously added ones will be kept).

Location The location of the cable path section can be specified as two points or imported from a NASTRAN file.

- **Define points** The **Start point** and the **End point** of the cable path section are defined by point names. These points must have been defined previously with a DP card (or by an external import).
- **Import from NASTRAN file** The name of the NASTRAN file and the property ID of the segments that have to be imported are required to import the cable path section.

Cable type This specifies the type of cable. There are two possibilities: User defined cable or a predefined cable from the internal Feko cable database:

- **If User defined cable is selected** The user has to enter all the cable properties in the **Cable properties** section which will then be enabled. The units of the individual parameters are included in the description.

  > **Note:** That the **Outer** radius of cable shield will be scaled by any active SF card.

  Also keep in mind that most of these parameters may depend on the frequency and thus one might use variables or expressions.
- **If a predefined cable type is selected** The section containing the cable properties will be disabled, and all required parameters will be retrieved automatically from an internal Feko cable database. There are several commonly found shielded cable types (up to now all coaxial cables) included.

Port termination This section is used to define the ports (for example, the two ends of the cable path section defined by this CA card). For each

port the user can decide if it is **terminated** by an (internal and external) impedance or if this end of the cable path section shall be **connected to another cable** path section.

The termination impedance value fields are only enabled if the corresponding port is terminated. The **internal** impedance terminates the inner conductor against the cable shield and the **outer** impedance terminates the cable shield against the ground plane.

> **Note:** That a complete cable path has to be terminated on both ends. (Detailed information is given below.)

Sampling point density

The cable path section will be subdivided into small segments for the computation of the induced currents and voltages. The electric and magnetic field strengths will be evaluated at each segment's centroid, so this setting influences the accuracy of the computed result, but also the computation time. The setting **Automatic determination** will choose the segment length automatically (which should be adequate for most cases). If the **maximum separation distance** is specified, then this value will override the automatic mechanism. Note that this manual value will be scaled by any active SF card.

Transmission line theory (TLT) is used in conjunction with the field calculation using the method of moments (MoM) to compute the voltage coupled-in at the termination impedances of a cable close to a conducting (metallic) ground. The cable itself is not taken into account when computing the external field distribution and it does not affect the field distribution at all, for example from a **field-viewpoint** of the scenario, the cable is not present at all. This is also the reason for the reduced number of unknowns in comparison to a full MoM solution: the cable itself is not modelled in the geometry and therefore not meshed into (wire) segments and it is therefore not necessary to introduce a very fine mesh on the ground plane underneath the cable. A further advantage of the transmission line approach is that multiple cable scenarios can be investigated in the same model (for example, different cable positions) without repetition of a time-consuming solution of the whole model (for example, MoM or MLFMM). Analysing another cable is similar to computing the near field at another point.

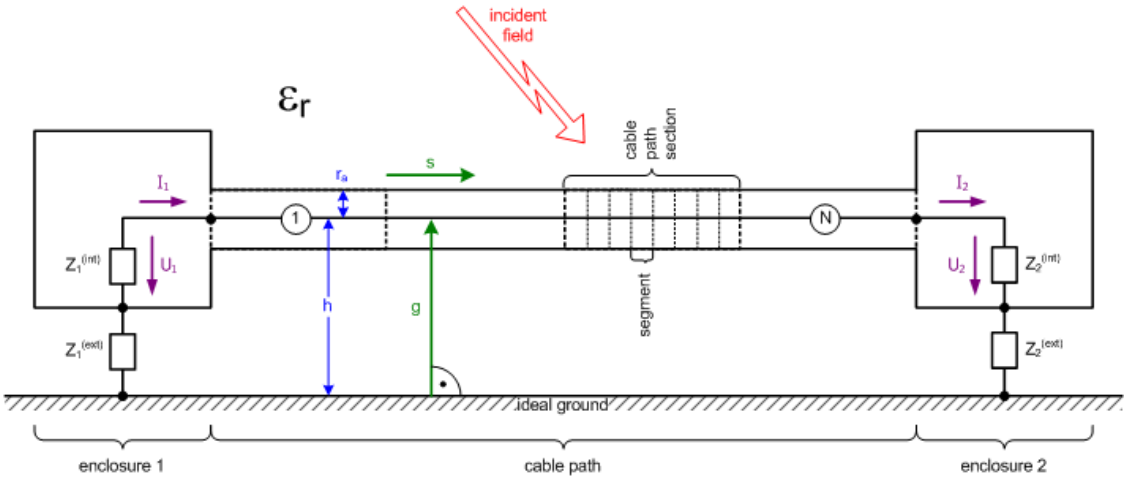


Figure 857: Complete scenario of a cable path.

The term cable path refers to the complete cable from its start point to its end point. Thus the cable path can consist of a single or multiple cable path sections. Each cable path section is then again subdivided into the segments which are used for the computation. Each complete cable path has to be terminated on both ends.

An arbitrary number of cable path sections can be defined. The complete scenario is shown in Figure 857. It consists of the cable path connecting two (virtual) enclosures with the termination impedances. The cable is illuminated by an external electromagnetic field (as caused by sources and other radiating structures in the model) which couples into the cable and causes the currents and voltages in the internal termination impedances. For the calculation to work properly the segment direction vector **s** and the ground vector **g** must be (almost) perpendicular. This figure also shows the cable path, the cable path sections (1 . .. N) and the segments.

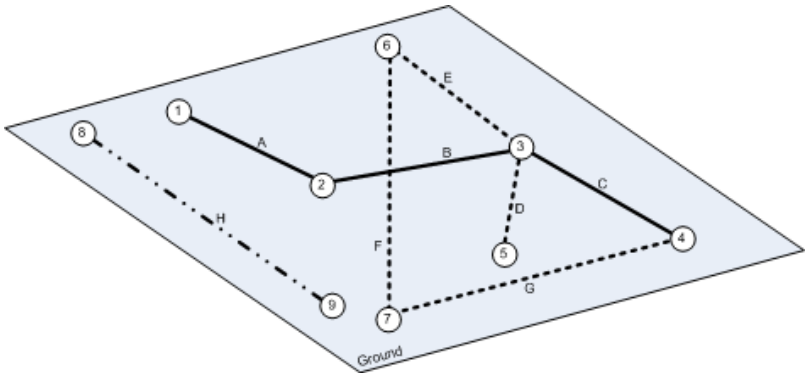


Figure 858: Possible ways to define cable paths.

Figure 858 shows the setup of a number of cable paths. There are three cable paths in total (distinguished by the line style):

Table 69: Cable path definitions

Path	Description
1	Cable path from point 1 to 4 consisting of the cable path sections A, B, C.
2	Cable path from point 5 to 4 consisting of the cable path sections D, E, F, G.
3	Cable path from point 8 to 9 consisting of the single cable path section H.

Note: That even if there are crossings (section B and F) or sections using the same points (for example, sections B, C, D and E regarding point 5) there will be no conducting connection between sections belonging to different cable paths.

As the cable paths will be assembled automatically by searching and matching the points' coordinates, the order of the CA cards determines how the cable path gets built. (The search is always started at the first CA card defining a termination impedance and then the cards will be processed in the order they appear in the input file.) For example, in order to get the situation as shown in Figure 858 the CA cards have to be in the following order:

- CA Defining section A (start cable path 1).
- CA Defining section B (continue cable path 1).
- CA Defining section C (end cable path 1).
- CA Defining section D (start cable path 2).
- CA Defining section E (continue cable path 2).
- CA Defining section F (continue cable path 2).
- CA Defining section G (end cable path 2).
- CA Defining section H (single segment cable path 3).

or (note the changed relative cable path number):

- CA Defining section A (start cable path 1).
- CA Defining section C (continue cable path 1).
- CA Defining section B (end cable path 1).
- CA Defining section H (single segment cable path 2).
- CA Defining section E (start cable path 3).
- CA Defining section D (continue cable path 3).
- CA Defining section F (continue cable path 3).
- CA Defining section G (end cable path 3).

but not in the following order:

- CA Defining section A (start cable path 1).
- CA Defining section B (continue cable path 1).
- CA Defining section D (end cable path 1).

- CA Defining section H (single segment cable path 2).
- CA Defining section E (start cable path 3).
- CA Defining section C (continue cable path 3).
- CA Defining section F (continue cable path 3).
- CA Defining section G (end cable path 3).

In the last case the cable path section D will be connected to the cable path section B which is not intended! (Cable path 1 would then start at point 1 and end at point 5 consisting of sections A, B, D and cable path 3 will then start at point 4 and end at point 4 consisting of the sections C, E, F, G.) Additionally in this case the cable path 3 will form a closed loop, but one has to keep in mind, that even in this case the two ends of the cable path are not connected at point 4.

Current limitations of the cable irradiation analysis using the CA card:

- For the built-in cable types, the frequency range is limited as this data is based on measurements only available for a certain frequency range. Currently the frequency range from 10 kHz up to 500 MHz is supported for all those cable types. (An error is given by Feko if one tries to set a frequency which is not in that range.) This restriction is not applied for user defined cables, since it is assumed that the user has supplied the required cable parameters for the frequency under consideration.
- The cable must be homogeneous, for example, the cable parameters may not vary for the sections belonging to the same cable path (this is enforced for all cables).
- Currently only single conductor coaxial cables are supported. A multi-conductor cable cannot be modelled.
- Cables cannot be used in connection with UTD, but any other method is possible to represent the external configuration (for example, a car body modelled with MoM or MLFMM).
- A reference plane acting as ground is required for the cable coupling algorithm. This is currently implemented in such a way that only metallic triangles and perfectly conducting ground planes (PEC-ground defined by a BO-card) are considered. Feko will give an error if the special Green's functions are used in connection with a cable analysis.
- Connections of cables with wires or crossings with wires are not allowed. Even if a cable path starts or ends at a point where also a wire segment starts or ends or if a cable path crosses a wire, there will be no electrical connection established between the cable and the wire at such a point.

Note: Also that this card is only intended to compute the coupling into the cable from an external field strength. Thus no additional voltage or current sources are supported at the ports.

Related tasks
Defining a Cable Path (CADFEKO)
Related reference
BO Card
DP Card
CD Card
CS Card
LC Card

CD Card

This card defines a cable cross section.

In the **Solve/Run** tab, in the **Cables** group, click the **Cable Cross section (CD)** icon.

Single Conductor Cable

This option defines a single cable.

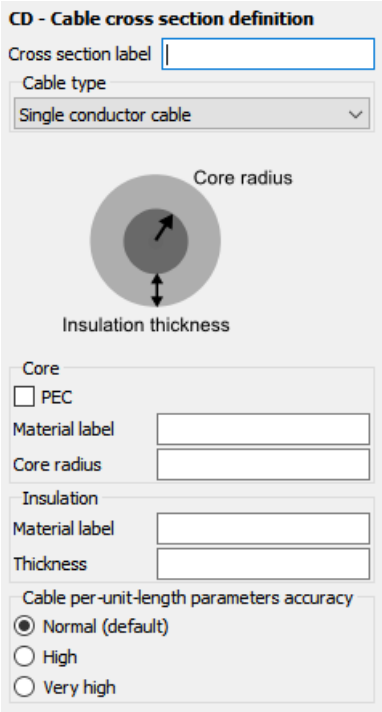


*Figure 859: The **CD - Cable cross section definition** dialog.*

Parameters:

Cross section label — The label of the cross section.

Core

- **PEC** — Select this option to set the core of the cable to PEC.
- **Material label** — The label of the metallic medium (as defined in the DI card) to be used for the core.
- **Core radius** — The radius of the core.

Insulation

- **Material label** — The label of the metallic medium (as defined in the DI card) to be used for the core.

	Thickness	The thickness of the material to be used as insulator.
Cable per-unit-length parameters accuracy	The **Cable per-unit-length parameters accuracy** can be increased from **Normal (default)** to **High** or **Very high**.	

Related tasks
Defining a Single Conductor Cable (CADFEKO)

Coaxial Cable

This option defines a coaxial cable.

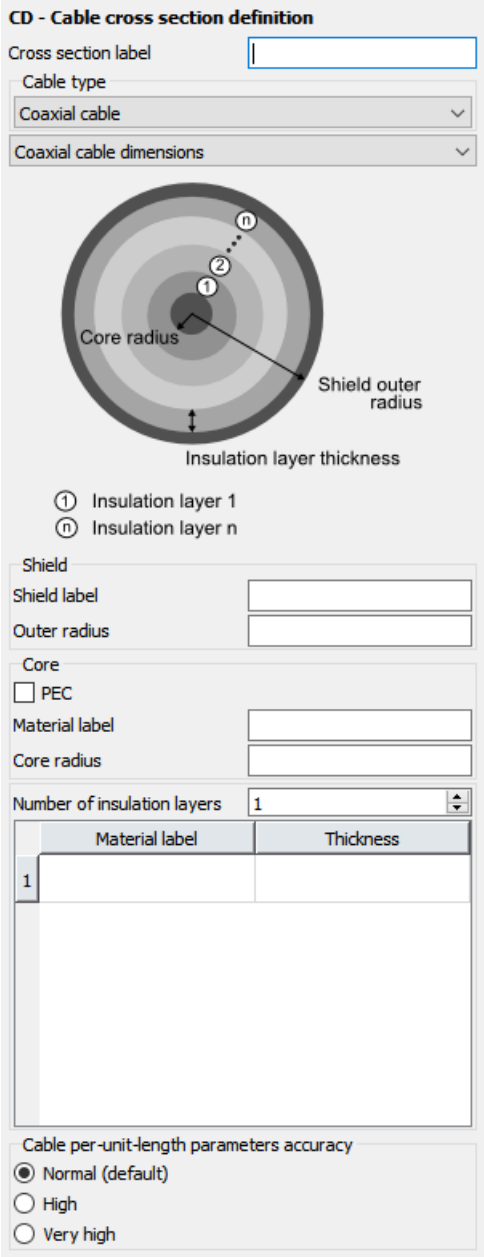


*Figure 860: The **CD - Cable cross section definition** dialog.*

Three types of coaxial cables are supported:

1. Coaxial cables defined by cable characteristics.
2. Coaxial cables defined by the cable dimensions.
3. Internally defined database of cables.

Parameters: Coaxial cables defined by cable characteristics

Cross section label	The label of the cross section.
Shield label	The label of the shield (as defined in the SH card).
Outer radius	The outer radius of the shield.
Magnitude of the characteristic impedance (Ohm)	The magnitude of the characteristic impedance of the cable.
Attenuation (dB/m)	Specify the attenuation of the coaxial cable in dB/m.
Velocity of propagation (%)	Specify the propagation speed through the coaxial cable relative to the speed of light.

Parameters: Coaxial cables defined by cable dimensions

Shield	**Shield label**	The label of the shield (as defined in the SH card) to be used
	Shield outer radius	The outer radius of the shield.
Core	**PEC**	Select this option to set the core of the cable to PEC.
	Material label	The label of the metallic medium (as defined in the DI card) to be used for the core.
	Core radius	The radius of the core.
Number of insulation layers	The number of the insulation layers in the cable.	
Material label	The label of the material (as defined in the DI card) to be used for the insulation layers.	
Thickness	The thickness of the insulation layers.	
Cable per-unit-length parameters accuracy	The **Cable per-unit-length parameters accuracy** can be increased from **Normal (default)** to **High** or **Very high**.	

Parameters: Internally defined database of cables

No parameters are specified, but a predefined coaxial cable type can be selected from the drop-down list. Several common types of shielded cables are included in this list.

Related tasks

Adding a Predefined Coaxial Cable from Industry (CADFEKO)
Defining a Coaxial Cable Using Cable Characteristics (CADFEKO)
Defining a Coaxial Cable Using Cable Dimensions (CADFEKO)

Ribbon Cable

This option defines a ribbon cable (with round cores only).

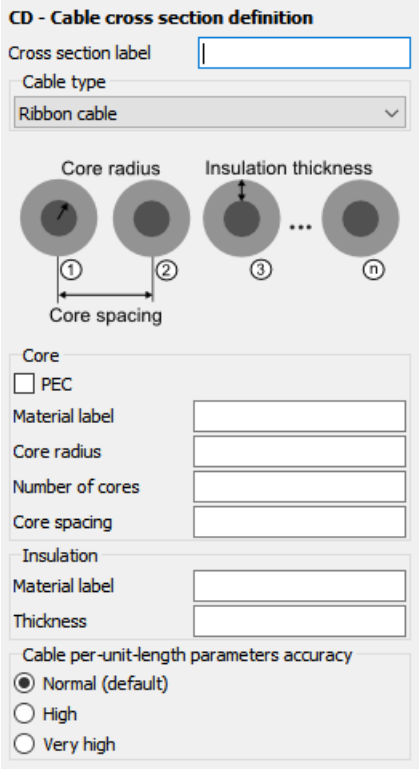


*Figure 861: The **CD - Cable cross section definition** dialog.*

Parameters:

Cross section label — The label of the cross section.

Core

- **PEC** — Select this option to set the core of the cable to PEC.
- **Material label** — The label of the metallic medium (as defined in the DI card) to be used for the core.
- **Core radius** — The radius of the core.
- **Number of cores** — The number of cables which constitute the ribbon cable.
- **Core spacing** — The spacing between the adjacent cores.

Insulation

- **Material label** — The label of the material (as defined in the DI card) to be used for the insulation layer.

	Thickness	The thickness of the insulation layer (optional).
Cable per-unit-length parameters accuracy	The **Cable per-unit-length parameters accuracy** can be increased from **Normal (default)** to **High** or **Very high**.	

Related tasks
Defining a Ribbon Cable (CADFEKO)

Twisted Pair Cable

This option defines twisted pair cables with specified insulation, radius, twist pitch length and direction.

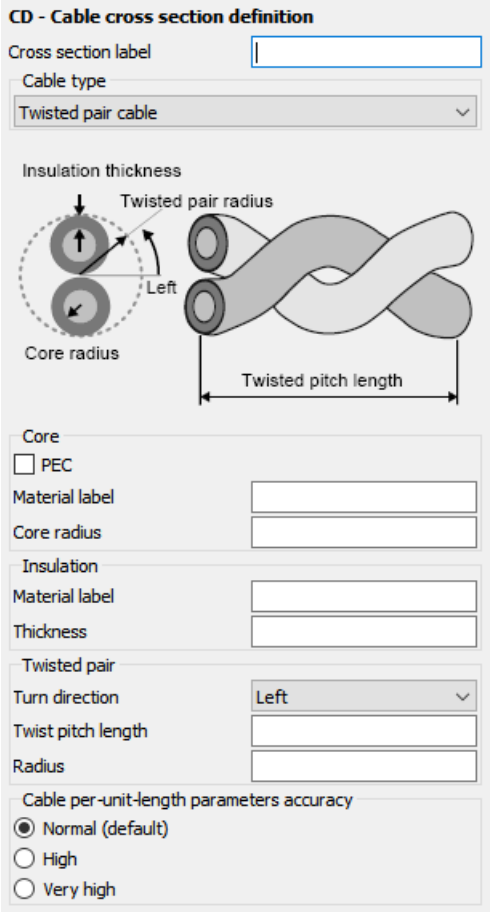


*Figure 862: The **CD - Cable cross section definition** dialog.*

Parameters:

Cross section label	The label of the cross section.	
Core	**PEC**	Select this option to set the core of the cable to PEC.

	Material label	The label of the metallic medium (as defined in the DI card) to be used for the core.
	Core radius	The radius of the core.
Insulation		
	Material label	The label of the material (as defined in the DI card) to be used for the insulation layer.
	Thickness	The thickness of the insulation layer (optional).
Twisted pair		
	Turn direction	Select left or right turn direction for the twisted pair.
	Twist pitch length	The axial length in which a twisted pair strand returns to its original relative position in a twisted conductor.
	Radius	The outer radius of the twisted pair cable.
Cable per-unit-length parameters accuracy	The **Cable per-unit-length parameters accuracy** can be increased from **Normal (default)** to **High** or **Very high**.	

Related tasks
Defining a Twisted Pair (CADFEKO)

Non-Conducting Element

This option defines a non-conducting fibre used as spacing elements and for additional strength to cables.

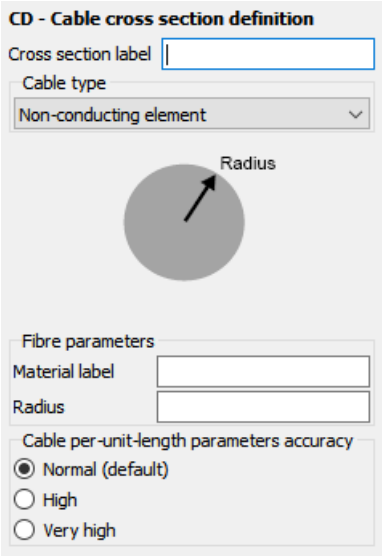


*Figure 863: The **CD - Cable cross section definition** dialog.*

Parameters:

Cross section label	The label of the cross section.
Material label	The label of the dielectric (as defined in the DI card) to be used for the non-conducting fibre.
Radius	The radius of the non-conducting fibre.
Cable per-unit-length parameters accuracy	The **Cable per-unit-length parameters accuracy** can be increased from **Normal (default)** to **High** or **Very high**.

Related tasks

Defining a Non-Conducting Element (CADFEKO)

Bundle (Mixed) Cable

This option allows the construction of complex multi-core cables based on existing cables created from the other cable types.

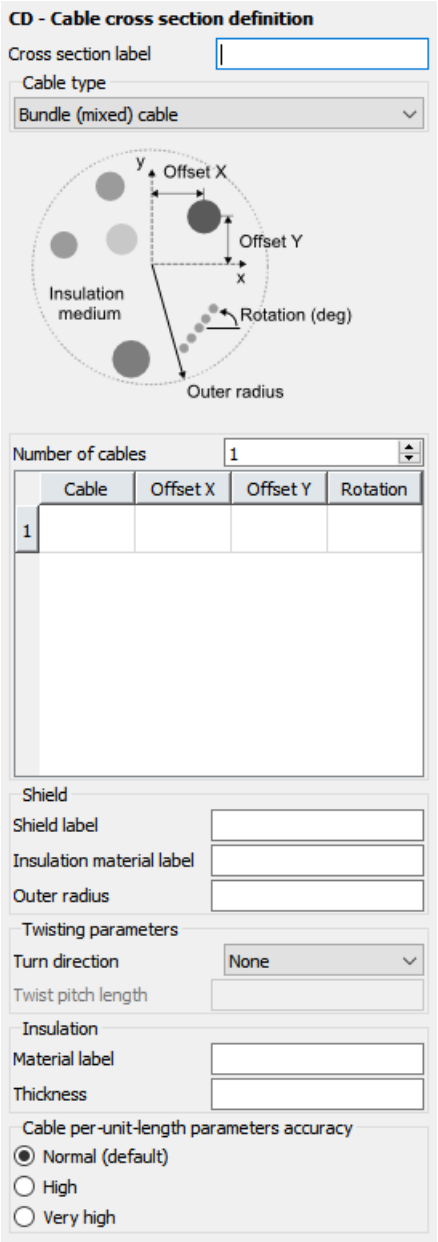


*Figure 864: The **CD - Cable cross section definition** dialog.*

Parameters:

Cross section label	The label of the cross section.
Number of cables	The number of cables that constitutes the bundle cable.
Cable	The cross section label.

Offset X	The x offset for the respective cable from the cable path.	
Offset Y	The y offset for the respective cable from the cable path.	
Rotation	The rotation of the respective cable relative to the bundle.	
Shield	**Shield label**	The label of the shield (as defined in the SH card) to be used (optional).
	Insulation material label	The label of the material (as defined in the DI card) to be used as the insulation.
	Outer radius	The outer radius of the bundle.
Twisting parameters	**Turn direction**	Select left or right turn direction for the twisted pair bundle.
	Twist pitch length	The axial length in which a twisted pair bundle returns to its original relative position.
	Core radius	The radius of the core.
Insulation	**Material label**	The medium name of an insulating sheath.
	Thickness	Thickness of the sheath layer
Cable per-unit-length parameters accuracy	The **Cable per-unit-length parameters accuracy** can be increased from **Normal (default)** to **High** or **Very high**.	

Related tasks
Defining a Cable Bundle (CADFEKO)
Related reference
DI Card
SH Card

CF Card

This card sets the type of integral equation for perfectly conducting metallic surfaces.

In the **Solve/Run** tab,in the **Solution settings** group, click the EFIE CFIE **Integral equation (CF)** icon.

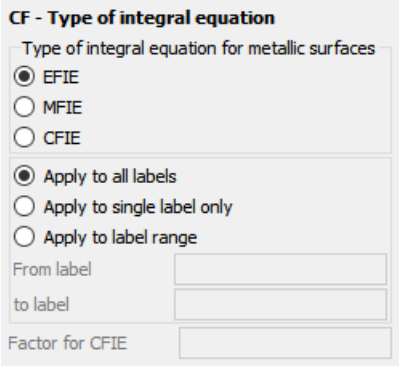


*Figure 865: The **CF - Type of integral equation** dialog.*

Parameters:

Type of integral equation for metallic surfaces	Chose between the EFIE (electric field integral equation) and the CFIE (combined field integral equation). See the comment below for more details.
Apply to all labels	The selected type of integral equation is applied globally to all metallic surfaces, irrespective of their label.
Apply to single label only	Here the selection of the type of integral equation applies to a single label only, which is entered into the field **From label**.
Apply to label range	Here the selection of the type of integral equation applies to a range of labels, which is entered into the fields **From label** and **to label**.
Factor of CFIE	In the CFIE formulation electric and magnetic terms are combined with a factor. Leaving this input field empty will select the default value of 0.2. It is generally not recommended to change this value.

The EFIE is the default in Feko if no CF card is used or for labels where no setting is made at the CF card. It is the most general formulation and can be applied to both open and closed bodies. The CFIE can only be used in connection with closed objects. The advantage is that the conditioning of the system of linear equations is better. In particular in connection with MLFMM the convergence can be improved if the CFIE is used for closed parts of an object. Note that the CFIE can be used together with the EFIE on the same object.

Note: All triangle normals should point away from the zero field region.

For the CFIE in addition to the fundamental restriction that the surface must be closed, these further conditions apply:

- The normal vector must point outwards (from the closed field free region into the domain of interest where there are sources and fields to be computed).
- Using symmetry in order to reduce the memory or run-time is not supported (it will be switched off automatically).
- The CFIE formulation for the MoM cannot be used together with the MoM/PO, MoM/UTD, or MoM/FEM hybrid methods.
- The CFIE formulation can be used only with the free space Green's function - using the spherical or planar multilayer Green's functions is not supported.
- When using the CFIE, dielectric bodies (solved using SEP, FEM or MoM/MLFMM) may be present in the model, though all of the CFIE surfaces must be perfectly conducting (no coating or skin effect and so forth).

Note that multiple CF cards can be used in order to specify for instance that the CFIE shall be used at multiple distinct labels which do not form a range. The setting for **Factor for CFIE** is global (not per label), the value read from the last CF card will be used.

Related tasks
Using the CFIE for Close Metallic Volumes (CADFEKO)

CG Card

The CG card defines the method to solve the matrix equation.

On the **Solve/Run** tab, in the **Solution settings** group, click the **Preconditioner** icon.

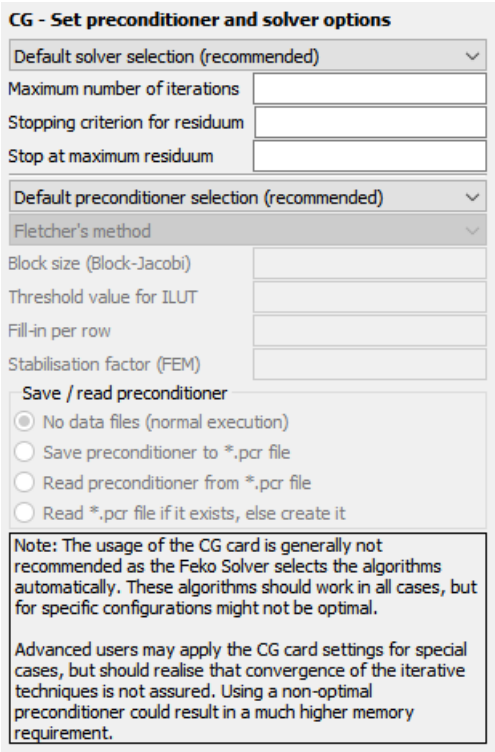


*Figure 866: The **CG - Set preconditioner and solver options** dialog.*

Normally the CG card should not be used. Feko automatically selects optimal solution techniques, preconditioners and other options depending on the problem type. These algorithms should be sufficient in all cases, but they might not be optimal for specific MLFMM and FEM configurations that use iterative solvers.

For these specific solutions, advanced users could apply the CG card after consultation with Feko technical support. Nevertheless convergence of the iterative techniques cannot be guaranteed. In addition the memory requirements could be higher than what is desired.

Warning: Any models derived from models containing a CG card should also be re-considered if they contain a CG card.

Parameters:

Matrix solution method:

- **Default solver selection (recommended)**. When this option is selected, then Feko will automatically select a suitable solver along with all its required parameters. The choice depends on whether Feko is executed sequentially or in parallel, but also which solution method is employed (for example direct LU decomposition solver for the MoM while an iterative solver is used for MLFMM or FEM). This option has, regarding the solver type, the same effect as not using a CG card,

but still allows the user to change the default termination criteria for the iterative solver types, or to change the preconditioner.

- **Gauss elimination (LINPACK routines)** Use Gauss elimination from the LINPACK routines.
- **Conjugate Gradient Method (CGM)**
- **Biconjugate gradient method (BCG)**
- **Iterative solution with band matrix decomposition**
- **Gauss elimination (LAPACK routines)** Use Gauss elimination from the LAPACK routines.
- **Block Gauss algorithm (matrix saved to disk)** The block Gauss algorithm is used (in case the matrix has to be saved on the hard disk, for example when a sequential out-of-core solution is performed).
- **CGM (Parallel Iterative Method)**
- **BCG (Parallel Iterative Method)**
- **CGS (Parallel Iterative Method)**
- **Bi-CGSTAB (Parallel Iterative Method)**
- **RBi-CGSTAB (Parallel Iterative Method)**
- **RGMRES (Parallel Iterative Method)**
- **RGMRESEV (Parallel Iterative Method)**
- **RCGR (Parallel Iterative Method)**
- **CGNR (Parallel Iterative Method)**
- **CGNE (Parallel Iterative Method)**
- **QMR (Parallel Iterative Method)**
- **TFQMR (Parallel Iterative Method)**
- **Parallel LU-decomposition (with ScaLAPACK routines)**. The parallel LU decomposition with ScaLAPACK (solution in main memory) or with out-of-core ScaLAPACK (solution with the matrix stored to hard disk). This is the default option for parallel solutions and normally the user need not change it.
- **QMR (QMRPACK routines)**
- **Direct sparse solver**. Direct solution method for the ACA or FEM (no preconditioning).

  When using the parallel , the factorisation type, which slightly impacts runtime and memory, can be specified.

Maximum number of iterations	The maximum number of iterations for the iterative techniques.
Stopping criterion for residuum	Termination criterion for the normalised residue when using iterative methods. The iterative solver will stop when the normalised residue is smaller than this value.
Stop at maximum residuum	For the parallel iterative methods, the solution is terminated when the residuum becomes larger than this value. The iterative solution will stop with an error message indicating that the solution has diverged.

Preconditioners

Refer to Preconditioners for MLFMM and Preconditioners for FEM for more information.

Save/read preconditioner

For the incomplete LU preconditioners used with the FEM the preconditioner can be computed only once and written to a `.pcr` file. Then for a subsequent solution it can be read from this file saving runtime. Since the FEM preconditioners depend only on the FEM part of the matrix, this method is useful when only the MoM part of a FEM/MoM problem has changed.

Iterative solutions are used in for example the MLFMM and the FEM. There could be cases where the residuum decreases but at a very low rate. Instead of waiting very long until the maximum number of iterations is reached, the user can press Ctrl+C or Ctrl+Break (under Windows) or send the SIGINT/SIGTERM signals (under UNIX) so that Feko will stop with the iterations and resume with the further processing (for example the far field computations) using the solution associated with the best residuum obtained so far. To really interrupt a Feko job Ctrl+C or Ctrl+Break must be pressed a second time (or the corresponding signal must be sent once more).

Related concepts

MLFMM Settings

Preconditioners for MLFMM

Preconditioners for FEM

CH Card

This card is used to group cable harness specific properties.

On the **Solve/Run** tab, in the **Cables** group, click the **Cable harness (CH)** icon.

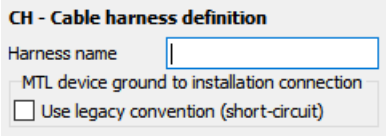


*Figure 867: The **CH - Cable harness definition** dialog.*

Parameters:

Harness name

The name of the cable harness.

Use legacy convention (short-circuited)

This check box can be used to change how the local circuit device ground and global installation concepts should be treated with respect to each other in an MTL cable harness definition.

For MTL the legacy convention enforces the device ground and installation to be short-circuited. If left unchecked, the device ground and installation are distinct concepts and may be open-circuited, short-circuited or connected through bonding circuit elements.

For combined MoM/MTL harnesses the device ground indicates the transitioning point from circuit to full wave solver. Connections beyond the device ground should be included in the full wave model.

CI Card

The CI card is used to define interconnections and terminations between cables.

In the **Solve/Run** tab, in the **Cables** group, click the **Interconnect cable (CI)** icon.

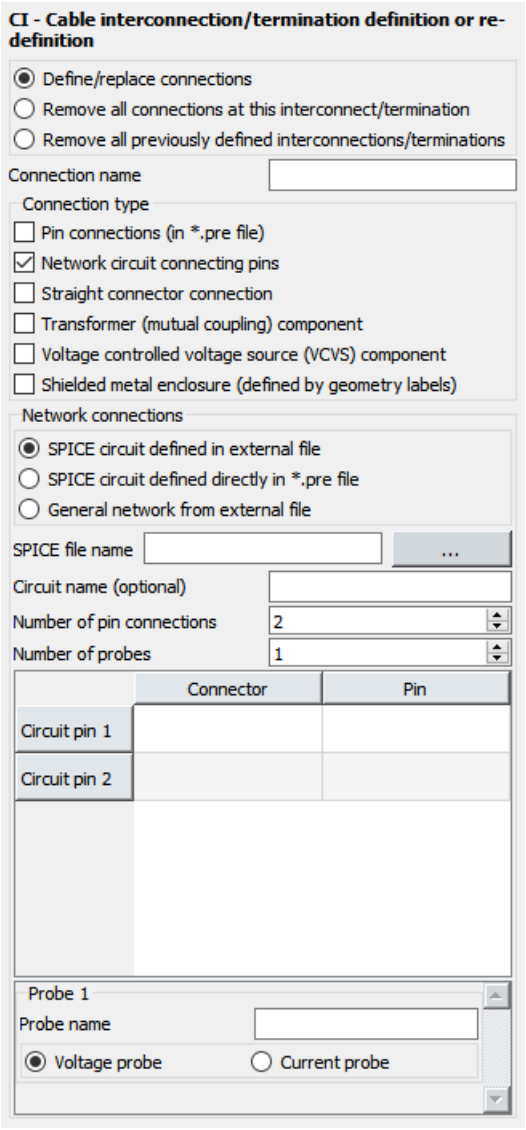


*Figure 868: The **CI - Cable interconnection/termination definition or re-definition** dialog.*

Parameters:

Define/ replace connections	New connection, replaces previous connections with the same name.
Remove all connections at this interconnect/termination	All previously defined connections (with this name) at this specific interconnection/termination are removed.
Remove all previously defined interconnections/terminations	All previously defined interconnections/terminations are removed.

Connection name

The name of the connection.

Pin connections (in `.pre` file)

The connection between pins are specified in the `.pre` file.

Pin

The pin number identifies the conductor associated with the cable path section as defined by **Connector label** in the CS card. If the pin is set to 0, the connector pin is connected to the local circuit device ground. If the pin is set to -1, the connector pin is connected to the global installation.

Loading

The following loading options between two pins are available: direct short, open, complex, series RLC, parallel RLC circuit and a 1-port Touchstone load specified in either a `.s1p`, `.z1p` or `.y1p` file.

Probes

A voltage and/or current probe may be placed between two pins.

Network circuit connecting pins

Note: This option is not supported in conjunction with a closed metallic surface.

Spice circuit defined externally and directly in `.pre` file

A SPICE circuit is connected between multiple pins. The SPICE circuit can be provided as an external file or it can be included directly into the `.pre` file.

SPICE file name

The file name of the SPICE circuit to be connected between two pins. The file name is required when the SPICE circuit is loaded from an external file, but should not be used when entering the SPICE circuit directly in the `.pre` file. When the SPICE circuit is included directly in the `.pre` file, a field will be displayed where the circuit can be entered.

Circuit name (optional)

The main sub-circuit name to be used from the `.cir` file. If left unspecified, the name should match the sub-circuit name.

Number of pin connections

The number of pin connections to be made.

Number of probes

The number of voltage and/or current probes.

Connector
The label of the connector to be connected to the SPICE circuit.

Pin
The pin of the connector to which the SPICE circuit will be connected to.

Probes
The type of probe, either voltage or current.

General network from external file
A Touchstone network is connected between multiple pins. The general network is provided as an external file with S, Y or Z parameters.

Touchstone file name
The file name of the Touchstone network to be added to the cable schematic.

Number of ports
The number of ports in the general network.

> **Note:** There are two pin connections for each port definition.

Connector +
The label of the connector to be connected to the general network.

Connector -
The label of the connector to be connected to the general network.

Pin +
The pin of the connector to which the general network will be connected to.

Pin -
The pin of the connector to which the general network will be connected to.

Straight connector connection This option is used when similar cables are connected.

Number of connections
The number of connections between similar cables.

Connector
The label of the connectors of the cables which will be connected.

Transformer (mutual coupling) component

This option is used when a transformer is connected in a circuit.

Number of transformer components
The number of transformers connected between similar cables.

Coupled inductor
Specify the connections for coupled inductor 1 and 2 of the transformer component.

Inductance
The inductance of the coupling inductor in Henry.

Connector
The label of the connector which the positive or negative pin of the coupling inductor is connected to.

Pin
The pin of the connector to which the positive or negative pin of the transformer coupling inductor is connected to.

Phase dot position
Select at which pin (positive or negative) the phase dot is positioned at for the coupling inductor.

Coupling
Specify the **coupling coefficient (K)** a value greater than 0 and less than or equal to 1.

Probes
The type of probe, either voltage or current for coupling inductor 1 and 2.

Voltage controlled voltage source (VCVS) component

This option is used to specify the connection of the voltage controlled voltage source in a circuit.

Number of VCVS components
The number of VCVS components between similar cables.

Connection pins

Connector
The label of the connector which the positive or negative pin of the VCVS is connected to.

Pin
The pin of the connector to which the positive or negative pin of the VCVS is connected to.

Control pins

Connector
The label of the connector which the positive or negative pin of the control (measurement) pin or is connected to.

Pin
The pin of the connector to which the positive or negative pin of the control (measurement) pin is connected to.

Control voltage
Specify the **Voltage gain** for the VCVS.

Probes
The type of probe, either voltage or current.

Closed metallic surface (defined by geometry labels)

Note: This option is not supported in conjunction with network circuit connecting pins.

This option is used to connect combined MoM/MTL cable paths to the same harness using a closed metallic surface with a label.

Number of combined MoM/MTL paths
The number of the combined MoM/MTL cable paths terminating on the metallic surface.

Number of labels
The number of geometry labels that define the closed metallic surface.

Labels
The list of geometry labels that define the closed metallic surface.

Connector name
All circuit connections should be made to this connector name and pin 0.

Adding probes in SPICE circuits

Voltage and current probes can be added to SPICE circuits so that these values become available in POSTFEKO. Additional circuitry is required in the SPICE circuit so that these values can be obtained for the probes. This will be explained using an example model illustrated in Figure 869 representing the SPICE circuit below.

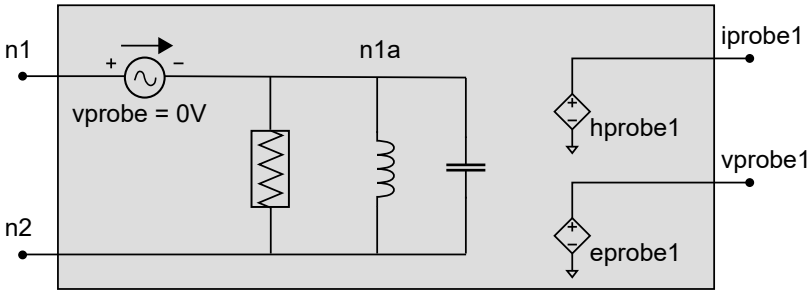


Figure 869: Example SPICE circuit with voltage and current probes.

```
* RLC parallel circuit

.SUBCKT SPICE_RL n1 n2 vprobe1 iprobe1

* RLC circuit
R2 n1a n2 25
L2 n1a n2 5e-3
C2 n1a n2 20e-9

* Define current probe (0V voltage source and current controlled voltage source)
vprobe n1 n1a ac 0V 0 dc 0
hprobe1 iprobe1 0 vprobe 1.0

* Define voltage probe (voltage controlled voltage source)
eprobe1 vprobe1 0 n1 n2 1.0

.ENDS SPICE_RLC
.end
```

Current probes are added to the SPICE circuit by adding a 0 V voltage source (**vprobe**) in series with other components in the branch where the current is to be probed. A current controlled voltage source (**hprobe1**) is then added between the global ground and a connection that is made available to the kernel as a probe (**iprobe1**). The kernel will load this pin with a 1 kΩ resistor and make the resulting voltage, that is directly proportional to the value of the current, available as the probe current (correctly scaled).

Voltage probes are added by simply adding a voltage controlled voltage source (**eprobe1**) between the global ground and a pin that is made available to the kernel as a probe. Similar to the current probe, the probe pin is loaded by the kernel with a 1 kΩ resistor so that the probed voltage can be made available in POSTFEKO.

CM Card

The CM card is used to couple Feko with the transmission line simulation programs CableMod or CRIPTE or the PCB tool PCBMod to calculate the coupling of electromagnetic fields into transmission lines. (The AC card is used for the case of radiation by these lines.)

In the **Request** tab, in the **Solution requests** group, click the **Cable fields (CM)** icon.

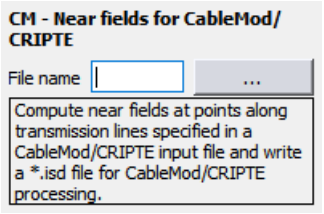


*Figure 870: The **CM - Near fields for CableMod/CRIPTE** dialog.*

Parameters:

File name	The name of the `.rsd` file created by CableMod, CRIPTE or PCBMod (enclosed in double quotation marks and starting at or after column 91)

The `.rsd` file contains geometry of the line. With the CM card Feko calculates the electric and magnetic nearfield at points along the line and write these to a `.isd` file for further processing by CableMod,CRIPTE or PCBMod. (The `.isd` file also contains additional data required by CableMod, CRIPTE or PCBMod, for example, the frequencies that were used during the solution.)

The complete geometry (without the transmission line) as well as the frequency and excitation(Ax cards) must be defined in Feko.

Related reference

AC Card

AX Cards

CO Card

This card specifies a dielectric or magnetic coating of wire segments or triangular surface elements.

In the **Home** tab, in the **Define** group, click the **Media** icon. From the drop-down list select the **Coating (CO)** icon.

> **Note:** The coating applies to all calculations following the CO card.

The CO card uses the material properties (label of the material) defined in the DI card. Therefore the DI card must be placed before the CO card. A layered medium is defined by referencing the labels of the materials to be used for the individual layers in the DL card.

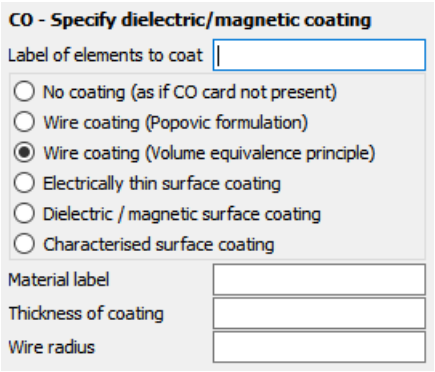


*Figure 871: The **CO - Specify dielectric/magnetic coating** dialog, set to **Wire coating (Volume equivalence principle)**.*

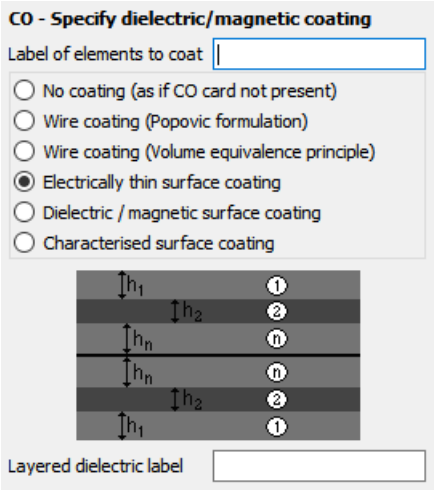


*Figure 872: The **CO - Specify dielectric/magnetic coating** dialog, set to **Electrically thin surface coating**.*

Parameters:

Label of elements to coat All segments or triangles with this label are coated.

No coating (as if CO card not present)	No coating present (as if the relevant label has no CO card). This is used to remove wire coatings from earlier solutions.
Wire coating (Popovic formulation)	In this case, the radius of the metallic core is changed internally to model the change in the capacitive loading of the wire and a corresponding inductive loading is added. The only restriction of this method is that the loss tangents of the wire coating and of the surrounding medium must be identical (for instance both media could be lossless).
Wire coating (volume equivalence principle)	This setting retains the radius of the metallic wire. The effect of the dielectric layer is accounted for by a volume polarisation current. The only restriction of this method is that the layer should not be magnetic in nature. This means that the relative permeability μ_r as well as the magnetic loss tangent $\tan\delta_u$ of the coating must be the same as those of the surrounding medium).
Electrically thin surface coating	This option adds multilayer dielectric/magnetic coatings to the surface triangles with the specified label. Different values for the permittivity and permeability of the layers are allowed, but the total coating must be electrically (that is relative to the wavelength in the coating) thin as well as geometrically thin (see the next item).
Dielectric / magnetic surface coating	This option adds electrically thick multilayer dielectric/magnetic coatings to the surface triangles with the specified label. This option requires that the total coating should be geometrically thin (relative to the triangle size, and as a result of the triangle size, also to the free space wavelength) as well as the curvature radius of the surface. This option may also be used when using the MoM / MLFMM and a single-layered, electrically thick, but geometrically thin coating is applied. • Only for closed structures with a PEC surface and the normal vector pointing towards the source(s). • Coating is applied to both sides of the PEC surface, since fields will be zero where there is no sources. • The accuracy of this option is higher when the propagation constant is high (high losses / permeability / permittivity).
Characterised surface coating	This option adds a characterised surface as a coating to the surface triangles with the specified label.
Material label	Label of the material (as specified in the DI card) that will be used as a coating.

Layered dielectric label	Label of the layered dielectric medium (as specified in the DL card) that will be used as a coating.
Thickness of coating	For surface coatings, the thickness h_n of each respective layer. For wire coatings this is the radius of the coating less the radius ϱ of the wire-core. This value is in metres and is scaled by the SF card.
Wire radius	Wire radius ϱ of the metallic wire, without layers, in metres (it is scaled by the SF card). This overrides the values specified in the IP card. This field is only applicable to wire coatings.
Reference direction	The reference direction of the characterised surface. The reference direction (U-Vector) is not required to be exactly in the plane of the face, since it is projected onto the face, but it should be approximately parallel to the face.

When using the Popovic formulation for wire coatings, the following restrictions apply:

- The loss tangent $\tan\delta$ of the layer (which is calculated from the conductivity σ and the relation $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$) has to be identical to the loss tangent of the surrounding medium.

  Note: The surrounding medium usually is free space, and it is specified with the EG card.

- Due to the change in the radius of the metallic core, no SK card should be active for the same label, otherwise the skin effect and/or the ohmic losses refers to the wire with changed radius.
- For pure dielectric layers (that is the relative permeability μ_r as well as the magnetic loss tangent $\tan\delta_\mu$ of the layer are equal to those of the surrounding medium) the option **Wire coating (Volume equivalence principle)** is recommended.

Note: For wire coatings, no surface triangles with the same label are allowed. Likewise for surface coatings, no segments with the same label are allowed.

If the option **Dielectric / magnetic surface coating** (the electrically thick coating) is used, it must remain consistent for the whole Feko run. (It cannot be enabled for one solution and disabled for the next.) It is, however, allowed to change the thickness and the medium parameters of the coating.

Related tasks
Applying a Coating to a Wire or Face (CADFEKO)
Related reference
DI Card
DL Card
SF Card

CR Card

This card specifies the orientation of a 3D anisotropic medium.

In the **Home** tab, in the **Define** group, click the **Media** icon. From the drop-down list select the **Orientation (CR)** icon.

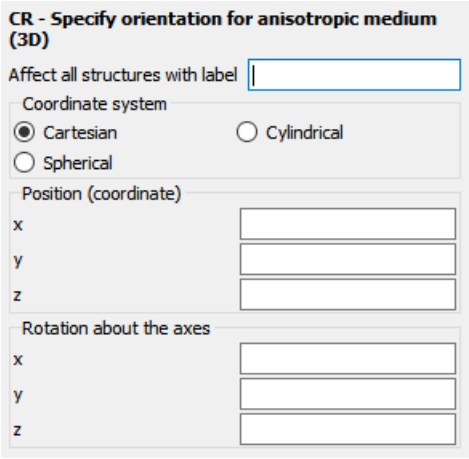


*Figure 873: The **CR - Specify orientation for anisotropic medium (3D)** dialog.*

Parameters:

Affect all structures with label	Regions with the specified labels will be affected.
Coordinate system	The coordinate system in which the 3D anisotropic medium orientation is defined.
Position (coordinate)	The X, Y and Z coordinates defining the origin of the workplane are entered in m. This value is affected by the scale factor of the SF card (if used).
Rotation about the axes	The angles with which the workplane is rotated around the X, Y and Z axes are entered.

Related tasks

Applying an Anisotropic Dielectric to a Region (CADFEKO)

CS Card

This card defines a cable path as well as the centre or reference location to which a cable cross section definition is applied.

On the **Solve/Run** tab, in the **Cables** group, click the **Cable path (CS)** icon.

A path my be specified using data points in the .pre file or loading a cable path from a NASTRAN file.

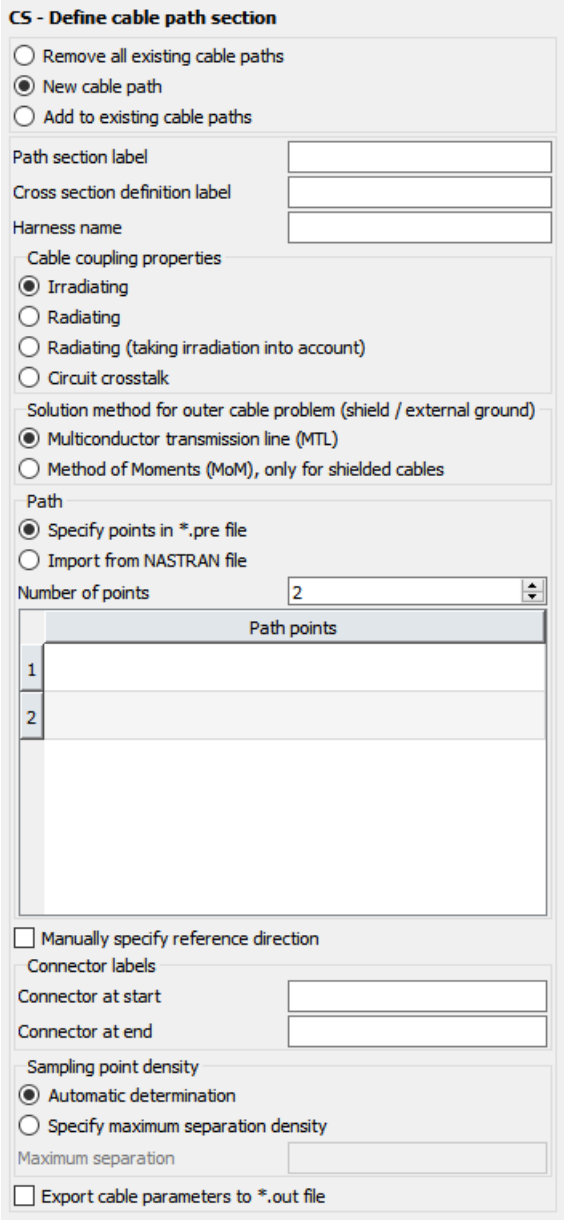


*Figure 874: The **CS - Define cable path section** dialog.*

Parameters:

Remove all existing cable paths If checked, all previously defined cable paths are removed. All the other input parameters are ignored.

New cable path	Defines a new cable path. All previously defined cable paths are replaced.	
Add to existing cable paths	An additional cable path is defined (that is the previously added cable paths will be kept).	
Path section label	The label of the path section.	
Cross section definition label	The label of the defined cross section to be applied to the path section.	
Harness name	The label of the cable harness containing the cable path.	
Cable coupling properties	**Irradiating**	When this option is selected, only the effect of external fields coupling into a cable will be considered.
	Radiating	When this option is selected, the effect of the currents radiating from the cables will be considered.
	Radiating (taking irradiation into account)	When this option is selected, the combined effect of external fields coupling into a cable as well as the effect of currents radiating from the cable will be considered.
	Circuit crosstalk	When this option is selected, the intra coupling between cables are considered (no external effects from fields or currents).
Solution method for outer cable problem (shield / external ground)	**Multiconductor transmission line (MTL)**	When this option is selected, the model will be solved with the multiconductor transmission line which is also hybridised with MoM or the MLFMM. The cable path should be within $\frac{\lambda}{5}$ of the conducting surface.
	Method of moments (MoM), only for shielded cables	When this option is selected, the model will be solved with the MoM/MTL solver. Any arbitrary cable path may be defined.
Path	**Specify points in *.pre file**	The points defining the cable path are specified in the .pre file. These points must have been defined previously with

a DP card. The number of points defining the cable path section in the .pre file is set by the **Number of points** option. The **Path points** (as defined with a DP card) are specified to define the cable path section.

Import from NASTRAN file

The NASTRAN **File name** is required to import the cable path section. The **NASTRAN segment property ID** of the segments is also required to be able to import the cable path section.

Manually specify reference direction

X	The X coordinate of the reference direction.
Y	The Y coordinate of the reference direction.
Z	The Z coordinate of cable the reference direction.
Twist angle [degrees]	The angle in degrees of the cable reference direction at the end of the cable path.

Connector labels

The start and end points of a cable path section are uniquely identified using the **Connector at start** and **Connector at end** labels. Currently cable path labels can be joined if all three of the following conditions are satisfied:

1. The labels share a node.
2. The labels use the same cable connector.
3. The labels use the same cable cross section definition.

Therefore after combining/reducing the number of cable path sections, each cable path consists of a section of cable with a unique cross section.

Note: There is no support for splitting of cable sections or combining of cable sections using loads.

Connection to device ground

Define if a cable path section's start and end points are connected to the local device ground.

Sampling point density

The cable path section will be subdivided into small segments for the computation of the induced currents and voltages. The electric

and magnetic field strengths will be evaluated at each segment's centroid. The setting **Automatic determination** will choose the segment length automatically (which should be adequate for most cases). If the maximum separation distance is specified, then this value will override the automatic mechanism. This manual value will be scaled by any active SF card.

> **Note:** This setting influences the accuracy and computation time of the computed result.

Export cable parameters to *.out file

When this item is checked the cable parameters such as inductance/capacitance matrices and transfer impedance/ admittances are exported to the .out file.

Related tasks

Defining a Cable Path (CADFEKO)

Related reference

CA Card

DP Card

SF Card

DA Card

This card controls the export of data to additional ASCII files. For example the currents can be exported to the .out file or S-parameters can be exported to a Touchstone file.

On the **Request** tab, in the **Output control** group, click the **Data output (DA)** icon.

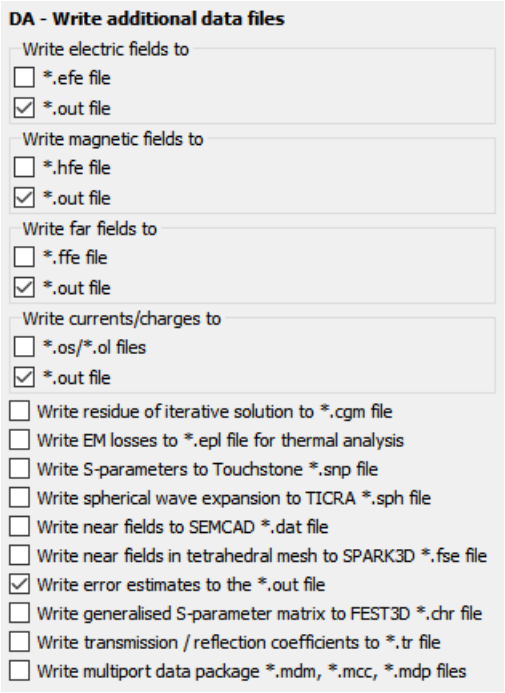


*Figure 875: The **DA- Write additional data files** dialog.*

The card allows to switch the export of data on or off. It only affects those cards that follow the DA card, such as the SP card for S-parameters. By default no such export files are created. The .out file is always created and the DA card is set, by default, to write all results to this file.

> **Warning:** For large datasets (such as surface currents at many frequency points) the .out file can become very large. The output of these results can then be deactivated with the DA card.

In order to display results in POSTFEKO only the binary output file (.bof file) is required. If the output of results to the .out file was deactivated, a header will still be written to this file to identify the type of computations that were requested.

Parameters:

Write electric fields to	The electric fields can be exported to a `.efe` file. The output of this result type to the .out file can also be activated/deactivated.
Write magnetic fields to	The magnetic fields can be exported to a `.hfe` file. The output of this result type to the .out file can also be activated/deactivated.

Write far fields to	The far fields can be exported to a `.ffe` file. The output of this result type to the .out file can also be activated/deactivated.
Write currents/charges to	The currents can be exported to a `.os`/`ol` file. The output of this result type to the .out file can also be activated/deactivated.
Write residue of iterative solution to *.cgm file	The residue of the iterative algorithm used to solve the matrix equation is stored in a `.cgm` file.
Write EM losses to *.epl file for thermal analysis	An element power loss `.epl` file is exported which contains the electromagnetic losses for each element in the model. The `.epl` file can be used in other thermal simulations with the NASTRAN `.nas` file and label mapping `.map` file.
Write S-parameters to Touchstone *.snp file	The S-parameters (SP card) are written to a file in Touchstone `.snp` format (v1.0). For each configuration containing an S-parameter data request, a separate Touchstone file is created. The file name will be of the form `<FEKO_base_filename>_<requestname>(k).snp` where n is the number of ports and k is an integer counter. The counter (k) is added to distinguish between the results of multiple requests with the same name and the same number of ports. For a one-port request without an SP card, the name of the AX card (source) is used.
Write spherical wave expansion to TICRA *.sph file	A spherical wave expansion of the far field as computed by Feko is exported to an SWE file (extension `.sph`) which can be imported into GRASP from TICRA (code for reflector antenna modelling). **Note:** The FF card must follow the DA card with spherical mode coefficients requested in the FF card.
Write near fields to SEMCAD *.dat file	The near fields can be exported to a SEMCAD `.dat` file. The output of this result type to the SEMCAD `.dat` file can be activated/deactivated.
Write near fields in tetrahedral mesh to SPARK3D *.fse file	The near fields calculated at the vertices and edge mid-points inside a tetrahedral mesh can be exported to a `.fse` file. The output of the result type to the `.fse` file can activated/ deactivated.
Write error estimates to the *.out file	The error estimates can be exported to the .out file.
Write generalised S-parameter matrix to FEST3D *.chr file	The generalised S-parameter matrix (GSM) for waveguide ports are exported to a FEST3D `.chr` file.

Write transmission / reflection coefficients to *.tr file	The transmission and reflection coefficients can be exported to a `.tr` file.
Write multiport data package to *.mdm, *.mcc, *.mdp files	The files required to do a multiport calculation are exported. The list of files that are packaged is written to the multiport data manifest (`.mdm`) file. A template input file containing the information to do a multiport calculation is written to the multiport combinations configuration (`.mcc`) file. The S-parameters, field and current requests are packaged in the multiport data package (`.mdp`) file. **Note:** This setting will also override S-parameter associated request DA card settings. For example, if we have an far field request associated with the S-parameter request, then the expectation is that the `.ffe` file(s) are generated as part of the multiport processor files, irrespective of its associated DA card setting.

More than one DA card is allowed in one input file. Therefore using the following sequence of control cards, with the appropriate options, will cause only specific blocks of the same data type to be exported to the data file.

```
DA ... ** Write near fields activated
FE ...
DA ... ** Write near fields deactivated
FE ...
```

With this sequence, the electric fields calculated with the first FE card can be written to the `.efe` file, but not those of the second FE card.

Related concepts
Advanced Settings for Far Field Requests (CADFEKO)
Advanced Settings for Near Field Requests (CADFEKO)
Related tasks
Requesting an Error Estimation (CADFEKO)
Adding an S-Parameter Configuration (CADFEKO)
Related reference
File Format History (.EFE, .HFE, .FFE, .OL, .OS and .TR)
General File Format (.EFE, .HFE, .FFE, .OL, .OS and .TR)

DI Card

This card can be used to define the frequency dependent or independent material characteristics of a dielectric medium, metallic medium or an impedance sheet.

In the **Home** tab, in the **Define** group, click the **Media** icon. From the drop-down list select the **Dielectric (DI)** icon.

The DI card is used for the MoM/MLFMM when using the surface current or volume current methods, or also for the FEM.

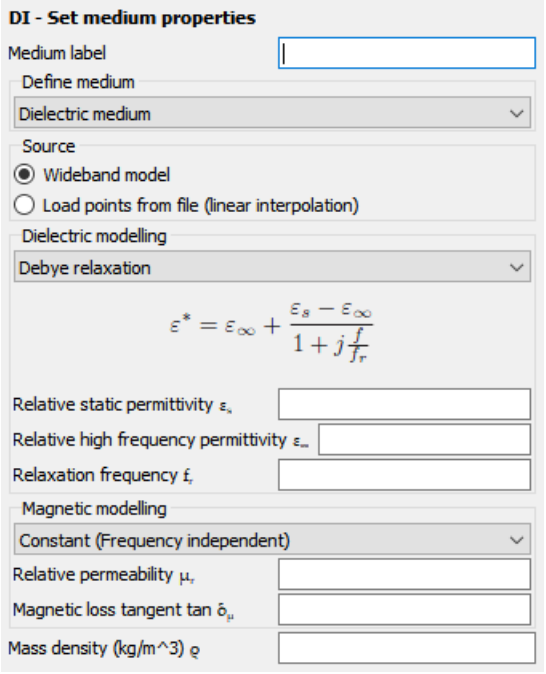


*Figure 876: The **DI - Set medium properties** dialog.*

Parameters:

Medium label	Label of the material to be defined.
Define dielectric medium	Define a dielectric medium by specifying a wideband model or loading points from a file and using linear interpolation. The following wideband models are available: **Constant (Frequency independent)**, **Debye relaxation**, **Cole-Cole**, **Havriliak-Negami** and **Djordjevic-Sarkar (Wideband Debye)**.
Define metallic medium	Define a metallic medium by specifying the frequency independent parameters or loading points from a file.
Define impedance sheet	Define an impedance sheet by specifying the frequency independent real/imaginary part of the impedance or loading points from a file.

Define anisotropic medium	Define an anisotropic medium by means of specifying either the **Full tensor**, **Diagonal tensor**, **Complex tensor** or **Polder tensor (ferrites)**.
Define characterised surface medium	Define a characterised surface medium by loading the transmission and reflection coefficients from a `.tr file`.(as specified in the DA card)
Source	Define the medium by means of a wideband model or loading points from a file.

Note: The same functionality for defining a dielectric, metallic or impedance sheet is also available in CADFEKO.

Related tasks
Creating a Dielectric (CADFEKO)

Dielectric Modelling

This option specifies the model (method) used to define a dielectric medium.

Frequency independent

The properties of the dielectric medium are specified as frequency independent.

Relative permittivity	Relative permittivity ε_r of the medium.
Dielectric loss tangent	Dielectric loss tangent $\tan\delta$ of the medium (this is an alternative way to specify the conductivity σ — the two loss terms are related by $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$ and have different frequency behaviour).
Conductivity	Conductivity σ in $\frac{1}{\Omega m}$ of the medium.

Debye Relaxation

The relaxation characteristics of gasses and fluids at microwave frequencies are described by the Debye model. It has been derived for freely rotating spherical polar molecules in a predominantly non-polar background.

Relative static permittivity	Relative permittivity ε_s of the medium.
High frequency dielectric constant	High frequency dielectric constant ε_∞ of the medium.
Relaxation frequency	The relaxation frequency f_r of the medium.

Cole-Cole

The model is similar to the Debye model, but uses one additional parameter to describe the material.

Relative static permittivity	Relative static permittivity ε_s of the medium.
High frequency dielectric constant	High frequency dielectric constant ε_∞ of the medium.
Relaxation frequency	The relaxation frequency f_r of the medium
Attenuation factor	Attenuation factor a of the medium.

Havriliak-Negami

This is a more general model and should be able to successfully model liquids, solids and semi-solids.

Relative static permittivity	Relative static permittivity ε_s of the medium.
High frequency dielectric constant	High frequency dielectric constant ε_∞ of the medium.
Relaxation frequency	The relaxation frequency f_r of the medium
Attenuation factor	Attenuation factor a of the medium.
Phase factor	Phase factor β of the medium.

Djordjevic-Sarkar

This is a particularly well suited broadband model for composite dielectrics.

Variation of relative permittivity	Variation of the relative permittivity $\triangle\varepsilon$ of the medium.
Relative high frequency permittivity	Relative high frequency permittivity ε_∞ of the medium.
Conductivity	Conductivity σ in $\frac{1}{\Omega m}$ of the medium.
Lower limit of angular frequency	The lower limit of the angular frequency for the medium, ω_1 of the medium.
Upper limit of angular frequency	The upper limit of the angular frequency for the medium, ω_2 of the medium.

Specify points in the *.pre file (linear interpolation)

This is a particularly well suited broadband model for composite dielectrics.

Frequency	The frequency for the specific data point.

Relative permittivity	Relative permittivity ε_r of the medium at a specific frequency.
Dielectric loss tangent	Dielectric loss tangent $\tan\delta$ of the medium (this is alternative way to specify the conductivity σ — the two loss terms are related by $\tan\delta = \frac{\sigma}{\omega\varepsilon_r\varepsilon_0}$ and have different frequency behaviour).
Conductivity	Conductivity σ in $\frac{1}{\Omega m}$ of the medium.

Related concepts
Dielectric Media Formulations

Magnetic Modelling

This option specifies the model (method) used to define a dielectric or metallic medium.

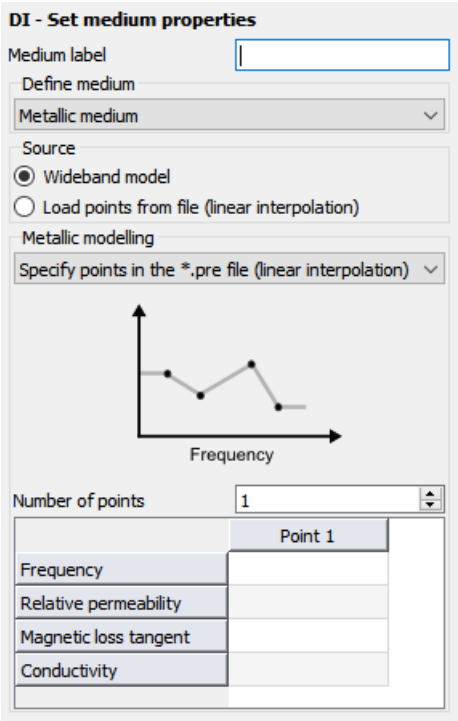


*Figure 877: The **DI - Set medium properties** dialog, set to **Metallic medium**.*

Constant (Frequency independent)

Relative permeability	Relative permittivity μ_r of the medium.
Magnetic loss tangent	Magnetic loss tangent $\tan\delta_u$ of the medium (the complex permeability is then given by $\mu = \mu_0\mu_r(1 - j\tan\delta_\mu)$).
Conductivity	Conductivity σ in $\frac{1}{\Omega m}$ of the medium.

Specify points in the *.pre file (linear interpolation)

Frequency	The frequency for the specific data point.
Relative permittivity	Relative permittivity μ_r of the medium at a specific frequency.
Magnetic loss tangent	Magnetic loss tangent $\tan\delta_u$ of the medium (the complex permeability is then given by $\mu=\mu_0\mu_r(1-j\tan\delta_\mu)$).
Conductivity	Conductivity σ in $\frac{1}{\Omega m}$ of the medium at a specific frequency.

Related concepts
Dielectric Media Formulations

Surface Impedance (Ohm)

This option specifies a user defined complex surface impedance Z_s.

> **Warning:** The impedance boundary condition for the MoM (also then for MLFMM and so forth) has certain limitations regarding the range of validity. Feko will apply the user defined surface impedance as is without regard for the specific configuration (frequency, radius of curvature of the structure and so forth).

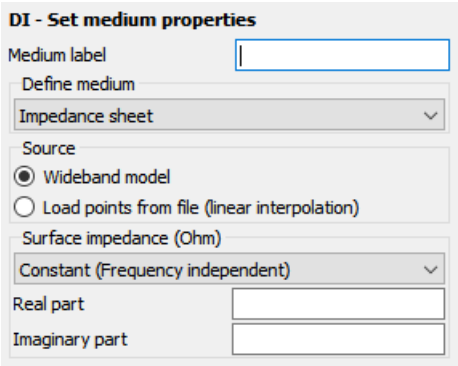


*Figure 878: The **DI - Set medium properties** dialog, set to **Impedance sheet**.*

For surfaces the unit of Z_s is Ω . For wire structures, the value used by Feko is in units of $\frac{\Omega}{\mathrm{m}}$ and results from the surface impedance expression by dividing it by $\pi\varrho$ where ϱ represents the wire radius.

The properties for the surface impedance are as follows:

Real part	The real part of the surface impedance Z_s in Ω (for triangles) or in $\frac{\Omega}{\mathrm{m}}$ for wires.
Imaginary part	The imaginary part of the surface impedance Z_s in Ω (for triangles) or in $\frac{\Omega}{\mathrm{m}}$ for wires.

Load Points From File (Linear Interpolation)

With this option the properties of dielectrics, metals and impedance sheets can be imported from file.

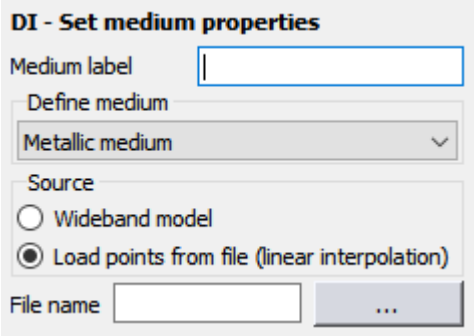


*Figure 879: The **DI - Set medium properties** dialog, set to **Load points from file**.*

A description of the XML format used to describe the medium properties, is given below.

When importing a medium from file, the following keywords are used:

Dielectric medium	freq, permittivity, diel_loss_tangent, mag_loss_tangent, conductivity and permeability.
Metallic medium	freq, conductivity, permeability and mag_loss_tangent.
Impedance sheet	freq, surf_imp_re and sur_imp_im.

For demonstrative purposes, the keywords `val_A`, `val_B` and `val_C` are used in the demo XML file given below as the same format is also applicable when defining a dielectric, metallic or impedance sheet.

```
<?xml version="1.0" encoding="UTF-8"?>
<materialDB creator="Altair Feko" date="2010-07-01" version="1.0">
   <material name="mediumA"                    val_B="7.0"     val_C="9.0"   >
       <dataPoint freq="2.0"   val_A="2.0"    val_B="6.0"                   />
       <dataPoint freq="3.0"   val_A="3.0"                                  />
       <dataPoint freq="4.0"   val_A="1.0"                                  />
       <dataPoint freq="5.0"                  val_B="5.0"                   />
       <dataPoint freq="6.0"   val_A="1.0"                                  />
       <dataPoint freq="8.0"                  val_B="6.0"                   />
       <dataPoint freq="9.0"   val_A="4.0"                                  />
   </material>
</materialDB>
```

In the following line, the static values for material with name `mediumA`, are defined:

```
<material name="mediumA"                    val_B="7.0"     val_C="9.0"   >
```

Next the frequency dependent data points are defined:

```
 <dataPoint freq="2.0"   val_A="2.0"    val_B="6.0"                   />
```

The internal XML parser then fills in the missing values in the frequency dependent data points with static values (if they were defined). If only static data points are defined and no frequency dependent

materials data points are found, then one data point will be generated with freq="0.0" by the XML parser and filled with the static values.

Therefore the above XML file will then be parsed as if it was specified by the user as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<materialDB creator="Altair Feko" date="2010-07-01" version="1.0">
   <material name="mediumA"                                          >
       <dataPoint freq="2.0"   val_A="2.0"    val_B="6.0"      val_C="9.0"  />
       <dataPoint freq="3.0"   val_A="3.0"    val_B="7.0"      val_C="9.0"  />
       <dataPoint freq="4.0"   val_A="1.0"    val_B="7.0"      val_C="9.0"  />
       <dataPoint freq="5.0"                  val_B="5.0"      val_C="9.0"  />
       <dataPoint freq="6.0"   val_A="1.0"    val_B="7.0"      val_C="9.0"  />
       <dataPoint freq="8.0"                  val_B="6.0"      val_C="9.0"  />
       <dataPoint freq="9.0"   val_A="4.0"    val_B="7.0"      val_C="9.0"  />
   </material>
</materialDB>
```

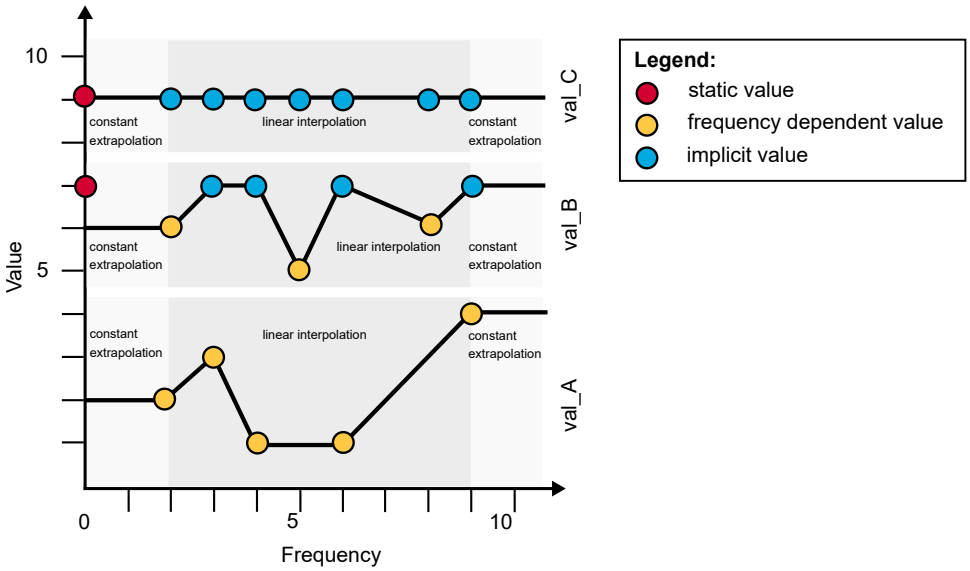


Figure 880: A graphical illustration showing the result of the parsed XML.

The **Mass density** is only used for specific absorption rate (SAR) calculations, but it must be specified and must have a value larger than 0.

3D Anisotropic Medium

This option specifies an anisotropic (3D) medium represented by a simple diagonal tensor, a nine element (full) tensor, complex tensor or a Polder tensor (for ferrites)

Diagonal tensor

Define the diagonal permittivity and permeability tensors by defining up to three dielectrics constituting the medium properties along the UU, VV and NN axes. If no linear dependencies exist between two axes, enter `0`. A value of `0` indicates empty or zero. Use the text, `Free space`, to indicate the free space medium.

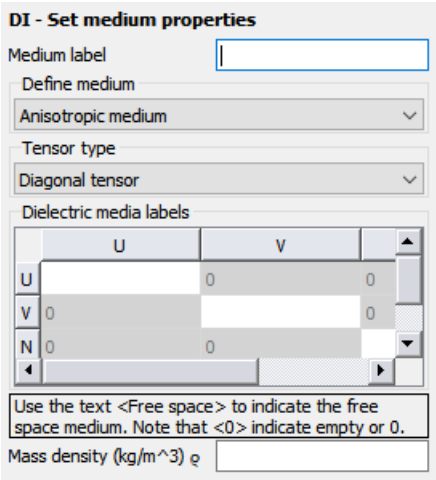


*Figure 881: The **DI - Set medium properties** dialog, for **Anisotropic medium**, set to **Diagonal tensor**.*

Full tensor

Define the permittivity and permeability tensors by defining up to nine dielectrics constituting the medium properties along the UU, UV, UN, VU, VV, VN, NU, NV and NN axes. If no linear dependencies exist between two axes, enter `0`. A value of `0` indicates empty or zero. Use the text, `Free space`, to indicate the free space medium.

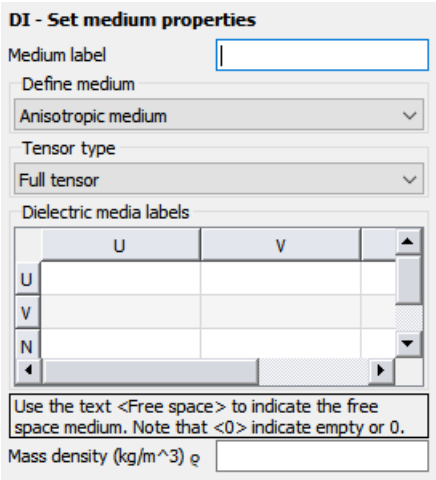


*Figure 882: The **DI - Set medium properties** dialog, for **Anisotropic medium**, set to **Full tensor**.*

Complex tensor

Define the permittivity and permeability tensors by using complex values. An entry in the tensor may be a complex number, pure real number or a pure imaginary number, but not 0.

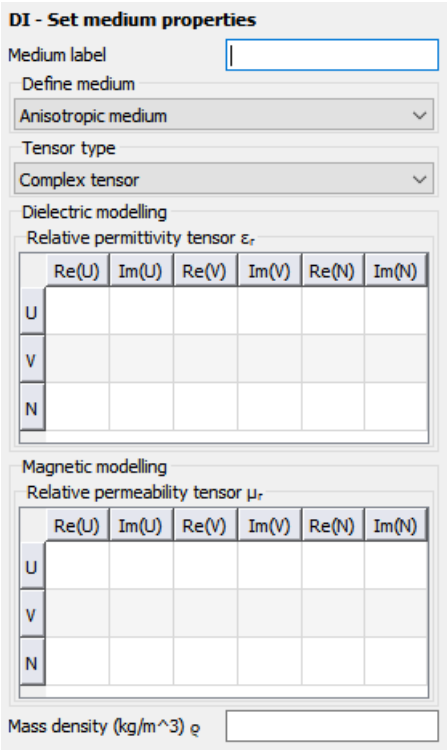


*Figure 883: The **DI - Set medium properties** dialog, for **Anisotropic medium**, set to **Complex tensor**.*

Polder tensor (ferrites)

Define a ferromagnetic material.

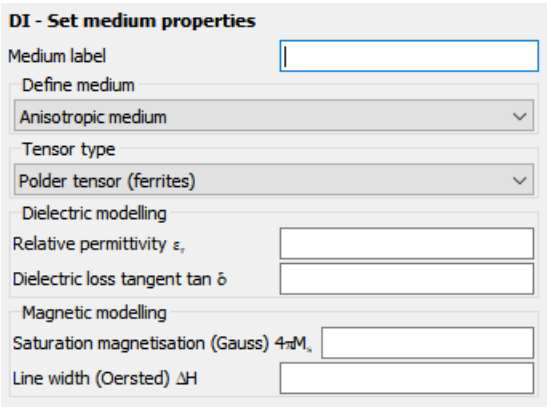


*Figure 884: The **DI - Set medium properties** dialog, for **Anisotropic medium**, set to **Polder tensor**.*

Relative permittivity	Relative permittivity ε_r of the medium.
Dielectric loss tangent	Dielectric loss tangent $\tan\delta$ of the ferrite.
Saturation magnetisation (Gauss)	Saturation magnetisation $4\pi M_S$ of the ferrite.

Line width (Oersted)	Line width $\triangle H$ of the ferrite.

Related concepts
Anisotropic Media Formulations

Characterised Surface Medium

Define a characterised surface medium by importing the transmission and reflection coefficients from a `.tr` file.

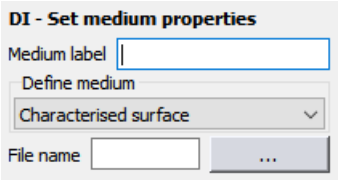


*Figure 885: The **DI - Set medium properties** dialog, for **Characterised surface medium**.*

Parameters

File name	The name of the `.tr` file used to define the characterised surface. See the DA card for more detail on the `.tr` file format.

DL Card

This card is used to define an isotropic or anisotropic layered medium by specifying the label of the material to be used for each layer.

In the **Home** tab, in the **Define** group, click the **Media** icon. From the drop-down list select the **Layered dielectric (DL)** icon.

Parameters:

Layered medium type	**Select either of the following two options:**	Isotropic layered medium Anisotropic layered medium

Related tasks

Creating an Isotropic Layered Dielectric (CADFEKO)

Isotropic Layered Medium

This option defines an isotropic layered medium by specifying the label of the material to be used for each layer.

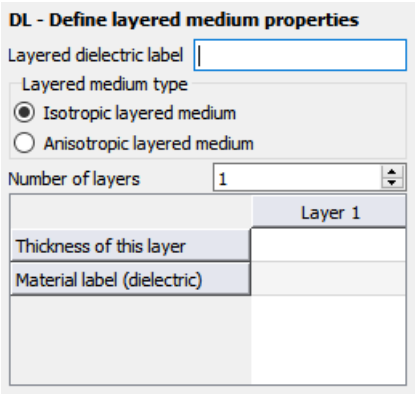


*Figure 886: The **DL - Define layered medium properties** dialog set to **Isotropic layered medium**.*

Parameters:

Number of layers	The number of isotropic layers defined for the layered medium.
Thickness of this layer	The thickness of the current layer in metres (if an SF card is present, this is always scaled).
Material label (dielectric)	The label of the material to be used for the current layer (as defined in the DI card).

Anisotropic Layered Medium

This option defines an anisotropic layered medium by specifying the label of the material to be used for each layer.

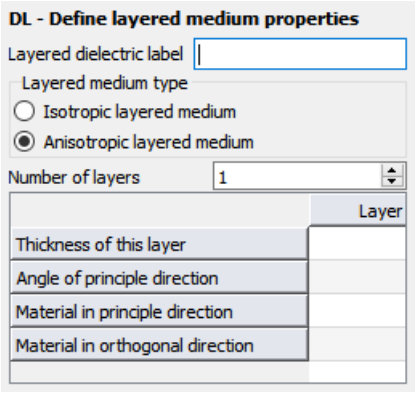


*Figure 887: The **DL - Define layered medium properties** dialog set to **Anisotropic layered medium**.*

Parameters:

Number of layers	The number of anisotropic layers defined for the layered medium.
Thickness of this layer	The thickness of the current layer in m (if an SF card is present, this is always scaled).
Angle of principal direction	The angle (in degrees) from which the principal direction is obtained.
Material in principal direction	The material label of the material to be used in the principal direction.
Material in orthogonal direction	The material label of the material to be used in the orthogonal direction.

EE Card

The EE card requests an a-posteriori error indicator whereby Feko can test the solution against an unconstrained physical test. The result is to give an indication of the region where local mesh refinement should be considered.

On the **Request** tab, in the **Solution requests** group, click the **Error estimation (EE)** icon.

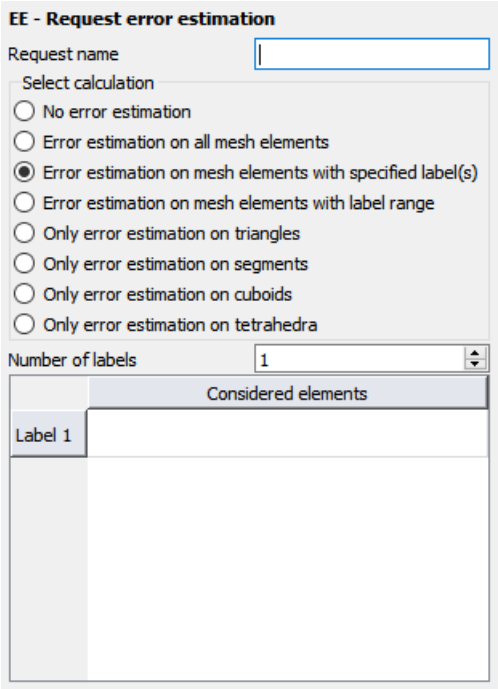


*Figure 888: The **EE - Request error estimation** dialog.*

Parameters:

Request name	The name of the request.
No error estimation	No error estimation output.
Error estimation on all mesh elements	Output error estimation on all mesh elements.
Error estimation on mesh elements with specified label(s)	Output error estimation on mesh elements with labels specified by the user.
Error estimation on mesh elements with label range	Output error estimation in the label range starting on **Start label** and ending on **End label**.
Only error estimation on triangles	Output only error estimation on triangles.
Only error estimation on segments	Output only error estimation on segments.

Only error estimation on cuboids	Output only error estimation on cuboids.
Only error estimation on tetrahedra	Output only error estimation on tetrahedra.

Related tasks
Requesting an Error Estimation (CADFEKO)

EN Card

The EN card indicates the end of the input file. It is essential and has no parameters.

On the **Home** tab, in the **Structure** group, click the  **End model (EN)** icon.

FD Card

This card sets the solver settings for the finite difference time domain (FDTD) solver.

On the **Solve/Run** tab, in the **Solution settings** group, click the **FDTD settings (FD)** icon.

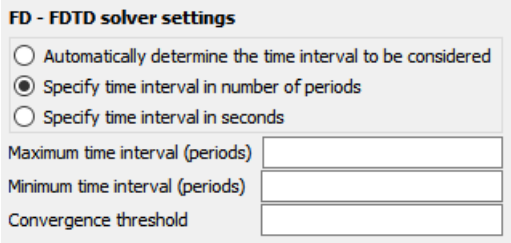


*Figure 889: The **FD - FDTD solver settings** dialog.*

Parameters:

Automatically determine the time interval to be considered
The time interval is determined automatically by the FDTD solver based on the time signals used by the configuration sources, the size of the computational domain and the material properties. An estimation is made for how much time is required for a pulse to propagate through the whole domain.

Specify the time interval in number of periods
The maximum and/or minimum time interval specified in sinusoidal periods. A period is defined as $\frac{1}{f_{centre}}$, where f_{centre} is the average between the upper and lower frequencies in the requested band.

Specify time interval in seconds
The absolute maximum and/or minimum time interval specified in seconds.

Convergence threshold
Convergence threshold for the FDTD solver. The simulation will be terminated if the threshold has been reached and the simulation time is larger or equal to the minimum simulation time (specified or automatically determined).

Related reference
FDTD Frequency Settings (CADFEKO)

FE Card

This card controls the calculation of near fields.

On the **Request** tab, in the **Solution requests** group, click the **Near field (FE)** icon.

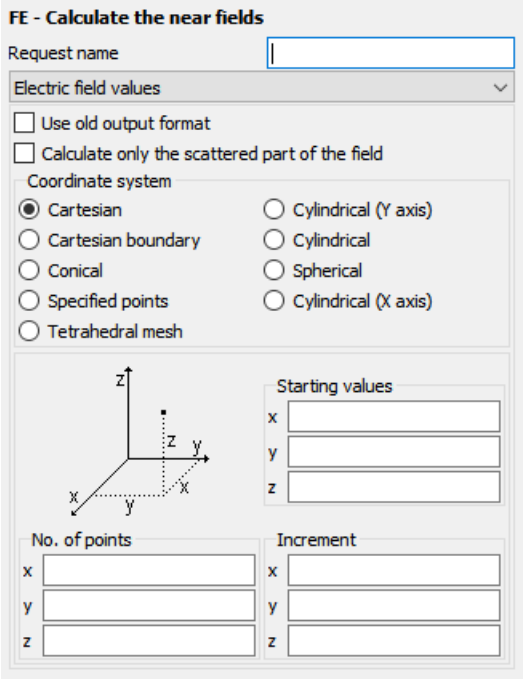


*Figure 890: The **FE - Calculate the near fields** dialog.*

Parameters:

Request name — The name of the request.

Select what to calculate

No field calculation	Field is not calculated.
Electric field values	Calculate the electric field, **E**.
Magnetic field values	Calculate the magnetic field, **H**.
Both electric field and magnetic field values	Calculate both electric and magnetic fields.
Electric field and SAR values (in cuboids)	Calculate the electric field and SAR values in the dielectric volume elements. For this option, no other parameters are required.
Electric vector potential	Calculate the electric vector potential, **F**.

Electric scalar potential	Calculate the electric scalar potential, φ.
Gradient of the scalar electric potential	Calculate the gradient of the scalar electric potential, $\nabla\varphi$.
Magnetic vector potential	Calculate the magnetic vector potential, **A**.
Magnetic scalar potential	Calculate the magnetic scalar potential, ψ.
Gradient of the scalar magnetic potential	Calculate the gradient of the scalar magnetic potential, $\nabla\psi$.

Use old output format

If this item is checked, the old format of the near field is used in the output file. This should only be used for compatibility with third party post processors. (POSTFEKO cannot extract SAR values from near fields in this format).

Calculate only the scattered part of the field

When this item is checked only the scattered part of the field/ potential is computed and written to the output file. Otherwise the total field/potential, that is the sum of the scattered and source contributions will be computed.

> **Note:** Depending on the formulation used in Feko the region where an impressed source is regarded as the incident field could differ.

For example when using the surface equivalence principle in the MoM to model dielectric bodies each source will act as the incident field only in that medium where the source is located. To elaborate, consider an SEP solution of a Hertzian dipole inside a dielectric body, A. In a nearby dielectric body, B (different label but could have the same or different medium properties), this source would not be considered an impressed source.

Coordinate system

In this group, the coordinate system for the calculation of the requested fields is specified.

By selecting **Cartesian**, **Cartesian boundary**, **Cylindrical**, **Spherical**, **Cylindrical (X axis)**, **Cylindrical (Y axis)** and **Conical**, additional groups will be shown for **Starting values**, **Increment** and **No. of points**.

If **Specified points** is selected, the **No. of field points** and the **Coordinates** of each near field point are required.

If **Tetrahedral mesh** is selected, the near field is calculated at the vertices and edge mid-points of the tetrahedra. No additional information is required.

For the Cartesian boundary option, to include a surface in the Cartesian boundary near field request, select the corresponding check box. Select the **Sample on edges** check box to enable sampling on the edges of the Cartesian boundary.

Note: All coordinates are in metres and all angles in degrees.

Scaling with the SF card is only applicable when the option **Modify all dimension related values** is checked in the SF card (default behaviour and highly recommended). In this case coordinates must be in metres after scaling.

Potentials cannot be computed with the FE card in the following cases:

- The UTD solver is used.
- The PO solver is used.
- The Green's functions for layered spheres or multi-layered planar media is used (but the free space Green's function is supported).

If the total potentials are requested, the potentials for the sources are added. These are not available for a plane wave (A0 card) or an impressed radiation pattern (AR card) and Feko will give an error in this regard. For a magnetic dipole (A6 card) the electric ring current model yields **A** and the magnetic current yields **F** and $\nabla\psi$. All the other potentials mentioned in **First drop-down list** are zero.

If output to `.efe` and/or `.hfe` files is requested with the DA card, then **A** and $\nabla\varphi$ are written to the `.efe` file, while **F** and $\nabla\psi$ are written to the `.hfe` file.

If a ground plane is used, calculation of the near fields in/below the ground plane is not possible. Requested points in the area z < 0 will be ignored.

The coordinates may be offset with the OF card. The OF card allows the near field on the surface of a sphere to be calculated where the centre of the sphere is not located at the origin of the coordinate system.

Related tasks
Requesting a Near Field (CADFEKO)
Related reference
A0 Card
A6 Card
AR Card
OF Card

Near Field Coordinate Systems (FE card)

The coordinate systems for the FE card have specific definitions/conventions. Use the appropriate coordinate system for the application.

Cartesian coordinates x, y, z

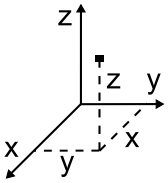


Figure 891: Field calculation in the Cartesian coordinate system.

Observation point:

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{187}$$

Unit vectors of the coordinate system:

$$\hat{\mathbf{x}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{\mathbf{y}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \hat{\mathbf{z}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{188}$$

Cylindrical coordinates around Z axis ρ, φ, z

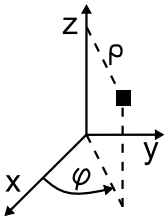


Figure 892: Field calculation in the Cylindrical coordinate system.

Observation point:

$$\mathbf{r} = \begin{bmatrix} \rho\cos\varphi \\ \rho\sin\varphi \\ z \end{bmatrix} \tag{189}$$

Unit vectors of the coordinate system:

$$\hat{\boldsymbol{\rho}} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{bmatrix} \qquad \hat{\boldsymbol{\varphi}} = \begin{bmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{bmatrix} \qquad \hat{\mathbf{z}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{190}$$

Spherical coordinates r, ϑ, φ

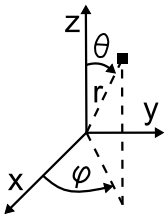


Figure 893: Field calculation in the spherical coordinate system.

Observation point:

$$\mathbf{r} = \begin{bmatrix} r\sin\vartheta\cos\varphi \\ r\sin\vartheta\sin\varphi \\ r\cos\vartheta \end{bmatrix} \tag{191}$$

Unit vectors of the coordinate system:

$$\hat{\mathbf{r}} = \begin{bmatrix} \sin\vartheta\cos\varphi \\ \sin\vartheta\sin\varphi \\ \cos\vartheta \end{bmatrix} \quad \hat{\boldsymbol{\vartheta}} = \begin{bmatrix} \cos\vartheta\cos\varphi \\ \cos\vartheta\sin\varphi \\ -\sin\vartheta \end{bmatrix} \quad \hat{\boldsymbol{\varphi}} = \begin{bmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{bmatrix} \tag{192}$$

Cylindrical coordinates around X axis r, φ, x

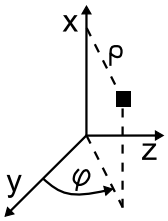


Figure 894: Field calculation in the Cylindrical coordinate system around the X axis.

Observation point:

$$\mathbf{r} = \begin{bmatrix} x \\ \rho\cos\varphi \\ \rho\sin\varphi \end{bmatrix} \tag{193}$$

Unit vectors of the coordinate system:

$$\hat{\boldsymbol{\rho}} = \begin{bmatrix} 0 \\ \cos\varphi \\ \sin\varphi \end{bmatrix} \quad \hat{\boldsymbol{\varphi}} = \begin{bmatrix} 0 \\ -\sin\varphi \\ \cos\varphi \end{bmatrix} \quad \hat{\mathbf{x}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{194}$$

Cylindrical coordinates around Y axis r, φ, y

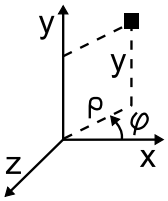


Figure 895: Field calculation in the Cylindrical coordinate system around the Y axis.

Observation point:

$$\mathbf{r}=\begin{bmatrix}\rho\cos\varphi\\ y\\ -\rho\sin\varphi\end{bmatrix} \tag{195}$$

Unit vectors of the coordinate system:

$$\hat{\boldsymbol{\rho}}=\begin{bmatrix}\cos\varphi\\ 0\\ -\sin\varphi\end{bmatrix}\qquad \hat{\boldsymbol{\varphi}}=\begin{bmatrix}-\sin\varphi\\ 0\\ -\cos\varphi\end{bmatrix}\qquad \hat{\mathbf{y}}=\begin{bmatrix}0\\ 1\\ 0\end{bmatrix} \tag{196}$$

Conical coordinates around the Z axis φ, z

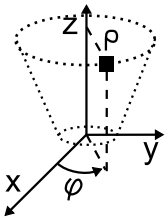


Figure 896: Field calculation in the Conical coordinate system.

This option is similar to the field calculation in cylindrical coordinates around the Z axis, where the radius r changes with the height z:

$$r(z)=r_0+\frac{\Delta r}{\Delta z}\cdot(z-z_0) \tag{197}$$

where z is within the range $z_0 \ldots z_0+n_z\cdot\Delta z$.

Observation point:

$$\mathbf{r}=\begin{bmatrix}\left(r_0+\frac{\Delta r}{\Delta z}\cdot(z-z_0)\right)\cdot\cos\varphi\\ \left(r_0+\frac{\Delta r}{\Delta z}\cdot(z-z_0)\right)\cdot\sin\varphi\\ z\end{bmatrix} \tag{198}$$

Unit vectors of the coordinate system:

$$\hat{\mathbf{r}}=\begin{bmatrix}\cos\varphi\\ \sin\varphi\\ 0\end{bmatrix}\qquad \hat{\boldsymbol{\varphi}}=\begin{bmatrix}-\sin\varphi\\ \cos\varphi\\ 0\end{bmatrix}\qquad \hat{\mathbf{z}}=\begin{bmatrix}0\\ 0\\ 1\end{bmatrix} \tag{199}$$

FF Card

This card controls the calculation of the far fields in spherical or Cartesian coordinates.

On the **Request** tab, in the **Solution requests** group, click the **Far field (FF)** icon.

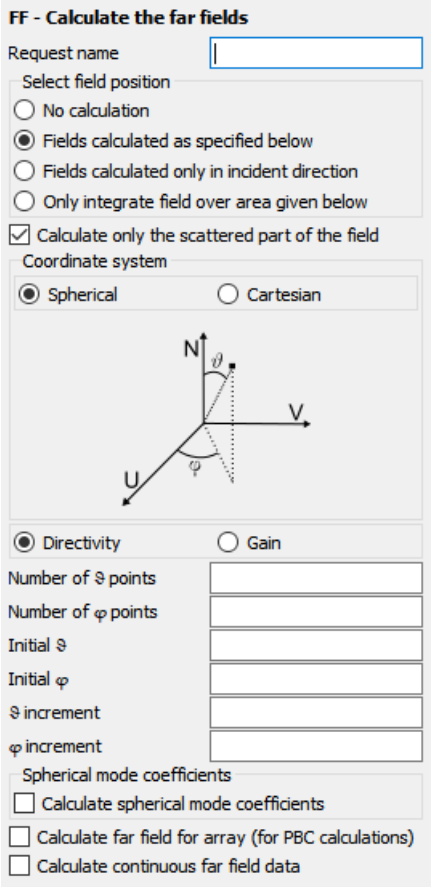


*Figure 897: The **FF - Calculate the far fields** dialog.*

Parameters:

Request name	The name of the request.
No calculation	Disables the calculation.
Fields calculated as specified below	The far field is calculated with the specified settings.
Fields calculated only in the incident direction	The far field is calculated only in the incident direction (used, for example, to calculate monostatic RCS). The rotation and phase reference parameters of the incident plane wave (specified at the A0 card) are used for the field calculation.
Only integrate field over area given below	The far field is calculated but it is not written to the .out file in order to limit its size. This option is meaningful when the individual values of the field strength (such as directivity and

	gain) are not required, but the total radiated power has to be calculated from the integral of the Poynting vector (see the discussion below), or if only the modal coefficients are required. If a `.ffe` file has been requested with the DA card, the field values used in this integration will be written to the file.
Calculate only the scattered part of the field	When this item is checked, the field radiated by the impressed sources (such as Hertzian dipoles) is not included. Normally one would not check this item so that the total field is calculated. The total field includes all source contributions except plane wave sources.
Coordinate system	Select to use the spherical coordinate system (ϑ, φ) or a Cartesian grid (u, v) to define the far field request points.
Directivity/Gain	Select the required quantity.
Number of ϑ/u points	The number of observation points in the ϑ/u direction. An empty field will be set to 1.
Number of φ/v points	The number of observation points in the φ/v direction. An empty field will be set to 1.
Initial ϑ/u	The angle ϑ_0 in degrees of the first observation point when using spherical coordinates, or the u component of the unit vector of the first observation point when using Cartesian coordinates.
Initial φ/v	The angle φ_0 in degrees of the first observation point when using spherical coordinates, or the v component of the unit vector of the first observation point when using Cartesian coordinates.
ϑ/u increment	Increment $\triangle\vartheta$ in degrees of the angle ϑ when using spherical coordinates. Increment in u (unitless) when using Cartesian coordinates.
φ/v increment	Increment $\triangle\varphi$ in degrees of the angle φ when using spherical coordinates. Increment in v (unitless) when using Cartesian coordinates.
Calculate spherical mode coefficients	Calculate the spherical mode coefficients of the far field.
Specify number of modes	When this option is unchecked, the number of modes is automatically determined when the spherical mode coefficients are computed. If this option is checked, the **Maximum mode index N** must be specified, which will affect the number of modes calculated.
Maximum mode index N	Specify the maximum number of modes to be calculated by means of the mode index.

Calculate far field for array (for PBC calculations)	Check this item if the far field is to be calculated for an array of elements. The **Number of elements in $\hat{u}_1$ direction (PE card)** and the **Number of elements in $\hat{u}_2$ direction (PE card)** should be specified if this option is chosen.
Calculate continuous far field data	Use interpolation to display continuous far field data.

Tip: To calculate monostatic radar cross section (RCS) for a number of directions of incidence for the plane wave source, select **Fields calculated only in incident direction**.

When using the FF card with **Number of ϑ points**, N_ϑ , and **Number of φ points**, N_φ, both larger than 1, the Poynting vector is integrated over the two spherical segments as follows:

- $\vartheta_0-\frac{1}{2}\cdot\Delta\vartheta\le\vartheta\le\vartheta_0+(N_\vartheta-\frac{1}{2})\cdot\Delta\vartheta$ and $\varphi_0-\frac{1}{2}\cdot\Delta\varphi\le\varphi\le\varphi_0+(N_\varphi-\frac{1}{2})\cdot\Delta\vartheta$
- $\vartheta_0\le\vartheta\le\vartheta_0+(N_\vartheta-1)\cdot\Delta\vartheta$ and $\varphi_0\le\varphi\le\varphi_0+(N_\varphi-1)\cdot\Delta\varphi$

In the case of an antenna the power provided by the voltage sources must be equal to the radiated power over the whole sphere. The total radiated power can be calculated using for example the following commands:

```
** Far field integration in angular increments of #delta (in degrees)
#delta=5
#nt=180/#delta + 1
#np=360/#delta + 1
FF: 3 : #nt : #np : 0 : 0 : 0 : 0 : #delta : #delta
```

The output in the .out file then reads as follows:

```
Integration of the normal component of the Poynting vector in the
angular grid DTHETA =    5.00 deg. and DPHI =    5.00 deg. ( 2701 sample points)
  angular range THETA        angular range PHI       radiated power
 -2.50 .. 182.50 deg.      -2.50 .. 362.50 deg.     5.60499E-03 Watt
  0.00 .. 180.00 deg.       0.00 .. 360.00 deg.     5.52821E-03 Watt
```

If the model is symmetrical, it is not necessary to integrate over the full sphere. If there are three planes of symmetry (as for a simple dipole in free space) the integration only needs to be done over an eighth of the sphere. The power then has to be multiplied by 8.

If an infinite ground plane has been specified, the calculation of the far fields below the ground plane is not possible. Observation points with z < 0 will therefore be ignored.

Use the OF card to specify an offset for the coordinate system's origin used in far field calculations.

Related tasks
Requesting a Far Field (CADFEKO)
Related reference
A0 Card
OF Card

Far Field Coordinate Systems (FF card)

The coordinate systems for the FF card have specific definitions/conventions. Use the spherical coordinate system in general and the Cartesian coordinate system when appropriate for the application.

Spherical coordinates ϑ, φ

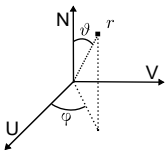


Figure 898: Field calculation in the spherical coordinate system.

The far field pattern of an antenna is assumed to be independent of the distance from the antenna. Specify the ϑ and φ angles of the direction to the observation point **r**.

Unit vectors of the coordinate system:

$$\hat{\mathbf{r}} = \begin{bmatrix} \sin\vartheta\cos\varphi \\ \sin\vartheta\sin\varphi \\ \cos\vartheta \end{bmatrix} \qquad \hat{\boldsymbol{\vartheta}} = \begin{bmatrix} \cos\vartheta\cos\varphi \\ \cos\vartheta\sin\varphi \\ -\sin\vartheta \end{bmatrix} \qquad \hat{\boldsymbol{\varphi}} = \begin{bmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{bmatrix} \tag{200}$$

Cartesian coordinates u, v

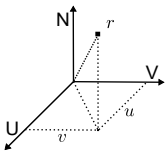


Figure 899: Field calculation in the Cartesian coordinate system.

The far field pattern of an antenna is assumed to be independent of the distance from the antenna. Specify the first two coordinates (u, v) of the unit vector to the observation point **r**.

Observation point on the unit sphere:

$$\hat{\mathbf{r}} = \begin{bmatrix} u \\ v \\ \sqrt{1-u^2-v^2} \end{bmatrix} \tag{201}$$

Convert between spherical and Cartesian coordinates:

$$\begin{aligned} u &= \sin\vartheta\cos\varphi \\ v &= \sin\vartheta\sin\varphi \\ \vartheta &= \arccos\left(\sqrt{1-u^2-v^2}\right) \\ \varphi &= \operatorname{atan2}(v,u) \end{aligned} \tag{202}$$

Spherical Mode Coefficients

Calculate spherical modes for use in a spherical mode equivalent source or export the modes to GRASP from TICRA.

The calculation of spherical mode coefficients is based on the far field values and the coefficients are written to the .out file.

Spherical modes have three indices s, m, and n with s = 1 for TE-modes, s = 2 for TM-modes, m = -N . . .N, and n = 1 . . .N, including a one-dimensional compressed indexing scheme j = 1. . . J and the normalisation of the modes and so forth. The input parameter **Maximum mode index N** can be used to manually set the maximum mode index N. Based on the choice for N a total number of $J = 2N(N + 2)$ modes will be computed. If the **Specify number of modes** check box is not checked, Feko will automatically calculate the modes up to a maximum mode index based on factors such as the model size, the individual power in each mode and the total power captured in all the modes.

> **Tip:** If only spherical modes are required, only a single far field point (position irrelevant) could be requested.

The origin for the modes is the same as for the far field computation in general (which is the global origin unless an OF card has been used to specify an offset). Due to the nature of the far field (propagating towards r = ∞) all computed mode coefficients refer to spherical cylinder functions $z_n^{(c)}$ with type c = 4.

> **Tip:** Use the DA card before the FF card to request the export of the spherical expansion modes to an SWE file (`.sph` file) for import into GRASP from TICRA.

GRASP is a reflector antenna modelling code, and by means of the SWE file export it is possible to model a horn antenna as feed in Feko and then export this feed structure for use in GRASP.

> **Note:** The gain in GRASP will only be correct if the radiated power in Feko has been set to 4π Watt.

The spherical mode expansion coefficients Q_{smn} as used by Feko are described in the AS card. In GRASP a slightly different convention is used:

- An additional factor $\frac{1}{\sqrt{8\pi}}$ is used.
- The coefficients are conjugate complex (that is GRASP assumes an $e^{-i\omega t}$ time dependency).
- The index m is exchanged with -m.

  > **Note:** In Feko the φ dependency is defined as $e^{jm\varphi}$ but in GRASP it is defined as $e^{-jm\varphi}$.

The above conversions are done automatically by Feko when exporting the SWE file, so that this can readily be imported into GRASP.

Related reference
OF Card

FR Card

The FR card sets the frequency/frequencies (in Hz), at which the solution will be obtained.

On the **Source/Load** tab, in the **Settings** group, click the **Frequency (FR)** icon.

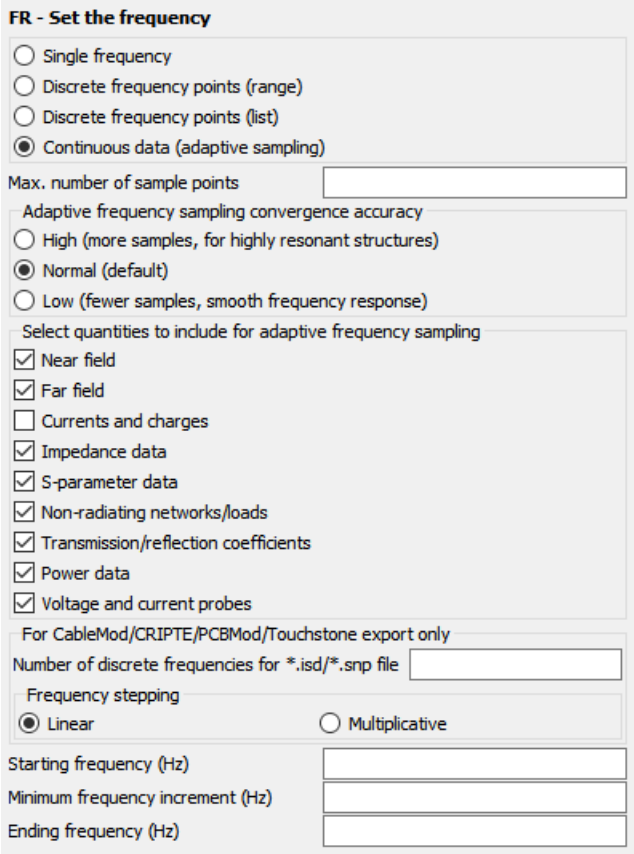


*Figure 900: The **FR - Set the frequency** dialog.*

The solution can be done for a single frequency, a range of discrete frequencies (linear or multiplicative stepping), a list of discrete frequencies or a continuous solution in a given frequency band with adaptive frequency interpolation with the option to set the convergence accuracy. For a continuous solution, only one FR card is allowed. Specific quantities of interest to the user may also be selected to be included for adaptive sampling. Unselected quantities will be calculated at the discrete solution frequency points.

Parameters:

Single frequency
Only a single frequency will be analysed.

Discrete frequency points (range)
If this item is selected the following parameters are applicable:

- **Number of frequency points**: For a discrete frequency range sweep, the number of frequency samples must be larger than 1.
- **Frequency scale**: In this group either **Linear** or **Multiplicative scaling** is selected. If

		Linear is selected, then consecutive frequencies differ with a fixed value, for example, the new frequency is the previous value plus the frequency increment. If **Multiplicative** is selected, then consecutive frequencies differ by a constant factor, for example, the new frequency is the previous value multiplied by the frequency factor.
	Specify by	If **Frequency increment/factor** is selected, the user specifies the increment or factor mentioned in the previous item. Thus, the number of frequencies and the start frequency then determine the ending frequency. If **Ending frequency** is selected, the user specifies this and the increment/factor is calculated.
Discrete frequency points (list)	If this item is selected, the following parameters are applicable:	
	Number of discrete frequencies	For a list of discrete frequency sweep, the number of frequency samples must be larger than 1. The list of discrete frequencies is applicable when a list of frequencies is required which is not linearly or logarithmically spaced.
Continuous data (adaptive sampling)	Select **interpolate** this item to use an adaptive frequency interpolation technique to obtain a continuous representation of the results in the given frequency band. When using this feature, the remaining parameters have the following meaning:	
	Max. number of sample points	Maximum number of discrete frequency points in this frequency band at which Feko may be executed (limitation to avoid convergence problems). If left empty, the default value of 1000 will be used.
	Adaptive frequency sampling convergence accuracy	This allows the user to manually set the adaptive frequency sampling convergence accuracy. Note that the default setting should be used for most cases, but in special cases where a structure with many resonances is being modelled a higher convergence accuracy should be used.

Select quantities to include for adaptive frequency sampling	The quantities selected will be included in the adaptive frequency sampling.
For CableMod/ CRIPTE/PCBMod/ Touchstone export only, Number of discrete frequencies for `.isd` / `.snp` file	This field is only relevant when the CM or SP card is used to create an `.isd` or `.snp` file respectively. The results are written to the `.isd` or `.snp` file for the number of discrete frequencies specified in this field.
Starting frequency (Hz)	Defines the start frequency of the simulation range.
Ending frequency (Hz)	Defines the end frequency of the simulation range.
Minimum frequency increment (Hz)	Minimum increment between adaptive samples, see the note below.

In order to obtain a continuous frequency response, the adaptive frequency interpolation technique obtains the solution at a set of discrete frequency points. They are automatically placed, for example using large frequency increments in regions with a smooth behaviour of the results, and much finer frequency increments close to resonances. Sometimes, for example when using a frequency dependent mesh, the Feko results versus frequency may contain small discontinuities. In these cases the adaptive algorithm cannot converge. (It will continue to refine the frequency increment as it tries to fit a smooth curve through the discontinuity and will only stop when the **Max. number of sample points** is reached.) One may avoid this by setting the **Frequency increment** to the minimum allowable separation distance between neighbouring frequency sample points. The value of the **Frequency increment** must be smaller than the resolution required to solve, for example, sharp resonances. If left empty, the default is:

$$\frac{1}{10000}(Ending\ frequency - Starting\ frequency) \tag{203}$$

If a discrete loop with more that one frequency is required then either the frequency increment or the ending frequency must be specified, but not both. If the end frequency is specified, the frequency increment is calculated from:

- for a linear frequency scale (additive increments):

$$\triangle_f = \frac{f_2 - f_1}{N_f - 1} \tag{204}$$

- for a multiplicative frequency scale (multiplicative increments):

$$fac = \left(\frac{f_2}{f_1}\right)^{\frac{1}{N_f - 1}} \tag{205}$$

where $\triangle_f$ is the frequency increment (linear stepping), *fac* the increment factor (multiplicative stepping), f_1 the start frequency, f_2 the ending frequency and N_f the number of frequencies.

When writing results at discrete frequencies to a `.isd` file, the frequency increment when a linear frequency scale is used is calculated similar to the case for $\triangle_f$ as shown above.

If more than one frequency is to be examined, then all the control cards up to the next FR card or EN card will be read into a buffer and are executed for each frequency.

Related concepts
Frequency Options (CADFEKO)
Related reference
CM Card
EN Card
SP Card

GF Card

This card includes the complexities of dielectric environments using special Green's functions. The Green's functions relates the fields in space to the sources.

On the **Home** tab, in the **Planes / arrays** group, click the **Plane / ground** icon. From the drop-down list, click the **Green's function (GF)** icon.

The Green's functions supported by Feko are as follows:

- **Homogeneous medium**: The dielectric properties for the entire problem space can be set. This is useful for modelling in a homogeneous medium that differs from free space.
- **Layered dielectric sphere**: A layered dielectric sphere located at the origin is taken into account with the Green's function.
- **Planar multilayer substrate**: A multilayer dielectric substrate in the XY plane is taken into account. The layers can have ground planes at any arbitrary z-values.

Using Green's functions to model the presence of these dielectric regions means that their influence is taken into account implicitly, using less computational resources than modelling them using either the volume- or surface equivalence principles.

Related tasks
Defining an Infinite Planar Multilayer Substrate (CADFEKO)
Related reference
Planar Multilayer Substrate Limitations

Homogeneous Medium

With this option the dielectric properties for the entire problem space can be set. This is useful for modelling in a homogeneous medium that differs from free space.

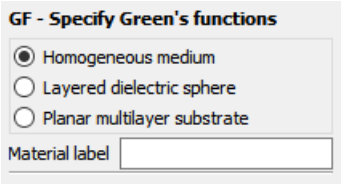


*Figure 901: The **GF - Specify Green's functions** dialog, set to **Homogeneous medium***

When this Green's function is selected, the EM problem under investigation is modelled inside an infinite space of the homogeneous medium. This is the standard "free space" Green's function similar to when the GF card is not used. The medium is normally free space, defined by material label "0", but different parameters can be set with the DI card.

Parameters:

Material label	The label of the homogeneous medium to be used, as defined in the DI card

Layered Dielectric Sphere

With this option a layered dielectric sphere located at the origin is taken into account with the Green's function.

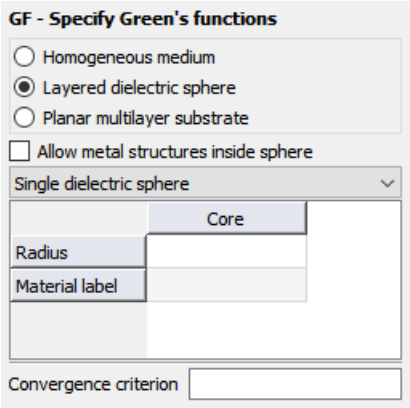


*Figure 902: The **GF - Specify Green's functions** dialog, set to **Layered dielectric sphere**.*

When this Green's function is selected, the EM interaction of a layered dielectric sphere located at the coordinate system centre is taken into account. With this option it is, for example, possible to analyse a mobile phone in front of a spherical shell model of the human head very efficiently.

Parameters:

Configuration list

The drop-down list allows selecting between a **Single dielectric sphere**, a **Core and a coating layer** and a **Core and three layers**.

> **Note:** If metal structures are included, the only options are **Single dielectric sphere** and **Core and two layers**.

Allow metal structures inside sphere

When this item is checked metallic structures can be present in the inner parts of the sphere.

Convergence criterion

Convergence criteria for the summation of the rows of Green's functions. If this field is 0 or undefined, a sensible standard criterion is used.

Radius

Radius of the sphere / layer in metres (is scaled by the SF card). For the layers, this is the total radius of the core plus layers up to that point. The highest numbered layer is on the outside of the sphere.

Material label

Label of the material (as defined in the DI card) to be used for the core / layer.

The scaling factor that is entered by the SF card is applied to the radius. Note that the surrounding medium is defined by material label “0”. By default the values of free space is used, but these parameters can be redefined in the DI card.

The Green’s function for a homogeneous or layered dielectric sphere can be used with metallic structures (treated with the MoM) either inside or outside the sphere (but not for example a wire from inside to outside). It can be used with dielectric bodies treated with the volume equivalence principle (for example the hand of a user around a mobile phone), but the dielectric bodies must be outside the sphere.

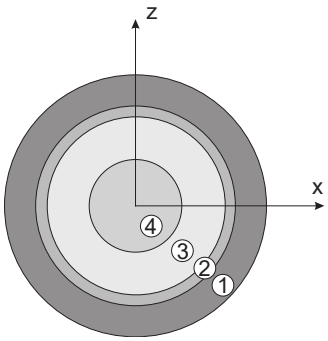


Figure 903: Example of a sphere consisting of 4 media (core and 3 layers) indicating the layer numbering.

Planar Multilayer Substrate

With this option a multilayer dielectric substrate in the XY plane is taken into with the Green's function.

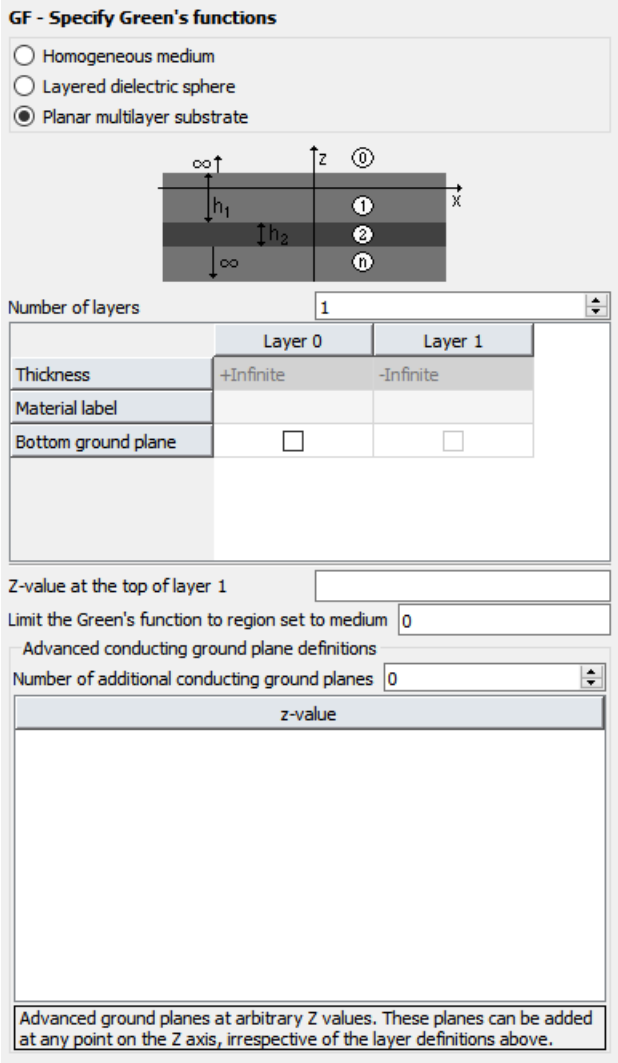


*Figure 904: The **GF - Specify Green's functions** dialog, set to **Planar multilayer substrate**.*

When this option is selected, the EM interaction of a layered dielectric substrate located in the XY plane is taken into account. This formulation has been popularised by its application to planar (microstrip) circuits and antennas, but it is applicable to a larger class of EM problems such as antennas underground.

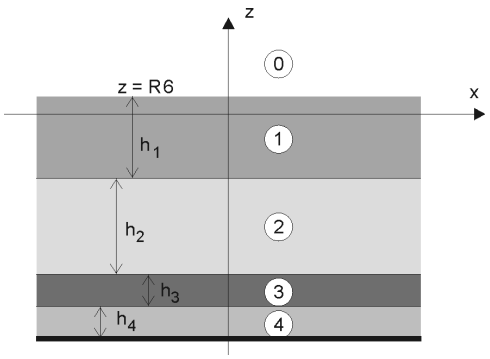


Figure 905: Example of a 4 layer substrate with a metallic ground plane

Parameters:

Number of layers	Number of layers in the substrate (layer 0 — the upper half-space) is not included in the number. As indicated in the figure of the card, layer 0 is the upper half-space, layer 1 is just beneath this, and so forth.
Thickness	Thickness of the layer (is scaled by the SF card).
Material label	Label of the material to be used for the layer (as defined in the DI card).
Bottom ground plane	By checking **Bottom ground plane** at the respective layer, a user can choose to have a ground plane at the bottom of the selected layer.
Z-value at the top of layer 1	The value of the Z coordinate at the transition between layer 0 and layer 1.
Limit the Green's function to region set to medium	An optional parameter which allows for the planar multilayer substrate to be limited to a specified dielectric region. A dielectric medium must be defined and a dielectric region set to this medium. (Refer to the DI card and DL card for more information.) The planar multilayer substrate is then limited to this dielectric region specified by the label of the dielectric medium (as defined in the ME card). For more information refer to the combination of MoM/SEP and planar Green's function in the CADFEKO section.
Number of additional conducting ground planes	Number of additional conducting ground planes to be added at arbitrary z-values.
Advanced ground planes at arbitrary z-values	The z-value at which the ground plane is to be added.

Structures can be located/orientated at any arbitrary position/orientation in one or more layers. The following restrictions apply:

- No metallic segment or triangle may cross a boundary between layers. More specifically, it must be positioned completely within one layer or at the boundary between the layers. For example, in the case of a metallic wire that penetrates a multilayer substrate, the meshing should ensure that there is a node on each interface between layers. This meshing is automatically done in CADFEKO.
- Dielectric triangles may not lie on the interface (boundary) between substrate layers.

Related reference

DI Card

DL Card

ME Card

KC Card

The KC card facilitates the transferral of connector names from CADFEKO to POSTFEKO.

On the **Solve/Run** tab, in the **Cables** group, click the **Connector (KC)** icon.

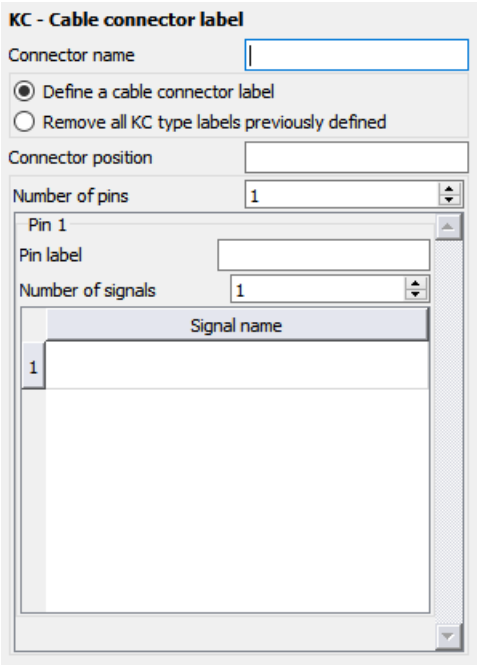


*Figure 906: The **KC - Cable connector label** dialog.*

Parameters:

Connector name	The name of the connector.
Define a cable connector label	Define a cable connector label with the following parameters.
Remove all KC type labels previously defined	This KC card does not define a load, but rather all previously defined KC labels are deleted. All the other input parameters of this card are ignored.
Connector position	Data point/node to access the geometrical coordinates of this connector.
Number of pins	The number of pins defined in the connector.

Related tasks

Defining Cable Connectors (CADFEKO)

KS Card

The KS card facilitates the transferral of signal names from CADFEKO to POSTFEKO.

On the **Solve/Run** tab, in the **Cables** group, click the **Signals (KS)** icon.

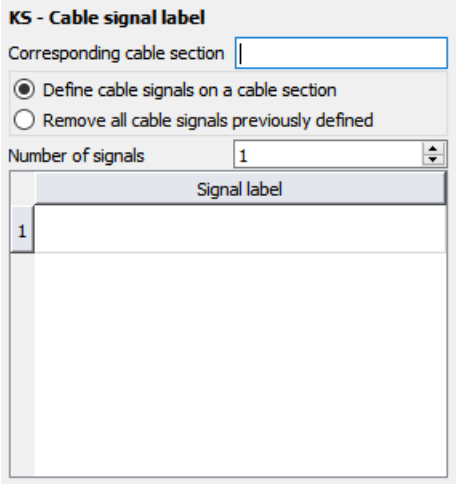


*Figure 907: The **KS - Cable signal label** dialog.*

Parameters:

Corresponding cable section	The label of the CS card that links to this KS card.
Define cable signals on a cable section	Define cable signals on a cable section with the following parameters.
Remove all cable signals previously defined	This KS card does not define a cable signal, but rather all previously defined cable signals are deleted. All the other input parameters of this card are ignored.
Number of signals	The number of signals being defined. It should correspond to the number of signals/connections for the respective cable instance.

Related reference
CD Card

L2 Card

This L2 card places a load (complex impedance) on a node.

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Vertex load (L2)** icon.

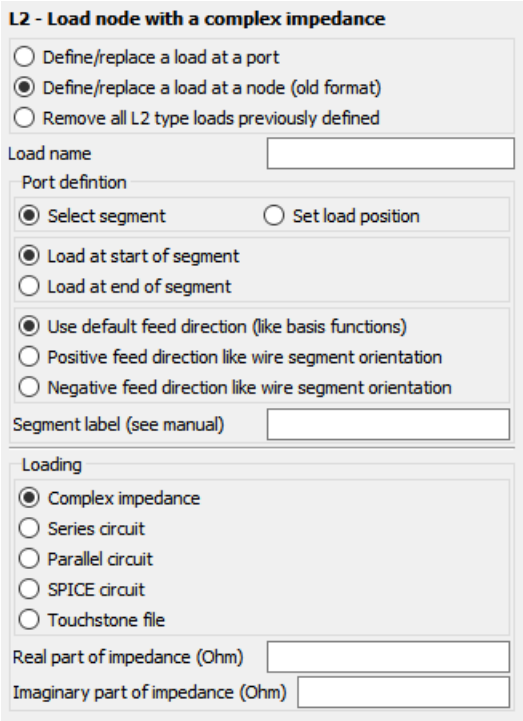


*Figure 908: The **L2 - Load node with a complex impedance** dialog.*

Parameters:

Define/replace a load at a port	Define/replace a complex impedance load at the port with name **Port** defined using a PT card.
Define/replace a load at a node (old format)	Define a load with the following parameters.
Remove all L2 type loads previously defined	This L2 card does not define a load, but rather all previously defined L2 loads are deleted. All the other input parameters of this card are ignored.
Load name	The name of the load.
Select segment	When this item is selected, then the **Segment label** text box becomes active. This text box specifies the label of the segment which shall be loaded (either start or end point as determined by the corresponding check box). The load has to be located at a node, either between two segments, or between a segment and a triangle, ground plane or polygonal plate. Only one segment with

	this label should be declared. If there is more than one segment with this label then all the corresponding segment vertices will be loaded.
Set load position	When this check box is activated, then the load node is determined by specifying its Cartesian coordinates in the **Coordinates of node** group. These values are in m and may be scaled by the SF card.
Load at start of segment	This option is only available when selecting the feed segment by label. If set, it indicates that the load location is at the start of the wire segment with a matching label.
Load at end of segment	This option is only available when selecting the feed segment by label. If set, it indicates that the load location is at the end of the wire segment with a matching label.
Use default feed direction (like basis function)	This option is only available when selecting the feed segment by label. If set, it indicates that the positive feed direction is according to the basis function setup in Feko. For wire/surface junctions (UTD plates, infinite ground, or meshed triangle surfaces), this direction is away from the wire onto the surface. For wire connections between two segments, this direction is from the segment with the lower index to the segment with the higher index.
Positive feed direction like wire segment orientation	This option is only available when selecting the feed segment by label. If set, then the positive feed direction is according to the orientation of the wire segment with the specified label.
Negative feed direction like wire segment orientation	This option is only available when selecting the feed segment by label. If set, then the positive feed direction is opposite to the orientation of the wire segment with the specified label.
Loading	*Complex impedance* Define the real and imaginary parts of the complex impedance in Ohm using **Real part of impedance (Ohm)** and **Imaginary part of impedance (Ohm)** respectively. *Series circuit* The resistor value in Ohm, inductor value in Henry and the capacitor value in Farad to be added as a series circuit. *Parallel circuit* The resistor value in Ohm, inductor value in Henry and the capacitor value in Farad to be added as a parallel circuit.

SPICE circuit
Specify the name of a one-port SPICE circuit to define a load between two pins. Define the SPICE circuit using the SC card.

Touchstone file
Specify a one-port Touchstone file (`.s1p`, `.z1p`, `.y1p`) to define a load.

> **Note:** If the load is added to a port that has a voltage source, the load is placed in series with the voltage source.

Related tasks
Adding a Load (CADFEKO)
Related reference
PT Card

LC Card

The LC card specifies complex, series and parallel circuits applied between connector pins and also between a connector pin and ground.

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Load cable (LC)** icon.

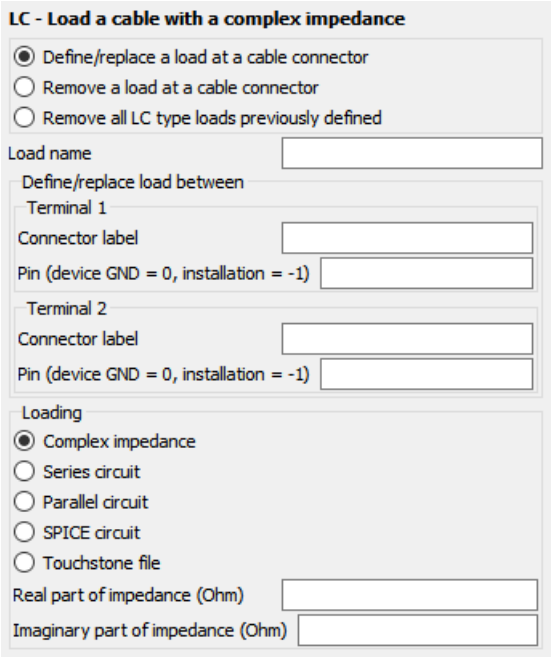


*Figure 909: The **LC - Load a cable with a complex impedance** dialog.*

Parameters:

Define/replace a load at a cable connector	Define/replace a load at a cable connector with the following parameters.
Remove a load at a cable connector	A load can be removed between two connector pins by defining the **Connector label** and **Pin**.
Remove all cable loads previously defined	All previously defined LC type loads are removed.
Load name	The name of the load.
Define/replace load between	A load can be placed between two connector pins by specifying the **Connector label** and pin number for both connectors. If the pin is set to 0, the connector pin is connected to the local circuit device ground. If the pin is set to -1, the connector pin is connected to the global installation.

Complex impedance	Define the real and imaginary parts of the complex impedance in Ohm using **Real part of impedance (Ohm)** and **Imaginary part of impedance (Ohm)** respectively.
Series circuit	The resistor value in Ohm, inductor value in Henry and the capacitor value in Farad to be added as a series circuit.
Parallel circuit	The resistor value in Ohm, inductor value in Henry and the capacitor value in Farad to be added as a parallel circuit.
SPICE circuit	Specify the name of a one-port SPICE circuit to define a load between two pins. Define the SPICE circuit using the SC card.
Touchstone file	Specify a one-port Touchstone file (`.s1p`, `.z1p`, `.y1p`) to define a load. **Note:** If the load is added to a port that has a voltage source, the load is placed in series with the voltage source.

For detailed conductor to cable connector pin relation, see the AK card.

Related reference
Cable Schematic Elements (CADFEKO)
AK Card

LD Card

This card specifies a distributed resistive, capacitive or inductive loading or a series combination of these loads for a segment(s).

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Distributed load (LD)** icon.

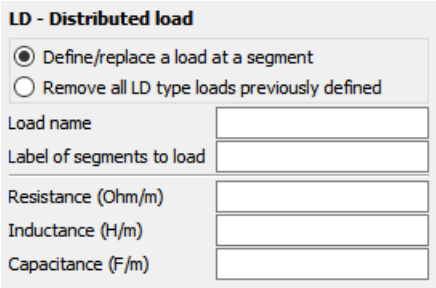


*Figure 910: The **LD - Distributed load** dialog.*

Parameters:

Define/replace a load at a segment	Define a load with the following parameters.
Remove all LD type loads previously defined	This LD card does not define a load, but rather all previously defined LD loads are deleted. All the other input parameters of this card are ignored.
Load name	The name of the load.
Label of segments to load	All segments with this label are subjected to distributed loading.
Resistance:	The distributed resistance in $\frac{\Omega}{m}$.
Inductance	The distributed inductance in $\frac{H}{m}$.
Capacitance	The distributed capacitance in $\frac{F}{m}$.

The combined impedance of the segment with length *l* is then

$$Z_s = (R' + j\omega L' + \frac{1}{j\omega C'}) \cdot l \qquad (206)$$

It should be noted that if the Capacitance (F/m) is left empty, it is treated as infinite, so that it does not contribute to the impedance.

The LD card may be combined with the LP, LS, LZ and the SK cards, but only one LD card may be used per label. If a second LD card is used, it replaces the values entered by the first card. This card has no significance for surface elements, even when these are assigned the same label.

Related tasks
Adding a Load (CADFEKO)
Related reference
LP Card
LS Card
LZ Card
SK Card

LE Card

This card specifies complex series and parallel circuits applied to an edge between surface triangles.

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Load edge (LE)** icon.

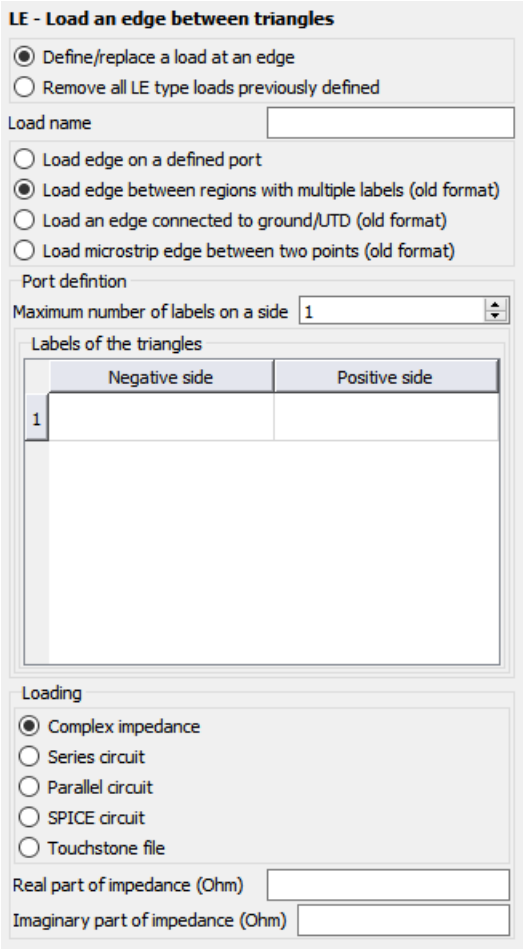


*Figure 911: The **LE - Load an edge between triangles** dialog.*

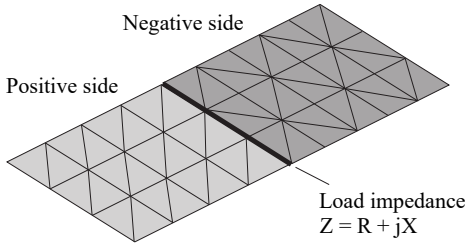


Figure 912: Application of the LE card.

Parameters:

Define/replace a load at an edge	Define a load with the following parameters.

Remove all LE type loads previously defined

This LE card does not define a load, but rather all previously defined LE loads are deleted. All the other input parameters of this card are ignored.

Load name

The name of the load.

Load edge on a defined port

The name of the referenced port defined using a PT card.

Load edge between regions with multiple labels (old format)

The edge between different regions is loaded with a complex impedance. The following parameters apply when this item is checked:

Maximum number of labels on a side	his indicates the maximum number of different labels used on either side of the load. Increasing this value will add rows to the table for the labels.
Negative side	The **Labels of triangles** on the one side of the load.
Positive side	The **Labels of triangles** on the other side of the load.

Load an edge connected to ground/UTD (old format)

Load the triangles with specified labels that are connected to a UTD surface or to a PEC ground plane (as specified with a BO or GF card). The following parameters apply when this item is checked:

Maximum number of labels on a side	This indicates the maximum number of different labels used on either side of the load. Increasing this value will add rows to the table for the labels.
Meshed surface represents positive feed side	If selected, the **Positive side** of the **Labels of triangles** is connected to ground. If not selected, the **Negative side** of the **Labels of triangles** is connected to ground.

Load microstrip edge between two points (old format)

This is a special microstrip port load. The load is placed on all edges on the line between two points (previously specified with DP cards) entered into the dialog. A GF card with a conducting ground plane must be present. The following parameters apply when this item is checked:

Start point of edge	The start point (not label) of the edge.
End point of edge	The end point (not label) of the edge.

Meshed surface represents positive feed side	If selected, the positive side of the microstrip edge between two points is connected to ground. If not selected, the negative side of the microstrip edge between the points is connected to ground.
Complex impedance	The real and imaginary part of the complex impedance in Ω.
Series circuit	The resistor value in Ω, inductor value in Henry and the capacitor value in Farad to be added as a series circuit.
Parallel circuit	The resistor value in Ω, inductor value in Henry and the capacitor value in Farad to be added as a parallel circuit.
SPICE circuit	Specify the name of a one-port SPICE circuit to define a load between two pins. Define the SPICE circuit using the SC card.
Touchstone file	Specify a one-port Touchstone file (`.s1p`, `.z1p`, `.y1p`) to define a load.

> **Note:** If the load is added to a port that has a voltage source, the load is placed in series with the voltage source.

See also the AE card for the excitation of such an edge. As shown in the figure above, the edge can consist of several single edges between regions specified by labels. (See the AE card for a discussion on the allowed configurations.) Alternatively the edge can be along a connection between triangles and a polygonal plate or a PEC ground plane, or it can be a microstrip feed line port. The impedance Z applies to the complete edge (all the single edges in parallel). The LE card can be combined with the AE card to specify both an impedance and a voltage source over the edge.

Note that the edge between the triangles does not need to be straight. One may, for example, specify a resistive connection between two half cylinders.

Related tasks
Adding a Load (CADFEKO)
Related reference
AE Card
BO Card
DP Card
GF Card
PT Card

LF Card

This card impresses a complex impedance between two points inside a FEM mesh.

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Load FEM (LF)** icon.

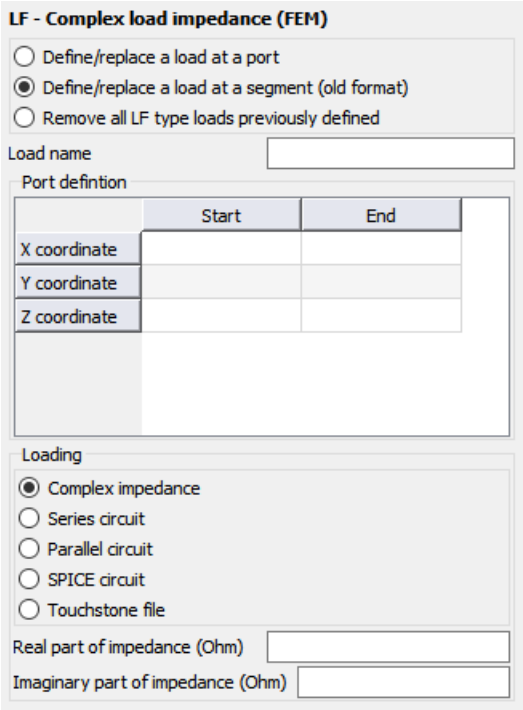


*Figure 913: The **LF - Complex load impedance (FEM)** dialog.*

Parameters:

Define/replace a load at a port	Define/replace a FEM load at the port with name **Port** defined using a PT card.
Define/replace a load at segment (old format)	Define a load with the following parameters.
Remove all LF type loads previously defined	This LF card does not define a load, but rather all previously defined LF loads are deleted. All the other input parameters of this card are ignored.
Load name	The name of the load.
Real part of impedance (Ohm)	The real part of the complex impedance in Ω.
Imaginary part of impedance (Ohm)	The imaginary part of the complex impedance in Ω.
Complex impedance	The real and imaginary part of the complex impedance in Ω.

Series circuit	The resistor value in Ω , inductor value in Henry and the capacitor value in Farad to be added as a series circuit.
Parallel circuit	The resistor value in Ω , inductor value in Henry and the capacitor value in Farad to be added as a parallel circuit.
SPICE circuit	Specify the name of an one-port SPICE circuit to define a load between two pins. Define the SPICE circuit using the SC card.
Touchstone file	Specify a one-port Touchstone file (`.s1p`, `.z1p`, `.y1p`) to define a load. **Note:** If the load is added to a port that has a voltage source, the load is placed in series with the voltage source.
Load position	The Cartesian coordinates of the start and end points of the load.

The complex impedance is impressed between two points inside the FEM mesh. The line may be positioned arbitrarily inside the FEM mesh, meaning it does not have to be coincident with tetrahedral edges, but the length between the points should be small compared to the shortest wavelength in the band of interest to obtain reasonable accuracy.

Note: The LF type load should not be used to model a short-circuit load. A short can be modelled by a metallic strip (meshed into triangles) inside a FEM region.

Related tasks
Adding a Load (CADFEKO)
Related reference
AF Card
PT Card

LN Card

This card defines a complex load to any non-radiating network port.

> **Note:** An LN load is always defined across the network port in series with the source. The port may be a non-radiating network port or a geometry port (segment/vertex/edge port).

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Load network (LN)** icon.

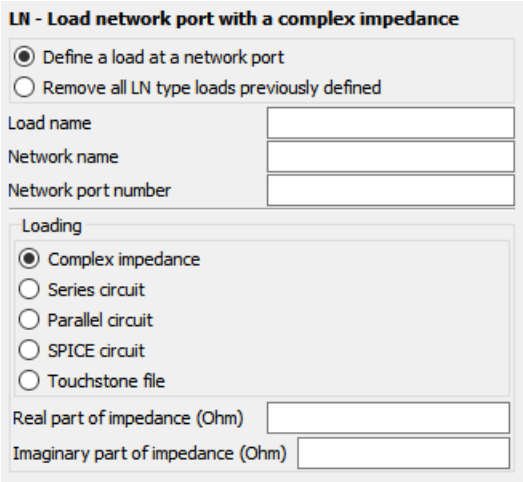


*Figure 914: The **LN - Load network port with a complex impedance** dialog.*

Parameters:

Define a load at a network port — Define a network load with the following parameters.

Remove all LN type loads previously defined — All previously defined LN type loads are removed. This replaces all network loads with open circuits. Note that setting the load impedance to zero creates a short circuit between the network terminals.

Load name — The name of the load.

Network name — The network or transmission line name, with the network port number, that uniquely identifies the connection terminals.

Network port number — The network port number, with the network or transmission line name, that uniquely identifies the connection terminals.

Loading

Complex impedance
: Define the real and imaginary parts of the complex impedance in Ohm using **Real part of impedance (Ohm)** and **Imaginary part of impedance (Ohm)** respectively.

Series circuit
The resistor value in Ohm, inductor value in Henry and the capacitor value in Farad to be added as a series circuit.

Parallel circuit
The resistor value in Ohm, inductor value in Henry and the capacitor value in Farad to be added as a parallel circuit.

SPICE circuit
Specify the name of a one-port SPICE circuit to define a load between two pins. Define the SPICE circuit using the SC card.

Touchstone file
Specify a one-port Touchstone file (`.s1p`, `.z1p`, `.y1p`) to define a load.

> **Note:** If the load is added to a port that has a voltage source, the load is placed in series with the voltage source.

Real part of impedance (Ohm)	The real part of the complex impedance in Ω.
Imaginary part of impedance (Ohm)	The imaginary part of the complex impedance in Ω.

Related concepts
Network Schematic View (CADFEKO)

LP Card

This card assigns a parallel circuit of discrete elements to a segment.

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Parallel load (LP)** icon.

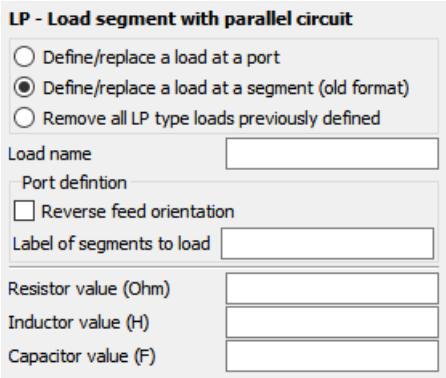


*Figure 915: The **LP - Load segment with parallel circuit** dialog.*

The figure below depicts the parallel circuit that can be assigned to a segment.

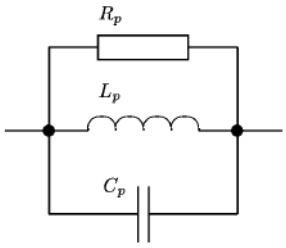


Figure 916: Sketch of the parallel circuit.

Parameters:

Define/replace a load at a port	Define/replace a parallel circuit at the port with name **Port** defined using a PT card.
Define/replace a load at a segment (old format)	Define a load with the following parameters.
Remove all LP type loads previously defined	This LP card does not define a load, but rather all previously defined LP loads are deleted. All the other input parameters of this card are ignored.
Load name	The name of the load.
Reverse feed orientation	By default, the vector of the voltage is orientated in the direction from the start of the segment to its end (the direction in which it was created). When this option is checked, the vector of the voltage is orientated in the opposite direction. (This is the

	direction of the current flow through the segment. The internal EMF (electromagnetic force) of the impressed voltage source is in the opposite direction.
Label of segments to load	All segments with this label are assigned the parallel circuit values specified below.
Resistor value (Ohm)	Value of the resistor in Ω
Inductor value (H)	Value of the inductor in H.
Capacitor value (F)	Value of the capacitor in F.

The impedance is then given by

$$Z_p = \frac{1}{\frac{1}{R_p} + \frac{1}{j\omega L_p} + j\omega C_p} \tag{207}$$

If the resistance is set to zero, then the resistance is interpreted as infinite, therefore in the parallel case it will not change the impedance. The same applies to the inductance.

The LP card may be combined with the LD, LS, LZ and the SK cards, but only one LP card may be used per label. If a second LP card is used, it replaces the values entered by the first one. This card has no significance for surface elements, even when these are assigned the same label.

> **Note:** Even though the circuit is a parallel circuit, the circuit as a whole is placed in series with the segment to which it is applied.

Related tasks
Adding a Load (CADFEKO)
Related reference
LD Card
LS Card
LZ Card
PT Card
SK Card

LS Card

This card assigns a series circuit of discrete elements in series to a segment.

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Series load (LS)** icon.

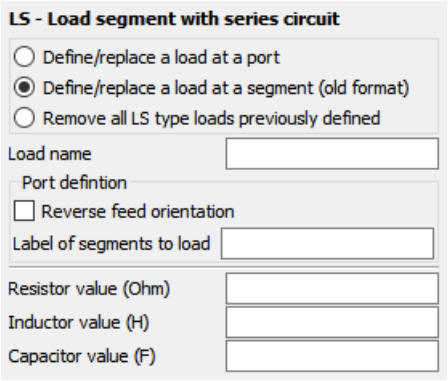


*Figure 917: The **LS - Load segment with series circuit** dialog.*

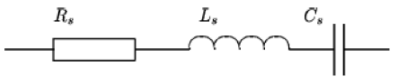


Figure 918: Sketch of the series circuit.

Parameters:

Define/replace a load at a port	Define/replace a series circuit at the port with name **Port** defined using a PT card.
Define/replace a load at a segment (old format)	Define a load with the following parameters.
Remove all LP type loads previously defined	This LS card does not define a load, but rather all previously defined LS loads are deleted. All the other input parameters of this card are ignored.
Load name	The name of the load.
Reverse feed orientation	By default, the vector of the voltage is orientated in the direction from the start of the segment to its end (the direction in which it was created). When this option is checked, the vector of the voltage is orientated in the opposite direction. (This is the direction of the current flow through the segment. The internal EMF (electromagnetic force) of the impressed voltage source is in the opposite direction.
Label of segments to load	All segments with this label are assigned the series circuit values specified below.

Resistor value	Value of the resistor in Ω.
Inductor value	Value of the inductor in H.
Capacitor value	Value of the capacitor in F.

The impedance is given by

$$Z_S = R_S + j\omega L_S + \frac{1}{j\omega C_S} \tag{208}$$

If a capacitance of zero is selected, it is interpreted as infinite capacitance, therefore in the case of the series circuit it is zero.

The LS card may be combined with the LD, LP, LZ and the SK cards, but only one LS card may be used per label. If a second LS card is used, it replaces the values entered by the first one. This card has no significance for surface elements, even when these are assigned the same label.

Related tasks
Adding a Load (CADFEKO)
Related reference
LD Card
LP Card
LZ Card
PT Card
SK Card

LT Card

This card assigns a resistor, inductor or capacitor in series to a voxel mesh for the finite difference time domain (FDTD) method.

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Port load (LT)** icon.

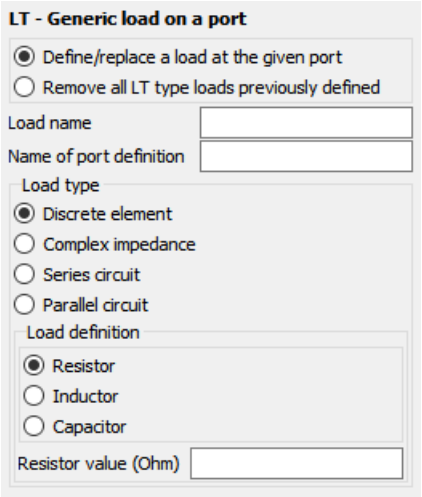


*Figure 919: The **LT - Generic load on a port** dialog.*

Parameters:

Define/replace a load at the given port	Define a load or replace a load at the given port with the parameters specified below.
Remove all LT type loads previously defined	This LT card does not define a load, but rather all previously defined LT loads are deleted. All the other input parameters of this card are ignored.
Load name	The name of the load.
Name of port definition	The name of the port.
Discrete element	A single resistor, inductor or capacitor is added to the port.
Resistor	Value of the resistor in Ω.
Inductor	Value of the inductor in H.
Capacitor	Value of the capacitor in F.
Complex impedance	The real and imaginary part of the complex impedance in Ω.
Series circuit	The resistor value in Ω , inductor value in Henry and the capacitor value in Farad to be added as a series circuit.

Parallel circuit	The resistor value in Ω , inductor value in Henry and the capacitor value in Farad to be added as a parallel circuit.
Resistor value (Ohm)	Value of the resistor in Ω.

Related tasks
Adding a Load (CADFEKO)

LZ Card

This card can be used to assign a complex impedance to a segment.

On the **Source/Load** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list, click the **Complex load (LZ)** icon.

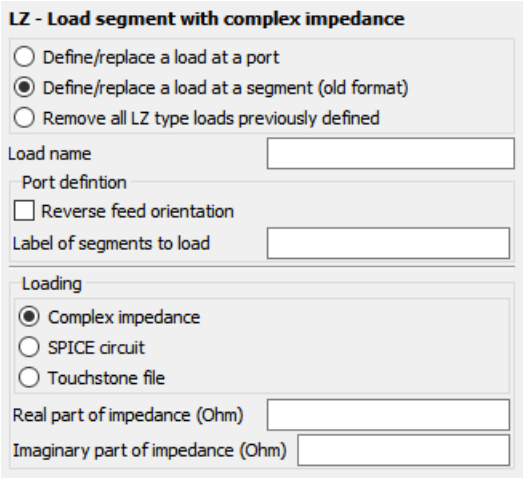


*Figure 920: The **LZ - Load segment with complex impedance** dialog.*

Parameters:

Define/replace a load at a port	Define/replace a complex impedance load at the port with name **Port** defined using a PT card.
Define/replace a load at a segment (old format)	Define a load with the following parameters.
Remove all LZ type loads previously defined	This LZ card does not define a load, but rather all previously defined LZ loads are deleted. All the other input parameters of this card are ignored.
Load name	The name of the load.
Reverse feed orientation	By default, the vector of the voltage is orientated in the direction from the start of the segment to its end (the direction in which it was created). When this option is checked, the vector of the voltage is orientated in the opposite direction. This is the direction of the current flow through the segment. The internal EMF (electromotive force) of the impressed voltage source is in the opposite direction.
Label of segments to load	All segments with this label are assigned the complex circuit values specified below.
Complex impedance	The real and imaginary part of the complex impedance in Ω.

SPICE circuit	Specify the name of a one-port SPICE circuit to define a load between two pins. Define the SPICE circuit using the SC card.
Touchstone file	Specify a one-port Touchstone file (`.s1p`, `.z1p`, `.y1p`) to define a load.

> **Note:** If the load is added to a port that has a voltage source, the load is placed in series with the voltage source.

The complex impedance value is a constant with respect to frequency. Frequency dependent impedances can be realised using the LS or the LP cards.

The LZ card may be combined with the LD, LP, LS and the SK cards, but only one LZ card may be used per label. If a second LZ card is used, it replaces the values entered by the first one. This card has no significance for surface elements, even when these are assigned the same label.

Related tasks
Adding a Load (CADFEKO)
Related reference
LD Card
LP Card
LS Card
PT Card
SK Card

MD Card

This card exports the model and solution coefficients to a `.sol` file.

On the **Request** tab, in the **Solution requests** group, click the **Model decomposition (MD)** icon.

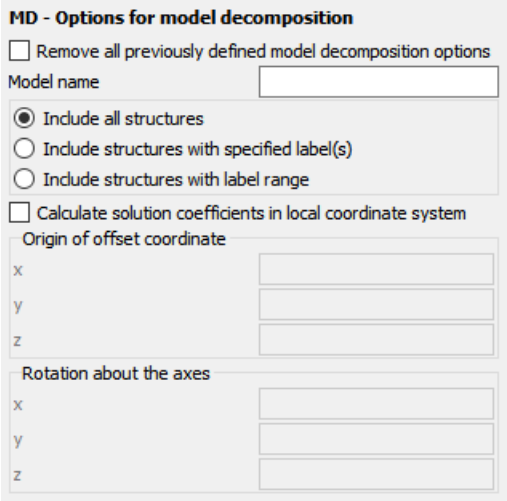


*Figure 921: The **MD - Options for model decomposition** dialog.*

A `.sol` file can be used to specify an impressed current source (AM card).

Parameters:

Remove all previously defined model decomposition options
All previously defined model decomposition options are removed.

Model name
The name of the model.

Include all structures
Model decomposition is done for all structures.

Include structures with specified labels(s)
Model decomposition is only done for structures with specified labels.

Include structures with label range
Model decomposition is only done for structures with a label in the range specified in the fields **Start at label** and **End at label**.

Calculate solution coefficients in local coordinate system
Select this option to calculate the model decomposition in a local coordinate system. Define the local coordinate system by specifying the offset from the origin and the rotation about the axis.

- **Origin of offset coordinate** — Specify the Cartesian coordinates of the transformed origin. These values are affected by the scale factor of the SF card if used.

Rotation about the axis	The angle of rotation a_x around the X axis, the angle of rotation a_y around the Y axis and the angle of rotation a_z around the Z axis in degrees.

Related reference

AM Card

SF Card

NW Card

This card defines a linear non-radiating network.

On the **Source/Load** tab, in the **Loads / networks** group, click the **General network (NW)** icon.

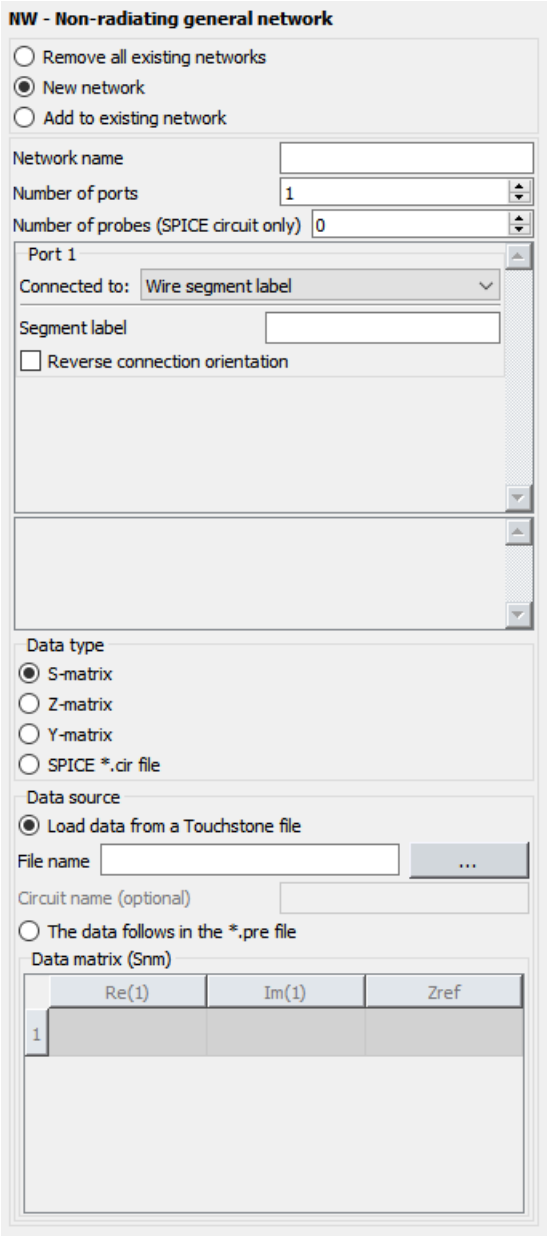


*Figure 922: The **NW - Non-radiating general network** dialog.*

The non-radiating general network provides functionality for combined analysis of electromagnetics with linear circuits (such as amplifiers, filters, matching networks). It is therefore possible to reduce computation time by breaking large problems into smaller element blocks. Cascading the solution of these blocks (represented by S-, Z- or Y-parameters), direct modelling of passive circuits using SPICE

and combining with Feko geometry, the complete (combined) problem solution can be found. The individual element solutions defined with the NW card neglect field coupling.

Parameters:

Remove all existing networks	All previously defined non-radiating networks are removed
New network	A new non-radiating network is created after removing all previously defined networks.
Add to existing network	A non-radiating network is created and added to any previously defined networks.
Network name	The name of the network.
Number of ports	A network can consist of any number of ports, but is required to have at least one port.
Number of probes (SPICE circuit only):	The number of current/voltage probes to be added to the network. A choice is given between a **Voltage probe** and a **Current probe**. The **Probe name** is the name of the defined probe.
Port n	Each port of a network can be connected to other network ports or geometry. Note that the orientation of a network port connection can easily be reversed for all connections except if connected to internal ports.

Wire segment label	The label of the segment to which the network port must be connected. If more than one segment has this label, the network port is connected to the last segment with this label.
Wire segment position	The segment is determined by specifying the Cartesian coordinates of the segment centre. These values are in metres and are scaled by the SF card if **Modify all dimension related values** is checked.
Internal port	The network name and the network port number to connect to.
Edge between regions with multiple labels	The positive and negative labels that define the edge where the network port has to be connected.
Edge connected to ground/UTD	The positive or negative labels that define the edge where the network port has to be connected.

Edge of microstrip between two points	The points that define the edge of the microstrip line where the network port has to be connected.
Vertex by segment label	The vertex is determined by specifying a segment label. Also select whether the start or end point of that segment should be used.
Vertex by position	The vertex is determined by specifying the Cartesian coordinates of the vertex.
FEM line port position	Input port attached to a FEM line port. The position of the FEM line port is specified by the start point and end point.
Port	The port is attached to a port with label defined using the PT card.

Data type

The network data can be specified with S-parameters, Z-parameters, Y-parameters or a SPICE `.cir` file.

Load data from Touchstone file

The network data can be loaded from a Touchstone file (in v1.1 format). The data in a Touchstone file is always defined in increasing order and at specific frequencies only. These may of course not directly coincide with the frequencies at which the Feko kernel is run. The solution is to interpolate both the magnitude and phase data by using cubic spline interpolation. The Feko frequency is considered out of bounds when it is more than 0.1% away from the lowest/highest frequency defined in the Touchstone file. In such an instance an error will be given and Feko will terminate. If the Feko frequency is within bounds, but not between points, no interpolation will be performed.

Load data from a SPICE file

A passive circuit network can be loaded from a SPICE file.

Absolute port reference

A `.cir` file is to be supplied containing the required SPICE circuit description. This description should include a sub-circuit definition (`..SUBCKT subnam N1 <N2 N3 ..>`) with its name identical to the current NW card name. Its external number of ports should also agree in number with the number of ports defined for this NW card.

Relative port reference

A `.cir` file is to be supplied containing the required SPICE circuit description. This description should include a sub-circuit definition (`..SUBCKT subnam N1p N1m <N2p N2m N3p N3m ..>`) with its name identical to the current NW card

	name. Its external number of ports should be double the number of ports defined for this NW card.
Circuit name (optional)	The sub-circuit name may be specified for SPICE networks.
The data follows in the *.pre file	For small networks with four ports or less, the network matrix can be inserted directly in the .pre file. The matrix is entered as real and imaginary components. S-parameters also require a real reference impedance to be specified for each port.

For more information regarding the placement of a load when a network port is connected to segment, vertex or edge port, refer to Figure 961 and Figure 962.

Connection Guidelines

Note the following guidelines regarding the connections between network ports:

- It is not necessary to specify all possible connections. If, for example, NWName1.Port1 is specified as connected to NWName2.Port1, it is not necessary to specify the reverse, that is the connection from NWName2.Port1 to NWName1.Port1. You should ensure that sufficient information is available to link all connected ports.
- If an internal port should be left open, then no connection should be entered.

Related tasks

Adding a General Network - Data Matrix (CADFEKO)

Adding a General Network - SPICE (CADFEKO)

OF Card

This card specifies an offset for the origin of the coordinate system for near and far field calculations. It also facilitates using only a part of the structure through label selection when calculating fields.

On the **Request** tab, in the **Solution requests** group, click the **Request options (OF)** icon.

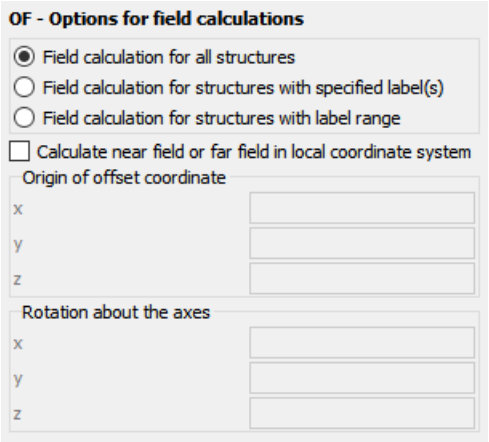


*Figure 923: The **OF - Options for field calculations** dialog.*

Parameters:

Field calculation for all structures
Near and far fields are calculated for all structures.

Field calculation for structures with specified labels(s)
Use label selection when calculating near and far fields. Only the currents on structures with specified labels are used during field computation.

Field calculation for structures with label range
Use label selection when calculating near and far fields. Only the currents on structures with a label in the range specified in the fields **Start at label** and **End at label** are used during field computation. (If a basis function extends over, for example, two triangles it is included if either triangle is in the specified range.)

Calculate near field or far field in local coordinate system
Select this option to calculate fields in a local coordinate system. Define the local coordinate system by specifying the offset from the origin and the rotation about the axis.

- **Origin of offset coordinate**: Specify the Cartesian coordinates of the transformed origin. These values are affected by the scale factor of the SF card if used.
- **Rotation about the axis**: The angle of rotation a_x around the X axis, the angle of rotation a_y around the Y axis and the angle of rotation a_z around the Z axis in degrees.

A possible application of the OF card is, for example, to calculate the near field on the surface of a sphere for which the centre is not positioned at the origin. The OF card transforms the origin of the coordinate system to the centre of the sphere, so that the near field calculation can be executed in spherical coordinates.

For monostatic RCS calculations, the plane wave origin and rotation settings take precedence over that defined by the OF card.

Related concepts

Advanced Settings for Near Field (CADFEKO)

Advanced Settings for Far Field (CADFEKO)

Related reference

FE Card

FF Card

SF Card

OM Card

This card can be used to calculate the weighted set of orthogonal current-modes which are supported on a conducting surface.

On the **Request** tab, in the **Configurations** group, click the **Characteristic modes** icon.

Characteristic mode analysis allows for a systematic CEM design approach, provides physical insight regarding antenna operating principles, can determine the resonating frequency of specific modes and determine the optimum feeding arrangements to excite these modes.

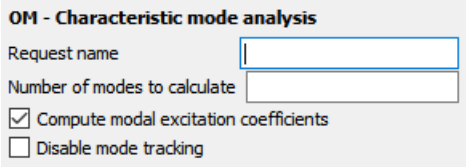


*Figure 924: The **OM - Characteristic mode analysis** dialog.*

Parameters:

Request name	The name of the request.
Number of modes to calculate	The maximum number of modes to calculate for the characteristic mode analysis.
Compute modal excitation coefficients	When this item is checked, in addition to the characteristic modes, modal excitation coefficients are computed given an excitation/ source.
Disable mode tracking	When this item is checked, characteristic mode tracking is disabled. This means that for each frequency, the modes will be sorted according to their dominance at that frequency. When unchecked, a logical mode will be tracked over the entire simulated frequency range.

The characteristic mode analysis (CMA) is supported for MoM/SEP examples containing dielectric and magnetic materials and metallic structures, such as metallic triangles and wires. CMA can also be used in conjunction with the planar multilayered Green's function as well as the planar Green's function aperture. No VEP, waveguide ports, MLFMM or FEM are allowed in conjunction with a characteristic mode analysis request.

Related tasks
Adding a Characteristic Mode Configuration (CADFEKO)

OS Card

With this card the currents on the surfaces and the segments can be extracted.

On the **Request** tab, in the **Solution requests** group, click the **Currents (OS)** icon.

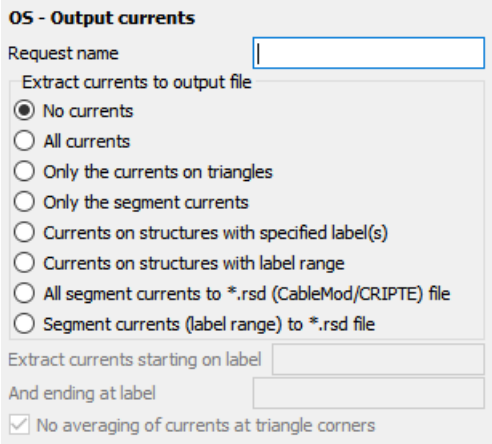


*Figure 925: The **OS - Output currents** dialog.*

Parameters:

Request name	The name of the request.
No currents	No current output (but does start calculation).
All currents	Output all currents on triangles (metallic and dielectric).
Only the currents on triangles	Only output the currents on surface triangles.
Only the segment currents	Only output the currents on wires.
Currents on structures with specified label(s)	Output the currents on segments and triangles with the specified labels.
Currents on structures with label range	Output all currents on segments and triangles in the label range specified by the fields **Extract currents starting on label** and **And ending at label**.
All segment currents to *.rsd (CableMod/CRIPTE) file	Export the currents on all segments to a `.rsd` file in CableMod/CRIPTE/PCBMod format (see the comment below).
Segment currents (label range) to *.rsd file	Export the currents on all segments with labels in the range specified by the fields **Extract currents starting on label** and **And ending at label** to a `.rsd` CableMod/CRIPTE/PCBMod file.
No averaging of currents at triangle corners	For the output of the magnitude of current densities at the vertices of triangles, neighbouring triangles with common vertices are identified and the current densities are then averaged over the neighbours. This ensures that a graphical representation of

the magnitude of the current density (found in the `.out` and `.os` files) is a smooth colour representation without discontinuities. Note that this setting has no effect on the graphical representation in POSTFEKO, the magnitude of the current density in POSTFEKO is always averaged. Averaging of the current densities at the vertices could potentially be very time consuming, particularly for structures containing a large number of triangles.

Multiple OS cards can be used to extract currents on multiple, non-consecutive, labels. The options where a `.rsd` file is written permit the creation of a `.rsd` file for use with the transmission line simulation programs CableMod or CRIPTE or the PCB tool PCBMod. The currents along all or selected segments are exported to the `.rsd` file (the file name without extension is the same as that of the `.fek` file). The `.rsd` file is an ASCII file and contains first a description of the geometry of the line, followed by blocks with the current information for each frequency. It can be read by CableMod, CRIPTE or PCBMod and can also be imported back into Feko to realise an impressed line source (see the AC card).

If the current of dielectric triangles (surface current formulation) is to be output by the OS card, both the equivalent electric and magnetic surface currents of the external problem are written to the output file. (The currents of the internal problem are different to those of the external problem only in that their sign is reversed.)

If requested by the DA card, an `.os` file will be created in addition to the currents written to the output file.

Related tasks
Requesting Currents (CADFEKO)
Related reference
AC Card
DA Card

PP Card

This card defines the phase shift of the excitation between one unit cell and the next for periodic boundary conditions. The unit cell for a PBC calculation is specified with the PE card.

On the **Home** tab, in the **Planes / arrays** group, click the **Periodic boundary** icon. From the drop-down list, click the **Periodic phase (PP)** icon.

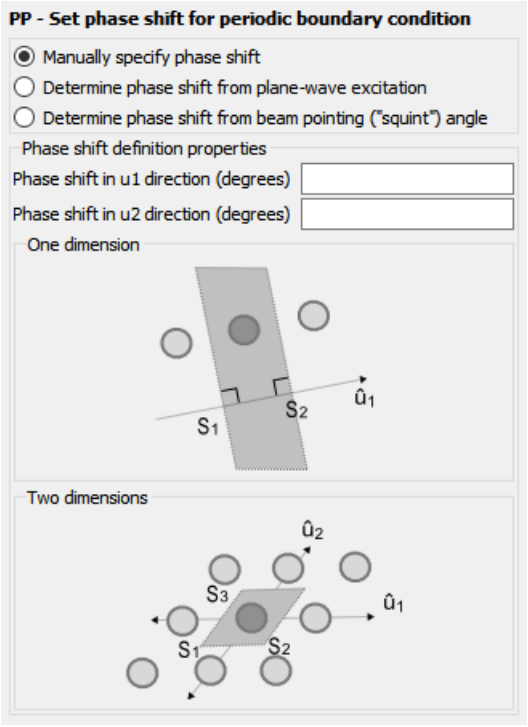


*Figure 926: The **PP - Set phase shift for periodic boundary condition** dialog.*

Parameters:

Manually specify phase shift	The phase shift is manually specified.
Determine phase shift from plane-wave excitation	When a plane wave is used as excitation the phase difference between the cells cannot be specified, but is determined by the excitation.
Determine phase shift from beam pointing ("squint") angle:	The phase shift is determined by specifying the theta and phi angle of the "squint" angle.
Phase shift in $\hat{u}_1$ direction (degrees)	Phase shift in the first direction, $\hat{u}_1$.
Phase shift in $\hat{u}_2$ direction (degrees)	Phase shift in the second direction, $\hat{u}_2$.

One dimension:

Theta angle (degrees)
Orientation of the squint angle. The angle, theta, in degrees is the angle between the squint angle and the plane defined by the $\hat{u}_1$ vector.

Two dimensions:

Theta angle (degrees)
Orientation of the squint angle. The angle, theta, in degrees is the angle between the squint angle and the $\mathbf{u}_1 = 0$ plane.

Phi angle (degrees)
Orientation of the squint angle. The angle, phi, in degrees is the angle between the squint angle and the plane defined by the $\hat{u}_1$ vector.

Note that multiple PP cards can be used in a model but only one PE card can be used.

Related tasks
Defining PBC (CADFEKO)

PR Card

This card defines a voltage or current probe along a cable.

In the **Solve/Run** tab, in the **Cables** group, click the **Probe (PR)** icon.

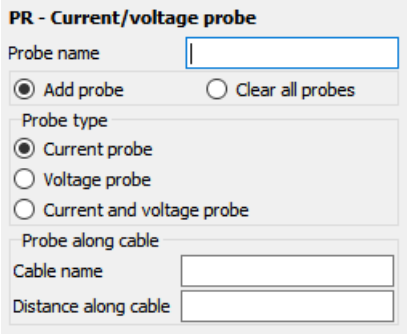


*Figure 927: The **PR - Current/voltage probe** dialog .*

Parameters:

Add probe		A probe is defined which is added to the previously defined probes.
Clear all probes		All previously defined probes along a cable path are removed.
Probe type	**Current probe**	A current probe is defined.
	Voltage probe	A voltage probe is defined.
	Current and voltage probe	A current and voltage probe is defined.
Probe along a cable	**Cable name**	The name of the cable path to which the probe will be added.
	Distance along cable	Position along the cable path from the start connector at which to monitor the probe.

Related tasks
Requesting Cable Probe Data (CADFEKO)
Related reference
CI Card
NW Card

PS Card

This card is for general program control such as storing the currents for re-use in a subsequent version of the model in order to save runtime.

On the **Request** tab, in the **Output control** group, click the **Data structures (PS)** icon.

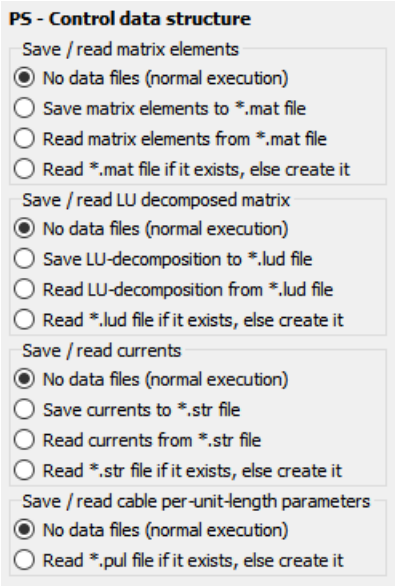


*Figure 928: The **PS - Control data structure** dialog.*

It is important to be familiar with the solution process of the MoM to understand this card. The solution of electromagnetic problems based on the MoM involves a setting up a system of linear equations, which by default is solved using an LU-decomposition and a subsequent backwards substitution. This card can be used to save the matrix of the system of linear equations, its LU decomposition, or the solution vector (which also includes PO currents and so forth). Such elements can also be loaded again.

This card can also be used to save the cable per-unit-length parameters between frequency runs to prevent the parameter recalculation for every frequency.

Parameters:

Save / read matrix elements
Select this option to save or read the matrix elements of the system of linear equations. This is typically not recommended, since the file will be large and the time to recompute the matrix elements is typically much shorter than the time for the LU-decomposition of the matrix.

Save / read LU decomposed matrix
Select this option to save or read the LU-decomposition of the system of linear equations. This option is useful if you want to solve a problem repeatedly with multiple different excitations (right-hand sides). Note that if you do this in one Feko run (that is one `.pre` file), then Feko keeps the LU-decomposition automatically in memory.

Save / read currents	Select this option to save or read the solution vector of the system of linear equations. The solution vector corresponds to the currents on the structure being simulated.
Save / read cable per-unit-length parameters	Select this option to save or read the cable per-unit-length parameters.

Feko always uses the most efficient computation when doing multiple solutions in one file. However, sometimes one might also do a solution, look at the results and then change certain parameters. Then the option to store the solution in the `.str` file and load it again or the similar option for the LU-decomposition are particularly useful. The option to save the currents is applicable when the solution remains unchanged (such as the geometry, material parameters, loads, frequency and sources), but when one wants to compute the near- or far fields with different options. The storage of the LU-decomposition is useful when only the right-hand side of the system of linear equations changes (for example the direction of incidence of a plane wave). Feko has built-in checks and reports a warning if, for example, currents were exported for one frequency and are later imported again for another frequency.

Note that the `.str` file can be used for MoM, MLFMM, PO, and FEM, while the `.mat` and `.lud` files are only applicable when the standard MoM is used. The `.mat` file can only be stored/read for sequential in-core solutions.

Note that models built with PREFEKO on different computers may not be identical due to very small rounding differences of different CPUs. It is therefore advisable to run PREFEKO only on one computer when using this card, to ensure consistency in the `.fek` files. The `.fek` files can then be copied to another computer if required. (For example, a user may calculate and store the current distribution for a large model on a fast workstation and later load this to calculate different near fields on a small PC. To ensure that the current solution is valid on the PC, the original `.fek` file should be generated on the PC and copied to the workstation.)

If the PS card is used to specify that data should be read from a file, the PS card may occur only once in the input file.

> **Tip:** Place the PS card right after the EG card.

Related concepts
Store and Reuse Solution Files (CADFEKO)

PW Card

The PW card specifies the radiated power or the source power. It can also be used to consider mismatch.

On the **Source/Load** tab, in the **Settings** group, click the **Power (PW)** icon.

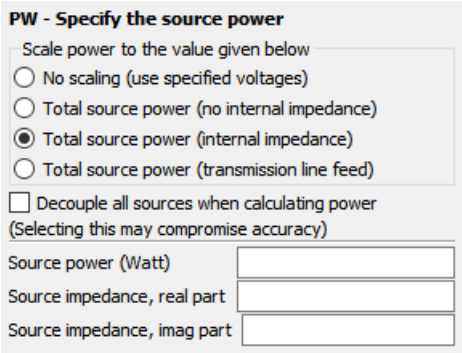


*Figure 929: The **PW - Specify the source power** dialog.*

When defining the excitation of an antenna, the source is normally specified as a complex voltage or current. By specifying the radiated/source power, Feko just scales the whole solution to arrive at the specified power. In addition a mismatch between the antenna input impedance and the internal impedance of the source or the characteristic impedance of a transmission line feed can be considered.

Parameters:

No scaling (use specified voltages)
PW card is not activated meaning the specified value of the voltage or current source is used.

Total source power (no internal impedance)
The PW card is active and the entire solution is scaled by a scaling factor so that the total source power (the sum of the power delivered by all the individual sources) is P_0 — the value specified in the **Source power (Watt)** field. Mismatch is not considered.

Total source power (internal impedance)
All voltage and/or current sources are assumed to have an internal impedance Z_i as specified by the parameters **Source impedance, real part** and **Source impedance, imag part**. The solution coefficients are scaled such that the total power supplied by the sources equals P_0 as discussed below. The mismatch loss in the source internal impedance reduces the antenna gain.

> **Note:** The solution coefficients, for example, refers to the current coefficients for surface triangles in a MoM or PO solution. These can also refer to the electric field coefficients in a FEM solution.

Total source power (transmission line feed)
All the antennas are assumed to be fed by transmission lines with a complex characteristic impedance Z_L as specified by the parameters **Charact. impedance, real part** and **Charact.**

	impedance, imag part. If there is a mismatch between Z_L and the antenna input impedance Z_a, a fraction of the incident power will be reflected back to the source.
Decouple all sources when calculating power	When this item is not checked and multiple impressed sources (that is, elementary dipoles with the A5/A6 cards, or impressed current elements with the AI/AV cards, and so forth) are present, the mutual coupling of all these sources, as well as the coupling of the sources with other structures such as ground (BO card), UTD surfaces, or MoM elements are taken into account when determining the source power. This is also the default if the PW card is not present. However, when this item is checked the mutual coupling is not considered. Ignoring the mutual coupling is acceptable when sources are spaced far apart or when accurate power values are not required. (Since gain and directivity are based on power, these values are then also possibly not very accurate.)
Source power	The total power P_0 in Watt supplied by all the voltage and/or current sources. In the case of transmission lines it is the total power of all forward travelling waves.

Details of the various possibilities with the use of the PW card are shown below.

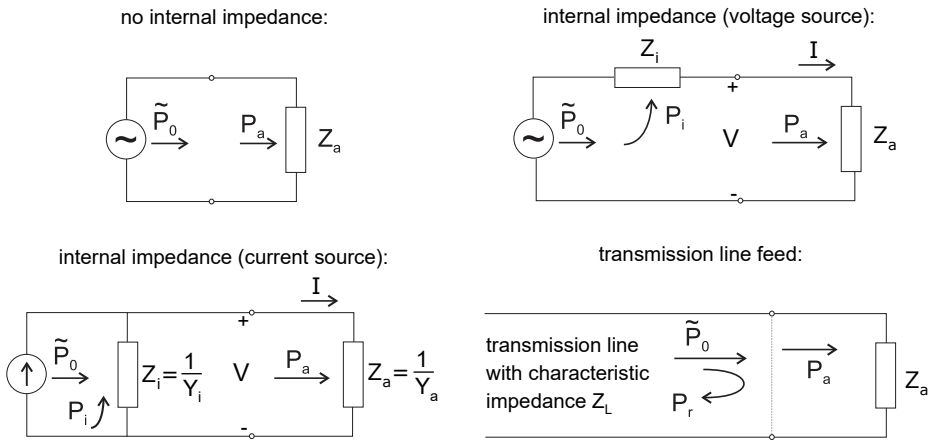


Figure 930: Possible applications of the PW card to determine the total power.

The options **Total source power (internal impedance)** and **Total source power (transmission line feed)** are allowed for voltage sources (the A1, A2, A3, A7, AE and AN cards) and the current source (AF card) which is used in a FEM dielectric. For models containing other sources such as elementary dipoles and impressed currents (AI, AV and AC cards), the option **Total source power (no internal impedance)** should be used. For plane waves **No scaling (use specified voltages)** should be used.

The power equations for different cases are discussed below. Consider, in general, that there are N voltage and/or current sources (such as in an array antenna) with open circuit voltages $U_{0,n}$ and short

circuit currents $I_{s,n}$ (before the scaling operation) where the parameter n is in the range 1. . .N. At each source there is an antenna input impedance $Z_{a,n}$ (as calculated during the Feko solution) to which power $P_{a,n}$ is transferred.

- **Total source power (no internal impedance)**:

  Using this option all the source power is delivered to the respective antennas, that is

  $$\tilde{P}_{0,n} = P_{a,n} \tag{209}$$

  as shown in the top left of the figure. To ensure that the total power is P_0, the power must be scaled with the factor

  $$S = \frac{P_0}{\sum_{n=1}^{N} \tilde{P}_{0,n}} = \frac{P_0}{\sum_{n=1}^{N} P_{a,n}} \tag{210}$$

  The currents on the structure are consequently scaled with the factor $\sqrt{S}$. There is no power loss.

- **Total source power (internal impedance)**:

  When this option is used the internal impedance Z_i of the voltage or current source is considered as shown in the top right and bottom left of the figure.

  For voltage sources the same current flows through the internal source impedance and the antenna input impedance. For the current source (AF card) the same voltage is applied across the internal source impedance and the antenna input impedance.

  The power dissipated in the impedance of the v^{th} voltage source is given by the relation

  $$P_{i,v} = P_{a,v} \frac{\mathrm{Re}\left(Z_{i,v}\right)}{\mathrm{Re}\left(Z_{a,v}\right)} \tag{211}$$

  The power dissipated in the impedance of the m^{th} current source is given by the relation

  $$P_{i,m} = P_{a,m} \frac{\mathrm{Re}\left(Y_{i,m}\right)^*}{\mathrm{Re}\left(Y_{a,m}\right)^*} \tag{212}$$

  The scaling factor S (to scale the total power supplied by the voltage and or current sources to P_0) is

  $$S = \frac{P_0}{\sum_{v=1}^{N_V} P_{a,v}\left(1 + \frac{\mathrm{Re}\left(Z_{i,v}\right)}{\mathrm{Re}\left(Z_{a,v}\right)}\right) + \sum_{m=1}^{N_m} P_{a,m} \frac{\mathrm{Re}\left(Y_{i,m}\right)^*}{\mathrm{Re}\left(Y_{a,m}\right)^*}} \tag{213}$$

  The combined loss caused by the mismatched antennas,

  $$P_{loss} = S\left(\sum_{v=1}^{N_V} P_{i,v} + \sum_{m=1}^{N_m} P_{i,m}\right) = S\left(\sum_{v=1}^{N_V} P_{a,v} \frac{\mathrm{Re}\left(Z_i\right)_v}{\mathrm{Re}\left(Z_a\right)_v} + \sum_{m=1}^{N_m} P_{a,m} \frac{\mathrm{Re}\left(Y_{i^*}\right)_m}{\mathrm{Re}\left(Y_{a^*}\right)_m}\right) \tag{214}$$

  reduces, for example, the antenna gain (but not the directivity).

- **Total source power (transmission line feed)**:

When this option is used each antenna (with input impedance $Z_{a,n}$) is considered to be excited by a transmission line with a complex characteristic impedance Z_L as shown in the bottom right figure. For most practical applications the transmission line will be lossless, resulting in a real characteristic wave impedance. For this lossless case the reflection factor is

$$\varrho_n = \frac{Z_{a,n} - Z_L}{Z_{a,n} + Z_L} \tag{215}$$

is taken into account when calculating the incident power at the feed point.

The total incident power for the n^{th} source is given by

$$\tilde{P}_{0,n} = \frac{P_{a,n}}{1 - |\varrho_n|^2} \tag{216}$$

and the reflected power by

$$P_{r,n} = |\varrho_n|^2 \tilde{P}_{0,n} = P_{a,n} \frac{|\varrho_n|^2}{1 - |\varrho_n|^2} \tag{217}$$

To ensure that the total incident power is P_0, the power is scaled with the factor

$$S = \frac{P_0}{\sum_{n=1}^{N} \tilde{P}_{0,n}} = \frac{P_0}{\sum_{n=1}^{N} \frac{P_{a,n}}{1 - |\varrho_n|^2}} \tag{218}$$

and the currents with the factor $\sqrt{S}$. As before the total reflected power

$$P_{loss} = s \sum_{n=1}^{N} P_{r,n} = s \sum_{n=1}^{N} P_{a,n} \frac{|\varrho_n|^2}{1 - |\varrho_n|^2} \tag{219}$$

reduces the gain of the antenna.

In the case of a lossy transmission line, the forward and backward travelling waves can be identified, but a proper definition of the power associated with such a wave is not possible. The power will constantly change along the length of the transmission line due to the losses. It is questionable whether using the PW card with a lossy transmission line makes any sense, but it is supported. In this case the forward travelling power P_0 is interpreted as the maximum available power at the end terminals of the transmission line. For a lossless transmission line this formulation is compatible with the above equations.

Related concepts
Power (CADFEKO)

RA Card

This card defines an ideal receiving antenna that can be placed anywhere in the model.

On the **Request** tab, in the **Solution requests** group, click the **Receiving antenna (RA)** icon.

The options that are supported by Feko are as follows:

- **Far field pattern**: An ideal receiving antenna is described with an impressed radiation pattern.
- **Near field aperture (single or multiple surfaces)**: An ideal receiving antenna is described with near field aperture(s).
- **Spherical modes**: A receiving antenna is described by means of spherical modes with three options.
  - **Far field approximation**: A receiving antenna is described by means of an impressed radiation pattern obtained internally from the spherical modes description.
  - **Spherical mode approximation**: Spherical mode expansions of the fields radiated and received by antennas are related directly to compute the coupling between them.
  - **Automatically determine approximation**: The Solver determines automatically if either the far field or spherical mode approximation should be used for the model.

Far Field Pattern

This option defines an ideal receiving antenna with an impressed radiation pattern.

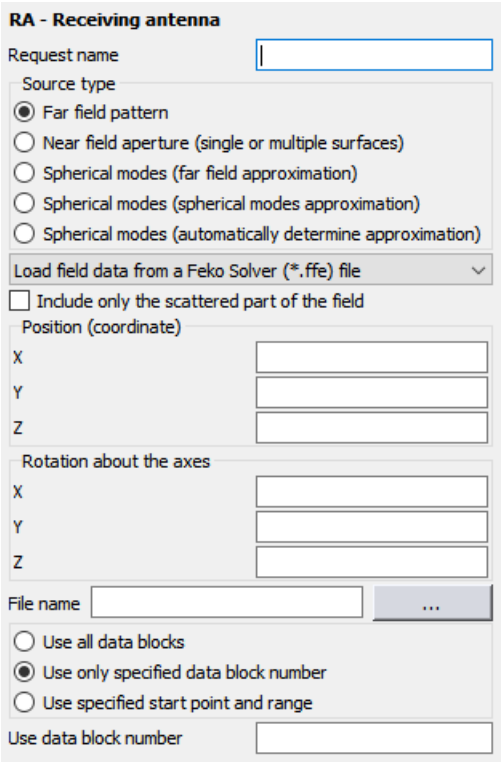


*Figure 931: The **RA - Receiving antenna** dialog.*

Parameters:

Request name	The name of the request.	
Load field data from	**Feko Solver (*.ffe) file**	Read the radiation pattern from an `.ffe` format file, created with DA and FF cards.
	CST far field scan (*.ffs)	Read the radiation pattern from a CST far field scan.
	The field data follows in the (*.pre) input file	The radiation pattern is specified in the `.pre` file following the RA card (the format of this file is described in the AR card). With this option FOR loops can be used to generate patterns from known functions.
	Use last pattern defined at previous RA card	When using multiple RA cards (different receiving antennas in the same model) then it is quite common that the shape of the pattern is identical. If this is the case it is allowed to load the pattern just once and at subsequent RA cards to check this option. Then the last defined pattern will be used and memory can be saved (no need to store it again). Note that it is still possible to set the receiving antenna position and orientation individually.
	external data file	Read the radiation pattern from an ASCII file (the format of this file is described in the AR card).
Include only the scattered part of the field	Consider only the scattered part of the field. If unchecked, the total field, that is the sum of the scattered and source contributions, are considered for calculation.	
Position (coordinate)	In this group the X, Y and Z coordinates of the receiving point (the position where the antenna is placed) are entered in metres. This value is affected by the scale factor of the SF card if used.	
Rotation about the axes	In this group the angles with which the imported pattern is rotated around the X, Y and Z axes are entered in degrees. These fields are referred to as a_x, a_y and a_z in the rest of this discussion.	
File name	The name of the `.ffe`, `.ffs` or ASCII input file.	
Use all data blocks	Import all data blocks from the specified `.ffe` or `.ffs` file. The data is interpolated for use at the operating frequency.	

Use only specified data block number
Use the data from the nth frequency block in the specified `.ffe` or `.ffs` file.

Use specified start point and range
Select a specific far field pattern in a `.ffe` or external file.

- **Start from point number** This parameter is only relevant when the data is read from a `.ffe` or an external file, and gives the line number of the first line to read from the input file. If the data must be read from the beginning of the file, the value in this field should be set equal to 1. This parameter is used when the `.ffe` file contains more than one pattern. For example, if the file contains the pattern at various frequencies, the correct pattern can be selected by setting this field to the appropriate value for each frequency. If the `.ffe` file is of a newer format and contains header data in addition to the data blocks, the point number refers to the actual point number excluding blank lines, comment lines and header lines.
- **Number of ϑ points** The number of ϑ angles in the pattern.
- **Number of φ points** The number of φ angles in the pattern.

The definition of the radiation pattern and the various different input formats supported for the RA card are identical to those of an impressed transmitting antenna (AR card). The Feko Examples Guide has more examples.

The ideal receiving antenna is considered to be decoupled from the model (that is the currents and charges are unchanged by including this antenna in the model) and should ideally be in the far field of radiating structures (so that incident plane wave approximations apply and the radial field component is small). Feko has internal checks to ensure these assumptions are valid.

Feko computes the power received by this ideal antenna assuming a perfect match, that is a complex conjugate load.

The relative phase of the received signal is also printed to the .out file, which indicates the relative phase of either the voltage or the current through the termination impedance (which is an ideal complex conjugate load). When using arrays of identical receiving antennas, then this relative phase information can be used to obtain the signal phase offsets of the various array elements (required for instance in the design of the feed network).

Related tasks
Requesting Receiving Antenna (Far Field Pattern) (CADFEKO)

Near Field Aperture

This option defines an ideal receiving antenna with a near field aperture (single or multiple surfaces).

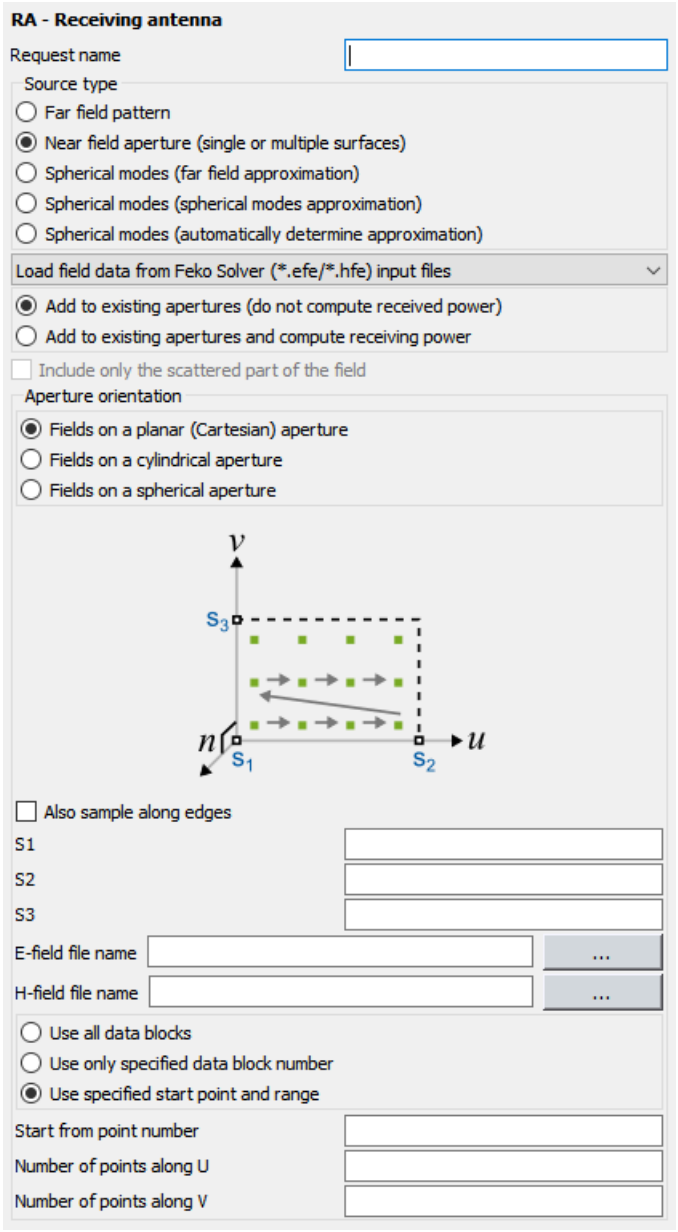


*Figure 932: The **RA - Receiving antenna** dialog.*

Parameters:

Load field data from

Feko Solver (*.efe/*.hfe) file

Read the field values from `.efe`/`.hfe` format files calculated by Feko. The files should contain values describing a single face. The field data files are requested with the DA card.

Cartesian boundary from Feko Solver (*.efe/*.hfe) file	Read the field values from `.efe`/`.hfe` format files calculated by Feko. The files should contain values describing a Cartesian boundary. The field data files are requested with the DA card.
CST near field scan (CST NFS)	Read the field values from CST NFS format files.
Sigrity (*.nfd) input file	Read the field values from a `.nfd` file.
MVG (*.mfxml) measurement file	Read the field values from a `.mfxml` file.
ASCII text file	Read the field values from an ASCII file.
The file data follows in the (*.pre) input file	The field values are specified in the `.pre` file. The format of this file is described in the AR card.

Add to existing apertures (do not compute received power)

This option is selected when a new near field aperture is defined which is added to previously defined apertures. The receiving power for the multiple near field apertures is not computed.

Add to existing apertures and compute receiving power

This option is selected if a single near field aperture or the last of multiple near field apertures is defined. All defined near field apertures defined before and including this one are taken into consideration and the receiving power is computed.

Include only the scattered part of the field

When this item is checked, only the scattered part of the field is considered for calculation. Otherwise the total field, the sum of the scattered and source contributions, are considered for calculation.

Aperture orientation

Fields on a ... aperture	Select a planar, cylindrical or spherical aperture.
Also sample along edges	Check this option to extend the samples up to the edges of the aperture. When it is unchecked, the samples will be offset from the edges of the aperture.
S1, S2 and S3	These text boxes are for input points (see the DP card) that define the orientation of the aperture. The figure on the dialog will depict the orientation. For a planar aperture the input points define the position of the origin and

the direction of the $\hat{u}_2$ and $\hat{u}_3$ directions respectively. (The field data is assumed to vary first along the $\hat{u}_2$ direction.)

> **Note:** The normal n is determined by S1, S2 and S3 and should correspond to the primary direction of power flow as seen when the aperture is in transmit mode. If S1, S2 and S3 are not carefully defined, it may result in negative received power.

For cylindrical and spherical apertures S1, S2 and S3 define the origin and the direction of the local Z axis and X axis respectively. All angles are relative to these axes and are obtained from the `.efe` and `.hfe` files, but the origin is determined from S1, S2 and S3.

E-field file name The input file name from which the electric field data must be read. This may be either an `.efe` file or a text file (with units V/m).

H-field file name The input file name from which the magnetic field data must be read. This may be either an `.hfe` file or a text file (with units A/m).

Use all data blocks Import all data blocks from the specified `.efe`/`.hfe`, `.nfd`, `.mfxml` or `.nfs` file. The data is interpolated for use at the operating frequency.

Use only specified data block number Use the data from the nth frequency block in the specified `.efe`/`.hfe`, `.nfd`. `.mfxml` or `.nfs` file.

Use specified start point and range Select a specific near field pattern in a `.efe`/`.hfe`, `.pre` or ASCII text file.

Start from point number
The number of the first field point to be used for the aperture. If set to 1, field values are read from the start of the file, for larger values the first point number-1 values

(`.efe` and `.hfe` files) or lines (text files) are ignored. This may be used, for example, if the data file contains the field values for more than one frequency. This corresponds to the line number if all non-data lines are stripped from the file. The **Start from point number** field is not used if the field data is obtained from the .pre input file

Number of points along ...
These two text boxes specify the number of field points along each of the two axes of the aperture.

Directory The directory where the CST near field scan (NFS) files are located.

File name The input file name from which the Sigrity (`.nfd`) or MVG (`.mfxml`) files are read.

Include fields on boundary When importing Cartesian boundary near field data from file, a face (or faces) of the near field boundary can be ignored. Clear the relevant check box to exclude the face.

Related tasks
Requesting Receiving Antenna (Near Field Pattern) (CADFEKO)

Spherical Modes

This option defines an ideal receiving antenna with spherical modes.

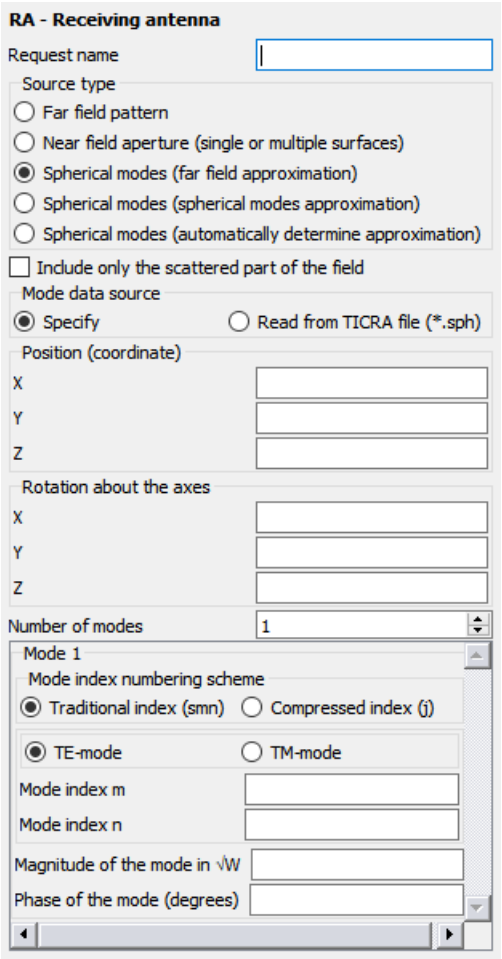


*Figure 933: The **RA - Receiving antenna** dialog.*

Parameters:

Request name — The name of the request.

Mode data source — The spherical modes can be entered directly in the `.pre` file or it can be imported from a TICRA (`.sph`) file.

Position (coordinate) — The coordinates of the origin r = 0 of the mode in metres. These values are optionally scaled by the SF card.

Rotation about the axes — The rotation of the spherical mode source about the X, Y and Z axes.

Number of modes — Specify the number of modes that will be entered in the `.pre` file.

Traditional index (smn) — If this option is selected, you can specify **TE-mode** (s = 1) or **TM-mode** (s = 2) and the indices m and n in the group below. Here n

	is the mode index in the radial direction and must be in the range 1, 2, . . . ∞ and m is the mode index in the azimuth direction φ. There is no distinction between even and odd modes (with $\cos(m\varphi)$ and $\sin(m\varphi)$ angular dependencies), but rather use the angular dependency $e^{jm\varphi}$. Thus the index m can also be negative, but it must be in the range -n . . . n.
Compressed index (j)	With this option, a compressed one-dimensional mode numbering scheme is used. The **Mode index j** is then specified as $$j = 2[n(n+1)+m-1]+s \quad (220)$$ where s = 1 for **TE-mode** and s = 2 for **TM-mode**. This unified mode numbering scheme allows the computation of an extended scattering matrix (with network and radiation ports). This index j then represents a unique port number in the scattering matrix.
Magnitude of the mode in $\sqrt{W}$	Absolute value of the complex amplitude of this specific spherical mode (due to the applied normalisation of the spherical modes, the unit of this amplitude is $\sqrt{W} = \sqrt{VA}$).
Phase of the mode (degrees)	The phase of the complex amplitude of this spherical mode in degrees.
Use all data blocks	Import all data blocks from the specified TICRA (`.sph`) file. The data is interpolated for use at the operating frequency.
Use only specified data block number	Use the data from the nth frequency block in the TICRA (`.sph`) file.

Related tasks

Requesting Receiving Antenna (Spherical Modes Pattern) (CADFEKO)

Related reference

AR Card

DA Card

DP Card

FF Card

SF Card

SA Card

This card controls calculations of the specific absorption rate (SAR) in a dielectric.

On the **Request** tab, in the **Solution requests** group, click the **SAR (SA)** icon.

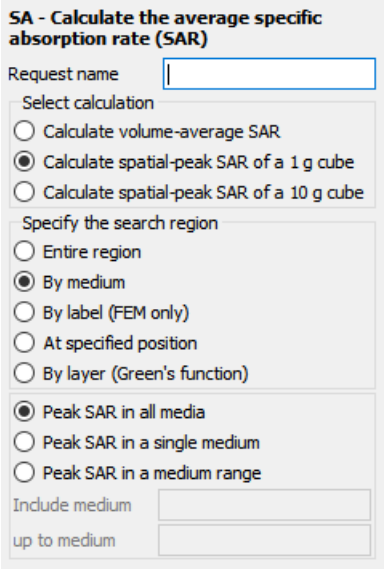


*Figure 934: The **SA - Calculate the average absorption rate (SAR)** dialog.*

Parameters:

Request name	The name of the request.
Select Calculation	One of three SAR values which could be of interest should be selected in this group.
Specify the search region	This group can be used to control, by medium number, label number or layer number (for the special Green's functions), which dielectric bodies are used for the specified calculation. It is also possible to specify a user defined position for the spatial average SAR computations.
Average/Peak SAR in all media/labels/layers	Select this option if the selected SAR calculation should be done on a **By label**, **By medium** or **By layer** basis. The whole body average SAR is also calculated. Selecting the volume by label is only valid for a FEM analysis.
Average/Peak SAR in a single medium/label/layer	The selected SAR calculation is obtained for the medium/label specified in the **Include medium**/**Include label** dialog.
Average/Peak SAR in a medium/label/layer range	The selected SAR calculation is performed on the label range as specified below in the input fields for **Include medium**/**Include label** and **up to medium**/**up to label**.

Centre of SAR cube	For the spatial average SAR computations using a specified position, the X, Y and Z coordinates of the cube centre should be specified here.

The required SAR calculation is performed, and the result saved in the `.out` file.

If the options **Calculate volume-average SAR** and **Entire region** are selected, the SAR, averaged over all media, is returned. If the options **Calculate volume-average SAR** and **By medium** are selected, the average SAR is calculated per medium and tabulated in the `.out` file. If a medium/label/layer range is specified, the SAR is averaged over the volume defined by the medium /label/layer range.

If a spatial-peak SAR calculation is requested, then spatial-peak SAR is computed, averaged over a mass of either 1 g or 10 g of tissue in the shape of a cube. By default, the search for the spatial peak SAR in the entire domain is returned, otherwise the spatial-peak SAR can be requested for regions in a specified medium/label/layer range or also at a user specified position.

When a special spherical or multilayer planar Green's function is used, then also spatial peak average SAR values can be computed (not volume average SAR). A selection is possible by a single layer number, a range of layer numbers, or by including the whole dielectric volume in the search.

Related tasks
Requesting SAR (CADFEKO)

SB Card

This card specifies an external magnetostatic bias field applied to a 3D anisotropic medium (ferrite).

In the **Home** tab, in the **Define** group, click the **Media** icon. From the drop-down list select the **Magnetic bias (SB)** icon.

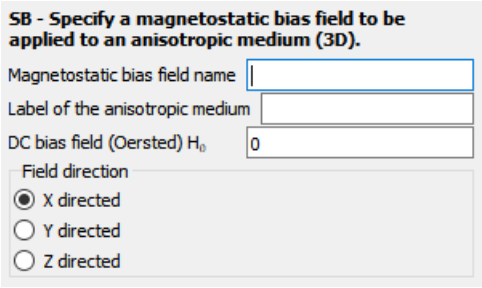


*Figure 935: The **SB - Specify a magnetostatic bias field to be applied to an anisotropic medium (3D)** dialog.*

Parameters:

Magnetostatic bias field name	The name of the magnetostatic bias field name to be defined.
Label of the anisotropic medium	Label of the anisotropic medium (ferrite) to which the magnetostatic bias field is applied.
DC bias field (Oersted)	The magnitude of the DB bias field in Oersted.
Field direction	The direction the magnetostatic bias field is applied to the anisotropic medium (3D).

Related tasks

Creating an Anisotropic Medium - Ferrite (CADFEKO)

SC Card

This card defines a SPICE circuit that can be used when defining a load.

On the **Home** tab, in the **Loads / networks** group, click the **Load** icon. From the drop-down list select the **SPICE circuit (SC)** icon. The circuit can be defined in the `.pre` file or reference a `.cir` file.

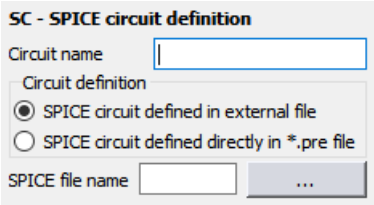


*Figure 936: The **SC - SPICE circuit definition** dialog.*

Parameters:

Circuit name	The name of the SPICE circuit.
SPICE circuit defined in external file	The SPICE circuit is provided as an external file.
SPICE circuit defined directly in *.pre file	The SPICE circuit is included directly in the `.pre` file.

SD Card

This card defines the impedance or admittance of an individual cable shield layer. The shield layers are used when defining a cable shield (SH card).

Define up to two SD cards to specify a single shield layer for a cable shield (SH card). Define up to four SD cards to specify two shield layers for a cable shield (SH card).

Related reference

DI Card

SH Card

Solid (Schelkunoff) Layer

This option defines a shield layer using the solid (Schelkunoff) model.

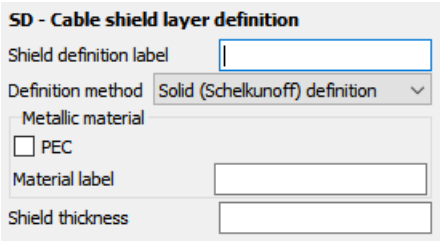


*Figure 937: The **SD - Cable shield layer definition** dialog, set to **Solid (Schelkunoff) definition.***

Parameters

Shield definition label	A name for the shield layer definition.	
Metallic material	**PEC**	Select this option to set the shield of the filaments to PEC.
	Material label	The label of the metallic material (as defined in the DI card) to be used for the shield.
Shield thickness	Define the thickness of the shield layer.	

Braided (Kley) Layer

This option defines a braided shield layer using the Kley model.

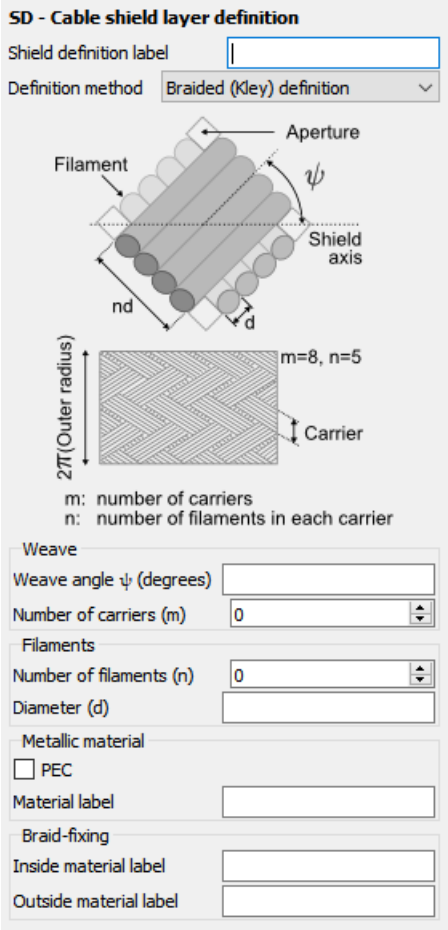


*Figure 938: The **SD - Cable shield layer definition** dialog, set to **Braided (Kley) definition**.*

Parameters

Shield definition label	A name for the shield layer definition.	
Weave definition	**Weave angle ψ (degrees)**	The weave angle of the shield.
Number of carriers (m)	The number of the carriers in the braided shield.	
Number of filaments (n)	The number of filaments in each carrier.	
Diameter (d)	The diameter of the filament.	
Metallic material	**PEC**	Select this option to set the shield of the filaments to PEC.

	Material label	The label of the metallic material (as defined in the DI card) to be used for the shield.
Inside material label		The label of the inside shield braid-fixing (protective) coating material.
Outside material label		The label of the outside shield braid-fixing (protective) coating material.

Braided (Vance) Layer

This option defines a braided shield layer using the Vance model.

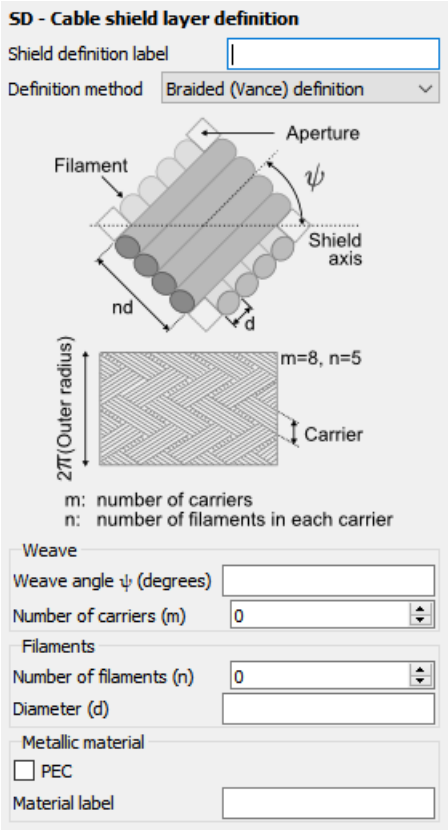


*Figure 939: The **SD - Cable shield layer definition** dialog, set to **Braided (Vance) definition**.*

Parameters

Shield definition label		A name for the shield layer definition.
Weave definition	**Weave angle ψ (degrees)**	The weave angle of the shield.
Number of carriers (m)		The number of the carriers in the braided shield.

Number of filaments (n) The number of filaments in each carrier.

Diameter (d) The diameter of the filament.

Metallic material

- **PEC** Select this option to set the shield of the filaments to PEC.
- **Material label** The label of the metallic material (as defined in the DI card) to be used for the shield.

Braided (Tyni) Layer

This option defines a braided shield layer using the Tyni model.

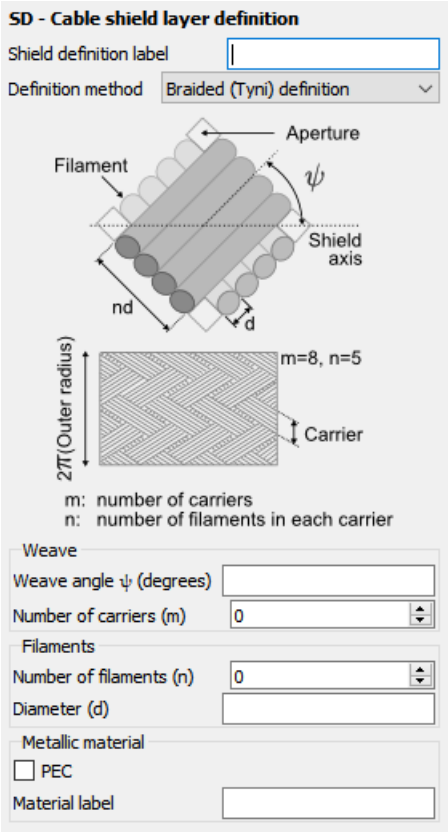


*Figure 940: The **SD - Cable shield layer definition** dialog, set to **Braided (Tyni) definition.***

Parameters

Shield definition label A name for the shield layer definition.

Weave definition

- **Weave angle ψ (degrees)** The weave angle of the shield.

Number of carriers (m)	The number of the carriers in the braided shield.
Number of filaments (n)	The number of filaments in each carrier.
Diameter (d)	The diameter of the filament.
Metallic material	

PEC	Select this option to set the shield of the filaments to PEC.
Material label	The label of the metallic material (as defined in the DI card) to be used for the shield.

Braided (Demoulin) Layer

This option defines a braided shield layer using the Demoulin model.

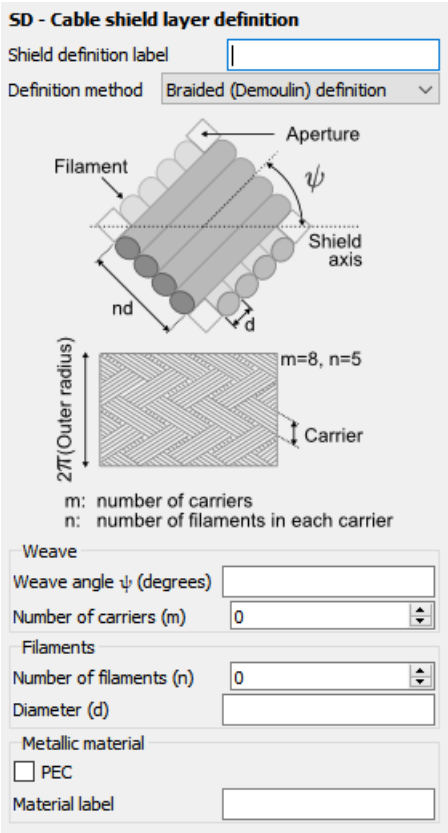


*Figure 941: The **SD - Cable shield layer definition** dialog, set to **Braided (Demoulin) definition.***

Parameters

Shield definition label	A name for the shield layer definition.

Weave definition	**Weave angle ψ (degrees)**	The weave angle of the shield.
Number of carriers (m)	The number of the carriers in the braided shield.	
Number of filaments (n)	The number of filaments in each carrier.	
Diameter (d)	The diameter of the filament.	
Metallic material	**PEC**	Select this option to set the shield of the filaments to PEC.
	Material label	The label of the metallic material (as defined in the DI card) to be used for the shield.

Define Layer Properties

This option defines the shield layer properties by either a manual definition in the .pre file or by loading from file.

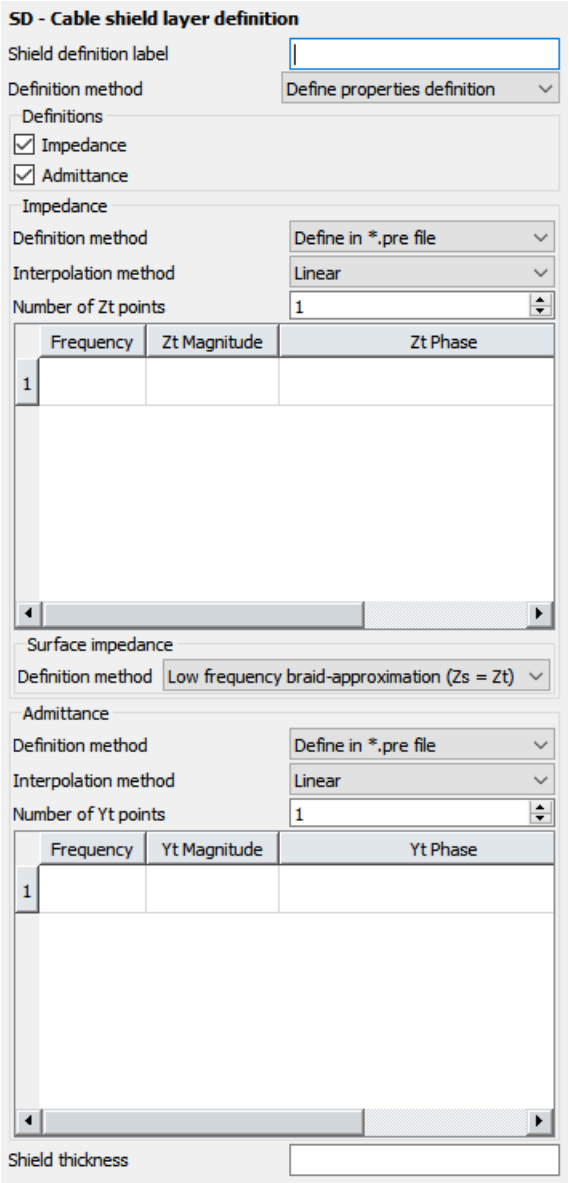


*Figure 942: The **SD - Cable shield layer definition** dialog, set to **Define properties definition.***

Parameters:

Shield definition label	A name for the shield layer definition.
Impedance	Select this option to include transfer and surface impedance definition.
Admittance	Select this option to include the transfer admittance definition.

Define in the *.pre file

The frequency dependent shield parameters based on measured transfer impedance, surface impedance and transfer admittance data can be specified in the .pre file.

Interpolation method	Choose an interpolation method for the data: • Default • Constant • Linear • Cubic spline • Rational (Thiele)
Number of Zt, Zs or Yt points	Total number of frequency points.
Frequency	The frequency at which the transfer impedance, surface impedance or transfer admittance is specified in the .pre file.
Zt, Zs or Yt Magnitude	The magnitude of the transfer impedance, surface impedance or transfer admittance defined in the .pre file.
Zt, Zs or Yt Phase	The phase of the transfer impedance, surface impedance or transfer admittance defined in the .pre file.

Load from file

The file format used when importing the shield properties from file is described in Load Shield Properties from a .XML File.

Surface Impedance - Solid (Metallic material)

PEC	Select this option to set the shield of the filaments to PEC.
Material label of metallic material	The label of the material (as defined in the DI card) to be used for the shield.

Surface Impedance (Zs = Zt)

Low frequency braid-approximation the data defined for the transfer impedance is also used for the surface impedance definition.

Transfer Capacitance

This option specifies the transfer admittance (Yt) of a shield layer in Farad per meter.

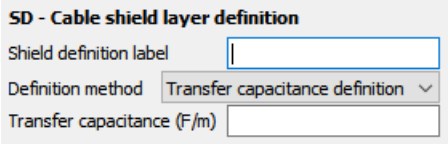


*Figure 943: The **SD - Cable shield layer definition** dialog, set to **Transfer capacitance definition**.*

Parameters

Shield definition label	A name for the shield layer definition.
Transfer capacitance (F/m)	Specify the capacitance in Farad per meter

SH Card

This card defines a cable shield consisting either of a single layer or two layers.

> **Note:** The impedance or admittance of a shield layer is defined using an SD card.

In the **Solve/Run** tab, in the **Cables** group, click the **Cable shield (SH)** icon.

Related tasks

Creating a Vance Cable Shield (CADFEKO)

Creating a Kley Cable Shield (CADFEKO)

Creating a Schelkunoff Cable Shield (CADFEKO)

Single Cable Shield

This option defines a cable shield with a single layer.

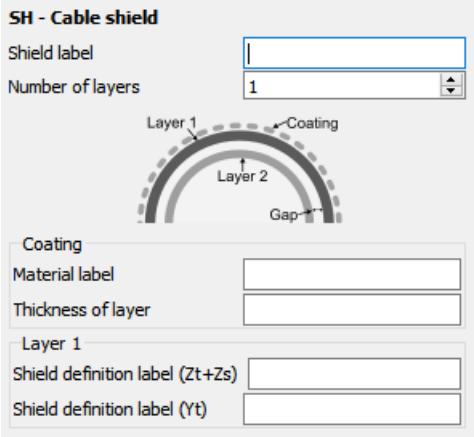


*Figure 944: The **SH - Cable shield** dialog, set to a **single layer**.*

Parameters:

Shield label	The name of the shield.
Number of layers	The total number of layers.
Material label	The label of the dielectric (as defined in the DI card) to be used as the coating for the cable.
Thickness of layer	Define the thickness of the dielectric coating.
Shield definition label (Zt + Zs)	The shield layer (SD card) to be used for the impedance part of layer 1.
Shield definition label (Yt)	The shield layer (SD card) to be used for the admittance part of layer 1.

Double Cable Shield

This option defines a cable shield with two layers.

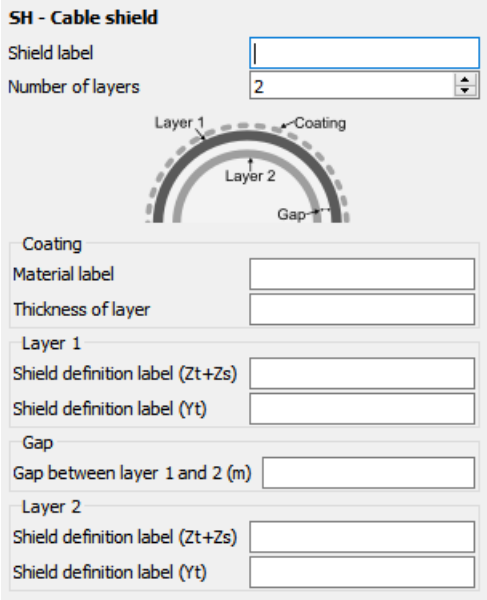


*Figure 945: The **SH - Cable shield** dialog, set to 2 layers.*

Parameters:

Shield label	The name of the shield.
Number of layers	The total number of layers.
Material label	The label of the dielectric (as defined in the DI card) to be used as the coating for the cable.
Thickness of layer for coating	Define the thickness of the dielectric coating.
Shield definition label (Zt + Zs)	The shield layer (SD card) that is to be used for the impedance part of layer 1.
Shield definition label (Yt)	The shield layer (SD card) that is to be used for the admittance part of layer 1.
Gap between layer 1 and 2 (m)	Air (free space) gap between layer 1 and layer 2 in meters.
Shield definition label (Zt + Zs)	The shield layer (SD card) to be used for the impedance part of layer 2.
Shield definition label (Yt)	The shield layer (SD card) to be used for the admittance part of layer 2.

Shield Layer Combinations

When specifying a cable shield, you need SD cards to specify the shield layer impedance or shield layer admittance. Several combinations of SD cards are supported when defining the shield layer impedance and shield layer admittance for a cable shield (SH card).

Table 70: Supported shield layer combinations when specifying the impedance and admittance for a shield layer.

Impedance Definition (Zt + Zs)	Admittance Definition (Yt)
Solid (Schelkunoff)	Not applicable
Braided (Kley)	Same as impedance definition Transfer capacitance Define properties
Braided (Tyni)	Same as impedance definition Transfer capacitance Define properties
Braided (Vance)	Same as impedance definition Transfer capacitance Define properties
Braided (Demoulin)	Same as impedance definition Transfer capacitance Define properties
Define properties	Same as impedance definition Define properties Transfer capacitance

Related reference
Solid (Schelkunoff) Layer
Braided (Kley) Layer
Braided (Vance) Layer
Braided (Tyni) Layer
Braided (Demoulin) Layer
Define Layer Properties
Transfer Capacitance

Solid (Schelkunoff) - Legacy

This option defines a shield using the solid (Schelkunoff) model.

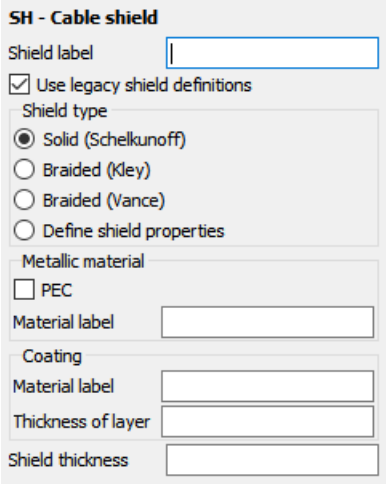


*Figure 946: The **SH - Cable shield definition** dialog, set to **Solid (Schelkunoff)**.*

Parameters:

Use legacy shields definitions — Unselect this option to define a layered shield using shield definitions from an SD card.

Metallic material

- **PEC** — Select this option to set the shield of the filaments to PEC.
- **Material label** — The label of the metallic material (as defined in the DI card) to be used for the shield.

Coating

- **Material label for coating** — The label of the dielectric (as defined in the DI card) to be used as the coating for the cable.
- **Thickness of layer for coating** — Define the thickness of the dielectric coating.

Braided (Kley) - Legacy

This option defines a braided shield using the Kley model.

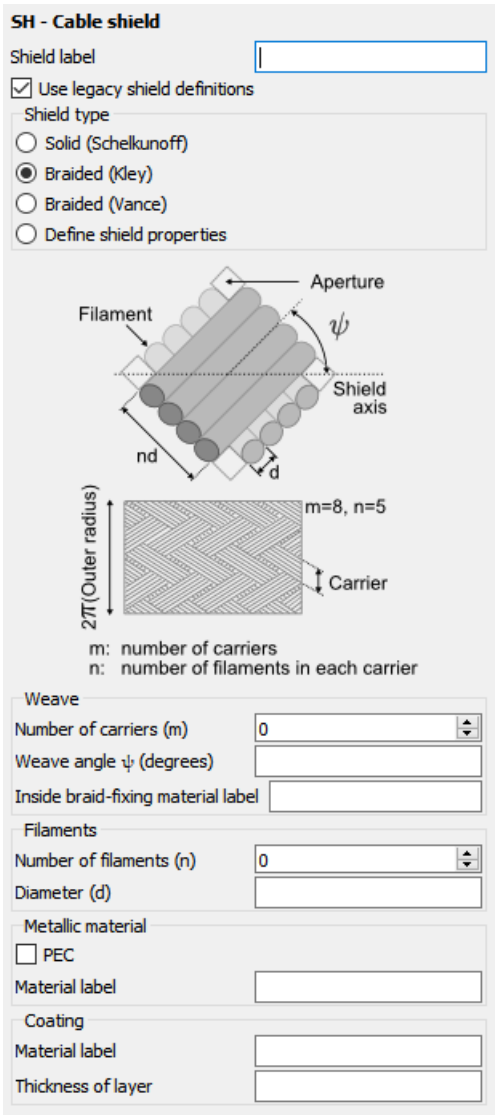


*Figure 947: The **SH - Cable shield definition** dialog, set to **Braided (Kley)**.*

Parameters:

Use legacy shields definitions	Unselect this option to define a layered shield using shield definitions from an SD card.
Number of carriers (m)	The number of the carriers in the braided shield.
Weave angle ψ (degrees)	The weave angle of the shield.
Inside braid-fixing material label	The label of the inside shield braid-fixing (protective) coating material.
Number of filaments (n)	The number of filaments in each carrier.

Diameter (d)	The diameter of the filament.	
Metallic material	**PEC**	Select this option to set the shield of the filaments to PEC.
	Material label	The label of the metallic material (as defined in the DI card) to be used for the shield.
Coating	**Material label for coating**	The label of the dielectric (as defined in the DI card) to be used as the coating for the cable.
	Thickness of layer for coating	Define the thickness of the dielectric coating.

Braided (Vance) - Legacy

This option defines a braided shield using the Vance model.

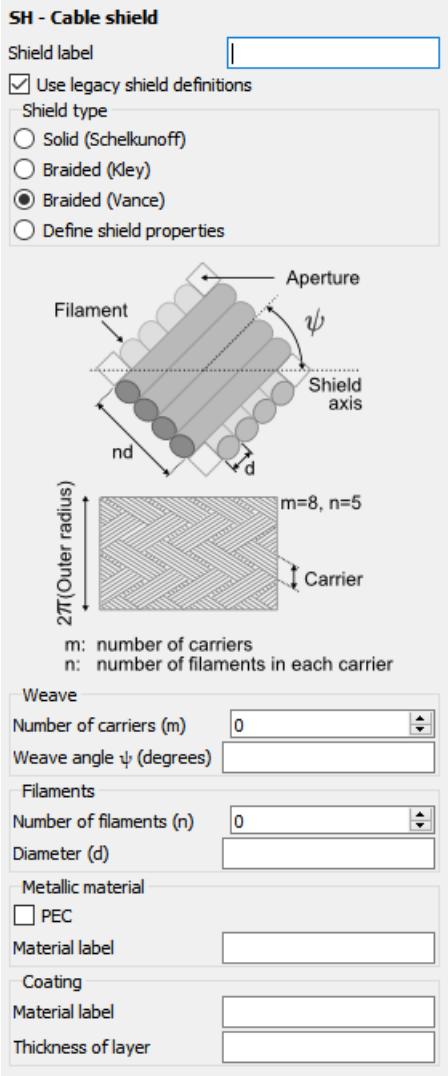


*Figure 948: The **SH - Cable shield definition** dialog, set to **Braided (Vance)**.*

Parameters:

Use legacy shields definitions	Unselect this option to define a layered shield using shield definitions from an SD card.
Number of carriers (m)	The number of the carriers in the braided shield.
Weave angle ψ (degrees)	The weave angle of the shield.
Number of filaments (n)	The number of filaments in each carrier.
Diameter (d)	The diameter of the filament.

Metallic material	**PEC**	Select this option to set the shield of the filaments to PEC.
	Material label	The label of the metallic material (as defined in the DI card) to be used for the shield.
Coating	**Material label for coating**	The label of the dielectric (as defined in the DI card) to be used as the coating for the cable.
	Thickness of layer for coating	Define the thickness of the dielectric coating.

Define Shield Properties - Legacy

This option allows you to define the shield properties by either a manual definition in the .pre file or by loading from file.

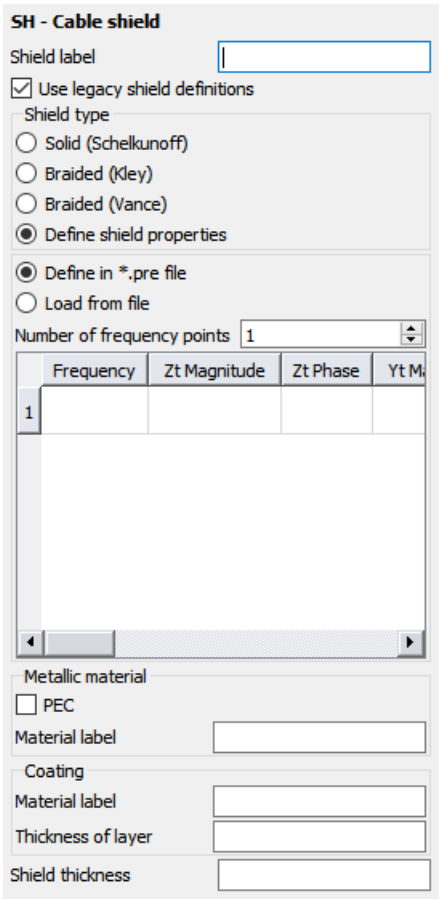


*Figure 949: The **SH - Cable shield definition** dialog, set to **Define in *.pre file**.*

Parameters:

Use legacy shields definitions
Unselect this option to define a layered shield using shield definitions from an SD card.

Define in *.pre file
Define the frequency dependent shield parameters based on measured transfer impedance and admittance data in the .pre file.

Load from file
The file format used when importing measured cable data from file is described in Load Measured Cable Data From File.

Frequency
The frequency at which the transfer impedance and admittance are specified in the .pre file below.

Zt Magnitude - Magnitude of transfer impedance
The magnitude of the transfer impedance defined in the .pre file.

Zt Phase - Phase of transfer impedance
The phase of the transfer impedance defined in the .pre file.

Yt Magnitude - Magnitude of transfer admittance
The magnitude of the transfer admittance defined in the .pre file.

Yt Phase - Phase of transfer admittance
The phase of the transfer admittance defined in the .pre file.

Metallic material

PEC
Select this option to set the shield of the filaments to PEC.

Material label
The label of the metallic material (as defined in the DI card) to be used for the shield.

Coating

Material label for coating
The label of the dielectric (as defined in the DI card) to be used as the coating for the cable.

Thickness of layer for coating
Define the thickness of the dielectric coating.

Shield thickness
The thickness of the shield.

Load Measured Cable Data From File

An example of an XML file containing fictitious measured data:

```
<?xml version="1.0" encoding="UTF-8"?>
<cableDB creator="name" date="2011-07-30" version="1.0">
<shielding name="shielding1">
<dataPoint freq="100e6" trans_imp_abs="5" trans_imp_phase="0" trans_adm_abs="0"
trans_adm_phase="2"/>
```

```
<dataPoint freq="300e6" trans_imp_abs="6" trans_imp_phase="2" trans_adm_abs="4"
trans_adm_phase="1"/>
<dataPoint freq="500e6" trans_imp_abs="4" trans_imp_phase="3" trans_adm_abs="3"
trans_adm_phase="2"/>
<dataPoint freq="700e6" trans_imp_abs="1" trans_imp_phase="5" trans_adm_abs="2"
trans_adm_phase="5"/>
</shielding>
</cableDB>
```

Related reference

DI Card

SD Card

SK Card

This card defines a skin effect, ohmic losses, dielectric sheet, characterised surface definition, arbitrary user defined impedance boundary condition on wire segments and surface elements or a dielectric surface impedance approximation. Layered dielectrics can also be defined.

In the **Home** tab, in the **Define** group, click the **Media** icon. From the drop-down list select the **Losses (SK)** icon.

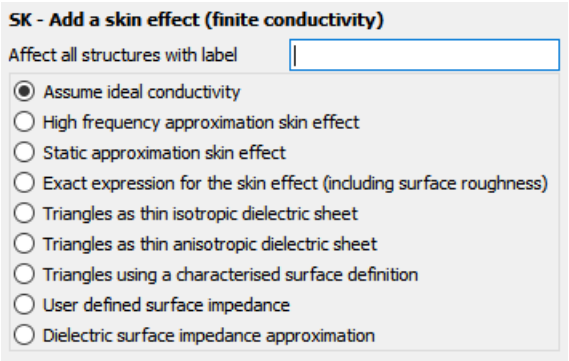


*Figure 950: The **SK - Add a skin effect (finite conductivity)** dialog.*

Parameters:

Affect all structures with label	The label to which the skin effect is applied is specified. All mesh elements with this label are assigned the skin effect.
Assume ideal conductivity	Ideal conductivity is assumed as if no SK card is specified for this label. All other parameters are ignored.
High frequency approximation skin effect	Apply the high frequency approximation skin effect to structures with the specified label.
Static approximation skin effect	Apply the static (ohmic losses) approximation skin effect to structures with the specified label.
Exact expression for the skin effect (including surface roughness)	Apply the exact expression for the skin effect (which includes the effects of surface roughness) to metallic surfaces and/or wires with the specified label.
Triangles as thin isotropic dielectric sheet	Treat metallic triangles with the specified label as thin isotropic dielectric layers (possibly consisting of multiple layers).
Triangles as thin anisotropic dielectric sheet	Treat metallic triangles with the specified label as thin anisotropic dielectric layers (possibly consisting of multiple layers).
Triangles using a characterised surface definition	Treat triangles with the specified label as a characterised surface (a surface defined by transmission and reflection coefficient data).
User defined surface impedance	Treat wire segments and surface elements with the specified label as an arbitrary user defined complex surface impedance.

Dielectric surface impedance approximation	Treat homogeneous dielectric regions in free space with the specified label as a dielectric surface impedance approximation.

Recommendations for Choosing the Label(s)

It is not recommended to set the same label for both triangles and segments with **Affect all structures with label**. Separate labels and a unique SK card for each label should be used. In addition all wires with the label **Affect all structures with label** must have the same radius. If this is not the case a unique label must be created for each radius.

Note: The skin depth is given by $\delta_{skin}=\sqrt{\frac{2}{\omega\mu\sigma}}$, where the radial frequency is $\omega=2\pi f$ and the permeability is $\mu=\mu_r\mu_0$.

Related tasks

Creating an Impedance Sheet (CADFEKO)

Creating a Characterised Surface (CADFEKO)

Applying a Thin Dielectric Sheet to a Face (CADFEKO)

Related reference

DI Card

DL Card

Skin Effect and Ohmic Losses

Specify the skin effect using either the exact expression, high frequency approximation or static approximation (Ohmic losses).

High Frequency Approximation Skin Effect

This option adds a high frequency approximation skin effect.

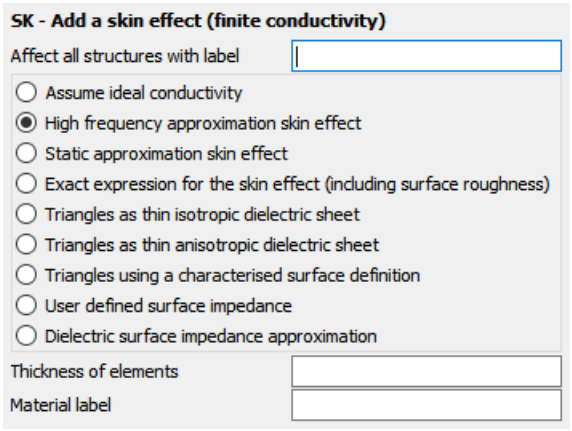


*Figure 951: The **SK - Add a skin effect (finite conductivity)** dialog.*

Note: The material parameters for the skin effect are defined with the DI card. The SK card then uses the label defined at the DI card.

Parameters:

Thickness of elements	The thickness d of the surface elements in metres (if an SF card is present, this is always scaled).
Material label	Label of the material which will be used (as specified in the DI card).

The required parameters are: μ_r, $\tan\delta_u$ and σ (defined with the DI card). If applied to surfaces then also the thickness d is required.

The following restrictions apply:

- A good conductivity is required, satisfying the condition $\sigma \gg \omega\varepsilon_0$.
- For wires the skin depth must satisfy the condition $\delta_{\text{skin}} < \varrho$ where ϱ is the wire radius. The surface impedance is given by $Z_s' = \frac{1}{2\pi\varrho}\sqrt{\frac{j\omega\mu}{\sigma}}$.
- For metallic surfaces the skin depth must satisfy the condition $\delta_{\text{skin}} < \frac{d}{2}$. The surface impedance is given by $Z_s = \frac{1}{2}\sqrt{\frac{j\omega\mu}{\sigma}}$.

Static Approximation Skin Effect

This option adds a static (Ohmic losses) approximation skin effect.

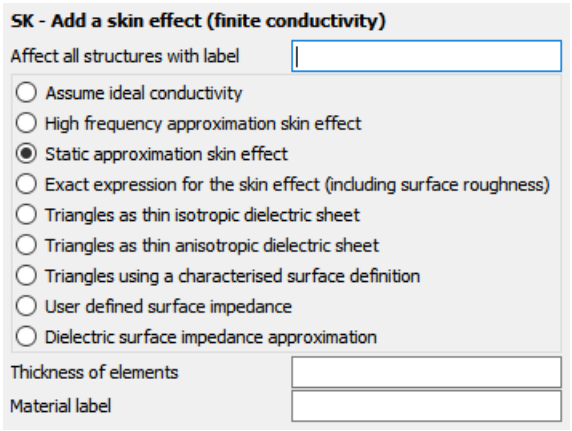


*Figure 952: The **SK - Add a skin effect (finite conductivity)** dialog.*

Note: The material parameters for the skin effect are defined with the DI card. The SK card then uses the label defined at the DI card.

Parameters:

Thickness of elements	The thickness d of the surface elements in metres (if an SF card is present, this is always scaled).
Material label	Label of the material which will be used (as specified in the DI card).

The required parameters are μ_r, $\tan\delta_u$ and σ (defined with the DI card). If applied to surfaces then also the thickness d is required.

The following restrictions apply:

- A good conductivity is required, satisfying the condition $\sigma \gg \omega\varepsilon_0$.
- For wires the skin depth must satisfy the condition $\delta_{skin} > \varrho$ where ϱ is the wire radius. The surface impedance is given by $Z_s' = \frac{1}{\pi\varrho^2\sigma}$
- For metallic surfaces the skin depth must satisfy the condition $\delta_{skin} > \frac{d}{2}$. The surface impedance is given by $Z_s = \frac{1}{\sigma d}$.

Exact Expression for the Skin Effect (Including Surface Roughness)

This option adds an exact skin effect (which includes the effects of surface roughness) to metallic surfaces and/or wires.

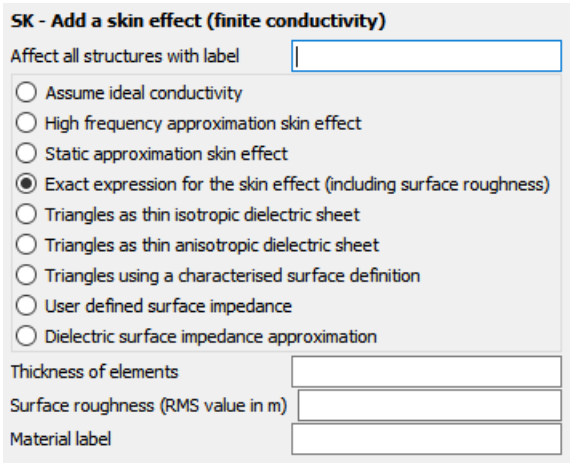


*Figure 953: The **SK - Add a skin effect (finite conductivity)** dialog.*

> **Note:** The material parameters for the skin effect are defined with the DI card. The SK card then uses the label defined at the DI card.

Parameters:

Thickness of elements	The thickness d of the surface elements in metres (if an SF card is present, this is always scaled).

Surface roughness (RMS value in m)	In many striplines, PCB or similar planar devices, when the frequency increases, the surface roughness of the employed metal (for example, copper) becomes larger than the skin depth. To model a conductor accurately, specify the surface roughness as an RMS value in m. Common values are typically in the range 50 nm to 100 µm. Specifying the surface roughness will increase the losses as the frequency increases. **Note:** Setting the surface roughness will not impact the total runtime or memory used.
Material label	Label of the material which will be used (as specified in the DI card).

The required parameters are μ_r, $\tan\delta_u$ and σ (defined with the DI card). If applied to surfaces then also the thickness d is required.

The following restrictions apply:

- A good conductivity is required, satisfying the condition $\sigma \gg \omega\varepsilon_0$.
- For wires with wire radius ϱ the surface impedance is given by

$$Z_{s'} = \frac{1-j}{2\pi\varrho\sigma\delta_{skin}} \frac{J_0[(1-j)\frac{\varrho}{\delta_{skin}}]}{J_1[(1-j)\frac{\varrho}{\delta_{skin}}]} \quad (221)$$

  where J_0 and J_1 are Bessel functions.
- For a sheet of thickness, d, with properties ε_c, μ_c with $\varepsilon_c = \varepsilon + \frac{\sigma}{j\omega}$ the wave propagation constant in the sheet is given by $\beta_c = \omega\sqrt{\mu_c\varepsilon_c}$ and the wave impedance in the sheet by $\eta_c = \sqrt{\frac{\mu_c}{\varepsilon_c}}$. The reflection coefficient at the interface between the sheet and the environment is defined as $\Gamma = \frac{E^-}{E^+} = \frac{\eta_o - \eta_c}{\eta_o + \eta_c}$. The surface impedance is given by

$$Z_{s'} = \eta_c \cdot \frac{1+\Gamma e^{-j2\beta_c d}}{1-\Gamma e^{-j2\beta_c d} + (\Gamma - 1)e^{-j\beta_c d}} \quad (222)$$

Triangles as Thin Isotropic Dielectric Sheet

This option defines a thin isotropic dielectric sheet (may consist of several layers).

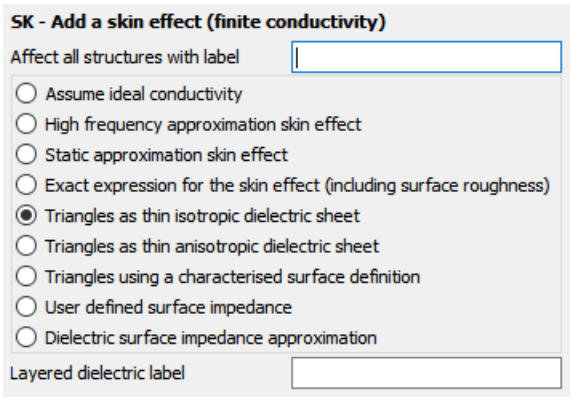


*Figure 954: The **SK - Add a skin effect (finite conductivity)** dialog.*

> **Note:** The material parameters for the isotropic dielectric sheet are defined with the DI and DL cards. The SK card then uses the label defined at the DL card.

Parameters:

Layered dielectric label	Label of the layered dielectric medium which will be used (as specified in the DL card).

This option is practical only for triangular surfaces, and not for wires. The required parameters are d, μ_r, $\tan\delta_u$ and ε_r as well as σ or $\tan\delta$ so that $\mu=\mu_r\mu_0$ and the complex dielectric constant $\varepsilon=\varepsilon_r\varepsilon_0(1-j\tan\delta)-j\frac{\sigma}{\omega}$. Usually either σ or $\tan\delta$ is entered as zero, but it is possible to specify both (for example to approximate a specific frequency response). Feko will give a warning which may be ignored.

The triangles with the label **Affect all structures with label** exist in a certain environment (ε_e, μ_e), which is usually specified with the DI card. The surrounding medium is denoted by label "0" and by default contains the parameters of free space. It is also possible to place the triangles within a dielectric body — in this case the environment is specified by modifying the parameters of material "0" in the DI card. An additional condition is that the triangles should be geometrically thin, that is d must be small relative to the lateral dimensions. The mesh size is determined by the wavelength in the environment (that is in the medium ε_e, μ_e). Therefore the layers must be thin relative to the wavelength in the environment.

When used with the MoM, the use of **Triangles as thin isotropic dielectric sheet** requires that the relations $\mu=\mu_e$ and $\varepsilon\neq\varepsilon_e$ are satisfied. For a single layer, the card consists of only one line. The surface impedance, as used by Feko, is then

$$Z_s=\frac{\beta}{2j\omega(\varepsilon-\varepsilon_e)\sin(\beta\frac{d}{2})} \qquad (223)$$

where $\beta=\omega\sqrt{\varepsilon\mu}$ is the complex propagation constant.

For multiple layers the card requires one line per layer with the parameters of the first layer on the same line as the card name. The approximate surface impedances of the different layers are added to determine the effective surface impedance.

When used with PO, it is not required that the relations $\mu = \mu_e$ or $\varepsilon \neq \varepsilon_e$ are satisfied. In this case the order of the layers is also significant. The layer on the side that the triangle normal vector points to, is specified in the first line with the remaining layers following in sequence.

Triangles as Thin Anisotropic Dielectric Sheet

This option defines a thin anisotropic dielectric sheet. The definition is similar to the isotropic dielectric sheet, except that the layers are anisotropic.

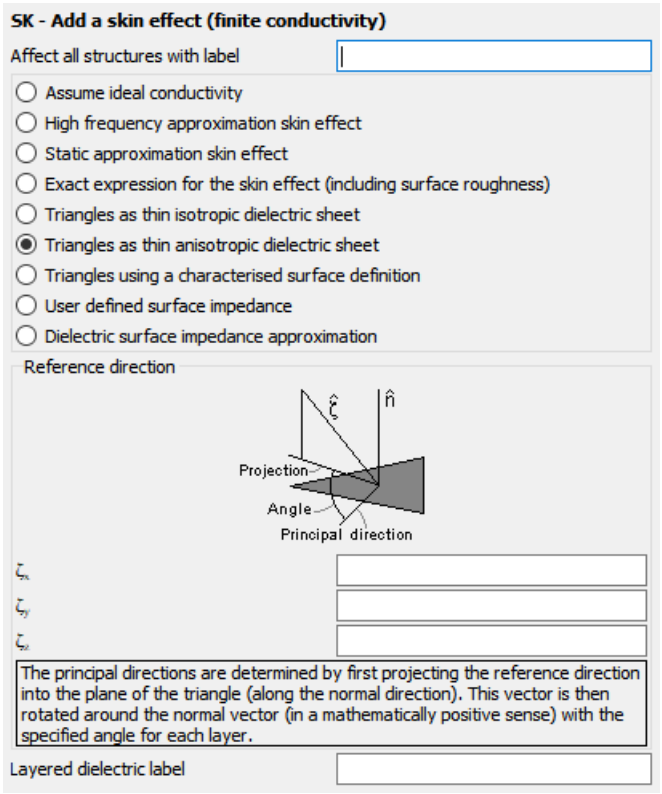


*Figure 955: The **SK - Add a skin effect (finite conductivity)** dialog.*

> **Note:** The material parameters for the anisotropic dielectric sheet are defined with the DI and DL cards. The SK card then uses the label defined at the DL card.

The principal direction in each layer is defined by the angle a (see **Angle** in the Figure 955) relative to the projection of the vector ζ (**Reference direction** group) onto the plane of the triangle (in the DI card). Here a is measured in the mathematically positive sense with respect to the normal vector of the triangle.

Tip: Use POSTFEKO to display the fibre direction and to confirm that the input file is correct.

In this case the card line is followed by an additional line for each layer. The medium properties in the *principle direction* is different from those in the *orthogonal direction* which lies in the plane of the triangle and orthogonal to the principal direction.

Parameters:

Reference direction	The X, Y and Z components of the vector ζ (used to define the principal direction, see Figure 955).
Layered dielectric label	Label of the layered dielectric medium which will be used (as specified in the DL card).

Triangles Using a Characterised Surface Definition

This option defines a characterised surface.

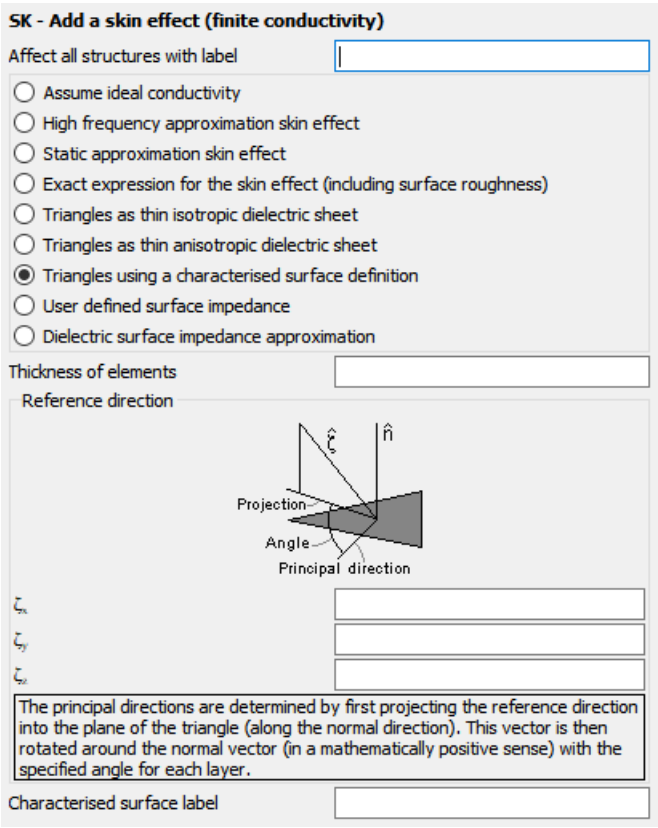


*Figure 956: The **SK Add a skin effect (finite conductivity)** dialog, set to **Triangles using a characterised surface definition**.*

Note: The material parameters for the characterised surface are defined with the DI card where it is imported from a `.tr` file. The SK card then uses the label defined at the DI card.

The principal direction in each layer is defined by the angle a (see Figure 956) relative to the projection of the vector ζ (**Reference direction** group) onto the plane of the triangle (in the DI card). Here a is measured in the mathematically positive sense with respect to the normal vector of the triangle.

In this case the card line is followed by an additional line for each layer. The medium properties in the principle direction is different from those in the orthogonal direction which lies in the plane of the triangle and orthogonal to the principal direction.

Parameters

Thickness of elements	The thickness of the triangles for the characterised surface. The thickness is used for MoM/MLFMM simulations and is disregarded for RL-GO.
Reference direction	The X, Y and Z components of the vector ζ (used to define the principal direction, see Figure 956).
Characterised surface label	Label of the characterised surface medium as specified in the DI card.

User Defined Surface Impedance

This option allows the definition of a complex user defined surface impedance, Z_s.

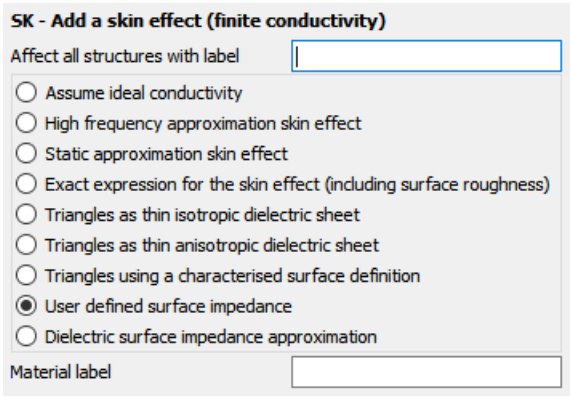


*Figure 957: The **SK - Add a skin effect (finite conductivity)** dialog.*

Note: The surface impedance (real and imaginary parts) are defined with the DI card. The SK card then uses the label defined at the DI card.

Parameters:

Material label	Label of the material which will be used as the defined surface impedance (as specified in the DI card).

For example, to model a solid dielectric object with relative permittivity ε_r and with conductivity σ at a specific angular frequency $\omega = 2\pi f$, the following surface impedance expression will be used:

$$Z_s = \sqrt{\frac{\mu_0}{\epsilon_0 \varepsilon_r - j\frac{\sigma}{\omega}}} \tag{224}$$

The user defined complex surface impedance Z_s relates to the surface current $\mathbf{J}_s$ and the E-field by $\mathbf{E}_{tan} = Z_s \mathbf{J}_s$ where $\mathbf{E}_{tan}$ is the tangential E-field and $\mathbf{J}_s$ is the surface current density.

Dielectric Surface Impedance Approximation

This option allows a dielectric region to be solved with the dielectric surface impedance approximation.

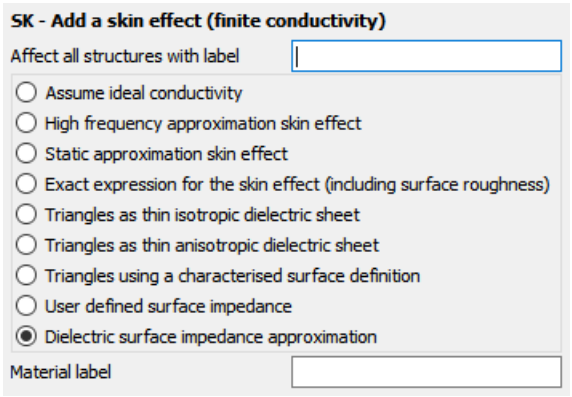


*Figure 958: The **SK - Add a skin effect (finite conductivity)** dialog.*

> **Note:** The surface impedance is calculated from the material properties defined at the DI card.

Parameters:

Material label	Label of the material (as specified in the DI card) which will be used for the calculation of the dielectric surface impedance.

The following dielectric surface impedance expression will be used:

$$Z_s = \sqrt{\frac{\mu_0}{\epsilon_0 \varepsilon_r - j\frac{\sigma}{\omega}}} \tag{225}$$

where the relative permittivity ε_r and conductivity σ are defined at a specific angular frequency $\omega = 2\pi f$.

SP Card

This card defines an S-parameter (S-matrix) request for active sources.

On the **Request** tab, in the **Configurations** group, click the **S-parameter (SP)** icon.

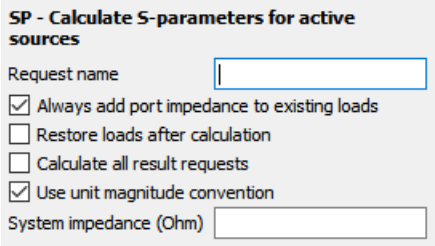


*Figure 959: The **SP - Calculate S-parameters for active sources** dialog .*

Parameters:

Request name	The name of the request.
Always add port impedance to existing loads	When S-parameters are computed, each port is automatically loaded by Feko with the S-parameter reference impedance of the port. If this option is checked, and the user has manually defined a load at a port, then the S-parameter load will be added to the existing load at the port. If this option is not checked, then Feko will automatically add the S-parameter reference loads at the various ports, but possible user defined loads of the same load type (see discussion below) will be overwritten (not added) at these ports.
Restore loads after calculation	As discussed above, Feko automatically adds loads to ports when computing S-parameters. With this option the behaviour can be controlled after the SP card processing is finished. When this option is enabled, the loads that were automatically added will be removed, and the load situation (for instance for a subsequent far field request) is the same as if the SP card was not used. Otherwise, all the loads as set during the SP card processing will remain in place afterwards.
Calculate all result requests	All result requests (for example, near field, far field and SAR) are calculated for each port excitation / loading scenario included in the S-parameter configuration.
Use unit magnitude convention	Select the **Use unit magnitude convention** check box to use the legacy definition for magnitudes when calculating the requests associated with the S-parameter configuration.
System impedance	The reference impedance in Ohm. This is used for all sources for which no impedance value is specified when defining the source. If this field is empty, it defaults to 50. Note that for

	waveguide sources (AW card) S-parameters are always related to the corresponding waveguide impedance.
Transmit power	The transmit power that is used when calculating the requests associated with the S-parameter configuration.

> **Note:** The transmit power specified does not affect the S-parameter matrix results.

All the ports must be defined before using the SP card. They are identified simply by defining excitation cards. Currently only A1, A2, A3, AE, AF, AN, AB and AW sources are supported. A1, A2 and A3 sources must be selected by label (not with position), and unique labels must be used (no other segments or triangles may have a label which is used for a port).

If the amplitude of any port is set to zero, it will be used as a receive port (or sink) but not as a source. For example, if only S_{21} and S_{11} are required for a two port network, one may set the amplitude of the source defining port 2 exactly to zero. Then S_{12} and S_{22} are not calculated. In some cases this could save considerable computation time.

The S-parameter load impedance for each of the port sources can be specified at the source itself. If no such impedance was specified, the system impedance (Ω) value specified with the SP card will be used (if this value is not specified it defaults to 50 Ω). This S-parameter load impedance will be added automatically to each port. The only exception is the waveguide port (AW card) and the modal port (AB card) where S-parameters are related directly to the corresponding waveguide impedance.

It must be noted that except for waveguide ports the SP card adds load impedances to all the ports. For A1, A2 and A3 sources it uses LZ type loads, for AN sources it uses LN type loads and for AE sources it uses LE type loads. If any similar loads were applied to the source position before the SP card these loads will either be added or overwritten. The addition or overwriting is set with the **Always add port impedance to existing loads** check box.

When execution continues after processing the SP card these loads will either be removed or kept, as controlled by the check box, **Restore loads after calculation**. This makes a difference when, for example, after the SP card the far field is computed with the FF card. If the loads are removed then the result for the far field pattern is the same as if there was no SP card (far field computed with ports unloaded by the S-parameter reference impedance). The disadvantage of restoring the loads is that the loads change after the SP card processing. For the MoM this means that the MoM matrix changes, and in order to compute the far field pattern, a full extra matrix computation and LU decomposition must be done. If the loads are kept, then further results are readily available (by re-using the LU decomposed matrix).

The original amplitudes and phases of the excitations will always be restored. It should, however, be noted that unlike near- or far field computations or other results, the amplitudes and phases of the excitations at the various ports do not influence the S-parameter results (except for the special case of setting the amplitude of a port to zero which indicates to Feko that this is a passive port only). This behaviour is consistent with the definition of S-parameters (results are normalised by the incident port signal). It should in particular also be noted that setting a phase of 180° for the excitation of a port does not change the direction of this port. One rather physically has to define the port with opposite orientation. When viewing a model in POSTFEKO then the arrows always indicate the positive source direction and the arrows will also not change direction when setting a 180° phase on the excitation.

Note that a request to compute S-parameters is not required for 1-port networks as the S_{11} (reflection coefficient for a 1-port network) data will be available since it is always calculated for voltage and current sources. For current and voltage sources, an additional S-parameter block will not be computed if the model consists of a 1-port network and the user requested an SP card (with unchanged reference impedance). For waveguide and modal ports, S-parameters are calculated by default in the same way that port impedances are calculated for voltage and current sources. When a redundant S-parameter request has been made, a warning will be displayed to indicate to the user that the SP card will be ignored. For n-port networks (with n > 1) while processing an S-parameter request, source, power and impedance data is not written to the `.out` and `.bof` files since this would be misleading as the effect of port loads would be included in the results.

Related tasks
Adding an S-Parameter Configuration (CADFEKO)

TL Card

This card connects a non-radiating transmission line between Feko geometry or other general non-radiating networks or transmission lines.

On the **Source/Load** tab, in the **Loads / networks** group, click the **Transmission line (TL)** icon.

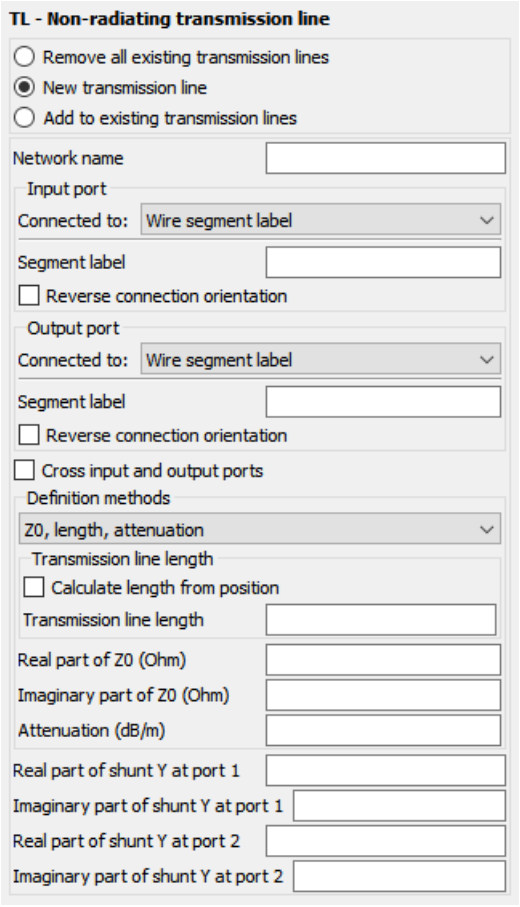


*Figure 960: The **Non-radiating transmission line** dialog.*

Note: Loads and sources can also be connected on a transmission line terminal using the LN and AN cards respectively.

Parameters:

Remove all existing transmission lines	If checked, all previously defined transmission lines are deleted. All the other input parameters are ignored.
New transmission line	Defines a new transmission line, all previously defined transmission lines are replaced.
Add to existing transmission lines	An additional transmission line is defined.

Network name The name of the transmission line.

Input port The input port (start of the transmission line) can be connected to geometry or other non-radiating ports in a number of ways.

Wire segment label	The label of the segment to which the transmission line port must be connected. If more than one segment has this label, the transmission port is connected to the last segment with this label.
Wire segment position	The segment is determined by specifying the Cartesian coordinates of the segment centre. These values are in metres and are scaled by the SF card if **Modify all dimension related values** is checked.
Internal port	The network name and the network port number of another network port that has to be connected.
Edge between regions with multiple labels	The positive and negative labels define the positive and negative terminals of the port connection.
Edge connected to ground/UTD	The positive or negative labels that define the edge where the network port has to be connected to.
Edge of microstrip between two points	The points that define the edge of the microstrip line where the network port has to be connected to.
Vertex by segment label	The vertex is determined by specifying a segment label. Also select whether the start or end point of that segment should be used.
Vertex by position	The vertex is determined by specifying the Cartesian coordinates of the vertex.
FEM line port position	The input port is attached to a FEM line port. The position of the FEM line port is specified by the start point and end point.
Port	The input port is attached to a port with label defined using the PT card.

Output port	Same as for **Input port**, but applies to the end of the transmission line.
Cross input and output ports	The positive port voltage is in the direction of the segment that it is connected to (from the start to the end point of the segment). Thus the input and output ports of the transmission line have unique orientations. If this item is checked the transmission line connecting the ports is crossed.
Calculate length from position	If checked, Feko determines the length based on the geometrical distance between the start and end points. Note that this feature is only available when both transmission line ports are connected to segments or vertices/nodes (the ports do not have to be the same type).
Transmission line length	The length of the transmission line in metres. This value is scaled with the scaling factor of the SF card.
Attenuation (dB/m)	Losses of the transmission line in dB/m. Note that since the propagation constant is taken as the propagation constant of the medium in which the start and end ports are located, the attenuation specified by this parameter is added to any losses of this medium. This factor is not affected by scaling specified with the SF card. This means that should a scaling factor which reduces the length of the transmission line be added, the total loss through the line will be less. (The length is now less and the loss per distance remained the same.)
Velocity of propagation (%)	The propagation speed through the transmission line relative to the speed of light.
Material label (dielectric)	The label of the dielectric medium (as defined in the DI card) used as the background medium for the transmission line.
Real part of Z_0 (Ohm)	Real part of the characteristic impedance of the transmission line in Ohm
Imaginary part of Z_0 (Ohm)	Imaginary part of the characteristic impedance of the transmission line in Ohm. Note that the characteristic impedance only defines the ratio between the voltage and current of the two waves propagating along the line. It does not specify any losses.
Real part of shunt Y at port 1	Real part of the shunt admittance at the input port in Siemens. (This admittance is across the port, connecting the two wires of the transmission line.)
Imaginary part of shunt Y at port 1	Imaginary part of the shunt admittance at the input port in Siemens.

Real part of shunt Y at port 2	Real part of the shunt admittance at the output port in Siemens. (This admittance is across the port, connecting the two wires of the transmission line.)
Imaginary part of shunt Y at port 2	Imaginary part of the shunt admittance at the output port in Siemens.

Any load impedance defined over a network[102] port with the LZ, LS, LP, LD, L2, LE, CO or SK cards are placed in series with the port. Parallel admittances can be defined directly at the TL card.

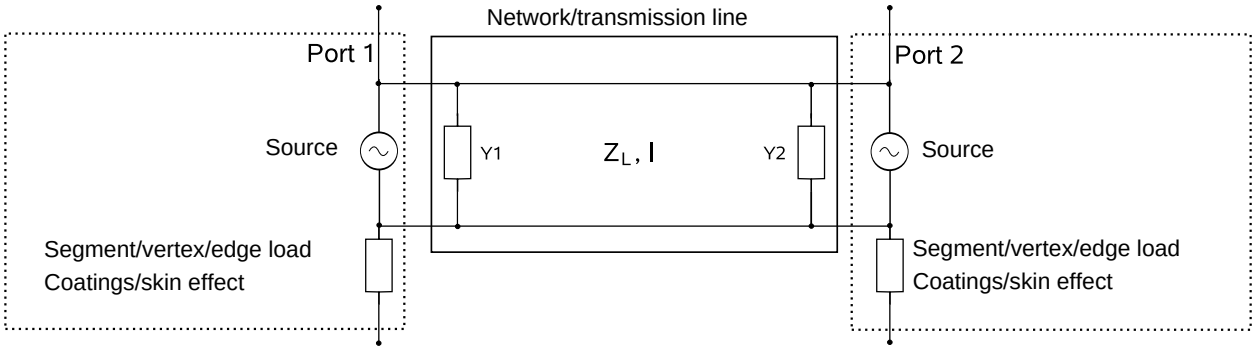


Figure 961: The load is placed in series with the network.

Any load impedance defined over a network port with the LN card is placed across the port.

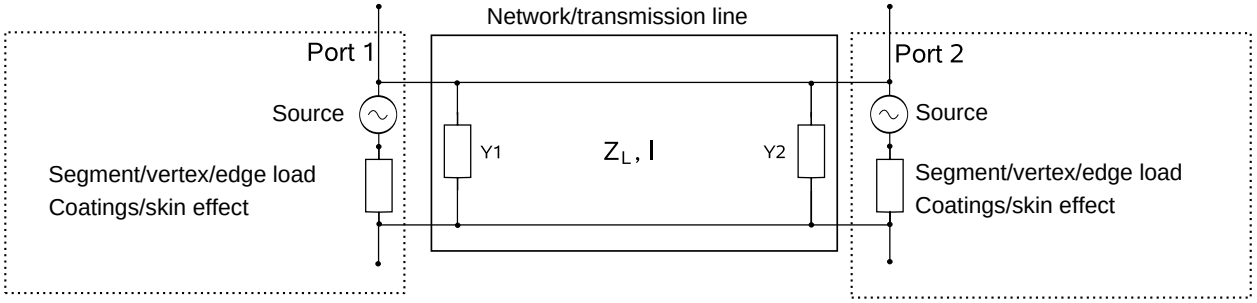


Figure 962: The load is placed across the network.

If a voltage source of type A1 or A3 is applied at one of the port segments, then this voltage source is assumed to be across the port (that is feeding the transmission line directly with an impressed voltage). If S-parameters are computed with respect to an excitation on a wire segment to which a TL card is connected, then the reference impedance is assumed to be in series with the source, but across the network port.

102. Network refers to general networks and transmission lines.

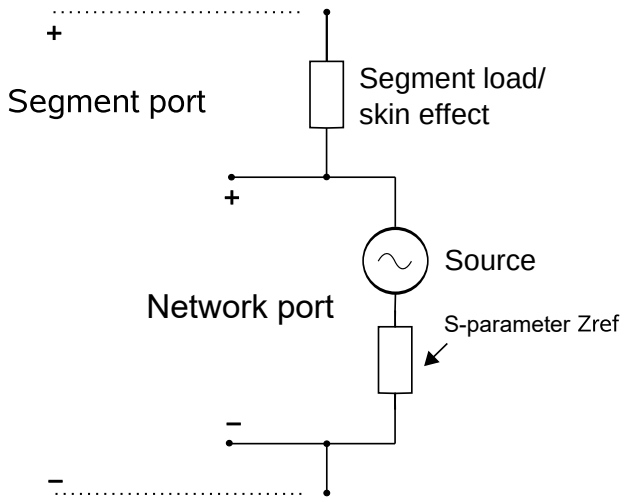


Figure 963: Load placement for S-parameter calculations on a segment port.

Note that the propagation constant and thus also the propagation loss of the transmission line is the same as that of the medium surrounding the port unless an additional loss tangent is specified in the Losses field. If this is free space the transmission line will be lossless. For transmission lines with a propagation constant that is higher than that of the surrounding medium, such as coaxial cables filled with dielectric material, the length of the transmission line should be reduced.

The following guidelines apply to determining the surrounding medium of a transmission line:

- When both ports of a transmission line are internal (not connected to geometry), the propagation constant of the background medium is used for the transmission line.
- If one port of the transmission line is internal, the propagation constant of the medium at the other port (connected to geometry) is used.
- Should both ports be connected to geometry, the medium of the input port (Port 1) is used for the propagation constant of the transmission line.
- Additionally, if a transmission line is located inside a planar multilayer substrate, the following applies:
  - If the transmission line is connected to geometry:
    - and lies inside a layer, the propagation constant of the medium of that layer is used.
    - and lies on the interface between two layers, the average medium between the two layers is used for the propagation constant.
  - If the transmission line is not connected to geometry then either the upper or lower medium is used depending on which one is lossless, or should both be lossless, the one with the greatest propagation constant, $|\beta|$, is used

Losses in the transmission line network (due to the shunt admittances or transmission line losses directly) are taken into account and will for instance reduce antenna efficiency or gain.

Related tasks
Adding a Transmission Line (CADFEKO)

Related reference

AN Card

LN Card

TR Card

This card is used to calculate the transmission and reflection coefficients for a plane wave interacting with a planar structure.

On the **Request** tab, in the **Solution requests** group, click the **Transmission / reflection (TR)** icon.

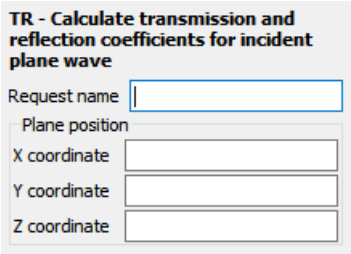


*Figure 964: The **TR - Calculate transmission and reflection coefficients for incident plane wave** dialog.*

Parameters:

Request name	The name of the request.
X, Y, Z coordinate	Cartesian coordinates of the position of the plane wave in metres. (is scaled by the SF card).

The transmission coefficient is defined as

$$\tau = \frac{E_t}{E_i} \tag{226}$$

The reflection coefficient is defined as

$$\rho = \frac{E_r}{E_i} \tag{227}$$

The figure below shows the incident, transmitted and reflected electric fields on a planar structure.

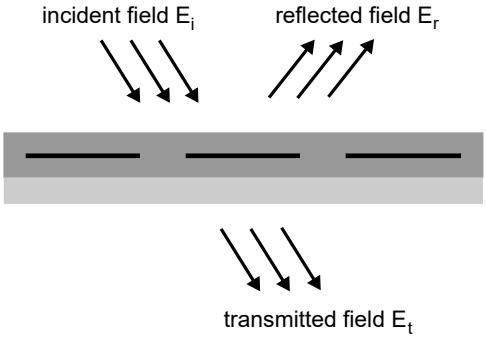


Figure 965: A plane wave interacting with a planar structure.

Note that for a transmission and reflection request, only a single plane wave source is allowed (that is no other sources are allowed in the model). The model should contain either of the following:

- A planar multilayer substrate without any other geometry/mesh in the model

- A 2D periodic boundary condition (PBC).

Related tasks

Requesting Transmission / Reflection Coefficients (CADFEKO)

WD Card

This card is used to define the dielectric properties of each of the windscreen glass layers. These layers are placed over the antenna elements by defining the relative position of the top layer to the reference plane.

In the **Home** tab, in the **Define** group, click the **Media** icon. From the drop-down list select the **Windscreen (WD)** icon.

The windscreen layers are defined in the direction of the reference plane's normal direction.

> **Note:** There are three cards that should be used together to create windscreen antenna models:
>
> 1. The WR card that defines the windscreen reference surface.
> 2. The WA card that defines the windscreen solution elements (antenna).
> 3. The WD card that defines the windscreen layered media (this card).

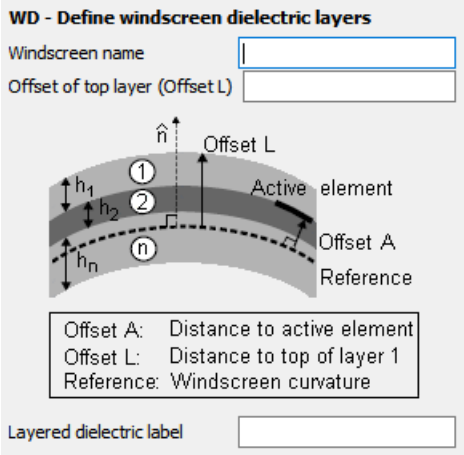


*Figure 966: The **WD - Define windscreen dielectric layers** dialog.*

Parameters:

Windscreen name	The name of the windscreen.
Offset of top layer (Offset L):	The distance from the reference to the top of layer 1.
Layered dielectric label	The label of the layered medium to be used, as defined in the DL card.

Related tasks
Creating a Windscreen Layer (CADFEKO)
Related reference
DL Card
WA Card
WR Card

How-Tos

A-2

A collection of how-tos are included that covers advanced concepts.

A-2.1 How to Use a Job Scheduling / Queuing System

Feko integrates into job scheduling and queuing systems such as Altair PBS Professional, Torque, IBM Platform LSF, Parallelnavi NQS, SLURM and Univa Grid Engine.

> **Note:** View the MPI documentation for the relevant MPI implementation in the `$ALTAIR_HOME\mpi` folder.

The job scripts depend on the level of complexity (for example, specifying the anticipated memory requirement or the expected run-time so that the job gets a SIGXCPU when exceeding the CPU time).

In the simplest case, you submit the job script to the right queue only, for example:

```
qsub job_script.sh
```

Inside the job script, make sure that RUNFEKO is called with the full path to the application and the `--use-job-scheduler` command line option to activate the queuing system integration with Intel MPI:

```
/opt/feko/bin/runfeko <filename> --use-job-scheduler
```

In this case, the machine file, that specifies the nodes to be used and the number of parallel processes, is obtained from the queuing system. Additional specifications in the job script, like the number of nodes to be used, should be done using the corresponding syntax of the queuing system.

When Intel MPI cannot be used, Feko can still be used with queuing systems. For instance, for Altair PBS Professional the batch systems provide a file with the list of hosts and number of CPUs which can be accessed by the environment variable `$PBS_NODEFILE`. Run Feko from the job script, for example with:

```
/opt/feko/bin/runfeko <filename> -np 16 --machines-file $PBS_NODEFILE
```

> **Note:** `filename` and the number of processes (16 in the example) can be a parameter of the script.

For advanced usage, you can make use of many features like merging `stdout` and `stderr` (recommended) into one file using:

```
#QSUB-eo
```

or a time limit can be set with a command such as the following (that sets a time limit of 3600 seconds per node):

```
#QSUB-l p_mpp_t=3600
```

For progress tracking purposes, in particular with longer jobs, it might also be useful to write `stdout` while the request is being processed, using a command like:

```
#QSUB-ro
```

Note: E-mail notifications can be sent when a job starts and finishes.

Feko includes the component QUEUEFEKO, which allows you to set up job scripts based on user-defined options and packs all the input files required for a run into an archive.

Related concepts
QUEUEFEKO Overview

Example Scripts

Three sample job scripts are included, one simple example one using the `--use-job-scheduler` option and two examples for using SGI MPT.

Script 1

```
#!/bin/bash
#
# Simple example for a job script for using Feko with a queuing system
# like Altair PBS Professional with the automatic Intel MPI integration.
#
#------------------------------------------------------------
# General settings for Altair PBS Professional
#PBS -l select=2:ncpus=1
#PBS -l walltime=1:30:00
#PBS -q workq
#PBS -V

# General settings for the Feko job
#   Name of the Feko file (without extension)
fekname=xxxxx
#   Name of the working directory
workdir=/scratch

#------------------------------------------------------------

echo "================================================="
echo "Feko running with file " $fekname "
echo "================================================="

# Go to the correct working directory (where we expect the input file)
cd $workdir

# Start the parallel Feko job
/opt/feko/bin/runfeko $fekname --use-job-scheduler

echo "---"
echo "Feko run finished"

# The end
exit 0
```

Script 2

```
#!/bin/bash
#
```

```
# Simple example for a script for using Feko with a queuing system
# like Altair PBS Professional.
# To be adapted according to the specific needs.
#
#------------------------------------------------------------
# Some useful QSUB commands (remove if not required)
#   Limit the maximum number of nodes
#QSUB-l mpp_p=8
#   Time limit per node in seconds
#QSUB-l p_mpp_t=60
#   Time limit for the total request in seconds
#QSUB-l mpp_t=60
#   Name of the request
#QSUB-r xxxxx.rqs
#   For accounting purposes, use a special account
#QSUB-A yyyyy
#   Merge stderr and stdout
#QSUB-eo
#   Direct the job log to a file
#QSUB-j xxxxx.log
#   Direct stdout to a file
#QSUB-o xxxxx.sto
#   Write stdout file while request is executed (allows monitoring)
#QSUB-ro

#------------------------------------------------------------
# General settings for the Feko job
#   Number of parallel processes to be used
nprocs=8
#   File with the nodenames to be used
machfile=$PBS_NODEFILE
#   Name of the Feko file (without extension)
fekname=xxxxx
#   Name of the working directory
workdir=/scratch

#------------------------------------------------------------

# Avoid immediate job abort when signal 26 (CPU time limit exceeded)
# is received (this gives Altair Feko time to clean up everything)
trap "" 26

echo "======================================================================"
echo "Feko running with file " $fekname " on " $nprocs " processes"
echo "======================================================================"
date

# Go to the correct working directory (where we expect the input file)
cd $workdir

# Start job accounting (if installed)
ja

# Start the parallel Feko job
/opt/feko/bin/runfeko $fekname -np $nprocs --machines-file $machfile

# Accounting output
ja -chl     # detailed output
ja -s       # summary
ja -t       # terminate job accounting

echo "---"
echo "Feko run finished"
```

```
date

# The end
exit 0
```

Script 3

```
#!/bin/bash
#
# Simple example for a script for using Feko with a queuing system
# like PBS.
# To be adapted according to the specific needs.
#
#------------------------------------------------------------
# General settings for the Feko job
#   Number of parallel processes to be used
nprocs=8
#   Name of the Feko file (without extension)
fekname=xxxxx
#   Name of the working directory
workdir=/scratch

#------------------------------------------------------------

echo "======================================================================"
echo "Feko running with file " $fekname " on " $nprocs " processes"
echo "======================================================================"

# Go to the correct working directory (where we expect the input file)
cd $workdir

# Start the parallel Feko job
/opt/feko/bin/runfeko $fekname -np $nprocs --machines-file $PBS_NODEFILE

echo "---"
echo "Feko run finished"

# The end
exit 0
```

A-2.2 How to Feed a Grounded Coplanar Waveguide

A method is presented on how to feed a grounded coplanar waveguide (GCPW) in CADFEKO. The same method can be used to feed a coplanar waveguide (CPW).

A grounded coplanar waveguide consists of a signal conductor placed between ground conductors but separated by slots. The conductors are fabricated on top of a dielectric substrate with a conductive ground plane at the bottom.

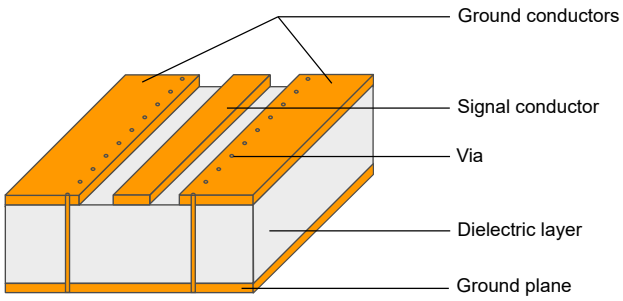


Figure 967: A graphical representation of a grounded coplanar waveguide (GCPW) with vias.

Constructing a GCPW

Create the grounded coplanar waveguide (GCPW) in CADFEKO.

1. Design the GCPW with a specific characteristic impedance using one of the following methods:
   - Use a third-party tool (transmission line calculator) to calculate the dimensions of the GCPW.
   - Calculate the dimensions of the GCPW using equations from literature.
2. Create the GCPW in CADFEKO using the dimensions obtained in Step 1.

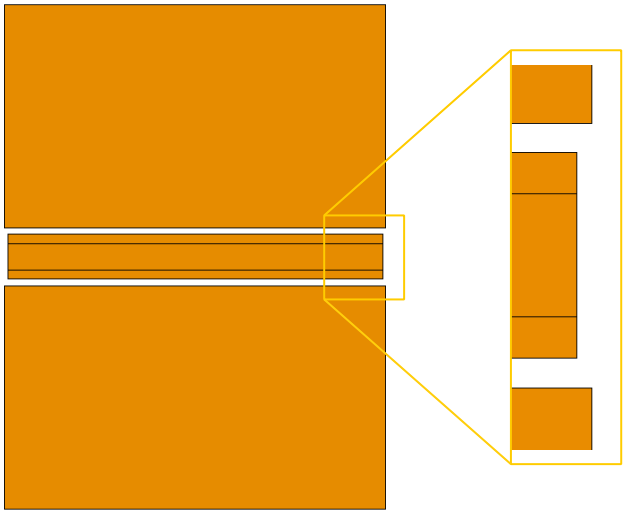


Figure 968: Top view of the GCPW. When creating the GCPW, make the signal conductor slightly shorter than the ground conductors, see enlargement on the right. Note that the ground plane is hidden.

> **Tip:** Use a planar multilayer substrate for the dielectric layer and ground plane.

3. Add vias to connect the ground conductors to the ground plane.

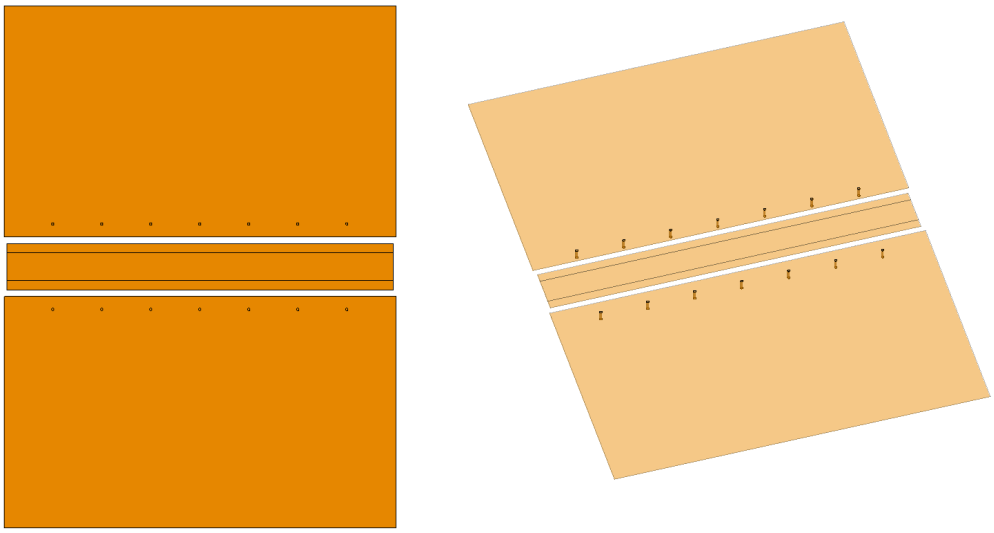


Figure 969: On the left, a top view of the GCPW with vias. To the right, an isometric view of the GCPW with vias, ground plane hidden and opacity set.

> **Tip:**
>
> - The via spacing should be between $\frac{\lambda}{8}$ and $\frac{\lambda}{10}$ to keep the ground at the same potential and for maintaining a stable impedance.
> - The distance of the vias to the center or ground edges affects the impedance. Too close and the vias interfere with the field distribution between the top and bottom plates. Too far and the parallel plate stray inductance increases.

Related tasks
Defining an Infinite Planar Multilayer Substrate

Constructing a Feed for the GCPW

Create the feed elements for the edge ports.

1. At each end of the signal conductor, create a connecting rectangle. These two rectangle faces are used when defining the edge ports.

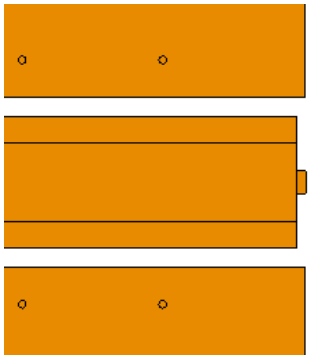


Figure 970: Top view of the GCPW showing the connecting rectangle at the end of the signal conductor. Note that the ground plane is hidden.

Note: Keep the width of the connecting rectangles less than $\frac{\lambda}{30}$ to prevent higher order modes or resonances over the width of the feeding edge.

2. At each end of the signal conductor, create a vertical plate that connects to the ground plane.

   The vertical plates at the end of the line mimics the surface of a connector that would need to connect to the lines as well as the ground plane.

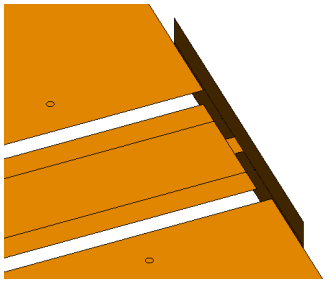


Figure 971: Isometric view of the GCPW showing the vertical plate. Note that the ground plane is hidden.

3. Union the geometry or model mesh.
4. At each end of the signal conductor, add an edge port at the edge between the connecting rectangles and the vertical plate.

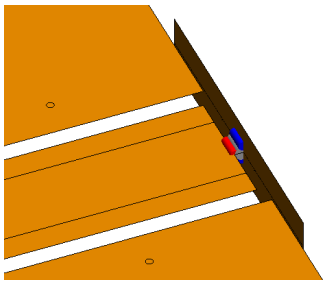


Figure 972: An isometric view of the GCPW showing the edge port. Note that the ground plane is hidden.

A-2.3 How to Estimate Memory Requirements for the MLFMM

A method is presented that allows you to estimate the memory requirements for a model solved using the multilevel fast multipole method (MLFMM).

The MLFMM memory requirement is directly proportional to $N\log_{10}(N)$, where N is the number of unknowns. When the frequency doubles, the triangle size in the mesh should be halved (to keep the same triangle size in terms of the wavelength). This causes the number of unknowns to increase by a factor of 4.

For example, if a model has 10 000 unknowns at 200 MHz, it will have 40 000 unknowns after meshing for 400 MHz. The increase in memory for the MLFMM solver is then:

$$\frac{40000\times\log_{10}(40000)}{10000\times\log_{10}(10000)}=4.8 \qquad (228)$$

For models with increasing electrical size, the above equation becomes less accurate. In addition to the number of unknowns, the total memory requirement also depends on the following:

- Uniformity of the mesh (similar element sizes for the whole mesh versus a mixture containing very fine element sizes)
- Number of mesh elements, specifically the storage of the triangle information (data such as position, normal, size)
- Number of parallel processes
- Number of nodes (hosts) used in a compute cluster scenario[103]
- Type and size of the preconditioner

> **Note:** This how-to was created using Feko version 2019 that includes several shared memory upgrades for parallel processing. Always ensure that you are using the latest Feko version.

Steps to Estimate the Memory

An optional parameter for the solution method allows you to calculate an estimate for the required memory.

1. On the **Solve/Run** tab, in the **Run/Launch** group, click the **Feko Terminal** icon.
2. Run the Solver using the intended number of processes using the special execution mode:

```
--estimate-resource-requirements-only
```

3. When the Solver execution is complete, open the `.out` file and find the following text block at the end of the file:

```
Peak memory usage during the whole solution: 61.707 MByte
 (refers to the master process only)
```

103. Some aspects of the solution are copied to each process and / or to each node.

```
 Sum of the peak memory of all processes:     1.928 GByte
 On average per process:                      61.698 MByte

 NOTE    48414: Memory requirement for a regular Feko run (without --estimate-
resource-requirements-only) is higher

 Memory estimate for a regular Feko run:      59.784 GByte (total for all
 parallel processes)
 On average per process:                      1.868 GByte
```

4. The estimated memory is given by `Memory estimate for a regular Feko run`.

Example

Solve a model with the file name `car_a.cfx` using 40 processes on a cluster with two hosts. Each host has a total of 20 available cores. The machines file name is `hosts.list`.

Open a Feko Terminal window on the host (or master node on the cluster) and execute the following command:

```
runfeko car_a -np 40 --machines-file hosts.list --feko-options --estimate-resource-
requirements-only
```

Notes on Usage

- If the model is to be solved on a cluster, the correct machines file should be used as some aspects of the solution are duplicated on each node.

  If the estimation algorithm is executed with the intended number of parallel processes, but only on a single node, the estimate is less accurate compared to execution over all the intended nodes.
- When comparing the execution time for the estimation algorithm to the solution of the model, the estimation algorithm is only a fraction of the total run time.
- The estimation algorithm requires roughly between $1/30^{th}$ and $1/3^{rd}$ of the memory for a complete solution.[104]
- If a memory error occurs during the estimate, the full solution will also fail on the same host / cluster that the estimation was performed with, for example:

  ```
  ERROR   32463: Not enough memory available for dynamic allocation
  ```
- An increase in the number of processes results in a less accurate estimate as some solution aspects are duplicated for each parallel process, but not included in the estimate.

  > **Note:** For example, solving a very large model comprising several tens of millions of triangles with:
  >
  > - 32 processes, the estimate was 90% accurate
  > - 256 processes, the estimate was 85% accurate
- The estimation is available for both SPAI and sparse LU preconditioners for sequential and parallel solutions.

104. Based on a few limited comparisons.

- Output requests (for example, far field requests) increases the memory requirement and are not included in the estimation.

Related concepts

Preconditioners for MLFMM

--estimate-resource-requirements-only

A-2.4 How to Construct a Conformal Patch Antenna

A method is presented on how to construct a conformal patch antenna in CADFEKO.

The conformal patch antenna consists of a finite dielectric substrate with a metallic (or PEC face) on the top surface and a PEC ground plane at the bottom. The patch is fed with a voltage source on a wire port. The patch antenna will be constructed as if attached to a larger metallic cylinder with a specific radius.

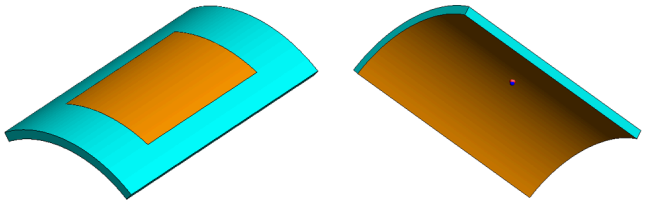


Figure 973: A graphical representation of a conformal patch antenna - top view (left) and bottom view (right).

Dimensions

The patch antenna has the following dimensions:

- Patch dimensions: 31.1807 mm x 46.748 mm
- Ground plane: 50 mm x 80 mm
- Substrate height: 2.87 mm
- Relative permittivity: 2.2
- Feed pin location is 8.9 mm from the centre of the model.
- The patch is bent onto a cylinder of radius 40 mm.
- Variable *alpha* is used for the angle subtended by the ground plane at the centre of the cylinder.
- Variable *beta* is used for the angle subtended by the patch at the centre of the cylinder.

Constructing a Conformal Patch Antenna

Create the conformal patch antenna in CADFEKO.

1. Set the model unit to millimetres.
2. Add a work plane.
   - **Origin**: (0, 0, -40)
   - **V vector**: (0, 0, 1)
   - **Label**: `workplane_hors`
3. Set the **workplane_hors** as the default workplane.

   > **Tip:** Expand **Workplanes** in the tree, open the right-click context menu and click **Set as default**.

4. Create a cylinder.
   - **Definition method**: **Base centre, radius, height**

- **Base centre (B)**: (0, 0, -40)
- **Radius (R)**: 40
- **Height (H)**: 80
- **Label**: `cylinder_out`

5. Create a copy (duplicate) of the **cylinder_out** part and modify the following:
   - **Radius (R)**: 40 - 2.87
   - **Label**: `cylinder_in`
6. Subtract **cylinder_in** from **cylinder_out**.

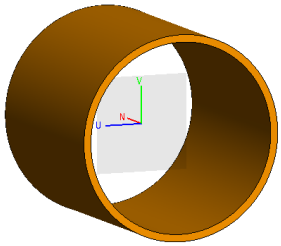


Figure 974: Remaining geometry after subtracting the two cylinders.

7. Define the following variables:
   - *alpha* = deg(50/40)
   - *beta* = deg(31.1807/40)
8. Split the **Subtract1** part:
   - **Plane**: **UN**
   - **Rotate split plane**, **N axis**: 90 - *alpha*/2

   This creates two parts, **Split_front1** and **Split_back1**.
9. Delete the **Split_back1** part.
10. Split the **Split_front1** part.
    - **Plane**: **UN**
    - **Rotate split plane**, **N axis**: 90 + *alpha*/2

    This operation results in two parts, **Split_back1** and **Split_front1**.
11. Delete the **Split_front1** part.

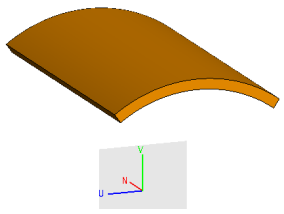


Figure 975: The finite solid outline of the patch antenna with correct dimensions and curvature, but with no material, face properties, or feed.

12. Create the leading edge of the patch.

a) Create a line.

- Use the **Global XY** workplane.
- **Start point**: (0, -46.784/2, 0)
- **End point**: (0, 46.784/2, 0)
- **Label**: `edge_of_patch`

13. Rotate the **edge_of_patch** part on the N axis through an angle of -*beta*/2 degrees.

14. Spin the **edge_of_patch** part on the N axis through an angle of *beta* degrees.

> **Tip:** Step 13 and Step 14 assume that **workplane_hors** is the default work plane.

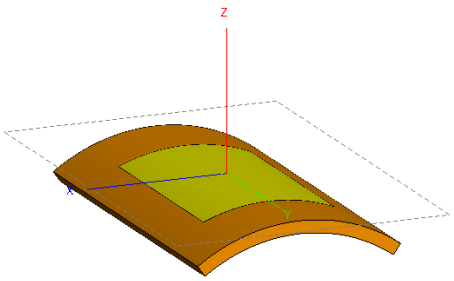


Figure 976: View after spinning the leading edge in the previous step.

15. Create the patch antenna feed.

a) Create a line.

- Use the **Global XY** workplane.
- **Start point**: (0, 0, -2.87)
- **End point**: (0, 0, 0)
- **Label**: `feed`

16. Position the feed line at the exact feed position.

a) Rotate the **feed** part through an angle of -*beta*/2 + deg(8.9/40).

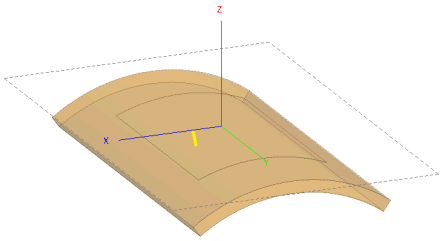


Figure 977: View of the patch antenna showing the feed line in yellow.

> **Tip:** Use one of the following methods to view the feed inside the model: Change the opacity of the model or use a cutplane.

17. Union all the parts.
18. Create a dielectric medium.
    - **Relative permittivity**: 2.2
    - **Label**: `substrate`
19. Set the region of **Union1** to **substrate**.
20. Set the bottom face of **Union1** to perfect electric conductor.
21. Set the top face of **Union1**, which corresponds to the metallic patch, to perfect electric conductor.
22. Create a **Wire port** on the feed line.

    Tip: Use a cutplane to add the wire port.

23. Create a **Voltage source** on **Port1** with default settings.
24. Set the **Frequency** to `3e9` Hz.
25. Create a full 3D far field request. Set the workplane (for this request) to the **Global XY** workplane.
26. Create the mesh.
    - **Mesh size**: **Standard**
    - **Wire segment radius**: `0.65`

    Note: If the meshing process takes more than two seconds, you may have neglected to set the **Model unit**. Change the **Model unit** to **Millimetres (mm)** and remesh the model.

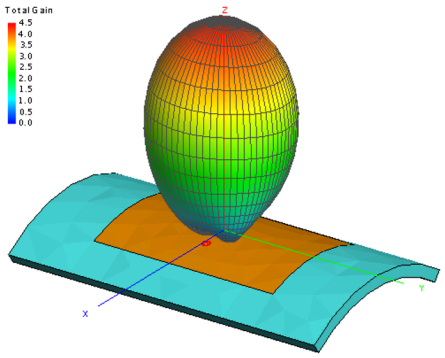


Figure 978: View of the 3D pattern in POSTFEKO.

A-2.5 How to Construct a Complementary Slotted Two-Arm Spiral Antenna

A method is presented on how to construct and feed a complementary slotted two-arm spiral antenna in CADFEKO.

A complimentary slotted two-arm spiral antenna consists of two spiral slots in a conducting plate. The two spirals are shifted by 180° with respect to one another resulting in spirals of equal width and equal gaps between the spirals.

Spiral antennas are widely used due to their consistent gain and input impedance over a wide bandwidth.

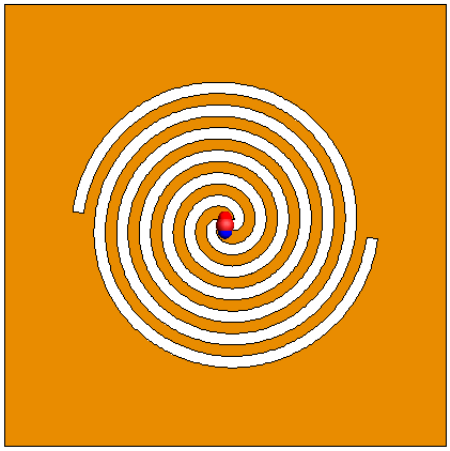


Figure 979: Top view of a complementary slotted two-arm spiral antenna with feed line and wire port.

Constructing the Spiral Antenna

Create the complementary slotted two-arm spiral antenna in CADFEKO.

1. Set the model unit to centimetres.
2. Create the following variables:
   - *a* = 1.1459 (The spiral growth ratio.)
   - *r1* = 1 (The base radius of the spiral.)
   - *r2* = 45/2 (The end radius of the spiral.)
   - *w* = 1.8 (The width of the slot.)
   - *width* = 70 (The width and depth of the conducting plate.)
   - *n1* = *r1*/(2 * *pi* * *a*)
   - *n2* = *r2*/(2 * *pi* * *a*)
   - *n* = *n2* - *n1* (The number of turns.)
   - *fmin* = 150e6 (The minimum frequency.)
   - *fmax* = 900e6 (The maximum frequency.)
   - *lambda0* = 100 * *c0*/*fmax* (A scale factor of 100 to compensate for working in centimetres.)

- *meshing* = min(*lambda0*/12, 1.3 * *w*) (The triangle length to be minimum of $\frac{\lambda}{12}$ or 1.3×slot width.)

3. Create a helix.

- **Definition method**: **Base centre, base radius, end radius, height, turns**
- **Origin of the helix (C)**: (0, 0, 0)
- **Base radius (Rb)**: *r1*
- **End radius (Rt)**: *r2*
- **Height**: 0
- **Turns (N)**: *n*
- **Label**: *Helix1*

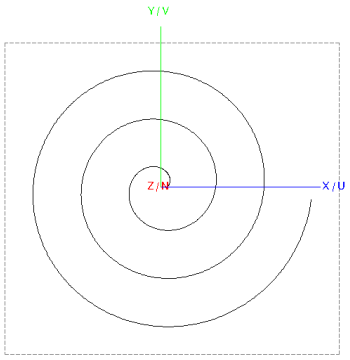


Figure 980: Top view of Helix1.

4. To add width to the spiral, create a copy (duplicate) of *Helix1*.

a) Rename the copy to *Helix2*.

b) Modify the start radius and end radius:

- **Base radius (Rb)**: *r1* + *w*
- **End radius (Rt)**: *r2* + *w*

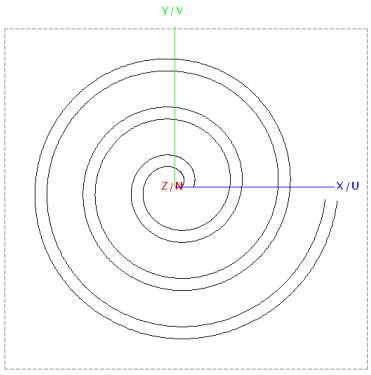


Figure 981: Top view of Helix1 and Helix2.

5. Connect the two helices to create a surface (loft).

a) Select *Helix1* and *Helix2* and loft the two curves to create a surface.

> **Note:** Do not select the **Reverse orientation** check box.

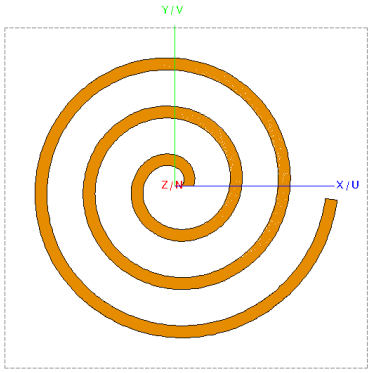


Figure 982: Top view of the lofted surface (spiral).

6. Create half of the feed.

a) Create a polygon.

- **Corner 1**: (*r1* + *w*/2, -*w*/2, 0)
- **Corner 2**: (0, -*w*/2, 0)
- **Corner 3**: (0, *w*/2, 0)
- **Corner 4**: (*r1* + *w*/2, *w*/2, 0)
- **Corner5**: (*r1* + *w*, 0, 0)
- **Label**: `Polygon1`

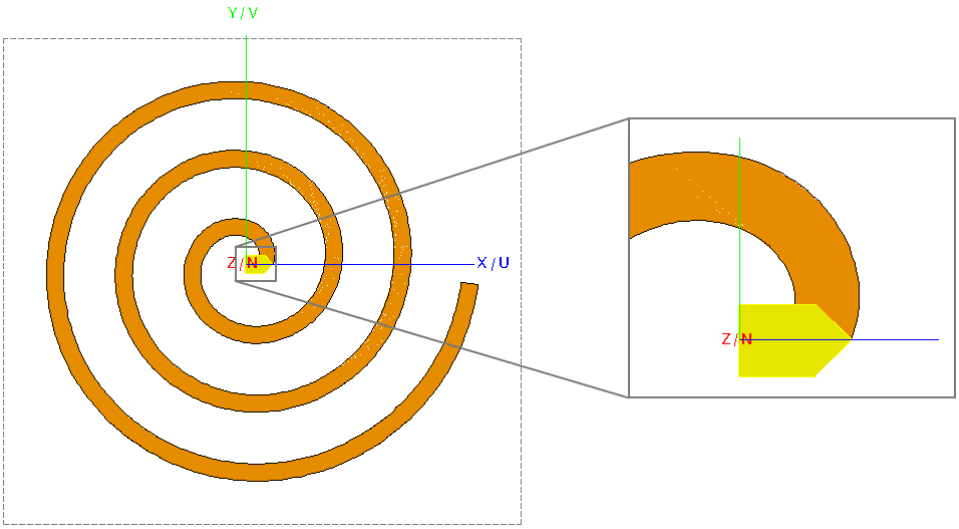


Figure 983: Top view of the lofted surface (spiral) showing half of the feed at the centre.

7. To ensure mesh connectivity between the feed and helix, union the lofted surface (*Loft1*) and the feed (*Polygon1*). The default label for the new union is *Union1*.
8. Simplify *Union1* to remove non-essential edges and faces.
9. Create the second arm of the spiral antenna.
   a) Copy and rotate *Union1* by 180° around the N axis.

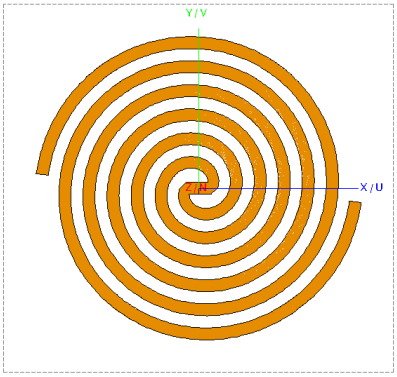


Figure 984: Top view of the two-arm spiral.

10. Union all parts of the model. Rename the union to *Union2*.
11. Create the conductive plate which will contain the complementary slotted spiral antenna.
    a) Create a rectangle.
       - **Definition method**: **Base centre, width, depth**
       - **Base centre (C)**: (0, 0, 0)
       - **Width (W)**: *width*
       - **Depth (D)**: *width*
       - **Label:**: `Rectangle1`
12. Create the spiral slots in the conductive plate.
    a) Subtract *Union2* from *Rectangle1*.

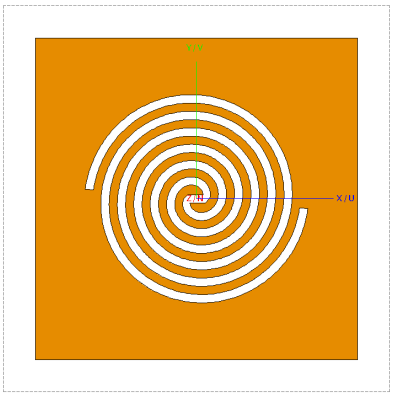


Figure 985: Top view of the two-arm spiral slots in a conductive plate.

13. Create the feed wire.

a) Create a line.

- **Start point**: (0, -*w*/2, 0)
- **End point**: (0, *w*/2, 0)

14. Add a wire port to the feed wire.

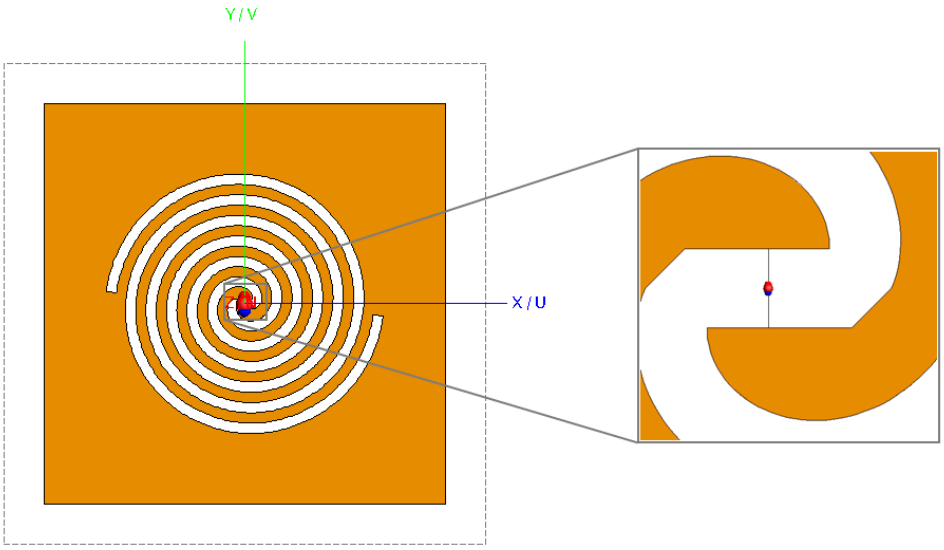


Figure 986: Top view of the complementary two-arm slot antenna showing the wire port at the feed line.

A complementary slotted two-arm spiral antenna with feed wire and wire port were constructed. Add requests, mesh the model and view the results in POSTFEKO.

A-2.6 How to Improve Convergence for the MLFMM

The MLFMM is an iterative solution method, and under certain conditions, the iterative solution may fail to converge. Several model or solution settings are presented that could improve the model's convergence behaviour.

Sometimes when using the MLFMM, the Solver stops with the error message:

```
ERROR 4673: Iterative solution of the system of linear equations failed, maybe try
 another pre-conditioner (solution settings).
```

The Solver stores the solution with the lowest residuum during the solution. Results are generated if this residuum is considered adequate, but the results may be less accurate if the stopping criterion for the residuum is not exactly met. In this case, the Solver output contains the warning:

```
WARNING 830: Maximum number of iterations reached without convergence, using in the
 following the solution with the smallest residuum.
```

Note: Warning 830 indicates possible inaccurate results due to inadequate convergence.

One or a combination of the following changes can be made to the model:

- Activate additional stabilisation for the MLFMM.
- Adjust the mesh.
- Change the preconditioner.
- Change the default box size.
- Use the CFIE for metallic structures.
- Use double precision.

If these options fail to bring convergence, consider if the model warrants using another solution method, or contact technical support for further assistance.

Tip: Highly lossy media (solved with the SEP) in most cases have poor convergence for the MLFMM. The FEM or VEP would be better suited to solving these materials.

Activating Additional Stabilisation for the MLFMM

Activating additional stabilisation may help to achieve convergence for the MLFMM.

Dielectrics, wires and other elements can be included in the model. If the poor convergence is due to something other than the metallic triangles, activating the stabilised MLFMM is unlikely to improve convergence.

Restriction: The stabilised MLFMM improves convergence only for metallic triangles.

Adjusting the Mesh

Slight adjustments to the mesh size (smaller or larger elements) could lead to improved convergence. If a model is discretised too finely or too coarsely, convergence could be negatively affected.

If a model is discretised too finely (smaller than $\frac{\lambda}{10}$) or mesh elements are too coarse (larger than $\frac{\lambda}{7}$), convergence could be negatively affected.

Tip: Reduce the radii of thick wire segments, or replace them with metallic strips (2D meshes) or cylinders (3D meshes) to improve convergence.

Changing the Preconditioner

The sparse LU is the default preconditioner. Changing to another preconditioner may help to achieve convergence for the MLFMM. Select one of the following preconditioners:

- Use the sparse approximate inverse (SPAI) preconditioner.
  - The default (accelerated SPAI) preconditioner is fast.
  - The non-accelerated SPAI preconditioner takes longer than its accelerated counterpart, but often converges better.
- Use the incomplete LU decomposition (ILU) preconditioner.

  Restriction: The ILU preconditioner is supported only for sequential solutions.

Changing the Default MLFMM Box Size

The MLFMM uses a boxing algorithm that encloses the entire computational space in a single box at the highest level, dividing this box in three dimensions into a maximum of eight child boxes and repeating the process iteratively until the side length of each child box is approximately a quarter wavelength at the finest level. Using a different box size at the finest level can sometimes facilitate convergence, although memory consumption could increase if the box size is increased.

The default box size is 0.23. A lower value decreases memory consumption while a higher value increases the memory consumption.

Tip: Try box size values between 0.2 and 0.35 wavelengths with increments of 0.02.

Using the CFIE on MLFMM Metallic Surfaces

The combined field integral equation (CFIE) uses both the electric field integral equation (EFIE) and magnetic field integral equation (MFIE). This produces a better-conditioned matrix leading to better convergence in general.

Note:

- Use the default preconditioner with the CFIE.
- CFIE can only be applied to surfaces bounding closed PEC structures.
- A mixture of CFIE and EFIE surfaces can be used.
- Sharp corners on CFIE surfaces can lead to inaccurate results.

Sharp corners should be meshed finer if CFIE is applied. If there is uncertainty, the EFIE should rather be applied around sharp corners. A rule of thumb is to apply the EFIE up to a few meshed triangles away from the sharp corners.

Using Double Precision

The Solver uses single precision by default- a single byte is used to store a complex number.

Use double precision when higher accuracy is required and to help resolve convergence issues. Double precision uses two bytes to store a complex number in the matrix. This increases the number of significant digits and reduces numerical noise.

Note:

- Double precision requires twice the memory compared to single precision.
- Double precision does not improve convergence for the stabilised MLFMM.

Related concepts

MLFMM Settings

Preconditioners for MLFMM

Local Mesh Refinement

General Solver Settings - Double Precision

Automatic Meshing

Related tasks

Using the CFIE For Closed PEC Regions

A-2.7 How to Improve Convergence for the MLFMM /FEM

The hybrid multilevel fast multipole method (MLFMM) / finite element method (FEM) is an iterative solution method and under certain conditions, the iterative solution may fail to converge. Several model or solution settings are presented that could improve the model's convergence behaviour.

Sometimes when using the MLFMM, the Solver stops with the error message:

```
ERROR 4673: Iterative solution of the system of linear equations failed, maybe try
 another pre-conditioner (solution settings).
```

The Solver stores the solution with the lowest residuum during the solution. Results are generated if this residuum is considered adequate, but the results may be less accurate if the stopping criterion for the residuum is not exactly met. In this case, the Solver output contains the warning:

```
WARNING 830: Maximum number of iterations reached without convergence, using in the
 following the solution with the smallest residuum.
```

Note: Warning 830 indicates possible inaccurate results due to inadequate convergence.

One or a combination of the following changes can be made to the model:

- Adjust the mesh.
- Change the default box size.
- Use the CFIE for metallic structures.
- Use double precision.
- Change the FEM to use first order basis functions.

Adjusting the Mesh

Slight adjustments to the mesh size (smaller or larger elements) could lead to improved convergence. If a model is discretised too finely or too coarsely, convergence could be negatively affected.

Tip: Reduce the radii of thick wire segments, or replace them with metallic strips (2D meshes) or cylinders (3D meshes) to improve convergence.

Changing the Default MLFMM Box Size

The MLFMM uses a boxing algorithm that encloses the entire computational space in a single box at the highest level, dividing this box in three dimensions into a maximum of eight child boxes and repeating the process iteratively until the side length of each child box is approximately a quarter wavelength at the finest level. Using a different box size at the finest level can sometimes facilitate convergence, although memory consumption could increase if the box size is increased.

The default box size is 0.23. A lower value decreases memory consumption while a higher value increases the memory consumption.

Tip: Try box size values between 0.2 and 0.35 wavelengths with increments of 0.02.

Using the CFIE on MLFMM Metallic Surfaces

The combined field integral equation (CFIE) uses both the electric field integral equation (EFIE) and magnetic field integral equation (MFIE). This produces a better-conditioned matrix leading to better convergence in general.

Note:

- Use the default preconditioner with the CFIE.
- CFIE can only be applied to surfaces bounding closed PEC structures.
- A mixture of CFIE and EFIE surfaces can be used.
- Sharp corners on CFIE surfaces can lead to inaccurate results.

Sharp corners should be meshed finer if CFIE is applied. If there is uncertainty, the EFIE should rather be applied around sharp corners. A rule of thumb is to apply the EFIE up to a few meshed triangles away from the sharp corners.

Using Double Precision

The Solver uses single precision by default- a single byte is used to store a complex number.

Use double precision when higher accuracy is required and to help resolve convergence issues. Double precision uses two bytes to store a complex number in the matrix. This increases the number of significant digits and reduces numerical noise.

Note:

- Double precision requires twice the memory compared to single precision.
- Double precision does not improve convergence for the stabilised MLFMM.

Changing the FEM to Use First Order Basis Functions

The FEM uses higher order (order two) basis functions by default. For large volumes, the higher order results in a much smaller number of tetrahedra in the mesh.

Related concepts

MLFMM Settings

Local Mesh Refinement

General Solver Settings - Double Precision

Automatic Meshing

Related tasks

Using the CFIE For Closed PEC Regions

A-2.8 How to Improve Convergence for the MoM / FEM

The hybrid method of moments (MoM) / finite element method (FEM) is an iterative solution method, and under certain conditions, the iterative solution may fail to converge. Several model or solution settings are presented that could improve the model's convergence behaviour.

Sometimes when using the FEM, the Solver stops with the error message:

```
ERROR 4673: Iterative solution of the system of linear equations failed, maybe try
 another pre-conditioner (solution settings).
```

The Solver stores the solution with the lowest residuum during the solution. Results are generated if this residuum is considered adequate, but the results may be less accurate if the stopping criterion for the residuum is not exactly met. In this case, the Solver output contains the warning:

```
WARNING 830: Maximum number of iterations reached without convergence, using in the
 following the solution with the smallest residuum.
```

Note: Warning 830 indicates possible inaccurate results due to inadequate convergence.

One or a combination of the following changes can be made to the model:

- Adjust the mesh.
- Change the preconditioner.
- Use double precision.
- Change the FEM to use first order basis functions.
- Change to the direct sparse solver for the FEM.

Adjusting the Mesh

Slight adjustments to the mesh size (smaller or larger elements) could lead to improved convergence. If a model is discretised too finely or too coarsely, convergence could be negatively affected.

Tip: Reduce the radii of thick wire segments, or replace them with metallic strips (2D meshes) or cylinders (3D meshes) to improve convergence.

Changing the Preconditioner

The multilevel LU / diagonal decomposition is the default preconditioner for the MoM / FEM. Changing to another preconditioner may help to achieve convergence for the FEM. Select one of the following preconditioners:

- Use the multilevel ILU / diagonal decomposition preconditioner. Set the **Fill-in level per row** to 200 and the **Stabilisation factor** to 1.
- Use the LU-decomposition of the FEM matrix.

  Note: This preconditioner is only available in EDITFEKO using the CG card.

Using Double Precision

The Solver uses single precision by default- a single byte is used to store a complex number.

Use double precision when higher accuracy is required and to help resolve convergence issues. Double precision uses two bytes to store a complex number in the matrix. This increases the number of significant digits and reduces numerical noise.

Note:

- Double precision requires twice the memory compared to single precision.
- Double precision does not improve convergence for the stabilised MLFMM.

Changing the FEM to Use First Order Basis Functions

The FEM uses higher order (order two) basis functions by default. For large volumes, the higher order results in a much smaller number of tetrahedra in the mesh.

Changing to the Direct Sparse Solver for the FEM

The direct sparse solver is not an iterative solution; therefore any convergence problems are avoided.

Related concepts

Preconditioners for FEM

General Solver Settings - Double Precision

Local Mesh Refinement

Automatic Meshing

A-2.9 How to Improve Convergence for the FDTD

The finite difference time domain (FDTD) is a solution method that may fail to converge under certain conditions. Several model or solution settings are presented that could improve the model's convergence behaviour.

Adding a 50 Ohm Load to the Feed Port

FDTD solutions, in general, converge faster when there are losses in the model. A 50 Ohm load can be added to the feed port but will affect the impedance (input reflection coefficient), far fields (gain only) and also the near fields.

In POSTFEKO the effect of this load can be removed from the impedance by selecting the **Subtract loading** check box in the result palette. The gain is lower when using a load but the directivity is unaffected.

Increasing the Total Time Interval

The maximum total time interval is the propagation time required for the time signal to pass through the domain. If the total time interval is too short, the signal has not yet propagated through the domain. Increase the total time interval to ensure the signal has propagated through the domain and therefore improving convergence.

Increase the **Time interval** to 5 to 10 times the original value to improve convergence.

> **Tip:** View the total time interval used internally by the Solver in the `.out` file.

Using Double Precision

The Solver uses single precision by default- a single byte is used to store a complex number.

Use double precision when higher accuracy is required and to help resolve convergence issues. Double precision uses two bytes to store a complex number in the matrix. This increases the number of significant digits and reduces numerical noise.

> **Note:** Double precision requires twice the memory compared to single precision.

Including Losses in the Dielectric

The FDTD shows slower convergence for lossless models, therefore including the dielectric loss could improve convergence.

Setting the Wire Radius to the Intrinsic Value

When a wire port is present in the model, convergence varies according to the wire thickness. Select the **Use intrinsic wire radius** check box to determine automatically the wire radius best suited for the voxel mesh.

Changing the Feed

If it practical for the specific model, change the feed. If a wire port is used, change to an edge port, and conversely.

Related concepts
General Solver Settings - Double Precision
Related tasks
Adding a Load
Intrinsic Wire Radius
Related reference
FDTD Frequency Settings - Total Time Interval

A-2.10 How to Reduce Computational Resources

Several tips and tricks are presented to reduce runtime and memory consumption. A few general tips are given as well as solution method-specific tips.

General Tips

General tips, which are not solution method-specific, are presented to reduce runtime and memory consumption.

- *Select the optimal solution method for the application*
  Select the solution method best suited to the model based on the electrical size of a problem, the geometrical complexity, and available computational resources.

- *Use adaptive (continuous) frequency sampling for frequency loops instead of linearly spaced sampling:*

  When computing results over a frequency band, using adaptive (continuous) frequency sampling can reduce the number of frequencies required to capture all the resonances in the result. You can initially limit the number of samples to get approximated results or use the low convergence accuracy option (fewer samples, smooth frequency response).

  > **Attention:** Using continuous (interpolated) frequency sampling with multiple solution requests in the same model could result in a long runtime.

  For example, requesting all currents over a frequency range requires interpolation of the current for each triangle/wire segment. The interpolation taking place results in extra frequencies to be computed as well as long delays between frequency samples.

- *Use coarse meshing in low current gradient or "shadow" areas*

  For example, if a plane wave is incident on an electrically large object and the radar cross section (RCS) is required, then the mesh size at the back of the object can be made coarser than the front of the object due to the very small currents at the back of the object.

  If critical output at the back of the object (away from the plane wave) is being requested, such as the near fields or the received power on an antenna at the back, then finer meshing may be required in this area.

  As the incident plane wave is uniform in phase, a mesh size coarser than the standard $\frac{\lambda}{10}$ [105] mesh size can be used. In some cases, useful results can be obtained with mesh sizes of around $\frac{\lambda}{4}$.

- *Use symmetry for hybrid methods*
  The hybrid methods are methods where the MoM is used in part, for instance MoM / PO or MoM / FEM. Geometric symmetry reduces the triangle integral calculation time and storage

105. λ is the free space wavelength.

required for the mesh elements, while electric and magnetic symmetry saves computation time and memory.

- *Use symmetry for large meshes*

  When symmetry is specified, only the part on one side of the symmetry plane is meshed and then mirrored to complete the model. Meshing is faster and the memory to store the geometry (~300 bytes/triangle) is reduced.

- *Store or re-use the solution*

  The Solver stores the current coefficients to a `.str` file by default. If no `.str` file was computed, the option can be enabled.

  If you want to change or add field computations or other outputs, then the Solver can read the solution from the `.str` file and only compute the new or changed outputs.

- *When calculating S-parameters, only enable the necessary ports*

  Each active port in an S-parameter calculation acts as a new solution. The additional solutions increase the runtime and the additional runtime depends on the solution method.

  For example, if for a two-port model, you are only interested in S_{11} and S_{21}, it is unnecessary to set port two as an active port.

- *Use domain decomposition (numerical Green's function) when many changes are made to only a small part of the model*

  For a large method of moments (MoM) model[106] where a small part of the model is changed multiple times, use the numerical Green's function (NGF). The NGF method solves the static domain (non-changing part) of the model once and then re-uses the solution (LU-decomposition) of this part together with the changing part (dynamic domain) for each new change. This method can reduce run time by a factor of 10 or more.

- *Use the DGFM to solve medium to large finite sized antenna arrays*

  The domain Green's function method (DGFM) is a method that takes first order coupling effects between antenna elements into account. However, a large saving in memory is obtained as the full matrix is not solved.

  The DGFM is used by default to solve antenna arrays defined using the finite antenna array tools unless the DGFM is de-activated

106. This include surface equivalence principle (SEP) and volume equivalence principle (VEP).

Tips for Using MLFMM

Several tips are presented to reduce runtime and memory consumption when using the multilevel fast multipole method (MLFMM).

- *Change the preconditioner*

  The sparse LU is the default preconditioner and it is the most memory intensive. Consider changing to another preconditioner.

  - Use the sparse approximate inverse (SPAI) preconditioner.
  - The default (accelerated SPAI) is fast.
  - The non-accelerated SPAI takes longer than its accelerated counterpart or the sparse LU.
  - Use the incomplete LU decomposition (ILU) preconditioner.

  > **Note:** The incomplete LU decomposition (ILU) preconditioner is only suitable for sequential solution.

- *Decrease the number of parallel processes*

  When using parallel solving for the MLFMM, the total memory increases with the number of processes due to parallel overhead[107]. Reduce memory usage by decreasing the number of parallel processes.

  > **Note:** Due to unavoidable communication between parallel processes, the solution could be slowed down by using too many parallel processes.

- *Reduce the box size*

  The default box size is 0.23 for all MLFMM models. The box size can be reduced to 0.2. Reducing the box size reduces memory and conversely.

- *Reduce runtime for the MLFMM*

  - Use the combined field integral equation (CFIE) on surfaces of the model that forms the boundary of a closed metallic region. The CFIE provides faster convergence on closed metallic problems but should be applied with caution on faces with many sharp corners.
  - If multiple plane waves are used, then each plane wave requires a new set of iterations. Reduce the number of angles to reduce the runtime.
  - In the case of RCS, the first incident angle of the plane wave should be set to a direction where the least number of multiple reflections on the model are expected. For example, for an aircraft, the first incident angle should not be set to directly illuminate the engine inlets. The Solver uses the solution from the previous angle as a phase corrected initial guess for the next incident angle. Subsequent angles, including those for cavities such as engine inlets, will then converge faster.

107. Parallel overhead is the time spent to coordinate parallel tasks.

Parallel MLFMM and Load Balancing

A load imbalance increases the memory requirements as well as the runtime. A load imbalance is detected by running the Solver with the special execution option, `--estimate-resource-requirements-only`.

The .out file will contain the following text, for example, `The local near field matrix with maximum size is located on process 3 where this maximum size = 3.44 x size of an ideal uniform distribution`.

A uniform mesh of a sphere has a very good load balance, which is very close to 1, while a large horn with fine corrugations has a load imbalance of more than 3 x size of an ideal uniform distribution.

Tips for Using PO or LE-PO

Several tips are presented to reduce runtime and memory consumption when using hybrid methods involving the physical optics (PO) and large element physical optics (LE-PO).

- *Decouple the MoM and PO parts if the coupling is negligible*

  If the interaction (coupling) between the MoM and PO parts are small, then the MoM and PO parts can be decoupled saving computational resources.

  > **Note:** It is recommended to always test this assumption by setting up a smaller model, for example, using a coarse mesh, specifying a single frequency and using a single source.

- *Use special ray tracing options*

  For applicable models, **set always illuminated** or **only illuminated from front** to reduce the ray tracing time (for example, a reflector antenna).

- *Minimize the number of MoM elements*

  If the coupling between the MoM and PO regions cannot be disabled, then reduce the computational resources by minimising the number of MoM elements (triangles/segments). Where possible use an equivalent source for the MoM part.

- *Use LE-PO*

  The LE-PO can be used in areas where the geometry is smoothly varying and several wavelengths away from the source. The LE-PO surfaces can be meshed using a mesh size of up to 2λ.

Tips for Using UTD

Several tips are presented to reduce runtime and memory consumption when using hybrid methods involving the uniform theory of diffraction (UTD).

- *Decouple the MoM and UTD parts if the coupling is negligible*
  If the UTD part of the model does not significantly influence the MoM part of the model, then the MoM and UTD can be decoupled. This removes the coupling matrix and saves computational resources.

- *Minimize the number of field points and MoM elements (or source points)*
  Each far field point or near field point requires a ray to be traced to that point from each MoM element or source via all the UTD interaction points. Reducing the number of field points and MoM elements (sources) reduces the computation time. A radiation pattern point source in a UTD solution is at least *M* times faster than a near field source where *M* is the number of near field points in the source.

- *Reduce the number of polygonal plates*
  The UTD method traces an optical ray from each MoM element (or source point) to each far field or near field point. Although no mesh elements are stored for UTD parts, a ray is traced to each edge, corner and plate centre of each polygonal plate in the model. Therefore removing unnecessary geometry from the UTD model reduces the runtime.

Tips for Using RL-GO

Several tips are presented to reduce runtime and memory consumption when using hybrid methods involving the ray launching geometrical optics (RL-GO).

- *Reduce the number of interactions*

  The MoM / RL-GO is an optical technique that is based on the principle of ray-launching. From the MoM elements or sources, rays are launched in all directions. Where a ray hits geometry with the RL-GO setting, a Huygens source is created.

  The RL-GO requires that Huygens sources are spaced a minimum distance apart to maintain accuracy. If there is more than one interaction specified (the default is three), additional rays are launched from the previous set of Huygens sources, resulting in more Huygens sources. Once all the interactions are computed, integration over all the Huygens sources is done to obtain the solution.

- *Decouple the MoM and RL-GO parts if the coupling is negligible*
  If the RL-GO part of the model does not significantly influence the MoM part of the model, then the MoM part can be decoupled from the RL-GO part to reduce computational resources.

- *Reduce the number of sources*
  Each MoM unknown or source element acts as a source for the RL-GO region. Rays are launched from each MoM element or source point. Using an equivalent source (for example, radiation pattern point source, spherical expansion mode source, Hertzian dipole) reduces

computational resources for the MoM / RL-GO as there is only a single source for the geometry part treated with the RL-GO.

Tips for Using FEM

Several tips are presented to reduce runtime and memory consumption when using hybrid methods involving the finite element method (FEM).

- *Decouple the FEM and MoM parts if the coupling is small*

  If the FEM part of the model does not have a significant influence on the MoM part of the model, then the MoM and FEM can be decoupled. This removes the coupling matrix and reduces computational resources.

  > **Note:** For FEM models containing dielectric surfaces (excluding surfaces for modal ports) on the outer boundary of the model, add an air layer with a thickness of at least $\frac{3\lambda}{10}$[108] around the model when using the decoupling method.

  The air layer reduces the number of surface triangle elements on the boundary between the FEM and the MoM. Resource requirements are reduced when the number of triangles decreases.

- *Use a surrounding dielectric air layer to reduce the number of surface triangles*

  If the outer surfaces require a mesh size much finer than $\frac{\lambda}{10}$, add a surrounding dielectric layer of air (relative permittivity of 1). This surrounding air dielectric creates a new outer surface that can be meshed with a size of roughly $\frac{\lambda}{10}$. The computational resources will be reduced with the reduced number of triangles on the outer FEM surface.

- *Model internal free space regions as PEC FEM regions*

  Set internal free space regions to perfect electric conductor (PEC) and set its solution method to FEM. This setting reduces the coupling matrix and as a result reduces the computational resources.

- *Use first order basis functions if the geometry requires a very fine mesh*

  If the FEM part of the model requires very fine meshing compared to the wavelength in the dielectric (for example, geometrically complex objects) then using first order basis functions can reduce the computational resources and often improve the convergence as well.

  If the MoM part is electrically large, then use the MLFMM / FEM instead.

108. λ is the free space wavelength.

Tips for Using FDTD

Some tips are presented to reduce runtime and memory consumption when using the finite difference time domain (FDTD).

- *Adjust the bounding box where applicable*
  Reduce the size of the finite difference time domain free space buffer to reduce the number of voxels in the mesh. Fewer voxels results in a reduction of the memory requirement and run time.

  > **Note:** Reducing the bounding box could reduce the accuracy.

- *Select near field request points to be within the voxel mesh*
  A voxel is stored for each near field point. If the near field point is outside the bounding box that would have been used without the request, the number of the voxels are increased. The increase in the number of voxels results in an increase in the computational resource requirements. If the near field points are closely spaced, the voxel mesh needs to be closely spaced as well.

A-2.11 How to Feed a Microstrip Line Using an Infinite Substrate

Various methods are presented on how to feed a microstrip line using an infinite planar multilayer substrate in CADFEKO.

A microstrip line consists of a signal conductor (transmission line) fabricated on top of a dielectric substrate with a conductive ground plane at the bottom.

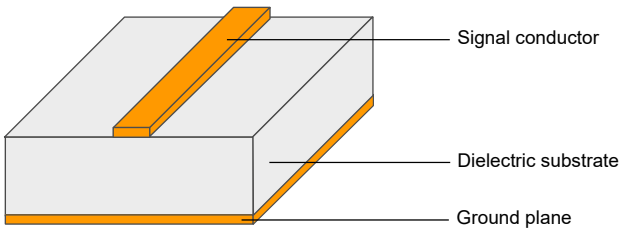


Figure 987: A graphical representation of a microstrip line using an infinite dielectric substrate.

The following feeding methods are discussed:

- wire port
- edge port
- microstrip port

Constructing a Microstrip Line

Create a two-port 50 Ohm microstrip line using a planar multilayer substrate in CADFEKO.

1. Design a 50 Ohm microstrip line on a substrate of height 0.787 mm and permittivity of 2.33, using one of the following methods:
   - Use a third-party tool (transmission line calculator) to calculate the dimensions of the microstrip line.
   - Calculate the dimensions of the microstrip line using equations from literature.
2. Create the microstrip line in CADFEKO using the dimensions obtained in Step 1.

   Tip: Use a planar multilayer substrate for the dielectric layer and ground plane.
3. To ensure finer meshing along the length of the microstrip line:
   a) Copy the side edges and translate the edges a distance of 1/7 of the microstrip line width.

Figure 988: The two side edges are translated from the edge a distance of 1/7 the microstrip line width.

a) Define a local mesh refinement on the two translated edges.

Constructing a Feed for a Microstrip Line

Three methods for feeding a microstrip line are presented.

Wire Port

The feed is constructed using a line and fed using a wire port.

A feed structure (a line) is added to both sides of the microstrip line to connect the microstrip line to the ground plane. A wire port is added to the feed structure.

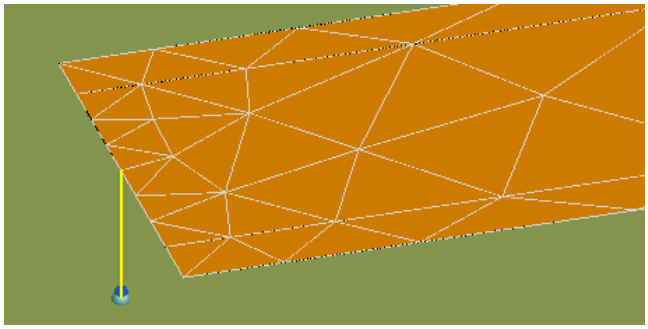


Figure 989: Isometric view of the partial microstrip line. The feed structure is a line (highlighted in yellow) and fed using a wire port.

Related concepts
Wire Ports

Edge Port

The feed is constructed using rectangles and polygons and fed using an edge port.

A feed structure is added at both sides of the microstrip line to connect the microstrip line and the ground plane. An edge port is added to the feed structure.

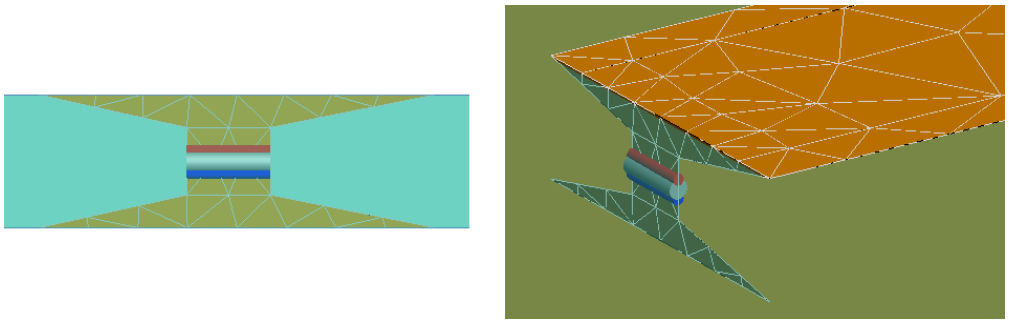


Figure 990: On the left, a side view of the feed structure. To the right, an isometric view of the partial microstrip line. The edge port is added to the feed structure.

Related concepts
Edge Ports

Microstrip Port

Different configurations using a microstrip port are considered.

The first configuration considered is to define a microstrip port on the full width of the microstrip line. For the second and third configurations, a feed structure is created using polygons to decrease the width of the microstrip port.

Narrow Version

To decrease the width of the microstrip port, a polygon is added to the microstrip line to act as the feed structure. For better accuracy, ensure that the width of the microstrip port is roughly $\frac{1}{5}$ of the microstrip line width.

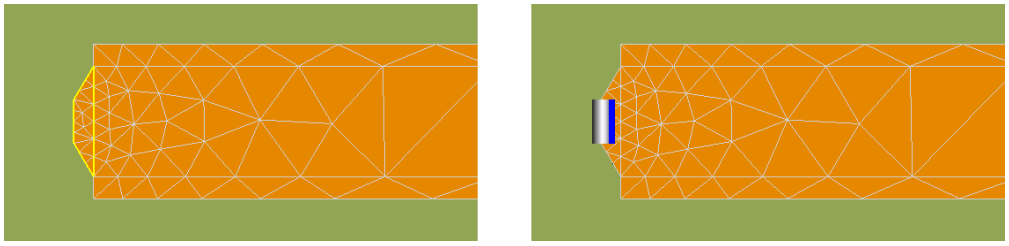


Figure 991: Top view of the partial microstrip line. On the left, the feed structure is indicated in yellow. To the right, the microstrip port is added to the feed structure.

Related concepts
Microstrip Ports

A-2.12 How to Connect an Antenna to a HyperMesh Generated Mesh

A workflow is presented on how to connect an antenna model designed in CADFEKO to a platform model that was meshed with Altair HyperMesh.

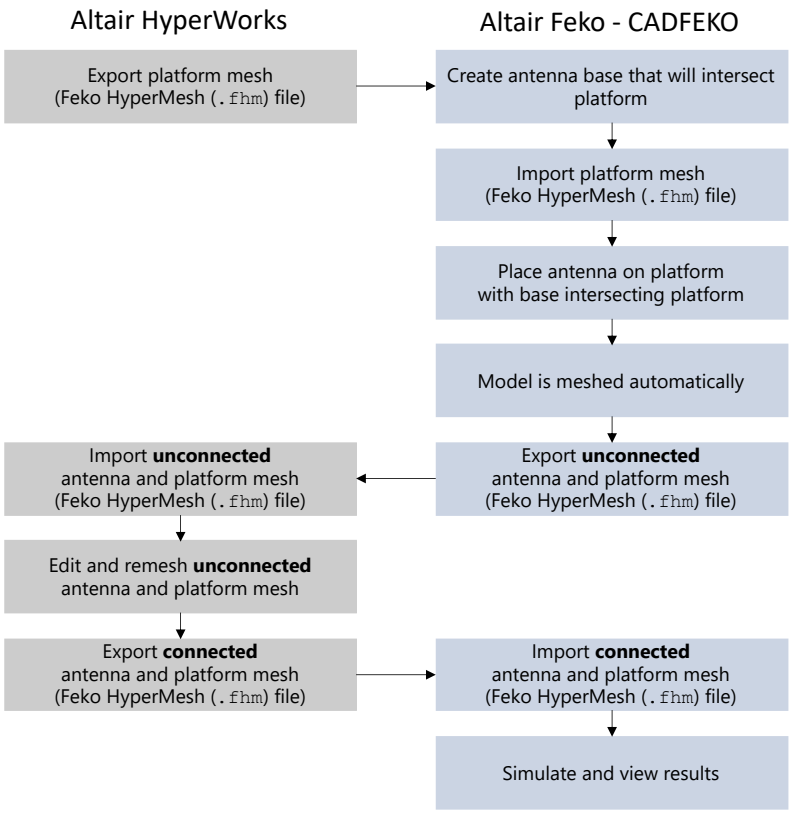


Figure 992: Workflow to connect a CADFEKO antenna to a platform model meshed with HyperMesh.

Example

As an example, a circular patch antenna designed in CADFEKO to operate at 1 GHz, will be placed on a platform (roof of a vehicle) that was meshed using HyperMesh.

Figure 993: On the left, the circular patch antenna. To the right, the platform (vehicle) model meshed with HyperMesh. Images not to scale.

Exporting the Platform Mesh in HyperWorks

The mesh of the platform is exported to a Feko HyperMesh (`.fhm`) file which contains the mesh elements, material properties, and assignment data.

1. Open Altair HyperWorks.
2. Open a HyperMesh model (`.hm`) file to be exported as the platform model.

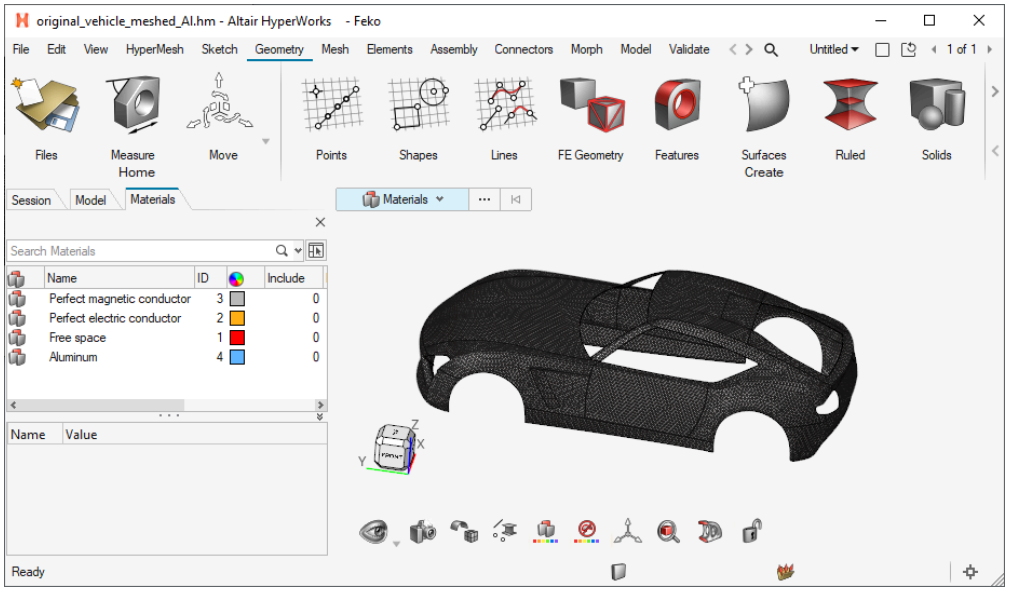


Figure 994: An example of a HyperMesh model. The vehicle is meshed and the material is defined as Aluminum and applied as a property to the mesh elements.

3. To export the mesh, click **File** > **Export** > **Solver Deck** to export the mesh.
   a) Specify a file name and save as type **Feko (*.fhm)**.

> **Tip:** Keep HyperWorks open since the antenna and platform will be remeshed.

Setting Up the Antenna in CADFEKO

To ensure mesh connectivity between the antenna mesh and platform mesh in the steps that follow, create a base for the antenna that will protrude and intersect the platform.

1. Open CADFEKO.
2. Open the antenna model.

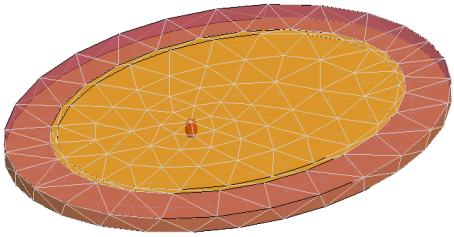


Figure 995: Circular patch antenna meshed using CADFEKO with mesh opacity at 60% showing the wire feed.

3. Create the antenna base and set the region medium to perfect electric conductor (PEC).

> **Tip:**
>
> 1. Create the base by making a copy of the antenna geometry.
> 2. Translate the base to sit below the antenna.

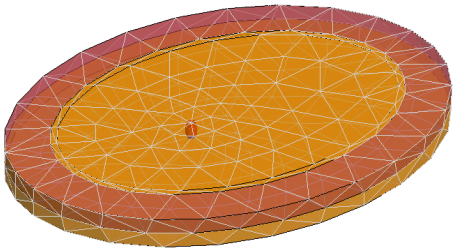


Figure 996: A base was created and the region medium set to PEC.

4. Ensure connectivity between the antenna and its base by using one of the following workflows:
   - Union the antenna and its base.
   - Move[109] the base into the antenna union. The **Move in** operation retains the label of the antenna in the tree.

Related concepts
Union

109. Drag the base into the antenna union and from the right-click context menu, select **Move in**.

Importing the Platform Mesh into CADFEKO

The mesh of the platform (`.fhm` file) is imported into CADFEKO model containing the antenna.

> **Note:** The `.fhm` file contains the mesh, material properties, and assignment data.

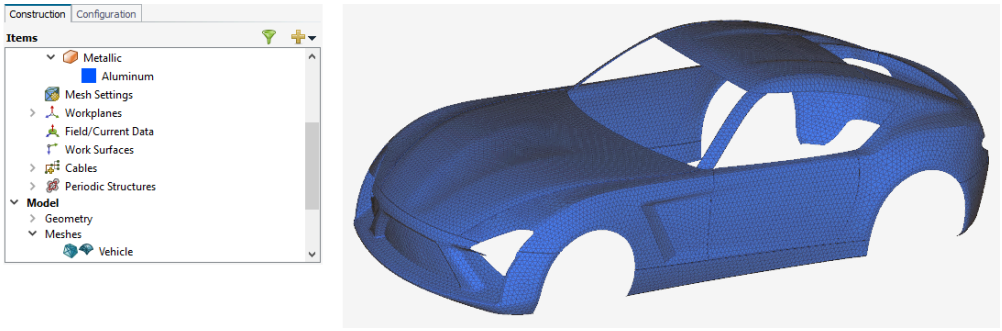


Figure 997: 3D view of the imported `.fhm` showing the imported mesh and its media properties.

The model now contains the antenna and the imported mesh.

Related tasks
Importing a Mesh

Placing the Antenna on the Platform in CADFEKO

The antenna is placed on the platform and remeshed.

1. Unlink the mesh of the antenna and transfer the solution entities to the new port.
2. Delete the geometry instance of the antenna (only the antenna mesh will be used going forward).
3. Place the antenna on the platform (roof of the vehicle) using the Align tool.

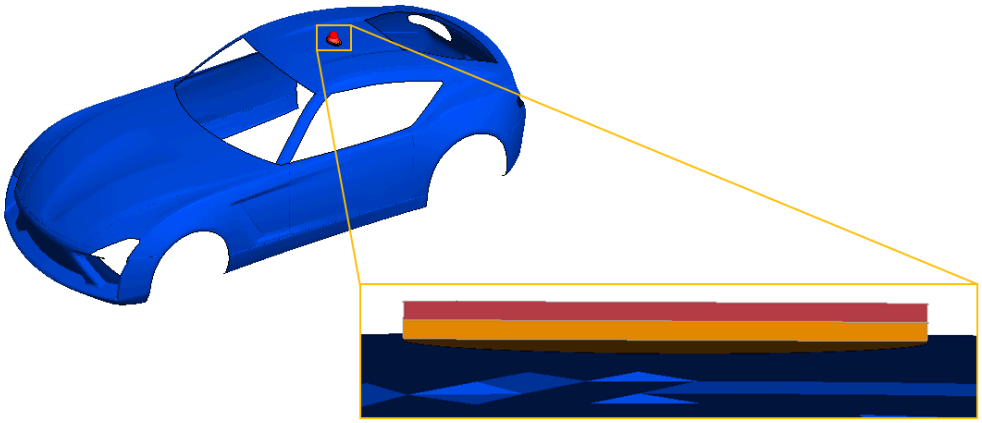


Figure 998: A 3D view showing the antenna placed on the roof of the vehicle. A cross-sectional view shows how the platform only intersects the PEC base.

> **Note:** There is no mesh connectivity between the antenna and the platform.

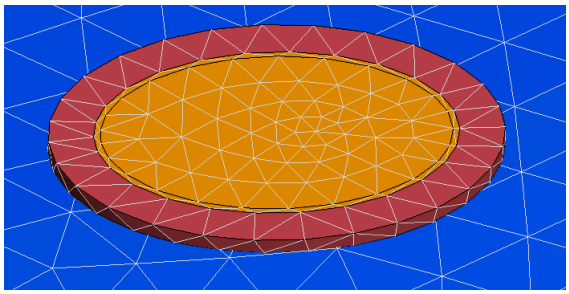


Figure 999: Antenna was remeshed automatically but there is no mesh connectivity yet.

Related concepts
Mesh Editing
Related tasks
Placing Geometry on Objects (Align)
Unlinking a Mesh

Exporting the Mesh in CADFEKO

The full model (antenna and platform) is exported to a Feko HyperMesh (`.fhm`) file.

Click **File** > **Export** > **Mesh** and export the antenna mesh and platform mesh to a `.fhm` file.

Related tasks
Exporting a Mesh

Remeshing the Antenna and Platform in HyperWorks

The platform and antenna base are unioned using the **Mesh Boolean** tool and remeshed.

1. Go back to Altair HyperWorks.
2. Create a new model.
3. Click **File** > **Import** > **Solver Deck** to import the `.fhm` file.

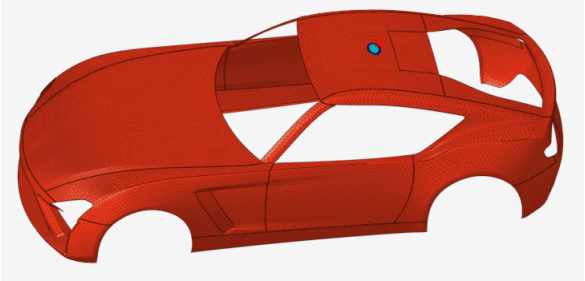


Figure 1000: 3D view of the imported `.fhm` file in Altair HyperWorks.

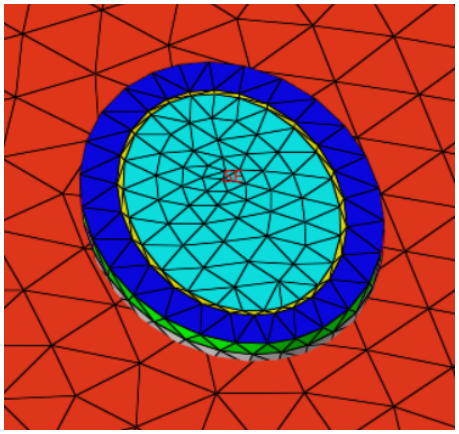


Figure 1001: Note that there is no mesh connectivity between the antenna mesh and platform mesh.

4. View the bottom face of the antenna base.

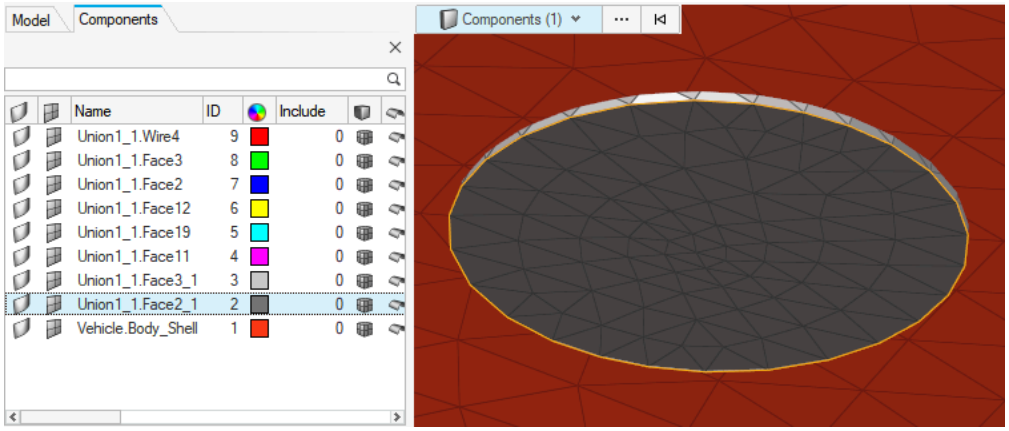


Figure 1002: 3D view showing the bottom face of the antenna base protruding through the underside of the roof.

5. Delete the bottom face of the antenna base.

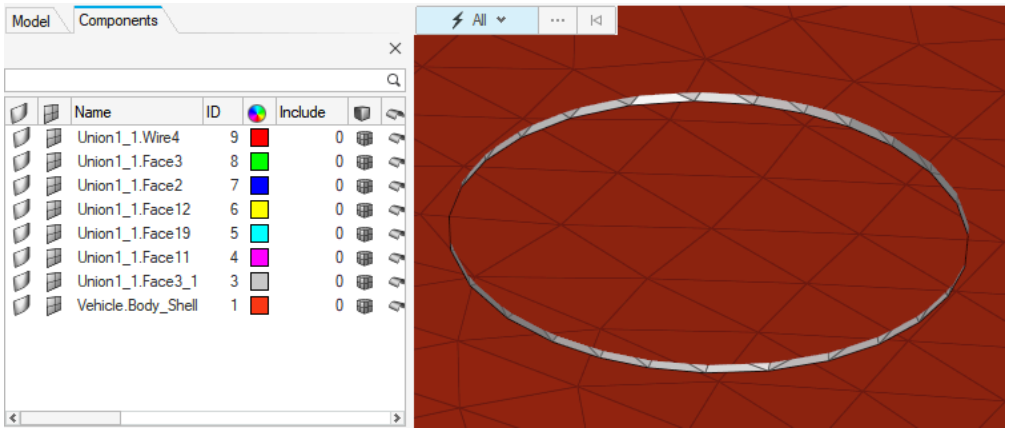


Figure 1003: 3D view showing the model after the bottom face was deleted.

6. To union the platform and antenna base, click **Elements** > **Edit** > **Mesh Boolean**.
   The **Boolean Operation** dialog is displayed.

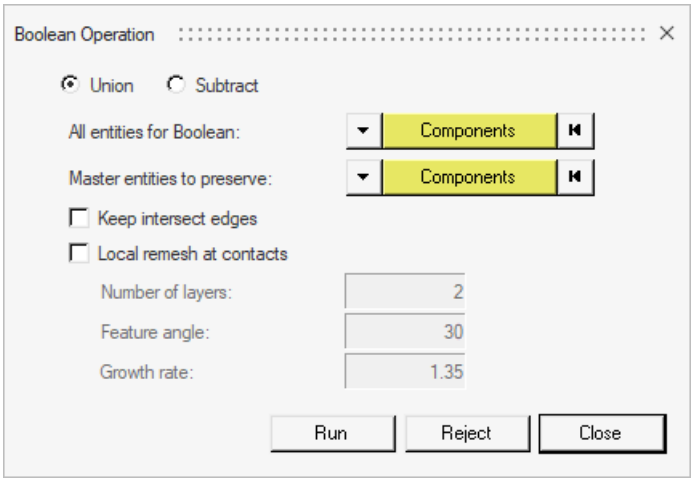


*Figure 1004: The **Boolean Operation** dialog.*

7. Confirm that **Union** is selected.
8. Next to **All entities for Boolean**, double-click **Components**.
   The **Components** toolbar is displayed.
9. Select the platform (vehicle) and the face that protrudes through the vehicle roof and click ✓.
10. On the **Boolean Operation** dialog, click the **Local remesh at contacts** check box.
11. Click **Run** to remesh the antenna and platform.

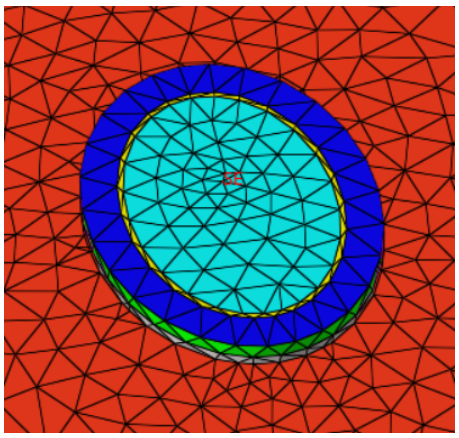


Figure 1005: The antenna and platform now have mesh connectivity.

12. Delete all antenna mesh elements that remain below the vehicle roof.
13. Click **File** > **Export** > **Solver Deck** and export to a `.fhm` file.

Importing the Connected Antenna and Platform in CADFEKO

The connected antenna mesh and platform mesh are imported into CADFEKO and simulated.

1. Import the `.fhm` file of the connected antenna base and platform.
2. Replace the original antenna mesh with the new imported antenna mesh.
3. Replace the original platform mesh with the new imported platform mesh.

   > **Note:** The **Replace with** mesh option allows you to transfer all settings applied to the old mesh, to the new mesh.

4. Activate the required solver settings.
5. Save the model.
6. Run the Solver and view the results.

Related tasks
Replacing a Mesh

A-2.13 How to Interpret Far Fields Calculated from a PBC Solution

This How-To helps to interpret far-fields (total and scattered) calculated with the periodic boundary condition (PBC) solution.

For the PBC solution, the default far field is calculated from the solution of a single unit cell radiating in free space. The PBC unit cell solution is obtained for an infinite array, meaning all the array element currents/fields are the same.

For a PBC unit cell, consider a truncated patch (derived from the component library) that radiates left-hand circular (LHC) polarisation.

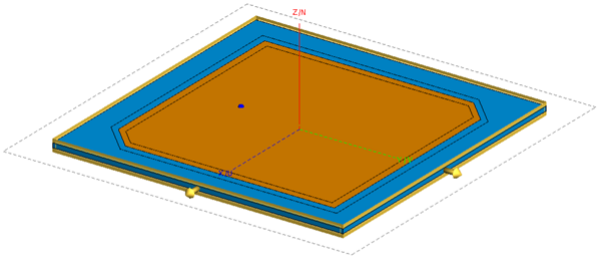


Figure 1006: The truncated patch PBC unit cell geometry.

Consider two cases:

- The PBC patch is exited with a voltage source with a specified beam pointing angle.
- The PBC patch is excited with a plane wave, and the port is loaded with 50 Ohm.

Voltage Source Excitation

The PBC solution is obtained by exciting the patch with a voltage to have a beam pointing angle {theta, phi} = {30, 0} degrees.

> **Note:** The unit cell beam will not necessarily point to the beam pointing angle.

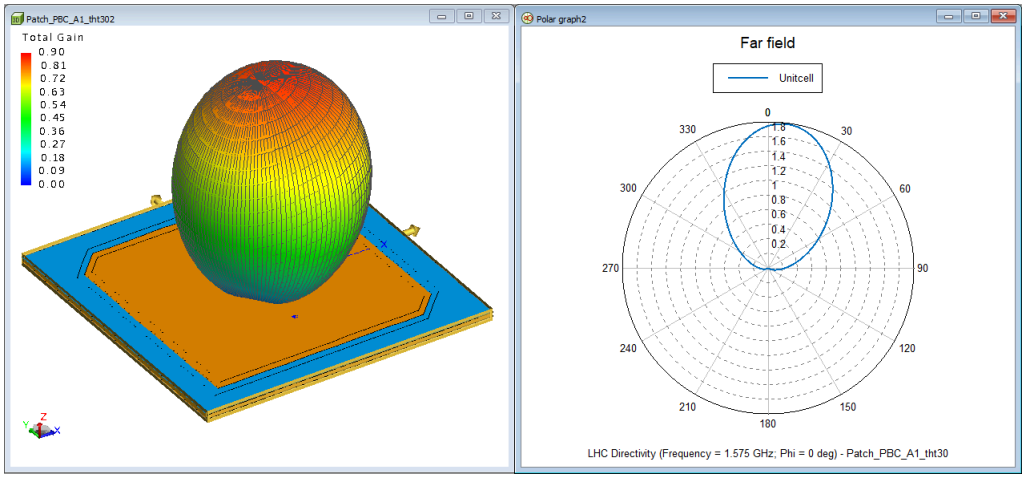


Figure 1007: The LHC far field directivity pattern of the unit cell.

It is only when requesting the far field[110] for a large finite array that the beam pattern will align with the specified pointing angle.

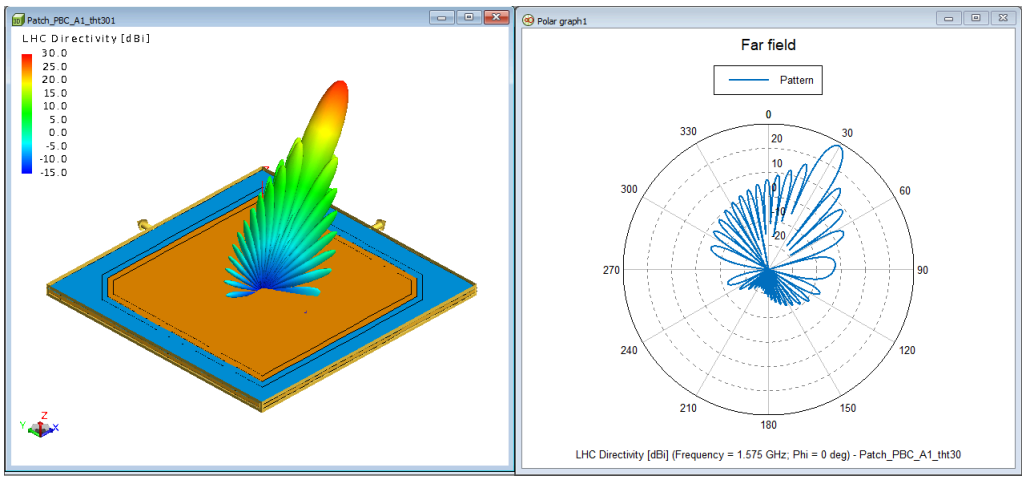


Figure 1008: The LHC far field directivity of a 21x21 element array excited uniformly. Only the unit cell geometry is displayed in the 3D view.

Plane Wave Excitation

The PBC solution is obtained for two sets of incident plane waves: one with left-hand circular polarisation (LHC) and another with right-hand circular (RHC) polarisation. Request the bi-static far field for the unit cell (default) and a 21x21 element array.

Figure 1009 shows the bi-static RCS pattern for a 21x21 array.

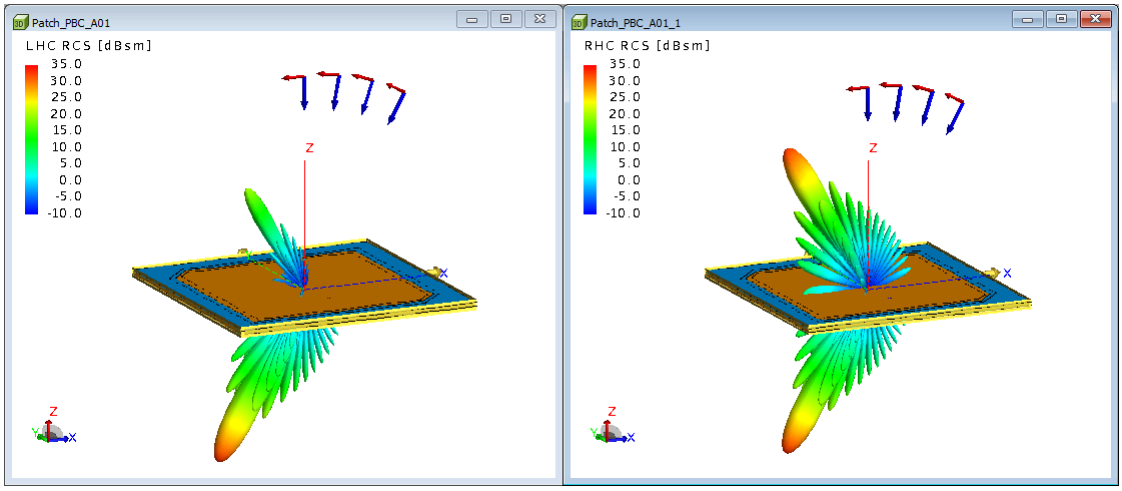


Figure 1009: The co-polarised bi-static RCS for the 21x21 array with incident angle (theta_i, phi_i} = {30, 0}. Left: LHC polarised plane wave; Right: RHC polarised plane wave.

110. Some advanced options are available on the **Request Far Fields** dialog (**Advanced** tab): option to calculate the far field from a finite array instead of one unit cell; option to calculate the scattered field only.

Again, only the unit cell geometry is displayed in the 3D view. The bi-static RCS includes only the scattered field from the array. It cannot include the plane-wave field contribution as the plane-wave E-field does not decay with 1/r. This is the reason for the large forward scatter beam below the array. A large forward scatter beam is needed to cancel the incident field.

The total near field, which includes the plane wave field contribution, below the unit cell will be zero. Numerically it will have some finite (but insignificant small) level due to the discrete nature of the current solution, typically >40 dB below the incident field level.

The patch array reflects less power when the incident polarisation is matched (LHC), compared to when the incident polarisation is mismatched (RHC).

This is consistent with the received power in the 50 Ohm load versus incident angle for LHC and RHC incident polarisation.

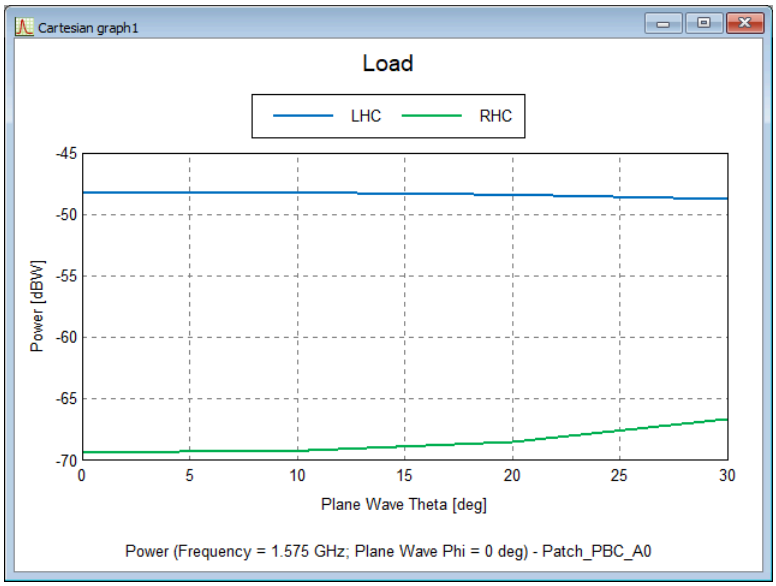


Figure 1010: The co-polarised bi-static RCS for the 21x21 array with incident angle (theta_i, phi_i} = {30, 0}. Left: LHC polarised plane wave; Right: RHC polarised plane wave.

More power is received when the incident polarisation (LHC) is matched to pattern polarisation (LHC) when the patch is excited.

Related concepts

Advanced Settings for Far Field Requests

Related tasks

Requesting a Far Field

Defining a Periodic Boundary Condition (PBC)

A-2.14 How to Install CADFEKO [LEGACY] Using the Standalone Installer

This How-To provides background on the standalone legacy CADFEKO installer as well as the installation process.

Background

From Feko 2023.1 and onward, legacy CADFEKO is not included in the Feko installer.

For some Feko users, however, there are models and workflows where the legacy CADFEKO interface may be required. To enable these Feko users to continue using the legacy CADFEKO interface for a period, a separate or "standalone " installation has been prepared. This installation will install only the files and environment needed to run the legacy CADFEKO interface.

As the standalone legacy CADFEKO installer does not install any other Feko components, a separate installation of Feko is required for any workflows requiring other Feko components (for example, the Solver or POSTFEKO).

Environment variables can be used to configure which Feko components should be used with legacy CADFEKO (for example, if multiple concurrent installations of Feko are installed).

How to Install Standalone Legacy CADFEKO

1. Install the full 2023.1 Feko installation or a Feko version newer than Feko 2023.1.

   > **Note:** If Feko 2023.1 or newer installation is already in place, or if only the legacy CADFEKO interface is needed, this step can be skipped.

2. Download the legacy CADFEKO installer (download link available in Altair Community article).
3. Install legacy CADFEKO.
   - Specify the path to the Feko installation on the **Choose Existing Feko Installation** installer panel (for example, `C:\Program Files\Altair\2023.1`).

If a 2023.1 or later Feko installation does not exist when installing legacy CADFEKO or legacy CADFEKO does not point to the correct version, complete the following steps:

1. Modify the `FEKOenvironmentFromSetup.lua` file located in the `%FEKO_HOME%` directory, where `%FEKO_HOME%` points to the Feko installation path (for example: `C:\Program Files\Altair\2023.1_Legacy_CADFEKO`).
2. Enter the path the full Feko installation to be used by legacy CADFEKO (as an example, using the default 2023.1 installation path):
   - Microsoft Windows
     - `setEnv([[FEKO_LEGACY_CADFEKO_FEKO_HOME]], [[C:\Program Files\Altair\2023.1\feko]], true);`
   - Linux
     - `setEnv([[FEKO_LEGACY_CADFEKO_FEKO_HOME]], [[/opt/feko/2023.1/altair/feko]], true);`

3. Add the line to specify the path to the legacy CADFEKO installation for the case where CADFEKO_BATCH needs to be called from RUNFEKO:
   - Microsoft Windows
     - `setEnv([[FEKO_LEGACY_CADFEKO_BINARY_PATH]], [[C:\Program Files\Altair\2023.1_Legacy_CADFEKO\feko\bin]], true);`
   - Linux
     - `setEnv([[FEKO_LEGACY_CADFEKO_BINARY_PATH]], [[/opt/feko/2023.1_Legacy_CADFEKO/altair/feko/bin]], true);`

How to Run the Solver Using Legacy CADFEKO

To run the Solver using legacy CADFEKO (locally or on a cluster), do the following:

1. See How to Install Standalone Legacy CADFEKO on how to install legacy CADFEKO.
2. Start legacy CADFEKO using one of the following workflows:
   - Start legacy CADFEKO, load a model and run the Solver.
   - Start legacy CADFEKO, open the (**Home** tab) and call the Solver using the command line.

How to Run the Solver Using Legacy CADFEKO_BATCH

To run the Solver using legacy CADFEKO_BATCH (locally or on a cluster), do the following:

1. See How to Install Standalone Legacy CADFEKO on how to install legacy CADFEKO.
2. Open a command prompt and specify the following environment variables:
   - FEKO_LEGACY_CADFEKO_FEKO_HOME = C:\Program Files\Altair\2023.1\feko (point to the full Feko installation)
   - `FEKO_LEGACY_CADFEKO_BINARY_PATH = C:\Program Files\Altair\2023.1_Legacy_CADFEKO\feko\bin` (point to the legacy CADFEKO installation)
   - `FEKO_LEGACY_CADFEKO=1` (to indicate that the legacy CADFEKO_BATCH should be used when RUNFEKO attempts to run CADFEKO_BATCH)
   - Run the Solver.

> **Note:** The following environment variables are set automatically when opening the Feko Terminal from legacy CADFEKO:
>
> - `FEKO_LEGACY_CADFEKO_FEKO_HOME = C:\Program Files\Altair\2023.1\feko`
> - `FEKO_LEGACY_CADFEKO_BINARY_PATH = C:\Program Files\Altair\2023.1_Legacy_CADFEKO\feko\bin`
> - `FEKO_LEGACY_CADFEKO=1`

A-2.15 How to Migrate Legacy CADFEKO Lua Scripts to the New CADFEKO API Format

The `ConvertLuaScript` utility can be used to migrate legacy CADFEKO Lua scripts to the latest CADFEKO API format.

Important:

- Migration is **only applicable** to CADFEKO Lua scripts created prior to version 2022.1.
- Migration is **not** applicable to POSTFEKO Lua scripts.

To download the utility and view the instructions, refer to Altair Community article for more details.

FAQ

A collection of frequently asked questions (FAQ) relating to advanced concepts are included.

A-3.1 What is the Maximum Character Path Length?

Feko supports a maximum character path length of 512 characters for Microsoft Windows when the registry key `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\LongPathsEnabled (Type: REG_DWORD)` exists and is set to 1.

The maximum path consists of a drive letter, colon, backslash, subfolders (separated by backslashes), name and a terminating null character.

For example:

```
C:\some_folder\name_of_fileNULL
```

where "NULL" is the invisible terminating null character for the current system codepage. For the above example, the path is 28 characters long.

If a path length exceeds the maximum path length of 512 characters, the too-long path can be reduced in the following ways:

- Rename a long file name.
- Reduce the folder depth by moving the folder.
- Use UNC pathing in Microsoft Windows.

A-3.2 Which Solution Methods Support GPU Acceleration?

Feko supports the use of a graphics processing unit (GPU) for simulation acceleration using the unified device architecture (CUDA) framework from NVIDIA for sequential (single CPU core) runs.

Note: Minimum requirements for the CUDA device:

- Compute capability of at least 3.5
- Driver installed on system must support CUDA 11.6.

The computational phases targeted for execution on a CUDA-based GPU show a significant speedup compared to standard CPU-based execution.

The following solution methods support GPU acceleration:

MoM

Includes SEP, VEP, planar Green's function (multilayer substrates) and thin dielectric sheets for single and double precision matrix solve (LU decomposition).

Support single or multiple GPUs.

CMA

Supports multiple GPUs.

RL-GO

For manual (fixed) ray launching grid and flat meshes (not curvilinear).

Supports single GPU.

FDTD

Supports single GPU.

Troubleshooting

Common problems that you may encounter are discussed as well as their solutions.

A-4.1 Crash When Using CADFEKO Over Remote Desktop

Problem

Clicking on **New Project** when using CADFEKO over a remote desktop connection, results in a crash.

Cause

3D support for remote desktop is disabled for the host machine's graphics card.

Solution

1. Enable 3D support on host machine for remote desktop.
   a) Open the Microsoft Windows **Start** menu.
   b) Type **Local Group Policy** and click **Edit group policy**.
   c) On the **Local Group Policy Editor** dialog, click **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Remote Session Environment**.
   d) Enable the following:
      - Use the hardware default graphics adapters for all Remote Desktop Services sessions
      - Prioritize H.264/AVC 444 graphics mode for Remote Desktop Connections
      - Configure H.264/AVC hardware encoding for Remote Desktop Connections
      - Configure compression for RemoteFX data
      - Configure image quality for RemoteFX Adaptive Graphics
      - Enable RemoteFX encoding for RemoteFX clients designed for Windows Server 2008 R2 SP1
      - Configure RemoteFX Adaptive Graphics

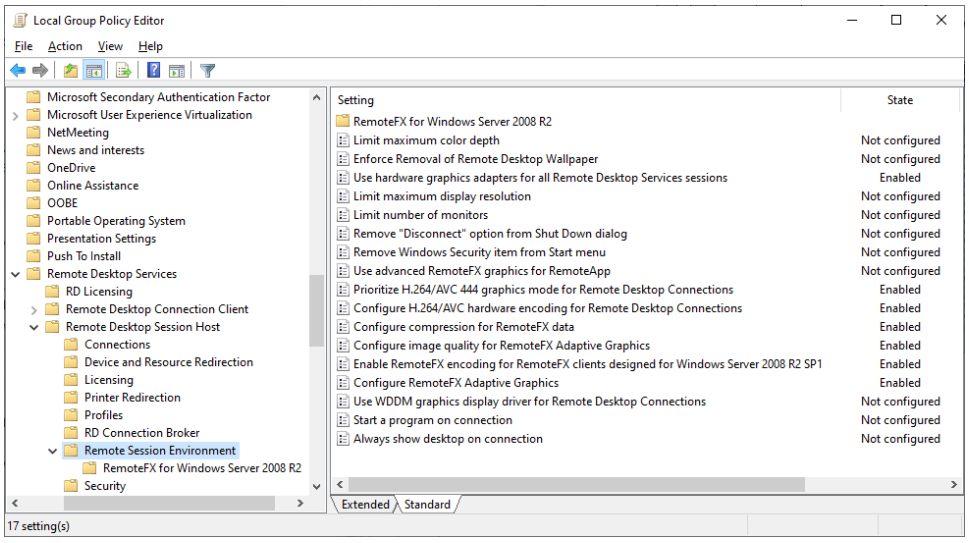


*Figure 1011: The **Local Group Policy Editor** dialog in Microsoft Windows.*

2. Download a special patch for NVIDIA graphics card drivers from https://community.altair.com/.

A-4.2 Drop-Down List Flashing with Mouse Over

Problem

A drop-down list on a dialog in CADFEKO "flashes" with mouse over.

Cause

The Microsoft Windows option to animate controls and elements inside dialogs are enabled.

Solution

Disable the option to animate controls and elements inside dialogs.

a) Open the Microsoft Windows **Start** menu.
b) Type **Performance Options** and click **Adjust the appearance and performance of Windows**.
c) On the **Performance Options** dialog, click **Custom**.
d) Unselect the **Animate controls and elements inside windows** check box.

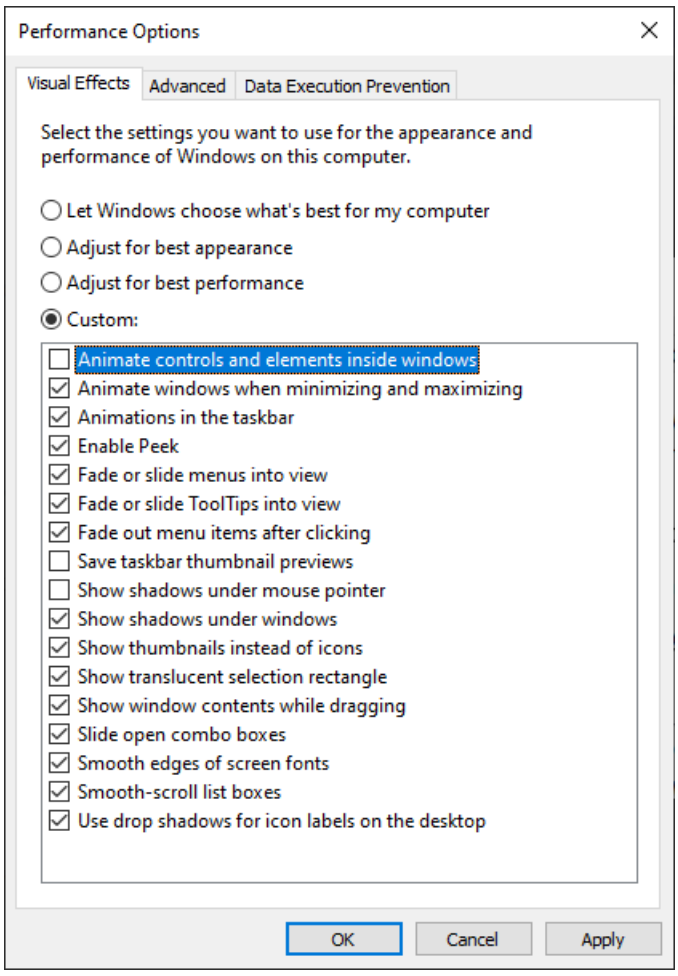


*Figure 1012: The **Performance Options** dialog.*

e) Click **OK** to close the **Performance Options** dialog.

A-4.3 CADFEKO [LEGACY] Model Fails to Open in CADFEKO

Problem

Opening an old CADFEKO [LEGACY] model in CADFEKO fails to convert to the new format.

Cause

CADFEKO uses advanced mapping algorithms to keep track of individual items when the geometry is modified. A model that was saved in an earlier CADFEKO [LEGACY] may not have all the mapping information available.

Remedy 1:

1. Open the model in a newer CADFEKO [LEGACY] version and resave the model[111].
2. Re-evaluate the geometry.
3. Save the model.
4. Reopen the model in CADFEKO.
   If the model still fails to open in CADFEKO then perform the steps below.

Remedy 2:

1. Follow the steps in 1 and 2.
2. Convert the geometry to a primitive.

   Tip:

   If the model consists of multiple parts[112], only one might need to be converted.
3. Save the model.
4. Reopen the model in CADFEKO.

Related tasks
Re-Evaluating Geometry

111. See How to run CADFEKO [LEGACY] for more information.
112. The highest-level items in the model are referred to as "parts".

Meshing

A-5

When meshing a model, you can either use the automatic meshing algorithm to calculate the appropriate mesh settings or you can specify the mesh sizes. When you specify the mesh sizes, the mesh sizes should adhere to certain guidelines.

A-5.1 Automatic Meshing

Automatic meshing takes into account the solution methods, ports, material properties and the simulation frequencies to automatically determine the appropriate mesh sizes.

It can be rather complex and a daunting task to specify the appropriate mesh types and mesh sizes for the different solution methods supported in Feko. An automatic meshing algorithm in CADFEKO takes into account the solution method, special properties applied, material properties and the simulation frequency to define appropriate mesh settings. You can select between coarse, standard or fine meshing, based on the required accuracy and the available resources (including time).

A custom mesh size can also be set for users who do not want to use the automatic meshing algorithm. All local mesh settings take precedence over the global and automatic settings.

Automatic Meshing for Wires

The automatic mesh sizes faces and edges when using the coarse, standard and fine mesh options are dependent on the solver method.

For wires, the wavelength (λ) is determined based on the maximum simulation frequency and the surrounding medium.

Type	Fine	Standard	Coarse
Method of moments (MoM)	$\frac{\lambda}{25}$	$\frac{\lambda}{12}$	$\frac{\lambda}{8}$

Automatic Meshing for Faces and Edges

The automatic mesh sizes faces and edges when using the coarse, standard and fine mesh options are dependent on the solver method.

If any bounding regions of a face have user-defined local mesh sizes, the face will inherit the smallest of the local mesh sizes. An edge will inherit the smallest mesh size from its bounding faces.

- Bounding face of a FEM region: The first order automatic mesh size of that region is used.
- Bounding face of a VEP region: The mesh size of that region is used.
- Metal face bounding a SEP region: The smallest wavelength in the media bounding the face is used to determine the mesh size.

In all other cases the largest wavelength for the bounding media is used to determine the mesh size. When higher order basis functions are used for the MoM, junction edges (where multiple faces share an edge) and open edges (where an edge only has one face bordering it) are meshed as for RWG or order 0.5.

Type	Fine	Standard	Coarse
MoM (RWG or 0.5 order basis function)	$\frac{\lambda}{16}$	$\frac{\lambda}{12}$	$\frac{\lambda}{8}$
MoM (1.5 order basis function)	$\frac{3\lambda}{16}$	$\frac{\lambda}{4}$	$\frac{3\lambda}{8}$
MoM (2.5 order basis function)	$\frac{\lambda}{3.2}$	$\frac{\lambda}{2.4}$	$\frac{\lambda}{1.6}$
MoM (3.5 order basis function)	$\frac{\lambda}{1.6}$	$\frac{\lambda}{1.2}$	$\frac{\lambda}{0.8}$
Physical optics (PO)	$\frac{\lambda}{10}$	$\frac{\lambda}{8}$	$\frac{\lambda}{6}$
Large element PO (LE-PO)	$\frac{9\lambda}{5}$	$\frac{9\lambda}{5}$	$\frac{9\lambda}{5}$
Ray launching geometrical optics (RL-GO)	∞	∞	∞
Faceted UTD	∞	∞	∞

Note: The ∞ symbol indicates that the mesh should accurately represent the geometry, for both flat and curvilinear mesh elements.

Automatic Meshing for Regions

The automatic mesh sizes for faces and edges when using the coarse, standard and fine mesh options are dependent on the solver method.

For regions, the wavelength (λ) is determined based on the maximum simulation frequency and the medium properties of the region. The specified lengths will be applied to the tetrahedron edge lengths.

Type	Fine	Standard	Coarse
Finite element method (1st order FEM)	$\frac{\lambda}{15}$	$\frac{\lambda}{10}$	$\frac{\lambda}{8}$
Finite element method (2nd order FEM)	$\frac{\lambda}{12}$	$\frac{\lambda}{8}$	$\frac{\lambda}{6}$
Volume equivalence principle (VEP)	$\frac{\lambda}{16}$	$\frac{\lambda}{12}$	$\frac{\lambda}{8}$

Automatic Meshing for Voxels

The automatic mesh sizes for voxels when using the coarse, standard and fine mesh options.

For voxels, the wavelength (λ) is determined based on the maximum simulation frequency and the medium properties of the region. The specified size is applied to the voxel edge length.

Type	Fine	Standard	Coarse
Finite difference time domain (FDTD)	$\frac{\lambda}{28}$	$\frac{\lambda}{20}$	$\frac{\lambda}{14}$

A-5.2 Meshing Guidelines Regarding Element Sizes

View the meshing guidelines regarding element sizes to prevent warning and errors given by the Solver.

Table 71: Segmentation warnings and errors.

Description	Warning	Error
Ratio of the segment length to the wavelength	$l > 0.3\lambda$	$l > 0.5\lambda$
Ratio of the segment radius to the segment length	$r > 0.3l$	$r > 1.0l$
Ratio of the triangle area to the wavelength squared (MoM RWG or order 0.5 basis function)	$A > \frac{1}{30}\lambda^2$	$A > \frac{1}{10}\lambda^2$
Ratio of the triangle area to the wavelength squared (MoM order 1.5 basis function)	$A > 0.21\lambda^2$	$A > 0.52\lambda^2$
Ratio of the triangle area to the wavelength squared (MoM order 2.5 basis function)	$A > 0.62\lambda^2$	$A > 1.40\lambda^2$
Ratio of the triangle area to the wavelength squared (MoM order 3.5 basis function)	$A > 0.97\lambda^2$	$A > 2.10\lambda^2$
Ratio of the triangle area to the wavelength squared for large element PO (if near fields present)	$A > 2\lambda^2$	$A > 6\lambda^2$
Ratio of wire radius to the triangle edge length at a connection point	$r \geq 3.33l$	$r \geq 5l$
Ratio of the cuboid[113] edge length to the wavelength	$l > \frac{1}{4}\lambda$	$l > \frac{1}{2}\lambda$
Ratio of the cuboid[113] edge length to the skin depth	$l > \frac{1}{5}\delta$	$l > \frac{1}{3}\delta$

113. Cuboids only supported in EDITFEKO.

Description	Warning	Error
Ratio of the tetrahedral face area to the wavelength squared - VEP, FEM (first order)	$A > 0.033\lambda^2$	$A > 0.108\lambda^2$
Ratio of the tetrahedral face area to the wavelength squared - FEM (second order)	$A > 0.047\lambda^2$	$A > 0.433\lambda^2$
Ratio of the FEM boundary tetrahedral face area to the wavelength squared	$A > 0.033\lambda^2$	$A > 0.108\lambda^2$
Ratio of the area of the triangle on a waveguide port to the smallest modal period squared	$A > \frac{1}{30}T^2$	$A > \frac{1}{10}T^2$
Ratio of the wire radius to the voxel size	$r > 0.49$	$r > 0.5$
Voxel aspect ratio	$AR \geq 15$	$AR \geq 200$

SAR Standards

Feko makes use of a local peak SAR algorithm.

The methodology in Feko for the computation of the spatial-average SAR of a cube at a given location is based on the recommendations by CENELEC[114] and the IEEE[115][116]. Feko cannot follow the standards literally for MoM and FEM calculations since they have been written for SAR calculations with an FDTD code. As a result, a local peak SAR algorithm was developed for Feko which is similar and has the same goals as the algorithm proposed in P1528.1[117], and also takes the intentions of P1528.1, ICNIRP and the IEEE (when they set the basic restrictions) into account.

114. CENELEC,"Basic Standard for the Measurement of Specific Absorption Rate Related to Human Exposure to Electromagnetic Fields from Mobile Phones (300 MHz--3 GHz)," Tech. Rep. EN 50 361, July 2001.

115. IEEE C95.3-2002, IEEE recommended practice for measurements and computations of radio frequency electromagnetic fields with respect to human exposure to such fields, 100 kHz-300 GHz, (Revision of IEEE C95.3-1991), January, 13th 2003.

116. IEEEP1529/D0.0, "Draft Recommended Practice for Determining the Spatial-Peak Specific Absorption Rate (SAR) Associated with the Use of Wireless handsets -- Computational Techniques," IEEE Standards Coordinating Committee 34, Subcommittee 2, 2002

117. IEEEP1528.1TM/D1.0"Draft Recommended Practice for Determining the Peak Spatial-Average Specific Absorption Rate (SAR) in the Human Body from Wireless Communications Devices, 30 MHz - 6 GHz: General Requirements for using the Finite Difference Time Domain (FDTD) Method for SAR Calculations", Prepared by the Working Group 2 of the TC34/SC2 Committee, January 2007.

Solution Control

A-7

Control the execution of Feko by specifying the memory management and environment variables.

A-7.1 Dynamic Memory Management

Use advanced settings and variables for memory management in Feko.

Manual Setting of Memory Usage for Feko

Memory is managed dynamically and automatically by Feko. For specific cases, set the memory to be used by Feko.

Feko has the ability to manage memory dynamically, that is the memory required for the geometry data, matrix elements as well as other stages of the solution is determined and allocated at run time. When Feko attempts to allocate memory, in principle the operating system offers a certain address space, which could either be physically installed memory (RAM), but also virtual memory (system swap spaces swapped to the hard disk).

If Feko starts to swap using virtual memory, then the whole solution process can be slowed down quite significantly. Feko also has an out-of-core solution which uses the data on disk in a much more efficient way. (The out-of-core technique is also used, of course, if the problem requires more memory than is available in both RAM and virtual memory.)

For solutions that do not fit into the available RAM, but do fit into the RAM plus virtual memory, the amount RAM to use should be set, less some margin for the operating system.

> **Note:** The out-of-core technique only applies for the MoM method as well as a hybrid techniques where the MoM is involved but where the backward coupling from the PO, UTD or RL-GO area is switched off.

Two variables, *maxalloc* and *maxallocm* were used in older versions of Feko, and for backwards compatibility reasons they are still supported. However, their usage is strongly discouraged.

The variable *maxallocm* sets the maximum memory in MBytes that can be allocated together by all Feko processes launched as part of one parallel Feko job per host.

For example, the definition `maxallocm = 400` will allow a maximum of 400 MByte of memory to be allocated on every host. If there are two hosts then the maximum memory over all processes and hosts that can be used would be 800 MByte. If this is insufficient, the matrix will be saved to the hard disk (if an out-of-core solution is feasible) or Feko will halt with an error message. For parallel versions of Feko the memory limit will apply for each process and is determined by dividing the value of *maxallocm* by the number of parallel processes on the same host.

Another variable, *maxalloc* is also supported. It has the same function as *maxallocm* except the memory limit is specified in Bytes and not MBytes.

Note: Should both maxalloc and maxallocm be specified in the same model, the variable maxalloc will be ignored.

In newer Feko versions (for both Windows and Linux) the memory management is fully automatic and the usage of virtual memory (swap space) is avoided as far as possible. On UNIX versions an environment variable *FEKO_MAXALLOCM* may be defined during the installation and set up internally by means of Lua initialisation scripts. The environment variable FEKO_MAXALLOCM specifies the physically installed RAM less an operating system reserve for a specific machine. The advantage then is that the models are machine independent - with two different computers with different memory, no variable *maxallocm* has to be changed in the model. Therefore the memory management is either fully automatic (Windows or Linux) or defined on a per machine basis (other UNIX versions).

Variables Automatically Set Correctly

Specific variables are used by Feko to set/estimate memory blocks. These variables should generally not be set by the user since PREFEKO estimates the variables correctly.

CAUTION: Only adjust the variables below if an explicit error message for these variables occurs.

maxanr	The maximum number of sources.
maxapo	The size of the memory block that is used to save the coefficients in the physical optics approximation. For `maxapo=0` the necessary amount will be dynamically allocated.
maxarang	The maximum number of ϑ or φ angles used with the AR card (excitation by a point source with a specified radiation pattern).
maxarpat	The maximum number of radiation pattern excitations (AR card) allowed simultaneously.
maxbsobnr	To accelerate the ray path search with PO the area under consideration is divided into boxes. Information pertaining to which box contains which object must be stored. A field of size *maxbsobnr* is used in this case.
maxcolayer	The maximum number of layers on a CO card which implements thin dielectric sheets.
maxdrnv	The maximum number of triangle elements that can be connected to a segment at an attachment point.
maxfepkts	The maximum number of points considered for the near field computation with the FE card when using specified points.

maxfoges	The maximum number of areas that are described by using the Fock theory.
maxgfmsia	The maximum number of entries in the interpolation tables for the Green's function of a planar substrate.
maxhacards	The maximum number of HA cards (used internally to set up microstrip ports) that may be present in the `.fek` file.
maxkanr	The maximum number of "internal" edges (also the number of basis functions) per triangle. It may be larger than 3 if more than two plates share an edge.
maxknonr	The maximum number of nodes that may lie against a segment.
maxl4cards	The maximum number of L4 type loads.
maxlab	The maximum number of labels in a model.
maxlecards	The maximum number of LE cards (which specify a load on an edge between triangles).
maxleedges	The maximum number of edges between two surface triangles that can be loaded with a single LE card.
maxlengz	Dimension of the interpolation table used for the planar multilayer Green's functions. This variable determines the maximum number of sample points in the z direction.
maxmedia	The maximum number of different media used for the treatment of dielectric bodies in the surface equivalence principle. The surrounding free space (medium 0) is not counted (with `maxmedia=1` one dielectric body can be treated).
maxnp	The maximum number of columns and rows which a block in the matrix consists of in the Block-Gauss algorithm which solves the matrix equation. Dynamically `3*maxnp*maxnp*16` Bytes are allocated for 3 blocks in the matrix.
maxnv	The maximum number of connection points between wires and surface triangles.
maxndr	The maximum number of triangles.
maxnka	The maximum number of edges between two triangles.
maxnkapo	The maximum number of edges in the physical optics approximation.
maxnkno	The maximum number of nodes between segments.

maxnlayer	The maximum number of layers for the special Green's function of a planar substrate.
maxnqua	The maximum number of dielectric cuboids.
maxnseg	The maximum number of segments.
maxntetra	The maximum number of tetrahedral volume elements for a FEM solution.
maxnzeile	The maximum number of basis functions in the MoM area.
maxpoka	The maximum number of bordering edges to the PO area.
maxpokl	The maximum number of wedges in the PO area.
maxpolyf	The maximum number of polygonal plates that can be used to represent a body in the UTD region.
maxpolyp	The maximum number of corner points allowed for a polygonal plate.
maxpovs	The maximum number of label to label visibility specifications set by VS cards (a card with a range sets a number of entries equal to the size of the range).
maxsklayer	The maximum number of layers at an SK card.
maxtlcards	The maximum number of TL cards.
maxutdzyl	The maximum number of cylinders in the UTD region.
nmat	The memory size that may be allocated for the matrix of the system of linear equations. For `nmat=0`, the necessary amount will be allocated dynamically. The allocation is not specified in Bytes, but in terms of the number of type DOUBLE COMPLEX numbers. (These require 16 Bytes each.) For example, 400 MByte is specified by setting `nmat = 400*1024*1024/16`. The same effect can be achieved by setting the variable *maxalloc*or *maxallocm*. Therefore it is unusual to assign a value to *nmat*.
npuf	The maximum number of control cards that may occur in a loop (for example in a frequency sweep).

A-7.2 Feko Environment Overview

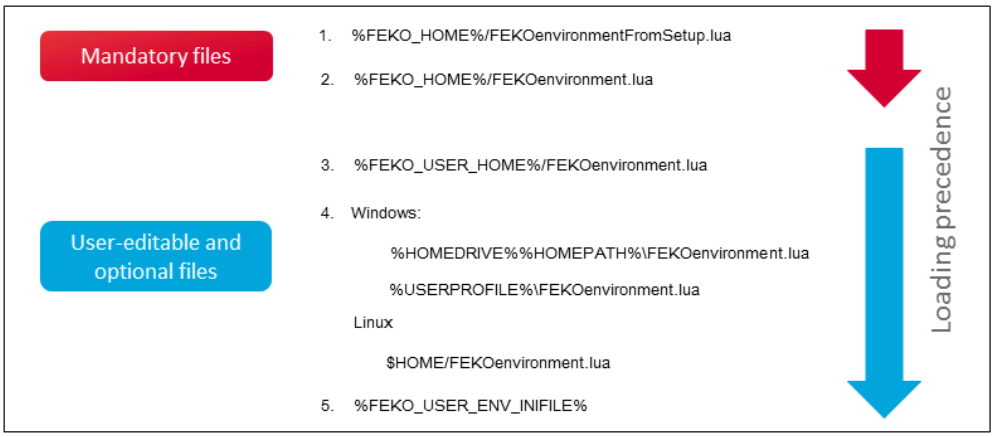
The Feko environment is setup using the Lua scripting language and internal functions. The environment setup is uniform across the different platforms.

Environment Settings Overview

The Feko environment is set up internally by means of Lua applications and internal functions

Each application is “self-aware”. It will detect and set up the environment based on its location. The default environment for the current installation will be loaded from a set of mandatory files. Any user-specific environment variables can then be added/changed in optional files loaded after the mandatory files. It allows for the user-specific environment variables to overwrite the global environment variables, rather than editing the file containing the global default environment variables

The Lua scripts are loaded in the following order:



1. *FEKO_HOME/FEKOenvironmentFromSetup.lua*
   This mandatory file is created at installation time. It contains the global default settings for the current installation. It is not advised to edit this file, unless a different setting is required than specified during installation.

2. *FEKO_HOME/FEKOenvironment.lua*
   This mandatory file is provided and managed by Feko to ensure correct functionality. This file may be updated by the update utility, so any changes to it may be lost.

3. *FEKO_USER_HOME/FEKOenvironment.lua*
   This is an optional file. It must be created by the user if and when required.

4. Windows:

   %HOMEDRIVE%%HOMEPATH%\FEKOenvironment.lua
   It must be created by the user if and when required. If it is not found, it will be silently ignored and operation continues.

   %HOMEDRIVE%%HOMEPATH%\FEKOenvironment.lua
   It must be created by the user if and when required. If it is not found, it will be silently ignored and operation continues.

   %USERPROFILE%\FEKOenvironment.lua
   It must be created by the user if and when required. If it is not found, it will be silently ignored and operation continues.

Linux:

$HOME/FEKOenvironment.lua
It must be created by the user if and when required. If it is not found, it will be silently ignored and operation continues.

5. *FEKO_USER_ENV_INIFILE*
It must be created by the user if and when required. If it is not found, it will be silently ignored and operation continues.

Functions for Environment-Related Tasks

- `getEnv(variable name, getExpanded)`

  Returns the value of the environment variable name.

Name	Description
variable name	Name of environment variable. *(String)*
getExpanded (optional)	*true*: If the value contains reference to other variables, get the expanded value. (default) *false*: Get the value as is with no extra expansion applied. *(Boolean)*
return value	Value of the environment variable (might be nil, if not set) *(String)*

- `setEnv(variable name, value, forceOverwrite)`

  Modifies the environment variable variable name to the specified value.

Name	Description
variable name	Name of environment variable. *(String)*
value	Value to be prepended. *(String)*
forceOverWrite (optional)	*parname*: Always set the value. Overwrite if variable already exists. *false:* Only set the value if variable does not exist. (default) *(Boolean)*

Name	Description
return value	-

- `prependEnv(variable name, value, delimReq)`

  Prepends (or sets, if not exists) the environment variable variable name with the specified value.

Name	Description
variable name	Name of environment variable. *(String)*
value	Value to be prepended. *(String)*
delimReq (optional)	Delimiter character/string to be used to separate values when concatenating (operating system default will be used, if not exists) *(String)*
return value	-

- `appendEnv(variable name, value, delimReq)`

  Appends (or sets, if not exists) the environment variable variable name with the specified value .

Name	Description
variable name	Name of environment variable. *(String)*
value	Value to be appended. *(String)*
delimReq (optional)	Delimiter character/string to be used to separate values when concatenating (operating system default will be used, if not exists) *(String)*
return value	-

List of Environment Variables

Control the execution of Feko with environment variables.

During the installation process, the environment variables are set correctly. The following environment variables may be set if it is required using Lua commands and internal functions:

ALTAIR_EXT_MATH_API
This variable can be used to specify if the Intel MKL libraries (default) or AMD AOCL libraries should be used when running a simulation.

Unset	MKL libraries are enabled (default).
MKL	Intel MKL libraries are enabled.
AOCL	AMD AOCL libraries are enabled.

ALTAIR_HOME
The Altair installation folder (FEKO_HOME points to the Feko folder inside ALTAIR_HOME.)

FEKO_AFFINITY_ONFLY
This variable enables the modification of the CPU mask for threads generated when converting from MPI to OpenMP. Supported values are:

Unset	Inactive (default)
0	Deactivate
1	Activate

FEKO_AFFINITY_PIN
This variable can be used to pin the processes to a core automatically or pin all processes to all cores detected. Supported values are:

Unset	Automatic pinning is enabled (default).
0	Pin all processes to all cores.
Any other value	Pinning is done automatically.

FEKO_AFFINITY_SET
This variable enables/disables affinity for Feko. Supported values are:

Unset	Affinity is disabled (default).
0	Deactivate affinity.
1	Enable affinity.
-1	Disable all affinity-related logic (including reporting and debug).

FEKO_AFFINITY_SINGLIFY
This variable controls whether the CPU mask is singlified to one resource, preventing the process from being migrated. Supported values are:

Unset	No singlify operation (default).
0	Deactivate
1	Activate

FEKO_CALCULATE_CONDITION_NUMBER

This environment variable provides control over the calculation of the MoM matrix. Allowed settings are as follows:

enabled	Calculate the MoM matrix condition number.
disabled	Do not calculate the MoM matrix condition number.
<empty> or any other value	Use the predefined Feko default - the condition number is calculated when feasible.

FEKO_CMDINFO

If this environment variable is set to the value 1, Feko writes additional data concerning the number and the value of the received command line parameters to the screen. This can be useful to trace errors in the parallel version of Feko used in connection with implementations of mpirun, mpiopt and mpprun.

FEKO_CSV_RESOURCE_REPORTING_PRESET

This variable is used for reporting preset selection for usage with a job scheduling system (for example, PBSpro). It is used to select an alternative way to integrate with the job scheduling system for resource reporting (for example CPU usage and memory). Feko will normally choose the correct default depending on the PBS version that is found on the target system. This variable is only applicable when used with the --use-job-scheduler option of RUNFEKO and Intel MPI on Linux (`FEKO_WHICH_MPI=11`).

pbs_attach	Use the pbs_attach tool of PBS to start the processes on the individual nodes. pbs_attach must be found by PATH or set FEKO_PBS_PATH to the directory where this tool is found (including a trailing slash "/").
pbs_tmrsh	Use the pbs_tmrsh tool of PBS to bootstrap the launching on the individual nodes. pbs_tmrsh must be found by PATH or set FEKO_PBS_PATH to the directory where this tool is found (including a trailing slash "/").
pbs_jmi	Use the JMI library to integrate with PBS directly (experimental).
<empty> or <not set>	Use the predefined Feko default which is based on the PBS detected at runtime.

FEKO_DATA_EXPORT_FORMAT

Use the n-th version format for the data export files (`.efe`, `.hfe`, `.ffe`, `.os`, `.ol`,`.tr`). Allowed values for n are as follows:

1 and 2	These versions was used up to Suite 6.1
3	This version was used from Suite 6.2.
4	This version was used from version 14.0.410
5	This version is used from version 2018.

"1" and "2" where "1" is the version used up to Suite 6.1. Version "2" was introduced with Suite 6.1. If not specified, the default is to use the latest supported version.

FEKO_HOME

This variable is set to the Feko installation path which contains the subdirectories such as `bin` and `license`. Note that it is not recommended to modify this environment variable.

FEKO_LICENSE_FILE

Only for legacy FEKO licenses. This variable is used to specify the location of the Feko licence file if it is not located in the default directory with the default name. The default name and location is secfeko.dat and `%FEKO_HOME%\license`.

FEKO_MACHFILE

The parallel version of Feko is started by running RUNFEKO with options -np x . When Feko is installed on a parallel computer or a computer cluster, the configuration of the cluster and the number of processes that should be run on each computer is specified during the installation. This can be overwritten for any Feko run by creating a so-called machines file and setting the environment variable FEKO_MACHFILE to point to this file.

FEKO_MACHINFO

If this parameter is set, Feko will write information about the machine precision to both the screen and the output file.

FEKO_MAXALLOCM

This environment variable is used to limit memory (in MByte) that Feko is allowed to use on the particular host. This environment variable is not needed or recommended for computers running Microsoft Windows or Linux operating systems. On others (such as UNIX systems) the variable is set at installation time. The value of this variable should usually be set equal to the physical memory minus some margin for the operating system. In a few cases a lower limit may be required, and should be set here.

FEKO_MPI_ROOT_FORCE

Use this environment variable to force Feko to use a custom MPI implementation instead of the MPI implementation included as part of the Feko installation. Set this environment variable to the path of the folder containing the MPI implementation / version.

FEKO_MPISTATISTICS

This environment variable provides additional information about the performance of the parallel version of Feko. There are three options:

1	Give a detailed report of the CPU and run times for the individual processes. It is, for example, possible to determine

	how much time each process required during the computation of the array elements.
2	Give as additional output the MFLOPS rate of each process (without network com- munication time). This is useful to determine the relative performance of nodes in a heterogeneous cluster.
4	Give information about the network performance (latency and bandwidth). This is very useful when configuring parallel clusters.

The options can be added in a binary fashion, for example setting the environment variable equal to 5 will print both the run times and network performance.

FEKO_NETWORK_DRIVE_MAPPING

This environment variable overrides the automatic mapping of shared network drives on Microsoft Windows during the MPI process startup.

1	Enable automatic shared network drive mapping on Microsoft Windows (default or not set).
0	Shared network drives are disconnected.

FEKO_PARALLEL_DEBUG

For parallel runs of Feko under UNIX, this environment variable can be set to 1 in order to see all the details and commands used in the parallel launching and machines file parsing. This is helpful for troubleshooting errors.

FEKO_RSH

When installing the parallel Feko version on a UNIX cluster, then communication between the nodes is required both at installation time (for example, checks on the remote nodes, remote copying of files and remote execution of utilities), but also when using Feko (remote launching of parallel Feko processes).

By default both the installation script and the parallel launcher will use the remote shell for this purpose (rsh for most UNIX platforms). A typical set up is then to use a `/.rhosts` file. But this is not quite secure, and you might prefer to rather use the secure shell ssh in connection with public key authentication (avoids having to type passwords all the time).

The actual remote shell executable (for example, rsh or remsh or ssh is determined during the installation procedure, and the environment variable FEKO_RSH is set to point to this executable. This can always be changed later (for example, using rsh for the installation as root, but then ssh for the users using the parallel Feko version or vice versa).

This command should be one of the following:

rsh	remote shell (might be “remsh” on some platforms)
ssh	secure shell
feko_run_local	Feko wrapper script for local runs

It is possible to set this on a user-per-user basis (see Environment Settings Overview). This environment variable is not used on Microsoft Windows systems.

FEKO_TMPDIR
This variable specifies the directory where Feko will write paging files, when using the out-of-core solution. In the past it was required that the definition ended in a backslash (Microsoft Windows) or a slash (UNIX). This is no longer required. For example, in UNIX it may be set as follows:

- `set FEKO_TMPDIR=` or by
- `export FEKO_TMPDIR`

FEKO_USER_HOME
This directory is used to write user specific initialisation files. This variable replaced FEKO_WRITE. It is provided to allow different users to save unique configurations, and for situations where the user does not have write access to the Feko directory. For Microsoft Windows systems this is normally `%APPDATA%\feko\xx.yy` and on UNIX systems it is usually set to `$HOME/.feko/xx.yy` during the installation. Here `xx.yy` represent the major and minor version numbers.

FEKO_SHARED_HOME
This directory is used to write files shared between Feko users on the same machine. For Microsoft Windows systems, this is by default set to `C:\ProgramData\altair\feko\xx.yy`, and on UNIX systems it is set to `$HOME/.feko/xx.yy` during the installation. Here `xx.yy` represent the major and minor version numbers.

FEKO_WHICH_MPI
Feko uses different MPI implementations for the different platforms and thus the different platforms require different command syntax to start Feko. RUNFEKO provides an interface that remains the same on all platforms. However, it must know which MPI implementation is used. This is done by setting the environment variable FEKO_WHICH_MPI (it is automatically set during installation) to one of the following options:

1	MPICH (Only supported on Linux)
4	SGI MPT (Deprecated)
11	Intel MPI
13	MS MPI
15	Open MPI (Only supported on Linux)

FEKO_WHICH_SPICE_EXECUTABLE
This variable specifies which SPICE solver should be used. When left unset (default),Feko uses HyperSpice. You can set this variable to point to the full path and executable name of the preferred SPICE solver.

Note: Supported SPICE engines: NGSPICE, LTSPICE and PSPICE.

FEKO_WHICH_SPICE_ENGINE
When FEKO_WHICH_SPICE_ENGINE is not set (default), Feko will attempt to auto-detect the SPICE engine that has been set using the FEKO_WHICH_SPICE_EXECUTABLE environment variable. The FEKO_WHICH_SPICE_ENGINE environment variable allows the user to specify the

SPICE engine so that FEKO does not need to perform the auto-detection. The following options are supported:

1	NGSPICE
2	PSPICE
3	LTSPICE

FEKO_WRITE_RHS

If this environment variable is set (value arbitrary), Feko writes the right side of the set of linear equations to a `.rhs` file. This is only useful for test purposes, such as when one wants to analyse this vector with another program.

Read MAT, LUD, RHS and STR Files

A-8

The `.mat` file, `.lud` file and `.rhs` file are not generated by default, but can be read externally.

The `.mat` file contains the elements of the method of moments (MoM) matrix A.

The `.lud` file is the LU decomposition of the MoM matrix A, which is used for the solution of the system of linear equations $Ax = y$.

The `.rhs` file contains the right hand side vector y representing the MoM excitation.

The `.str` file contains the solution vector x of the system of linear equations $Ax = y$. This is useful for adding further output requests (for example, adding a far field request) without rerunning the Solver from scratch.

A-8.1 Read MAT and LUD Files Using Fortran

Read the `.mat` file that contains the elements of the MoM matrix A and the `.lud` file that contains the LU decomposition of the method of moments matrix A.

The `.mat` and `.lud` files are binary files, written in the native platform coding (for example, little-endian coding on the INTEL / AMD CPUs), and have a Fortran block structure using the COMPLEX data type (in either single or double precision).

Reading can be done again from Fortran using the following code fragment:

```
CHARACTER MD5_CHECK*32
 INTEGER VERSION
 LOGICAL FILE_SINGLE_PRECISION
 INTEGER ROWS, COLUMNS
 INTEGER I, J

 OPEN (19, FILE="filename", FORM='UNFORMATTED')

 READ (19) VERSION
 READ (19) MD5_CHECK
 IF (VERSION.GE.2) THEN
   READ (19) FILE_SINGLE_PRECISION
 ELSE
   FILE_SINGLE_PRECISION = .FALSE.
 END IF
 READ (19) ROWS
 READ (19) COLUMNS

 DO J=1, COLUMNS
   IF (FILE_SINGLE_PRECISION) THEN
     READ (19) (MATRIX_S(I,J), I=1, ROWS)
   ELSE
     READ (19) (MATRIX(I,J), I=1, ROWS)
   END IF
 END DO
```

```
    CLOSE (19)
```

with these additional variables:

MATRIX
: a two dimensional array at least ROWS*COLUMNS in size to store the data in double precision (declared as DOUBLE COMPLEX).

MATRIX_S
: a two dimensional array at least ROWS*COLUMNS in size to store the data in single precision (declared as COMPLEX).

The command above,

```
        READ (19) (MATRIX_S(I,J), I=1, ROWS)
```

reads a complete column of the matrix at once.

File Structure

The structure of the `.mat` and `.lud` files are as follows:

```
    | length=4        | VERSION (4 bytes)                 | length=4        |
    | length=32       | MD5_CHECK (32 bytes)              | length=32       |
   (| length=4        | FILE_SINGLE_PRECISION (4 bytes)   | length=4        |) -- Only
present if VERSION >= 2
    | length=4        | ROWS (4 bytes)                    | length=4        |
    | length=4        | COLUMNS (4 bytes)                 | length=4        |
    | length=ROWS*es  | MATRIX(:,1) (ROW*es bytes)        | length=ROWS*es  |  -- es is
8 or 16 bytes depending on precision.
    | length=ROWS*es  | MATRIX(:,2) (ROW*es bytes)        | length=ROWS*es  |
    ...
    | length=ROWS*es  | MATRIX(:,COLUMNS) (ROW*es bytes)  | length=ROWS*es  |
```

Here each record of interest is preceded by a length field that indicates the size (in bytes) of the record.

> **Note:** The size of the length field is 4 bytes.

When reading these files using an external utility, such as one written in C or MATLAB, these length fields must also be considered. They can either be ignored or can be used to detect errors in the reading process.

A-8.2 Read the RHS File

Read the `.rhs` file containing the right-hand side vector y representing the method of moments (MoM) excitation.

Use the following code fragment to read the `.rhs` file:

```
OPEN (23,  FILE=FNAMEEXT, FORM='UNFORMATTED')
READ  (23,ERR=200) (Y(I), I=1, NSZEILE)
```

where:

NSZEILE
Number of basis functions for MoM as obtained from the `.out` file or the `.mat` file.

Note: Vector array Y is always DOUBLE COMPLEX (even if single precision is requested).

A-8.3 Read the STR File

Read the `.str` file that contains the solution vector x of the system of linear equations Ax=y.

Export the `.str` file using one of the following workflows:

- Use the PS card in EDITFEKO.
- Saving the `.str` file in CADFEKO.

The `.str` file is saved as a binary file, but can be converted into ASCII format by using the `str2ascii` utility. When using the "-r" option, a special `.str` file format is created that allows the re-import of the file into Feko. This gives you the power to visualise, for instance, characteristic modes.

Use the syntax:

```
str2ascii <filename.str> -r > ascii.str
```

This results in a file `ascii.str` with an ASCII format of the `.str` file which Feko can read again. The file contains the complex basis function coefficients in real and imaginary format.

```
   8.2676511990e-05    -5.9745463799e-05
   1.2218137786e-04    -2.4529271790e-04
  -2.7489665728e-05     1.6734208727e-04
   9.3781476258e-05    -1.1630237673e-04
   1.0987524472e-04    -1.8260254976e-04
```

Related concepts
Saving the `.str` file CADFEKOSaving the `.str` file CADFEKO
Related reference
PS Card

A-8.4 MAT2ASCII Utility

Run the `mat2ascii` utility from the command line to convert a `.mat` file to ASCII format. The `mat2ascii` utility is supported for both the 64-bit Windows and Linux platforms.

Tip: The `mat2ascii` utility only supports `.mat` files generated with a sequential run.[118]

118. If `.mat.0`, `.mat.1` ... are generated, the files are for a parallel run.

The `mat2ascii` utility is called from the command line using:

```
mat2ascii FILENAME [b]
```

FILENAME
: The name of the `.mat` file (including the file extension).

b
: [Optional] The number of the block you want to convert to ASCII. Typically this is for a run over frequency where there is a block for each frequency.

  The default is the first block, `b=1`.

An example of how to run the `mat2ascii` utility using the command line and redirecting the output to a `.txt` file:

```
mat2ascii example.mat 3 > example_fr_3.txt
```

Integration With Other Tools

Feko integrates with various products within Altair HyperWorks Products such as HyperStudy. Integration with third-party products is also supported through the powerful scripting and plug-in infrastructure.

A-9.1 HyperStudy Integration

Part of the Altair HyperWorks Products, HyperStudy is a solver neutral design tool, which enable you to explore, understand and improve your design using methods such as design of experiments (DOE), response surface modelling and optimisation.

> **Note:** HyperStudy is not installed with Feko, it can be installed by running the Altair Simulation main installer.

Further details are available in the HyperStudy documentation and Example I.3 in the Feko Example Guide.

Registering a Solver Script

Register Feko as a solver script in HyperStudy.

Integration is usually seamless, except when different installation versions of HyperStudy and Feko are used. In this case, it may be necessary to register the solver script.

1. In HyperStudy, click **Edit** > **Register Solver Script**.
2. Click **Add Solver Script** > **Feko** and click **OK** to close the dialog.
3. In the **Path** field, browse to the location of the Feko installation that should be registered.
4. Select `runfeko.exe` in the `feko\bin\` folder.
5. Click **OK** to close the dialog.

Output Responses

Define the design output responses in HyperStudy.

HyperStudy offers various methods to define the design output responses that are read from the Feko simulation results. Three of these methods include:

1. Running a POSTFEKO extraction Lua script after each Feko simulation. The Lua script writes the axis and quantity values from a visible Cartesian graph and polar graph trace to a `hst_output.hstp` file that is read by HyperStudy.

   > **Note:** It is required to have a POSTFEKO session with a matching name as the `.cfx` file in the same directory.

   - This approach is recommended if the data is available in POSTFEKO and can be plotted on a Cartesian graph or polar graph.
2. Running a Lua script after each Feko simulation. The Lua script writes the values to a `hst_output.hstp` file that is read by HyperStudy.
   - This approach is for advanced custom optimisation goals, for example, bandwidth, beamwidth and average gain.
3. Reading the values directly from the Feko `.out` file.

- This approach parses the `.out` file and reads the values at the specified offset in the file. Any reformatting of the file structure may result in incorrect values being read.

Note: While setting up the HyperStudy project, a template Lua file is created upon **Import Variables**.

The Lua file name matches the `.cfx` file, but has `_extract.lua` appended, for example, `mymodel.cfx_extract.lua`. Edit this file to add functions to calculate and write the required output response values to the `hst_output.hstp`.

Optimisation Workflows

Two optimisation workflows in HyperStudy are mentioned and the fit surface based optimisation is discussed in detail.

HyperStudy offers various workflows for performing optimisation tasks, for example:

- Conventional optimisation
- Fit surface based optimisation

Conventional Optimisation

A simulation is run at each iteration of the optimisation.

Fit Surface Based Optimisation

A design of experiments (DOE) is computed and used to generate a fit surface response, which is used for running the optimisation.

This workflow is one of the advantages that HyperStudy offers over using the optimisation engine that is available directly in Feko. It is advantageous in the following cases:

1. Multi-goal optimisation where goal weighting needs investigation.
2. Multi-goal optimisation where trade-off analysis is needed.
3. Whenever it is necessary to compare different optimisation algorithms.
4. Where a stochastic analysis is also required (can be run using the same HyperStudy fit surface that is used for the optimisation).

The workflow is made up of three parts:

1. *DOE*

   The design of experiments is the main computational effort during this workflow. Feko is run for each permutation of the design parameters and the corresponding design responses are calculated. It is recommended to use the **Modified Extensible Lattice Sequence (MELS)** method for this analysis because it is well suited to space filling and can also be extended if more samples are required.

   It is also possible to run a secondary DOE, which can be used as a testing matrix for the fit surface. In this case, it is recommended to chose a different method like **Latin HyperCube** or **Hammersley** for the validation DOE.

2. *Fit Surface*

   The fit surface response maps the relationship of the design variables to the output responses (computed in the DOE). The result is a continuous description of how the output responses vary with respect to changes in the design variables, and can be used to predict the optimum design.

   HyperStudy offers a **FAST** method which should be used to compute the fit surface. It compares the different methods and parameters and selects the best configuration for each response.

3. *Optimisation Using the Fit Surface*

   Optimisation using the fit surface typically takes a few seconds to run, which is one of the main advantages of this workflow. The computational effort is shifted from the optimisation to the DOE, which makes it possible to run multiple optimisations at a negligible additional computational cost.

   HyperStudy offers a variety of different optimisation methods to choose from. The **Global Response Search Method (GRSM)** or **Adaptive Response Search Method (ARSM)** is recommended.

Fit Surface Based Optimisation

The workflow for setting up an optimisation that uses a fit surface in HyperStudy is described.

The workflow is the following:

1. *Setup*

   Specify the `.cfx` file, **Import variables**, define the lower and upper bounds of the design variables and define the output responses.

2. *Design of experiments (DOE)*

   Use **MELS** and set the **Number of Runs**[119].

3. *Testing DOE (optional)*

   Add an additional DOE, use **Latin HyperCube** or **Hammersley** and set the **Number of Runs** to between 30 and 50.

4. *Fit surface*

   Add a fit response, set the **Input** matrix to the DOE. If a test DOE was used, add this as a second matrix with the type set to **Testing**. Use the **FAST** method, **Evaluate Tasks** and check the accuracy of the fit surface.

119. The required number of runs depends on number of design variables, but also how quickly the responses change in the design space. HyperStudy suggests a minimum number of runs, but in most cases this will need to be increased.

5. *Optimisation*

   Define the design goal **Objectives** and **Constraints**, change the **Evaluate From** field to the fit surface for all design responses. Select the optimisation method and **Evaluate Tasks**. The optimum can be found highlighted in green in the **Iteration History** table.

6. *Implement the optimum model*

   Select the row of design variable values of the optimum in the **Iteration History**, copy them, add a new DOE, set the **Number of Runs** to 1 and press **Apply**. Edit the run matrix, select the design variables row and paste the optimum values here. Run the DOE. The Feko model with the optimum design variable values is created and run.

Related concepts

Define the Output Responses

Check Accuracy of Fit Surface

Fit Surface Accuracy

Verify the accuracy of a fit surface response.

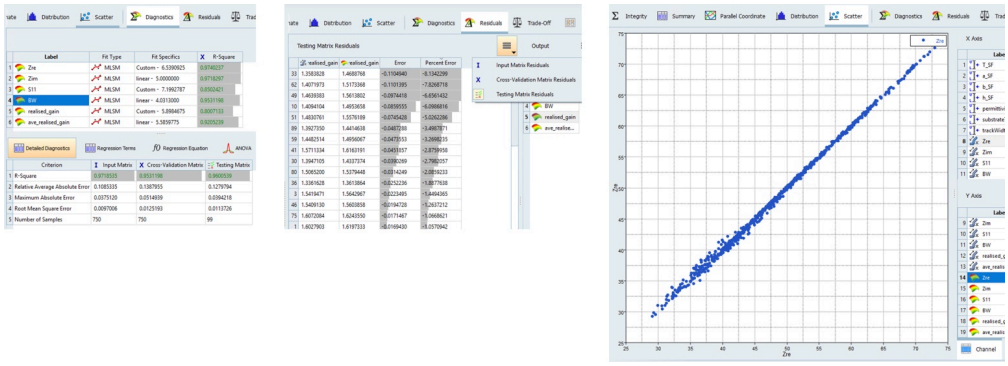


Figure 1013: Checking the fit surface accuracy.

HyperStudy has three methods to verify the accuracy of the fit response:

1. *Diagnostics*

   Diagnostics shows various statistics to help assess the accuracy and errors associated with the fit surfaces for each response. These include **R-Squared**, which for a perfect fit would be equal to 1, but values ≥ 0.9 are typically sufficient.

2. *Residuals*

   Residuals show the error and percentage error between the actual responses computed by Feko and the corresponding values of response from the fit surface.

3. *Scatter Plot*

   A scatter plot can be used to plot the simulated responses against the corresponding surface response values. Ideally, the scatter plot should show a one-to-one plot over the range of

the response, and is therefore a good visual indicator of how significant any errors may be and also in what range of the response values they may occur.

Note: If the accuracy of the fit surface is insufficient, add an additional DOE to extend the initial DOE.

In the **Specifications**, select the **Use Inclusion Matrix** check box and import the values from the initial DOE.

A-9.2 Third-Party Integration with Optenni Lab

Optenni Lab is a matching circuit generation and antenna analysis software tool. It includes automatic impedance matching tools and antenna bandwidth estimation.

A macro is provided that creates a link between CADFEKO and Optenni Lab.

On the **Home** tab, in the **Scripting** group, click the **Application macro** icon. From the drop-down list, click the **Optenni Lab: Port Matching** icon.

A Touchstone file containing the S-parameters for the system is generated by Feko and sent to Optenni Lab. A matching network can be created for each of the ports in the Touchstone file. These matching networks are then sent back to CADFEKO to be included in the model.

The Optenni Lab has three run modes:

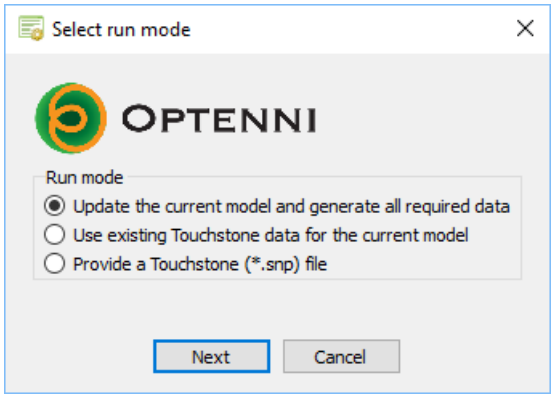


*Figure 1014: The **Select run mode** dialog.*

1. Update the model that is currently open and generate all of the required data for Optenni Lab to generate matching networks. A resulting model will be generated where all of the matched ports are reconnected through the matching networks.
2. Use a Touchstone file that has previously been generated by the currently open model and use Optenni Lab to generate matching networks. A resulting model will be generated where all of the matched ports are reconnected through the matching networks.
3. Specify an independent Touchstone file that is not associated with the open model. A matching network will be generated, but will not be connected to any ports in the model.

For multiport models, or models containing multiple configurations, user input may be required. In these cases, a dialog is generated to ask for the required input. At this point, you can provide the active ports, reference impedance, sampling and frequency range settings.

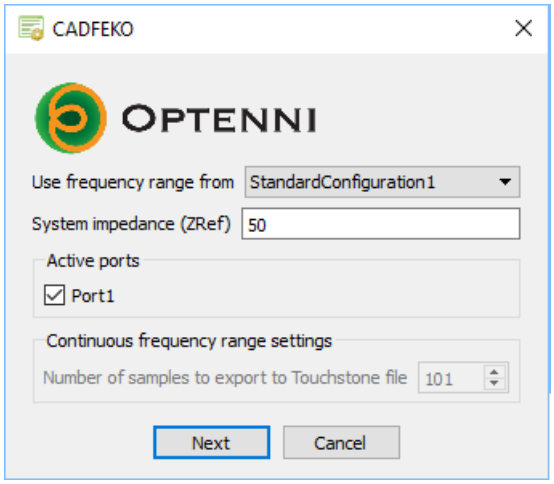


*Figure 1015: The **Optenni Lab** dialog.*

SPICE3f5

A-10

Use the correct structure, convention and syntax for a SPICE circuit definition in Feko.

A-10.1 General Structure and Conventions

A SPICE circuit is described by a set of element lines that defines the circuit topology and element values, and a set of control lines that defines the model parameters and the run controls.

> **Note:** The SPICE circuit in Feko describes the circuit only (subset of a standard SPICE circuit) therefore no run controls should be specified.

The first line in the input file must be the title and the last line must contain only the text, ".END".

Each element in the circuit is specified by an element line that contains the element name, the circuit nodes to which the element is connected and the values of the parameters that determine the electrical characteristics of the element. An example SPICE `.cir` file is as follows:

```
Matching circuit
* Note that the subcircuit name should correspond to the name
* of the general network in CADFEKO (or in the *.pre file).
.SUBCKT MatchingNetwork n1 n2
* The next two lines describe a capacitor and an inductor
c1 n1 0 2.17pF
l1 n1 n2 42.29n
.ENDS MatchingNetwork
.END
```

The sections that follow contain extracts from the Berkeley SPICE manual regarding SPICE syntax. Only a small subsection of the commands is presented and the descriptions are incomplete. Please refer to the Berkeley SPICE manuals for a more detailed description of the required syntax.

A-10.2 SPICE Syntax

Title line

The title line must be the first line in the input file and is required. Its contents will be printed verbatim as the heading for each section of output.

.END line

The ".END" line must always be the last in the input file and is required.

Comments

The asterisk in the first column indicates that the line is a comment line. Comment lines may be placed anywhere in the circuit description.

.SUBCKT line

A circuit definition begins with a ".SUBCKT" line. The subcircuit name is identified by "SUBNAM" and "N1, N2..." are the external nodes, which cannot be zero. The general form of this line as follows:

```
.SUBCKT SUBNAM N1 <N2 N3...>
```

The group of element lines, which immediately follows the ".SUBCKT" line, defines the subcircuit. The last line in a subcircuit definition is the ".ENDS" line. Control lines may not appear within a subcircuit definition. Subcircuit definitions may contain anything else, including other subcircuit definitions, device models and subcircuit calls. Any device models or subcircuit definitions included as part of a subcircuit definition are strictly local (that is such models and definitions are not known outside the subcircuit definition). In addition, any element nodes not included on the ".SUBCKT" line are strictly local, with the exception of 0 (ground) which is always global.

.ENDS line

The ".ENDS" line must be the last one for any subcircuit definition and has the following general form:

```
.ENDS <SUBNAM>
```

The subcircuit name, if included, indicates which subcircuit definition is being terminated. If omitted, all subcircuits being defined are terminated. The name is needed only when defining nested subcircuits.

.INCLUDE lines

Portions of circuit descriptions will often be reused in several input files by using the ".INCLUDE" line with the following general form:

```
.INCLUDE filename
```

Commonly used models and subcircuits can be copied as if the copied file appeared instead of the ".INCLUDE" line in the main file. There is no restriction on the file name imposed by SPICE beyond those imposed by the local operating system.

Resistors

Resistors can be included as follows:

```
RXXXXXXX N1 N2 VALUE
```

"N1" and "N2" are the two element values. "VALUE" is the resistance (in Ohm) and may be positive or negative, but not zero.

Capacitors

Capacitors can be included as follows:

```
CXXXXXXX N+ N- VALUE
```

"N+" and "N-" are the positive and negative element nodes respectively and "VALUE" is the capacitance in Farad.

Inductors

Inductors can be included as follows:

```
LXXXXXXX N+ N- VALUE
```

“N+” and “N-” are the positive and negative element nodes respectively and “VALUE” is the capacitance in Henry.

Coupled (Mutual) Inductors

Coupled inductors can be included as follows:

```
KXXXXXXX LYYYYYYY LZZZZZZZ VALUE
```

“LYYYYYYY” and “LZZZZZZZ” are the names of the two coupled inductors. The parameter “VALUE” is the coefficient of coupling, K, which must be greater than zero and less than or equal to one.

Lossless transmission lines

Lossless transmission lines can be included as follows:

```
TXXXXXXX N1 N2 N3 N4 Z0=VALUE <TD=VALUE> <F=FREQ <NL=NRMLEN>
```

“N1” and “N2” are the nodes at port one while “N3” and “N4” are the nodes at port two. “Z0” is the characteristic impedance. “N2” and “N4” are usually used as the ground connections of the transmission line.

The length of the line may be expressed in either of two forms. The transmission delay, “TD”, may be specified directly (for example as “TD=10 ns”). Alternatively, a frequency F may be given, together with “NL”, the normalised electrical length of the transmission line with respect to the wavelength in the line at the frequency “F”. If a frequency is specified but “NL” is omitted, a value of 0.25 is assumed (that is the frequency is assumed to be the quarter-wave frequency). Although both forms for expressing the line length are indicated as optional, one of the two must be specified.

List of Acronyms and Abbreviations

View the list of commonly used acronyms in Feko.

ACA	Adaptive Cross-Approximation
API	Application Programming Interface
ARSM	Adaptive Response Surface Method
ASCII	American Standard Code for Information Interchange
AVI	Audio Video Interleave
BMP	Bitmap
CAD	Computer-Aided Design
CEM	Computational Electromagnetics
CFIE	Combined Field Integral Equation
CPU	Central Processing Unit
CSV	Comma Separated Value
CUDA	Compute Unified Device Architecture
DGFM	Domain Green's Function Method
EFIE	Electric Field Integral Equation
EMC	Electromagnetic Compatibility
EPS	Encapsulated PostScript
EMF	Enhanced Metafile Format
FDTD	Finite Difference Time Domain
Feko	FEldberechnung bei Körpern mit beliebiger Oberfläche (Field computations involving bodies of arbitrary shape)
FEM	Finite Element Method
FFmpeg	Fast Forward Moving Pictures Expert Group
FSS	Frequency Selective Surface

GA	Genetic Algorithm
GIF	Graphic Interchange Format
GO	Geometrical Optics
GPU	Graphics Processing Unit
GRSM	Global Response Surface Method
HOBF	Higher Order Basis Functions
IP	Internet Protocol
JPEG	Joint Photographic Experts Group
KBL	KabelBaumListe (Harness Description List)
LE-PO	Large Element Physical Optics
MFIE	Magnetic Field Integral Equation
MFLOPS	Million FLOating Point operations per Second
MLFMM	Multilevel Fast Multipole Method
MoM	Method of Moments
MPI	Message Passing Interface
NASTRAN	NASA Structural Analysis
NURBS	Non-Uniform Rational B-Spline
NGF	Numerical Green's Function
PBC	Periodic Boundary Condition
PCB	Printed Circuit Board
PC	Personal Computer
PDF	Portable Document Format
PEC	Perfect Electric Conductor
PMC	Perfect Magnetic Conductor
PNG	Portable Network Graphics
PO	Physical Optics
PSO	Particle Swarm Optimisation

QQVGA	Quarter-Quarter Video Graphics Array
QVGA	Quarter Video Graphics Array
RSH	Remote SHell
RWG	Rao-Wilton-Glisson
SAR	Specific Absorption Rate
SEP	Surface Equivalence Principle
SPICE	Simulation Program With Integrated Circuit Emphasis
SSH	Secure Shell Host
SSPI	Security Support Provider Interface
SVGA	Super Video Graphics Array
SXGA	Super Extended Graphics Array
TIF	Tagged Image File
UTD	Uniform Theory of Diffraction
VEP	Volume Equivalence Principle
VGA	Video Graphics Array
XGA	Extended Graphics Array

Summary of Files

Feko creates and uses many different file types. It is useful to know what is stored in the various files and weather they were created by Feko and if it is safe to delete them. The files are grouped as either native files that have been created by Feko or non-native files that are supported by Feko. Non-native files are often exported by Feko even if the formats are not under the control of the Feko development team.

A-12.1 Native Files

Feko native files are files created and used by Feko. The format of these files is maintained by Feko.

List of Native Files and Descriptions

A list of native files with descriptions is given.

`._n` (for example, … `._14`, `._15` …)
: Page (temporary storage) file for the matrix in the sequential version of Feko with out-of-core solution.

`.afo`
: Continuous frequency results created by ADAPTFEKO.

`.bof`
: Binary version of the output file which is used for post processing.

`.cfm`
: CADFEKO mesh file (exported file containing CADFEKO mesh and variables to be imported by PREFEKO).

`.cfx`
: Native CADFEKO model file (contains geometry, mesh, solution settings, optimisation settings etc.)

`.cgm`
: Contains the size of the residue that results from the iterative algorithm which solves the matrix equation and the number of iterations. This file is only generated on request by a DA card.

`.csr`
: Contains the sparse near field matrix (stored in CSR format). To free memory, the sparse near field matrix is dumped to file.

`.dbg`
: When using the UTD, it is possible to request an optional output file containing a large amount of additional data (and may therefore be very large), see the UT card.

`.efe`
: File containing the electric field strengths. Contains both the position and the complex components of the electric field strength vectors. This file is only generated on request by a DA card.

`.epl`
: File containing the EM losses per element. This file is only generated on request by a DA card.

`.fek`
: Output file from PREFEKO -- serves as the input file for Feko.

`.ffe`
: File containing the far field data. This file is only generated on request by a DA card.

.fhm
: Feko HyperMesh file containing mesh data. Import or export `.fhm` in Feko using standard mesh import and export settings. Use a `.inc` file with exact file name as the `.fhm` file to include material data.

.gfe
: Interpolation table of the electric field strengths for the Green's function of a layered sphere.

.gfh
: Interpolation table of the magnetic field strengths for the Green's function of a layered sphere.

.hfe
: File containing the magnetic field strengths. Contains both the position and the complex components of the magnetic field strength vectors. This file is only generated on request by a DA card.

.inc
: Include file for PREFEKO.

.log
: Log file created by OPTFEKO.

.lud
: File in which the elements of the LU-decomposed MoM matrix are stored (only generated on request of a PS card.

.mat
: File in which the matrix elements of the linear equation system, are stored (only generated on request of a PS card.

.mdp
: A file which contains the S-parameters and additional request data for a multiport calculation. The file can be unpacked or created by The Multiport Processor .

.mdm
: File containing the list of files to be packaged to a .mdp file.

.mcc
: File containing the information for a multiport calculation. This is an .xml file which describes the content of the excitations, loads and networks connected to the multiport network from a .mdp file.

.mcr
: File containing the port scaling coefficients,voltages and currents as well as the scaled field results if requested from a multiport calculation by the multiport processor.

.opt
: Input file for the program OPTFEKO.

.ofc
: Paging files for the array elements used with sequential and parallel out-of-core solution. (To avoid the 2 GByte file size limit; or on parallel systems with a distributed file system, several files may be used. These are distinguished by adding numbers to the file name.)

.ol
File containing the surface charges and the charges in the segments. The data includes the physical position and the complex charge density. This file is only generated on request by a DA card.

.ops
File used internally by OPTFEKO to check if the existing `.fek` and `.bof` files may be reused through the --restart x option.

.os
File containing the surface currents and the currents in the segments. The data includes the physical position and the complex components of the current density vectors. This file is only generated on request by a DA card.

.out
The ASCII output file generated by the Feko solver, in which the results of all the calculations and messages can be found in a human-readable format.

.pcr
Exported ILU preconditioner for the FEM.

.pfg
POSTFEKO graph file. (The `.pfg` file is also used to store optimisation process information used for graphing in POSTFEKO after/during an optimisation run.)

.pfs
POSTFEKO session file.

.pre
Input file for PREFEKO.

.pul
File containing the cable per-unit-length parameters.

.ray
When using UTD or RL-GO an optional ray file can be requested. This file is not required when visualising rays in POSTFEKO

.rhs
File containing the right hand side vector in the system of linear equations.

.sha
File storing shadowing information for the PO.

.sol
File containing the model solution coefficients (generated on request from an MD card).

.str
File in which the coefficients of the basis functions are stored for reuse (generated on request from a PS card.

.tr
File containing the transmission and reflection coefficients.

.vis
: When multiple reflections are used with the PO formulation Feko determines which basis functions have line of sight visibility. Since this calculation may require significant run time, this information can be saved to a `.vis` file for reuse.

File Format Details

The history and format of Feko native files, such as the `.efe`, `.hfe`, `.ffe`, `.ol`, `.os` and `.epl` are described.

File Format History

A list of notable changes to the format of the `.efe`, `.hfe`, `.ffe`, `.ol`, `.os` and `.epl` is provided.

Version	Description of Changes	File Type	Release Version
2	New file formats. Used as baseline for all future changes.	• `.efe` • `.hfe` • `.ffe` • `.ol` • `.os`	6.1
3	Support for characteristic mode analysis: • `Characteristic Mode Index`	• `.efe` • `.hfe` • `.ffe` • `.ol` • `.os`	6.2
4	Support for configuration name in the header block: • `Configuration Name`	• `.efe` • `.hfe` • `.ffe` • `.ol` • `.os`	14.0.410
5	Support for exporting transmission / reflection coefficients to a `.tr` file.	• `.tr`	2018
5	Support for aperture sources in a Cartesian boundary coordinate system: • `Coordinate System`	• `.efe` • `.hfe`	2018

Version	Description of Changes	File Type	Release Version
	◦ `Cartesian Boundary`	• `.ffe`	
5	Support for incident wave polarisation angles: • `Polarisation Type` • `Ellipticity` • `Polarisation Angle`	• `.tr`	2018
5	Support for exporting characterised surface definitions to a `.tr` file.	• `.tr`	2018
6	Support for Cartesian boundary coordinate system for aperture sources: • `Coordinate System` ◦ `Cartesian` • `No. of [$$$] Samples` ◦ `U` ◦ `V`	• `.ffe`	2018.2
7	Support for exporting element power loss to a `.epl` file.	• `.epl`	2019.1
7	Support for Cartesian boundary near field requests: • `Excluded Faces Key`	• `.efe` • `.hfe`	2019.2
8	Support for antenna efficiency: • `Efficiency`	• `.ffe`	2020.1.1

General File Format

A common data structure exists for the `.efe`, `.hfe`, `.ffe`, `.ol`, `.os`, `.tr` and `.epl` files, which should be understood to use or post-process these files externally.

The general structure of the file formats consists of the following:

- **Header Block**

  The header block contains general information of the entire file. It includes information such as the file type, file format, the source file and when the file was written. Only one header block may be created and must be located at the top of the file. Header lines are denoted by two hash symbols `##`, followed by the key/value pairs allowed by each file type for the header block sections. The delimiter used between a key and value is a ":". Header lines consists of multiple lines of the form:

  ```
  ##<key>: <value>
  ```

- **Comments**

  Comments may be placed anywhere in the file. They are denoted by two asterisks `**` in the form:

  ```
  ** <comment_string>
  ```

  indicating that the rest of that line is to be ignored.

- **Solution Block**

  Any number of solution blocks (typically one per request per frequency) can follow. A solution block consists of the following:

  - **Solution Block Header**

    This section contains information that describes the data block and includes information such as the frequency, coordinate system, the request name and column headers. Solution block headers are denoted by a single hash symbol `#`, followed by the key/value pairs allowed by each file type for the solution block header sections. The format is then `#Key: Value` for each solution header block line.

  - **Data Block**

    The data block contains space delimited values. Values are given in scientific notation (for example, 1.23E-001).

The column headers are part of the solution block header and must be in the following format:

```
#no of header lines: M
#"Column 1: Line 1" "Column 2: Line 1" ... "Column N: Line 1"
#"Column 1: Line 2" "Column 2: Line 2" ... "Column N: Line 2"
...
#"Column 1: Line M" "Column 2: Line M" ... "Column N: Line M"
```

Column headers differ from other solution block header lines in that they do not have a key/value pair format. Column header lines also start with a single hash `#`, but is then followed by the column titles surrounded in quotation marks.

Units

The following units are used:

- All dimensions are measured in metres (“m”).
- All angles/phases are measured in degrees (“deg”).
- Far Fields
  - E-Fields are measured in “V”.
  - Gain / Directivity is measured in “dBi”.
  - RCS is measured in decibel square metres (“dBsm”).
- For Triangle Currents / Charges
  - Surface Current Densities (Electric) are measured in Ampere per metre (“A/m)”.
  - Surface Current Densities (Magnetic) are measured in Volt per metre (“V/m”).
  - Surface Charges are measured in Coulomb per square metre (“C/m^2”).
- For Segment Currents / Charges
  - Wire Currents are measured in Ampere (“A”).
  - Wire Charges are measured in Coulomb per metre (“C/m”).
- Near Fields
  - E-Fields are measured in “V/m”.
  - H-Fields are measured in “A/m”.

Electric / Magnetic Near Fields (.EFE / .HFE)

The data structure for the `.efe` and `.hfe` files are described.

The following fields are available in the header block:

Table 72: Fields in the header block of the `.efe` and `.hfe` files.

Key	Required	Description
File Type	Yes	Describes the type of the file. The file type can be any of the following: • Electric Near Field • Magnetic Near Field
File Format	Yes	Denotes the file syntax version (such a "4"). If not present, it defaults to version 1 (files pre-dating Feko 6.1).
Source	Optional	Denotes the base filename of the source where this data comes from.
Date	Optional	Date and time of data export in format "YYYY-MM-DD hh:mm:ss" (that is 24-hour format)

The following fields are available in the solution block header:

Table 73: Available fields in the solution block header of the `.efe` and `.hfe` files.

Key	Required	Description
Configuration Name	Optional	The configuration name, if present.
Request Name	Optional	The explicit name given to that solution request (as denoted in the `.pre` file). If none is specified, POSTFEKO uses a default name of `request_N` (where `_N` is replaced with a number for each unnamed request) during import.
Frequency	Yes	Frequency in Hz for which the following data was measured/computed.
Coordinate System	Optional	Coordinate system in which the axes are defined: • Cartesian (default) • Cylindrical • Spherical • Cylindrical (X axis) • Cylindrical (Y axis) • Conical

Key	Required	Description
		• Cartesian Boundary
Origin	Optional	Origin of the data coordinate system in form (x, y, z) (always in Cartesian coordinates; based on global origin). If no origin is provided, assume (x, y, z) = (0, 0, 0).
U-Vector	Optional	Indicates a point on the U axis relative to the Origin. If none is specified, it is assumed that the U axis coincides with the X axis.
V-Vector	Optional	Indicates a point on the V axis relative to the origin. If none is specified, it is assumed that the V axis coincides with the Y axis.
No. of [$$$] Samples	Yes	The number of samples in each axis direction. The "[$$$]" term is replaced by the following: • X/U • Y/V • Z/N • Phi • Theta • Rho • Radius • Cuboid
Result Type	Optional	For electric near fields (`.efe`), this specifies whether the output is one of the following: • Electric Field Values (default) • Magnetic Vector Potential • Gradient of Scalar Electric Potential • Electric Scalar Potential For magnetic near fields (`.hfe`), this specifies whether the output is one of the following: • Magnetic Field Values (default) • Electric Vector Potential • Gradient of Scalar Magnetic Potential • Magnetic Scalar Potential
Excluded Faces Key	Only applicable to Cartesian near field	Specify an integer that is a bit-wise interpretation of the following: • 0: Default - All faces included • 1: +N face excluded • 2: -N face excluded

Key	Required	Description
	boundary request.	• 4: -V face excluded • 8: +V face excluded • 16: -U face excluded • 32: +U face excluded
Spatial Units	Optional	Specify the units in which the position is defined.
Result Units	Optional	Specify the units in which the results are given.
No. of Header Lines	Optional	Number of header lines to read. The column header lines must follow this line. If this value is not specified, assume a value of "1".

	Yes	For header lines, the following format should be used:
		`#No. of Header Lines: M`
		`#"Column 1: Line 1" "Column 2: Line 1" ... "Column N: Line 1"`
		`#"Column 1: Line 2" "Column 2: Line 2" ... "Column N: Line 2"`
		`...`
		`#"Column 1: Line M" "Column 2: Line M" ... "Column N: Line M"`

The default column headers all have the same structure. The column headers depend on the coordinate system that was defined, which can be found in the **Coordinate System** value. For each column, the text "$$$" will be replaced with the appropriate value depending on the **Result Type**. The column headers for each coordinate system follows below.

Cartesian [vector]:

```
X Y Z Re($$$x) Im($$$x) Re($$$y) Im($$$y) Re($$$z) Im($$$z)
```

Cartesian [scalar]:

```
X Y Z Re($$$) Im($$$)
```

Cylindrical (X axis) [vector]:

```
Rho Phi X Re($$$rho) Im($$$rho) Re($$$phi) Im($$$phi) Re($$$x) Im($$$x)
```

Cylindrical (X axis) [scalar]:

```
Rho Phi X Re($$$) Im($$$)
```

Cylindrical (Y axis) [vector]:

```
Rho Phi Y Re($$$rho) Im($$$rho) Re($$$phi) Im($$$phi) Re($$$y) Im($$$y)
```

Cylindrical (Y axis) [scalar]

```
Rho Phi Y Re($$$) Im($$$)
```

Cylindrical [vector]:

```
Rho Phi Z Re($$$rho) Im($$$rho) Re($$$phi) Im($$$phi) Re($$$z) Im($$$z)
```

Cylindrical [scalar]:

```
Rho Phi Z Re($$$) Im($$$)
```

Spherical [vector]:

```
Radius Theta Phi Re($$$r) Im($$$r) Re($$$theta) Im($$$theta) Re($$$phi) Im($$$phi)
```

Spherical [scalar]:

```
Radius Theta Phi Re($$$) Im($$$)
```

Conical [vector]:

```
Rho Phi Z Re($$$rho) Im($$$rho) Re($$$phi) Im($$$phi) Re($$$z) Im($$$z)
```

Conical [scalar]:

```
Rho Phi Z Re($$$) Im($$$)
```

Note: All of the coordinate system descriptions contain a section of text, "$$$". This text is not meant to be included in the file, but rather to serve as a place holder.

The contents of the "$$$" place holder will depend on the type of data being presented and the type of file being written. For any of the above systems, replace "$$$" with any of the following:

Table 74: Fields to replace "$$$" with in the `.efe` files:

`.efe`	**Electric Near Field Files**
E	for electric field values
A	for magnetic vector potential values
grad(PHI)	for gradient of the scalar electric potential values
PHI	for scalar electric potential values

Table 75: Fields to replace "$$$" with in the `.hfe` files:

`.hfe`	**Magnetic Near Field Files**
H	for magnetic field values
F	for electric vector potential values

`.hfe`	**Magnetic Near Field Files**
grad(PSI)	for gradient of the scalar magnetic potential values
PSI	for scalar magnetic potential values

Cartesian Near Field Boundary

The sampling order is as follows for a Cartesian near field boundary request:

Xmin → Xmax → Ymin → Ymax → Zmin → Zmax

An example of a simplified file, with two samples in each dimension, and the Zmin face excluded from the request:

```
...
#Coordinate System: Cartesian Boundary
...
#Excluded Faces Key: 1
...
#   "X"   "Y"   "Z"   ...
    0     0     0     ... ** X_min
    0     1     0     ...
    0     0     1     ...
    0     1     1     ...
    1     0     0     ... ** X_max
    1     1     0     ...
    1     0     1     ...
    1     1     1     ...
    0     0     0     ... ** Y_min
    1     0     0     ...
    0     0     1     ...
    1     0     1     ...
    0     1     0     ... ** Y_max
    1     1     0     ...
    0     1     1     ...
    1     1     1     ...
    0     0     0     ... ** Z_min
    1     0     0     ...
    0     1     0     ...
    1     1     0     ...
```

Far Fields (.FFE)

The data structure for the `.ffe` files is described.

The following fields are available in the header block:

Table 76: Fields in the header block of the `.ffe` file.

Key	Required	Description
File Type	Yes	Describes the type of the file. For far fields, the value must be **Far Field**.
File Format	Yes	Denotes the file syntax version (such as "4"). If not present it defaults to version 1 (files pre-dating Feko 6.1).
Source	Optional	Denotes the base filename of the source where this data comes from.
Date	Optional	Date and time of data export in format "YYYY-MM-DD hh:mm:ss" (that is 24-hour format).

The following fields are available in the solution block header:

Table 77: Available fields in the solution block header of the `.ffe` file.

Key	Required	Description
Configuration Name	Optional	The configuration name, if present.
Request Name	Optional	The explicit name given to that solution request (as denoted in the .pre file). If none is specified, POSTFEKO will use a default name of "request_N" (where "_N" is replaced with a number for each unnamed request) during import.
Frequency	Yes	Frequency in Hz for which the following data was measured/computed.
Coordinate System	Optional	Coordinate system in which the axes are defined: • Spherical (default) • Cartesian
Origin	Optional	Origin of the data coordinate system in form "(x, y, z)" (always in Cartesian coordinates; based on global origin). If no origin is provided, assume "(x, y, z)" = "(0, 0, 0)".
U-Vector	Optional	Indicates a point on the U axis relative to the Origin. If none is specified, it is assumed that the U axis coincides with the X axis.

Key	Required	Description
V-Vector	Optional	Indicates a point on the V axis relative to the Origin. If none is specified, it is assumed that the V axis coincides with the Y axis.
No. of [$$$] Samples	Yes	The number of samples in each axis direction. The "[$$$]" term is replaced by the following, depending on the coordinate system: • Theta • Phi • U • V
Result Type	Optional	For far fields (`.ffe`), this specifies whether the output is one of the following: • Gain (default) • Directivity • RCS • Far Field Values
Incident Wave Direction	Required for RCS blocks	This is a (Theta, Phi) pair indicating the angle where an infinite plane source is originating from. This field is only required if the **Result Type** is RCS.
Characteristic Mode Index	Required for characteristic mode blocks.	The value is an integer value larger than 0, which indicates the mode index of the block that follows.
Spatial Units	Optional	Specify the units in which the position is defined.
Result Units	Optional	Specify the units in which the results are given.
Efficiency	Optional	Specify the antenna efficiency. Only applicable when the result type is gain/directivity with the field request "total fields".
No. of Header Lines	Optional	Number of header lines to read. The column header lines must follow this line. If this value is not specified, assume a value of "1".

	Yes	For header lines, the following format should be used:
		`#No. of Header Lines: M`
		`#"Column 1: Line 1" "Column 2: Line 1" ... "Column N: Line 1"`
		`#"Column 1: Line 2" "Column 2: Line 2" ... "Column N: Line 2"`
		`...`
		`#"Column 1: Line M" "Column 2: Line M" ... "Column N: Line M"`

The default column headers all have the same structure. The first two column headers depend on the coordinate system that was defined, which can be found in the **Coordinate System** value. The remaining column headers will depend on the far field type that was defined, which can be found in the **Result type** value. The column headers for each coordinate system and result type follows below.

Gain:

- Spherical coordinates:

```
Theta Phi Re(Etheta) Im(Etheta) Re(Ephi) Im(Ephi) Gain(Theta) Gain(Phi)
 Gain(Total)
```

- Cartesian coordinates:

```
U V Re(Etheta) Im(Etheta) Re(Ephi) Im(Ephi) Gain(Theta) Gain(Phi) Gain(Total)
```

Note: Gain is in dB (last three columns).

Directivity:

- Spherical coordinates:

```
Theta Phi Re(Etheta) Im(Etheta) Re(Ephi) Im(Ephi) Directivity(Theta) ...
... Directivity(Phi) Directivity(Total)
```

- Cartesian coordinates:

```
U V Re(Etheta) Im(Etheta) Re(Ephi) Im(Ephi) Directivity(Theta) ...
... Directivity(Phi) Directivity(Total)
```

Note: Directivity is in dB (last three columns).

RCS (that is either a bistatic or monostatic radar cross section problem):

- Spherical coordinates:

```
Theta Phi Re(Etheta) Im(Etheta) Re(Ephi) Im(Ephi) RCS(Theta) RCS(Phi) RCS(Total)
```

- Cartesian coordinates:

```
U V Re(Etheta) Im(Etheta) Re(Ephi) Im(Ephi) RCS(Theta) RCS(Phi) RCS(Total)
```

Note: RCS components (last three columns) are linear. They are in m^2.

Far Field Values (unknown result type, that is, the results do not pertain to an antenna problem, nor to an RCS problem):

- Spherical coordinates:

```
Theta Phi Re(Etheta) Im(Etheta) Re(Ephi) Im(Ephi)
```

- Spherical coordinates:

```
U V Re(Etheta) Im(Etheta) Re(Ephi) Im(Ephi)
```

Surface Charge Density (.OL)

The data structure for the `.ol` files are described to understand and correctly post-process these files externally.

The following fields are available in the header block:

Table 78: Fields in the Header Block of the `.ol` file.

Key	Required	Description
File Type	Yes	Describes the type of the file. For surface charge density, the value must be **Charges**.
File Format	Yes	Denotes the file syntax version (such as "4"). If not present it defaults to version 1 (files pre-dating Feko 6.1).
Source	Optional	Denotes the base filename of the source where this data comes from.
Date	Optional	Date and time of data export in format "YYYY-MM-DD hh:mm:ss" (that is 24-hour format).

The following fields are available in the solution block header:

Table 79: Available fields in the solution block header of the `.ol` file.

Key	Required	Description
Configuration Name	Optional	The configuration name, if present.
Request Name	Optional	The explicit name given to that solution request (as denoted in the .pre file). If none is specified, POSTFEKO will use a default name of "request_N" (where "_N" is replaced with a number for each unnamed request) during import.
Frequency	Yes	Frequency in Hz for which the following data was measured/computed.
No. of [$$$] Samples	Yes	The number of samples in each axis direction. The "[$$$]" term is replaced by the following: • Electric Charge Triangle • Magnetic Charge Triangle • Segment Charge
Spatial Units	Optional	Specify the units in which the position is defined.
Result Units	Optional	Specify the units in which the results are given.

Key	Required	Description
No. of Header Lines	Optional	Number of header lines to read. The column header lines must follow this line. If this value is not specified, assume a value of "1".

	Yes	For header lines, the following format should be used:
		`#No. of Header Lines: M`
		`#"Column 1: Line 1" "Column 2: Line 1" ... "Column N: Line 1"`
		`#"Column 1: Line 2" "Column 2: Line 2" ... "Column N: Line 2"`
		`...`
		`#"Column 1: Line M" "Column 2: Line M" ... "Column N: Line M"`

For the column headers of charges, the only difference between triangles and segments is an optional additional field that stores the total surface area (for triangles) or the total length (for segments).

Triangles:

```
Num X Y Z Re(Q) Im(Q) [optional] Surface Area
```

Segments:

```
Num X Y Z Re(Q) Im(Q) [optional] Length
```

Surface Current Density (.OS)

The data structure for the `.os` files is described .

The following fields are available in the header block:

Table 80: Fields in the Header Block of the `.os` file.

Key	Required	Description
File Type	Yes	Describes the type of the file. For surface charge density, the value must be **Currents**.
File Format	Yes	Denotes the file syntax version (such as "4"). If not present it defaults to version 1 (files pre-dating Feko 6.1).
Source	Optional	Denotes the base filename of the source where this data comes from.
Date	Optional	Date and time of data export in format "YYYY-MM-DD-hh:mm:ss" (that is 24-hour format).

The following fields are available in the solution block header:

Table 81: Available fields in the solution block header of the `.os` file.

Key	Required	Description
Configuration Name	Optional	The configuration name, if present.
Request Name	Optional	The explicit name given to that solution request (as denoted in the .pre file). If none is specified, POSTFEKO will use a default name of "request_N" (where "_N" is replaced with a number for each unnamed request) during import.
Frequency	Yes	Frequency in Hz for which the following data was measured/computed.
No. of [$$$] Samples	Yes	The number of samples in each axis direction. The "[$$$]" term is replaced by the following: • Electric Current Triangle • Magnetic Current Triangle • Segment Current
Spatial Units	Optional	Specify the units in which the position is defined.
Result Units	Optional	Specify the units in which the results are given.

Key	Required	Description
No. of Header Lines	Optional	Number of header lines to read. The column header lines must follow this line. If this value is not specified, assume a value of "1".

	Yes	For header lines, the following format should be used:
		`#No. of Header Lines: M`
		`#"Column 1: Line 1" "Column 2: Line 1" ... "Column N: Line 1"`
		`#"Column 1: Line 2" "Column 2: Line 2" ... "Column N: Line 2"`
		`...`
		`#"Column 1: Line M" "Column 2: Line M" ... "Column N: Line M"`

The column headers for currents are dependent on the element type.

Triangles (Electric currents):

```
Num X Y Z Re(Jx) Im(Jx) Re(Jy) Im(Jy) Re(Jz) Im(Jz) ...
... Abs(Jcorn1) Abs(Jcorn2) Abs(Jcorn3) ...
... Re(Jx_c1) Im(Jx_c1) Re(Jy_c1) Im(Jy_c1) Re(Jz_c1) Im(Jz_c1) ...
... Re(Jx_c2) Im(Jx_c2) Re(Jy_c2) Im(Jy_c2) Re(Jz_c2) Im(Jz_c2) ...
... Re(Jx_c3) Im(Jx_c3) Re(Jy_c3) Im(Jy_c3) Re(Jz_c3) Im(Jz_c3)
```

Triangles (Magnetic currents):

```
Num X Y Z Re(Mx) Im(Mx) Re(My) Im(My) Re(Mz) Im(Mz) ...
... Abs(Mcorn1) Abs(Mcorn2) Abs(Mcorn3) ...
... Re(Mx_c1) Im(Mx_c1) Re(My_c1) Im(My_c1) Re(Mz_c1) Im(Mz_c1) ...
... Re(Mx_c2) Im(Mx_c2) Re(My_c2) Im(My_c2) Re(Mz_c2) Im(Mz_c2) ...
... Re(Mx_c3) Im(Mx_c3) Re(My_c3) Im(My_c3) Re(Mz_c3) Im(Mz_c3)
```

Segments:

```
Num X Y Z Re(Ix) Im(Ix) Re(Iy) Im(Iy) Re(Iz) Im(Iz)
```

Transmission / Reflection Coefficients (.TR)

The data structure for the `.tr` files are described to understand and correctly post-process these files externally.

The following fields are available in the header block:

Table 82: Fields in the header block of the `.tr` file.

Key	Required	Description
File Type	Yes	Describes the type of the file. For a transmitted and reflected wave, the value must be **Transmission Reflection Coefficient**.
File Format	Yes	Denotes the file syntax version (such as "5"). If not present it defaults to version 1 (files pre-dating Feko 6.1).
Source	Optional	Denotes the base filename of the source where this data comes from.
Date	Optional	Date and time of data export in format "YYYY-MM-DD-hh:mm:ss" (that is 24-hour format).

The following fields are available in the solution block header:

Table 83: Available fields in the Solution Block Header of the `.tr` file.

Key	Required	Description
Configuration Name	Optional	The configuration name, if present.
Request Name	Optional	The explicit name given to that solution request (as denoted in the .pre file). If none is specified, POSTFEKO will use a default name of "request_N" (where "_N" is replaced with a number for each unnamed request) during import.
Frequency	Yes	Frequency in Hz for which the following data was measured/computed.
Origin	Optional	Origin of the data coordinate system in form "(x, y, z)" (always in Cartesian coordinates; based on global origin). If no origin is provided, assume "(x, y, z)" = "(0, 0, 0)".
U-Vector	Optional	Indicates a point on the U axis relative to the Origin. If none is specified, it is assumed that the U axis coincides with the X axis.
V-Vector	Optional	Indicates a point on the V axis relative to the Origin. If none is specified, it is assumed that the V axis coincides with the Y axis.

Key	Required	Description
Polarisation Type	Yes	Indicates the polarization type, for example, linear, elliptical, circular.
Reference Plane Position	Optional	The reference plane position for the phase reference of a transmission / reflection request. If none is provided, assume "(x, y, z)" = "(0, 0, 0)".
Lattice Vector Origin	Yes (with PBC)	Periodic boundary condition lattice vector starting point in the form "(x, y, z)" (always in Cartesian coordinates; based on global origin). The field is required for periodic boundary condition files, but may be omitted when a file is created using planar substrates.
Lattice Vector U1	Yes (with PBC)	Periodic boundary condition lattice vector U1 in the form "(x, y, z)" (always in Cartesian coordinates; based on global origin). The field is required for periodic boundary condition files, but may be omitted when a file is created using planar substrates.
Lattice Vector U2	Yes (with PBC)	Periodic boundary condition lattice vector U2 in the form "(x, y, z)" (always in Cartesian coordinates; based on global origin). The field is required for periodic boundary condition files, but may be omitted when a file is created using planar substrates.
No. of [$$$] Samples	Yes	The number of samples in each axis direction. The "[$$$]" term is replaced by the following: • Incident Theta • Incident Phi • Polarisation Angle
Spatial Units	Optional	Specify the units in which the position is defined.
Result Units	Optional	Specify the units in which the results are given.
No. of Header Lines	Optional	Number of header lines to read. The column header lines must follow this line. If this value is not specified, assume a value of "1".

	Yes	For header lines, the following format should be used:
		`#No. of Header Lines: M`
		`#"Column 1: Line 1" "Column 2: Line 1" ... "Column N: Line 1"`
		`#"Column 1: Line 2" "Column 2: Line 2" ... "Column N: Line 2"`
		`...`
		`#"Column 1: Line M" "Column 2: Line M" ... "Column N: Line M"`

The column headers for transmission / reflection coefficients when surface characterisation is being performed.

Characterised surface format:

```
Incident Theta  Incident Phi  Polarisation Angle ...
... Re(R_CrossPol) Im(R_CrossPol) Re(R_CoPol) Im(R_CoPol) ...
... Re(T_CrossPol) Im(T_CrossPol) Re(T_CoPol) Im(T_CoPol)
```

Element Power Loss (.EPL)

The data structure for the `.epl` files are described to understand and correctly post-process these files externally.

The following fields are available in the header block:

Table 84: Fields in the header block of the .epl file.

Key	Required	Description
File Type	Yes	Describes the type of the file. For a transmitted and reflected wave, the value must be **Power Loss**.
File Format	Yes	Denotes the file syntax version (such as "5"). If not present it defaults to version 1 (files pre-dating Feko 6.1).
Source	Optional	Denotes the base filename of the source where this data comes from.
Date	Optional	Date and time of data export in format "YYYY-MM-DD-hh:mm:ss" (that is 24-hour format).

The following fields are available in the solution block header:

Table 85: Available fields in the Solution Block Header of the `.epl` file.

Key	Required	Description
Configuration Name	Optional	The configuration name, if present.
Frequency	Yes	Frequency in Hz for which the following data was measured/computed.
No. of Element Loss Samples	Yes	The number of element loss samples.
Element Type	Yes	Specify the mesh element type when exporting elemental power loss: • Segment • Triangle • Tetrahedron • Unidentified
No. of Header Lines: 1	Optional	Number of header lines to read. The column header lines must follow this line. If this value is not specified, assume a value of "1".

Key	Required	Description

	Yes	For header lines, the following format should be used:
		`#No. of Header Lines: 1`
		`#"Column 1: Line 1" "Column 2: Line 1" ... "Column 7: Line 1"`
		`#"Column 1: Line 2" "Column 2: Line 2" ... "Column 7: Line 2"`
		`...`
		`#"Column 1: Line M" "Column 2: Line M" ... "Column 7: Line M"`

The column headers for element power loss (`.epl`) file:

```
Num  X  Y  Z  Loss Power  Size  Label
```

Multiport Combinations Configuration (.MCC)

The data structure for the `.mcc` file is described to understand and setup the file for a multiport calculation correctly. A template file is created when a multiport data package file is generated from a S-parameter configuration.

Table 86: Elements of a multiport combinations configuration file.

Element	Required	Description
MCC	Required	The main grouping element with the following attributes: *Version* The version of the `.mcc file`. *creationDate* The creation date of the file. *solverVersion* The version used to generate the `.mcc file`. template file.
referenceData	Required	The grouping element for the data elements that can be reused in multiple multiport combinations elements to define different combinations to process. • units • sources • loads • packages • networks
units	Required	The *units* element with the following attributes: *current* The unit used for the current values. Default is Ampere (A). *voltage* The unit used for the voltage values. Default is Volt (V). *impedance* The unit used for impedance. Default is Ohm. *capacitance* The unit used for capacitance. Default is Farad (F). *inductance* The unit used for inductance. Default is Henry (H).

Element	Required	Description
sources	Required	The grouping element for the *source* elements used to excite a multiport network.
source	Required	A *source* element used to define an excitation with the following attributes: *name* The name of the source. *type* The type of source either *constantVoltage* for a voltage source or *constantCurrent* for a current source. *magnitude* The magnitude of the source. *phase* The phase in degrees of the source. *referenceImpedance* The reference impedance of the source. *sourceImpedanceName* The name of the *load* element used as in internal source impedance.
packages	Required	The grouping element of the *package* elements which references the multiport data package `.mdp` file.
package	Required	The package element used to describe a multiport data element with the following attributes: *name* Name of the multiport data package element. *dataPackage* The path to the `.mdp` file. *touchstoneFileName* The name of the Touchstone file used as the base network for the multiport calculation. *numberOfPorts* The number of ports for the multiport network.
frequencies	Required	The grouping element for the *frequency* element used in the multiport data package with the following attributes: *type* The frequency type definition *points* or *range*.

Element	Required	Description
		minimum The minimum frequency defined in the list. *maximum* The maximum frequency defined in the list.
frequency	Required	The *frequency* element containing a single value.
quantities	Optional	The grouping element for the additional quantities to be scaled by multiport calculator in the multiport data package. The following attributes are used: *indexCharacter* The character in the *sourceFile* attribute of the *quantity* element which indicates the port index to which the quantity file maps to. *wildcardCharacter* The character in the *sourcefile* attribute of the *quantity* element that indicates the active port which was used to generate the quantity file.
quantity	Optional	The quantity element referencing either multiple field data files or a single data file in the multiport data package file with the following attributes: *type* The quantity type *farfields*. *name* The name of the quantity. *sourceFile* The name of the source file in the multiport data package. *portIndex* The port index which the quantity maps to if the wild card method is not used to identify a group of data files in the multiport data package.
loads	Optional	The grouping element for the *load* element which defines the loads attached to a multiport network.
load	Optional	The load element with the following attributes: *name* The name of the load.

Element	Required	Description
		type The type of load complex,series,parallel or Touchstone. *re* The real part of a complex load. *im* The imaginary part of the complex load. *resistor* The resistor value for a series or parallel load type. *inductor* The inductor value for the series or parallel load type. *capacitor* The capacitance value for the series or parallel load type. *filename* The path to the one port Touchstone file.
networks	Optional	The grouping element for the *network* elements.
network	Optional	The *network* element describing the properties of a Touchstone network with the following attributes: *name* The name of the network. *filename* The path to the Touchstone file. *numberOfPorts* The number of network ports.
combinations	Required	The grouping element of the *combination* elements each describing a multiport calculation.
combination	Required	The element that describes the connections to a multiport data package for a single calculation. Multiple of these elements can be defined. The following attributes are required: *name* The name of the multiport combination. *packageName* The name of the package element to be used in the multiport calculation defined in the packages group.

Element	Required	Description
combinationFrequencies	Optional	The grouping element for a custom range of frequencies to be used in a multiport calculation. If no *frequency* elements are defined inside the group all frequencies are used inside the *package* element.
combinationPortSetup	Required	The grouping element defining the network connections, excitations and loads connected to the multiport network in the data package. The group can contain a *port* and *nwInstance* elements.
port	Required	The element indicating the type of connection to a multiport network with the following attributes: *index* The port index of the network in the multiport data package. *sourceName* The name of the source element attached to the port defined in the sources group. *loadName* The name of the load attached to the port defined in the ports group. *nwInstanceName* The name of the *nwInstance* element or network that the port is connected to. *portIndex* The port index of the *nwInstance* network that the port connects to.
nwInstance	Optional	The network instance element that describes what is connected to an additional Touchstone network. The network instance also contains port elements.

Example of a Multiport Combinations Configuration File

This section shows an example of a multiport combinations configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<MCC xmlns="http://www.altair.com/altair/feko/ioxml/iomcc"
     version="1"
     creationDate="2022-09-06 08:23:51"
     solverVersion="Altair Feko - Solver (seq) Version 2022.2-21418 from 2022-09-05">
  <referenceData>
    <units current="A" voltage="V" impedance="Ohm" capacitance="F" inductance="H"/>
    <sources>
```

```
      <source name="VoltageSource1" type="constantVoltage" magnitude="1" phase="0"
referenceImpedance="0"/>
    </sources>
    <loads>
        <load name="Load1" type="complexImpedance" re="50" im="0"/>

  <load name="Load2" type="seriesRLCImpedance" capacitor="0" resistor="50" inductor="0"/
>
        <load name="Load3" type="touchstone" filename="<path to touchstone file>"/>

  <load name="Load4" type="parallelRLCImpedance" capacitor="0" resistor="50" inductor="0"/
>
    </loads>
    <networks>
      <network name="Amp1" filename="two_port_amp.s2p" numberOfPorts="2"/>
    </networks>
    <packages>
      <package name="case1_three_port_dipole_array" dataPackage="3_dipole_array.mdp"
       touchstoneFilename="3_dip_mdp_example_SP_dipole_array.s3p" numberOfPorts="3">
        <frequencies type="points" minimum="  1.00E+09" maximum="  1.00E+09">
          <frequency  value='1000000000'/>
        </frequencies>
        <quantities indexCharacter="#" wildcardCharacter="*">
          <quantity  type="farFields" name="FarField1"
                     sourceFile="FarField1 - S-parameter Port# (*).ffe"/>
        </quantities>
      </package>
    </packages>
  </referenceData>
  <combinations>
    <combination name="Example" packageName="case1_three_port_dipole_array">
      <combinationFrequencies>
      </combinationFrequencies>
       <combinationPortSetup>
        <port index="1" sourceName="VoltageSource1" loadName="Load1"/>
        <port index="2" nwInstanceName="Amp_inst1" portIndex="2"/>
        <port index="3" loadName="Load2"/>
        <nwInstance name="Amp_inst1" networkName="Amp1">
            <port index="1" sourceName="VoltageSource1"/>
        </nwInstance>
      </combinationPortSetup>
    </combination>
  </combinations>
</MCC>
```

Other Supported File Formats

The data structure for other file types are provided in order to understand and correctly use or post-process these files externally.

`.cgm` files

This file contains the number of iterations and the resulting residue from the iterative solving process of the matrix equation.

`.snp` files

The Touchstone S-parameter file name contains the number of ports in the model. The extension is, for example, `.s1p` for a 1-port, `.s2p` for a 2-port. The file contains a header (following the "#" character) that specifies the frequency unit, the parameter type, the data format and the normalising impedance for all the ports. This is followed by the data lines (which may be repeated for multiple frequencies):

1-port:	frequency	$\lvert S_{11}\rvert$ $\angle S_{11}$			
2-port:	frequency	$\lvert S_{11}\rvert$ $\angle S_{11}$	$\lvert S_{21}\rvert$ $\angle S_{21}$	$\lvert S_{12}\rvert$ $\angle S_{12}$	$\lvert S_{22}\rvert$ $\angle S_{22}$
3-port:	frequency	$\lvert S_{11}\rvert$ $\angle S_{11}$	$\lvert S_{12}\rvert$ $\angle S_{12}$	$\lvert S_{13}\rvert$ $\angle S_{13}$	
		$\lvert S_{21}\rvert$ $\angle S_{21}$	$\lvert S_{22}\rvert$ $\angle S_{22}$	$\lvert S_{23}\rvert$ $\angle S_{23}$	
		$\lvert S_{31}\rvert$ $\angle S_{31}$	$\lvert S_{32}\rvert$ $\angle S_{32}$	$\lvert S_{33}\rvert$ $\angle S_{33}$	
4-port:	frequency	$\lvert S_{11}\rvert$ $\angle S_{11}$	$\lvert S_{12}\rvert$ $\angle S_{12}$	$\lvert S_{13}\rvert$ $\angle S_{13}$	$\lvert S_{14}\rvert$ $\angle S_{14}$
		$\lvert S_{21}\rvert$ $\angle S_{21}$	$\lvert S_{22}\rvert$ $\angle S_{22}$	$\lvert S_{23}\rvert$ $\angle S_{23}$	$\lvert S_{24}\rvert$ $\angle S_{24}$
		$\lvert S_{31}\rvert$ $\angle S_{31}$	$\lvert S_{32}\rvert$ $\angle S_{32}$	$\lvert S_{33}\rvert$ $\angle S_{33}$	$\lvert S_{34}\rvert$ $\angle S_{34}$
		$\lvert S_{41}\rvert$ $\angle S_{41}$	$\lvert S_{42}\rvert$ $\angle S_{42}$	$\lvert S_{43}\rvert$ $\angle S_{43}$	$\lvert S_{44}\rvert$ $\angle S_{44}$

where $\lvert S_{11}\rvert$ is the absolute value and $\angle S_{11}$ is the phase (in degrees) of the given parameter.

> **Note:** The 2-port file is formatted on a single line and in a different order than for cases with more ports. This is consistent with the Touchstone file format specification version 1.0.

When exporting results to Touchstone format, the Solver writes out the port labels to commented lines indicated by "!".

The port labels are written out in chronological order in the following format:

```
! %Port labels%
! %PortLabel|ModeText[.m][.n][.RotationText]
```

where the port name is absent or empty, the line will only consist of the "%" symbol.

For example:

```
! %Port labels%
! %PORT1|TE.1.1.rotation(0)
! %PORT2|TE.1.1.rotation(0)
! %PORT2|TE.1.1.rotation(90)
! %AEPort
! %
! %A1Port
```

`.sph` files

This file makes use of the native SWE file format of TICRA as used in their code GRASP.

A-12.2 Non-Native Files

Various files are used and even exported by Feko that are not native to Feko, meaning that the format is not controlled by Feko. The format of these files are subject to change by third parties.

List of Non-Native Files and Descriptions

A list of non-native files with descriptions is given.

.3di
: Artwork file containing data to model IC packages and substrates.

.cdb
: ANSYS mesh file which can be imported with the IN card.

.chr
: FEST3D file containing the generalised S-parameter matrix and may be exported using the DA card.

.cir
: SPICE file which describes a circuit and may be included as a non-radiating general network by means of the NW card, CI card and SC card.

.dxf
: AutoCAD geometry file which may be imported using the IN card. (Arbitrary surfaces from meshed `.dxf` files may be imported. Lines and polyline surfaces can also be import and meshed - see IN card.)

.fim
: FEST3D file containing the waveguide modes which may be imported using the AW card.

.ffs
: CST far field scan file which may be imported by using the RA and AR cards.

.gbr
: Gerber file describing printed circuit boards.

.inp
: ABAQUS mesh file.

.isd
: Data file containing the field distribution calculated by Feko for coupling with CableMod, CRIPTE or PCBMod.

.mfxml
: MVG file containing field data which may be imported by the RA and AP cards.

.nas
: NASTRAN geometry file that may be imported using the IN card.

.neu
: Geometrical data file which is exported by the program Femap.

.nfd
: Sigrity file containing field data which may be imported by the RA and AP cards.

.rei
: An XML file for coupling of Feko with Altair PollEx. The `.rei` file containing the radiated emission data of a PCB calculated in PollEx may be imported using the AJ card (PCB source).

.rsd
: File for coupling of Feko with CableMod, CRIPTE or PCBMod. It is usually created by CableMod, CRIPTE or PCBMod, but can also be created by Feko if requested with the OS card (field calculation along lines).

.snp
: Touchstone format (v1.0) S-parameter file as created by the DA card. The n refers to the number of ports.

.sph
: Spherical wave expansion (SWE) as used by the reflector antenna code GRASP from TICRA (may be exported during a Feko solution using the DA card or used to define a spherical mode excitation in a Feko solution as part of an AS card).

.x_b
: Binary version of a Parasolid model file.

.x_t
: ASCII version of a Parasolid model file.

Common Errors and Warnings

A Feko Errors, Warnings and Notes Reference Guide is available as a reference for messages that may be encountered in Feko.

View the Feko Errors, Warnings and Notes Reference Guide at the following locations for the PDF and WebHelp:

- `Altair/2024.1/help/feko/messages/pdf/Altair_Feko_Error_Warning_Reference_Guide.pdf`
- `Altair/2024.1/help/feko/messages/html/index.htm`

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z