



User's Manual

Suite 6.1

July 2011

Contents

I The FEKO Suite

1	Introduction to the FEKO Suite	1-1
1.1	FEKO Overview	1-1
1.1.1	FEKO solution engine	1-1
	Method of Moments	1-2
	MLFMM	1-3
	Adaptive Cross Approximation (ACA)	1-4
	Uniform Theory of Diffraction	1-4
	Geometrical Optics (ray launching)	1-4
	Physical Optics	1-4
	Large Element Physical Optics	1-4
	Finite Element Method	1-5
	General non-radiating networks	1-5
	Periodic boundaries	1-5
1.1.2	FEKO Suite components	1-5
1.2	Platforms	1-7
1.3	Examples	1-7
1.4	Changes in this release	1-7
1.5	Contacting your distributor or EMSS	1-8
2	General modelling guidelines	2-1
2.1	Program flow	2-1
2.2	Modelling and meshing guidelines	2-1
2.2.1	Definitions and terms	2-1
2.2.2	Meshing guidelines regarding element sizes	2-2
2.2.3	Meshing guidelines regarding connectivity	2-4
2.3	Dielectric solids	2-6
2.4	Checking the validity of the results	2-7

II Working with CADFEKO

3	Introduction to CADFEKO	3-1
3.1	CADFEKO overview	3-1
3.1.1	CADFEKO Getting started page	3-1
3.1.2	CADFEKO window	3-1
3.1.3	The CADFEKO ribbon	3-3

3.1.4	CADFEKO files	3-4
3.1.5	Preferences	3-4
	3D View rendering and colours	3-4
	Messages and log files	3-5
3.1.6	Checking for updates to the FEKO Suite	3-5
3.1.7	Batch Meshing - CADFEKO from a command line	3-6
3.2	The 3D view	3-7
3.2.1	3D view interaction	3-7
	Rotation	3-7
	Zooming	3-7
	Panning	3-7
	Zoom to selection	3-7
	Snapping to special points	3-8
3.2.2	Manipulating items	3-8
	Selecting items	3-8
	Deleting items	3-9
	Point entry	3-10
3.2.3	Display options	3-10
	View modes	3-10
	Cutplanes	3-11
	Selective viewing	3-12
	Colouring options	3-12
	Element visibility controls	3-13
	Named points, annotations, solution entities and settings	3-13
3.2.4	View settings (View tab)	3-13
	Viewing angle and distance	3-13
	Vertical Z-lock	3-14
	View undo and redo	3-14
3.3	The model or contents tree	3-14
3.3.1	Icons displayed in the model tree and details tree	3-15
3.3.2	Medium icons in the details tree	3-20
3.4	Project tools	3-20
3.4.1	Distance between points	3-20
3.4.2	Angle between points	3-21
3.4.3	Calculator	3-21
3.4.4	Notes editor	3-21
3.4.5	Archiving models	3-22
3.5	Short-cut keys	3-22

4 Creating and meshing geometry in CADFEKO

4-1

4.1	Model geometry properties	4-1
4.1.1	Setting units	4-1
4.1.2	Geometry extents	4-1
4.2	Variables and named points	4-2
4.2.1	Variables	4-2
4.2.2	Named points	4-4
4.3	Creating geometry	4-4
4.3.1	General geometry primitive dialog behaviour	4-5
4.3.2	The workplane	4-6
	Coordinates set in a given workplane	4-7
4.3.3	Creating solid primitives	4-7
4.3.4	Creating surface primitives	4-10
4.3.5	Creating curve primitives	4-13
4.3.6	Import points	4-18
4.4	Operations on geometry	4-19
4.4.1	Boolean operations	4-20
	The Union operation	4-21
	The Intersect operation	4-21
	The Subtract operation	4-21
4.4.2	Operations on parts	4-21
	Projection and imprinting points	4-21
	Split parts	4-22
	Exploding parts	4-22
	Simplify tool (removing detail)	4-23
	Stitch sheet parts	4-24
4.4.3	Creation operations	4-24
	Spinning and sweeping parts	4-24
	Loft surfaces	4-26
4.4.4	Administrative operations	4-26
	Reversing face normals	4-26
	Assemblies	4-27
	Copy objects	4-27
	Special copy options	4-28
	Collapse - removing the creation history of a part	4-28
	Re-evaluating the geometry	4-28
4.5	Transformations	4-29
4.5.1	Transformations on geometry parts	4-29
	Rotate	4-29
	Translate	4-30

	Mirror	4-30
	Scale	4-30
	Align	4-30
4.5.2	Transformations on mesh parts	4-31
4.5.3	Multiple (transformed) copies	4-31
4.6	Cable modelling	4-31
4.6.1	Cable cross sections	4-32
4.6.2	Cable shield	4-33
4.6.3	Cable path	4-34
4.6.4	Requesting a cable harness request	4-36
	The Cable tab	4-36
	The <i>Excitations and loads</i> tab	4-37
4.7	Geometry validation	4-37
4.8	Creating meshes	4-38
4.8.1	Automatic meshing	4-39
	Wires	4-39
	Faces and edges	4-39
	Regions	4-40
4.8.2	Custom meshing	4-40
4.8.3	Advanced meshing options	4-40
4.8.4	Specifying local mesh sizes	4-41
4.8.5	Editing mesh vertices	4-42
4.8.6	Creating mesh triangles	4-43
4.8.7	Reversing normals	4-43
4.9	Mesh refinement	4-44
4.9.1	Point refinement	4-44
4.9.2	Polyline refinement	4-44
4.10	Mesh manipulation	4-45
4.10.1	Mesh element radius	4-45
4.10.2	Relabelling mesh elements	4-45
4.10.3	Merging coincident vertices	4-46
4.10.4	Merging elements (removing vertices)	4-46
4.10.5	Merging mesh parts (union for meshes)	4-47
4.10.6	Remove duplicate and collapsed elements	4-47
4.11	Mesh validation	4-48
4.11.1	Mesh information	4-48
4.11.2	Finding problems in the mesh	4-48
4.11.3	View Distorted and Oversized mesh elements	4-48
4.11.4	View free mesh edges and segment nodes	4-49

4.11.5	Find intersecting mesh triangles	4-49
5	Importing and exporting	5-1
5.1	Importing and exporting geometry	5-1
5.1.1	Importing CADFEKO models	5-1
5.1.2	Working with Parasolid	5-2
	Import of Parasolid models	5-2
	Export of Parasolid models	5-2
5.1.3	General CAD formats	5-3
	Importing general CAD	5-3
	Importing AutoCAD DXF	5-3
	Exporting general CAD formats	5-5
5.1.4	Faults on geometry	5-5
5.2	Importing meshes	5-5
5.2.1	Importing *.fek files	5-5
5.2.2	Importing general mesh formats	5-6
5.3	Exporting meshes	5-6
5.4	Printing and image export	5-6
6	Defining the solution configuration in CADFEKO	6-1
6.1	Using dielectric and lossy media	6-1
6.1.1	Defining media	6-1
6.1.2	Setting material properties on geometry	6-10
6.1.3	Setting properties on mesh elements	6-15
6.1.4	Displaying dielectric media, coatings and thin sheets	6-17
6.1.5	Displaying the model view, simulation mesh and overlay	6-17
6.2	Setting the frequency	6-17
6.3	Setting the total source power	6-19
6.4	General networks and ideal transmission lines	6-20
6.4.1	Ideal transmission lines	6-20
6.4.2	General networks	6-21
6.4.3	Connecting, exciting and loading general networks	6-22
6.5	Sources independent of the geometry	6-23
6.5.1	Plane waves	6-23
6.5.2	Electric and magnetic point dipoles	6-24
6.5.3	Point sources with a specified pattern	6-25
6.5.4	Impressed spherical mode excitation	6-27
6.5.5	Impressed aperture excitation	6-29
6.5.6	Impressed current	6-30
6.6	Setting ports, sources and loads on geometry	6-31

6.6.1	Ports	6-31
	Wire ports	6-33
	Edge ports	6-34
	Microstrip ports	6-36
	Waveguide ports	6-37
	FEM line ports	6-39
	FEM modal ports	6-41
6.6.2	Sources	6-42
	Voltage sources	6-42
	Current sources	6-43
	Waveguide excitations	6-44
	FEM modal excitations	6-45
6.6.3	Loads	6-45
6.7	Infinite planes — Planar Green's functions and ground planes	6-46
6.8	Symmetry	6-49
6.9	Periodic boundary conditions	6-50
6.10	Numerical Green's function (NGF)	6-51
6.11	Solution control and special solver settings	6-52
	6.11.1 Setting solution parameters on faces	6-56
	6.11.2 Viewing special solution parameters on faces	6-57
6.12	Requesting results	6-57
	6.12.1 Currents	6-58
	6.12.2 S-parameters	6-59
	6.12.3 Transmission/reflection coefficients	6-60
	6.12.4 Far fields	6-61
	6.12.5 Near fields	6-64
	6.12.6 SAR	6-65
	6.12.7 Cable harness	6-66
	6.12.8 Ideal receiving antenna	6-67
6.13	Error estimation	6-68
6.14	Mesh manipulation	6-69
6.15	CEM validation of the CADFEKO model	6-69
6.16	Running the FEKO solver components	6-69
	6.16.1 Component parameters	6-70
6.17	Additional solution configurations	6-74
6.18	Working with CADFEKO models in EDITFEKO	6-74
	6.18.1 Disabling the CADFEKO solution configuration	6-75
	6.18.2 Setting units	6-75
	6.18.3 Referencing elements	6-75

6.18.4	Using variables and named points in EDITFEKO	6-76
6.18.5	Setting dielectric parameters	6-76
7	Setting up optimisation in CADFEKO	7-1
7.1	Defining optimisation searches	7-1
7.1.1	Setting the optimisation method and stopping criteria	7-2
7.1.2	Defining optimisation parameters	7-3
7.1.3	Defining optimisation masks	7-5
7.1.4	Setting up optimisation Goals	7-7
7.2	Optimisation Goal types	7-11
7.2.1	The Impedance Goal	7-11
7.2.2	The Near field Goal	7-12
7.2.3	The Far field Goal	7-14
7.2.4	The S-parameter Goal	7-15
7.2.5	The SAR Goal	7-17
7.2.6	The Power Goal	7-17
7.2.7	The Receiving antenna Goal	7-17
7.2.8	The Transmission/reflection Goal	7-18
7.3	The global Goal: combining and weighting of multiple Goals	7-19
7.3.1	Goal weighting	7-20
7.3.2	Goal combination tools	7-20
7.4	Specifying special optimisation solver settings	7-20

III Working with POSTFEKO

8	Introduction to POSTFEKO	8-1
8.1	Environment	8-1
8.2	Start page	8-3
8.3	Application menu	8-4
8.4	Ribbon tabs: tools and controls	8-5
8.4.1	Default tabs	8-5
8.4.2	Context-sensitive tabs	8-6
	3D View context-sensitive tabs	8-6
	2D View context-sensitive tabs	8-6
8.5	System operations	8-7
8.5.1	Application Button and quick access toolbar	8-7
8.5.2	Save and load project sessions	8-7
8.5.3	Preferences	8-7
8.5.4	Animation	8-8

8.5.5	Import and export features	8-9
	Importing	8-9
	Exporting	8-10
8.5.6	Storing data and using stored data	8-11
8.5.7	Math scripting using Lua in POSTFEKO	8-11
9	Using POSTFEKO	9-1
9.1	Launching POSTFEKO	9-1
9.2	Typical user scenarios	9-1
9.2.1	Managing projects/models	9-1
9.2.2	Add results to a view	9-2
9.2.3	Convert graph types	9-3
9.2.4	Using 2D graphs (Cartesian, Smith and polar)	9-4
	The <i>Display</i> tab	9-4
	The <i>Trace</i> tab	9-5
	The <i>Measure</i> tab	9-6
	The <i>Format</i> tab	9-8
9.2.5	Using 3D views	9-9
	The <i>Display</i> tab	9-9
	The <i>Mesh</i> tab	9-12
	The <i>Result</i> tab	9-15
	The <i>Animate</i> tab	9-18
9.2.6	Using legends	9-19
9.2.7	Use the project browser	9-21
9.2.8	Manually specify axis properties, ranges and captions	9-21
9.2.9	Using math traces	9-23
9.2.10	Using math scripts	9-23
9.2.11	Generating reports	9-25
	Exporting and copying data, images and animations	9-25
	Quick reports	9-25
	Generating a report from a template	9-26
9.3	Errors and warnings	9-27
9.3.1	Warning icon	9-27
9.3.2	POSTFEKO workflow error and warning dialog	9-27
9.3.3	Error and warning logging dialog	9-28
9.4	Getting Help	9-28
10	Scripting guide for POSTFEKO	10-1
10.1	The interface between POSTFEKO and the scripting environment	10-1
10.2	Pulling data into the scripting environment	10-2

10.3	The structure of a dataset	10-2
10.3.1	Axes	10-4
10.3.2	Quantities	10-5
10.3.3	Metadata	10-6
10.4	Processing and modifying a dataset	10-6
10.4.1	Indexing a single element in a dataset	10-6
10.4.2	Iterating through all elements in a dataset	10-7
10.5	Creating a custom dataset	10-7
10.6	Structures for internal dataset types	10-8
10.6.1	Near field	10-9
10.6.2	Far field	10-11
	Directivity	10-12
	Gain	10-12
	Realised Gain	10-12
10.6.3	Source	10-13
10.6.4	Load	10-14
10.6.5	Network	10-15
10.6.6	S-Parameter	10-16
10.6.7	Power	10-17
10.6.8	Transmission and reflection coefficients	10-18
10.6.9	Specific Absorption Rates (SAR)	10-19

IV Working with EDITFEKO

11	Introduction to EDITFEKO	11-1
11.1	Toolbars	11-2
11.1.1	File control toolbar	11-2
11.1.2	The edit and window toolbars	11-3
11.2	Menu items in EDITFEKO	11-3
11.2.1	File menu	11-3
11.2.2	Edit menu	11-3
11.2.3	Accessing Geometry Control card definition panels	11-4
11.2.4	Run menu	11-4
11.2.5	Window menu	11-5
11.2.6	Help menu	11-5
11.3	Preferences in EDITFEKO	11-5
11.3.1	Appearance tab	11-5
11.3.2	Options tab	11-6
11.4	Inserting and editing input cards and variables	11-6

11.4.1	Generating input cards	11-6
11.4.2	Variable editor	11-8
11.5	Important keystrokes	11-9
12	PREFEKO language and concepts	12-1
12.1	PREFEKO	12-1
12.2	Structure of the *.pre input file	12-1
12.2.1	Order of the cards in the *.pre file	12-1
12.2.2	Format of cards in the *.pre file	12-1
12.3	Guidelines for creating geometry in EDITFEKO	12-3
12.3.1	Meshing guidelines regarding connectivity	12-3
12.3.2	Reducing GUI mesh sizes	12-4
12.4	Usage and concept of labels	12-4
12.5	Symbolic variables	12-6
12.6	Built-in functions	12-7
12.7	FOR/NEXT loops	12-12
12.8	IF/ELSE/ENDIF constructs	12-13
12.9	EXIT command	12-14
12.10	PRINT commands	12-14
12.11	Symbolic node names	12-15
13	Geometry cards	13-1
13.1	** card	13-3
13.2	BL card	13-4
13.3	BP card	13-6
13.4	BQ card	13-8
13.5	BT card	13-10
13.6	CB card	13-12
13.7	CL card	13-14
13.8	CN card	13-16
13.9	DD card	13-17
13.10	DK card	13-18
13.11	DP card	13-20
13.12	DZ card	13-21
13.13	EG card	13-23
13.14	EL card	13-25
13.15	FM card	13-27
13.16	FO card	13-29
13.17	FP card	13-30
13.18	HC card	13-31

13.19 HE card	13-32
13.20 HP card	13-34
13.21 HY card	13-35
13.22 IN card	13-36
13.23 IP card	13-55
13.24 KA card	13-56
13.25 KK card	13-57
13.26 KL card	13-61
13.27 KR card	13-62
13.28 KU card	13-64
13.29 LA card	13-66
13.30 MB card	13-67
13.31 ME card	13-68
13.32 NU card	13-71
13.33 PB card	13-73
13.34 PE card	13-75
13.35 PH card	13-76
13.36 PM card	13-78
13.37 PO card	13-80
13.38 PY card	13-84
13.39 QT card	13-86
13.40 QU card	13-87
13.41 RM card	13-89
13.42 SF card	13-93
13.43 SY card	13-95
13.44 TG card	13-96
13.45 TO card	13-99
13.46 TP card	13-101
13.47 UT card	13-103
13.48 UZ card	13-107
13.49 VS card	13-108
13.50 WA card	13-111
13.51 WG card	13-112
13.52 WR card	13-113
13.53 ZY card	13-114
14 Control cards	14-1
14.1 ** card	14-5
14.2 Ax Cards	14-6
14.3 A0 card	14-9

14.4 A1 card	14-12
14.5 A2 card	14-13
14.6 A3 card	14-15
14.7 A4 card	14-17
14.8 A5 card	14-19
14.9 A6 card	14-20
14.10 A7 card	14-21
14.11 AB card	14-22
14.12 AC card	14-23
14.13 AE card	14-25
14.14 AF card	14-28
14.15 AI card	14-29
14.16 AK card	14-31
14.17 AN card	14-34
14.18 AP card	14-35
14.19 AR card	14-41
14.20 AS card	14-46
14.21 AV card	14-52
14.22 AW card	14-54
14.23 BO card	14-61
14.24 CA card	14-62
14.25 CD card	14-67
14.26 CF card	14-71
14.27 CG card	14-73
14.28 CM card	14-77
14.29 CO card	14-78
14.30 CS card	14-80
14.31 DA card	14-82
14.31.1 General File Format	14-83
14.31.2 *.efe / *.hfe file: Electric / Magnetic near fields	14-84
14.31.3 *.ffe file: Far fields	14-87
14.31.4 *.ol file: Surface charge density	14-89
14.31.5 *.os file: Surface current density	14-90
14.31.6 Other supported file formats	14-92
14.32 DI card	14-93
14.33 DL card	14-97
14.34 EE card	14-99
14.35 EN card	14-100
14.36 FE card	14-101

14.37 FF card	14-106
14.38 FR card	14-109
14.39 GF card	14-112
14.40 L2 card	14-118
14.41 L4 card	14-119
14.42 LC card	14-120
14.43 LD card	14-121
14.44 LE card	14-122
14.45 LF card	14-124
14.46 LN card	14-125
14.47 LP card	14-126
14.48 LS card	14-127
14.49 LZ card	14-128
14.50 NW card	14-129
14.51 OF card	14-132
14.52 OS card	14-133
14.53 PP card	14-135
14.54 PS card	14-137
14.55 PW card	14-139
14.56 RA card	14-143
14.57 SA card	14-145
14.58 SH card	14-147
14.59 SK card	14-150
14.60 SP card	14-155
14.61 TL card	14-157
14.62 TR card	14-160
14.63 WD card	14-161

V Working with SECFEKO

15 The program SECFEKO	15-1
15.1 Introduction to SECFEKO	15-1
15.2 Displaying licence information	15-1
15.3 Managing floating licences	15-3
15.3.1 Concept of a preferred licence	15-4
15.3.2 Managing floating licence servers	15-5
15.3.3 Allow or deny usage of certain floating licences	15-5
15.4 Determining machine codes and creating licence requests	15-6
15.5 SECFEKO command line	15-6

VI Working with QUEUEFEKO

16 Introduction to QUEUEFEKO	16-1
16.1 QUEUEFEKO system overview	16-1
16.2 Creating and extracting packages	16-1
16.2.1 Package configuration files	16-2
16.2.2 Including package files	16-2
16.2.3 FEKO options	16-3
16.2.4 CLUSTER options	16-3
16.2.5 Using encryption (optional)	16-4
16.2.6 Generate package	16-5
16.2.7 Adding the package to the execution queue	16-5
16.2.8 Extract package	16-5
16.2.9 Decryption of package (if required)	16-6
16.2.10 View results	16-6
16.3 Setting preferences	16-6

VII Preprocessing and the FEKO solver

17 The preprocessor PREFEKO	17-1
17.1 Description	17-1
17.2 Running PREFEKO	17-1
18 The FEKO solution kernel	18-1
18.1 Introduction	18-1
18.2 Running the FEKO kernel	18-1
18.2.1 Running the sequential version	18-1
18.2.2 Running the parallel version	18-2
18.2.3 Running on a remote host	18-6
General settings and usage	18-6
19 Description of the output file of FEKO	19-1
19.1 Geometric data	19-1
19.2 Excitation	19-5
19.3 Currents and charges	19-8
19.4 Finite conductivity	19-10
19.5 Near fields and SAR	19-11
19.6 Far fields and receiving antenna	19-12
19.7 S-parameters	19-16
19.8 Computation time and peak memory	19-17

VIII FEKO Utilities

20 The optimiser OPTFEKO	20-1
20.1 Introduction	20-1
20.2 The optimisation method	20-1
20.2.1 Simplex (Nelder-Mead)	20-2
20.2.2 Particle swarm optimisation (PSO)	20-5
20.2.3 Genetic algorithm (GA)	20-8
20.2.4 Grid search	20-10
20.3 Sensitivity analysis	20-12
20.4 Running OPTFEKO	20-13
21 The program ADAPTFEKO	21-1
21.1 Description	21-1
21.2 Running ADAPTFEKO	21-1
21.3 The *.pre input file for adaptive sampling	21-1
22 The program TIMEFEKO	22-1
22.1 Description	22-1
22.2 The *.pre input file for TIMEFEKO	22-1
22.3 The *.tim input file	22-2
22.3.1 Defining the pulse form	22-2
22.3.2 Definition of the frequency block	22-5
22.3.3 Definitions of the normalisation	22-6
22.3.4 Definition of the excitation output	22-6
22.3.5 Definition of a time point	22-6
22.4 Running TIMEFEKO	22-7
22.5 TIMEFEKO output	22-7
23 The FEKO software updater	23-1
23.1 Introduction	23-1
23.2 GUI update utility	23-1
23.3 Command-line update utility	23-2
23.4 Proxy settings - advanced	23-3
23.5 Creating a local update repository	23-3
IX Appendix	
24 Advanced modelling and solution control	24-1
24.1 Utilisation of symmetry	24-1

24.1.1	Geometric symmetry	24-1
24.1.2	Electric symmetry	24-1
24.1.3	Magnetic symmetry	24-2
24.1.4	Example of the application of symmetry	24-2
24.1.5	Special enforcement of symmetry: Even – odd method	24-3
24.2	Dynamic memory management	24-4
24.2.1	Telling FEKO how much memory can be used	24-4
24.2.2	Variables that are automatically set correctly	24-5
24.3	Environment variables and registry keys	24-7
25	Summary of files	25-1
26	SPICE3f5 general structure, conventions and syntax	26-1
27	CADFEKO geometry faults	27-1
28	Copyright notices and acknowledgements	28-1
28.1	Copyright to Voronoi	28-1
28.2	Copyright to Berkeley Spice 3f5	28-1
28.3	Copyright to SuperLU	28-2
28.4	Copyright of libcurl	28-2
28.5	Qwt project usage	28-3
28.6	HOOPS and Parasolid	28-3
28.7	MeshSim	28-3
28.8	FFmpeg (export video files)	28-3
28.9	Microsoft	28-3
28.10	Copyright of libXML2	28-4
X	Index	
	Index	I-1

Part I

The FEKO Suite

1 Introduction to the FEKO Suite

The name FEKO is an abbreviation derived from the German phrase *FEldberechnung bei K rpern mit beliebiger Oberfl che*. (*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.

1.1 FEKO Overview

FEKO is a software Suite intended for the analysis of a wide range of electromagnetic problems. Applications include EMC analysis, antenna design, microstrip antennas and circuits, dielectric media, scattering analysis and many more. The kernel provides a comprehensive set of powerful computational methods and has been extended for the analysis of thin dielectric sheets, multiple homogeneous dielectric bodies and planar stratified media. Figure 1-1 illustrates some of the numerical analysis techniques available in FEKO and the types of problems for which they are intended.

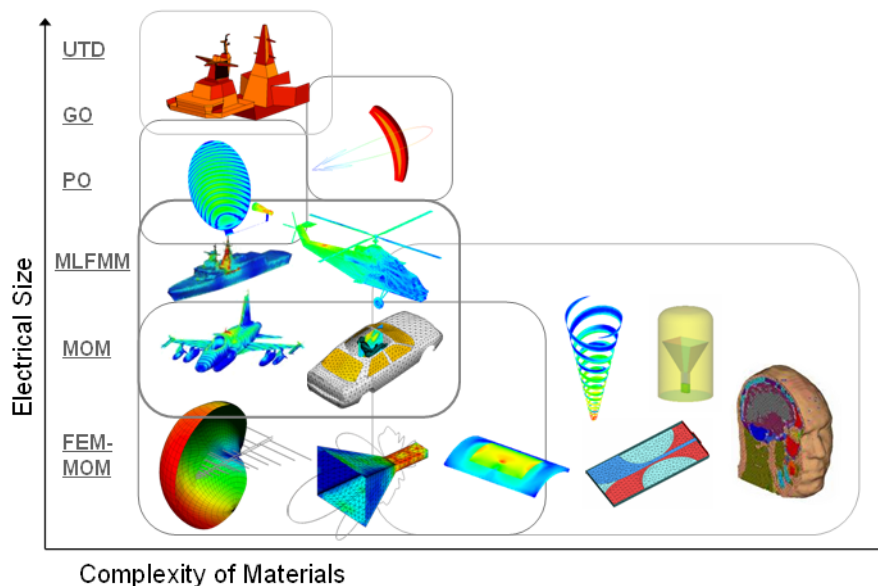


Figure 1-1: Illustration of the numerical analysis techniques in FEKO.

1.1.1 FEKO solution engine

The Method of Moments (MoM) technique forms the basis of the FEKO solver. Other techniques such as the Multilevel Fast Multipole Method (MLFMM), the Finite Element Method (FEM) Uniform Theory of Diffraction (UTD), Geometrical optics (ray launching) and Physical Optics (PO) have been implemented to allow the solving of electrically large problems and inhomogeneous dielectric bodies of arbitrary shape. Special approximations and acceleration techniques are available for problems of specific types.

FEKO provides for parallel processing usage on a range of workstations, servers and clusters. The performance for each platform, operating system and deployment method has been optimised for the delivery of accurate and timely results.

Method of Moments

The core of the program FEKO is based on the *Method of Moments* (MoM). The MoM is a full wave solution of Maxwell's integral equations in the frequency domain. An advantage of the MoM is that it is a "source method" meaning that only the structure in question is discretised, not free space as with "field methods". Boundary conditions do not have to be set and memory requirements scale proportional to the geometry in question and the required solution frequency. The following special extensions have been included in FEKO's MoM formulation to enable the modelling of magnetic and dielectric media.

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 discretized using triangles.

Volume Equivalence Principle (VEP): The VEP allows the creation of dielectric bodies from cuboids (in EDITFEKO) or tetrahedra. More basis functions are typically required than for the SEP, but neighbouring cuboids or tetrahedra may have differing electric and magnetic properties.

The volume equivalence principle is associated with a volume mesh and general usability is inhibited by the order $O(N^{2..3})$ memory and CPU-time scaling with number unknowns N . There are however special cases where the VEP is advantageous over the SEP or the FEM/MoM:

- The formulation is stable at low frequencies
- It also displays good stability and convergence properties for an iterative solution with the MLFMM
- It is well- suited to inhomogeneous, thin dielectric bodies

Note that the VEP is not supported together with dielectric solution methods (SEP, FEM, VEP with cuboids, special Green's functions) and periodic boundary conditions.

Planar Green's Functions for Multilayered Media: Multilayered dielectric media may be modelled with Green's functions, e.g. substrates for microstrip architecture. The special Green's function formulation implements 2D infinite planes with finite thickness to handle each layer of the dielectric. Conducting surfaces and wires inside the dielectric layers have to be discretized, but not the dielectric layers themselves. Metallic surfaces and wires can be arbitrarily oriented in the media and are allowed to cross multiple layers. (Calculations using Green's functions are accelerated by using interpolation tables.)

Thin Dielectric Sheets: Multiple layers of thin dielectric and anisotropic sheets can be analysed as a single layer in FEKO. Typical applications are the analysis of radome covered antennas and windscreens of automobiles.

Dielectrically Coated Wires: FEKO implements two methods for the modelling of 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 is restricted in that the loss tangent of the layer has to be identical to the loss tangent of the surrounding medium.

- Pure dielectric layers (i.e. relative permeability of the layer equals that of 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. The only restriction on the method is that the layering may not be magnetic.

Real Ground: Real ground can be modelled with the reflection coefficient approximation or the exact Sommerfeld formulation.

Windscreen: Multiple windscreen antennas on multiple glass definitions can be analysed in one model. This solution method is much faster than modelling the windscreen, antennas and layers using any other technique.

Planar Green's function aperture A planar Green's function slot or aperture interface is discretised to yield a more efficient solution. Simulations using this method are much faster than discretising the finite size ground plane that surrounds a slot or aperture, since the number of triangles can be greatly reduced.

MLFMM

The MLFMM is an alternative formulation of the technology behind the MoM and is applicable to much larger structures than the MoM, making full-wave current-based solutions of electrically large structures a possibility. This fact implies that it can be applied to most 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. 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. In the MoM framework the MLFMM is implemented through a process of aggregation, translation and disaggregation of the different levels.

The MoM treats each of N basis functions in isolation, thus resulting in an N^2 scaling of memory requirements (to store the impedance matrix) and N^3 in CPU-time (to solve the linear set of equations). It is thus clear that processing requirements for MoM solutions scale rapidly with increasing problem size. The MLFMM formulation's more efficient treatment of the same problem results in $N * \log(N)$ scaling in memory and $N * [\log(N)]^2$ in CPU time. In real applications this reduction in solution requirements can range to orders of magnitude. Significant effort has also been invested in improving the parallel MLFMM formulation to achieve exceptionally high efficiency when distributing a simulation over multiple processors.

Note that the MLFMM can now be applied in the hybrid FEM/MoM framework in FEKO to reduce computational resource requirements associated with the MoM part.

Adaptive Cross Approximation (ACA)

The ACA is a fast method similar to the MLFMM but is also applicable to low frequency problems or when using a special Green's function. It approximates the impedance matrix by constructing a sparse H-matrix (only a few selected elements are computed).

Uniform Theory of Diffraction

FEKO hybridises the current-based accurate MoM with the UTD in the truest sense of the word with the coupling between the MoM and UTD being maintained in the solution, i.e. modifying the interaction matrix and ensuring accuracy. A practical example would be a changing input impedance of a dipole treated with the MoM, in close proximity to a large structure treated with the UTD. Frequency does not influence the memory resources required for UTD treatment of a structure as only points of reflection from surfaces and diffraction from edges or corners are considered without meshing the structure. Edge and corner diffraction, double diffraction and creeping waves (cylinders) are taken into account. 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 length in the order of a wavelength or to single cylinders. The UTD is thus quite well suited to the analysis of ships at radar or electronic wave frequencies, but not well suited to the analysis of complex objects with curved surfaces, e.g. automobiles.

Geometrical Optics (ray launching)

The Geometrical optics (ray launching) is a ray-based method intended for the consideration of electrically large dielectric and perfect electrically conducting structures in applications like the analysis of lens antennas. The GO method is hybridised with the MoM in a similar fashion to the UTD. The GO method in FEKO employs ray-launching and transmission, reflection and refraction theory to model the interaction between the dielectric region and the MoM.

Physical Optics

PO is formulated for use in instances where electrically very large structures are modelled. PO is an asymptotic high frequency numerical method of the same nature as the UTD. Users will typically attempt a solution with the MoM at first and when they realise that the structure is electrically too large to solve with their available resources (platform memory, time) they will turn to one of the asymptotic high frequency techniques.

Large Element Physical Optics

Large element PO is formulated for use in instance where electrically very large smooth structures are modelled. This method is only to be used when there are no discontinuities in the incident field (e.g. if the incident field closely represents a point source). Large element 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. If the large element PO is used together with the MoM method, the MoM and PO regions are to be decoupled. Note that the same options that are available for PO are also valid for large element PO.

Finite Element Method

The FEM is applicable to the modelling of electrically large or inhomogeneous dielectric bodies, which are not efficiently solvable with FEKO's extensions to the MoM. The FEM is a volume meshing technique that employs tetrahedra to accurately mesh arbitrarily shaped volumes where the dielectric properties may vary between neighbouring tetrahedra. FEM modelling is advantageous in these instances because FEM solution matrices are sparse, where MoM matrices are densely populated, making FEM matrices significantly more scalable with an increase in frequency.

The MoM/FEM 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 technique makes use of the strengths of both the MoM and the FEM in the following ways:

- The MoM is used for the efficient modelling of open boundary radiating structures where no 3D space discretisation is required.
- The FEM is used for the efficient modelling of inhomogeneous dielectric bodies in terms of field distributions inside the volume.

General non-radiating networks

General networks (defined using network parameter matrices) as well as ideal non-radiating transmission lines may be used in FEKO simulations. These non-radiating networks may be interconnected (cascaded) and excited or loaded directly at the ports. 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.

Periodic boundaries

Large, equally-spaced periodic structures may be simulated in FEKO using an infinite periodic boundary approach. This approach may be used to provide an accurate accelerated solution for many applications like frequency selective surface analysis and large array analysis.

1.1.2 FEKO Suite components

The graphical user interface consists of the components:

- CADFEKO is used to create and mesh the geometry and to specify the solution settings and calculation requirements in a graphical environment.
- EDITFEKO is used to construct advanced models (both the geometry and solution requirements) using a high level scripting language which includes repetitive FOR loops and conditional IF-ELSE statements.
- POSTFEKO reads results from binary output files (*.bof) and can display the results on 2D graphs or in combination with the geometry in 3D views. POSTFEKO is also used to visualise optimisation results during and after optimisation, as well as the meshed geometry of the FEKO model, with excitations, field requests points etc. before the actual FEKO run.
- QUEUEFEKO facilitates the creation of packages which can be transported to remote cluster machines where the package is placed in an execution queue (such as PBS).
- FEKO_UPDATE is a command line tool that can be used to check if updates are available from a master (internet) or local repository. The FEKO GUI update tool is an interactive application that allows the user to set preferences regarding the automatic download of updates.
- SECFEKO_GUI is a visualisation of the FEKO licence manager. See SECFEKO for more details on the licence management tool.

Other components that form part of the FEKO Suite do not provide a graphical interface. These are concerned with the analysis and solution of the electromagnetic problem as defined in the GUI components, or the maintenance and administration of the Suite. Components are launched indirectly from the GUI components, but may also be launched from a command line. The solution components are fully supported by a large range of platforms.

- PREFEKO processes the model and prepares the input file (*.fek) for the FEKO solution kernel.
- FEKO is the actual solver code. The ASCII (*.out) and binary (*.bof) output files generated by FEKO contain all the solution information.
- OPTFEKO is a tool that is used for the optimisation of a FEKO model according to specific requirements. OPTFEKO calls the FEKO solver as required during optimisation.
- TIMEFEKO provides a Fourier-transform based time-domain analysis mechanism for FEKO. TIMEFEKO calls the FEKO solver as required during the solution process.
- ADAPTFEKO is used in the generation of continuous adaptively sampled results. ADAPTFEKO is called as required by the FEKO kernel when continuously sampled results are required.
- CADFEKO_BATCH is a command line tool that can be used to modify variable values in a CADFEKO model file from a command-line interface without launching the CADFEKO GUI.
- SECFEKO is the FEKO licence manager and shows all the licences in the specified licence file (secfeko.dat) for node locked licences or connects to the floating licence servers and retrieves licence information.

1.2 Platforms

The FEKO kernel components are available on PC's and a wide variety of workstations. The GUI components CADFEKO, EDITFEKO, and POSTFEKO are available on PC's running MS Windows or Linux. All pre- and post processing must thus be performed on a PC, while the actual computationally intensive field calculations can be performed on a workstation, parallel cluster or on the PC itself as required. FEKO includes a remote launching facility to make such a remote execution easy to use from within the GUI running on the PC.

1.3 Examples

First time users are advised to view the Demo Example video provided with the FEKO installation and work through the *Getting Started Manual* (additional demonstration videos are available on the FEKO website¹). The *Getting Started Manual* gives a basic introduction to the different components of the FEKO Suite and introduces the basic workflow required for effective FEKO use by helping the user complete a set of step-by-step projects. It is also recommended that new users read the introductory sections, and the chapter on modelling guidelines (see section 2) in this document carefully.

Various simple FEKO examples that show the application of a selection of features are discussed in the *ExamplesGuide.pdf* document. More examples, models and specific information may be obtained on the FEKO website.

Advanced examples based primarily on geometry creation and solution control using scripting in the EDITFEKO interface may be found in the FEKO installation directory under:

`examples/Miscellaneous/ScriptingExamples`

The examples are explained and some results are presented in `ScriptingExamples.pdf` guide that may be found in the same directory.

1.4 Changes in this release

Changes to the functionality of the code in this release with respect to the previous release of September 2010 (Suite 6.0) are indicated by adding a column in the margin. The changes are indicated in two ways:

- | Sections that have changed from those in the previous version of the manual.
- | Sections that were newly added to this version of the manual.

¹www.feko.info

1.5 Contacting your distributor or EMSS

You can find the distributor for your region at

<http://www.feko.info/contact.htm>

Alternatively, for technical questions, please send an email to

feko_support@emssusa.com	for North America
feko_support@emss.de	for Europe
feko_support@emss.co.za	for all other regions

or, for activation codes and licence queries, to

feko_license@emssusa.com	for North America
feko_license@emss.de	for Europe
feko_license@emss.co.za	for all other regions

2 General modelling guidelines

2.1 Program flow

Models are generally constructed in CADFEKO. The model information is saved to the *.cfx file and the workspace layout to the *.cfs file. Next the user runs PREFEKO — launched from the CADFEKO *Run* menu — which processes the *.cfm and *.pre files (created automatically by CADFEKO during the save operation) and generates the *.fek file. The *.fek file is the input to the solution kernel, FEKO. (When running FEKO from the CADFEKO *Run* menu, PREFEKO is executed automatically if the user has not yet done so.) The FEKO output is stored in the binary *.bof file from which the results can be viewed in POSTFEKO. The results are also stored in the *.out file.

Where an optimisation has been defined in CADFEKO, the relevant optimisation information is written to the *.opt file and optimisation-specific visualisation quantities are saved to a special POSTFEKO graph file (*.pfg). The user may launch the optimiser (OPTFEKO) from the CADFEKO *Run* menu. The optimiser will automatically launch the other FEKO Suite components as required during the optimisation process. During the optimisation process, general iteration-specific results as well as optimisation process-specific information can be viewed in POSTFEKO. Once the optimisation process has completed, the optimum model as well as a full set of results are available for viewing in POSTFEKO.

For advanced models, the user may elect to edit the *.pre file. This allows using the scripting commands and provides complete control of the solution process. (See the section working with CADFEKO models in EDITFEKO (see section 6.18) and the advanced discussion on program flow when using EDITFEKO (see section 12.1)).

If the *.pre file is manually edited, it is a good idea to validate the model in POSTFEKO before executing FEKO or OPTFEKO.

2.2 Modelling and meshing guidelines

2.2.1 Definitions and terms

Below are a number of definitions that are used frequently in this manual:

Segment : A short section of a wire (short in comparison with the wavelength).

Cuboid : A volume element used to model dielectric and magnetic solids with the volume equivalence method in the MoM. It has 90° corners similar to a cube, but does not need to have equal side lengths. (Cuboid mesh elements can only be created in EDITFEKO.)

Tetrahedron : 3D tetrahedral shaped volume element used for discretisation of dielectric regions to be solved with the FEM.

Polygon : A planar surface element with straight edge boundaries. This can be a primitive polygon (which will be subdivided into triangles) or a polygonal plate (which is not discretised and will be solved with the UTD).

Vertex : Any end point of a mesh segment or corner of a mesh element (triangle, tetrahedron, etc.)

Node : The point where two segments are joined is called a node. One basis function is assigned to each node.

Edge : In the geometry an edge is any free curve (these are also-called wires) or any boundary curve of a surface. All free curves are discretised into wire-segments during meshing.

When used in connection with triangular mesh elements on a surface, the term edge refers to the common line between two adjacent triangles. (If three triangles share two vertices, there are two edges associated with these triangles.) If the surface is a metal, then one basis function is assigned to each edge. If the surface is a dielectric to be solved with the surface current method, then two basis functions are assigned to each edge, one for the equivalent electric current density and one for the equivalent magnetic current density. A free edge belongs to only one triangle. Unless this is in a ground plane or on a polygonal plate, no current flows across this edge.

Connection point A connection point is where a segment is joined to a triangle. The end of the segment is connected to the vertex of the triangle. A basis function is assigned to each connection point.

Conducting surfaces are subdivided into triangles, and wires into segments. For dielectrics, there are a number of possibilities (see section 2.3). Using the surface current method the surface of the dielectric solid is subdivided into triangles, whereas with the volume current method, solid dielectrics are subdivided into cuboids. For the FEM, the mesh is based on tetrahedral volume elements. Thin dielectric sheets are meshed into triangles located along the middle of the sheet.

For structures that employ special solution methods (UTD, PO and GO - ray launching) special meshing rules apply. Metal faces that employ the UTD approximation are not meshed. Metal or dielectric surfaces that employ the PO approximation use a triangular meshing similar to the standard meshing.

The meshing for surfaces to be considered using Geometrical optics (ray launching) is a triangular mesh, but the required mesh size should in general be frequency-independent. For GO surfaces, the largest possible mesh size should be used that will provide an acceptable representation of the surface geometry.

2.2.2 Meshing guidelines regarding element sizes

During meshing the following general rules should be adhered to:

Segments

- The segment length l should be smaller than a tenth of the free space wavelength.
- Note also that the segment current flows only in the axial direction. Thus segments should not be too short relative to the wire radius. Ideally the segment length should be at least four times the radius.
- When modelling a conductive surface by means of a wire grid, the radius should be chosen so that the wire area in one direction is approximately the same as the area of the original surface. This leads to

$$r \approx \frac{l}{2\pi}, \quad (2-1)$$

where r is the radius and l is the segment length which should be about a tenth of a wavelength.

Surface mesh elements

- For MoM and PO mesh elements, the area A of each triangular element should be smaller than $\frac{\lambda^2}{70}$. For triangles that are approximately equilateral this means the side length s should be shorter than approximately $\frac{\lambda}{(5...6)}$. Depending on the geometry and the required accuracy, more or less triangles may be needed. If the memory constraints allow it, an edge length of $\frac{\lambda}{(8...10)}$ is preferred.
- For large element PO mesh elements, the area A of each triangular element should be smaller than $2\lambda^2$ when near field requests are present. The allowed edge length for when near field requests are present should be 2λ . If only far field requests are present, the triangles only need to represent the geometry accurately, regardless of frequency or wavelength.
- For dielectric surfaces to be solved using the GO (ray launching) method, the maximum triangular mesh element size should be chosen such that the geometry of the surface is well represented. For this method, the mesh size should be chosen independent of the solution frequency, and should be purely a function of the accuracy of the geometrical representation.
- The edge length of dielectric cuboids has to be small in comparison with the wavelength *in the dielectric* as well as the skin depth

$$\delta = \sqrt{\frac{2}{\omega\mu\sigma}}. \quad (2-2)$$

Due to the staircase approximation used in representing the model geometry, a mesh size of at least the minimum between a tenth of the wavelength and a tenth of the skin depth is recommended.

Volume mesh elements

When meshing the FEM region into tetrahedral volume elements, the element size (edge length of the tetrahedra) should be about a fifth of the wavelength *inside of the dielectric medium in question*. For the elements right on the FEM/MoM interface, a finer element size of about a tenth of the medium wavelength is recommended. The reason that a coarser mesh can be employed inside the medium is that higher order basis functions are employed for the FEM solution.

Sometimes the overall memory requirement for a solution may be reduced by adding a small air region or buffer around the actual dielectric object. This region is also meshed into tetrahedral elements (i.e. the buffer region is also solved with the FEM). Since the wavelength in air is larger than in the dielectric, larger tetrahedral elements can then be used in the buffer regions and at the FEM/MoM interface. This reduces the memory requirement for the FEM/MoM coupling arrays. This memory reduction is typically much higher than the additional memory required to solve the additional tetrahedral elements added in the air buffer zone.

In some cases accurate modelling of the geometry requires significantly finer mesh elements than specified by the guidelines above. (For low frequencies in particular, the segmentation rule of a tenth of a wavelength is often much too coarse to yield a reasonable representation of the geometry.) One example of where finer discretisation may be required is where a wire runs parallel to a conducting plate. If the wire is closer than a tenth of a wavelength to the plate, the size of the triangles in the direction orthogonal to the wire should be similar to the distance from the wire to the plate in order to give an accurate representation of the surface charge distribution. Another case where finer discretisation may be required is on waveguide ports, where the mesh size must be small enough to capture the field distribution of the highest mode which is included in the modal expansion of the port. More information may be found in the discussion of waveguide ports (see section 14.22).

If the segmentation rules are not adhered to, the errors and warnings listed in Table 2-1 will be reported by the FEKO kernel.

Table 2-1: Segmentation warnings and errors.

<i>Description</i>	<i>Warning</i>	<i>Error</i>
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	$A > \frac{1}{30}\lambda^2$	$A > \frac{1}{10}\lambda^2$
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 edge length to the wavelength	$l > \frac{1}{4}\lambda$	$l > \frac{1}{2}\lambda$
Ratio of the cuboid edge length to the skin depth	$l > \frac{1}{5}\delta$	$l > \frac{1}{3}\delta$
Ratio of the tetrahedral face area to the wavelength squared (inner mesh elements)	$A > 0.047\lambda^2$	$A > 0.433\lambda^2$
Ratio of the tetrahedral face area to the wavelength squared (boundary surface mesh elements)	$A > 0.033\lambda^2$	$A > 0.108\lambda^2$
Ratio of the area of the triangle on a waveguide port to the smallest modal period squared	$A > \frac{1}{30}T^2$	$A > \frac{1}{10}T^2$

2.2.3 Meshing guidelines regarding connectivity

The MoM

FEKO approximates the current in terms of basis functions associated with edges, nodes and connection points as defined above (see section 2.2.1). To ensure electrical connectivity, triangles must therefore share an edge. Similarly, segments must connect to other segments at nodes or to mesh triangles at vertices. Some examples are shown in Figure 2-1.

The UTD and GO

For the special case, where the UTD method is applied to unmeshed polygon plates, there is no defined connectivity between plates that share an edge.

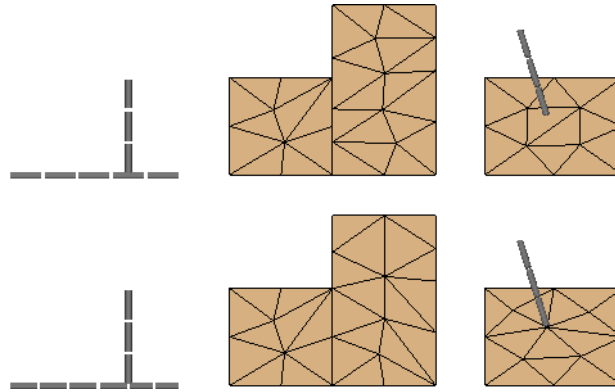


Figure 2-1: Example of mesh connectivity: unconnected at the top, connected at the bottom

For GO, the connectivity should generally be maintained, though this is not required. When meshing the faces of a GO region, it is good practice to ensure that the meshes on faces that touch do align to achieve a good geometric representation.

The FEM

When meshing dielectric solids into tetrahedral elements for the FEM, the faces of adjacent tetrahedra must match. In addition, when modelling conducting surfaces in or on the FEM region, the metallic triangles must match the faces of the tetrahedral volume elements. See Figure 2-2.

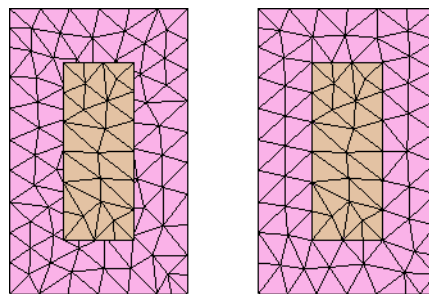


Figure 2-2: Example of FEM element connectivity: invalid left, correct right

General

CADFEKO generally enforces meshing rules regarding connectivity for each part. Therefore, connected items should be unioned (see section 4.4.1) together before meshing them. It is, however, still important that the model is of decent quality. If, for example, a wire is attached to a surface, but due to numerical error it is more than the model tolerance (see section 4.1.2) away from the actual surface CADFEKO will not create a vertex in the surface mesh at the attachment point. (It is possible to union two objects that do not touch each other at all.) The wire will then not be considered electrically connected to the surface during the solution phase.

The FEKO kernel has several checks built in and will give errors if, for example, the mesh contains overlapping triangles. CADFEKO also has several checks which the user can perform before trying to solve the model (see section 4.7).

2.3 Dielectric solids

There are numerous ways to model dielectric objects in FEKO, three of which apply to arbitrarily shaped bodies:

- **Surface equivalence principle:**
Method of Moments (MoM) generally uses the surface equivalence principle for modelling of dielectric bodies. In this method, interfaces between different homogeneous regions are subdivided into a surface mesh using triangular elements. Basis functions are applied to these elements for the equivalent electric and equivalent magnetic surface currents. Boundary conditions result through the use of equivalent sources.

For models constructed in CADFEKO this method is used by default on all dielectric regions (see section 6.1.2) that are not explicitly meshed into tetrahedral volume elements (see section 4.8).

When working with models in EDITFEKO, medium regions are defined with the ME card (see section 13.31). This uses the normal vector of the triangles to distinguish the respective dielectric media on either side of the mesh elements. It is generally advisable to check the respective media regions in POSTFEKO before running the FEKO solver.

- **Volume equivalence principle:**
MoM can also be applied with the volume equivalence principle. Here the volume is subdivided into cuboidal elements in EDITFEKO and tetrahedral elements in CADFEKO. In principle, each element can be assigned a different material property. Inside the element the polarisation current is unknown. Normally a volume would have many more unknowns than a surface mesh, such that this method would require more memory. However, this technique is very suitable for thin sheets and is also very stable for low frequencies. The formulation is very stable, and thus when using MLFMM the number of iterations is typically small. In EDITFEKO is accessed using the DK (see section 13.10), DZ (see section 13.12) or QU (see section 13.40).
- **Finite element method:**
As an alternative to the MoM, the Finite Element Method (FEM) is also available in FEKO. It requires that 3D volumes are discretised into tetrahedral elements. FEM matrices are sparse, as opposed to the MoM, and the memory requirement for a FEM volume mesh is much less than a MoM volume mesh of the same model. This method is automatically used for regions in the model that contain tetrahedral elements.

In CADFEKO models, the medium properties are specified when defining the dielectric medium (see section 6.1.1). In EDITFEKO, medium properties are specified with the DI card (see section 14.32).

For the surface equivalence principle, it is possible to define metallic triangles on the surfaces and triangles and segments within the dielectric regions. With the volume equivalence principle, there must be a small space between the cuboids and any conducting triangles on the surface of the dielectric. Metallic triangles can also be located inside FEM regions, but they have to coincide with tetrahedral surfaces.

Two additional methods that are available for dielectric bodies are Physical optics and Geometrical optics (ray launching). These are approximate methods with special limitations and application methods.

The model geometry (e.g. metallic wires and surfaces) does not necessarily have to be embedded in free space. In CADFEKO the properties of the free space (see section 6.1) medium may be changed. In EDITFEKO the EG, DI and GF cards can be used to specify the material parameters of the surrounding medium.

Special dielectrics and infinite planes

Apart from the general formulations applicable to dielectrics, there are a number of special methods to account for dielectric sheets and coatings as well as special dielectric bodies in FEKO:

- Thin dielectric sheets:
The volume equivalence principle is applied and the resulting equivalent currents approximated by a surface current.
- Dielectric coatings:
Metallic wires or triangular surface patches can have a thin dielectric coating.
- Dielectric half-space e.g. ground surface:
In this case the reflection coefficient method is used. This is activated by using infinite planes (see section 6.7) under *Solution* in CADFEKO or the BO card (see section 14.23) in EDITFEKO.
- Spheres consisting of one or more dielectric layers:
A special Green's function can be activated by using the GF card (see section 14.39) in EDITFEKO.
- Planar multilayer substrate:
A multilayer planar substrate (with or without a perfectly conducting ground planes at the top and bottom) is added to the model using infinite planes (see section 6.7) under *Solution* in CADFEKO or the GF card (see section 14.39) in EDITFEKO.
- Windscreen:
The active windscreen antenna elements can be activated by using the WA card (see section 13.50) in EDITFEKO. The dielectric windscreen reference plane is defined by the WR card (see section 13.52) in EDITFEKO. Dielectric properties of the glass layers can be defined by the WD card (see section 14.63).

In CADFEKO, thin dielectric sheets and coatings are applied to faces (see section 6.1.2). In EDITFEKO, thin dielectric sheets are defined by applying the SK card (see section 14.59) and coatings by applying the CO card (see section 14.29) to triangle labels.

2.4 Checking the validity of the results

Once a calculation has been completed with FEKO, the results have to be checked or confirmed.

There are a number of ways of doing this:

- Comparison with exact results, if these are available.
- Comparison with results that have been published in the literature.

- Comparison with results generated using another computational method.
- Comparison with measured results.
- Plausibility, e.g. negative real input impedances do not exist.

If these possibilities are not available, then the following process may be tried:

- After a calculation with FEKO, repeat it with a finer mesh. The number of elements should be at least 1.5 times greater than with the initial calculation. If there is a large difference in the results, then the results cannot be considered correct. In this case the model should be refined, either by improving the meshing, or by consideration of other factors that may influence the results, for example the validity of the techniques used. Any warnings provided by the FEKO kernel during the solution phase should be carefully considered as these are often an indication of the source of inaccuracies in results.
- Perform a power balance check. The power fed into an antenna through the source must be equal to the sum of the radiated power and any losses in the antenna material. The radiated power is automatically calculated for the specified sector if the required far field calculation has two or more angles in each angular direction, while the losses in materials are always calculated. These values can be extracted from the *.out file and used to confirm the power balance.

Part II

Working with CADFEKO

3 Introduction to CADFEKO

CADFEKO is the component of the FEKO Suite that facilitates the creation and set up of FEKO models in a graphical or CAD environment. This involves defining and meshing of geometry as well as specifying the electromagnetic parameters and solution configuration. CADFEKO also makes provision for the definition and launching of an optimisation process on the defined model.

CADFEKO supports parametric model construction. If the model is constructed using variables, the entire model can be modified by changing the values of these variables. This is used, for example, to adapt the size of an antenna to a required frequency or to make provision for model changes during optimisation. CADFEKO maintains the construction history so that, for example, a union operation is automatically updated if any of the individual objects involved in the union are modified afterwards.

Similarly, the user defines media properties and applies them to the relevant parts of the model. All of these properties can then be modified by changing the applicable medium parameters.

3.1 CADFEKO overview

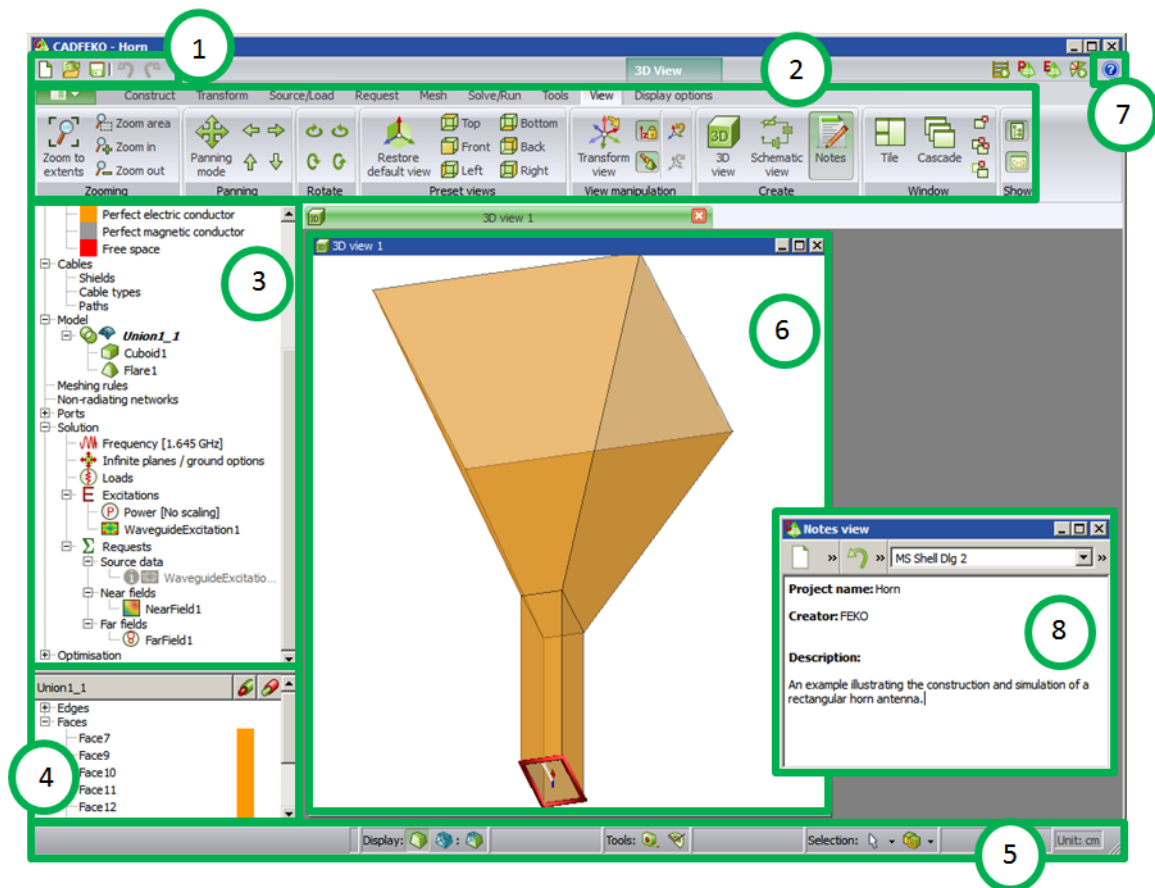
3.1.1 CADFEKO Getting started page

When starting CADFEKO, the CADFEKO start page will be displayed, giving quick access to *Create a new model*, *Open an existing model* and a list of recently opened models. Links to the PDF's for the FEKO suite are also available here along with FEKO introduction videos. It is recommended that these videos are watched by first time users before attempting this example.



3.1.2 CADFEKO window

The various main elements and terminology of the CADFEKO window will be briefly described. These terminology will be used in the examples to follow.

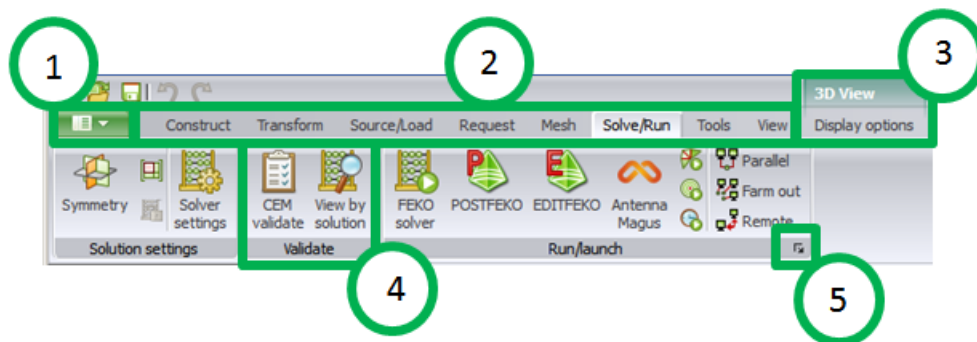


1. **Quick access toolbar** These items give the user quick access to controls such as *New model*, *Open model*, *Save model*, *undo* and *redo* actions (grouped at the left side of the toolbar) as well as launching the FEKO solver, POSTFEKO (for the display of the results obtained by the FEKO solver), EDITFEKO and PREFEKO (grouped at the right side of the toolbar, next to the help button [7] and called *Application Launcher*).
2. **Ribbon** The ribbon contains the application menu, default tabs, contextual tabs and contextual commands.
3. **Model tree** The model tree contains context menus to manage parametric variables, creation of named points, workplanes, defining media, defining cables, mesh refinement, non-radiating networks, adding of ports and excitations, setting the frequency, requesting calculations and setting up optimisation runs.
4. **Details tree** The details tree contains the geometry object details (edges, faces and regions). Custom solution and mesh settings may be set here.
5. **Active statusbar** The active statusbar gives the user quick access to general display settings, tools and selection method and type.
6. **3D view** The 3D view enables the user to visualise the geometry and solution settings (such as far field requests, etc.). Additional visualisations such as cutplanes and symmetry can also be displayed.

- 7. Help** The *Help* button gives the user quick access to the FEKO manuals. Context sensitive help is available in all FEKO Suite GUI components by pressing <F1> at any time.
- 8. Notes view** The notes view can be used to document a model. Additional comments, explanations or descriptions can be added.

3.1.3 The CADFEKO ribbon

The CADFEKO ribbon consists of several elements. Please take note of the terminology as it will be used extensively in the examples to follow.



- 1. Application menu** The application menu contains the following commands: *New model*, *Open model*, *Save as...*, *Archive*, *Import*, *Export*, *Check for updates*, *Rendering options* and *Preferences*.
- 2. Default tabs** The default tabs are always visible and contain general commands.
- 3. Contextual tabs** The contextual tabs display context sensitive tabs with commands relevant to the selected view (3D view or schematic view). A coloured tab marker bar above the tabs indicates the current context.
- 4. Group of commands** Similar actions or commands are contained in a group.
- 5. Dialog launcher** Clicking on the dialog launcher will launch a dialog with additional settings that relate to that group.

A message window at the bottom displays messages about user interaction such as geometry creation, meshing, source configuration, etc. It also provides details regarding warnings and error messages. Errors and warnings in the message window will provide links to the corresponding geometry objects in the details tree which resulted in the error or warning. When clicking on the given link in the message window, the respective geometry object will be selected in the model tree, the 3D view and in the details tree.

3.1.4 CADFEKO files

Only the *.cfx file is required to open the model, but when saving the model, CADFEKO automatically saves the *.cfs file — containing the workspace (views, cut planes, etc.) — and the *.cfm and *.pre files — used when solving models (see section 6.16). When optimisation settings have been defined in the CADFEKO model, CADFEKO will also automatically create/save the *.opt and the *.pfg files that contain the relevant information used (in conjunction with the *.pre and *.cfx files) during optimisation.

3.1.5 Preferences

The settings anchor on the application menu provides a number dialogs that allows the user to customise CADFEKO by setting default preferences.

Figure 3-1 shows the *Default settings* dialog. A variety of options can be set, from default model unit settings to display settings.

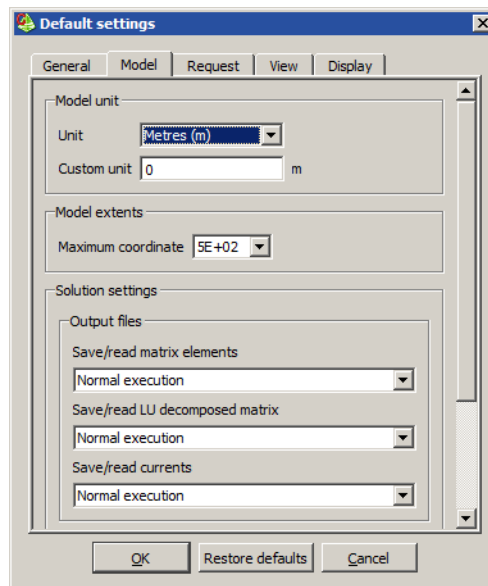
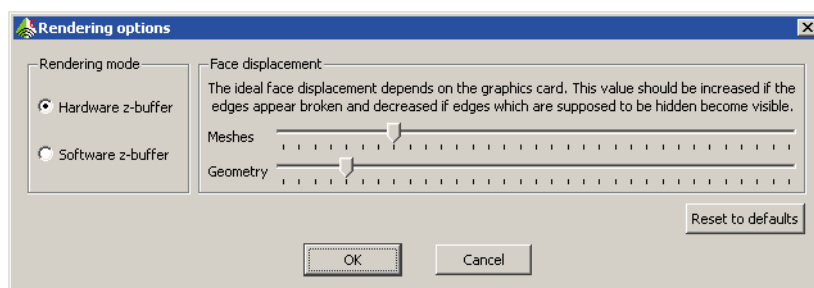


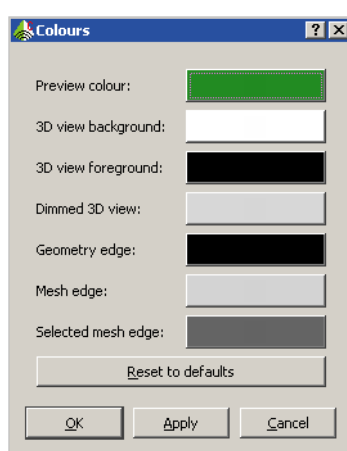
Figure 3-1: The *Default settings* dialog.

3D View rendering and colours

The *Rendering mode* group shows the algorithm used to remove hidden lines. These settings can be configured manually and this may improve the display accuracy, but it can have a significant impact on the rendering performance and memory usage. Hardware rendering (*Hardware z-buffering*) is only enabled when available on the user's system. The sliders change the face displacement for geometry and meshes. The face displacement provides a trade-off between edges appearing broken and supposedly hidden lines being visible. The relative positions of the sliders also determine what is visible if both the geometry and mesh are displayed. Different settings may be required for different view directions. These options are stored in the CADFEKO configuration file.

Figure 3-2: *Rendering options* dialog

The *Colours* dialog allows the user to change the default colours that CADFEKO uses in the 3D view. The *Reset to defaults* button restores the default colours. Figure 3-3 displays the currently available options with the default colours used by CADFEKO.

Figure 3-3: The *Colours* dialog

Messages and log files

In addition to the messages and errors in the message window, CADFEKO creates a text log file for each session in the `logs` subdirectory of the `FEKO_USER_HOME`. If CADFEKO encounters an internal error, the log file for the session is copied to `CADFEKO.ERROR.LOG` and the current model to `CADFEKO.ERROR.CFX`. If these files exist, they will be overwritten. If possible, these error files should be included when reporting any problem to FEKO support.

3.1.6 Checking for updates to the FEKO Suite

Select *Check for updates* on the application menu and click *Check for updates* on the *FEKO update* dialog to poll the FEKO website for update information. This operation checks if any updates are available compared to the installed components. It does not send any information to the web-site.

If automatic updates have been enabled on the *Settings* tab, CADFEKO polls the web-site each time a GUI component is launched if the specified interval between updates have elapsed. Updates can also be downloaded from a local repository for cluster machines or where internet access is not possible. For more information see section 23.

3.1.7 Batch Meshing - changing the model variable from a command line

A stand-alone batch meshing tool can be called from a command line. This tool makes provision for the re-evaluation and meshing of an existing CADFEKO model (after reassigning specific variable values) without launching the graphical user interface.

This functionality is used by OPTFEKO during the optimisation process, and also provides for the manipulation of CADFEKO models and the creation of meshes from CADFEKO models directly from a third-party tool, or a command line script. The required variables and values are included directly in the command line string.

The CADFEKO batch-mesher is launched using the command:

```
cadfeko_batch <filename> [options]
```

Where <filename> is an existing CADFEKO model (*.cfx) (with or without the extension). The path may be included in the <filename> string. After re-evaluation and meshing, the modified CADFEKO model will be saved, replacing the existing *.cfx file as well as the *.cfm, *.pre, *.opt and *.pfg files (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 will not be overwritten).

If the batch-mesher is called with no variable assignments in the options, then the indicated file will simply be re-meshed and saved without any changes to the variable values in the model. This will create (or overwrite existing) *.pre, *.opt, *.cfm and *.pfg files (where applicable).

If the new variable values indicated in the command line cause an error during re-evaluation or meshing, the changes will be aborted (no changes are made to the model) and an error reported. If suspect entities are found in the CADFEKO model after re-evaluation, the re-evaluation and meshing will be completed, and the new model and mesh created, but an error will also be reported (this error will be reported for all suspect items, even if they were not introduced due to changes made by the batch-mesher).

If multiple assignments are included in the command line options for the same variable, then the last value assigned to that variable will be used for that variable before model re-evaluation, meshing and saving.

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, but each variable must be indicated in the same way (i.e. 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 graphical user interface. In this mode, additional information regarding the progress of each phase of the model re-evaluation and meshing is included in the screen output.

3.2 The 3D view

3.2.1 3D view interaction

The 3D views are used to display and interact with the model. CADFEKO distinguishes between a click operation (mouse down and up at approximately the same location) and click and drag (the mouse is moved a minimum distance while the button remains down). Left clicking is used for selection (see section 3.2.2) and point entry (see section 3.2.2). Right clicking opens a context-sensitive menu with operations for the current selection.

Rotation

The model is rotated by clicking and dragging the mouse.

Zooming

It can be zoomed by pressing <Shift> while clicking and dragging the mouse (upward movement zooms in, while downward movement zooms out; horizontal movement is ignored). Rolling the mouse wheel also zooms. (Pressing <Shift> while rolling the mouse wheel slows down zooming.)

CADFEKO tries to zoom in / out on the object below the mouse pointer, hence rolling the mouse wheel while the cursor is not in the centre of the screen will also have a panning effect. After a zoom operation the model will rotate around a point that is more or less in the middle of the screen and at a distance approximately equal to that between the current viewpoint and the model geometry.

Panning

The display can be panned (moved inside the display window) by clicking and dragging while pressing the <Ctrl> key or clicking and dragging with the middle mouse button.

Zoom to selection

When searching for specific parts or elements in the geometry or mesh, the *Zoom to selection* tool may be used. This tool is available in the context menu of any mesh or geometry element in the model or details tree. The *Zoom to selection* tool will modify the panning and zoom factor of the current 3D window such that the selected elements are centred in the view. The rotation of the view is not adjusted - elements in the selection may therefore be hidden behind other mesh or geometry. These hidden elements can easily be identified by manually rotating the model without changing the selection, or by setting the model *opacity*.

Snapping to special points

CADFEKO uses a powerful auto-snapping algorithm. Snapping enables the selection of special points from the 3D view. When pressing <Ctrl><Shift> and hovering over the model with the mouse cursor, special snapping points near the mouse cursor will be indicated by blue dots. The active snapping point is indicated by a black dot and can be selected by pressing <Ctrl><Shift> and clicking on the current point. The special snapping points include: named points, geometry points, geometry face centre, geometry edge centre, mesh vertices and the grid.

When snapping to the workplane, a small grid will be displayed at the mouse cursor. When two objects overlap, pressing <Ctrl><Shift> and repeatedly pressing <Tab> may be used to toggle between their respective snapping points.

When snapping is used to align a new workplane, it should be noted that the history of the starting point and the route followed to the destination point (for example the orientation of an edge), affects the orientation of the workplane.

To specify how points will be selected from the 3D view when using mouse-click based point entry (see section 3.2.2), right click on the 3D view and select *Snapping settings*. The *Snapping settings* dialog is shown in Figure 3-4.

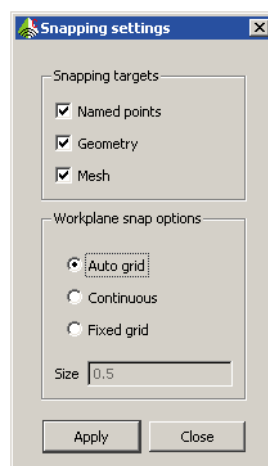


Figure 3-4: The snapping settings dialog

3.2.2 Manipulating items

Selecting items

Items are selected by clicking either in the tree or on the item representation in any 3D view. Selected items are highlighted in the tree and in the 3D view. When parent objects are selected in the tree, CADFEKO displays wire-frame outlines of these objects in the 3D view. These items are not themselves part of the model, but it is useful to be able to distinguish the different parents in a composite part.

Three selection methods are supported.

Single selection: Select the single item under the mouse cursor.

Rectangle selection: Select all items in a rectangle. The user has to click once to place the first corner of the rectangle, then move the mouse cursor and select a second time to indicate the second point in the rectangle.

Polygon selection: A polygon of selection area can be defined using successive clicks. All elements inside the polygonal area will be selected.

The selection type is set to *Auto* selection by default, but the user can set the selection type manually on the *Tool* tab or the status bar. Auto selection cycles through the available selection types.

The *Undo selection* and *Redo selection* buttons on the *Tools* tab allow reverting back / forward to the previous / next selection states, respectively. These states are stored in a separate undo / redo list. This list is cleared each time the model is changed (for example, when adding / deleting / modifying geometry or variables). Note that just like the model manipulations, selections are independent of 3D views (i.e. they are shared between them).

Pressing <Ctrl> while selecting items, in the tree or a 3D view, will add them to the current selection (or remove them if they are selected). Pressing <Shift> while clicking on items in the tree will select everything of the same type in the range from the first item of the current selection. Since the order of items is not as apparent in the 3D view, pressing <Shift> while selecting will merely add / remove the new item.

All selected items must be of the same type, i.e. it is impossible to select a part and some of its faces at the same time. However, simultaneous selection of faces, regions and edges belonging to different parts is allowed.

Deleting items

CADFEKO does not allow deletion of objects that are being used by other items - this is to maintain model consistency. For example, it is not possible to delete a variable that is used in the definition of a medium or to delete a medium that is applied to some part of the geometry. When a delete operation fails, CADFEKO will provide an indication of where the item is used. That item must then first be removed or edited so that it no longer depends on the original item that is to be deleted.

Note that sometimes the dependency may be indirect. For example, consider a variable used to set a local mesh size on *Face1* of *Cuboid1*. The cuboid is then unioned with another and subtracted from some other object. If the user now tries to delete the variable, the message indicates the full tree path to the face (*Subtract1.Union1.Cuboid1.Face1*), but this entity is no longer accessible. At this point it is not possible to remove the local mesh size setting except if the original objects are copied (see section 4.4.4) out of the tree and the last few creation steps are redone. Thus, all geometry dependent settings which are likely to change should be applied as late as possible in the model definition process.

A similar situation can occur when a solution configuration is defined in CADFEKO and then disabled (see section 6.18) after the user has edited the *.pre file. The user can no longer edit or remove any of these settings, but CADFEKO must keep the model consistent since the user can at any time enable the solution again. This can be a problem if the user wants to, for example, modify a variable that is also used in the rest of the model, but this change cannot be made since

it would result in an invalid state in the solution configuration. The only way around this is to re-enable the solution configuration (CADFEKO automatically renames the edited *.pre file), make the necessary change and then manually revert to the renamed *.pre file.

Point entry

The input dialogs of some geometry related and result request related fields allow the entry of field values based on <Ctrl><Shift> + mouse click in the 3D view or tree (for variables and named points). If such point entry fields have focus, their background colour will be yellow. By clicking on the 3D view, the coordinates of the selected point are entered into the field and the focus is shifted to the next field. This allows the spatial definition or editing of geometry or solution requests based on a series of clicks on the 3D view (or tree) and one click on *Create* in the dialog. For some one-dimensional input fields (such as the radius of the sphere primitive), a value is calculated based on the distance between the specified point and the coordinates or values already entered into other fields. Where an input field specifies a direction vector, the vector is considered to extend from the relevant origin to the selected point. If <Ctrl><Shift> is pressed while moving the mouse over the screen (without clicking), the fields values are updated continuously. This is called *Preview mode*. The active fields show the values that would be entered if the mouse was clicked at that position. The values in the active fields are displayed using an italic font to indicate that preview mode is active.

Locking and preview of point entry fields

Fields which accept multiple values from point entry have *Lock* buttons next to them. If such a button is toggled on (down), that field will maintain its current value, and will not be updated based on point selection. The lock mechanism is used when there is a need, for example, to click on points in the workplane without modifying the value in the *n*-direction. Different named points may be individually referenced in the three component fields of a point. Each component is then determined by projecting the specified point along the required dimension. The locking mechanism provides a useful method to realise such multi-point references.

3.2.3 Display options

View modes

Two view modes and a overlay option are supported in CADFEKO that allows the user to inspect the model.

Model view: The model (geometry and mesh) is displayed. This is the default display mode and should be used for model creation and manipulation.

Simulation mesh view: The simulation mesh, the mesh that the kernel will use, is displayed. This option is useful when the mesh needs to be inspected.

Overlay: The overlay options shows a bit of the view mode that is not selected. In model view mode, the simulation mesh edges will be displayed, but not the faces. In simulation mesh mode, the model will be displayed semi-transparent so that the model and simulation mesh can be compared.

Cutplanes

The cutplane dialog for a specific view can be opened by clicking on the *Specify cutplanes* toolbar button or by right-clicking in the 3D view and selecting *Cutplanes*. The first cutplane is created automatically when the dialog is opened (shown in Figure 3-5). Additional cutplanes can be added or deleted using the *Add* and *Remove* buttons. Unchecking the *Active* item on any of the cutplanes deactivates that cutplane without losing its settings. The *Flip* button reverses the operation of the cutplane, hiding the visible region, and showing the invisible region.

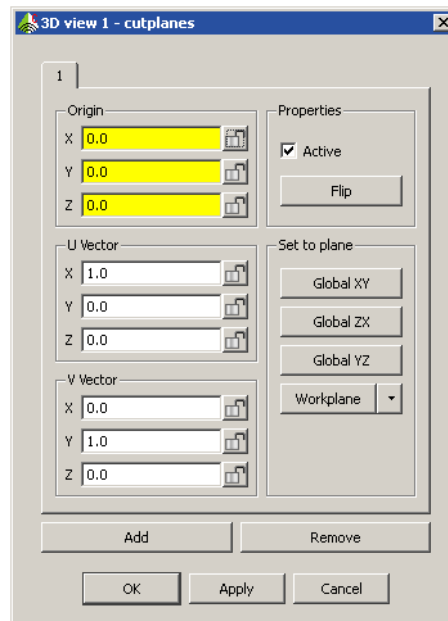


Figure 3-5: The *Cutplane* dialog

The cutplane is specified in a similar manner to the workplane (see section 4.3.2), using an origin and two vectors. These fields all use the standard point entry (see section 3.2.2). Thus the definition of the cutplane may be interactively modified. Before applying any changes to a cutplane, a plane preview is shown in all 3D windows (and will accept point entry from any 3D view), but when applied, the cutplane applies only to the current window.

The cutplane is updated with each change in the dialog. Closing the dialog with *Cancel* reverts to the previous state (i.e. when the *Apply* or *OK* button was last clicked).

If a solid is cut, while the current mode selection (see section 3.2.2) is *Select geometry parts*, a surface (in the colour of the internal region of the solid) is displayed in the plane of the cut where the solid is intersected, even when the geometry is not coloured by medium (see section 3.2.3). Such surfaces cannot be selected, and clicking on them will select geometry that lies behind the cutplane. When the selection mode is anything other than *Select geometry parts*, CADFEKO displays cut solids in a similar fashion to shells (showing only the actual faces) allowing visible selection-access to the internal faces and edges.

The cutplane icon (on the right hand side of the status bar) is activated if the view under the mouse cursor has any active cutplanes. Like the snap options, this icon applies to the view under the mouse cursor, and not the active view.

Selective viewing

Where items are obscured by other items in the 3D view, it can be difficult to set up cutplanes to view these items. In addition, 3D view representations of items such as field calculation requests can cause a general clutter of the view window. In order to tailor views, specific items (geometry, mesh or solution entity items) can be hidden selectively. Hidden items are removed from all 3D views, but are still part of the model and will be exported to the *.cfm and *.pre files and will be considered in any computation process. (Meshing a hidden object will result in a visible mesh part — even if a hidden mesh part with that name existed beforehand.) Hidden items are indicated with grey icons in the model tree.

Items can be shown or hidden by selecting them in the 3D view or tree and choosing *Show/Hide* from the pop-up menu, or by choosing *View → Show/Hide selected items* from the main menu. If the selection contains visible and hidden items, selecting *Show / Hide* toggles the hidden state of each of the selected item. Selecting *View → Show all* in the main menu shows all hidden items (geometry, meshes and solution entities) in the model, independent of their current state. The *Show all geometry* and *Show all meshes* options show all geometry and mesh parts respectively. Ports can be shown/hidden at the port level or the level of the individual mesh/geometry instances.

Note that the visibility buttons on the 3D view toolbar (see section 3.2.4) apply to all items of that type in the selected view only and take priority over the shown/hidden status of all entities of that type, i.e. if the visibility is off in the 3D view, selecting *Show all* will not influence the display.

Item viewing properties are not saved as part of the model or session and if the model is saved and reloaded, all items will become shown (visible) again.

Colouring options

The available colouring options are described below.

Colour by element normal: All parts are drawn with the same colour. The two sides of faces are coloured differently to indicate the normal direction of the faces. On the geometry, the normal side is coloured green, while the reverse side is coloured red. On the mesh, the normal side of each element is coloured blue, while the reverse side is coloured brown.

Colour by region medium: Each region medium is indicated according to the media list under *Media* in the model tree (see section 6.1 and section 6.1.2). In the mesh view, 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 will have the free space colour when viewed from outside, but the view from inside (utilising a cutplane or after some elements of the region boundary have been deleted) will be the colour of the dielectric/metallic medium of the inside region. In the geometry view, regions are, however, 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 defined as coating for that wire respectively.

Colour by face medium: The faces are displayed according to the material type of each face. There is no differentiation of the colouring according to the direction from which faces are viewed on the mesh. In the mesh view, 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. As an example, an object in free space will have the colour of free space (red by default) on the outside of the object.

Faces and wires can also have coatings applied to them. The colour of the coating is used by default, but can be disabled by toggling the *Coatings* button in the *Style* group.

Element visibility controls

The model and simulation mesh visibility controls are controlled using the controls in the *Model visibility* and *Simulation mesh* groups. These controls allow the user to specify how the elements should be rendered (edges, faces, etc.).

Named points, annotations, solution entities and settings

The visibility of named points (see section 4.2.2), requested results (see section 6.12) that have a representation in the 3D view, port annotations as well as symmetry settings (see section 6.8) and periodic boundary conditions (see section 6.9) can be controlled.

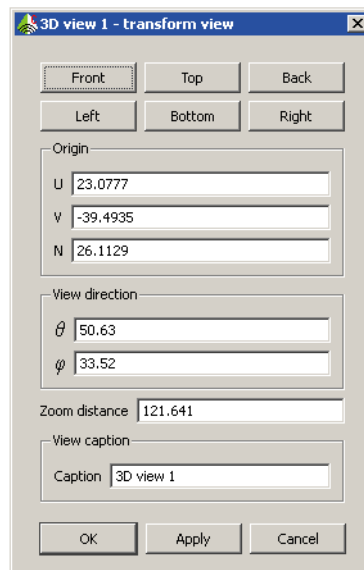
When *Show/hide port annotations* is on, a tag will be displayed at each active port, indicating the port name, and the names of the excitations and loads that are applied at that point. Annotations are shown for both the geometry and mesh manifestations of each port, and are only displayed if the ports are shown.

The visibility of Symmetry and Periodic boundary conditions can be set using the toggle buttons, but these settings are overridden when the Symmetry or Periodic boundary condition settings dialogs are open, in which case the relevant preview will always be shown.

3.2.4 View settings (View tab)

Viewing angle and distance

The *Transform view* dialog (shown in Figure 3-6) opens with the current view settings represented by the position of the *Origin* (defined as the centre of the view), the spherical angular coordinates θ and φ that specify the *View direction* and the current radial *Zoom distance*. See section 3.2.4 for an indication of the icon used for the *View transformation* dialog. These values are updated each time the 3D view is changed. This means that rotating, moving or zooming while the *transform view* dialog is open causes the values in the relevant fields to be updated.

Figure 3-6: *Transform view* dialog

Vertical Z-lock

The *Vertical Z axis* button controls the rotation of the model in such a way that the *z*-axis remains vertical on the screen (i.e. locking it in place). The next two buttons zoom to the full model extent or to a selected window, respectively. (To zoom to a window after clicking this button, click and drag the mouse, without releasing the mouse button, to define the window.) The *Restore default view* button resets the view angles to the standard position.

View undo and redo

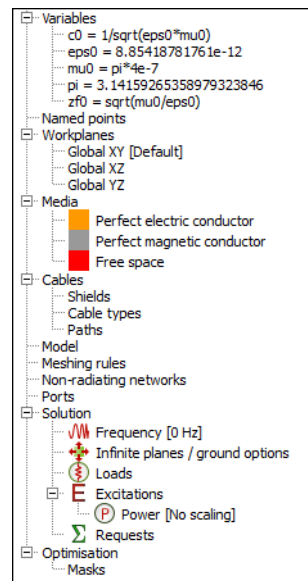
Undo view action and *Redo view action* buttons (these can also be accessed with the short-cut keys <Alt><←> and <Alt><→>) are also provided. These apply to all view manipulations in the current view. View manipulations history is stored in a separate list from the one used to store model manipulations. View operations can therefore be undone independent of geometry modifications. The *Undo / Redo* buttons are disabled if there are no additional undo / redo operations available in the view transformations history.

3.3 The model or contents tree

The model tree provides a complete representation of the current model.

The top (or contents) tree (shown in Figure 3-7), contains the lists of variables, named points, workplanes and media, as well as an overview of the geometry, meshes, non-radiating general networks, transmission lines, ports, solution requests and the optimisation configuration. Note that the model tree is sorted alpha-numerically within each branch.

The entire hierarchy of geometry creation can be seen in the *Geometry* branch of the contents tree. Where objects are derived from existing ones (for example, the individual objects used in a Boolean operation or the original object before a split operation), the original (parent) objects are removed from the top level of the model and listed as sub-branches under the new object in

Figure 3-7: The *model tree* or *contents tree* in CADFEKO

the model tree. The term *part* is used for highest-level items. These can be at the root level under *Geometry* or in the top level of an assembly and are the objects that are visible in the 3D view.

Right clicking on any entry in the model tree will open an appropriate pop-up menu. Double clicking on an item in the model tree will display the *Properties* for that item (except for some headings, in which case a new item is added).

All branches in the geometry branch of the model tree below the selected item can be expanded / collapsed by pressing <*> or selecting *Collapse / Expand all* from the pop-up menu. Collapsing an entry will show all the direct parents only. To hide the direct parents (but keep the rest of the state) use the <-> key. Likewise the <+> key expands only the current level. The model tree may also be navigated using the arrows on the keyboard.

When specific items are hidden (see section 3.2.3) they are displayed with greyed icons in the model tree.

The bottom (or details) tree shows detailed information (faces, edges, regions, transforms, etc.) of a single part, when selected in the contents tree (the selected part is highlighted, labelled with a bold italic font). The lists of faces, edges and regions only apply to top-level parts — these items are not listed for parent objects in sub-branches of the geometry tree.

The tree area can be hidden by dragging the slider all the way to the left. When the tree area is hidden, it can be shown by dragging the slider to the right or by selecting *View → Tree* from the CADFEKO menu.

3.3.1 Icons displayed in the model tree and details tree

Icons next to each item in the trees indicate their type, media, the application of transforms, mesh properties, etc. (The default names also indicate the type, but after renaming, for example, *Assembly1* to *Antenna*, the assembly indicator icon become very useful.) These icons are listed.

Primitive icons:

	– Cuboid
	– Flare (cut-off pyramid)
	– Sphere
	– Cylinder
	– Cone
	– Polygon
	– Rectangle
	– Ellipse
	– Parabolic surface
	– Line
	– Polyline curve
	– Analytical curve
	– Elliptical arc
	– Parabolic arc
	– Hyperbolic arc
	– Bézier curve
	– Spline curve through points
	– Helix or spiral
	– Imported (Parasolid) CAD body
	– Surface body (e.g. created with face copy or explode)
	– Curve (edge / wire) body (e.g. created with edge copy or explode)
	– Part that has been converted to a primitive
	– Mesh part

Medium icons:

	– Create dielectric medium
	– Create metallic medium
	– Create layered dielectric



– Create impedance sheet



– Create windscreen medium

Geometry operator icons:



– Boolean union



– Boolean intersection



– Boolean subtraction



– Split



– Object created by spinning / rotating parent



– Object created by sweeping / extruding parent



– Object created by sweeping along a path



– Loft (ruled surface) created between two edges



– This geometry contains edges created by projecting other items onto it



– This geometry contains imprinted points



– Object created by stitching sheet parts



– Simplified geometry (redundant items removed)



– This is an assembly

Indicator icons:



– This item has been transformed



– This part contains dielectric regions



– This face lies on a dielectric region (shown in details tree)



– This item is suspect — it could not be mapped



– The indicated part/region/face/edge contains faults.



– Local mesh properties set on regions, faces or edges (shown in details tree)



– Local wire radius (shown in details tree)



– This icon indicates the target from which an object was subtracted from.

Geometry transforms:

-  – Translation (shown in details tree)
-  – Rotation (shown in details tree)
-  – Align (shown in details tree)
-  – Scale (shown in details tree)
-  – Mirror (shown in details tree)

Network icons:

-  – Add general network
-  – Add transmission line

Ports, loads and excitations:

-  – Wire port
-  – Edge port
-  – Microstrip port
-  – Waveguide port
-  – FEM modal port
-  – Line port
-  – Complex impedance load
-  – Parallel point circuit load
-  – Series circuit load
-  – Voltage source
-  – Current source (can be applied to a line port in a FEM region)
-  – Waveguide mode excitation
-  – FEM modal excitation
-  – Electric point source
-  – Magnetic point source
-  – Point source with a specified pattern

-  – Plane wave excitation
-  – Aperture field excitation
-  – Impressed current excitation
-  – Spherical mode excitation
-  – Frequency specification
-  – Power setting

Calculation requests:

-  – Current request
-  – Far field request
-  – Near field request
-  – SAR request
-  – S-parameter request
-  – Error estimation request
-  – Request to perform cable analysis
-  – Request to calculate the power received by an ideal antenna

Optimisation icons:

-  – Add an impedance optimisation goal
-  – Add a near field optimisation goal
-  – Add a far field optimisation goal
-  – Add a S-parameter optimisation goal
-  – Add a SAR optimisation goal

Icons used in the details tree heading:

-  – For wires and surfaces, the core medium; for tetrahedra, the medium
-  – The layered medium applied as coating
-  – For surfaces, the medium on the normal side; for wires, the surrounding medium
-  – Only used for surfaces; the medium on the back side

3.3.2 Medium icons in the details tree

In addition to the indicator icons, showing special properties of faces, regions, edges and parts, medium information is shown using specifically coloured blocks in additional columns next to the entries in the details tree (shown in Figure 3-8). The colour used for each medium can be set by right clicking on the medium in the *Media* list and choosing *Change display colour*. The colours indicated in the details tree coincide with the colourings that will be applied in the 3D view when choosing one of the colour-by-media options (see section 3.2.3).

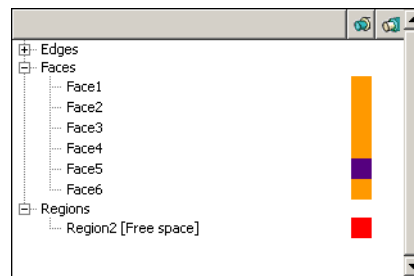


Figure 3-8: The *Details tree* in CADFEKO

For geometry regions there is a single column indicating the region medium (PEC, free space or one of the defined dielectrics).

For geometry faces there are two columns — the first indicates the face medium (PEC, one of the metallic media, a *Layered dielectric*, if it is a thin dielectric sheet or an *Impedance sheet*). The second indicates the *Layered dielectric* if a coating is applied to the face.

For wires there are also two columns. The first indicates the wire medium (PEC or one of the metallic media) and the second the coating.

For tetrahedral meshes, there is only one column which indicates the medium (one of the defined dielectrics).

For surface meshes there are the two columns (of the same type as for geometry faces) and two additional columns which indicate the medium (PEC, free space or one of the defined dielectrics) on the front and back respectively.

For segment meshes there is, in addition to the two columns displayed for wires, a column indicating the surrounding medium.

3.4 Project tools

3.4.1 Distance between points

The *Measure distance* tool can be opened by clicking the *Measure distance* icon on the *Tools* tab. The absolute distance between two points together with the relative x, y and z distance can be determined by pressing <Ctrl><Shift> and clicking with the mouse cursor on any two respective points in the model. The *Measure distance* dialog is shown in Figure 3-9.

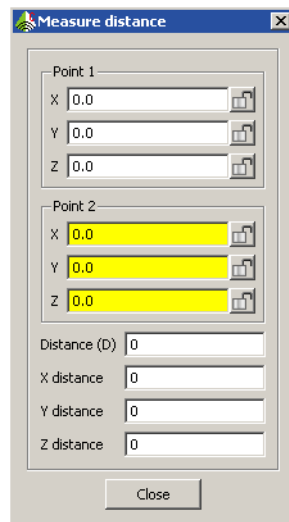


Figure 3-9: The measure distance dialog

3.4.2 Angle between points

The *Measure angle* tool can be opened by clicking the *Measure angle* icon on the *Tools* tab. The angle between three points can be determined by holding down <Ctrl><Shift> and clicking with the mouse cursor on any three points in the model.

3.4.3 Calculator

The *Calculator* allows the evaluation and testing of variable and named point based expressions without modifying any model entries. The calculator is located on the *Tools* tab on the ribbon.

The format of the result can be controlled. *Scientific* uses exponential notation, for example, 0.01 becomes 1.0e-2. *Engineering* format is similar to *Scientific*, except that the exponent part is always a multiple of 3. Thus 0.01 becomes 10.0e-3. *Decimal* uses a fixed notation without an exponent. This is not recommended for numerically small numbers, for example, with 5 decimals, 1.0e-6 becomes 0.000001 which means all information is lost. The *Decimals* field gives the number of digits after the decimal point.

3.4.4 Notes editor

CADFEKO provides a rich text editor tool (shown in Figure 3-10) in which the user can add comments to a model. The editor is opened from the standard toolbar or by toggling the Notes editor icon on the *View* tab. The notes view opens in its own window, allowing usage of multiple monitors.

Unlike the 3D views, there is only one notes editor.

The toolbar at the top of the noted editor provides buttons to *Clear* and *Print* the notes, for *Undo* / *Redo* changes as well as the standard *Cut*, *Copy* and *Paste* commands. Multiple undo / redo commands are supported. There is no facility to *Cancel* any changes to the text, but this may be achieved with the *Undo* command.

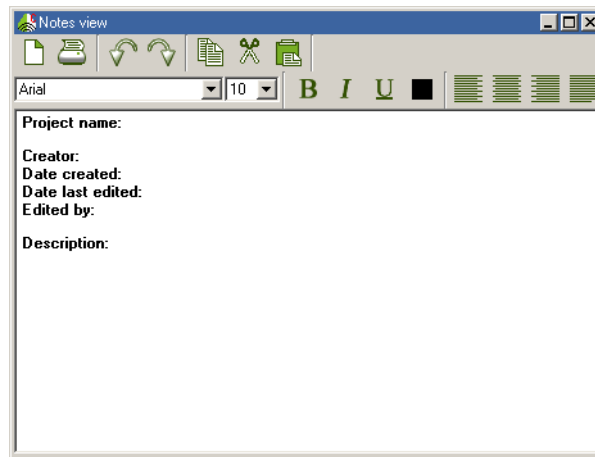


Figure 3-10: The notes editor

The editor allows a choice of font name, font size, font type (bold, italic, underlined) and text colour as well as setting the justification. If no notes have been previously saved with the model, then a basic template is loaded (as shown in Figure 3-10).

The contents of the notes editor is also written to the *.pre file for users that need to create the model in CADFEKO and then change the model using EDITFEKO. The contents are written to the top of the *.pre file as a series of comments.

3.4.5 Archiving models

When working with complex models, it is sometimes useful to create a backup of the model in a specific state that may be returned to later, before continuing. CADFEKO provides a facility to store and retrieve different versions of the project files. Select *Archive* → *Archive model* from the application menu and enter a comment to go with this version of the model files. The current model is saved and copied to the archive directory. Note that external data files, such as the pattern data used for patterned point sources (see section 6.5.3) and ideal receiving antennas (see section 6.12.8) which can change independently and are not considered part of the CADFEKO model are not included in the archive.

Select *Archive* → *Revert from archive* from the application menu to open the *Revert model* dialog which lists the archived versions with their time stamps and comments. Select a version and click *Revert* to revert to that version. Note that the revert operation cannot be undone and the model files at the time of the revert operation are overwritten by those retrieved from the archive. A model should therefore usually be archived before reverting to a previously archived version.

Finally selecting *Archive* → *Delete from archive* allows deleting archived versions to save disk space. Multiple archives can be selected and deleted simultaneously.

3.5 Short-cut keys

The keyboard short-cut keys that are available in CADFEKO are listed in Table 3-1 and Table 3-2.

Table 3-1: Short-cut keys (running other FEKO components)

<i>Short-cut key</i>	<i>Description</i>
<Alt><1>	Run EDITFEKO
<Alt><2>	Run PREFEKO
<Alt><3>	Run POSTFEKO
<Alt><4>	Run FEKO
<Alt><6>	Run OPTFEKO
<Alt><7>	Run TIMEFEKO

Table 3-2: CADFEKO short-cut keys

<i>Short-cut key</i>	<i>Description</i>
<F1>	Context-sensitive help for the dialog / window that has focus.
<F2>	Rename selected item.
<F9>	Create workplane.
<Del>	Delete selected items.
<Shift><Ins>	Paste clipboard text.
<Ctrl><Ins>	Copy selected text.
<Shift><Del>	Cut selected text.
<Ctrl><A>	Select all items of the current selection type.
<Ctrl><C>	Copy selected text/image.
<Ctrl><E>	Export image.
<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>	Create mesh.
<Ctrl><Shift><M>	Quick mesh - mesh with the current settings without requiring confirmation.
<Ctrl><N>	Create new model.
<Ctrl><3>	Create new 3D view.
<Ctrl><O>	Open model.
<Ctrl><P>	Print
<Ctrl><S>	Save model.
<Ctrl><V>	Paste.
<Ctrl><X>	Cut selected text.
<Ctrl><Y>	Redo model creation / modification.
<Ctrl><Z>	Undo model creation / modification.
<Alt><←>	Undo view manipulation.
<Alt><→>	Redo view manipulation.

Table 3-3: CADFEKO tree and message window short-cut keys

<i>Short-cut key</i>	<i>Description</i>
<Alt><T>	Open/closes the tree
<Alt><W>	Open/closes the message window

Table 3-4: CADFEKO create solids short-cut keys

<i>Short-cut key</i>	<i>Description</i>
<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

Table 3-5: CADFEKO create surfaces short-cut keys

<i>Short-cut key</i>	<i>Description</i>
<S,1>	Create a polygon
<S,2>	Create a rectangle
<S,3>	Create an ellipse
<S,4>	Create a paraboloid
<S,5>	Create a NURBS

Table 3-6: CADFEKO create curves/arcs short-cut keys

<i>Short-cut key</i>	<i>Description</i>
<V,1>	Create a straight line
<V,2>	Create a polyline
<V,3>	Create a fitted spline
<V,5>	Create a Bezier curve
<V,6>	Create an analytical curve
<A,1>	Create an elliptic arc
<A,2>	Create a parabolic arc
<A,3>	Create a hyperbolic arc
<A,4>	Create a helix

Table 3-7: CADFEKO modify short-cut keys

<i>Short-cut key</i>	<i>Description</i>
<U>	Union the selected parts
<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 curve parts to form a loft surface

Table 3-8: CADFEKO view short-cut keys

<i>Short-cut key</i>	<i>Description</i>
<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

4 Creating and meshing geometry in CADFEKO

In this chapter the various methods and concepts related to the creation and manipulation of geometry and meshes in CADFEKO are discussed.

4.1 Model geometry properties

4.1.1 Setting units

To set the unit used for all distances and dimensions in CADFEKO, click on the *Model unit* button on the *Construct* tab. Besides the standard units provided in the dialog, the user can specify an arbitrary unit conversion factor with respect to metres. For example, the model unit should be 1×10^{-6} when working in micrometres. The unit is displayed on the status bar (see section 3.1).

Changing the unit does not directly modify any numbers specified in CADFEKO, but rather the interpretation of all numbers (created before and after the unit change) — hence the unit can be changed at any time during construction of the model.

4.1.2 Geometry extents

Operations on geometry (such as checking if two points are at the same physical location) require a numeric tolerance. This tolerance is dependent on the model size. For example, microwave structures may require dimensions that differ by only few micrometers, but this level of accuracy is clearly not required for studies of antennas on large ships. CADFEKO therefore allows the specification of the model size — click on the *Extents* icon on the *Construct* tab. The model extent is the same in all coordinate directions and therefore only one value needs to be specified. This is the *Maximum coordinate* which gives the largest offset — in either direction — along any of the three axes. For example, if the *Maximum coordinate* is 500, the entire geometry must fit inside a 1000x1000x1000 box centred at the origin. The size of the extents are specified in the CADFEKO model unit (see section 4.1.1).

The tolerance on the model dimensions can be determined by taking the *Maximum coordinate* divided by 5×10^8 . If coordinate values differ by more than this amount, they are considered unique. Note, however, that values are only guaranteed to be considered identical if the distance between them is less than a hundredth of the model tolerance. (Between these values is a range where the uniqueness and connectivity cannot be guaranteed.)

The default value for the geometry extents is 5E+02 (500). It is recommend that this value be used except if the model is large (for example, when modelling an automobile in mm model units), or small with a requirement for high dimension-accuracy (accurate CAD model and / or mesh). For extent settings other than the default value of 500, exported Parasolid models will not be in the same unit used in CADFEKO (see section 5.1.2.)

The extents box limitation applies to all geometry. For example, an intersection between two spheres cannot be performed if either sphere exceeds the size box even if the resultant intersection would be within the extents boundary. CADFEKO will display an error if the geometry exceeds the specified size. In this case, the extents can be changed without having to close the dialog that caused the error and the operation re-attempted.

4.2 Variables and named points

4.2.1 Variables

CADFEKO supports fully parametric models. Thus most input fields can be specified using variables or mathematical expressions such as $1+\sqrt{x}$, where x is a user defined variable. The expressions are stored as part of the model and when a variable is changed, all values and items depending on that variable are re-evaluated and updated. It is therefore very easy to adapt a model using variables and a little planning, for example, to modify the geometry based on the frequency. Since it is possible to edit geometry objects, variables can be introduced after constructing the model.

While the geometry is fully parametric, the mesh is not (storing expressions for mesh vertices would have significant implications on memory requirements).

In general, operations at the part level are fully parametric, but operations that apply to individual elements are not. When working with mesh elements, most input fields still accept expressions (for example, adding +1 to a coordinate of a vertex will move that vertex by one unit). These expressions, however, are not maintained, but converted to numerical values when the operation is executed.

Defining and editing variables

Variables may be added by double clicking on the *Variables* entry in the tree or selecting *Model* → *Add variable* from the main menu. The first character of a variable name must be alphabetic ('a' through 'z' or 'A' through 'Z') or the underscore; the remaining characters may be numeric ('0' through '9').

Variable names are case insensitive, i.e. 'a' and 'A' are treated as the same character. The *Expression* defines the value of the variable and may be a simple number (such as 1.23) or a mathematical expression — which may use round brackets, the operators +, -, *, \, ^ (exponential notation), as well as the functions listed in Table 4-1 and Table 4-2 and may reference other defined variables.

If there is an error in the expression, an error message will be displayed in the message window when the *Create* or *Apply* button is clicked. The variable can only be added/modified when the expression is valid. A short comment for the variable can be added to the *Comment* field. Note that the adding of comments for variables are optional. Predefined variables will already contain a comment. The *Evaluate* button evaluates and tests the validity of the expression without closing the dialog. The result is maintained until the next time the expression is evaluated and is not updated automatically when the expression is changed.

Variables that have already been created can be changed by double clicking on the variable itself (or by right clicking and selecting *Properties*). CADFEKO will display a *circular dependencies* error if a variable is changed in such a way that it depends on itself. All items with defined dependencies on the variable will be updated automatically. If this results in invalid geometry (for example, a sphere radius that becomes zero or an intersection or split that becomes empty), an error is written to the message window and the variable modification will not be accepted. The *Modify variable* dialog may be left open while editing the item which gave the error. Thereafter focus reverts back to the *Modify variable* dialog. Variables can be deleted via the context menu.

Table 4-1: Trigonometric functions supported in CADFEKO expressions

sin cos tan cot	trigonometric functions (arguments expected in radians)
arcsin arccos arctan arccot	trigonometric inverse functions (results given in radians)
atan2	atan2(y, x) yields arctan(y/x) in the range $-\pi \dots \pi$
sinh cosh tanh	hyperbolic functions

Table 4-2: Mathematical functions supported in CADFEKO expressions

fmod	fmod(a, b) returns the remainder of the division a/b
deg	converts radians to degrees
rad	converts degrees to radians
sinh cosh tanh	hyperbolic functions
log	logarithm to base 10
ln	natural logarithm
exp	exponential function
sqrt	square root
abs	absolute value
step	step(x) is 1 when $x > 0$; otherwise it is 0
ceil	rounded upwards
floor	rounded downwards
min max	min(a, b) or max(a, b) gives the minimum / maximum of the two arguments

The delete operation will fail for all variables that are still referenced in any other part of the model (for example, in defining other variables or geometry). A variable or multiple variables may also be copied. Select the variables, right click and select *Copy* from the context menu. The copy of the variable will be indicated by appending *_1* to the variable's original name.

The variables in Table 4-3 are pre-defined as part of a new model. These variables may be deleted and / or modified just like any other variable and only have an effect if the user explicitly refers to them.

Multiple variables may also be modified before re-evaluating by right clicking on *Variables* in the tree and selecting *Modify multiple variables*. The *Modify multiple variables* dialog is shown in Figure 4-1.

Table 4-3: Predefined variables

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

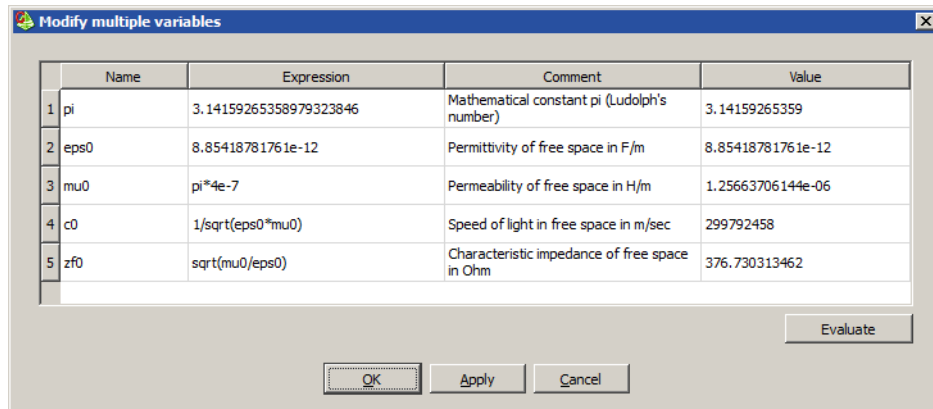


Figure 4-1: Modify multiple variables dialog

4.2.2 Named points

Named points can be created by double clicking on the *Named points* entry in the tree, by right clicking it and selecting *Add named point*, or by selecting *Add point* on the *Construct* tab. Named points may be copied by right clicking and selecting *Copy* from the context menu. The copy of the named point will be indicated by appending *_1* to the point's original name.

Named points are effectively vector variables and are typically used to create geometry that should change when the point is modified (see section 3.2.2). The *Add named point* dialog contains input fields for the three coordinates of the point in the global coordinate system.

The *x*, *y* and *z* components of a point may be accessed using a dot followed by the required component, for example, '*P.x*' is the *x* component of point *P*. Points may also be constructed using the '*pt*' command. For example, the expression '*pt(1,1,1) + pt(2,1,1)*' will result in a point definition of (3,2,2). Only the subtract and add operations are allowed between two points (the corresponding components are added / subtracted) and a point may be multiplied with or divided by a scalar. The '*abs*' function may also be applied to points. This function gives the distance from the origin to the point, i.e. the vector length if the point is interpreted as a scalar value. Currently no other operations may be applied to points.

4.3 Creating geometry

CADFEKO provides various tools for the creation of basic geometric forms from which a more complex geometry can be constructed. These basic forms are referred to as 'primitives'. Geometry primitives can be categorised into three groups.

1. Solid primitives

2. Surface primitives and
3. Curve primitives

The basic concepts involved in the creation of geometry primitives are common to all primitive types.

4.3.1 General geometry primitive dialog behaviour

A typical primitive creation dialog (the *Create cuboid* dialog) is shown in Figure 4-2.

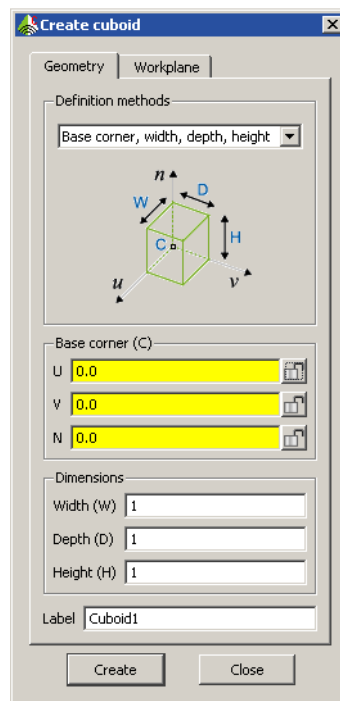


Figure 4-2: The geometry tab of the *Create cuboid* dialog

Workplane: All geometry objects have a fixed orientation according to their workplane. Note that any modifications to a primitive's workplane, will only affect that specific geometry.

Definition methods: Many geometry primitives support creation using different parameters. The easiest definition depends on the information available for the geometry. Each definition is also accompanied by an image that illustrates the dimensions required for a complete and unique definition of the primitive.

Points and dimensions: All coordinate and dimension fields support general point entry (see section 3.2.2). All required dimensions are constrained (for example, the width specified for a cuboid primitive must be non-zero). If a value is applied that does not meet the specific constraints, CADFEKO will display an error in the message window and highlight the offending field during creation or when application of changes is attempted. Where negative values are acceptable (most dimension and angle fields), they are interpreted in the opposite direction. (If a value of 1 implies a length in the direction of the positive u-axis, then a value of -1 will be valid, and interpreted as a length in the direction of the negative u-axis). For an angle field, if 10 implies rotation in the positive ϕ direction, then a value of -10 is interpreted as rotation in the negative ϕ direction.

Label: The *Label* field allows entering the name of the new object. This is visible in the model tree and can be changed at any time after creation. The *Label* may be chosen freely, but two identically labelled objects are not allowed in the same level of the tree (irrespective of type). Note that labels used in FEKO are considered case insensitive.

Create / Close: A new object is created when the *Create* button on the creation dialog is clicked. The dialog can be closed by clicking *Close* or by pressing <Esc>.

4.3.2 The workplane

In CADFEKO, there are three predefined workplanes namely: Global XY, Global XZ and Global YZ. Any of these workplanes can be at any time be set as the default workplane by right-clicking on the respective workplane in the contents tree and selecting *Set/Unset as default* from the pop-up menu. After a workplane is set as the default workplane, it will be indicated by the text, *[Default]* in the model tree. The default workplane will be used as the starting workplane for the dialogs concerning primitive creation, operators and transforms. If necessary, the workplane can be modified on any *Workplane* tab for primitive creation, operators and transforms. (The easiest is to use point entry - <Ctrl><Shift> + click on workplane in the tree.)

The workplane is determined by the orientation of the axis which is displayed as a blue *X/U*, green *Y/V* and red *Z/N* tangential lines. When an axis is obscured by geometry, it will still be visible as a dotted line in their respective colour.

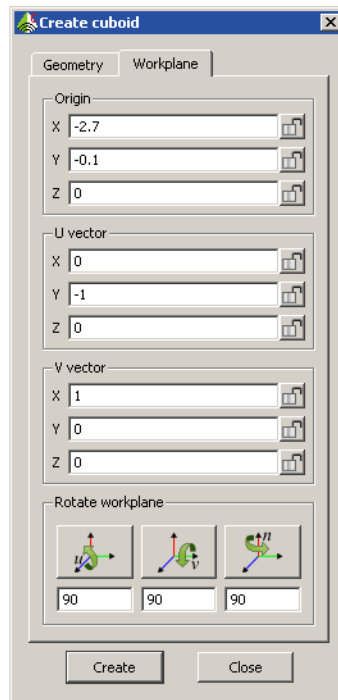
The workplane display can be switched off by clicking *Show/Hide* workplane on the 3D view toolbar. This action will also hide the dynamic grid. The grid is dynamic as the tick marks are determined by the amount to which the model is zoomed in or out. When the model is zoomed out, the interval between the tick marks will be increased. An extra option is available under the *Show/Hide* workplane to hide the tick marks on the dynamic grid.

The workplane is specified in terms of an *Origin* which defines the position, and the *U vector* and *V vector* directions which define the orientation of the workplane. The workplane can also be rotated around the *U*, *V* or *N* axis. The angle (in degrees) for rotation can be entered underneath the icons (shown in Figure 4-3). Note that these buttons perform actions on the values entered in *U* and *V*.

A workplane can be easily modified by snapping to existing faces, edges and wires (see section 3.2.1).

Only the directions of the vectors are relevant, i.e. using the vector (2,0,0) is the same as using (1,0,0). When calculating the *v*-axis, the component of *V vector* parallel to the *U vector* is removed to ensure that the two axes are orthogonal. Thus, the *V vector* need not be specified exactly orthogonal to the *U vector*, though it may not be specified exactly parallel to *U vector*. The *n*-axis is normal to the *uv*-plane and in the direction $u \times v$.

The *Origin*, *U vector* and *V vector* fields may be specified using the <Ctrl><Shift> (see section 3.2.2) and clicking with the mouse. This means that the current workplane may be used to enter values which will define the settings of a new workplane. (Pressing <Ctrl><Shift> and clicking on the view while any of the *Origin*, *U vector* or *V vector* fields has focus will modify its values.)

Figure 4-3: The workplane tab on the *Create cuboid* dialog

Coordinates set in a given workplane

All primitive geometry objects and solution entities with spatial properties in CADFEKO are defined in the default workplane environment, unless the default workplane was modified on the *workplane* tab. When a workplane is defined (see section 4.3.2), the coordinate system is set in the defined workplane environment.

4.3.3 Creating solid primitives

The *Create solid* group on the *Construct* tab contains tools for the creation of cuboids, flares (a possibly-truncated pyramid), spheres, cylinders and cones. (A typical solid primitive dialog is shown in Figure 4-2.) Solid primitives are by default perfect electric conductors, but can be changed to dielectric or shell structures. This is done by setting region properties (see section 6.1.2).

For cuboid primitives, all faces lie in the principle planes of the indicated coordinate system. For flares, the base is aligned with the *uv*-axes and the height is in the direction of the *n*-axis.

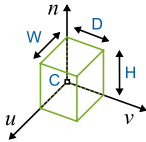
For cylinders and cones, a number of special definition options are available in order to aid flexibility in the creation of complex geometries. The various primitive definition options are depicted and described below. If the *Radius* field for a sphere or cylinder is specified using a point, the surface of the sphere or the extended cylinder will pass through that point. For a cone the radius is determined as the distance between the origin of the local coordinate system and the projection of the point in the local *uv*-plane. To specify a sharp tipped cone, set the *Top radius* field to '0'.

When creating any of the solid primitives, surface primitives or line primitives, default values are added to the geometry dimension fields. The default values are dependant on the zoom level in

the 3D view and are only given to enable a preliminary preview of the primitive in the 3D view. The exceptions to these default values are the *polygon* and *fitted spline*.

Definition options for solid geometry objects

Cuboid - Method 1



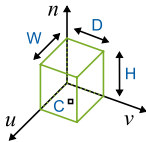
Base corner (C): One corner of the cuboid.

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.

Cuboid - Method 2



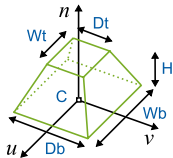
Base centre (C): The centre of the base of the cuboid.

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 - Method 1



Base centre (C): The centre of the flare base.

Bottom width (W_b): The width of the base in the u-axis direction.

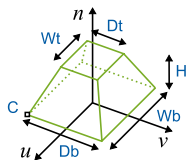
Bottom depth (D_b): The depth of the base in the v-axis direction.

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.

Flare - Method 2



Base corner (C): A corner of the flare base.

Bottom width (W_b): The width of the base in the u-axis direction.

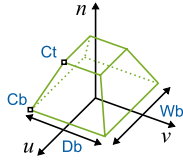
Bottom depth (D_b): The depth of the base in the v-axis direction.

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.

Flare - Method 3



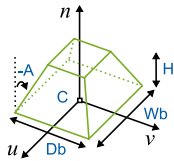
Base corner (C_b): A corner of the flare base.

Top corner (C_t): A corner point 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.

Flare - Method 4



Base centre (C_b): The centre of the flare base.

Bottom width (W_b): The width of the base in the u-axis direction.

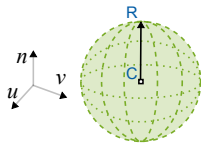
Bottom depth (D_b): The depth of the base in the v-axis direction.

Height (H): The height of the flare, in the n-axis direction.

Flare angle (AU): The angle of the flare from the u-n plane.

Flare angle (AV): The angle of the flare from the v-n plane.

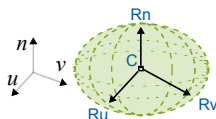
Sphere - Method 1



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

Radius (R): The radius of the sphere.

Sphere - Method 2 (ellipsoid)



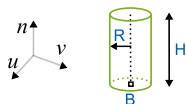
Centre (C): The centre point of the sphere.

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 - method 1

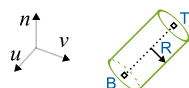


Base centre (B): The centre of the cylinder base.

Radius (R): Cylinder radius (parallel to the uv plane).

Height (H): Cylinder height in the n-direction measured from B .

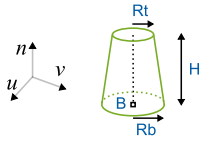
Cylinder - method 2



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).

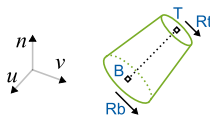
Cone - method 1

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).

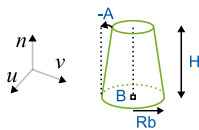
Cone - method 2

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 (perpendicular to the line from B to T).

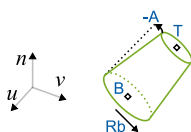
Cone - method 3

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.

Cone - method 4

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.

4.3.4 Creating surface primitives

Flat polygons, elliptical disks and paraboloid surfaces are supported as surface primitive types. More complex surfaces may be created using the loft, spin, sweep or path sweep (see section 4.4.3) operations on line primitives (see section 4.3.5).

The polygon creation dialog is shown in Figure 4-4. All of the specified points must lie in a plane — the simplest way to ensure this is to align the workplane with the plane of the polygon and ensure that all *n*-coordinate values are exactly equal (usually zero).

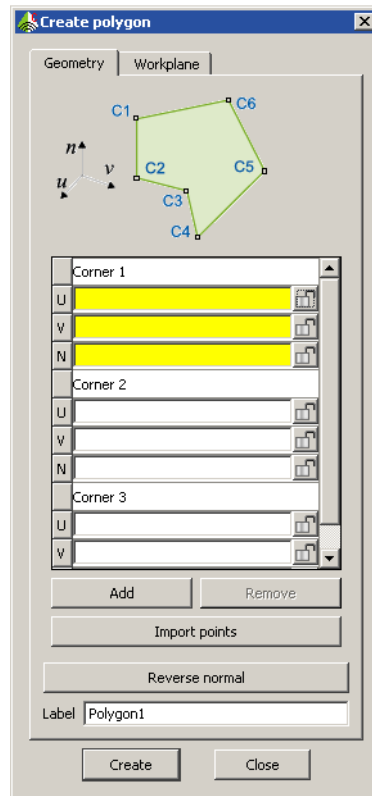


Figure 4-4: The dialog used to create polygons

Clicking the *Add* button adds an additional corner to the list directly below the one that has focus or, if no corner has focus, at the end of the list. The polygon is created by connecting the lines in the specified order and the edges are not allowed to cross. The corner with focus is represented in the 3D display with a blue square. If the last coordinate is entered using the mouse (see section 3.2.2), CADFEKO automatically adds another corner and moves the focus to the added corner.

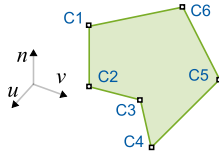
The polygon primitive provides an *Import points* option (see section 4.3.6) to populate the corner point definitions based on an external file.

The *Reverse normal* button reorders the points in such a way that the normal vector (determined in a mathematically positive sense from the direction of the edges) is reversed. When *Create* is clicked, all empty points are removed automatically, and the surface is created.

The definition options for the different surface primitives are depicted and described in the section that follows.

Definition options for surface primitive objects

Polygon

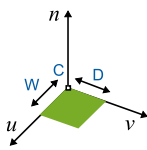


Corner 1 (C_1): The first corner of the polygon.

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 .

Rectangle - Method 1

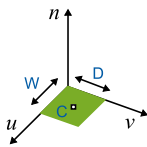


Base corner (C): A corner of the rectangle.

Width (W): The width of the rectangle.

Depth (D): The depth of the rectangle.

Rectangle - Method 2

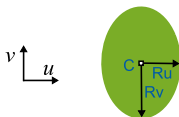


Base centre (C): The centre of the rectangle.

Width (W): The width of the rectangle.

Depth (D): The depth of the rectangle.

Ellipse

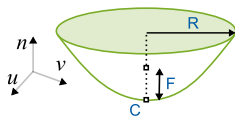


Centre point (C): The centre of the ellipse.

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



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* of the paraboloid is the distance F from the centre point (C) to the focal point. If this is negative, the paraboloid is oriented towards the or $-n$ axis. The focal depth is related to the height (h) by

$$F = \frac{R^2}{4H}.$$

with the height (H) defined as the distance from the centre point (C) to the centre of the aperture of the paraboloid.

NURBS

Specify the order of the Bezier curves: The degree of the Bézier curve in the $\hat{u}$ -direction and $\hat{v}$ -direction.

Point position: The position of each surface control point is specified.

Weight: The weight of each surface control points is specified.

4.3.5 Creating curve primitives

Curve primitives in CADFEKO can be used either as building blocks for constructing geometry (for example, using the spin, sweep or loft operations) or as free-standing conducting wires. In addition to the general curves resulting from intersections between faces of shell bodies, CADFEKO supports a number of simple primitives. These include single straight lines, polylines, analytical curves, fitted splines, Bézier curves, elliptical arcs, hyperbolic arcs, parabolic arcs and helices / spirals. The creation of curve primitives is very similar to the creation of solids and surfaces.

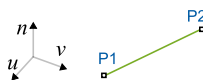
The polyline and fitted spline primitives make use of a list of points. These points may be manually specified, or imported from an external text file (see section 4.3.6). Here, as for polygon surfaces, new fields are created if the last field is entered with the mouse, and blank points are removed before constructing the geometry.

The helix primitive may be used to create conical spirals by setting the top and bottom radii to different values, or flat spirals by setting the height to zero. The direction of rotation of the helix can be specified using the *Left handed* checkbox. When this box is checked, a left-handed spiral will be generated, when it is un-checked, a right-handed spiral will be created. The handedness of the spiral is defined based in the direction from the start point to the end point of the spiral.

The definition options for the different line primitives are depicted and described below.

Definition options for line primitive objects

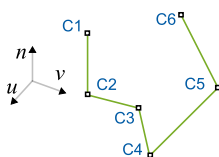
Line



Start point (P_1): The starting point of the line.

End point (P_2): The end point of the line.

Polyline

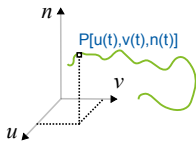


Corner 1 (C_1): The starting point of the polyline.

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.

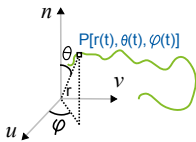
Analytical curve - Cartesian



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 .

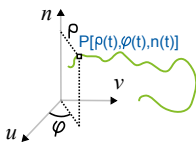
Analytical curve - Spherical



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 .

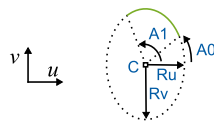
Analytical curve - Cylindrical



Parametric interval: Interval over which the analytical curve is parametrically defined.

Cylindrical description: The Cylindrical description of the analytical curve in the ρ , ϕ and n dimensions as a function of variable t .

Elliptic arc - method 1



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.

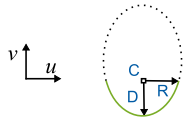
End angle (A1): The angle, from the positive u -axis direction where the arc ends.

Elliptic arc - method 2 (V Major axis direction)

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

Conditions for creating a valid elliptic arc:

$$\text{if } R \leq D : \sqrt{1 - \frac{R^2}{D^2}} \leq \epsilon < 1 \quad (4-1)$$

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

Elliptic arc - method 2 (U Major axis direction)

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.

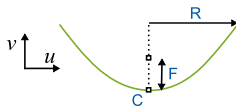


Eccentricity: The eccentricity of the ellipse on which the elliptical arc section lies. The eccentricity must be less than 1 to specify a valid ellipse.

Conditions for creating a valid elliptic arc:

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

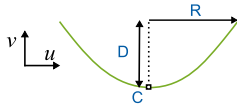
where D is denoted by the depth, R by the aperture radius R and ϵ the eccentricity.

Parabolic arc - method 1

Base centre (C): The centre of the parabola on which the arc lies.

Focal depth (F): The focal depth of the parabola.

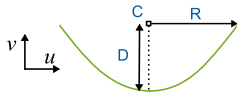
Radius (R): The radius of the aperture of the parabola.

Parabolic arc - method 2

Base centre (C): The centre of the parabola on which the arc lies.

Radius (R): The radius of the aperture of the parabola.

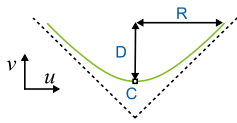
Depth (D): The distance from the apex of the parabola to the centre of the aperture.

Parabolic arc - method 3

Aperture centre (C): The aperture centre of the parabolic arc section.

Radius (R): The radius of the aperture of the parabola.

Depth (D): The distance from the apex of the parabola to the centre of the aperture.

Hyperbolic arc - method 1

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 hyperbola.

Eccentricity: The eccentricity of the hyperbola on which the hyperbolic arc section lies.

Conditions to create a valid hyperbolic arc:

$$1 \leq \epsilon \leq \sqrt{1 + \frac{R^2}{D^2}} \quad (4-3)$$

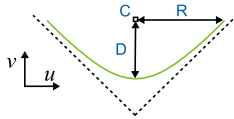
where D is denoted by the depth, R by the aperture radius R and ϵ the eccentricity.

Hyperbolic arc - method 2

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 hyperbola.



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 that not all values greater than one specifies a valid hyperbola.

Conditions to create a valid hyperbolic arc:

$$1 \leq \epsilon \leq \sqrt{1 + \frac{R^2}{D^2}} \quad (4-4)$$

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

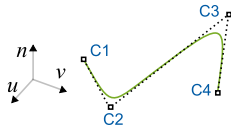
Bezier curve

Corner 1 (C₁): The starting point of the curve.

Corner 2 (C₂): The first control point of the Bézier curve (the curve does not necessarily pass through this point).

Corner 3 (C₃): The second control point of the Bézier curve (the curve does not necessarily pass through this point).

Corner 4 (C₄): The end point of the curve.

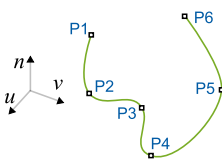


Fitted spline

Point 1 (P₁): The starting point of the curve.

Point 2 (P₂): 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.



Helix - method 1

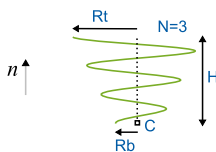
Origin (C): The centre point of the helix base.

Base radius (R_b): The radius of the helix base (parallel to the uv-plane).

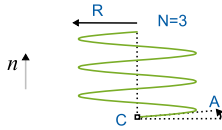
End radius (R_t): The radius of the helix top (parallel to the uv-plane).

Height (H): The height of the helix, in the n-axis direction.

Turns (N): The number of turns for the helix (the rotation direction is chosen based on the *Left handed* checkbox option).



Helix - method 2



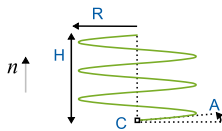
Origin (C): The centre point of the helix base.

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 for the helix (the rotation direction is chosen based on the *Left handed* checkbox option).

Helix - method 3



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. (In this method, the number of turns is related to the height (*H*) and pitch angle (*A*) chosen).

Note that the local wire radius (see section 4.8.4) is specified on an edge (as listed in the details tree) rather than on the curve primitive directly. Specifying the radius for a wire curve (any free edge that is not a face border) that was created by geometry operations (such as the intersection of two crossing faces) is done in the same way.

4.3.6 Import points

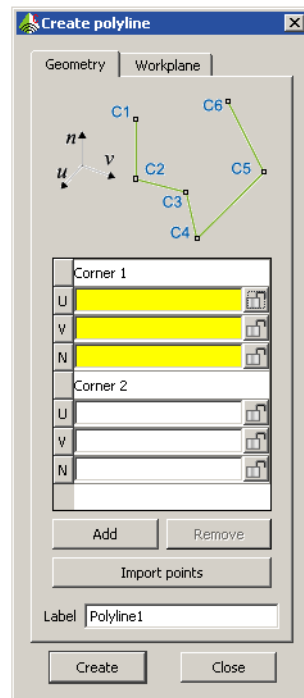
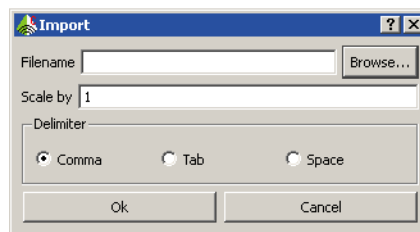
Some primitives (such as the polygon, polyline or fitted spline), as well as the cable path definition for cable analysis and the points list definition for the imprint points tool, are defined based on a list of coordinate points. These points may be entered manually (using the keyboard or based on mouse clicks) or imported from an external file that contains a list of the coordinates. In the creation dialogs of primitives that support the importing of point definitions from an external file, an *Import points* button is available (Figure 4-5 shows the *Import points* button on the *create polyline* dialog).

If the *Import points* button is clicked, the *Import points* dialog, shown in Figure 4-6, is opened to allow file location and data format selection.

A scaling factor may be applied to the coordinates as specified in the import file before they are used (this may be required, for example, to convert to the current *model unit* setting). This scaling is linearly applied to all coordinates, and cannot be specified only in a specific coordinate direction.

File format for imported points

The coordinates may be specified using either a *comma*, *tab* or *space* delimited file format. The format must be set on the *Import points* dialog. The expected file format comprises three columns

Figure 4-5: *Create polyline* dialogFigure 4-6: *Import points* dialog

(for each of the three Cartesian coordinates) separated by the specified delimiter, with each coordinate set on a separate line. Any errors in the file format will be reported when the file selection is applied and the points are imported. Errors in the individual imported values will be reported during geometry/solution request creation or when the changes are applied. This allows the addition/removal or editing of specific points before application.

It is very important to note that CADFEKO does not maintain any form of linkage to files specified as the source for point coordinate definitions. During the import process, the values represented in the source file are simply used to automatically populate the relevant dialog fields. In order to reflect changes made in the source file, the point import process will have to be repeated.

4.4 Operations on geometry

Complex geometry is created from a sequence of operations starting with simple primitives. This sequence is represented in the tree (see section 3.3). It is possible to select and modify objects at any level in this tree, or change variables upon which the geometry depends. If parent items are selected, CADFEKO displays a temporary wire-frame representation of all selected items.

All items higher up in the geometry tree are re-evaluated each time an item is changed. During

this process CADFEKO may not be able to maintain the identities of some regions / faces / edges, for example, where multiple faces are derived from the same original face during Boolean operations. These items are then marked *suspect* (indicated by a question mark next to the representation of the items in the details tree — see section 3.3.1) as a warning that properties set may not have been maintained. The reason that an item has become suspect is displayed if the mouse pointer is moved over the item in the tree. After ensuring that the properties are correct or have been corrected, the suspect setting on the relevant regions / faces / edges may be reset using *Set not suspect* from the pop-up menu. Some solution configuration items (for example ports) that depend on the model will also be marked suspect if the model changes in such a way that they are no longer valid. These must also be checked and the *suspect* setting removed. It is advisable to resolve all suspect items before launching the FEKO solver or optimiser, as the loss of certain properties on the model geometry may change the electromagnetic problem description dramatically and have a large effect on computed results.

Any combination of objects or the variables that they depend on can be selected and modified. All operations are available from the *Construct* and *Transform* tabs.

The geometry modification operations are listed below.

Boolean operations *Union, Subtract and Intersect*

Operations on parts *Project, Imprint points, Split, Explode, Simplify, Stitch sheet parts*

Creation operations *Spin, Loft, Sweep, Path sweep*

Property operations *Reverse face normals* and *Properties* (which opens an edit dialog similar to the create dialog.)

Administrative operations *Rename(<F2>), Copy(<Ctrl+K>), Delete(<Del>)*

In addition to these operations, transformations (see section 4.5) (*Scale, Mirror, Rotate, Translate*) can be applied to geometry objects.

4.4.1 Boolean operations

The *Construct* tab provides access to the Boolean operations *Union, Subtract* and *Intersect*. The part/s to which the operation will apply must be selected before requesting a Boolean operation. The boolean operations are inactive when no parts are selected. (Surfaces and wire bodies are also considered parts.) Boolean operations cannot be applied to parents or the regions / faces / edges of parts.

When adjacent objects in a model are slightly misaligned, there may be faces and / or edges that have very small overlapping (supposedly separate objects) or non-overlapping (supposedly connected objects) sections as shown in Figure 4-7. If these sections are of the same order as the model tolerance, Boolean operations involving these objects may fail, or result in very small faces or gaps. The snap facility can be used to ensure that these items line up correctly. Similar problems can occur when an item is split very close to a boundary.

Note that it is now possible to add and remove items from an union and other boolean operators without having to copy the contents out to the root and recreating the boolean operation.

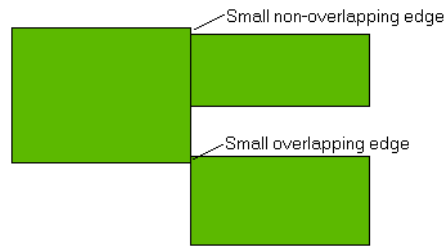


Figure 4-7: Slightly misaligned surfaces showing a short overlapping edge as well as a short non-overlapping edge.

The Union operation

Union combines all selected parts. In CADFEKO the *Union* operation is used to define connectivity between parts. Parts that touch, but are not unioned are considered not to be physically connected. When meshing, incorrect connectivity may result in invalid meshes (overlapping or misaligned meshes) that will generate errors during the FEKO solution.

The Intersect operation

For *Intersect*, the result is the common part of all the selected parts.

The Subtract operation

For *Subtract*, all the selected parts are subtracted from a final part, which must be selected when prompted for. When only one part is selected, the only Boolean operation that can be launched is *Subtract*. After the selected parts have been subtracted from the target part, the target part (the part which was subtracted from) will be indicated by a *T* in the tree.

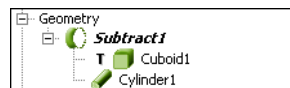


Figure 4-8: The *T* indicating the target part in the subtract operation.

Note that regions are taken into account during the subtract operation. Subtracting a shell structure (a free space region - implying an empty/hollow closed structure) may generate new regions, faces or edges on intersecting geometry, even though no geometry will be removed. Subtracting a solid structure (a region that is not set to free space) will result in geometry being removed from any intersecting parts.

4.4.2 Operations on parts

Projection and imprinting points

In some cases it is desirable to create specific points, edges or faces on a given geometry. For example, to provide accurate and valid attachment points for other structures on the model,

to manually force a finer mesh size along a curve or to create conducting patches on dielectric objects.

The *Imprint points* operation allows the placement of specified points on the selected part. Points can only be imprinted on one part at a time. The *Imprint points* dialog allows a list of points to be specified using standard point entry (see section 3.2.2) or based on a list of points imported from an external file (see section 3.2.2) in either global or local coordinates. The specified points are projected onto the closest point on the selected part — either on a face or an edge. Points may not be imprinted on top of existing points. The imprint operation creates a new entry in the tree (allowing access to the part without points), but uses the name of the parent object, as there is only one parent for this case. In CADFEKO 1 it was possible to specify vertices on the *Edge properties* dialog. This is no longer the case — such points must be created using the *Imprint points* operation.

Multiple parts can also be projected onto another part by selecting them, activating the *Project* operation and then selecting a target part to project onto. This sequence is similar to the subtract operation. All the projected parts appear as parents of the resulting part, i.e. they are no longer present as individual parts of the model. All of the edges of the selected parts are projected onto the faces of the target part. Any part (curves, surfaces and solids) can be projected onto any other part containing faces. (Spheres do not have edges, hence projecting a sphere has no effect other than removing it from the model.) Where the projected edges form closed paths, new faces will be created.

The projection direction is determined based on the normals of the faces of the target. Projecting edges onto convex curved target faces will thus tend to reduce their size and / or perspective. Convex surfaces may also “shadow” other surfaces in that all points on the projected edge may project onto the curved face even though it may seem as projection onto another face should also result. Edges are only projected onto the normal side of faces.

Edges where the projection crosses itself or turns back on itself are not allowed.

Split parts

Selecting the *Split* button on the *Construct* tab will open a dialog where the split plane can be specified by means of an origin, a plane (UV, UN or VN) and the angle of rotation around the axes.

The split operation creates two new parts (named *Split_back...* and *Split_front...*, where the *Front* side of the split is in the direction of the positive *n* axis of the split plane) for each selected part. The two halves of each split are derived from independent copies of the original part — if parent items are modified for one half, the other half remains unchanged and has to be modified separately. If synchronised changing is required, the original part must be created using variables. The split plane used in a split operation may be modified by selecting the new part and selecting *Properties* from the pop-up menu.

Exploding parts

Selecting *Explode* from the *Transform* tab will explode all selected geometry parts. Separate new surface parts are created for each face and free edge of the original parts. The new parts represent a snapshot of the geometry at the time it was exploded — they are not parametric.

Simplify tool (removing detail)

Redundant faces and edges may be deleted by selecting them and pressing the <Del> key or selecting *Delete* from the pop-up menu. Faces can only be redundant if they have the same medium (i.e. metal, free space or the same dielectric medium — see section 6.1) 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. Since the outside medium is free space, faces can only be removed from closed regions if the internal medium is set to *Free space*.

These items are permanently removed, but it is possible to copy the original object (see section 4.4.4) in which the faces/edges still exist out of the tree to restore them. If a component or variable is changed in such a way that the object is re-evaluated, suspect faces (see section 4.4) will not be removed and some deleted faces/edges may reappear.

Note that edges are not redundant if the normals of the faces on either side of the edge are in opposite directions. For this reason the front and back sides of faces are coloured differently in the *Default* view and it is possible to reverse the face normals (see section 4.4.4) of these faces. The normals of mesh triangle elements created in CADFEKO are in the same direction as the faces they originate from.

CADFEKO provides a tool for the automatic removal of redundant faces and edges. Select *Simplify* from the *Transform* tab to open the *Simplify geometry* dialog.

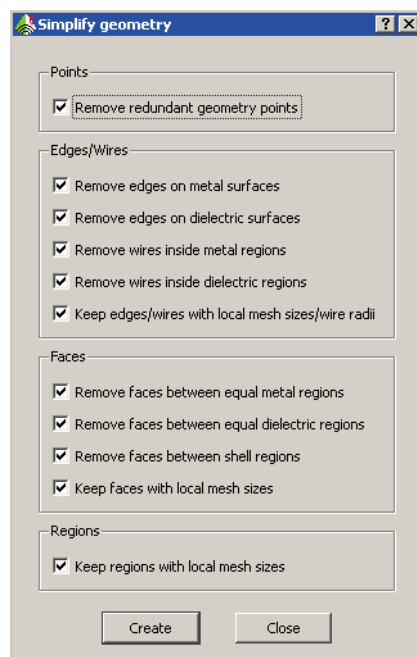


Figure 4-9: The simplify dialog

This operation allows removing specific types of items from the selected part/s. The simplified geometry will be electromagnetically the same as the original, but may not have the same meshing constraints. If, for example, an imprinted point is removed, there will no longer be a guaranteed mesh vertex at this location. (Note that imprinted points that are not attached to any edges are considered redundant.) Faces cannot be deleted unless the regions they separate can be merged. The same applies to edges on the boundaries of faces and geometry points at the ends of edges.

By default the simplify operation does not remove redundant regions, faces or edges on which local mesh properties (see section 4.8.4) are set. To remove them, the various *Keep ... with local*

mesh sizes options must be unchecked. For example, consider the union of two spheres of the same dielectric medium shown in Figure 4-10, case (a). A local mesh size is set on *Region1*. If the union is simplified with *Keep regions with local mesh sizes* checked, the result is as shown in case (b). Since the region contains local properties, it cannot be removed and the face between it and the centre region can also not be removed. If *Keep regions with local mesh sizes* is unchecked, the result is as shown in case (c).

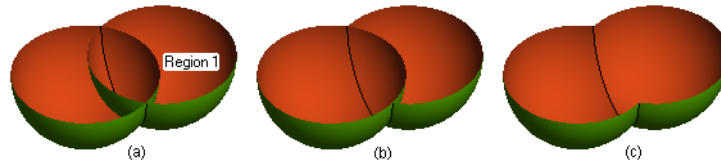


Figure 4-10: Illustrations of the simplify operation

Stitch sheet parts

It is possible that geometry that should be connected, is not, or there are small sections that overlap. This is most often due to problems encountered with tolerances during geometry import and should not be encountered for geometry created in CADFEKO. These geometry elements (sheet parts only) can be stitched together using the *Stitch* tool. Sheet parts that are within the specified tolerance is then considered to be connected and meshed correctly.

Note that the stitch tool can lead to a very strange (seemingly contradictory) display of geometry in CADFEKO. This happens due to the tolerance of some edges, faces and nodes being large (as specified in the *Stitch* tool) and thus they can be displayed anywhere within the tolerance area. The mesh will not have this strange display since the mesh elements have a very small tolerance.

In some circumstances, the *Stitch* tool may be used as a replacement for the *Union* operation. Generally the *Stitch* operation is faster and more efficient than the *Union* operation, but it may only be employed for connected sheet parts.

4.4.3 Creation operations

Spinning and sweeping parts

Selecting *Sweep*, *Path sweep* or *Spin* on the *Construct* tab sweeps the selected parts along a specified vector or path (these operations are also often referred to as ‘extrude’) or spins (rotates) them around a specified axis, respectively. *Spin* and *sweep* operations can only be applied to parts.

The sweep / spin operation is applied separately to each of the selected parts. If multiple parts are swept / spun, the resulting new parts have no defined relation to each other based on the spin / sweep operation, i.e. the defined operation is applied to each item independently and the spin / sweep parameters can be modified independently afterwards. The *Path sweep* tool cannot be applied to multiple parts.

Only parts that contain edges and / or faces (not solids or closed regions) may be spun / swept / path swept. 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 (for example, the T-plate union in Figure 4-11 is not an acceptable part for a spin / sweep / path sweep operation as the connecting edge borders three faces). Sweeping or spinning a curve results in a surface, while applying these operations to a surface results in a solid.

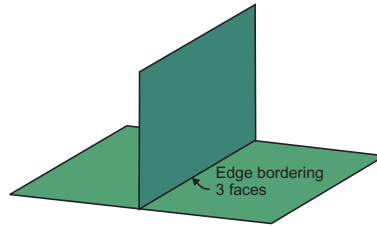


Figure 4-11: T-plate example of an edge bordering three faces so that it cannot be swept / spun

When the *Sweep* operation is selected, the *Sweep geometry* dialog allows specification of the sweep vector in terms of a start and end point (the path is taken as the straight line between these two points). These fields accept standard point entry (see section 3.2.2). It is not possible to sweep bodies in a direction that is tangential to any of its edges or in the plane of its faces.

The *Path sweep* operation is only available when a part that can be swept is selected. The *Path sweep* dialog is shown in Figure 4-12. The tool requests that the path for the sweep operation be selected. Any part consisting of only free edges and curves that form a joined, non-overlapping path may be chosen as a sweep path. The sweep may be adjusted by assigning a *Twist angle* (in degrees) and a *Scale factor*. These factors specify a scaling and rotation to be applied along the sweep path, beginning with unity scaling and 0 degree rotation and linearly increasing, ending at the specified angle and scale factor at the end of the sweep path. *Normal* or *Parallel* sweeps may be selected. The *Parallel* sweep maintains the orientation of the part during the sweep process, while a *Normal* sweep maintains the angle between the sweep object and the sweep path.

When a sweep path is selected, CADFEKO automatically determines the most appropriate sweep direction. This can, however be changed by selecting the *Start from other end of path* checkbox.

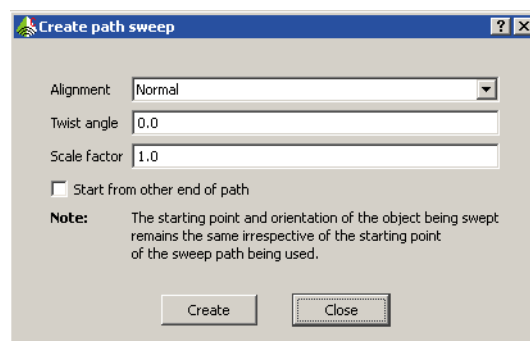


Figure 4-12: The *Path sweep* dialog

The *Spin geometry* dialog allows specification of the axis of rotation (an origin and a direction) and a rotation angle. The angle is taken in the mathematically positive sense around the specified axis and is specified in degrees. (In CADFEKO radian angles are only used in the arguments and results of the trigonometric functions and their inverses.) The *Set to ... axis* buttons allow quick selection of common axes. Note that the local axis buttons (*U*, *V*, *N*) are only available if all selected parts have the same local coordinate system.

The normal directions of the faces that result from a spin / sweep operation depend on the direction of the curves to which the operation is applied.

There are a number of restrictions on the relation between the part and the spin axis. No free edges may be coincident with the axis, nor are they allowed to intersect the axis at any point other than at the ends of the curve. For full spins (360°) of parts with faces, the spin axis may not intersect the face at only one point, unless it is at an end point of an edge. The axis may be coincident with any edge of the sheet, provided that the entire edge lies on the axis. In addition, no edge may touch the axis tangentially as shown in Figure 4-13 — not even if the edge is broken at this point. (Again, this is allowed IF the entire edge lies on the axis.)

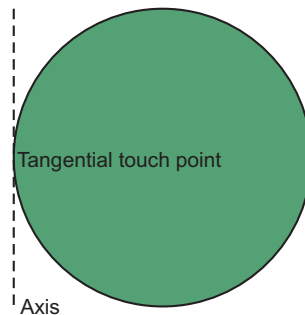


Figure 4-13: Example of a surface with an edge that just touches the axis of rotation

The *sweep* and *spin* operators may also be applied with an arbitrary orientation by making use of workplanes. This is set on the *Workplane* tab of the *Sweep* and *Spin* dialogs.

Loft surfaces

The *Loft* operation forms a smooth surface by connecting two curve parts with straight lines. (This is sometimes called a ruled surface, but it is not a faceted model — unless one or both curves are polylines). The curve bodies involved in the loft may consist of a number of edges, for example, a polyline. The two lofted curves must, however, have an equal number of edges (the loft can be considered to connect each pair of edges). If this is not the case, points can be imprinted (see section 4.4.2) on one of the curves to satisfy this requirement. Curves with more than two edges joining at one point cannot be used in a loft operation. While the loft tool is active, the 3D view preview shows how the curves will be connected. The *Loft* dialog allows reversing the start and end points of one curve in the case where the loft between the two curves is not indicated in the desired direction.

4.4.4 Administrative operations

Reversing face normals

Face normals can be reversed by selecting *Reverse normals* from the right-click pop-up menu in the details tree representation of the face/s or by choosing *Reverse normals* from the *Transform* tab. (The normals of all selected faces — in the tree and 3D view — are reversed, even though only one part's faces are shown in the details tree at any given time.)

Currently, it is not possible to reverse the normal on a body with a single, closed face, for example, a sphere. If such a normal must be reversed, the simplest option is to project a circular line onto

the body to create two open faces. The normals of these faces can then be reversed and the separating edge deleted or removed using the *Simplify* operation (see below).

Assemblies

Assemblies are used to organise the geometry and mesh parts. They become part of the geometry structure and are shown as the first level under *Geometry* in the tree. To create a new assembly, select the required items, right click in the tree or 3D view and select *Assembly* → *Create*. Only parts (displayed objects) can be added to an assembly. (Mesh and geometry parts cannot be added to the same assembly.) Items are moved between assemblies by selecting *Assembly* → *Move to* → ... in the pop-up menu. Similarly *Assembly* → *Move out* moves the selected items back to the root level in the tree. An item cannot be in two assemblies at the same time.

Deleting an assembly removes the entire assembly and its contents from the model. Similarly, selecting an item in an assembly and pressing <Del> will remove it from the model. (It is very different from *Assembly* → *Move out*.) Selecting *Disassemble* on an assembly moves all of the items inside it back to the root level and deletes the assembly — in effect removing the assembly without deleting the contents.

Operations (Boolean, transforms, splitting, etc.) are applied to the individual items. If multi-parent operations are applied to items in one assembly, the result is also in that assembly. Where such operations are applied to items from more than one assembly, the result is placed at the root level.

Assembly names are part of the full label / name, for example, *Assembly1.Union4.Face12*. This is the reference required in EDITFEKO to reference elements on this specific face. Note that all names in CADFEKO are case insensitive, i.e. *Face1* and *face1* are the same.

Copy objects

Geometry objects (including parents of objects that are shown in the tree) can be copied to new root-level parts by selecting the items and selecting *Copy* from the *Transform* tab (or using the short-cut <Ctrl><K>). For example, if a spherical shell is created by subtracting one sphere from another (which will remove both original spheres from the model and list them as parents of the new part created by the Boolean operation), the inner sphere can be copied to create a new root-level part which fits inside the shell. Copied items are created in the root level, even if the original items form part of an assembly. It is also possible to make multiple transformed copies of parts (see section 4.5).

The new part is completely independent of the existing one and neither will change if the other is modified. If it is desired that they change together, the original object should be created using variables as all field expressions used in the primitive definition are maintained during the copy operation.

It is also possible to select faces or edges of parts and select *Copy*. In this case new parts are made for each selected item. This allows, for example, an edge to be copied from a complex object and used to loft to another edge. Note that these copies are snapshots of the model when the copy is made — they are not linked to the parent object in any way and are not parametric.

Special copy options

In addition to the basic copy functionality, CADFEKO provides a number of special copy options. These special copy options are available on the *Transform* tab and *Copy special* → *Copy...* from the context menu.

The *Copy and translate*, *Copy and rotate* and *Copy and mirror* options allow the inclusion of a transformation in the copy operation.

The *Copy original* operation can be used to retrieve faces and edges that have been deleted from a part. The copy of a part generated using the *Copy original* operation will include all of these removed faces and edges.

Collapse - removing the creation history of a part

CADFEKO stores the entire creation history of every part allowing the user to modify any point in the creation history. While extremely powerful this requires a significant amount of memory and in some cases also a significant amount of processing time.

For example, if a small component of a union operation is modified, CADFEKO needs to recreate the parent parts in order 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 will never change. (It is, for example, unlikely that the model of a specific motor vehicle will be modified to another model, but it is common to place different antennas on such a vehicle.) Hence it should not be required to re-evaluate the vehicle each time a small part of the geometry is changed.

CADFEKO provides the option to convert a part to a primitive using the *Collapse* tool. This removes the entire creation history and stores the model as is. If this part is then used in other operations it does not have to be re-evaluated since the 'primitive' part is available immediately. This can save a significant amount of processing time and also reduce the possibility that edges / faces or regions cannot be mapped and will become suspect (see section 4.4).

The *Collapse* operation can be undone, but once the model is saved and reloaded the creation tree can only be accessed using a previously saved or archived copy (see section 3.4.5) of the model which still contains the tree information.

Re-evaluating the geometry

CADFEKO uses advanced mapping algorithms to keep track of individual items when geometry is modified. Since this information was not available in models created using CADFEKO 1, 1.1 or 2, the user is given the option to re-evaluate the geometry tree in order to generate this information when opening in a newer CADFEKO version. This re-evaluation option is automatically launched when an older model is loaded for the first time in CADFEKO, but the user may initiate a re-evaluation at any time by selecting *Re-evaluate* from the *Transform* tab.

Note that all the mapping information is not available in some models created in earlier CADFEKO versions, thus it may not be possible to map all items during the re-evaluation. These items are then marked suspect (see section 4.4). All properties (such as local mesh sizes) set on such

suspect items will be lost during re-evaluation. In addition, faces or edges that were deleted, may re-appear if they cannot be successfully mapped. The user therefore needs to evaluate his situation carefully. If the model geometry is likely to change in the future, it is recommended that the geometry be re-evaluated and that all suspect items resolved before making any changes to the model or setting any additional properties.

If the model geometry will not change, it may be better to keep the existing model and all its settings. Note, however, that this model should not be changed in any way that requires re-evaluation of the tree. For example if the model contains a motor vehicle with a monopole antenna unioned to it, changing the length of the monopole would trigger an automatic re-evaluation of the tree as re-execution of the union operation between the vehicle and the modified monopole is required to maintain the model consistency. (For memory reasons, CADFEKO does not store the entire model in each level of the tree.) Complex geometry in a CADFEKO model created in an early CADFEKO version that is not re-evaluated should in general be converted to a primitive form (see section 4.4.4).

Depending on the complexity of the model, this operation may be a quite lengthy operation. It is possible to cancel this operation, but the cancel process may also require a significant amount of time.

It is possible that during geometry re-evaluation, faults will be identified in the model that were not previously displayed. This is particularly true of models built in previous versions of CADFEKO, or models containing imported geometry. These faults identified during re-evaluation may provide additional information that may be useful in repairing a model.

4.5 Transformations

Geometry objects or mesh parts can be transformed by selecting them and selecting the transform from the *Transform* tab. The *Transforms* entry in the details tree contains a list (in creation order) of all the operations applied to each object. These may be edited (by double clicking on the specific operation) or deleted (by pressing the <Del> key). A transform may be applied to a selection of multiple objects, but it is in fact applied separately to each object. Therefore, changing the transformation afterwards requires selecting and editing the transform of each object separately. A transformation can, of course, be defined in terms of variables. In this case, changing the variable value will modify all the items to which the transform was applied accordingly.

The transforms now also make use of transforms which enable the application of the transforms with arbitrary orientation.

4.5.1 Transformations on geometry parts

Rotate

The *Rotate* operation requires an origin, axis direction, workplane and a rotation angle (in degrees). The angle is measured in a mathematically positive sense around the axis. The dialog, shown in Figure 4-14, also allows the modification of the workplane on the *Workplane* tab.

Translate

The *Translate* dialog, shown in Figure 4-14, requires a start and end point from which it calculates the translation distance and direction.

Mirror

The *Mirror* operation requires an origin and a plane. This is specified via the dialog shown in Figure 4-14.

Scale

The *Scale* transformation requires an origin (set in the global workplane) and a scale factor as shown in Figure 4-15. Scaling is done around the origin specified on the scaling dialog, not around the centre or origin of the object or the model.

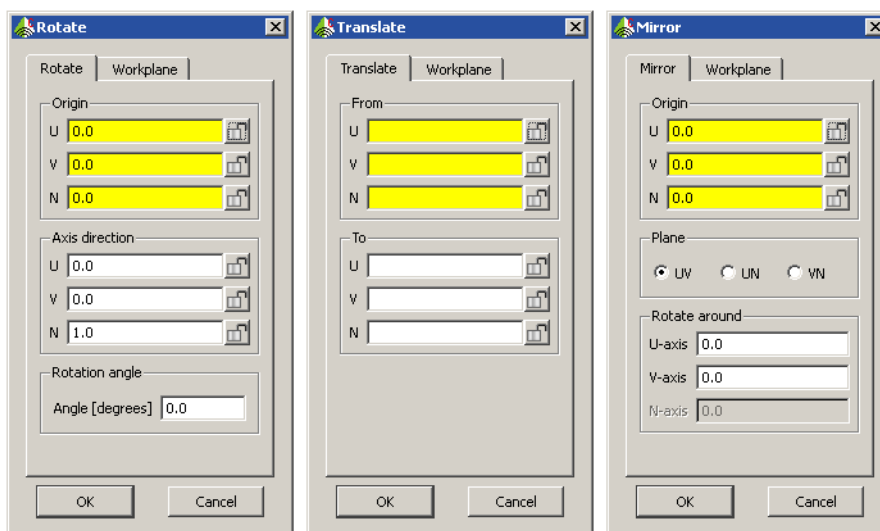


Figure 4-14: The rotate, translate and mirror dialog

Align

The *Align* dialog (shown in Figure 4-15) requires an origin and an axis direction for both the source and destination workplane. This transformation may be used to easily place objects on structures as it performs a rotate and a translate as a single transformation.

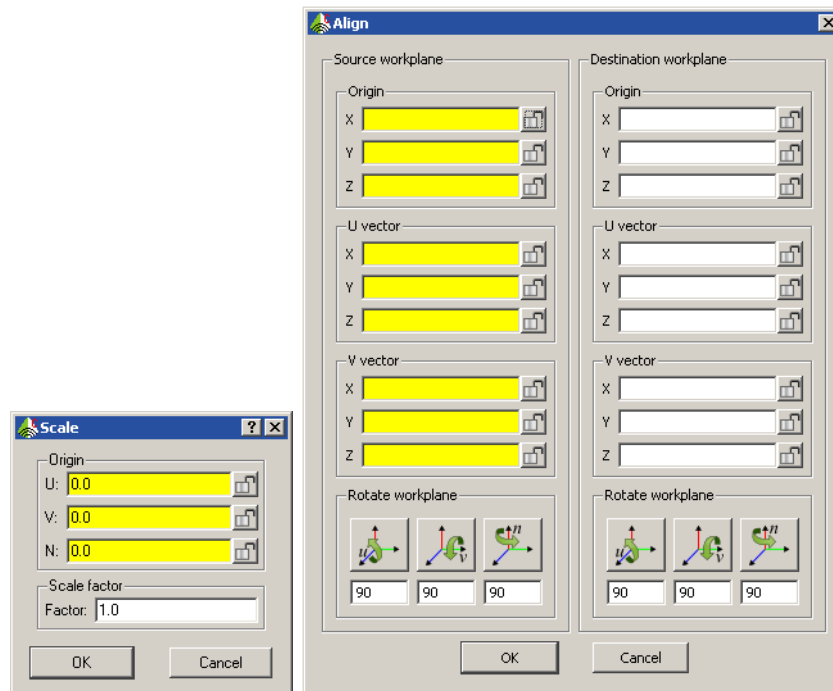


Figure 4-15: The scale and align dialog

4.5.2 Transformations on mesh parts

Mesh parts can be transformed in a similar manner to geometry parts. The transform applies to the entire part and may be edited or removed. If a vertex of a transformed mesh is edited, the transformed coordinates are displayed and user input is assumed to be in the transformed space. If the transform is modified, the effect of the modification is also applied to the edited vertex.

4.5.3 Multiple (transformed) copies

Parts (geometry or mesh) may be copied and transformed multiple times via a single command. Select *Copy special* from the pop-up menu or *Copy* from the *Transform* tab. A special transform dialog that contains a *Number of copies* field is launched. This will make the specified number of copies of each selected part.

If a rotation is specified, the n^{th} new part will be rotated through n times the specified rotation angle. For translation, the n^{th} new part will be translated by n times the specified translation distance. For the mirror operation only one copy is allowed. (A second copy would be back on the original and can be achieved with the standard *Copy* option.)

4.6 Cable modelling

Cables with arbitrary cable cross sections and shields may be included in the CADFEKO model. These user defined cables may also be excited or loaded directly and a calculation may be requested by adding a *Cable harness* request to the model.

The workflow for when defining cable modelling in CADFEKO is as follows:

- Define the cable type cross section (*Single conductor*, *Ribbon*, *Specified coax*, *Predefined coax* and *Cable bundle*). The defined cable type will be shown in the contents tree under *Cables*. If the cable is enclosed by a shield, a shield must be defined and its label added to the cable cross section definition. The defined shield will be shown in the contents tree under *Shields*.
- Define the cable path. The defined cable path will be shown in the contents tree under *Cables* and indicated in blue in the 3D view.
- Request a cable harness calculation by defining the solution method, whether the cable is irradiating/radiating and adding excitations and loads to the cable. The calculation request will be shown in the contents tree under *Solution*→*Requests*.

4.6.1 Cable cross sections

The following cable cross sections may be defined, *Single conductor*, *Ribbon*, *Specified coax*, *Predefined coax* and *Cable bundle*. These are available on the *Construct* tab, *Cables* group.



Figure 4-16: The *Cable cross sections* button

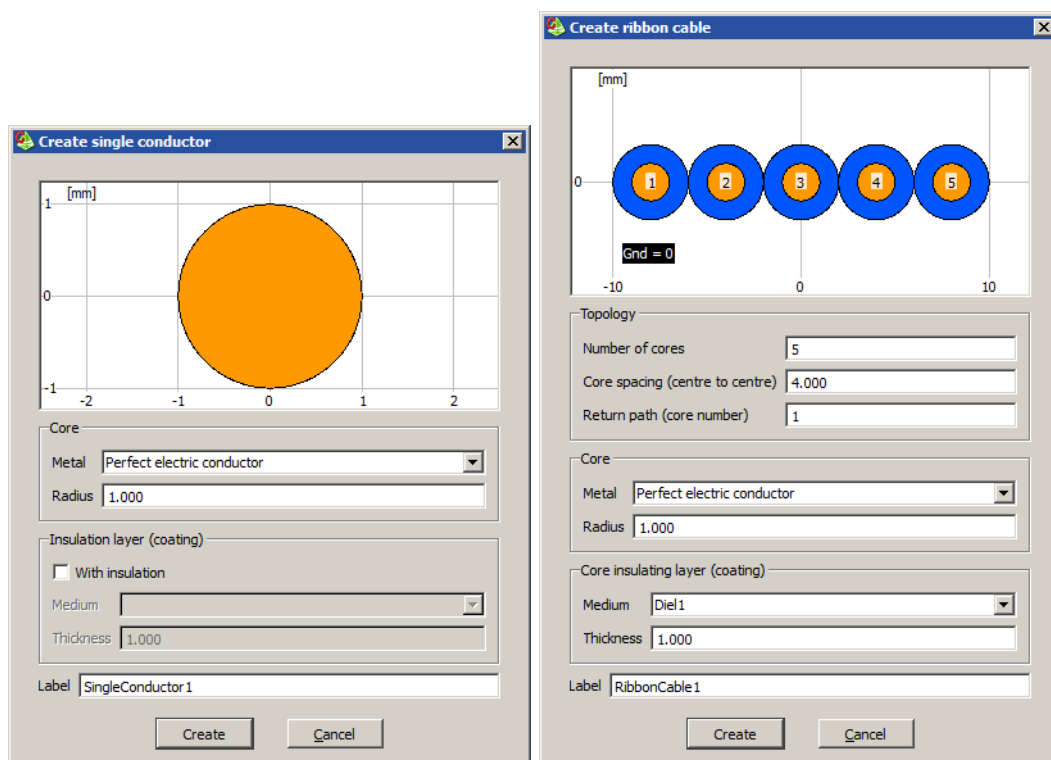


Figure 4-17: The *Create single conductor* and the *Create ribbon cable* dialog

Create single conductor: **Core** The core of the single conductor is specified by defining the metal of the core and its radius.

- Insulation layer (coating)** If the *With insulation* option is checked, the core is covered by a coating with the medium given by *Medium* and the thickness specified by *Thickness*.
- Create ribbon cable:**
- Topology** The ribbon cable topology is specified by the *Number of cores*, *Core spacing (Centre to centre)* and the *Return path (core number)*.
- Core** The core of a single conductor is specified by defining the metal of the core and its radius.
- Core insulating layer (coating)** A core insulating layer is specified by *Medium* and *Thickness*.
- Create coaxial cable:**
- Core** The core of the single conductor is specified by defining the metal of the core and its radius.
- Core insulating layers** Core insulating layers may be added by clicking on the *Add* button. The *Core insulating layer* is defined by the *Medium* and the *Thickness*.
- Shield** The shield enclosing the cable.
- Add a predefined coaxial cable:** A predefined cable type can be selected. Note that predefined cables are currently limited to frequencies from 10 kHz to 500 MHz.
- Create bundle:**
- Cables contained in bundle** A cable bundle may be created containing arbitrary cable cross sections. Defined cable cross sections may be added to the cable bundle by pressing on the *Add* button.
- Shield type** The following options are available *Not shielded, embedded in background medium*; *Not shielded, embedded in a dielectric* and *Shielded, dielectric filled*.
- Insulation medium** The insulation medium may be defined
- Outer radius** When the option *Compute automatically* is selected, the outer radius of the cable bundle is automatically calculated.
- Shield** The shield enclosing the coaxial cable.

4.6.2 Cable shield

The following cable shields may be defined in CADFEKO, *Braided (Kley)*, *Solid (Schelkunhoff)* and *Define shield properties*. These are available on the *Construct* tab, *Cables* group

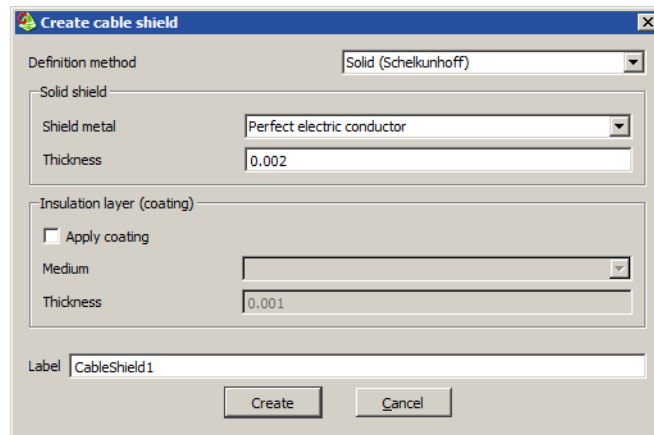


Figure 4-18: The *Cable shield* button

Braided (Kley)

Weave: The number of carriers in the weave and its weave angle must be specified by the user.

Insulation layer (coating): When the *Apply coating* option is selected, the cable is given a coating specified by *Medium* with a thickness specified by *Thickness*.

Figure 4-19: The *Create cable shield* dialog

Braid-fixing materials: When the option *Apply braid-fixing material* is selected, the *Inside braid-fixing material* may be specified. If a coating has not been specified, an *Outside braid-fixing material* may also be specified.

Filaments: The number of filaments, its diameter and the metallic medium to be used for the filaments are specified.

Solid (Schelkunhoff)

Solid shield: The *Shield metal* and the *Thickness* of the solid shield is required to be specified by the user.

Insulation layer (coating): When the *Apply coating* option is selected, the cable is given a coating specified by *Medium* with a thickness specified by *Thickness*.

Define shield properties

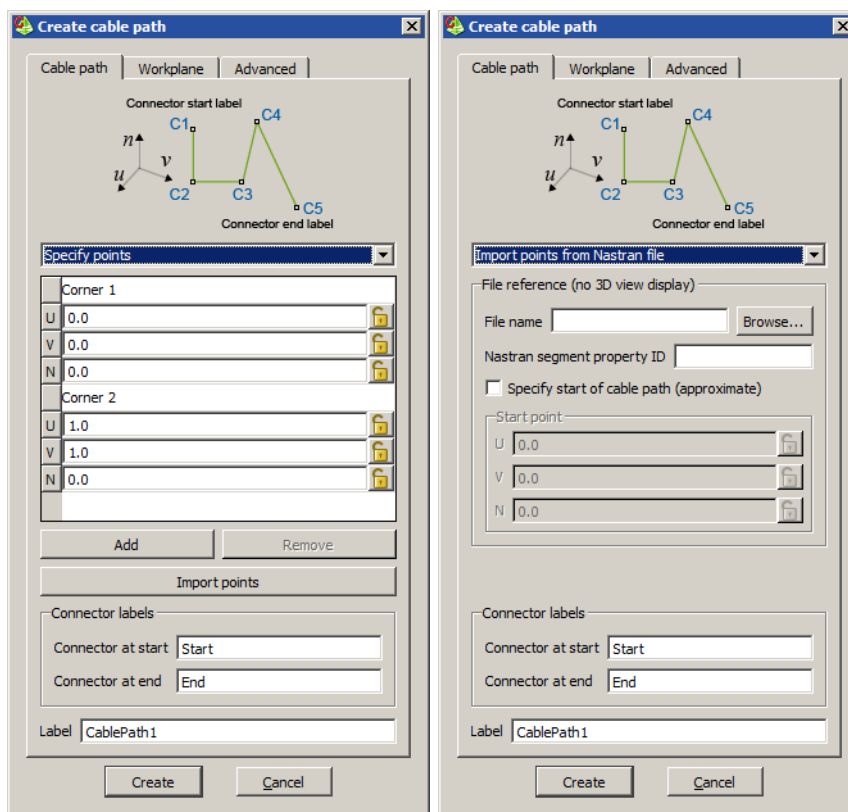
Solid shield: The *Shield metal* and the *Thickness* of the solid shield is required to be specified by the user.

Insulation layer (coating): When the *Apply coating* option is selected, the cable is given a coating specified by *Medium* with a thickness specified by *Thickness*.

Frequency dependent properties: The frequency dependent properties may be defined for the cable or loaded from a file. See the SH card (see section 14.58) for an XML example regarding the file format when importing cable properties from file.

4.6.3 Cable path

A cable path may be defined by specifying the points or importing the cable path from a file. To define a cable path, select the *Construct* tab, *Cables* group.

Figure 4-20: The *Cable path* buttonFigure 4-21: The *Create cable path* dialog

Cable path tab

The path of the cable is specified as a series of straight line sections. Corner points can be inserted or removed from the points list using the *Add* and *Remove* buttons. If point entry (see section 3.2.2) is used to specify the last point in the list a new point is added automatically. Point coordinates may also be imported from an external text file using the *Import points from Nastran file* option (see section 4.3.6). The cable path may cross itself (or other cable analysis paths), but there is no electrical contact at these locations.

Connector at start: The label to specify the start of the cable path. This connector label uniquely identifies the connection of a cable path section.

Connector at end: The label to specify the end of the cable path. This connector label uniquely identifies the connection of a cable path section.

Label: The label of the cable path.

Advanced tab

Sampling point density: The cable path section will be subdivided into small segments for the computation of the induced currents and voltages. The default sizes can be changed by selecting

Specify maximum separation distance and specifying a value for the *Maximum separation distance*.

4.6.4 Requesting a cable harness request

A cable harness request may be included in the CADFEKO model to calculate the coupling into a cable near a large conducting surface (see section 6.12.7).



Figure 4-22: The *Cable harness* button

The Cable tab

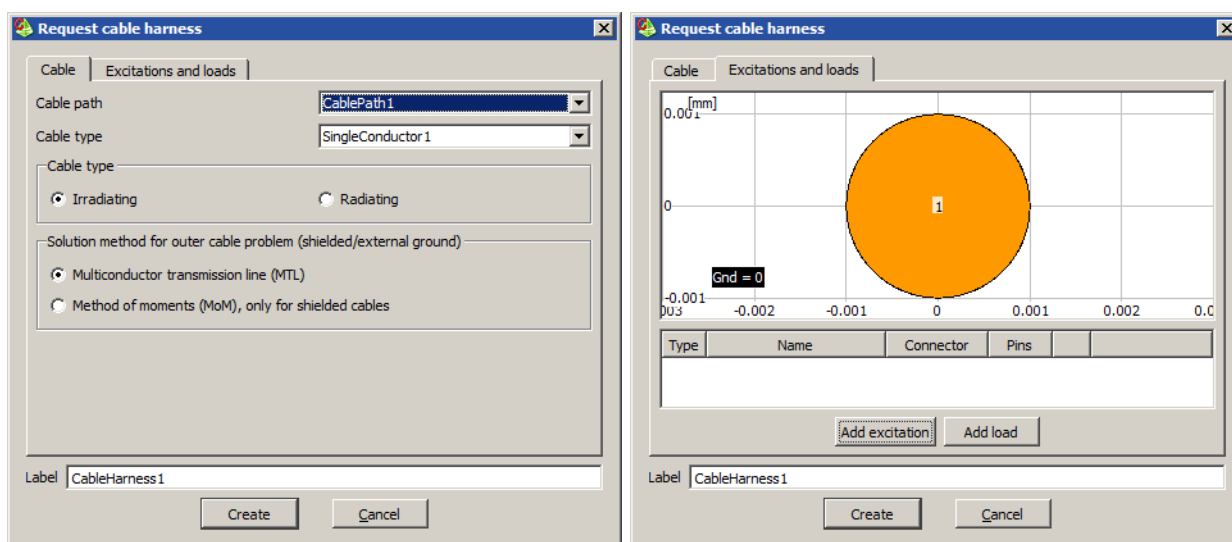


Figure 4-23: The *Request cable harness* dialog

Cable path and type: The labels of the defined cable path and the defined cable cross section type are required to be specified.

Cable type: Irradiating When this option is selected, only the effect of external fields coupling onto a cable will be considered

Radiating When this option is selected, the effect of the currents radiating in the cables will also be considered

Solution method for outer cable problem: Multiconductor transmission line (MTL) When this option is selected, the model will be solved with the multiconductor transmission line theory which is also hybridised with the 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 by means of the hybrid MoM/MTL solver. Any arbitrary cable path may be defined.

Label: The label of the cable harness request.

The *Excitations and loads* tab

Excitations and loads may be added to the cables configuration by clicking on the *Add excitation* and *Add load* button on the *Excitations and loads* tab on the *Request cable harness* dialog (shown in Figure 4-23).

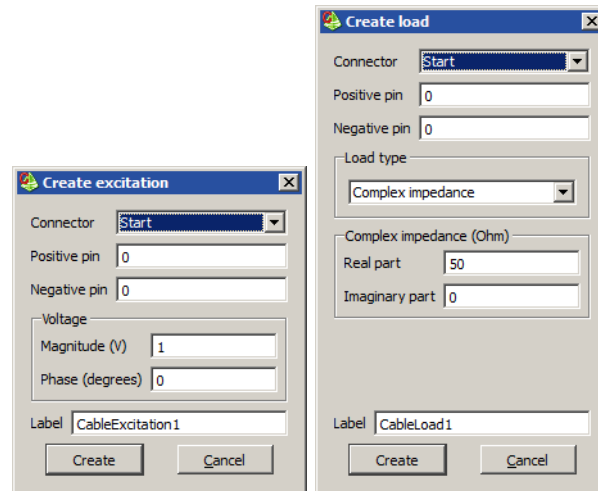


Figure 4-24: The *Create excitation* and *Create load* dialogs

Connector: The connector label uniquely identifies the connection of a cable path section. These connector labels are specified when a cable path is defined.

Positive/Negative pin: The positive and negative pins between which the excitation or load is to be placed.

Adding excitations or loads to a cable requires knowledge of the conductor to cable connector pin relation. See the AK card (see section 14.16) regarding the pin numbering.

4.7 Geometry validation

CADFEKO provides tools to perform basic consistency checks on the geometry. These tools may be applied globally (to all geometry in the model) or locally (only to the current selection — not available if the selection is empty).

Since properly connected meshes are only guaranteed for single parts, it is important to ensure that different parts do not clash. Parts clash if there is any contact between them or if one part is completely inside another. Select the *Mesh* tab and click on the *Clashing geometry* button.



Figure 4-25: The *Clashing geometry* button

Any parts containing clashing geometry will be selected and listed in the message window. *Find* → *Find in whole model* → *Unconnected geometry edges* and *Find* → *Find in selection* → *Unconnected geometry edges* finds and selects all the edges that are free (not attached to any face) or

attached to only one face within the indicated scope. If this operation selects an edge that seems to be connected to multiple faces, it means that there is more than one edge at this location and that the faces do not make electrical contact. This usually indicates a problem in the model.

4.8 Creating meshes

A mesh is a discretised representation of a geometry model. All geometry that is used during a simulation must first be meshed before a solution can be obtained. The accuracy of results depend greatly on generating a mesh that is appropriate for the problem being solved and as such, generating a correct mesh is very important.



Figure 4-26: The *Create mesh* button

The mesh dialog can be opened by clicking on the button on the *Mesh* tab on the *Create mesh* button (see Figure 4-26) or by pressing <Ctrl><M>. If all of the mesh settings have already been verified, the user can also run a *Quick mesh* that skips the dialog and performs the mesh without further dialogs or prompts. To run a *Quick mesh*, press <Ctrl><Shift><M>.

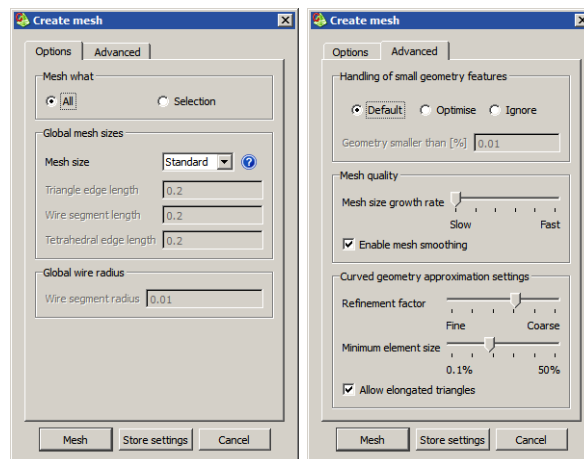


Figure 4-27: The *Create mesh* dialog

Figure 4-27 shows the *Create mesh* dialog. The various mesh settings can be set, including what must be meshed and how finely it must be meshed. Advanced features such as ignoring geometry and curvature refinement settings can be found on the *Advanced* tab. It is possible to indicate if all parts or only the currently selected parts in the geometry should be meshed. Unlike other dialogs, the selection can be changed while the dialog is open. This allows, for example, meshing different parts with different mesh sizes without closing the dialog.

The *Wire segment radius* field specifies the radius which is applied to all wires for which no local radius is specified. This value must be specified when wires are present irrespective of any other mesh settings.

4.8.1 Automatic meshing

Automatic meshing takes several model properties into account and generates a mesh that is appropriate for that configuration. Some local refinement may still be necessary (see section 4.8.4), but in most cases the automatic meshes will give a reasonable mesh. Meshes are generally built relative to the wavelength of an electromagnetic wave in the medium of propagation. Each solution method has different requirements and sometimes the model itself will influence the accuracy of results. The following model properties are considered:

Frequency: The shortest wavelength corresponds to the highest simulation frequency. Note then that the frequency range for the simulation must be set before automatic meshes can be generated.

Solution method: Depending on the solution method being used to solve the problem, different mesh requirements may be needed. For example, a FEM model requires settings for tetrahedra, a MoM solution requires settings for triangles and wires, while a hybrid solution needs to take both into account.

Dielectric properties: The dielectric properties of the media in the model will affect the propagation speed of an electromagnetic wave in a medium, which will in turn affect the wavelength. Dielectric media are taken into account in all cases *except* in the case where infinite layers are being used. In these cases, local refinement must be applied.

Geometry curvature: Even in cases where a finer mesh isn't required for accurate solution results, it may still be required to accurately model aspects of the geometry. Automatic meshes will attempt to reasonably conform to the original geometry, the settings of which can be modified on the *Advanced* tab.

Local mesh refinement settings may still be applied to individual components in a model. Any local mesh refinement settings will be respected, meaning that a user's setting will never be overwritten.

Wires

For wires, the wavelength (λ) is determined based on the maximum simulation frequency and the surrounding medium.

Table 4-4: Automatic meshing for wires.

Type	Fine	Normal	Coarse
	$\frac{1}{30}\lambda$	$\frac{1}{20}\lambda$	$\frac{1}{12}\lambda$

Faces and edges

If any of the bounding regions have a user-defined local mesh size, then the face will inherit the smallest of these local mesh sizes. If a face is the bounding face of a FEM region, then the first order automatic mesh size of that region will be used. If a face is the bounding face of a VEP region, then the mesh size of that region will be used. In all other cases the smallest wavelength for the bounding media is determined.

Table 4-5: Automatic meshing for faces and edges.

Type	Fine	Normal	Coarse
Method of Moments (MoM)	$\frac{1}{16}\lambda$	$\frac{1}{12}\lambda$	$\frac{1}{8}\lambda$
Physical Optics (PO)	$\frac{1}{10}\lambda$	$\frac{1}{8}\lambda$	$\frac{1}{6}\lambda$
Large Element PO (LE-PO)	λ	λ	λ
Geometric Optics (GO)	λ	λ	λ

Regions

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 tetrahedral edge lengths.

Table 4-6: Automatic meshing for regions.

Type	Fine	Normal	Coarse
Finite Element Method (1 st order FEM)	$\frac{1}{15}\lambda$	$\frac{1}{10}\lambda$	$\frac{1}{6}\lambda$
Finite Element Method (2 nd order FEM)	$\frac{1}{12}\lambda$	$\frac{1}{8}\lambda$	$\frac{1}{6}\lambda$
Volume Equivalence Principle (VEP)	$\frac{1}{16}\lambda$	$\frac{1}{12}\lambda$	$\frac{1}{8}\lambda$

4.8.2 Custom meshing

For finer control over the simulation mesh, the lengths of the edges of triangles and tetrahedra as well as the lengths of wire segments can be specified. This type of control can be used if the automatic meshing does not provide an appropriate mesh.

Triangle edge length: The *Triangle edge length* field specifies the default mesh size for the edges of triangles. Note that some edges may be as much as 30% longer than the specified length. (See section 4.11.1.)

Wire segment length: The *Wire segment length* field specifies the maximum length for wire segments. All edges that do not form the boundary of a face are assumed to be conducting wires and meshed into segments.

Tetrahedral edge length: The *Tetrahedral edge length* field specifies the default mesh size for the edges of tetrahedral. Note that some edges may be as much as 30% longer than the specified length. (See section 4.11.1.)

4.8.3 Advanced meshing options

The following options are available on the *Advanced* tab of the *Create mesh* dialog (shown in Figure 4-27).

Handling of small geometry features: This group allows special treatment of small geometry details. The *Geometry smaller than* field specifies the limit of what is considered a small feature as a percentage of the size of the part that it belongs to.

The *Default* behaviour is to mesh these structures normally. This will result in an accurate representation of the geometry, potentially using very small elements. *Optimise* is useful where the geometry has long narrow slivers or faces that are close together. If this option is selected, CADFEKO will try to improve the mesh quality of small features. Finally, it is possible to *Ignore* small features. Small details are then ignored at a possible cost of accurate geometry representation. This option may at times allow the meshing of faces that otherwise cannot be meshed with the default settings. Note that ignoring small features does not work for closed edges. Such edges can, however, be divided by imprinting points (see section 4.4.2).

Enable mesh smoothing: If *Enable mesh smoothing* is checked, an additional smoothing algorithm is applied. This will result in a better quality mesh, but will increase meshing time.

Mesh quality: The *Mesh size growth rate* controls how quickly the mesh size changes. *Fast* allows an abrupt jump from small to large elements, while for *Slow* each triangle will not be more than twice the size of any other triangle that it is connected to.

Curved geometry approximation settings: The *Refinement factor* indicates how closely a mesh must conform to the geometry. For geometries with many small details, a setting of *Fine* will require smaller triangles which will reduce simulation performance and perhaps even give errors. To limit the size of these elements, the *Minimum element size* can be set as a percentage of the average edge length on that part.

The option to allow elongated triangles means that the resulting mesh will make use of long, thin triangles where required. This could lead to a reduction in the number of mesh elements, depending on the type of geometry being meshed.

In general, the mesh size is not allowed to be smaller than the maximum coordinate divided by 1×10^8 . This is the limit of the numerical precision of the geometry. For very small models (which require correspondingly small mesh sizes), the geometry extents (see section 4.1.2) should be decreased. The mesh size is also not allowed to be larger than four times the extents. (For more information regarding the required and suggested mesh sizes for different mesh elements, see section 2.2.2.)

4.8.4 Specifying local mesh sizes

Sometimes an accurate solution requires a very fine mesh in some part of the model. Additionally, when meshing curved structures, a finer mesh may be required for an accurate representation of the geometry. Rather than mesh the entire model at this size, the user may specify a local mesh size for these parts.

The sizes specified on the *Create mesh* dialog are used on all items which do not have a local mesh size.

For regions, faces or edges, local mesh parameters can be specified in the *Mesh size* group on the respective *Properties* dialogs. This group contains a *Local mesh size* field which must be checked if a local mesh size is to be set. An icon will be displayed next to all items in the details tree on which local mesh parameters have been set to indicate this.

If a region that has a local mesh size set on it is meshed into tetrahedra, that size will also be applicable to the bordering faces. An additional finer mesh size can, of course, be specified on these faces. (The minimum of all applicable local mesh sizes are used.) Likewise, setting a mesh size on a face also sets that size on its bounding edges. If a finer mesh size is specified on, for example, an edge of a face, then the triangles of the face will adhere to this length along the edge, even though the rest of the face may have a much larger mesh size.

The final mesh size on any item is the minimum of all local sizes applicable to it. As an extreme example, if an edge borders a face which is the boundary of a region and all three items have a local mesh size, the final mesh size on the edge is the minimum of all three these sizes. On faces, the local mesh size may be larger than the global mesh size. Note, however, that if no local mesh size is specified on the bordering edges, the global mesh size applies to them. The triangles on the edge of the surface will then have the global mesh size. For regions, the local mesh size is only used when meshing the region into tetrahedra. If the local mesh size on a region is set, then volume meshing of that region still will occur at the smaller of the specified local and global mesh sizes.

For wires, a local wire radius may be specified on the *Edge properties* dialog. Note that wires are displayed as lines in the 3D view and that the radius is not represented. The radius for meshed segments are displayed by default. An icon next to the edge in the details tree indicates that a local wire radius has been set.

Multiple selections

When launched, the properties dialog shows the current state of the properties of the selected items. If some selected items have a local mesh size and others not, or if the specified size is different, the *Local mesh size* item is in a ternary (i.e. not defined) state. (Thus the field is referred to as a tri-state field.) If this is the case, the parameters of the different elements in the selection are not updated when clicking *OK* or *Apply*. This field can, however, still be checked or unchecked to set (override) the properties on all selected items. Note that it is possible to cycle through all three states by clicking on the field.

4.8.5 Editing mesh vertices



Figure 4-28: The *Mesh vertex selection* button

A mesh vertex can be modified by selecting the vertex on the unlinked simulation mesh (mesh vertex must be visible and mesh vertex selection on). Right-click and select *Properties* from the popup menu. This opens the *Modify vertex* dialog which contains the coordinates of the vertex (in global coordinates). The input fields accept point entry (see section 3.2.2) and expressions, but these are converted to numerical values once the dialog is applied and closed. The vertex point can be moved by a given amount (rather than to a specific position) by adding the required offset to each field. For example, changing *X* from 1.2 to 1.2+0.5 moves the point 0.5 units in the *x* direction.

4.8.6 Creating mesh triangles

Sometimes manual mesh fixing is required. This may be because the mesh contains a number of holes or because some bad elements were selected and deleted. Unlike deleting vertices (see section 4.10.4), deleting elements leaves a hole in the mesh surface.

For this reason, mesh triangles can be manually constructed by selecting the *Mesh* tab and clicking on the *Create triangle* button.



Figure 4-29: The *Create triangle* button

This operation is only allowed if the selection contains a single mesh label (in which case the new element is added to this label), or a single mesh part (in which case the new element is added to a new label created under this part). The *Create mesh triangle* dialog allows specification of the three corners of the triangle. Since these three fields are standard point entry (see section 3.2.2) fields, all the different snap options (see section 3.2.1) are available.

Manually constructed triangles may be “moved” into an existing label. Select all the elements of the existing label as well as all the new elements. (If the elements were selected by label, click the *Select mesh element* button to convert the selection to individual elements.) Now right click on the selected elements and select *Rename*. This will merge all the selected elements to one label which can then be renamed. If all elements of a label are renamed, that label is removed — the original label name can then be used for the newly created label. Alternatively, mesh labels may be selected, and the *Merge* tool applied from the right-click menu. This will merge all of the selected mesh elements into a single mesh label within the same mesh part.

4.8.7 Reversing normals

Some features in FEKO make use of the normal vectors of elements. For example, dielectrics are specified by giving the medium on each side of the bordering triangles. For this purpose, the normal side of a triangle or polygon is defined as the side towards which the normal vector is pointing. (If the start point of the normal vector is located on the triangle, the normal side is the one closest to the arrowhead.) In CADFEKO the normal sides of mesh elements are coloured blue, while the backs (away from the normal) are brown when the *Face normal medium* display option is selected (*Display options*, *Colour* menu button).

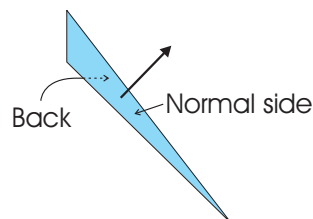


Figure 4-30: The normal and back sides of a triangle

When the selection contains triangles or UTD polygonal plates (or labels or parts that contain these elements), selecting the *Transform* tab and clicking on the *Reverse normals* button will invert the normals of all the selected triangles / polygons.

Figure 4-31: The *Reverse normals* button

Normal vectors can be shown — as lines with a red tip — per 3D view by activating the *Show / hide normals* button on the 3D view toolbar.

Reversing the normals of mesh triangles on dielectric boundaries can adversely affect the consistency of the model. CADFEKO tries to keep the model consistent. For example, if the normals of all triangles in a given label are reversed, CADFEKO swaps the dielectric media on the two sides. However, if only some triangles are modified, CADFEKO cannot ensure consistency and it is up to the user to ensure that the model remains consistent.

4.9 Mesh refinement

CADFEKO provides the functionality for mesh refinement around a point and a polyline. Note that display of these meshing rules may be enabled/disabled by clicking on the *Display options* context tab and clicking on the *Meshing rules* button.

Figure 4-32: The *Meshing rule* button

4.9.1 Point refinement

If a point refinement is required in the mesh, select the *Mesh* tab and click on the *Point refinement* button.

Figure 4-33: The *Point refinement* button

4.9.2 Polyline refinement

If polyline refinement is required in the mesh, select the *Mesh* tab and click on the *Point refinement* button.

Figure 4-34: The *Polyline refinement* button

4.10 Mesh manipulation

4.10.1 Mesh element radius

If the selection contains individual segments or mesh labels containing segments, the segment radii can be changed by right clicking on the selection and selecting *Properties*. If the selected items have different values, the *Radius* field will initially be blank. This is the ternary state — other items on this dialog can then be changed without modifying the radii of all the selected elements. If a value is entered in this field, the radius of all selected items will be set to this value.

Mesh segments can be displayed with their actual radius in the individual 3D views. Click the down arrow next to the *Show/hide mesh* icon on the 3D view toolbar (see section 3.2.4) and ensure that *Show segment radii* is checked. (Selecting this item when it is checked will uncheck it.) A representation of coatings that have been applied to mesh segments can also be shown by activating the *Show segment coating* option.

When changing the mesh viewing mode to the *Colour by face medium* mode (see section 3.2.3), the radii of all visible mesh segments is automatically shown (the *Show segment radii* option is automatically activated). This display may be manually deactivated. In a similar manner, when changing the mesh viewing mode to the *Colour by coating* mode, the coating of all visible mesh segments is automatically shown.

4.10.2 Relabelling mesh elements

Any selection of mesh elements can be relabelled provided they are of the same type and belong to the same mesh part. Since the full label is constructed as *Assembly.Part.Label*, it is not possible to relabel elements on different parts. Note that the individual elements represented in the 3D view, not the labels in the tree, must be selected. (If the selection is done by mesh part or mesh label, switching to element selection automatically selects all elements belonging to the selected parts / label — see section 3.2.2). If the original elements did not have the same settings (media, using PO, etc.) the new label is marked suspect as CADFEKO cannot automatically determine which of the settings to use.

Multiple mesh labels can also be merged (grouped under the same name) by simply selecting the mesh elements in the details tree and clicking on *Merge* in the right-click pop-up menu. Merging of elements is also only possible within in a single mesh part.

Labels are used to reference specific elements when setting certain properties. To specify unique properties on an element or collection of elements, those elements need to be relabelled. As an example, consider a simple wire monopole attached to a plate, created as *Union1*. The resulting mesh contains two labels: *Face1* for the triangles on the plate and *Wire5* for the segments on the monopole. If, for example, only the bottom half of the dipole is coated with a dielectric, only these segments must be relabelled before applying the coating. (A better approach to this example would be to split the original wire — by imprinting a point — and apply the coating to the resulting lower edge.)

When the last element with any label is removed (deleted or relabelled), the label is removed. Thus if a selection containing all elements from more than one label is renamed, the total number of labels is reduced.

Care must be taken when remeshing parts where the mesh has been relabelled. Since the entire mesh part is replaced by the new mesh, the label created by relabelling the elements is lost. If the example above is remeshed, there will again be only the labels *Face1* and *Wire5*. Therefore, relabelling elements should be delayed till as late as possible in the model preparation process.

4.10.3 Merging coincident vertices

Connectivity in FEKO relies on vertices of adjacent mesh elements being within a small tolerance of each other. CADFEKO allows merging all vertices within a user-specified tolerance. To do this, select one or more mesh parts and select the *Mesh* tab and click on *Merge vertices* to open the *Merge coincident vertices* dialog.



Figure 4-35: The *Merge vertices* button

Here a *Tolerance* can be specified. Any two points separated by less than this distance are then merged to the coordinates of one of the original vertices (as opposed to taking the average position) resulting in a single vertex.

If the *Snap to geometry points* or *Snap to named points* fields are checked, mesh vertices lying within the specified tolerance of these points will be snapped to them. For example, if a named point lies between two mesh vertices that are less than the specified tolerance from each other they will be merged at this point. If the snap options are checked, vertices lying within the tolerance from geometry or named points will be included in the number of merged vertices listed in the message window — even if they originally coincided exactly with the points.

Merging points can lead to degenerate (collapsed) triangles. CADFEKO tries to avoid this by giving a warning if the tolerance is large relative to the mesh sizes. It also automatically removes all degenerate elements after merging the vertices.

4.10.4 Merging elements (removing vertices)

The automatic mesh removal is meant primarily to merge vertices that should be the same, but differ due to numeric tolerances. If the points are further apart, it may be required to control which point is moved and which point remains fixed. It is also often required to remove specific elements where the distance between points may be large compared to the triangle size elsewhere in the model.

This is done by deleting vertices. This is a two-step process, similar to the subtract operation. First select all the vertices which must be removed and press <Del>. In the second step, a final vertex must be selected when prompted. All the originally selected vertices are then merged with the final vertex. CADFEKO automatically removes all elements that become collapsed (degenerate) during this operation.

4.10.5 Merging mesh parts (union for meshes)

When mesh parts have been imported as separate mesh parts, it is not possible investigate mesh connectivity or add edge port between these parts. Meshes can be merged into a single mesh by selecting the mesh parts in the model tree and selecting the *Merge* option from the context menu (right-click on the mesh parts) or merge the meshes by selecting the *Mesh* tab and clicking on *Merge meshes*.



Figure 4-36: The *Merge meshes* button

4.10.6 Remove duplicate and collapsed elements

Duplicate elements

Duplicate elements within a mesh part can be deleted automatically. Select the required mesh parts and select the *Mesh* tab and click on the *Remove duplicates* button. (Generally duplicate elements should only occur in imported meshes or where CADFEKO meshes were edited manually.)



Figure 4-37: The *Remove duplicates* button

If duplicate elements have the same label, CADFEKO deletes all but one. If the elements do not have the same label, it may be important to delete a specific element. This is controlled with the *Remove duplicate mesh elements* dialog shown in Figure 4-38. The items are ordered by pressing <Ctrl> and dragging the numbers in the left-hand column. For each set of duplicates an element whose label is highest on this list is retained and all others are deleted. (There could be more than two identical elements.)

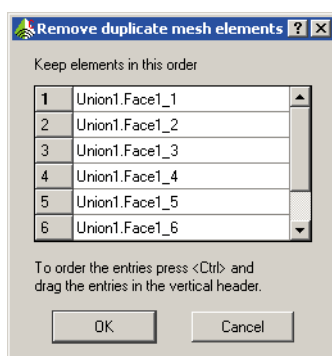


Figure 4-38: The remove duplicate mesh elements dialog.

Collapsed elements

Selecting the *Mesh* tab and clicking on the *Remove collapsed* button will delete all degenerate elements (elements where two or more nodes coincide).



Figure 4-39: The *Remove collapsed* button

4.11 Mesh validation

4.11.1 Mesh information

Selecting *Mesh* tab and clicking on the *Info* button shows a histogram of the edge length distribution (as well as the average edge length and the standard deviation of the edge lengths) and the number of the different elements in the current mesh selection. (The selection may contain mesh parts, labels or elements.)



Figure 4-40: The *Info* button

The mesh info can also be displayed by right-clicking on *Meshes* in the tree and selecting *Info* from the pop-up menu. Alternatively, the mesh info can be displayed by double clicking on the meshed model in the tree. The spread gives an indication of the quality of the mesh and shows how many edges are longer than a desired length. The numbers of the different elements determine the problem size and influence the required run time.

4.11.2 Finding problems in the mesh

Since the accuracy of the FEKO results depends on the quality of the mesh, CADFEKO provides some mesh validation tools. These tools may be applied globally (to all mesh parts in the model) or locally (only to the currently selected mesh parts - only available if the current selection is not empty).

4.11.3 View Distorted and Oversized mesh elements

Selecting the *Mesh* model and clicking on the *Distorted elements* button checks the current selection (mesh parts, labels or elements) for distorted elements.

After the operation is applied, the identified inappropriate elements are selected and listed in the message window. The selection *Undo* and *Redo* operations can be used to see which elements were selected before and after the test.

Figure 4-41: The *Distorted elements* button

Distorted mesh elements are specified in terms of the minimum internal angle. In an ideal mesh all internal angles are 60° (all angles in CADFEKO — except the arguments of trigonometric functions used in expressions — are in degrees). If any of the three angles in a mesh element are very small, the element is a sliver. Such elements can be removed by deleting vertices (see section 4.10.4).

Figure 4-42: The *Oversized elements* button

Oversized elements are found based on the maximum edge length.

4.11.4 View free mesh edges and segment nodes

Another very important issue regarding mesh quality is connectivity. In addition to the tests on the geometry, CADFEKO provides tools for finding unconnected edges and segment vertices. This is done by selecting *Find* → *Find in whole model/Find in selection* → *Free mesh triangle edges* or *Find* → *Find in whole model/Find in selection* → *Free segment nodes* from the main menu. These operations test only the selected elements, and will not report free edges / vertices if the element is connected to another unselected element.

Triangle edges are considered to be free if they are not attached to other triangles. (An edge is only connected to a triangle if both end points coincide with corner points of the triangle.) Here the two vertices at the ends of the edge are selected. These vertices may then be merged, moved or modified as required to make the model consistent. Again, free edges may be perfectly normal such as on the edge of a finite boundary.

Using the *Find* → *Find in whole model/Find in selection* → *Free segment nodes* tool, all vertices which are not attached to another segment or a triangle corner within the indicated scope are selected. These vertices may be valid, for example, at the ends of wires. However, if two such points lie very close together, they should be merged.

4.11.5 Find intersecting mesh triangles

Intersecting mesh triangles indicates all triangles that intersect, but are not connected. This is useful for identifying areas of poor mesh quality - particularly in imported meshes. This tool highlights all mesh elements that overlap (entirely or partially) or cut each-other and are not connected at the point of intersection. After the tool is applied, all parts that contain intersecting mesh elements are highlighted in the tree, and the individual problematic mesh elements are highlighted (shown in yellow) in the 3D view. The selection may be deleted, but typically each intersection should be considered and repaired according to the intended mesh representation.



Figure 4-43: The *Intersecting triangles* button

5 Importing and exporting

5.1 Importing and exporting geometry

There are a number of ways to import and export geometry in CAFEKO. CAFEKO can import geometry from other CAFEKO models, Parasolid and other CAD formats. Importing and exporting CAFEKO and Parasolid models do not require any special licences to be activated, but translation to and from other formats does require extra licences.

5.1.1 Importing CAFEKO models

CAFEKO is able to import other CAFEKO models. The import allows the user to select whether geometry, meshes, solution entities or optimisation settings should be included in the import process or not. Click on the *Application menu* button and hover over the *Import* menu anchor.



From the *Application menu* select *Import* → *CAFEKO model (*.cfx)*.

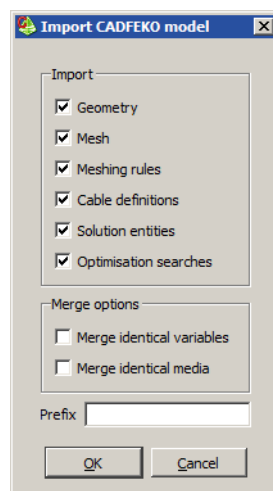


Figure 5-1: The import CAFEKO model dialog.

Variables, named points and media are always imported. An option is provided to merge variables and media that are shared between the model being imported and the destination model (i.e. variables and named points with the same name and value in both models). If there are naming conflicts between names of imported entities and existing entities in the model (variables, named points, geometry, meshes or solution entities), an error message will list the offending entities and the import will be terminated. This problem can be overcome by adding an optional prefix (in the *Prefix* field) that will be pre-pended to all imported entity names in order to ensure uniqueness. When variables or media are merged, variables and media with the same name and the same value result in a single instance of that variable or media after importing.

It should be noted that the frequency, infinite planes and various other settings (e.g. meshing settings in the *Create meshing* dialog) are not considered during the import process and will not affect the destination model in any way.

5.1.2 Working with Parasolid

CADFEKO is based on the Parasolid solid modelling kernel and therefore models can be imported and exported from/to the native Parasolid format without any translation.

Import of Parasolid models

Parasolid models can also be imported into CADFEKO. Click on the *Application menu* button and hover over the *Import menu* anchor. From the *Application menu*, select *Geometry → Parasolid*. This asks for a file name and then opens an *Import Parasolid model* dialog where the scale factor to be applied during importing can be modified. This *Scale by* factor is the factor by which the Parasolid model is scaled before it is imported. If the Parasolid model was created by exporting from CADFEKO and the *.info file is available, the dialog will display the value in the file. This value can then be changed. For example, if an antenna model was constructed in mm and is to be imported into an automobile model constructed in m, a scale factor of 0.001 would be appropriate (independent of any scale factor required by the definition of the model extents). Note that importing a large model with a scale factor other than unity may be a very slow process.

Isolated vertices (acorns) in a Parasolid model are not the same as named points (see section 4.2.2) in CADFEKO. These vertices are not imported, but CADFEKO does write the coordinates to the message window during the import process. If required, the relevant named points can then be created manually.

Export of Parasolid models

Click on the *Application menu* button and hover over the *Export menu* anchor. From the *Application menu* select *Geometry → Parasolid*. This opens the *Export Parasolid model* dialog, where the option of *Text* or *Binary* format can be set and the Parasolid version to export to can be chosen. Once the export options have been chosen, a dialog prompting for the name of the file to export to will be launched. CADFEKO exports the latest Parasolid version by default, but older versions of Parasolid can also be selected. Only the final geometry is exported. Exporting and importing the same model will result in the loss of the entire creation tree (this is similar to the make primitive (see section 4.4.4) operation).

Parasolid models are inherently limited to a 1000x1000x1000 unit box centred at the origin. CADFEKO introduces a scaling factor to make this more flexible (see section 4.1.2). The *Scale factor* field on the *Export Parasolid model* dialog displays 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. In order 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. If, however, the model is smaller 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.

5.1.3 General CAD formats

CADFEKO can also import other CAD formats using the Spatial InterOp translators. Currently ACIS, CATIA V4, CATIA V5, AutoCAD, IGES, Parasolid, Pro/ENGINEER, STEP and Unigraphics are supported import formats. The following are supported CAD export formats using the Spatial InterOp translators: ACIS, CATIA V4, CATIA V5, IGES, Parasolid and STEP. AutoCAD DXF files can also be imported (some advanced DXF options are not supported).

Note that all CAD import translator modules are licenced separately. If some import/export menu items are disabled, these modules are not activated in the licence. If you wish to evaluate or purchase any of these import/export modules, please contact your distributor.

Importing general CAD

From the *Application menu*, select *Geometry*. If an import menu item is selected, CADFEKO requests the source file name and then opens the *Import ... model* dialog. The *Units* group on this dialog displays the units of the imported model and allows setting the *Destination* units. The destination unit determines a scale factor and may be set different to the unit of the current model to realise the required scaling effect. If CADFEKO cannot determine the unit of the imported model it assumes a default value. It may be required to select a different unit to make the imported model fit inside the Parasolid size limit discussed regarding the import of Parasolid geometries.

Since all imported CAD models are converted to a Parasolid format during the import process, all the comments for Parasolid imports also apply to imports from general CAD formats. Due to the differences in the internal representations used by different CAD formats, format conversion can have unexpected results. In particular, adjoining surfaces sometimes do not line up correctly due to tolerance differences and models that use a numerical representations can cause faults during scaling. Though CADFEKO does provide some basic diagnosis tools, no advanced CAD fixing functionality is available natively in CADFEKO.

All model format conversions performed during a CAD import is logged in the file `CADimport.log` in the directory `%FEKO_USER_HOME%/logs`. The information in this file is useful in cases where the import conversion fails.

Importing AutoCAD DXF

AutoCAD DXF file can also be imported (no export). After selecting the file to import, a dialog displays the contents of the file and various other options. It is possible to select only selective components from the file by unchecking the check box of the unwanted geometry. The *Advanced*

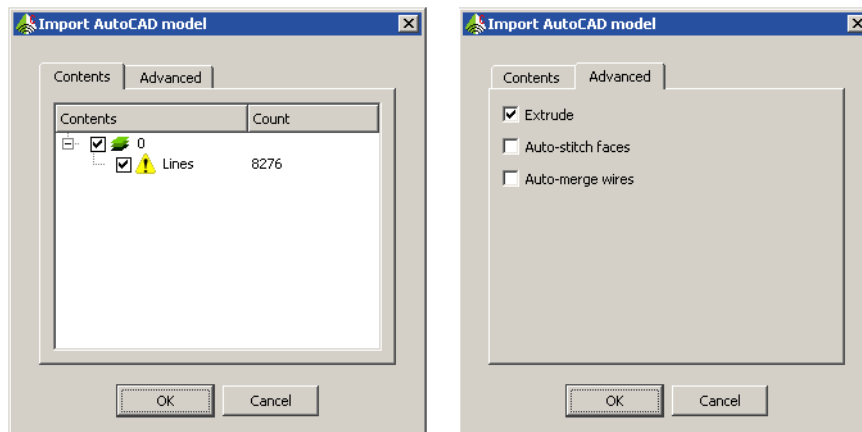


Figure 5-2: AutoCAD DXF import dialog (Contents and Advanced tab)

tab contains special options to extrude, to stitch faces together and the auto-merging of wires during the import process.

Note that not all DXF entities are currently supported. The supported entities are:

- 3D face
- Arc
- Circle
- Ellipse
- Line
- Polyline and polyface mesh (3D)

Entities that are not supported are:

- Point
- Spline
- 3D solid
- Trace
- Hatch
- Text
- Dimensional annotations

Exporting general CAD formats

Click on the *Application menu* and hover over the *Export* menu anchor. From the *Application menu* select *Geometry*. If an export format is selected, the options and information available for the particular format is displayed. Clicking on *OK* allows the user to select the file to export to.

Scaling is very important when translating between the different CAD file formats (ACIS, Catia, IGES, Pro Engineer, STEP, Unigraphics, Parasolid) and this is often the source of many importing errors. During an export a dialog displays the difference in the number of units displayed in CADFEKO and the number of units that would be seen in the program where the file is imported. The difference in the number of units is caused by the fact that CADFEKO does not perform any scaling during the export (scaling could cause tolerance errors during the subsequent import process). During an import, CADFEKO also displays the scaling that will be applied during the import process - the scaling can be changed by changing the CADFEKO model unit or geometry extents. If a model does not import as expected the scaling should be varied in an attempt to import the geometry correctly.

5.1.4 Faults on geometry

It is important that the imported model is of a decent quality. If the imported model has overlapping surfaces and/or gaps between supposedly connected surfaces, these will not be corrected during the import process and the resulting CADFEKO model will also be of poor quality. Refining such models may be a considerable task after importing. CADFEKO will try to import geometry with faults and also indicate what the fault is. (A tool-tip with details regarding the fault will be shown if the mouse pointer is held over the fault icon.) The user should consider each fault and try to correct the model as far as possible. In order to locate specific faults in the geometry, the entity of interest may be selected in the tree or in the details tree, and the zoom to selection tool (see section 3.2.1) used. The *Transparency mode* under the geometry view options may also be activated to allow visual location of the selected entity in the 3D view.

In certain cases it is possible to use the model without correcting a geometry fault, but this is not advised. A list of possible faults and solutions are given in CADFEKO geometry faults (see section 27).

5.2 Importing meshes

CADFEKO can import a variety of mesh formats. The import tools are accessed by clicking on the *Import* menu anchor. From the *Application menu* select *Mesh* → ...

5.2.1 Importing *.fek files

When importing *.fek files, only the mesh parts (wire segments, triangles, polygonal plates and tetrahedra) are imported; information regarding the solution setup is completely ignored. As CADFEKO specifies dielectric medium boundaries based on face labels, it appends an underscore and a number to create unique labels if the *.fek file contains different dielectric boundaries with the same label. In addition, since the imported labels are listed under a new mesh part, the labels in the imported model contain an additional level. Medium information and segment radii are retained during the import.

5.2.2 Importing general mesh formats

CADFEKO imports other mesh formats by running PREFEKO and importing the resulting *.fek file. Currently NEC, FEMAP neutral files (boundary surfaces bordered with line curves are imported as polygonal plates), NASTRAN, meshed AutoCAD DXF files (only the LINE and POLYLINE structures which defines segments and triangles), STL, PATRAN, Ansys CBD, Concept files, ABAQUS meshes, GiD meshes (for GiD meshes, hexahedra elements are ignored) and a special ASCII format are supported. (For more detail on the ASCII format, see the IN card (see section 13.22)).

For all formats (except *.fek files), CADFEKO opens an additional *Import mesh* dialog. Here the user can set a scale factor and control which elements to import. (For example, boundary elements in FEMAP neutral files may be either part of the geometry — i.e. they should not be imported — or intended as polygonal plates — i.e. they must be imported.) *Quadrangles* are divided into triangles during the import process. Only Ansys CBD files support segment radius information — for all other formats a default radius must be specified on the *Import mesh* dialog. (The default radius is also used for Ansys segments where a radius is not defined in the file.) Different radii can be set on the segments after importing. Finally, for meshed AutoCAD DXF files, the LINE elements are divided in segments according to the value of the *Segment length* field. If the LINE elements must not be subdivided, this value must be larger than the longest line. Since these formats do not support specifying dielectric media, all segments, triangles and polygonal plates are PEC structures in free space and all tetrahedra have the medium *Unknown*.

It is also possible to import meshes ‘manually’ by constructing a simple *.pre file that imports the mesh in EDITFEKO, running PREFEKO and then importing the *.fek file in CADFEKO. This approach provides more control over the import process, and gives access to some of the more advanced import options, for example to only include certain labels.

5.3 Exporting meshes

CADFEKO is able to export NASTRAN mesh formats. It is exported directly from CADFEKO and does not make use of PREFEKO.

5.4 Printing and image export

CADFEKO provides a tool to export the current 3D view to various image formats (*.jpg, *.bmp, *.xpm, *.png and *.eps), as well as the direct printing of the current 3D view to a printer.

Image export may be accessed through the *Image export* option on the *Tools* tab ribbon. Printing of images may be accessed through the *Print* option on the application menu.

The resolution of exported images may be specified and the exported image will be saved in the chosen location. All printed and exported images will contain the current active 3D view, with the current view-direction, zoom factor, colouring and visibility settings.

6 Defining the solution configuration in CADFEKO

It is possible to completely define models and request solution results from within CADFEKO. A standard model includes: a physical model; ports that define entry / exit points for energy; excitations on the ports that define the behaviour of the energy; and a mesh that represents all of these components. The creation of these components is discussed, along with instructions on how to request results for later processing in POSTFEKO. It is also possible to run the FEKO solver directly from CADFEKO and to edit CADFEKO models using EDITFEKO.

6.1 Using dielectric and lossy media

Media (representing different materials e.g. metals, dielectrics etc.) must be defined before they can be used in the model definition.

Default media

The first three entries in the *Media* tree are *Perfect electric conductor*, *Perfect magnetic conductor* (currently only used when setting magnetic ground planes) and *Free space*. The properties of the *Perfect electric conductor* and *Perfect magnetic conductor* cannot be edited nor deleted, but the properties of *Free space* may be edited. This provides for simulation where the free space media is not a vacuum.

User-defined media

All user defined media/materials are listed under *Media* in the contents tree. Though each of the different media types is listed separately, the medium names must be globally unique and contain less than 43 characters. The coloured square next to each medium entry indicates which colour is used to represent this medium in the 3D display (when the *Colour by region medium*, *Colour by face medium* or *Colour by coating medium* colouring options are selected (see section 3.2.3)) and in the details tree (see section 3.3.2). The display colour for a medium can be changed by selecting *Change display colour* from the medium's pop-up menu. (The selected colours are saved as part of the model in the *.cfx file and when the model is opened, the colours will therefore be maintained, even if the session file is not available.)

All media (except the default media — *Perfect electric conductor*, *Perfect magnetic conductor* and *Free space*) can be renamed in the tree or on the respective *Properties* dialogs. The name change will be reflected in all instances where the dielectric is applied. Media may also be deleted, provided they are not used by any other item.

6.1.1 Defining media

Dielectric media

Dielectric media can be applied to geometry regions (see section 6.1.2) and used as building blocks in the definition of *Layered dielectrics* and planar substrates.

Dielectric media are created by right clicking on *Media* in the tree and selecting *Dielectric* or by selecting the *Construct* tab and clicking on the *Media* menu button.



Figure 6-1: The *Dielectric face* button

Changing the *Name* field at the bottom of the *Create dielectric medium* dialog has the same effect as renaming the medium in the tree.

In cases where the geometry of a model is not available, dielectric properties may be set on mesh faces. Dielectric media are set on the mesh in terms of the medium on the front side (normal - displayed blue in the *Default view* colouring) and back side (away from the normal - displayed brown in the *Default view* colouring) of the mesh. In this case, the onus rests on the user to ensure that the dielectric definitions are consistent (e.g. the medium specified on the front sides of two connected mesh triangles cannot differ, except if the normal of one of the elements has been reversed.) This approach should only be followed for imported meshes with no geometry representation.

A dielectric may be defined by specifying the definition method for the dielectric and magnetic modelling.

Dielectric modelling of the dielectric is defined by means of the following dielectric definition methods on the *Dielectric modelling* tab of the *Create dielectric medium* dialog:

- Frequency independent

The dielectric modelling of frequency independent dielectric media is defined in terms of the *Relative permittivity* (ϵ_r), *Relative permeability* (μ_r), *Magnetic loss tangent* ($\tan\delta_\mu$), and either the *Dielectric loss tangent* ($\tan\delta$) or *Conductivity* (σ). The loss tangent and conductivity both specify dielectric losses, but have different frequency behaviour. The selection depends on the specific problem. 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. Expressions may be used in any of the dialog fields.

The effective permittivity of the dielectric is given by

$$\epsilon_{eff} = \epsilon_0 \epsilon_r (1 - j \tan \delta) \quad (6-1)$$

or

$$\epsilon_{eff} = \epsilon_0 \epsilon_r - j \frac{\sigma}{\omega} \quad (6-2)$$

- 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. The method is defined in terms of the *Relative static permittivity* (ϵ_s), *Relative high frequency permittivity* (ϵ_∞) and the *Relaxation frequency* (f_r).

$$\epsilon^* = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1 + j \frac{f}{f_r}} \quad (6-3)$$

- Cole-Cole

The Cole-Cole model is similar to the Debye model, but uses one additional parameter to describe the material. The Cole-Cole method is defined in terms of the *Relative static permittivity* (ϵ_s), *Relative high frequency permittivity* (ϵ_∞), *Relaxation frequency* (f_r) and the *Attenuation factor* (α).

$$\epsilon^* = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1 + (j \frac{f}{f_r})^{1-\alpha}} \quad (6-4)$$

- Havrillak-Negami

The Havrillak-Negami is a more general model and should be able to successfully model liquids, solids and semi-solids.

The Havrillak-Negami method is defined in terms of the *Relative static permittivity* (ϵ_s), *Relative high frequency permittivity* (ϵ_∞), *Relaxation frequency* (f_r), *Attenuation factor* (α) and the *Phase factor* (β).

$$\epsilon^* = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{\left[1 + (j \frac{f}{f_r})^{1-\alpha}\right]^\beta} \quad (6-5)$$

- Djordevic-Sarkar

The Djordevic-Sarkar is particularly well suited as a broadband model for composite dielectrics. It is defined in terms of the *Variation of real permittivity* ($\Delta\epsilon$), *Relative high frequency permittivity* (Δ), *Conductivity* (σ), *Lower limit of angular frequency* (ω_1) and the *Upper limit of angular frequency* (ω_2).

$$\epsilon^* = \epsilon_\infty + \left[\frac{\Delta\epsilon}{\log_{10} \frac{\omega_2}{\omega_1}} \right] \frac{\ln \frac{\omega_2 + j\omega}{\omega_1 + j\omega}}{\ln(10)} - \frac{j\sigma}{\omega \epsilon_0} \quad (6-6)$$

- Specify points (linear interpolation)

The dielectric properties of the material can be 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. Parameters required are *Frequency*, *Relative permittivity* (ϵ_r) and either the *Loss tangent* ($\tan\Delta$) or *Conductivity* (σ).

Magnetic modelling of the dielectric is defined by means of the following dielectric definition methods on the *Magnetic modelling* tab of the *Create dielectric medium* dialog:

- Non magnetic

For non magnetic modelling, the relative permeability (μ_r) is set to 1.0 and the magnetic loss tangent ($\tan\delta_\mu$) is set to 0.0.

- Frequency independent

The magnetic modelling of frequency independent dielectric media is defined in terms of the *Relative permeability* (μ_r) and the *Magnetic loss tangent* ($\tan\delta_\mu$). The effective permeability of the dielectric is given by

$$\mu_{eff} = \mu_0 \mu_r (1 - j \tan \delta_\mu). \quad (6-7)$$

- Specify points (linear interpolation)

The dielectric properties of the material can be 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. Parameters required are *Frequency*, *Relative permeability* (μ_r) and *Magnetic loss tangent* ($\tan\delta_\mu$).

Alternatively, a file may be imported containing the dielectric and magnetic properties of the dielectric.

- Import from file

Note that the following keywords are used in the XML file format namely, *freq*, *permittivity*, *diel_loss_tangent*, *conductivity*, *permeability* and *mag_loss_tangent*.

For demonstrative purposes, the keywords *val_A*, *val_B* and *val_C* will be used instead in the demo XML file given below as the same format is also applicable when defining a metallic medium or impedance sheet.

```
<?xml version="1.0" encoding="UTF-8"?>
<materialDB creator="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 will then fill 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.

Thus the above XML file will then be parsed as if it was specified by the user as:

```
<?xml version="1.0" encoding="UTF-8"?>
<materialDB creator="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>
```

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.

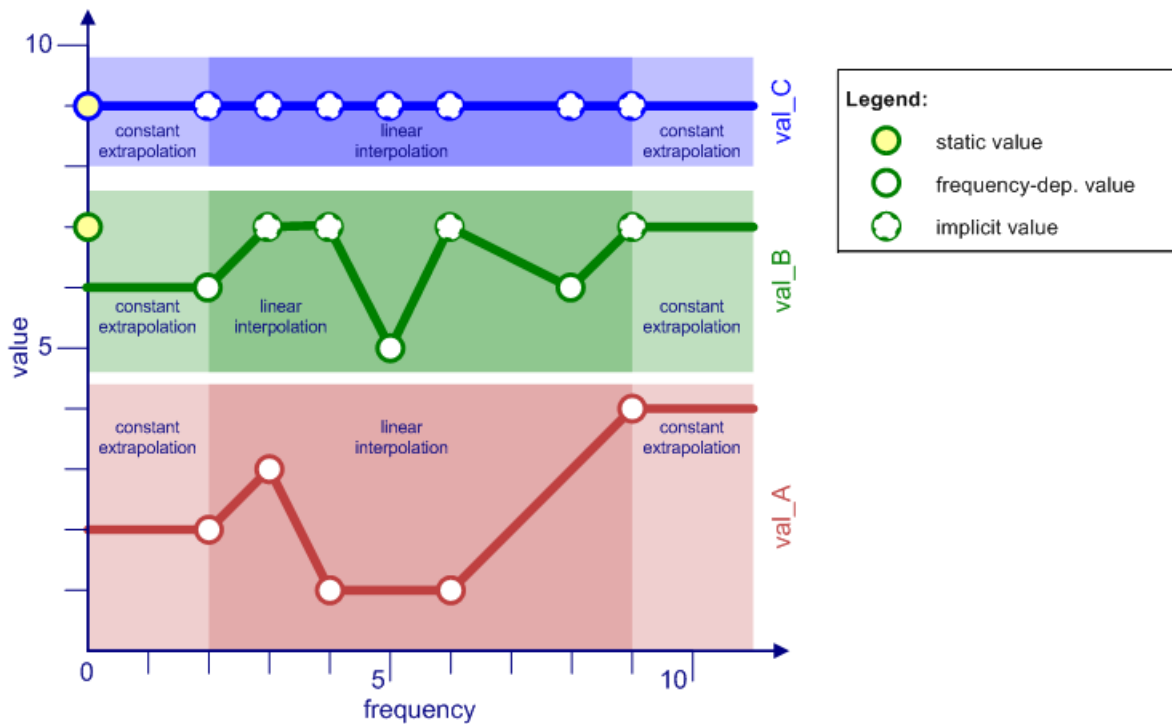


Figure 6-2: A graphical illustration showing the result of the parsed XML file

Metallic media

Conducting (skin effect) losses are specified by setting the face media of the relevant face/s to *Metallic*. (This cannot be done for faces bordering perfectly electric conducting regions.) *Metallic* media are created by right clicking on *Media* in the tree and selecting *Metallic medium*, or by selecting the *Construct* tab and clicking on the *Media* menu button.



Figure 6-3: The *Metallic medium* button

Note that it is not possible to apply a user defined metallic medium directly to a solid region (except for the special case of the *Perfect electric conductor* medium). The medium of the internal region should rather be set to free space (or dielectric) and a 'thick' metallic medium property applied to the faces bounding the region. ('Thick' here implies that the property should be set much thicker than the medium skin depth (as defined in Equation 6-8 in the section that deals with parameters on faces (see section 6.1.2).)

Metallic media are defined in terms of the (i) *Conductivity* (σ), *Relative permeability* (μ_r) and the *Magnetic loss tangent* ($\tan\Delta_\mu$), as frequency dependent data points or by importing the medium from file. See the Dielectric media (see section 6.1.1) for the XML format. Keywords for metallic media are *freq*, *conductivity*, *permeability* and *mag_loss_tangent*.

In this manual the phrase *conducting media* collectively refers to *Metallic* media as well as the *Perfect electric conductor*.

Media library

The media library consist of a list of predefined and user defined media. Predefined media may be imported from the media library to be used in the CADFEKO model. Media may also be added from the CADFEKO model to the Media library to be used in future CADFEKO models.

A medium is added from the *Media library* by selecting the *Construct* tab, clicking on the *Media* menu button and selecting *Media library* from the menu. The *Media library* dialog may also be launched by right-clicking on *Media* in the tree and selecting *Media library* from the popup menu.



Figure 6-4: The *Media library* button

The *Media library* has a filter functionality whereby a user can filter through the list of media for a required medium. If more information regarding the medium is required, the *Advanced* view mode may be used to display the medium properties of the selected medium.

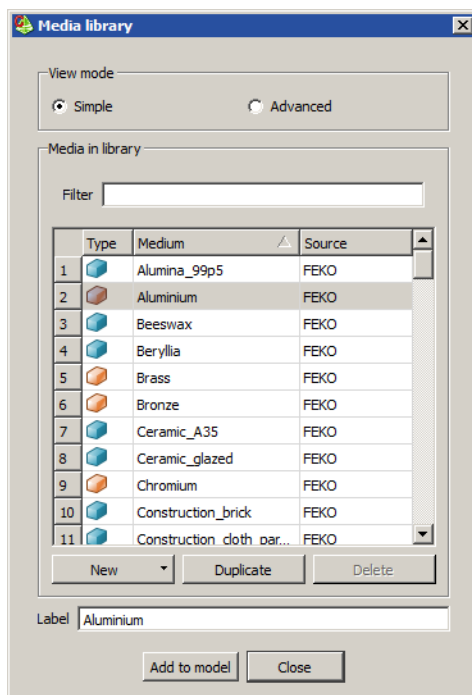


Figure 6-5: The *Media library* dialog

Layered dielectrics

Surfaces that define the boundary between free space regions can be modelled as thin dielectric sheets or as coated conductors. This is done by applying user defined *Layered dielectric* media (which consist of an arbitrary number of dielectric layers of a specified thickness) to the surfaces. Note that the *Dielectric* media constituting the various layers must be defined before the *Layered dielectric* can be defined.

Layered dielectric media are created by right clicking on *Media* in the tree and selecting *Layered structures* → *Layered dielectric*, or by selecting the *Construct* tab and clicking on the *Media* menu button.



Figure 6-6: The *Layered dielectric* button

The *Create layered dielectric* dialog contains a table with fields for the *Thickness* and *Medium* of each layer. Only *Dielectric* media are available in the drop-down list for selection of the *Media* for each layer. Clicking the *Add* button inserts a layer directly after the currently selected layer and *Remove* deletes the selected layer from the list. Thus it is possible to insert new layers between any two layers. Expressions may be used in the *Thickness* fields.

Coatings and thin dielectric sheets must be geometrically or electrically thin depending on the selected solution. During EM validation of the model (see section 6.15), CADFEKO will give warnings if the thickness limits are approached and errors if the thickness limits are exceeded.

Layered dielectric (anisotropic)

Layered dielectric (anisotropic) are created by right clicking on *Media* in the tree and selecting *Layered structures* → *Layered dielectric (anisotropic)* or by selecting the *Construct* tab and clicking on the *Media* menu button.



Figure 6-7: The *Layered dielectric (anisotropic)* button

An anisotropic layered dielectric may be defined by specifying the *Thickness*, *Principal direction (deg)*, *Material in principal direction* and the *Material in the orthogonal direction*.

Note that the *Dielectric* media constituting the various anisotropic layers must be defined before the *Layered dielectric* can be defined.

Impedance sheets

Impedance sheet structures can be used to apply a surface impedance to surfaces that define the boundary between free space regions.

Impedance sheets are created by right clicking on *Media* in the tree and selecting *Impedance sheet* or by selecting the *Construct* tab and clicking on the *Media* menu button.



Figure 6-8: The *Impedance sheet* button

These “media” define the real and imaginary parts of the *Surface impedance* (effectively the ratio between tangential electric field on the surface and the electric surface current). In theory, the same effect can be achieved by applying an appropriately defined *Metallic* medium to this surface, but where only the surface impedance is known the properties of the equivalent metallic medium need not be derived.

The following options are available for defining impedance sheets: *Frequency independent*, *Specify points (linear interpolation)* or *Importing medium from file*. See the Dielectric media (see section 6.1.1) for the XML format. Keywords are *freq*, *surf_imp_re* and *surf_imp_im*.

Windscreen

Windscreens can be used to model multiple windscreen antennas on different windscreen dielectric definitions. Windscreens are created by right clicking on *Media* in the tree and selecting *Windscreen layers* or by selecting the *Construct* tab and clicking on the *Media* button.



Figure 6-9: The *Windscreen medium* button

When creating a windscreen antenna/analysis, the user needs to define the following three components:

- Windscreen antenna.
- Windscreen reference plane.
- Windscreen dielectric/definition.

Note that these three components should share the same windscreen name.

When defining an active windscreen antenna element, the user needs to select the required geometry in the tree. After right-clicking on the respective face/wire in the details tree, select *Properties*. Under the *Solution* tab, select *Windscreen* under the *Solve with special solution method*. Setting the *Element type* to *Active element*, will allow the user to specify the offset (offset A) from the reference to the active element. Note that this offset is specifically required because of the limitations of a finite mesh in combination with the curvature and rotation of the model, compared to a smooth surface.

The windscreen analysis is only supported for MoM. Other techniques such as MLFMM, FEM, PO and GO (ray launching) are not supported. Multiple windscreens with multiple glass definitions are allowed in one model. The active windscreen antenna elements should always be defined tangential w.r.t. to the dielectric windscreen reference plane (i.e. the different glass layers).

The windscreen reference may now be defined. Select the required geometry in the tree that will be used as reference. After right-clicking on the respective face in the details tree, select *Properties*. Under the *Solution* tab, select *Windscreen* under the *Solve with special solution method*. Set the *Element type* to *Reference element*.

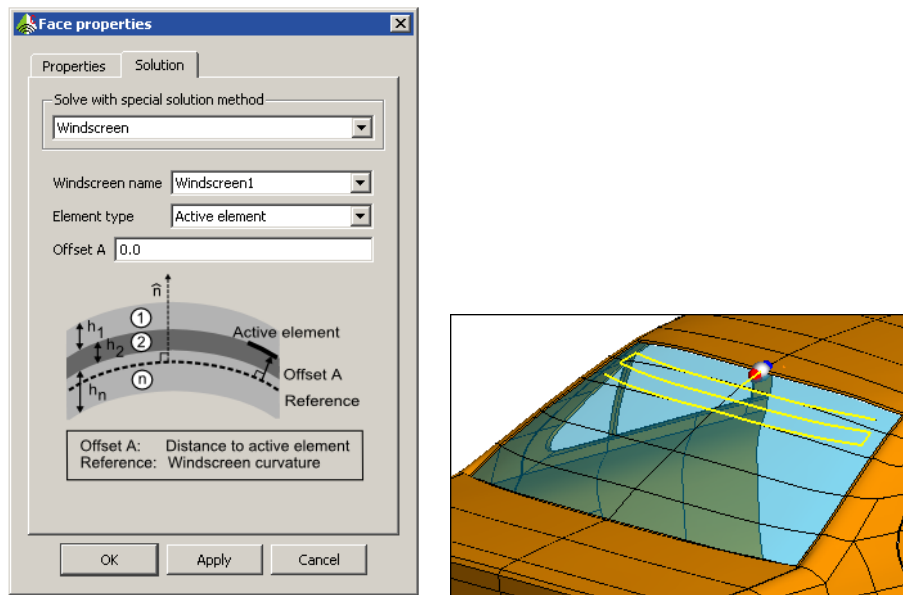


Figure 6-10: The CADFEKO dialog for defining a windscreen antenna

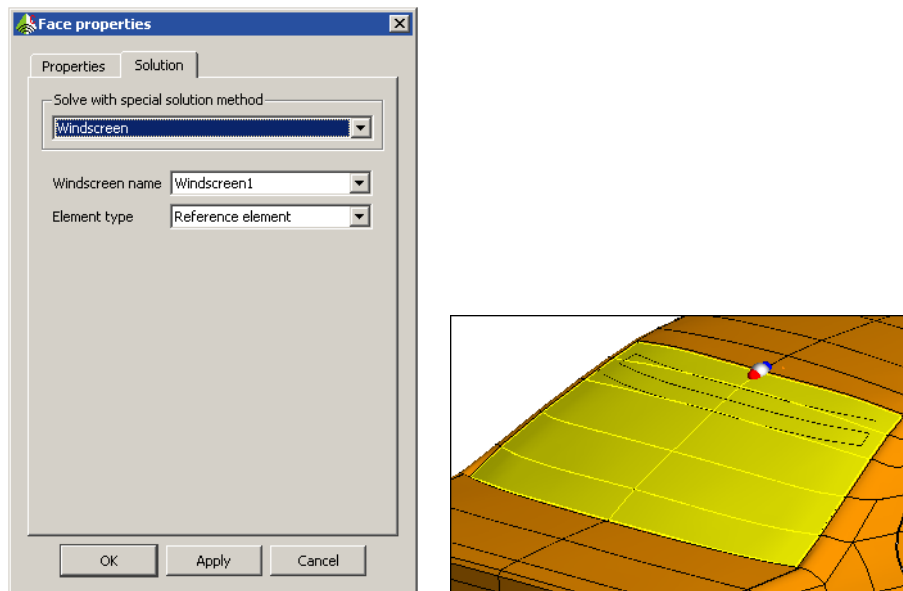


Figure 6-11: The CADFEKO dialog for defining a windscreen reference

Finally, the dielectric properties of the glass layers can be defined. A dielectric medium can be defined by right-clicking in the tree on *Media* and selecting *Create dielectric medium*. Next, a layered dielectric is defined by right-clicking again on *Media* in the tree and selecting *Create layered dielectric*. Enter a label and a thickness for the layered dielectric. The layered dielectric can now be used in the definition of the windscreen.

The windscreen can be created by a number of ways:

1. Select the *Construct* tab and click on the *Media* button. Select the *Windscreen medium* from the drop down menu.
2. In the tree, right-click on *Media* and select *Windscreen layers* from the popup menu

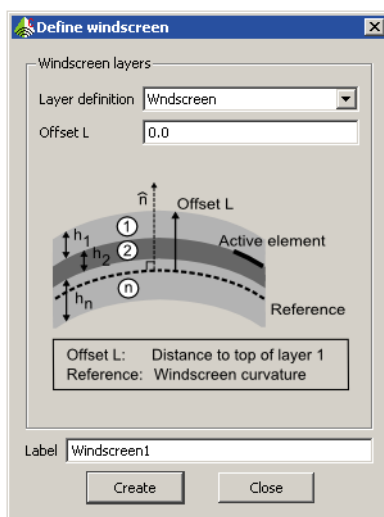


Figure 6-12: The CADFEKO dialog for defining a windscreen

On the *Define windscreen* dialog, enter the label of the layered dielectric in the *Layer definition* field. Enter the distance from the *Reference* to the top of layer 1 in the *Offset L* field.

6.1.2 Setting material properties on geometry

By default all items are perfectly electric conducting (and will be solved using the MoM). Material properties are set on regions, faces or edges / wires.

Regions are fully enclosed volumes that are listed in the details tree and may be selected in the details tree or selected in the 3D view by means of the *Auto selection* mode. Some imported models use surface parts to define the boundaries of solid regions. These models have no regions and it is not possible to specify solid dielectrics. However, if the surface parts of such a model are unioned to indicate connectivity, those parts that form closed surfaces will automatically be identified as regions in the resultant union part. (Where the unioning of imported faces does not results in closed region/s, the *Stitch sheet parts* tool may be used.)

Faces are the individual surface's of the part and are also listed in the details tree when the part is selected. Faces may be selected from either the details tree or by clicking in the 3D view when face selection (see section 3.2.2) is active. The term “face” is used to distinguish it from “surface”, which in CADFEKO refers to a 2D primitive such as a polygon.

Edges are also listed in the details tree when the part that contains them is selected. They include the edges of faces as well as free curves (also called “wires”).

Where possible, medium properties are retained during all geometry operations. For example, if a face is split in two by a subtract operation, both the resulting faces will inherit the properties of the parent; or, if two overlapping faces are intersected, the resulting face will have all the properties common to both parents. If it is not possible to merge the properties of the parents, the items are marked suspect (see section 4.4) and a warning is provided indicating which properties could not be resolved.

Dielectric volumes are indicated by applying a *Dielectric* media to the relevant geometry region.

Conducting losses are accounted for by applying a user defined *Metallic* media to the relevant conducting surfaces.

Coatings and thin dielectric sheets are specified by applying user defined *Layered dielectric* media (constructed from individual pre-defined *Dielectric* media layers) to surfaces or wires (free edges).

Impedance sheets (used to represent metal surfaces in cases where only the surface impedance per unit area is known for a surface) can be modelled by applying an *Impedance sheet* medium to the surface. (Although an impedance sheet is not strictly speaking a medium, user defined *Impedance sheets* are also listed under *Media* in the tree, and are applied, identified and displayed in a similar manner to the other media.)

Windscreens are indicated by applying *Windscreen* media to the relevant wires and faces.

Properties on regions

To set properties on regions, select one or more regions and select *Properties* from the pop-up menu. This opens the *Region properties* dialog shown in Figure 6-13.

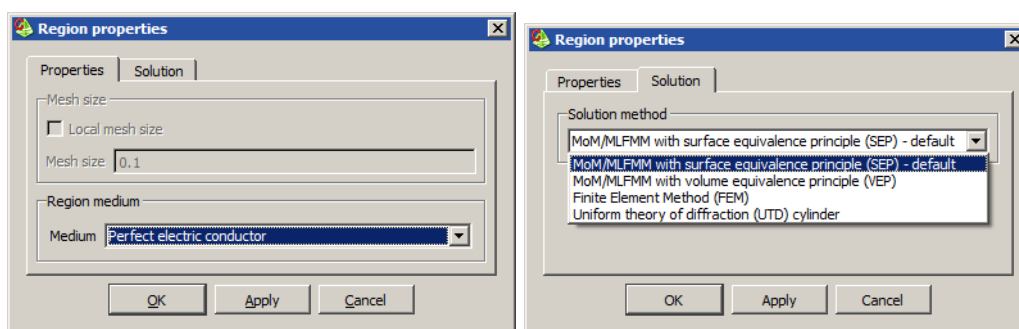


Figure 6-13: The *Region properties* dialog

On the *Properties* tab, the medium type of the selected region can be set. It may be *Perfect electric conductor*, *Free space* or any pre-defined *Dielectric*.

Perfect electric conductor: *Perfect electric conductor* (PEC) is the default media for newly-created regions. The local mesh size may be changed on the faces enclosing the region. It is possible to delete faces that border a PEC region. Deleting a face that forms part of the boundary of a region effectively removes the region.

Free space: Setting a region to *Free space* effectively converts it to a hollow shell (when setting a region to *Free space*, by default the faces bounding the region are set to PEC). It is possible to delete faces that border free space regions (if the media outside of the region is also free space). Deleting a face that forms part of the boundary of a region effectively removes the region, as the remaining face/s are no longer closed. Some face properties (e.g. a thin dielectric sheet) can only be set on faces bordering free space regions.

Dielectric: The *Dielectric* region medium is only available if the *Media* list contains dielectric media (see section 6.1.1). (New dielectrics may be defined while the *Region properties* dialog is open — the *Dielectric* medium will then immediately become available in the region properties dialog.) Unlike label names (which must be unique) the same medium can be applied to multiple regions. By default, the faces bounding a region that is denoted as *Dielectric* assume the same dielectric property as the region. This may be modified to any conductive medium (not a dielectric or layered dielectric medium) by setting the face properties for each face (see section 6.1.2). The faces bounding a region that is denoted as *Dielectric* can only be deleted under certain conditions. If the face defines a border between two different media, it cannot be deleted.

Where geometry operations introduce intersection of existing regions, and the parent regions have conflicting material properties, the resulting region/s will be marked suspect.

Initially the region properties dialog shows the current state of each property for the selected item(s). If multiple regions are selected, fields which are not the same for all dialogs are left blank. The properties represented by blank field are not modified when changes are applied. This allows, for example, modifying the mesh size for a number of different dielectric regions simultaneously without changing the media properties of the individual regions.

On the *Solution* tab of the *Region properties* dialog, the solution method for the selected region may be set (shown in Figure 6-13).

MoM/MLFMM with surface equivalence principle (SEP) - default: The selected region will be solved with MoM/MLFMM.

MoM/MLFMM with volume equivalence principle (VEP): The selected region will be solved by employing tetrahedra to accurately mesh arbitrarily shaped volumes with dielectric properties and solved by means of the MoM/MLFMM.

Finite Element Method (FEM): The selected region will be solved by employing tetrahedra to accurately mesh arbitrarily mesh shaped volumes with dielectric properties and solved by means of the finite element method (FEM).

Uniform theory of diffraction (UTD) cylinder: The selected cylinder region will be solved with the uniform theory of diffraction (UTD).

Properties on faces

The *Face properties* dialog (shown in Figure 6-14) has two tabs: *Properties* and *Solution*. The medium properties for the selected face/s are set on the *Properties* tab. Special solution options for the chosen face can be set on the *Solutions* tab.

The *Face medium* options available in the *Medium* drop-down list depend on the media of the two regions bordering the selected face/s. If there is free space on both the front and back sides of the face/s, the options will be *Perfect electric conductor*², *Lossy conducting surface*, *Impedance sheet*, *Thin dielectric sheet* and *Default*. (Some of these options are only available once the corresponding media have been defined.)

²*Perfect electric conductor* replaces the *Metal surface* option used on this dialog in CADFEKO 1.

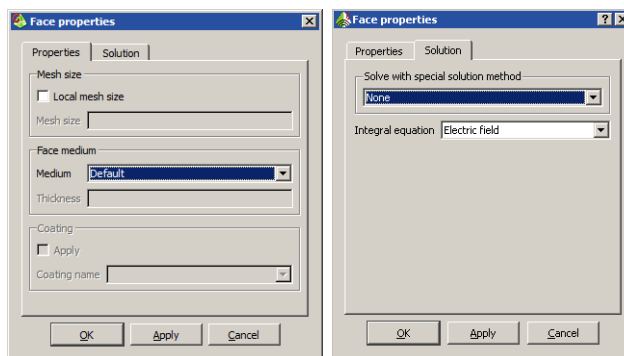


Figure 6-14: The *Face properties* dialog showing the face properties and solution settings options

The *Default* medium for faces that do not bound a region and for faces that bound a PEC or free space region is the same as *Perfect electric conductor*. The *Default* medium for faces that bound a dielectric region is the same as *Dielectric boundary*. If the medium selection is set to *Default* on a face that bounds a region, it will take on the default properties according to the region medium *Type* setting.

If a face is indicated as a *Lossy conducting surface*, a metallic *Medium* and a thickness must be specified. Note that it is not possible to set general *Metallic* media directly on regions. A lossy metallic region can be effectively modeled by setting the boundary faces of Free space region to lossy surfaces that are much thicker than the skin depth (more than $\times 5$) at the frequencies of interest, where the skin depth is defined as

$$\delta_{skin} = \sqrt{\frac{2}{\omega \mu \sigma}}. \quad (6-8)$$

The *Impedance sheet* (see section 6.1.1) and *Thin dielectric sheet* (see section 6.1.1) face types will respectively list all defined *Impedance sheets* and *Layered dielectric* media in the *Medium* options drop-down list.

The *Impedance sheet* may be used as an alternative method for defining conductive surfaces, while the *Thin dielectric sheet* is used to model flat multilayer dielectric structures such as automotive windows.³

The *Face type* selection is retained when a bordering region is changed. This may lead to invalid settings. For example, if a *Lossy conducting surface* is set on a face bordered by free space, this setting becomes invalid if either bordering region is changed to a *Perfect electric conductor*. Face properties that become invalid due to changes in region media will cause a warning to be reported in the message window and the face will become suspect (see section 4.4). Though the media properties of such suspect faces are displayed as usual in the details tree, the *Face type* field in the properties dialog enters a ternary state (even if all selected faces have the same setting). Thus the region that caused the invalid setting can be reset without affecting the face settings (which will then become valid again). The face will be marked suspect during this process and must be set “not suspect” manually once the problem has been resolved.

In addition to the media type settings, *Coatings* may be applied to faces. Coatings are applied on both sides of the face. Hence this option is only available when the face has free space on both

³The kernel may refer to the SK card in error messages dealing with *Impedance sheet* and *Thin dielectric sheet* specifications.

sides. The *Coating name* must be selected from a list of *Layered media*. The coating is applied such that layer 1 (according to the layered dielectric definition) is on the outside, as shown in Figure 6-15.⁴

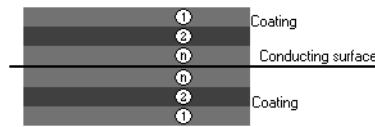


Figure 6-15: Coatings are applied to both sides of conducting surfaces.

On the *Solution* tab, the option to solve the selected face with a special solution method, is set. The following options are available:

- None
- Physical optics (PO) - full ray-tracing
- Physical optics (PO) - always illuminated
- Physical optics (PO) - only illuminate from front
- Large element PO - full ray-tracing
- Large element PO - always illuminated
- Large element PO - only illuminated from front
- Geometrical optics (GO) - ray launching
- Uniform theory of diffraction (UTD)
- Windscreen
- Planar Green's function aperture

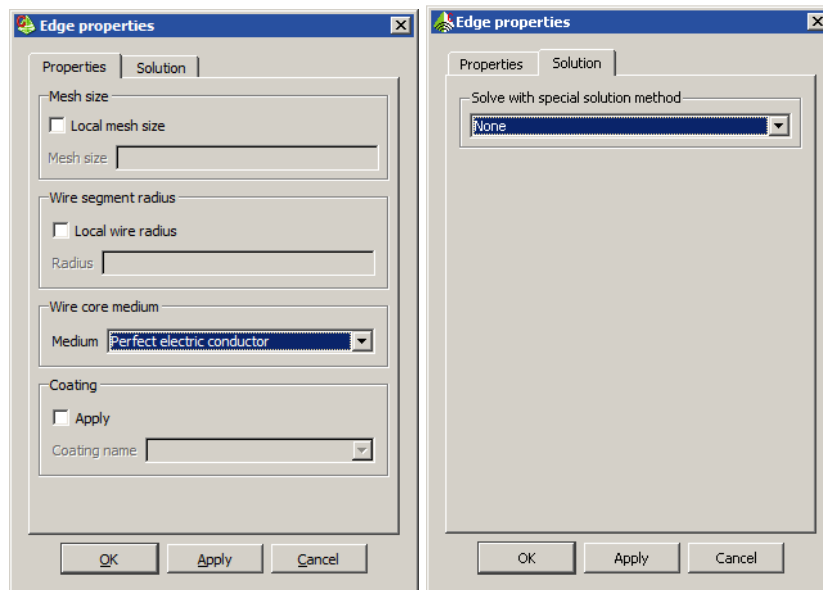
Properties on edges

The *Edge properties* dialog (shown in Figure 6-16) provides the option to set a local wire radius (overriding the default setting on the *Create mesh* dialog) for any selected edge in the model. If the selected edge/s are wires (i.e. free edges that do not define the boundary of a face), *Wire segment radius*, *Wire core medium* and *Coating* options are available in addition to the local mesh size option.

Conducting losses on wires can be accounted for by selecting the pre-defined metal in the *Medium* list drop-down list.

For a coating to be applied to a wire, the *Coating name* is selected from a list of the pre-defined *Layered dielectric* media. Currently only single layer coatings can be applied to wires.

⁴The kernel may refer to the CO card in error messages regarding coatings.

Figure 6-16: The *Edge properties* dialog

On the *Solution* tab, the option to solve the selected edge with a special solution method is available. The options available are:

- None
- Windscreen

6.1.3 Setting properties on mesh elements

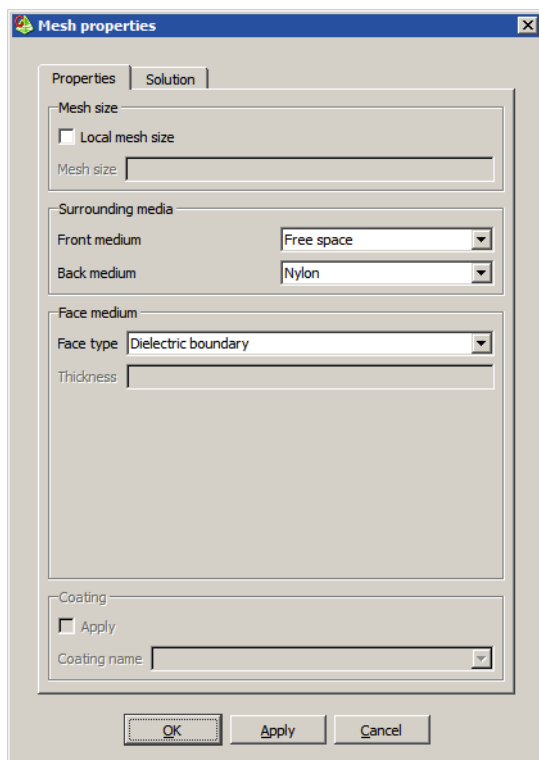
Material properties should be set on the geometry or on a geometry mesh. Once the mesh action is performed, a resulting *simulation mesh* will be generated and indicated by a blue mesh icon. Any major changes to the geometry will automatically remove the simulation mesh, since it will no longer accurately represent the geometry.

It is not possible to directly edit a simulation mesh. However, it is possible to unlink a simulation mesh from its geometry and edit it as though it is a base geometry mesh. This type of mesh can also be obtained by importing a model that is saved in a mesh format. It is possible to set media properties on these meshes.

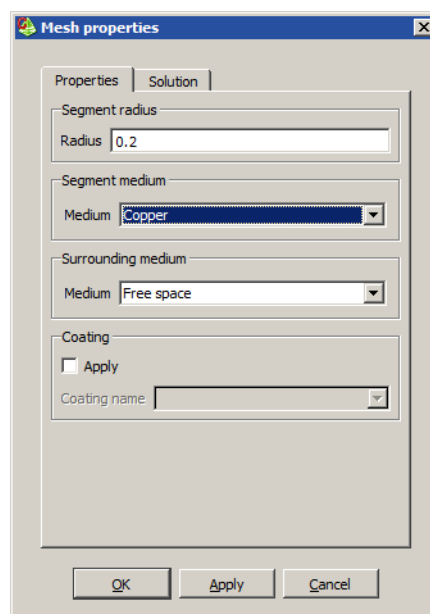
Select the required element labels (not the individual elements) in the details tree and access the *Properties* from the right-click context menu. Properties can only be set on one type of mesh element at a time (for example, if the selection contains mesh triangles and wire segments then the properties dialog will not be available). According to the type of elements selected, the relevant mesh properties dialog (as shown in Figure 6-17) will be launched.

For triangles, the medium is specified on each side. The media on either side of the mesh elements are specified on the same dialog (rather than on a separate region dialog for faces that bound a region). The *Face medium* and *Coating* properties may be set as for geometry faces.

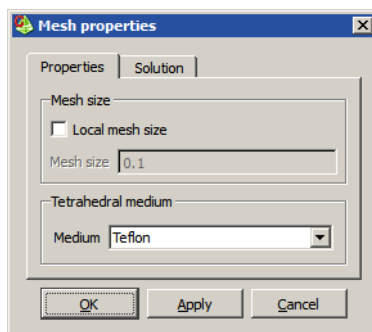
For segment labels in the mesh, the surrounding medium can be set on the *Mesh properties* dialog. The wire segment core medium and coating can be specified.



Triangle element/s



Wire segment/s



Tetrahedral element/s

Figure 6-17: The *Mesh properties* dialogs for triangle, wire segment and tetrahedral elements.

For tetrahedra mesh elements, only the volume medium can be specified by referencing the label of a user defined dielectric medium.

If any medium is changed in such a way that the *Face medium* or *Coating* setting on a mesh becomes invalid, that setting immediately reverts to a ternary state and has to be updated manually.

6.1.4 Displaying dielectric media, coatings and thin sheets

The display options may be accessed on the *3D view* context tab, *Display options* contextual tab. To select the required colouring option to colour the display according to the dielectric regions/face media, click on the *Colour* button. The display of coatings in the 3D view may be enabled/disabled by clicking on the *Coatings* button.



Figure 6-18: The (a) *Colour* button and (b) the *Coatings* button

In addition to the special 3D view colouring options, the face and coating parameters (media, local mesh settings etc.) are indicated using icons in the tree (see section 3.3.2). Faces on dielectric regions are indicated with an icon in front of the item in the tree.

Mesh segments are displayed with their actual radius in the individual 3D views. This radius may include the coating radius. For disabling the display of the segment coating, click on the *Coatings* button on the *3D view* context tab, *Display options* contextual tab.

6.1.5 Displaying the model view, simulation mesh and overlay

6.2 Setting the frequency

Select the *Source/Load* tab and click on the *Frequency* button or double-click on *Solution* in the tree to open the *Solution frequency* dialog (shown in Figure 6-20 and Figure 6-21). There are four *Frequency range* options.



Figure 6-19: The *Frequency* button

Single frequency: Allows setting a specific solution frequency

Continuous (interpolated) range: All requested results are calculated using adaptive sampling in the range *Start frequency* to *End frequency*. The sampling algorithm (see section 21) uses finer sampling in areas where the results change rapidly to ensure that all resonance effects are calculated accurately. Since all the requested results are interpolated, it is best to use this technique with as few result requests

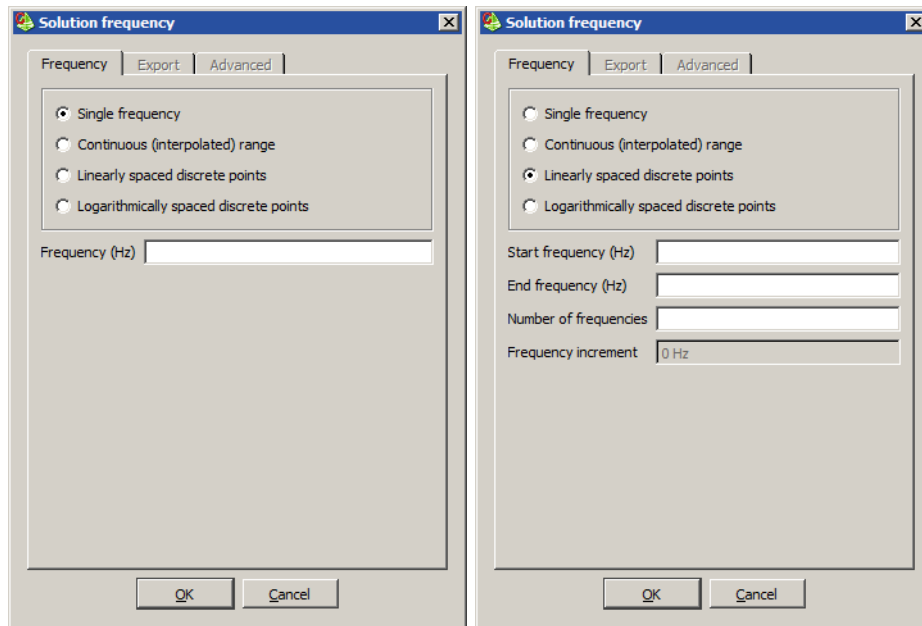


Figure 6-20: The solution frequency dialog used to define the solution frequencies. The *Single frequency* and *Linearly spaced discrete points* options are shown.

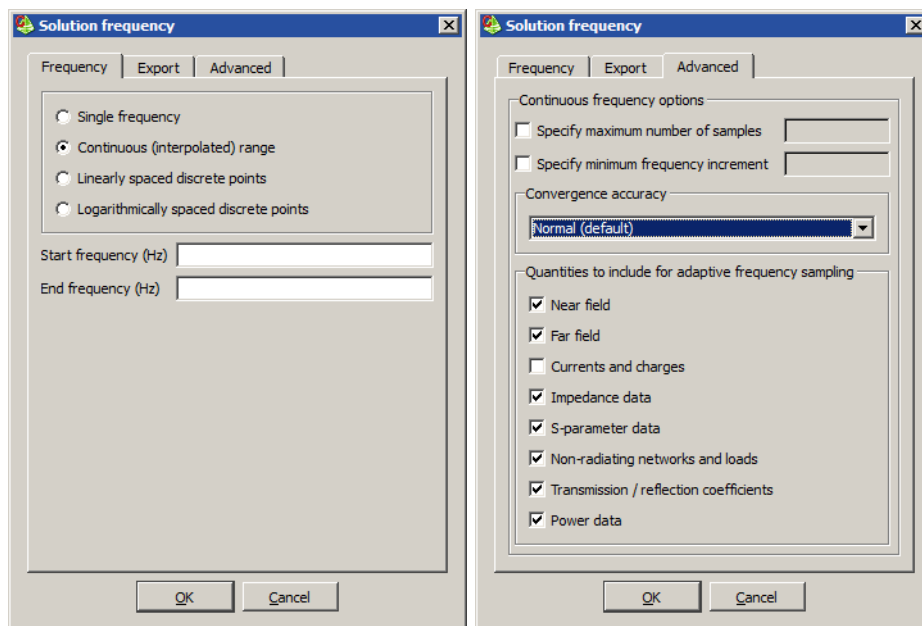


Figure 6-21: The solution frequency dialog showing the *Continuous (interpolated) range* and the *Advanced* tab.

as possible. The *Maximum number of samples* option limits the number of solutions and hence the runtime, but the results may be inaccurate if not fully converged. The *Minimum frequency increment* is used to limit how far FEKO refines the frequency. This is mostly useful if there are small discontinuities in the results. In general, the user should use the default settings for these two options.

On the *Advanced* tab of the *Solution frequency*, the following *Continuous frequency options* can be set: *Specify maximum number of samples*, *Specify minimum frequency increment*, *Specify sampling for exported data files*. The fre-

quency stepping can also be set as *Linear* or *Logarithmically*. An option is also available to set the *Convergence accuracy* as the following:

High more samples, highly resonant structure

Normal default

Low fewer samples, smooth frequency response

A user may also elect to include certain quantities for adaptive frequency sampling. Selecting specific quantities of interest will include it in the adaptive sampling. Unselected quantities will be calculated at the discrete solution frequency points.

Linearly spaced discrete points: The user can specify a fixed number of linearly spaced points between the *Start frequency* and the *End frequency*. CADFEKO calculates and displays the increment. This is typically used when the exact frequencies are known where the solution is required.

Logarithmically spaced discrete points: The user can specify a fixed number of logarithmically spaced points between the *Start frequency* and the *End frequency*. CADFEKO calculates and displays the multiplicative factor. This is typically used when the exact frequencies are known where the solution is required.

6.3 Setting the total source 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. FEKO can therefore scale the result to yield the desired source power level.

Select the *Source/Load* tab and click on the *Power* button or double-click *Power* under *Excitations* in the tree to control the source power (the *Power settings* dialog is shown in Figure 6-23).



Figure 6-22: The *Power* button

Note that FEKO uses peak magnitude for all complex values — hence excitations (i.e. voltage sources and current sources) must be specified with peak magnitude (as opposed to root mean square values) if no power scaling is done. Note however that power settings are specified as time-averaged values.

No power scaling: This is the default setting. If this option is selected, FEKO will calculate the results using the specified source magnitudes. Note that plane wave excitations have infinite extent and therefore infinite power. If a model contains plane wave excitations, the *No power scaling* option must be selected.

Total source power: If *Total source power* is selected, FEKO will 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* field. No mismatch is taken into account. This option can be used with any excitation except plane waves.

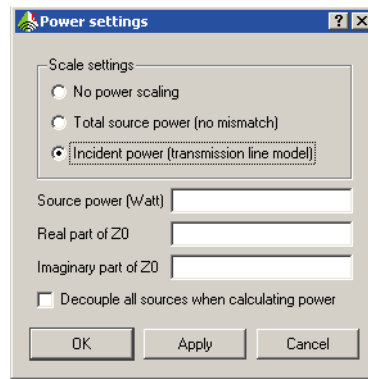


Figure 6-23: The power settings dialog used to control the source power options

Incident power (transmission line model): If *Incident power (transmission line model)* is selected, all structures are assumed to be 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, some of the incident power will be reflected back to the source. This is the mismatch loss. This option may be used for models that contain only voltage source excitations. See illustration in the PW card (see section 14.55)

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 very time consuming. If *Decouple all sources when calculating power* is checked, the mutual coupling is not considered when calculating the source power. This is acceptable for sources which (in terms of the computation wavelength) are relatively far from each other and from other structures in the model, or 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 checked.)

6.4 General networks and ideal transmission lines

General networks (based on network parameter descriptions) and ideal transmission lines may be included in the CADFEKO model. These network components may be interconnected and/or connected to the model geometry and may also be excited or loaded directly.

6.4.1 Ideal transmission lines

Ideal transmission lines may be added to the CADFEKO model by selecting the *Save/Load* tab and clicking on the *TX line* button or right-clicking on the *Non-radiating networks* branch on the tree and clicking on the *TX line*.



Figure 6-24: The *TX line* button

The *Add transmission line* menu is shown in Figure 6-25. The option to automatically determine the length of the transmission line based on the physical distance between the two points, is available.

The parameters of the transmission line may be specified, and a label may be assigned. Note that the transmission line length is specified in terms of the model unit (default metre).

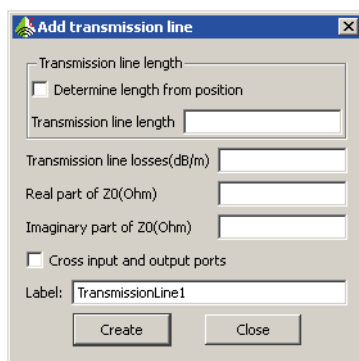


Figure 6-25: The Add transmission line dialog

After a transmission line has been created, it will be shown under the *Non-radiating networks* branch of the tree, and the transmission line properties may be edited by double-clicking on this representation to open the *Modify transmission line* menu.

6.4.2 General networks

General non-radiating networks may be added to the CADFEKO model by clicking on the *Source/Load* tab and clicking on the *Source/Load* tab and clicking on the *Network* button or right-clicking on the *Non-radiating networks* branch and clicking on *Network*.



Figure 6-26: The *Network* button

The *Add general network* menu is shown in Figure 6-27. The network may be specified in terms of S- Y- or Z-parameters imported from an external Touchstone file, a direct component-based network imported from a SPICE *.cir file or manually specified in CADFEKO.

If a Touchstone or network file is used to specify the network parameters, the number of ports must be indicated (this may be any number greater than or equal to 1) and the reference impedances for the ports will be determined from the Touchstone file for the case of S-parameters. If network parameters are specified manually in CADFEKO, a maximum of 4 ports may be used, and the S-parameter reference impedances for each port should be specified (these reference impedances must be real and need not be the same for all ports) when S-parameters are used. Note that only the Touchstone format v1.0 is supported.

If a SPICE *.cir file is used to specify the network (e.g. RLC network), the following are required: the number of ports and the label for the general network. Note that the number of ports and the general network label in CADFEKO need to correspond to the number of ports and subcircuit label in the *.cir file.

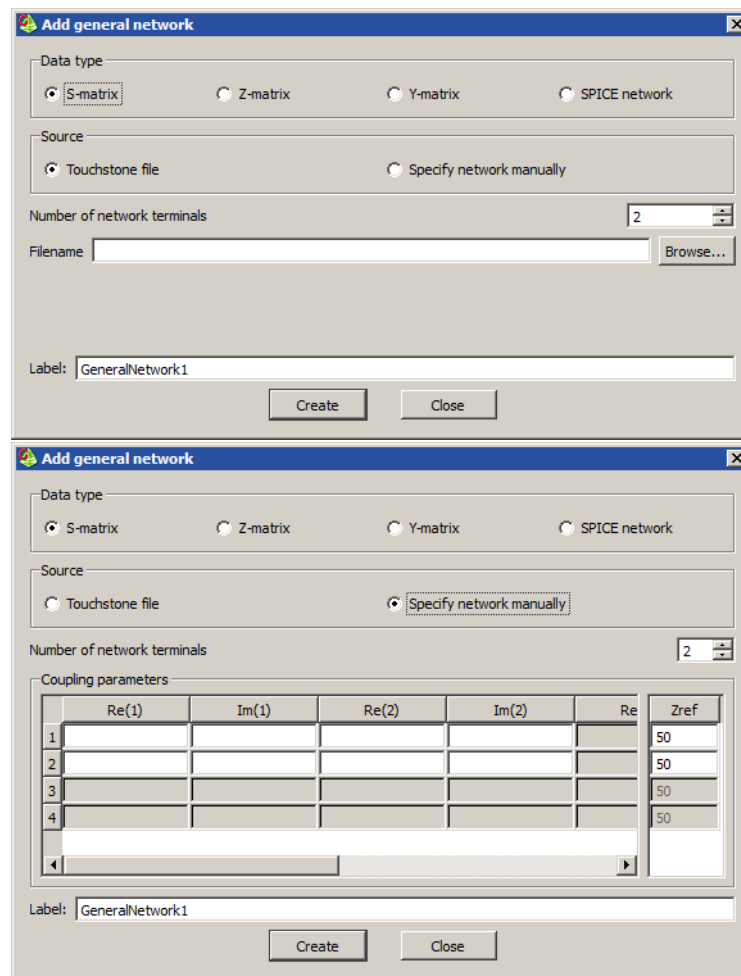


Figure 6-27: The *Add general network* dialogs for Touchstone file and manual parameter definition.

The *.cir file should contain a description of the circuit defined in Berkeley SPICE3f5 syntax. Please ensure that the correct syntax is used since FEKO only uses a subset of the Berkeley SPICE3f5 syntax (see section 26). Also note that currently only linear circuits are supported because FEKO is a frequency domain solver.

After a general network has been created, it will be shown under the *Non-radiating networks* branch of the tree, and the definition of the network may be edited by double-clicking on this representation to open the *Modify general network* menu.

6.4.3 Connecting, exciting and loading general networks

General non-radiating networks and Transmission lines that have been defined in a model may be inter-connected or connected to geometry/mesh ports in the model. In order to specify the interconnection of networks, transmission lines and ports, select the *View* tab and click on the *Schematic view* button. The schematic tools may be accessed by clicking on the *Schematic* context tab.

Connections between networks, transmission lines and ports can be created by clicking on the *Connector* icon in the *Connect non-radiating networks*. A connection between two elements can be created by clicking on the connector point (indicated by a white dot) and dragging the

Figure 6-28: The *Schematic view* button

connection until the mouse cursor is over the desired second connection point. When clicking at the second, a connection between wires will be indicated by a black dot. Selected networks, transmission lines, ports and connections will be indicated by a dotted outline. Connections between elements can be deleted by selecting the respective element and pressing <Delete> or by making use of the *Delete* icon.

Loads and excitations may be connected to general non-radiating network ports and transmission line ports in exactly the same way as they are connected to geometry and mesh ports (see section 6.6).

6.5 Sources independent of the geometry

6.5.1 Plane waves

Select the *Source/Load* tab and click on the *Plane wave* button or right-click on *Excitation* in the tree and select *Add plane wave* to specify a plane wave excitation.

Figure 6-29: The *Plane wave* button

All other sources (listed separately under *Excitations*) will also be considered active during the computations for each plane wave incident direction. While defining or editing a plane wave source, a preview of the incident and polarisation directions is shown in the 3D view. The preview indicates the polarisation of the E-field (green arrow) and the direction of propagation of the plane wave (blue arrow) for each of the defined incident directions. Once the field is created, the E-field polarisation is depicted by a red arrow.

The *plane wave* source has two modes of operation.

Single incident wave: If *Single incident wave* is selected, a single plane wave is added to the existing sources. Multiple *Single incident wave* excitations can be added to create specific field distributions.

Loop over multiple directions: If *Loop over multiple directions* is selected, FEKO calculates a solution for each specified direction of incidence. The incident direction is specified in a spherical coordinate system, in terms of the angles θ and ϕ (in degrees). The user must specify the *Start* angle, *End* angle and angle *Increment* for each angular coordinate. CADFEKO calculates and displays the resulting number of incident directions. The actual end angle is determined based on the start angle, the increment and the number of samples and may not coincide exactly with the specified end angle. Note that with the *Loop over multiple directions* a single plane wave source may also be defined (by setting the *Start* and *End* angles the same). It is currently not possible to define multiple independent plane wave excitation loops (more than a single incident direction) in a single CADFEKO solution setup.

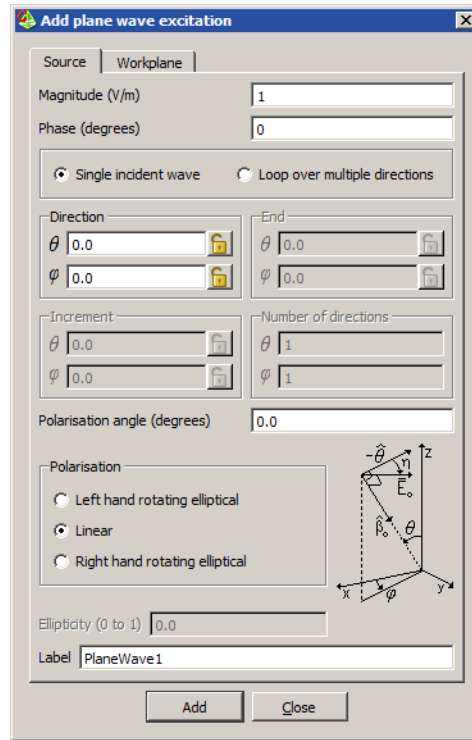


Figure 6-30: The incident plane wave excitation dialog

Polarisation angle: *Polarisation angle* specifies the angle η , in degrees, — measured in a right-handed sense around the direction of propagation — from $-\hat{\theta}$ to the polarisation vector $\vec{E}_0$ (linear polarisation) or major axis (elliptical polarisation). The angle fields in the dialog accept point entry (see section 3.2.2) from the 3D view. For elliptical polarisation the *Ellipticity* must be larger than 0 (linear polarisation) and smaller than or equal to 1 (circular polarisation).

The electric field strength of the incident field is given by

$$\vec{E}_i(\vec{r}) = [\vec{E}_0 + j\nu(\vec{E}_0 \times \hat{\beta}_0)] \cdot e^{-j\hat{\beta}_0 \cdot \vec{r}} \quad (6-9)$$

where ν is the ellipticity, $\hat{\beta}_0$ is the direction of incidence and $\vec{E}_0$ is the polarisation vector.

On the *Workplane* tab, a new workplane can be created. As a result, the plane wave is interpreted in the chosen/specified workplane. When a new workplane was not set, the plane wave will be influenced by the default workplane.

6.5.2 Electric and magnetic point dipoles

Incremental point sources may be added by selecting the *Source/Load* tab and click on the *Electric dipole/Magnetic dipole* button or by right-clicking on *Excitation* in the tree and selecting *Electric point source / Magnetic point source*.

The dialogs for definition of electric or magnetic point sources are shown in Figure 6-32. The *Position* and *Orientation* fields (the angles are in degrees) on these dialogs accept point entry (see section 3.2.2).



Figure 6-31: The (a) electric point dipole and (b) the magnetic point dipole buttons

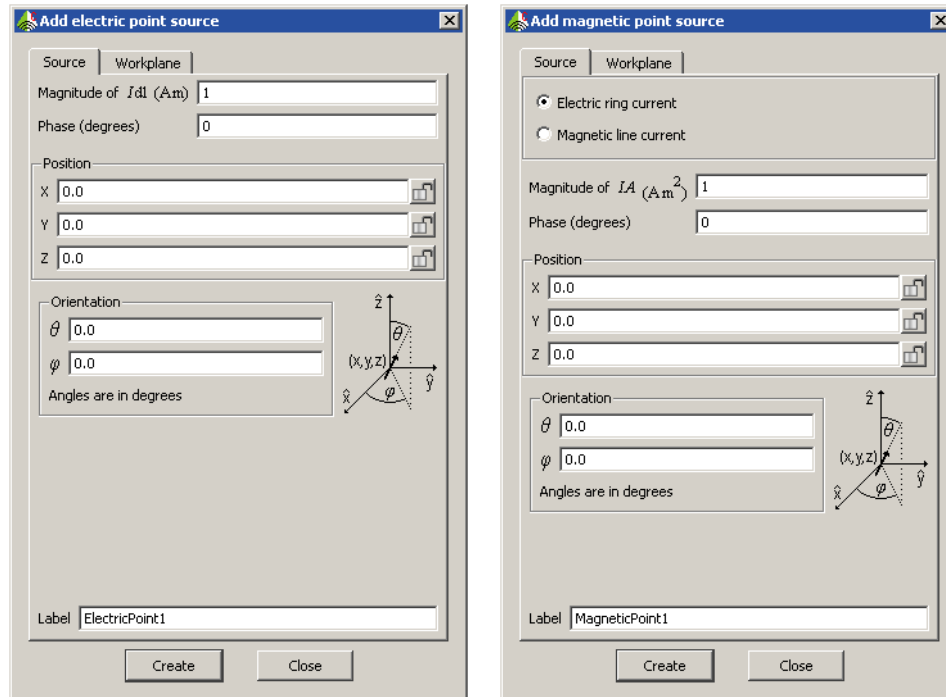


Figure 6-32: The electric and magnetic point source dialogs

The electric point source

The *Electric point source* represents an elementary dipole element with the specified orientation, magnitude and phase.

The magnetic point source

A magnetic point source can be either an *Electric ring current* (the magnitude is specified as the product of the loop current and loop area) or a *Magnetic line current* (the magnitude is specified as the product of the dipole length and the magnetic current). The electric ring current and magnetic line current radiate the same near and far fields, though the radiated potentials differ. The selection of which of these sources to use is dependant on the application.

6.5.3 Point sources with a specified pattern

To add a point source with a specified far field pattern, select the *Source/Load* tab and click on the *Radiation pattern* or right-click on *Excitation* in the tree and select *Radiation pattern point*.

The *Add radiation pattern point source* dialog is shown in Figure 6-34.

The specified *Magnitude scale factor* and *Phase offset* is applied to each field point in the entire pattern.

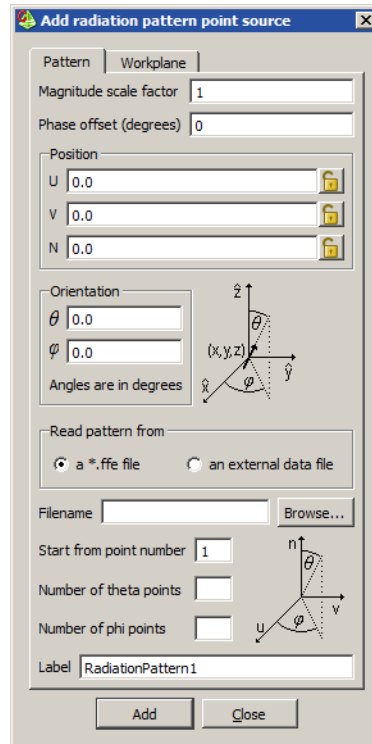
Figure 6-33: The *Radiation pattern* button

Figure 6-34: The radiation pattern point source definition dialog

On the *Workplane* tab, a new workplane can be created. As a result, all the fields are interpreted in the chosen/specified workplane. When a new workplane was not set, the *Position* field on the *Pattern* tab will be influenced by the default workplane.

Radiation pattern data may be imported from an **.ffe* file (created by FEKO) or from an ASCII data file. The AR card can be viewed for more detail on the ASCII format (see section 14.19).

The user must ensure that the pattern information in the data file is applicable to the solution frequency of the current model. Since far field patterns are typically frequency dependent, models with radiation pattern sources will usually have a single solution frequency only. If the radiation pattern was calculated using a frequency sweep in FEKO, the **.ffe* file will contain multiple patterns. The *Start from point number* field may then be used to indicate the position in the file of the first line of the required pattern. For example, if the pattern was originally calculated for 50 field directions in a frequency sweep, the pattern for the third frequency of the original solution may be selected by setting the *Start from point number* field to 101.

Data file handling

The pattern information is not loaded into CADFEKO, only a reference to the *Filename* is stored. The indicated file will therefore be re-processed each time PREFEKO is executed. This implies that it is possible to overwrite the pattern data file and recompute the FEKO solution without

making any changes to the model (though the user needs to keep track of the changes to the file and to ensure that the file can be located by FEKO during each computation).

The pattern data file is not archived by CADFEKO, since it may be in a different directory and may be used by other models. If the CADFEKO model is saved to a different directory using the *Save as* function, CADFEKO will first attempt to find the file in the same relative path and, failing this, try to find it in the original, absolute path. CADFEKO will then (as part of the *Save as* operation) update the file reference/s to the first valid location found.

6.5.4 Impressed spherical mode excitation

An *Impressed spherical mode* excitation can be defined based on pre-calculated spherical modes. The excitation is added by selecting the *Source/Load* tab and clicking on the *Spherical modes* button or by right-clicking on *Excitation* in the tree and selecting *Spherical modes*.



Figure 6-35: The *Spherical modes* button

The *Add spherical mode source* dialog (shown in Figure 6-36) allows the specification of the source position, orientation and the spherical mode expansion to be excited. The source data may be imported from an external TICRA *.sph file or specified manually.

When importing the source data from an external TICRA (*.sph) file, the following additional options are available such as scaling the magnitude of the data file and offsetting the phase.

For manual specification, each mode must be defined separately per line of the table on the dialog (also shown in Figure 6-36).

When specifying the spherical modes manually, individual modes may be added or removed using the *Add* and *Remove* buttons. The options for manual specification of the spherical modes are detailed as follows:

Propagation direction: This determines if the spherical waves propagate *Inward* (the model is illuminated with modes propagating towards $r = 0$, i.e. spherical Hankel function of the first kind $z_N^{(3)} = h_N^{(1)}$) or *Outward* (the modes radiate towards $r = \infty$, i.e. spherical Hankel function of the second kind $z_N^{(4)} = h_N^{(2)}$). This option is only available 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.

Index scheme: The *Normal* scheme uses the traditional smn index. If this option is selected, the user can specify *TE-mode* ($s = 1$) or *TM-mode* ($s = 2$) and the indices M and N in the indices columns. Here N is the mode index in radial direction and must be in the range $1, 2, \dots, \infty$ and M is the mode index in the azimuth direction φ . We do not distinguish between *even* and *odd* modes (with $\cos(M\varphi)$ and $\sin(M\varphi)$ angular dependencies), but rather use the angular dependency $e^{jM\varphi}$. Thus the index M can also be negative, but it must be in the range $-N \dots N$.

The *Compressed* scheme uses a compressed one-dimensional mode numbering scheme. The J mode index is then specified in the index column. Here

$$J = 2 [N(N + 1) + M - 1] + s \quad (6-10)$$

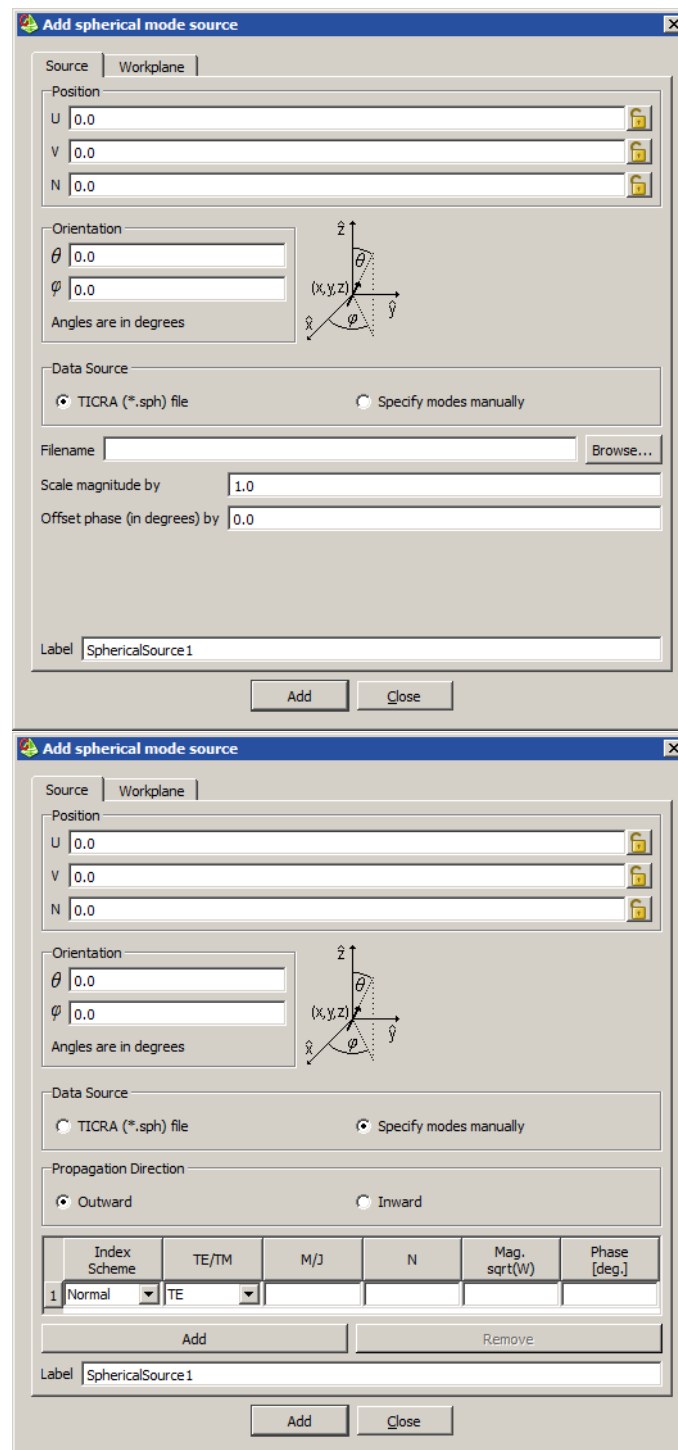


Figure 6-36: The *Add spherical mode source* dialog for excitations based on an external file

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.

Mag. 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 (deg.): The phase of the complex amplitude of this spherical mode in degrees.

6.5.5 Impressed aperture excitation

Aperture excitations may be used to specify a planar, cylindrical or spherical aperture of measured or calculated field values that are impressed as an excitation. The field values are converted into an equivalent array of electric and magnetic dipoles in the solution.

Aperture excitation is added by selecting the *Source/Load* tab and clicking on the *Aperture field* button or by right-clicking on the *Excitation* in the tree and selecting *Aperture field* to launch the *Add aperture excitation* dialog (shown in Figure 6-38).



Figure 6-37: The *Aperture field* button

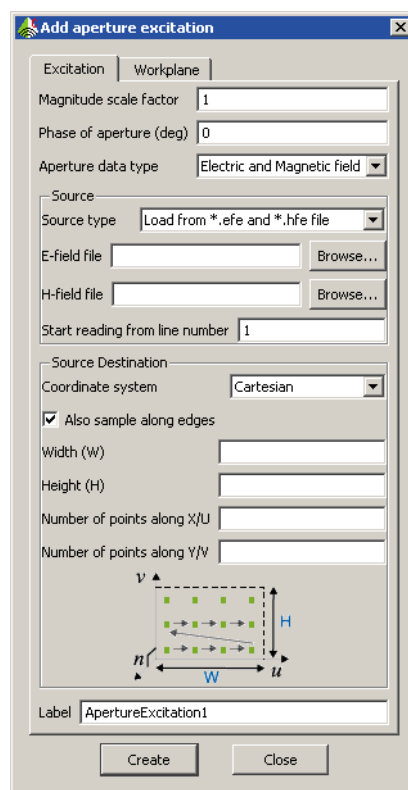


Figure 6-38: The *Add aperture source* dialog

In this dialog the source may be defined as follows:

Magnitude scale factor: The field magnitudes in the files will be multiplied by this factor. This is useful when using data files where the field values are in different units (e.g. $\mu\text{V}/\text{m}$)

Phase of the aperture (deg): This phase (in degrees) will be added to the phase of the fields as defined in the data files.

Aperture data type: The type of data to be used to generate the aperture excitation is specified here. An aperture excitation may either be defined based on only impressed electric or magnetic field distribution or a combination of both electric and magnetic fields.

Source: The data for the E and H fields may be imported from *.efe and/or *.hfe files generated by a previous FEKO solution (see section 6.12.5 and FEKO User Manual (section 14.31) for information on generating these files) or a more general ASCII file. The ASCII file format is discussed in FEKO User Manual (section 14.18). Note that the units for when reading data from an ASCII file are V/m and A/m for the E-field and H-field respectively. The first line number in the file from which data should be read may be specified in the *Start reading from line number* field. This may be used, for example, when a single file contains information for multiple aperture excitations, or when additional comment lines are inserted at the top of an ASCII data file that should be ignored. If both electric and magnetic field data is read, the starting line in the files will be the same.

Source Destination: The coordinate system in which the field data is specified, as well as the number of field points in each coordinate direction is specified here. It is important to note that data may only be specified in Spherical or Cylindrical coordinates when both electric and magnetic fields are used. An additional *Also sample along edges* option is provided. This should be used depending on how the physical location of the sample points relate to the aperture defined in the *Source Destination*. When checked the outer sample points are assumed to lie on the edges, when unchecked the sample point lie half an increment away from the edges. When using multiple aperture excitations in a single model, sample points should not lie on the edges of any two apertures that share a common side, otherwise two elementary dipoles elements with the same location and polarisation will be included. If this is the case the power calculation in FEKO will fail and incorrect results may be calculated.

Note that the position of the aperture excitation is by default at the global workplane. The position may be adjusted by modifying the workplane on the *Workplane* tab.

6.5.6 Impressed current

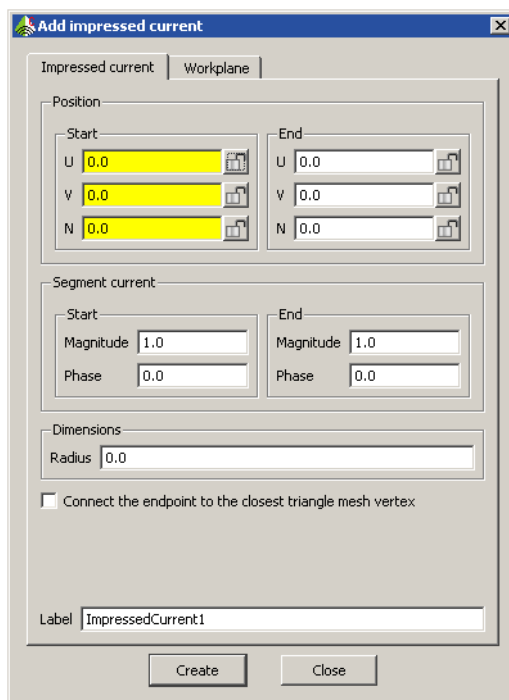
An *Impressed current* may be defined as an excitation in a model. This is a special excitation type, and it is subject to a number of limitations (see section 14.15). The impressed current excitation is defined on the *Add impressed current* dialog (shown in Figure 6-40). This dialog is launched by selecting the the *Source/Load* tab and clicking on the *Impressed current* button or by right-clicking on the *Excitation* tab and selecting *Impressed current*.



Figure 6-39: The *Impressed current* button

The position of the start and end points of the current segment, as well as the magnitude and phase (in degrees) of the current at the start and end points can be defined. The radius of the impressed current must also be specified in the *Radius* field. This radius value may not be zero, and the unit is determined by the model unit.

A *Connect the end point to the closest triangle vertex* option is provided. When checked, the end position of the impressed current cannot be specified, but will be determined based on the closest triangle mesh vertex available during the solution. This is useful when the impressed current should terminate (and be connected to) the model geometry. There must be a mesh vertex at the connection point, but it is not always possible to predict the exact location of the mesh vertex to which the excitation should be connected. When using this option, care should

Figure 6-40: The *Add impressed current* dialog

be taken to confirm that the excitation does indeed connect at the required triangle vertex. (This can be visually confirmed by running PREFEKO and observing the impressed current excitations in POSTFEKO.)

Impressed current excitations are shown as red arrows in the 3D view. The preview arrow is located between the start and end points, and has the specified radius of the impressed current. When the *Connect the end point to the closest triangle vertex* option is used, the impressed current is represented by a red sphere at the starting point of the impressed current. The radius of the sphere is determined by the radius of the impressed current.

6.6 Setting ports, sources and loads on geometry

Ports and loads are mathematical representations for where energy can enter or leave a model. A port can either serve as a source, or as a sink. Loads and excitations may be applied to a port in a variety of ways to best model the equivalent real-world problem.

In CADFEKO, voltage sources and discrete loads are not applied directly to the model geometry or mesh, but rather to ports which must be defined on the geometry or mesh before adding the required source or load. Ports effectively define the locations where excitations and loads may be attached to the geometry.

6.6.1 Ports

There are six types of ports that can be defined in CADFEKO: Wire ports, Edge ports, Microstrip ports, Waveguide ports, FEM line ports and FEM modal ports. Each has a different application and advantages or restrictions. The FEM ports, for example, may only be applied in FEM regions.

Waveguide ports may be used with most solution methods, but only have certain predefined shapes (i.e. rectangular, circular and coaxial). Using the correct port for a particular model can help reduce resource requirements and provide more accurate results. Figure 6-41 shows the ports that can be created.

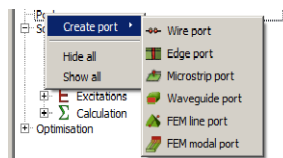


Figure 6-41: List of ports that can be added. The list is obtained by right-clicking on the *Ports* entry in the model tree.

All ports (irrespective of type) are listed under *Ports* in the tree. Each port contains sub-instances, indicating the geometry- and/or mesh-linked manifestations of that port. Generally ports are created on geometry items and such ports initially contain only a geometry instance. When the geometry part to which the port is applied is meshed, a mesh instance is automatically created. If the mesh part to which a port is connected is deleted, then the mesh instance of that port will automatically be deleted if the port also has at a geometry instance. If the port does not have a geometry instance, the port must first be removed before the mesh part can be deleted.

The tree representation for each port contains separate icon representations for the geometry instance and/or each of the mesh instances of the port. (It is not possible for a port to have multiple concurrent mesh instances - this was possible in previous versions of CADFEKO under certain special circumstances. If a model created in a previous CADFEKO version which has multiple mesh instances is opened in a newer CADFEKO version, separate mesh ports will be automatically created for each of the mesh port instances and a message indicating the changes provided.) It is generally not necessary to view the separate instances of a port themselves, though if the port is suspect (see section 4.4) or contains a fault (see section 5.1.4), the suspect or fault reason may only be displayed in the tool-tip for the individual instances.)

Ports may be created directly on meshes. This option is only available from the *Mesh* → *Create port* menu and should only be used for imported meshes or in cases where the geometry is no longer available. If a port is created directly on the mesh, it will only contain a mesh instance. Ports created on meshes derived from CADFEKO geometry will be automatically removed if the geometry is re-meshed, and the mesh is replaced.

If a port contains a geometry instance, selecting *Properties* on the pop-up menu of the main port item opens the properties of the geometry instance. If the port contains only a single mesh instance, this opens the properties of that instance. The properties of any specific instance may be opened by expanding the port in the tree and selecting *Properties* on the pop-up menu of that instance.

The *Label* field is used to uniquely identify each port. The label may be changed using the *Rename* entry in the pop-up menu of the main port item or the *Properties* dialog of any geometry or mesh instance of the port.

Some of the mesh refinement tools may remove elements on which ports are set. The affected ports are then marked suspect and must be edited (and correctly re-assigned to geometry / mesh items) to correct this setting.

The geometry and mesh instances associated with a port can be individually deleted. A port is deleted when its last instance is deleted. If a port is applied to a mesh part (as described above)

or the geometry instance is deleted after meshing, that port contains only mesh instances. Any mesh part that a port refers to cannot be remeshed unless the port is removed completely.

Note that only the mesh instances of ports are written to the *.pre and *.cfm files.

The convention is that if a positive voltage is applied to a port, the current flows through the port from the negative side to the positive side.

Wire ports

Wire ports are created on wire edges, i.e. free edges that do not form a face boundary. Wire ports may be created in a number of ways

1. Select the *Source/Load* tab and click on the *Wire port* button
2. Select the wire in the details tree and select *Create port* → *Wire port* from the pop-up menu
3. Right-click on the *Ports* tree entry and select *Create port* → *Wire port* from the pop-up menu



Figure 6-42: The *Wire port* button

The *Create wire port (geometry)* dialog is shown in Figure 6-43. During the creation/editing of a port, a preview of the current position of the port is indicated with a small sphere in the 3D view.

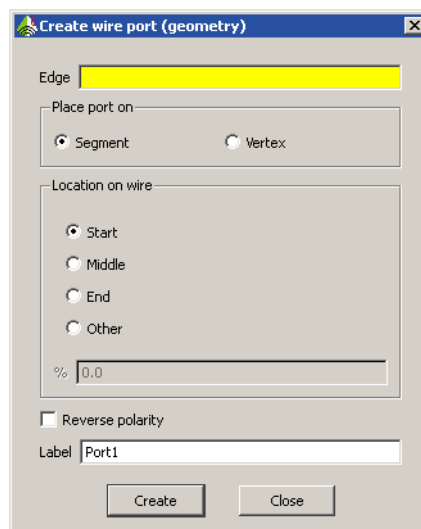


Figure 6-43: The *Create wire port* dialog for geometry instances

Edge: Determines which wire the port is located on. This can be changed by selecting an edge in the 3D view or details tree. To select an edge in the 3D view, the edge selection mode must be active. (The selection mode is changed to edge selection when the dialog is opened, but the user may have changed it while the dialog is open.) If the dialog is opened from the right-click pop-up menu of a single wire in the details tree, the selected edge label is automatically entered into the *Edge* field.

Place port on: When the wire is meshed, the mesh instance of the port can be placed on a single segment or on the vertex between two segments. This group allows specification of how the port should be created. *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 influence on the input impedance.

Location on wire: This group is used to specify where on the selected wire the port is to be located. If *Segment* points are specified at the end points, the first (specially shortened) segment on this end is used. With *Other* selected, the user may specify an arbitrary position along the wire in terms of the position as a percentage of the total wire length. 0% is interpreted as placement at the start point and 100% at the end point. If the wire is modified after the port has been applied in such a way that there is still only one wire with this label, the port will maintain the same relative position along the wire. Thus if the port was one third from the end of a wire and the wire is shortened to work at a different frequency, the port remains one third from the end of the shorter wire. The absolute position of a port may be fixed by entering a named point or a 'pt' expression (see section 4.2.2) in the % field. The port is then located at the projection of this point onto the wire. If the wire is modified after application of the port, the point will remain as close as possible to this absolute position. This field accepts standard point entry (see section 3.2.2)

Reverse polarity: The option to reverse polarity of the port can be set by checking *Reverse polarity*.

If a model is modified such that the wire label to which a port is applied no longer exists, or if the wire is divided into multiple parts during a geometry operation, the port may be marked suspect. The user must then edit it and select the appropriate single edge to which the port should apply. If the wire is split into multiple edges in the case where the port position has fixed using a named point, the port will be located on the edge where the projection is closest to this point where possible.

Vertex ports are now allowed to be set on the end of wires that are connected to infinite ground planes and UTD plates.

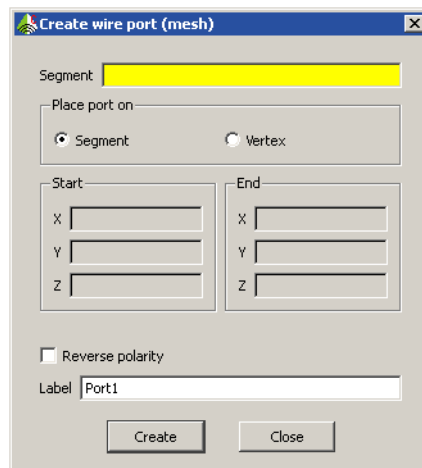
Wire ports can be set directly on the mesh by selecting the *Source/Load* tab and clicking on the *Wire port* button. Note that ports should generally be applied to the geometry, mesh instances should only be directly defined on imported and unlinked meshes.

The *Create wire port (mesh)* dialog (shown in Figure 6-44) is similar to the *Create wire port* dialog for geometry. The *Locate port on* group allows direct *Segment / Vertex* selection. When the *Segment* or *Vertex* field is active (according to the *Locate port on* selection), the user can select a segment/vertex by clicking in the 3D view. (The correct selection mode (see section 3.2.2) must be active.) The *Segment/Vertex* field reflects the wire label to which the port will be applied, while the start and end point coordinates of the segment or the coordinates of the vertex are displayed in the read-only fields below this.

In the 3D display the segment on the positive side of the port is indicated with a red cylinder and the one on the negative side with a blue cylinder. Ports which are set on vertices (as opposed to segments) are identified by an additional small sphere at the vertex. If a vertex port is at a connection to a surface, only the segment display is coloured.

Edge ports

Edge ports are created along an edge defining the boundary between two sets of faces. All the faces referenced in the port definition must belong to the same part for the port definition to be

Figure 6-44: The *Create wire port (mesh)* dialog for mesh instances

valid.

Select the *Source/Load* tab and click on the *Edge port* button or by right-clicking on *Ports* in the tree and selecting *Edge port*.

Figure 6-45: The *Edge port* button

The *Create edge port (geometry)* dialog — shown in Figure 6-46 — contains two lists of labels. The *Positive faces* list defines the set of faces on the positive side of the port, while the *Negative faces* list defines the set of faces on the negative side.

When the dialog is launched, the *Positive faces* list is populated based on the current selection. If the current selection contains a face selection, then the list is automatically populated with all of the selected face labels. If the selection contains edges, then the labels of all of the associated faces are also listed. If the selection contains only parts, then no faces are added to the initial list. (Note that this initial list of faces may contain labels of faces from different parts, and is simply intended as a starting point for the port definition. Some labels will need to be moved to the *Negative faces* list and other labels may need to be added or entirely removed to specify a valid port definition).

Faces that use the UTD solution method may also be used in the definition of an edge port, but one side of the port must consist only of one valid UTD face, and all faces on the other side of the defined port must be standard MoM faces.

Where an infinite ground plane is present in the model, any edges that lie in the plane may be excited with respect to that ground plane. The *Connect to infinite ground* checkboxes on the dialog allow the positive or negative side of the port to connect to the infinite ground.

Add/Remove: The user may add and remove face labels from each of the lists as required.

Move to negative/positive face: Faces can be switched between the lists by clicking the button between the lists (double-clicking on an entry will also shift that entry to the other list)

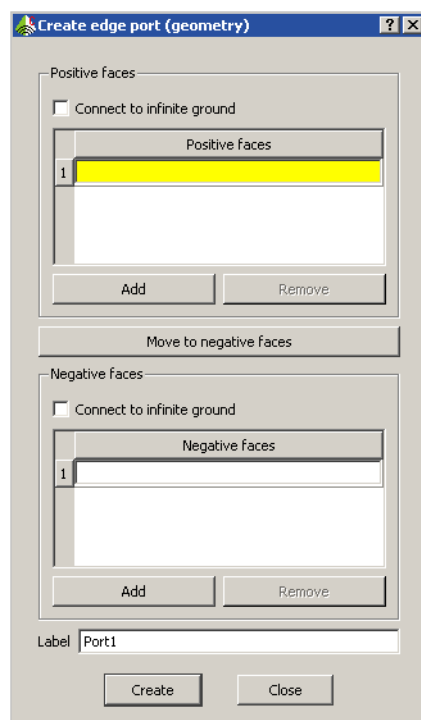


Figure 6-46: The *Create edge port* dialog for geometry instances.

Connect to infinite ground: When this option is selected, the infinite ground will form the positive/negative side of this edge excitation (according to the selection). Only one side of the excitation may be connected to ground at a time, and all faces in the list on the other side of the port must have edges that lie in the plane of the infinite plane.

Only one entry in either of the lists of face labels may be selected at a time. The background of the entry that has the selection focus is coloured yellow. While one of the list entries has focus, clicking on a face in the 3D view or on the details tree representation of a face will enter the face's label into the selected field. In this way, the relevant face labels may be added to the lists to complete the port definition.

During creation/editing of an edge port, the boundary between the two sets of faces (the actual location of the port) is highlighted with a grey cylinder in the 3D view. The edge is marked on the side of the positive faces with a second, red cylinder while the negative faces are marked with a third, blue cylinder.

Note that the edge on which a port is defined need not be straight — it may even close on itself, for example to create a port at the centre of a thick dipole modelled using a cylinder meshed into triangle elements rather than wire segments.

For the edge port, the mesh instance creation dialog (launched from *Geometry* → *Create port* → *Edge port* on the main menu) and the 3D display are exactly the same as for the geometry instance.

Microstrip ports

Microstrip ports are used to represent feed lines on microstrip structures. These ports are specified on an edge or set of edges that form a continuous, straight, horizontal (i.e. lie in a constant

z plane in the global coordinates) edge bordering only one face. In addition, the model must contain a planar dielectric substrate with a conducting ground plane at the bottom.

Select the *Source/Load* tab and click on the *Microstrip port* button.



Figure 6-47: The *Microstrip port* button

The *Create microstrip port (geometry)* dialog (shown in Figure 6-48) contains a single list of edges (similar to the face lists of the *Edge port* discussed above).

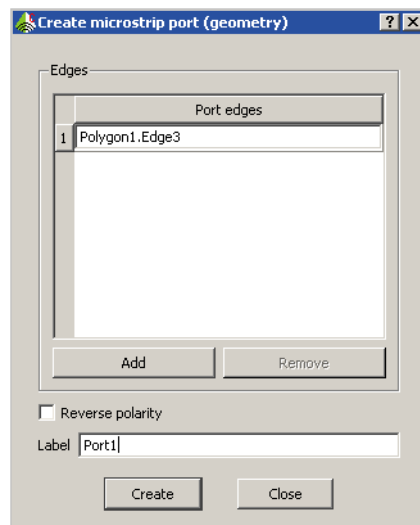


Figure 6-48: The *Create microstrip port* dialog for geometry instances.

This port is applied by selecting the *Source/Load* tab and clicking on *Microstrip port* or by selecting a set of edges and selecting *Create port* → *Microstrip port* from the pop-up menu in the details tree. The port is previewed and displayed in the 3D view in a similar fashion to the edge port. The orientation or polarity of the microstrip port is shown in the 3D view port preview. If the excited edge is highlighted blue, the positive side of the port is connected to the edge and the negative side to the ground plane. If the edge is highlighted red, then the negative side of the port is connected to the edge. The orientation of the port can be reversed by checking or un-checking the *Reverse polarity* checkbox.

A microstrip port may be created directly on the mesh. The mesh microstrip port dialog is accessed by selecting the *Source/Load* tab and clicking on the *Microstrip* button. On the mesh instance dialog (shown in Figure 6-49) the edge to which the port is to be applied is specified in terms of its start and end points. These must be mesh vertices and must be selected by clicking on the relevant vertices in the 3D view while the *Start vertex* and *End vertex* fields are active. (Mesh vertices must be displayed.) The vertex fields then display the label of the elements associated with the appropriate vertices. The display is the same as for the geometry instance.

Waveguide ports

Waveguide ports are used to define the planes of excitation for waveguide structures. Three basic waveguide cross-sections are supported namely:

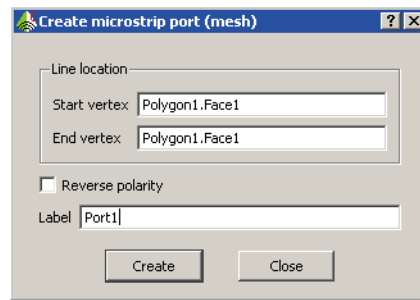


Figure 6-49: The *Create microstrip port* dialog for mesh instances

1. Rectangular waveguide
2. Circular waveguide
3. Coaxial waveguide

These ports are specified on a single face with the correct shape. In order to apply a port to a face, the face must be flat, must not contain any internal edges, must not form the boundary of a PEC or dielectric region, and must not have any special material properties (e.g. Dielectric coating) or be solved using special solution methods (e.g. UTD).

Waveguide ports may be created by selecting the *Source/Load* tab and clicking on the *Waveguide port* button.



Figure 6-50: The *Waveguide port* button

The *Create waveguide port* dialog (shown in Figure 6-51) allows the selection of a single face to which the port is to be applied. When a face is selected, a propagation direction and reference direction for the port are automatically chosen and displayed in the port preview. If these are not as desired, they can be changed by selection the *Propagation direction opposite to normal* and by setting the relevant rotation angle in the *Rotate reference direction* list respectively.

On the *Advanced* tab of the *Create waveguide port dialog* (shown in Figure 6-52) certain special settings may be chosen. By checking the *Manually set the number of modes considered* option, the maximum mode indices can be set. If this is not selected, then FEKO will automatically determine the maximum number of modes to consider. By selecting *Manually set the reference vector* the automatically chosen reference vector for the port can be overridden. In this case, the reference vector is specified from the centre of the port face in the direction of the specified vector.

Waveguide ports may also be applied directly to mesh faces. In order to do this, select the *Source/Load* tab and click on the *Waveguide port* button. Specification of a waveguide port on the mesh is very similar to the specification of a waveguide port on geometry in that CADFEKO automatically calculates the relevant port shape based on the shape of the selected mesh face (the mesh must represent one of the supported waveguide shapes sufficiently or CADFEKO will give an error). For direct application of a waveguide port to the mesh, the reference direction of the port should be manually specified in terms of a vector from the centre of the excitation face.

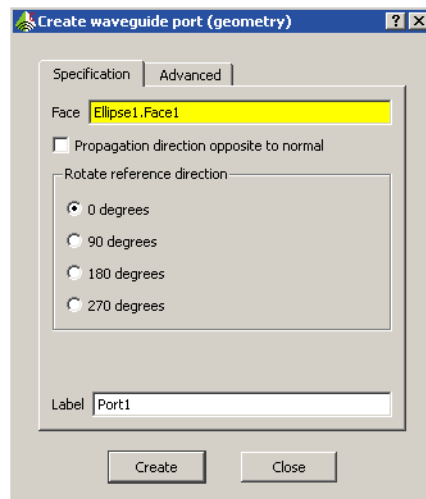


Figure 6-51: The *Create waveguide port* dialog for geometry instances.

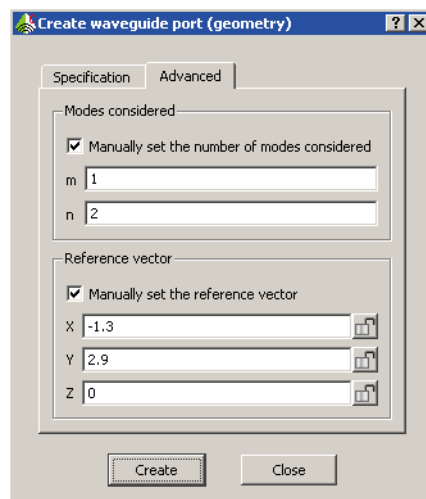


Figure 6-52: The *Advanced* tab of the *Create waveguide port* dialog for geometry instances.

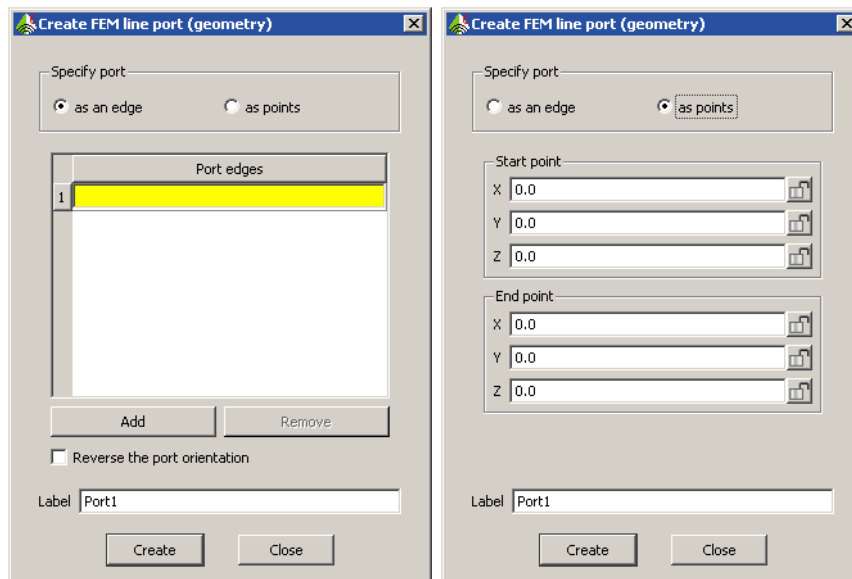
The waveguide port differs from other geometry-linked ports in CADFEKO in that when no excitation is applied to the port it will still be considered in the FEKO solution. A waveguide port without any waveguide excitation(s) (see section 6.6.2) applied will be considered as an ideal waveguide termination in all parts of the solution (i.e. all energy that propagates into the waveguide port will be absorbed and not reflected).

FEM line ports

FEM line ports are used to define the location of impressed current source excitations and loads in the FEM region. A FEM line port may be added by selecting the *Source/Load* tab and clicking on the *FEM line port* button or by right-clicking on the *Ports* branch of the tree and selecting *Create port* → *FEM line port*.

The *Create FEM line port (geometry)* dialog is shown in Figure 6-54.

The port position may be specified in one of two ways — based on the location of a free edge (or a connected set of free edges that form a continuous straight line) inside of a dielectric region in

Figure 6-53: The *FEM line port* buttonFigure 6-54: The *Create FEM line port (geometry)* dialog for the definition of a FEM line port in the FEM region

the model, or based on coordinates defining the beginning and end points of the FEM line port (in global coordinates).

If the port location is specified based on an edge (or set of connected edges) the orientation of the port is determined automatically. The orientation can be visually determined from the 3D view preview of the port, and may be reversed if required by checking the *Reverse the port orientation* option. If the geometry on which the port is defined is modified (translated or scaled etc.) the FEM line port location and size will be adjusted accordingly. If the FEM line port definition is based on coordinates, the user must ensure that the coordinates remain correct.

During meshing, all edges that are associated with a FEM line port definition will be excluded in the mesh, but are used to determine the start and end points of the FEM line port. The FEM line port has a mesh instance which may be directly defined or edited (after it is generated from a geometry port during meshing). The mesh instance of the FEM line port is defined based on either the start and end coordinates, or by attaching the start and end point of the FEM line port to specific mesh vertices within (or on the boundary of) a tetrahedral FEM mesh. Mesh instances of the FEM line port that are generated based on geometry ports during meshing will be defined based on mesh vertices where possible, or based on the coordinate values directly.

The *Create FEM line port (mesh)* dialog is shown in Figure 6-55.

Where the mesh instance of a port is defined based on mesh vertices it will be maintained through operations on the mesh (scaling, translation etc.). Where the mesh instance of the FEM line port is based on coordinate points, the user must ensure that the port definition is as intended. (Note — during meshing, mesh instances of FEM line ports that are based on coordinate point definitions are generated with the label '*user defined*' and do not inherit their name from the geometry instance of the port. This indicates that CADFEKO will not maintain this port position

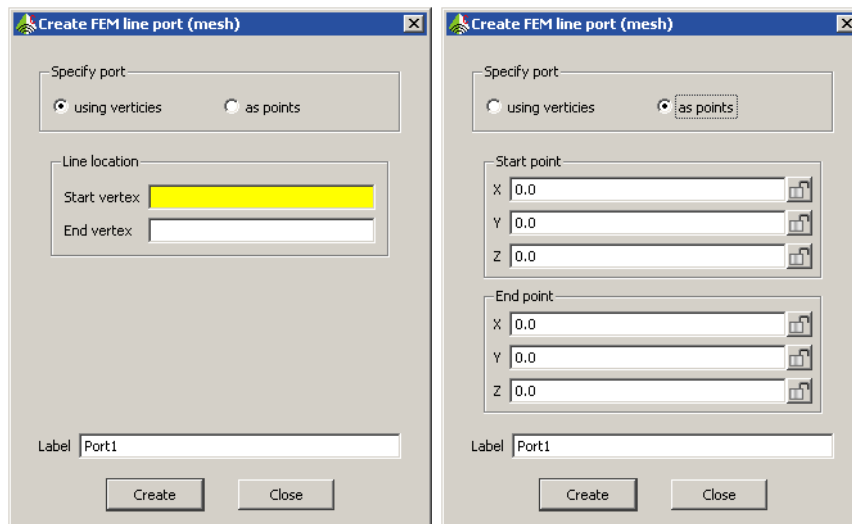


Figure 6-55: The *Create FEM line port (mesh)* dialog for the definition of a FEM line port in a tetrahedral mesh region

through mesh manipulations.)

Loads (see section 6.6.3) and current sources (see section 6.6.2) may be applied to a FEM line port and FEM line ports may be included in S-parameter calculations (see section 6.12.2).

If a model created in a version of CADFEKO released before the Suite 5.4 FEKO release, any *FEM current source* excitations in the model will automatically be converted into an equivalent FEM line port with an applied current source excitation (see section 6.6.2). This should not influence the solution, but the line port will have to be maintained by the user through any changes made to the model.

FEM modal ports

FEM modal ports are used to apply a modal port boundary condition on the boundary of a finite element (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 may 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 excitations in the model. In order to apply a FEM modal port to a face, the face must be flat and on the boundary of a FEM region.

A *FEM modal port* may be added by selecting the *Source/Load* tab and clicking on the *FEM modal port* button or by right-clicking on the *Ports* branch of the tree and selecting *Create port → FEM modal port*.



Figure 6-56: The *FEM modal port* button

The *Create FEM modal port* dialog (shown in Figure 6-57) allows the specification of the port either as a list of faces or as points.

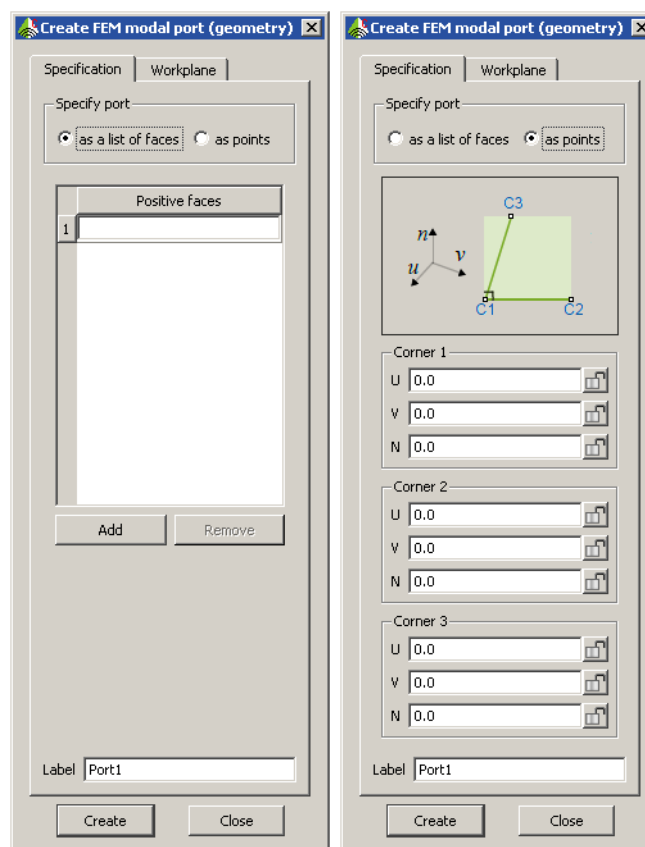


Figure 6-57: The *Create FEM modal port* for geometry instances.

6.6.2 Sources

When adding an excitation, a new item will appear under the *Calculation* part of the tree to indicate that source data will be calculated by simply adding an excitation. Note that when only the reflection coefficient or input impedance is required, it is not required to add an S-parameter request.

Voltage sources

The voltage source excitation can be applied to any wire, edge, line, network or transmission-line port to define the excitation of that port. Any port of these types to which a voltage source (or load) is not applied will have no effect on the solution — though it may be used in the definition of a port for S-parameter calculations.

In order to define a voltage excitation, select the *Source/Load* tab and click on the *Voltage source* button or right-click on *Excitation* in the tree and select *Voltage source*.



Figure 6-58: The *Voltage source* button

A voltage source may also be added by right-clicking on a defined port in the *Ports* tree and

selecting *Add voltage source* (in this case, the port label is automatically selected in the *Port* field on the voltage source dialog — the port selection may, however, be changed freely).

The voltage source creation dialog is shown in Figure 6-59. The dialog contains a *Port* field which lists all the ports in the model to which a voltage source may be applied. The list of valid ports will include all defined ports, and when the excitation is applied, it will be reflected in all of the mesh and geometry instances of that port.

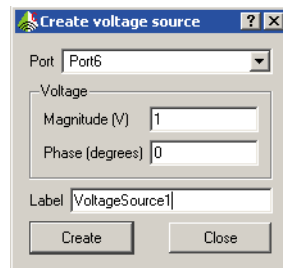


Figure 6-59: The dialog for the application of a voltage source to a port.

The input fields for the voltage magnitude and phase define the excitation to be applied. The specified voltage gives the potential difference between the positive side of the port relative to the negative side. A positive voltage will result in positive current flowing out of the positive side and into the negative side of the port.

Current sources

The current source may be applied to a line port (see section 6.6.1) in a dielectric region to be solved with the FEM in order to realise an impressed current excitation (this is equivalent to the *FEM current source* excitation available in FEKO versions before the Suite 5.4 release).

Add a current source by selecting the *Source load* tab and clicking on the *Current source* button.



Figure 6-60: The *Current source* button

The *Add current source* dialog is shown in Figure 6-61. The excitation is specified based on a constant current (*Magnitude* and *Phase*) that is applied in a filament along the line defined by the line port to which it is attached. The line port should therefore be short relative to a wavelength in the medium in which it is located.

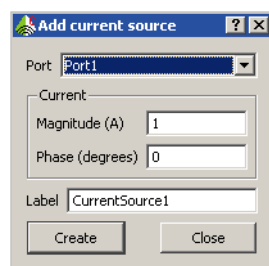


Figure 6-61: The dialog for the application of a current source to a line port.

An intrinsic limitation of the impressed current source 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.

Waveguide excitations

Specific waveguide mode excitations can be applied to waveguide ports (see section 6.6.1) that have been defined in the model. A waveguide mode excitation is added by selecting the *Source/Load* tab and clicking on the *Waveguide excitation* button or by right-clicking on *Excitations* in the tree and selecting *Waveguide excitation*.



Figure 6-62: The *Waveguide excitation* button

The *Add waveguide excitation* dialog is shown in Figure 6-63.

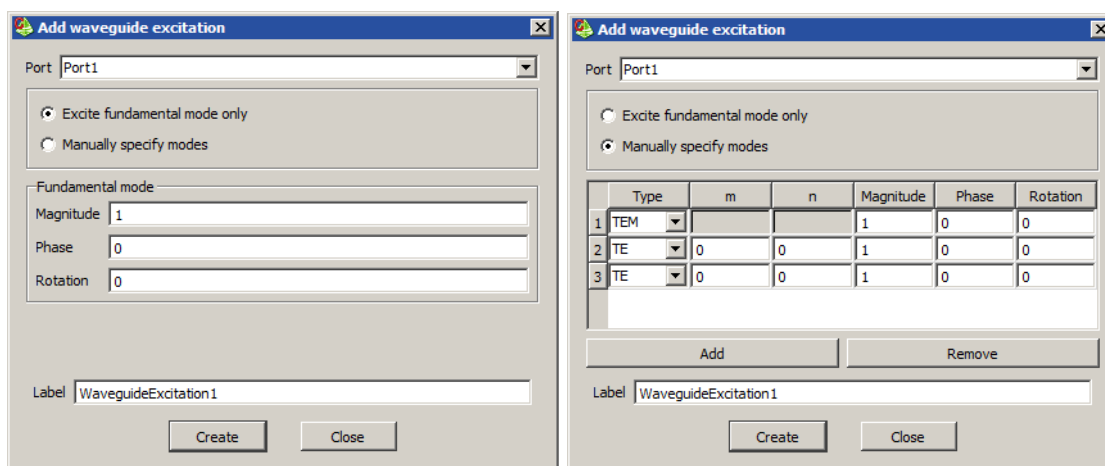


Figure 6-63: The dialog for the application of a waveguide excitation to a waveguide port (options for manual and fundamental mode selections shown).

The *Excite fundamental mode only* option should be selected if the fundamental propagating mode at the solution frequency must be excited (the FEKO kernel will compute this mode based on the waveguide size and cross-section during the solution). The *Magnitude* and *Phase* (in degrees) of the excitation, as well as the physical *Rotation* of the mode (also in degrees) can be specified. The rotation of an excited mode is relative to the reference direction of the port to which the mode is applied and for rectangular ports, the rotation must be a multiple of 90 degrees.

If a specific mode or a specific set of modes are to be excited, the *Manually specify modes* option should be selected. In this case, the type and indices (where applicable) as well as the magnitude, phase and physical rotation of the mode(s) must be entered in the table. Modes may be added or removed from the list using the *Add* and *Remove* buttons. It is important to note that if multiple modes are applied to a single waveguide port (as part of one waveguide excitation definition or in separate definitions) the modes will all be applied simultaneously during the solution phase. When using waveguide ports as part of an S-parameter calculation, the waveguide excitations that are applied to the port will not influence the S-parameter calculation (see

section 6.12.2). During the S-parameter calculations, only the mode(s) specified as part of the S-parameter request will be considered.

FEM modal excitations

The FEM modal excitation can be applied to FEM modal ports (see section 6.6.1) that have been defined in the model. A FEM modal excitation is added by selecting the *Source/Load* tab and clicking on the *FEM modal excitation* or by right-clicking on *Excitations* in the tree and selecting *FEM modal excitation*.



Figure 6-64: The *FEM modal excitation* button

The *Add FEM modal excitation* dialog is shown in Figure 6-65.

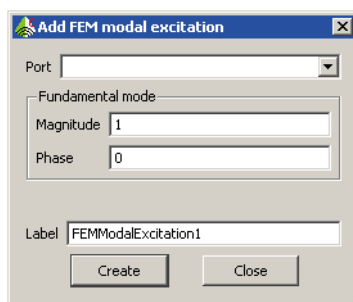


Figure 6-65: The dialog for the application of a FEM modal excitation to a FEM modal port.

If the FEM modal excitation is defined, the modal port will be excited with the fundamental mode of the associated long guided wave structure of the FEM modal port. When no excitation is defined, the modal port will act as a passive port (sink) for fields incident on the port.

6.6.3 Loads

An impedance load may be applied to any wire, edge, line, network or transmission-line port.

The *Create load* dialog (shown in Figure 6-67) is launched by selecting the *Source/Load* tab and clicking on the *Add load* button or by double clicking on *Loads* in the tree, or by right-clicking on a specific port and selecting *Add load* (if this last method is used, then the port is automatically selected in the ports list on the *Create load* dialog — though the selection may be changed freely).



Figure 6-66: The *Add load* button

There are three types of loads that may be specified.

Complex impedance: A frequency-independent load consisting of a constant real and imaginary part. This load type can be applied to wire, edge, stripline, network or transmission-line ports.

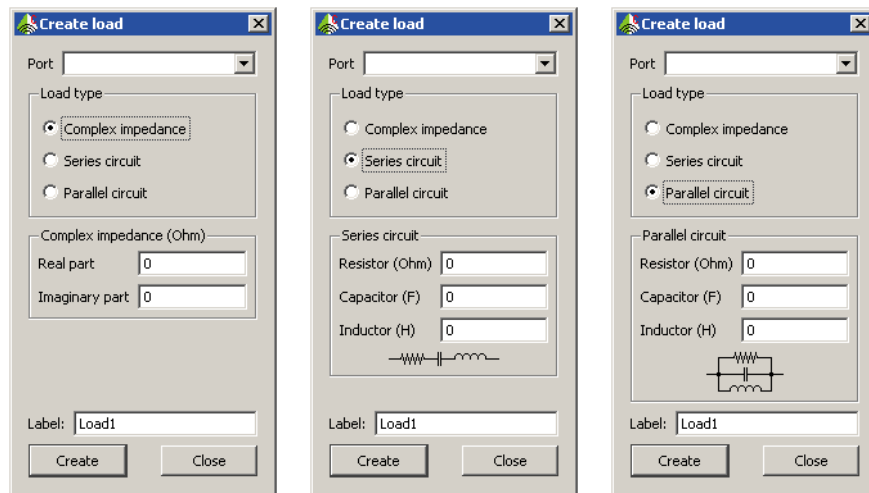


Figure 6-67: The impedance load dialog showing the options for the 3 types of loads that may be applied to a port.

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 wire ports. The load impedance is given by

$$Z_s = R + j\omega L + \frac{1}{j\omega C}. \quad (6-11)$$

Parallel circuit: A frequency dependent load consisting of a resistor (R), capacitor (C) and inductor (L) connected in parallel. This load can only be applied to wire ports. The load impedance is given by

$$Z_p = \frac{1}{\frac{1}{R} + \frac{1}{j\omega L} + j\omega C} \quad (6-12)$$

where the resistance or inductance is taken as infinite when set to 0 (i.e. it does not contribute to the impedance).

If multiple loads, or loads and sources are applied to the same port, they are placed in series. This also applies to the *Parallel circuit* load — only the three elements of the load are connected in parallel.

6.7 Infinite planes — Planar Green's functions and ground planes

FEKO can include the effect of an infinite ground plane and/or planar substrates.

Select the *Construct* tab and click on the *Infinite structures* button and select *Infinite plane* from the dropdown menu or double-click *Infinite planes* under *Solution* in the tree to open the *Infinite planes / ground options* dialog (shown in Figure 6-69).



Figure 6-68: The *Infinite plane* button

No ground (homogeneous free space medium): This is the default for any new model. When this option is selected, the model is solved in an homogeneous environment filled with the *Free space* medium. The properties of the *Free space* medium may be edited in the media tree as required.

Perfect electric (PEC) ground plane at $z=0$: This provides for the inclusion of an infinite ground plane that coincides with the $z = 0$ plane (in the global coordinate system). Perfect electric conductor is used as the ground medium. In the reflection coefficient approximation, a reflected component is added to each field. This technique is a lot faster than the exact Sommerfeld integral method, but is also less accurate. For real grounds all conducting structures must be more than a tenth of a wavelength above the ground. (For perfect ground, structures may be connected to the ground, but not below it.)

Perfect magnetic (PMC) ground plane $z=0$: This provides for the inclusion of an infinite ground plane that coincides with the $z = 0$ plane (in the global coordinate system). Perfect magnetic conductor is used as the ground medium. In the reflection coefficient approximation, a reflected component is added to each field. This technique is a lot faster than the exact Sommerfeld integral method, but is also less accurate. For real grounds all conducting structures must be more than a tenth of a wavelength above the ground. (For perfect ground, structures may be connected to the ground, but not below it.)

Homogeneous half space in region $z < 0$ (reflection coefficient approximation): This provides for the inclusion of an infinite ground plane that coincides with the $z = 0$ plane (in the global coordinate system). The medium is selected from the list in the *Ground medium* group. This list contains all of the user-defined *Dielectric* media. In the reflection coefficient approximation, a reflected component is added to each field. This technique is a lot faster than the exact Sommerfeld integral method, but is also less accurate. For real grounds all conducting structures must be more than a tenth of a wavelength above the ground. (For perfect ground, structures may be connected to the ground, but not below it.)

Homogeneous half space in region $z < 0$ (exact Sommerfeld integrals): This provides for the inclusion of an infinite ground plane that coincides with the $z = 0$ plane (in the global coordinate system). The ground medium is selected from the list in the *Ground medium* group. This list only contains all of the user-defined *Dielectric* media (Perfect conductors are not supported in the Sommerfeld ground). The Sommerfeld integral solves the exact boundary condition using the appropriate Green's function. This method supports conducting structures inside the ground, but no mesh element that crosses the surface of the ground may be connected to an element below the ground. For example, partly buried wires are allowed as long as there is a vertex at the position where the wire passes through the surface of the ground.).

Planar multilayer substrate: This adds an infinite planar multilayer substrate to the model. The user can define the *Medium* and *Thickness* of each layer in the planar structure. A ground plane may be specified at the bottom of each layer by selecting *Bottom* from the *Ground plane* dropdown list. If a groundplane is enabled at the bottom of a layer, it will be indicated by an orange line underneath *Media preview*. The model is then solved using planar Green's functions to account for the layered structure. The *Medium* lists contain all the user defined *Dielectric* media. Layers can be added and removed with the *Add* and *Remove* buttons. Conducting planes can be added on top of layer 1 or below the bottom layer by checking the appropriate fields in the *Conducting ground plane* group. Note that *Layer 0* is infinitely thick and it is ignored if there is a conducting plane at the top of the layered structure. The thickness of the bottom layer is only relevant if there is a conducting plane at the bottom. Otherwise this layer extends towards infinity. The planar structure is always orthogonal to the global z -axis, the vertical position can be modified by specifying the *Z-value* at the

top of layer 1. Planar substrates may be used without ground planes in complex (layered) models (for example models of real earth); with a ground plane at the bottom (for example in microstrip applications) and with both conducting top and bottom planes (for example in stripline applications).

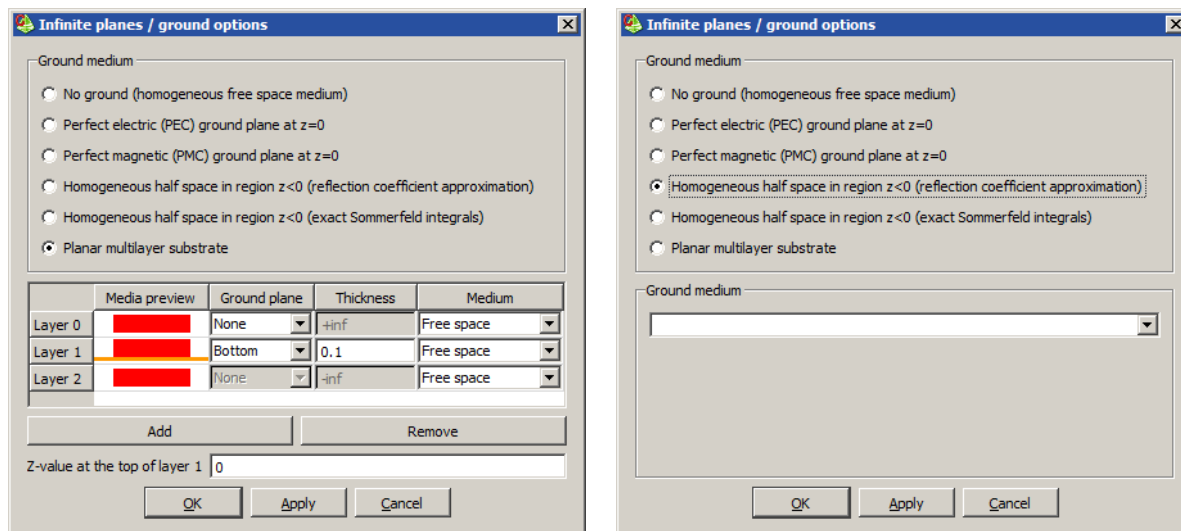


Figure 6-69: The infinite planes dialog for the application of ground planes and planar substrates

The infinite ground plane is displayed in the 3D model view.

6.8 Symmetry

CADFEKO allows the specification of planes of symmetry that may be used to accelerate and reduce the memory requirements for the solution of a problem.

To add symmetry to a model select the *Solve/Run* tab and click on the *Symmetry* button.



Figure 6-70: The *Symmetry* button

The *Symmetry definition* dialog is given in Figure 6-71.

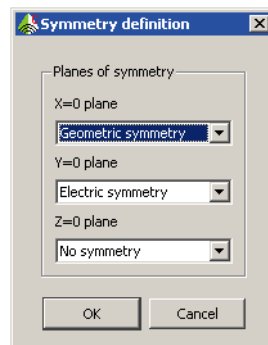


Figure 6-71: The *Symmetry definition* dialog

During meshing CADFEKO will validate that the geometry to be meshed does indeed adhere to the specified model symmetry (both geometric symmetry as well as symmetry of excitation and loads where magnetic/electric symmetry is used). If the model is found not to adhere to the specified model symmetry then CADFEKO will abort the meshing process, and provide a list of the parts of the model that break the symmetry.

When writing the mesh to the *.cfm file (during save or exporting from the *File* → *Export* → *CADFEKO Mesh* dialog) the symmetry of the mesh is verified. If the mesh does not adhere to the symmetry (both geometric and excitation/loads), then the *.cfm file is written, but the symmetry is excluded. In this case a warning is given with the reasons why the symmetry cannot be applied.

It is possible that the geometry and/or mesh of a model appear to be symmetrical but the CADFEKO symmetry tests indicate that this is not so. In this case it may be more appropriate to delete half of the model, and use the mirror operation to ensure a symmetrical model rather than trying to resolve each asymmetry. This is particularly relevant when working with imported CAD geometry or imported meshes where there may be slight differences with respect to tolerances.

CADFEKO indicates the current symmetry of the model in the 3D view (as shown in Figure 6-72 for three planes of symmetry). This preview is always shown while the *Symmetry definition* dialog is open, but may be deactivated by toggling the *Show symmetry planes* icon in the relevant 3D view. The preview is coloured according to the symmetry type (green for geometric symmetry, orange for electric symmetry and grey for magnetic symmetry).

When using magnetic or electric symmetry planes, all loads, excitations, media properties etc. must be defined such that the model properties are identically symmetrical. For geometric planes

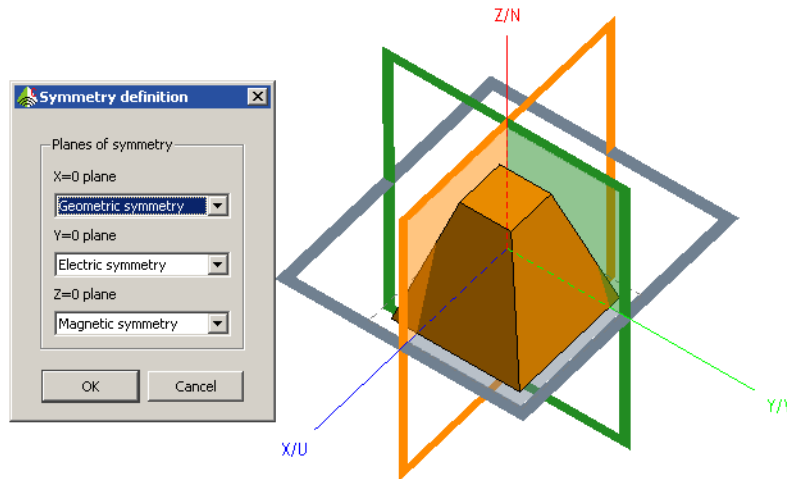


Figure 6-72: Preview of symmetry planes in the 3D view

of symmetry, the geometry must be symmetric though symmetrical loading, excitation and media properties are not required. It is important to note that geometry-linked ports (wire ports, waveguide ports, edge ports, line ports) may affect meshing and must therefore always have symmetrical counterparts around any defined symmetry plane, irrespective of the symmetry plane type.

6.9 Periodic boundary conditions

Periodic boundary conditions may be specified in CADFEKO. Like symmetry, periodic boundary conditions are considered a model setting and may be accessed by selecting the *Infinite structures* tab and selecting *Periodic boundaries* from the dropdown menu.

Figure 6-73: The *Periodic boundaries* button

The dialogs for 1D and 2D periodic boundary conditions are shown in Figure 6-74.

The definition of the unit-cell of the periodic boundary condition solution is based on vectors. For the 1D case, the start- and end-point of a single vector is required. Periodicity is then defined based on two planes passing through these start and end points, and normal to the vector formed between them. The vector used to define 1D periodicity can have any orientation, but must have a non-zero length. For the 2D 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 on which both vectors lie (see Figure 6-75). The vectors that define the unit-cell for 2D periodicity must have non-zero length, and cannot be oriented in the same direction.

CADFEKO allows the specification of the phase shift to be applied in the direction of each of the vectors defining the unit-cell. The specified values for the phase-shift are only used if a plane-wave excitation is not present. (If phases are specified and a plane wave excitation is present in the solution, the FEKO kernel will return an error during solution.)

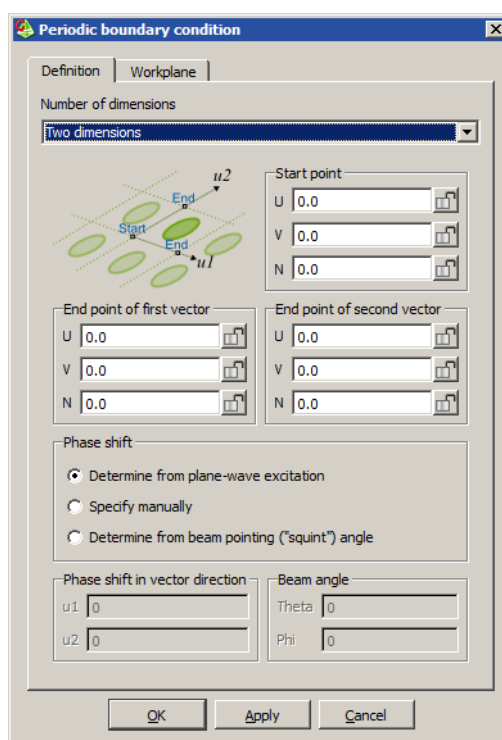


Figure 6-74: The Periodic boundary condition dialog (1D and 2D)

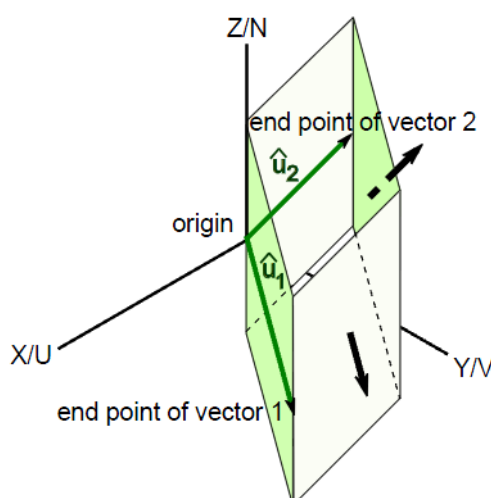


Figure 6-75: The Periodic boundary condition for 2D periodicity

For array modelling using periodic boundary conditions, the beam (squint) angle can be specified by defining the theta and phi angle. The phase along the periodic lattice vectors will then be computed automatically to ensure the specified beam direction.

6.10 Numerical Green's function (NGF)

CADFEKO allows the numerical Green's function (NGF) to be applied to a MoM problem consisting of a dominant, fixed static part and a smaller dynamic part. A user may elect to save the static interaction matrix to a *.ngf file. The CPU time may then be reduced by reusing the *.ngf file to obtain solutions for various dynamic domains

The Numerical Green's function may be activated by selecting the *Solve/Run* tab and clicking on the *NGF* button.



Figure 6-76: The *Numerical Green's function* button

Note that domain decomposition is recommended for MoM models consisting of a large static part and a smaller dynamic part. The static part must be large in relation to the dynamic part to obtain a reduction in CPU time.

The static parts of the MoM model is defined on the *numerical Green's function settings* dialog (see Figure 6-77) on the *Solve/Run* tab.

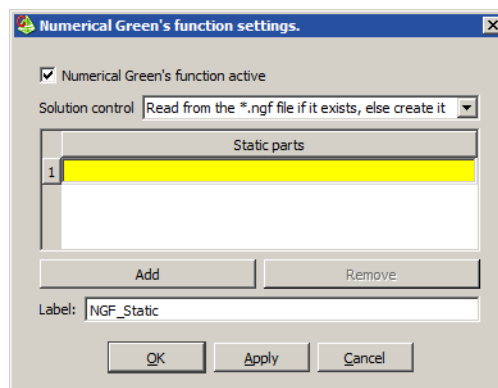


Figure 6-77: The numerical Green's function settings dialog

6.11 Solution control and special solver settings

CADFEKO provides a number of options for the different solution techniques available in the FEKO Suite. These options are advanced and the implications of each setting should be well understood before use. The inappropriate application of these settings may result in poor result accuracy or inefficient calculations. For most cases, the default solver settings are appropriate and no changes are necessary before solving the model.

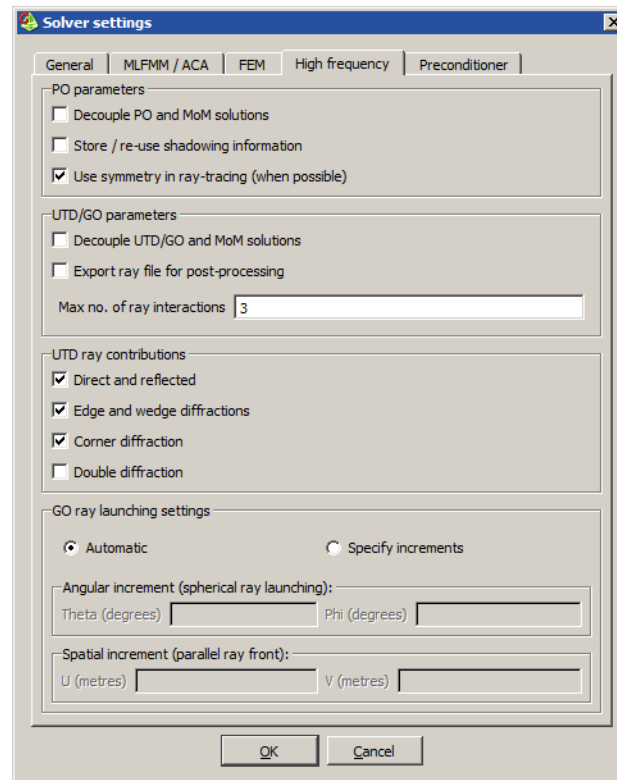
The solver settings may be accessed by right-clicking on *Solution* in the tree and selecting *Solver settings* or by selecting the *Solve/Run* tab and clicking on the *Solver settings* button.



Figure 6-78: The *Solver settings* button

The *General* tab

On the *General* tab, setting *Data storage precision* to *Single precision* forces FEKO to store certain memory-critical arrays in single precision. *Single precision* is recommended unless the FEKO

Figure 6-79: The *Solver settings* dialog.

kernel gives warnings and suggests that *Double precision* be used (this might happen for instance at very low frequencies where increased accuracy is required).

For low frequency modelling, the low frequency stabilisation for MoM may be activated in the *Low frequency modelling* group.

For large models, a significant amount of time can be saved if the solution coefficients are saved during the solution phase. Note that for smaller models (where the run time is short), storage of the solution coefficients is typically not required. Storage of the solution coefficients creates large files that need to be maintained by the user.

The reading/creation of the matrix elements (*.mat), LU-decomposition (*.lud) and solution vector of the linear equations (*.str) may be enabled in the *Output files* group.

The *MLFMM / ACA* tab

The MLFMM may be activated on the MLFMM / ACA tab by selecting *Solve model with the fast multipole method (MLFMM)*. For most cases the default settings are appropriate. The MLFMM is typically used to solve electrically large, high frequency problems faster and with less memory resources than the method of moments (MoM). If monostatic RCS is requested, solution time can be decreased if the angular increment is less or equal to 5° . Note that the MLFMM can currently not be used simultaneously with a multi-layer Green's function in the same model. MLFMM may be used with the hybrid FEM/MoM.

The parameters on the *MLFMM / ACA* tab are only enabled if solution using the MLFMM is activated. The MLFMM is based on a hierarchical tree-based grouping algorithm, and FEKO

automatically determines the ideal number of levels for each model. If the solution does not converge, advanced users may try to obtain convergence by modifying the *Box size in wavelengths* option. A starting point of 0.23 is recommended and values should not be much less than this. (The value specifies the box size in fractions of the wavelength.) Alternatively, the user can select *Advanced solver settings* and set the number of iterations or the stopping criteria for the iterative process or even select a different preconditioner. Any text field on this tab may be left empty. It should be emphasised again that setting these parameters can have significant consequences, both in accuracy and solution time — users without a working knowledge of the MLFMM or iterative solvers should preferably only use the default settings. The behaviour of the MLFMM can also be influenced by selecting the combined field integral equation, see section 6.11.1.

The adaptive cross approximation (ACA) may be activated on the MLFMM / ACA tab by selecting the *Solve model with the adaptive cross-approximation*. The ACA is a fast method similar to the MLFMM but is also applicable to low frequency problems or when using a Green's function in the model. If the solution does not converge, advanced users may try to obtain convergence by selecting the *Advanced solver settings* and setting the number of iterations, stopping criteria for the iterative process or even selecting a different preconditioner.

The advantages of the ACA is that it is done on the matrix level and has no low frequency breakdown like MLFMM. It is not restricted to free-space Greens function like MLFMM and each matrix-vector multiplication is very fast.

The *FEM* tab

The *FEM* tab is used to control the FEM (finite element method). The FEM is used if the model contains tetrahedral mesh elements (see section 4.8). The following conductor properties of metallic surfaces inside a FEM region are supported: skin effect, thin dielectric sheet, impedance sheet and electrically thin surface coating. If the FEM and MoM are de-coupled, the imperfect conductor surface may be on the FEM boundary. If *Decouple from MoM (use FEM absorbing condition) solutions* is checked, the FEM region (the tetrahedral elements and any conducting surfaces on their boundaries) will not influence the MoM solution. This decoupling may save significant run time, but is only valid if the FEM and MoM regions are electrically far enough apart.

Switching to *First order* elements reduces the required memory and run time as well as the accuracy. To get the same accuracy using First order elements, the mesh size must be reduced — normally such that the solution actually requires more memory and run time than the larger mesh with second order elements. Thus switching to *First order* elements is only recommended if the mesh is already very fine, for example to account for highly inhomogeneous media or very complex geometry.

The *High frequency* tab

The *High frequency* tab provides settings for the high-frequency asymptotic methods available in FEKO (Physical optics, the UTD (uniform theory of diffraction) and Geometrical optics (ray launching)).

Any changes to the high frequency settings should be based on the physical considerations of each model.

Solution using PO, UTD or GO is activated for each face (see section 6.11.1). If *Decouple PO and MoM solutions* or *Decouple UTD/GO and MoM solutions* are checked, the PO or UTD/GO structures are not taken into account when calculating the MoM currents. In this case, for example, the input impedance of a horn fed parabolic reflector, where the horn is solved using the MoM and the reflector using PO, will be the same as that of the MoM horn in free space. In general, where the MoM and PO/UTD/GO regions are electrically far apart, and far field quantities are of interest, decoupling the solutions may save a significant amount of memory and run time without significantly affecting the results.

During calculations using the PO formulation, a large amount of the time is spent in determining which surfaces are illuminated from each source. This information may be saved by activating the *Store / re-use shadowing information* to speed up subsequent runs (re-use is only possible if the physical geometry of the model is unchanged). Storage of the shadowing information, however, does lead to huge *.sha files on disk.

The option *Use symmetry in ray-tracing (when possible)*, is set by default for PO. If *Export ray file for post-processing* is checked, FEKO outputs the rays used during the UTD or GO solution process to a *.ray file. The ray-tracing information is also stored in the *.bof file and the rays may be viewed in POSTFEKO. It should be noted that *.ray files can be very large, especially when using a MoM / UTD solution that has not been decoupled and there are more than a few hundred mesh elements in the MoM region.

The *Max. no. of ray interactions* field limits the number of reflections/transmission interactions taken into account for each ray that is tracked during the solution process. For example, a ray path that includes a double diffraction and one reflection has three interactions, or a path that includes a diffraction and three reflections has four interactions. The options in the *Select ray contributions* group allow selection of the special UTD ray interactions that are to be taken into account. (Note that some combinations go hand in hand. For example, the contributions of edge and corner diffractions in the UTD solution are usually of the same order and they should be switched on or off together.) Selecting more contributions or increasing the number of ray interactions will increase both accuracy and computation time. The chosen settings should therefore be a compromise between accuracy and solution time.

For geometrical optics (ray launching), special ray launching settings may be accessed in the *Dielectric GO ray launching settings* section. When *Specify angular increment* is selected, the θ and ϕ angular resolution for ray launching and the increments for the parallel ray front in the *U* and *V* direction may be manually specified. Though the runtime for a problem involving GO may be decreased by using this option, it may influence the accuracy of the solution. Manual specification of the angular increment should only be used in specific conditions after the implications have been carefully considered.

The *Preconditioner* tab

Advanced users may elect to use the *Direct sparse solver* or the *Iterative solver* for FEM models which do not give a solution with the default settings. For the iterative solver it is possible to set the maximum number of iterations, termination criterion and a *Preconditioner*. The *Stabilisation factor* only applies to the incomplete (*Multilevel ILU / diagonal decomposition*) preconditioner and may improve convergence in critical cases. The value should be between 0 and 1. Note that any field on the *Preconditioner* tab may be left empty.

6.11.1 Setting solution parameters on faces

By default, FEKO will employ method of moments based (MoM) solution methods for all surfaces (dielectric and conductive) in a model. Other solution methods may, however, be set individually on each face in the model. For example, an area of the model may be solved with PO. The solution method is specified on the *Solution* tab of the *Face properties* dialog for the relevant geometry faces or on the *Mesh properties* dialog for the relevant triangle labels. (Properties set on the geometry will be transferred to the mesh during meshing.)

The different solution methods available are mutually exclusive, and the choice between the special solution methods is reflected in the *Solve with special solution method* drop-down list. The available options are:

None: No special solution method is used and the face will be solved using the default, method of moments solutions.

Physical optics (PO): The high frequency physical optics method is used when calculating the currents on the face.

full ray-tracing With this method, a significant fraction of the solution time is spent on ray tracing (determining which elements are visible from each non-PO radiating element). The advanced PO options (see section 6.11) allow manual optimisation of ray tracing.

always illuminated This option is applicable when the model is such that the entire surface will always be illuminated (such as for a basic reflector antenna analysis). Shadowing is not analysed, and non-PO or large element PO surfaces are considered transparent for this purpose.

only illuminate from front This option is applicable to all closed bodies, but the normal vector must point outward (away from the closed body) from each face. Smaller mesh elements are required close to the shadow boundaries.

Large element PO: The high frequency large element physical optics method is used for large smooth areas when calculating near and far fields. This method is only to be used when there are no discontinuities in the incident fields (e.g. close to a point source). If the large element PO is used together with the MoM method, the MoM and PO regions are to be de-coupled. Note that the same options as given above for PO (full ray-tracing, always illuminated etc.), are also valid for large element PO.

Geometrical optics (ray launching): The geometrical optics (ray launching) method is used. This method is applicable to the simulation of thin dielectric or metallic objects, such as radomes and dielectric lenses. The use of ray-launching GO is also supported for thin dielectric sheets, coatings and anisotropic media. Note that the use of plane wave as an excitation is also supported (required for RCS requests).

Uniform theory of diffraction (UTD): The UTD will be used for this face. This method is applicable to flat electrically large PEC faces.

Windscreen: The windscreen will be used for this face. This method is applicable to the simulation of multiple windscreen antennas on various glass definitions.

Planar Greens function aperture: The planar Green's function aperture modelling will be used for this face. This method is applicable to the simulation of a slot/aperture in an infinite electrical conductor.

When the properties are viewed for a selection that contains multiple faces, if any of these options are different for the faces in the selection, that field is opened in ternary state. The *Integral equation* field allows selection of the formulation to be used during the solution phase. By default the *Integral equation* is set to *Electric field* (EFIE). This is the most general formulation and can be applied to both open and closed bodies. The *Combined field* (CFIE) setting can only be used on closed objects, and then it is generally more stable. Note that the triangle normal should point away from the region with zero field. Where applicable, the CFIE is particularly useful when used with the MLFMM as it generally requires less memory and convergence can be much faster than the EFIE based solution. It is possible to use the *Combined field* setting on a subset of the model, but the surface must be a closed PEC surface and all the normals must point outward (away from the zero field region).

6.11.2 Viewing special solution parameters on faces

Keeping track of different solution parameters that have been set on different regions of the geometry can become difficult. CADFEKO provides a *View by solution* tool, which allows highlighting of all the geometry with the same solution settings in the 3D view, while the display of all other geometry is semi-transparent. This tool allows the user to quickly check that the correct solution parameters have been set before solution. The *View by solution* tool (shown in Figure 6-81) may be enabled by selecting the *Solve/Run* and clicking on the *View by solution* button.



Figure 6-80: The *View by solution* button

6.12 Requesting results

Before running a simulation, the required results must be requested. The CADFEKO model contains a list of requested calculations. These are listed in the tree under *Requests*. The same set of results are calculated for all frequencies with the specified loads and excitations. The following operations may be performed on individual requests:

- Requests that are represented in the 3D view may be hidden/shown.
- Requests that are represented in the 3D view may be included/excluded without deleting them by selecting *Include/exclude* from the popup menu. Excluded solution requests are indicated by the addition of a small red cross to the icon and will not be included in the FEKO solution (excluded solutions cannot be used during optimisation).
- Requests may be copied (to create similar requests).

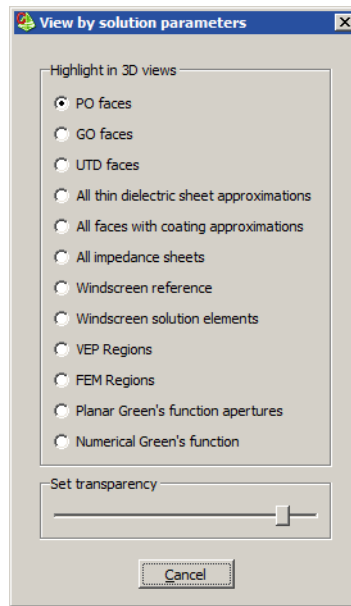


Figure 6-81: The *View by solution parameters* dialog.

- Requests may be renamed. (The names or labels of some requests are referenced in POST-FEKO. OPTFEKO also uses the labels of solution requests to uniquely identify the results applicable to optimisation).
- Requests may be deleted from the pop-up menu of the tree item, or using the <Del> key.
- Requests may be edited by double-clicking on them or selecting *Properties* from the pop-up menu.

All results may be requested from the the pop-up menu on *Requests* or by double clicking the appropriate header if other requests of this type have already been defined and are present in the model.

6.12.1 Currents

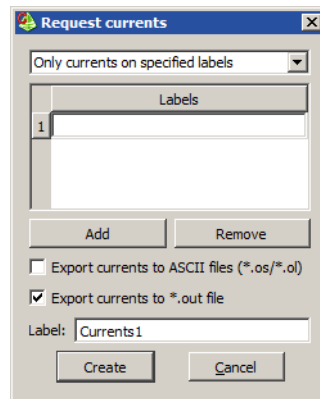
The currents that should be provided in the output of the simulation (and therefore be available for analysis or visualisation) are specified in the *Request* dialog (shown in Figure 6-83). This dialog is launched by selecting the *Request* tab and clicking on the *Currents* button.



Figure 6-82: The *Currents* button

All currents: All currents will be computed and stored in the simulation output.

Only currents on specified labels: The currents on the segments and triangles with the specified labels are computed.

Figure 6-83: The *Request currents* dialog

Only segment currents: The currents on segments are computed.

Only triangle currents: The currents on triangles are computed.

Export currents to ASCII file (.os/*.ol):* The currents and charges that are stored in the solution output are also written to a *.os and *.ol files respectively which can be read by other post-processors.

*Export currents to *.out file:* In addition to the inclusion of the currents in the standard solution output (used by POSTFEKO), the currents are written to the text *.out file.

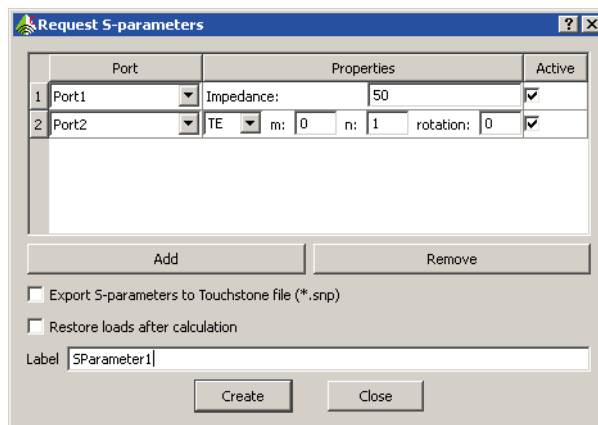
Though currents must be included in the output to enable the display of current distributions, in large models the unnecessary storage of currents can lead to large output files. When using adaptive sampling (see section 6.2) the unnecessary storage of currents may also increase the number of frequencies required for solution convergence. In general, only the currents that are needed should therefore be stored.

6.12.2 S-parameters

The calculation of S-parameters between an arbitrary number of ports may be requested by adding an S-parameter solution request. A S-parameter request may be added by selecting the *Request* tab and clicking on the *S-parameter* button.

Figure 6-84: The *S-parameters* button

The *Request S-parameters* dialog (shown in Figure 6-85) contains a table where the user can specify the different ports to be considered in an S-parameter extraction. For all ports that are not waveguide ports, the reference *Impedance* can be specified (if no reference impedance value is given, a default reference impedance of 50 Ohm is used). For waveguide ports, type (TE/TM/TEM), indices and rotation of the mode that is to be considered can be specified. The *Add* and *Remove* buttons can be used to add or remove ports to the list. If the *Active* field of any port in the table is checked, the port is used as a source. Otherwise it is only a receiving port. For

Figure 6-85: The *Request S-parameters* dialog.

example, if the list contains two ports and only *Port 1* is active, FEKO will calculate S_{11} and S_{21} but not S_{12} and S_{22} . If *Port 2* is also active, S_{12} and S_{22} will also be calculated.

The port numbers in an S-parameter solution request are indexed based on the order of appearance in the port list on the S-parameter request dialog, and not on the label of the selected port.

If *Export S-parameters to Touchstone file (*.snp)* is checked, the calculated S-parameters included in the solution output are written to an additional *.snp file. When exporting S-parameters to a Touchstone file it is important to note that FEKO does not normalise the S-parameter values to a global reference impedance, but rather provides the values referenced to the impedance specified on each port. Some tools that use the Touchstone format, however, assume that all values are referenced to a common impedance. This may result in misinterpretation of the S-parameter results provided by FEKO. When exporting S-parameters for use in a tool that supports only a single reference impedance, the reference impedance provided at each port in the S-parameter solution request should be set the same to ensure the correct interpretation.

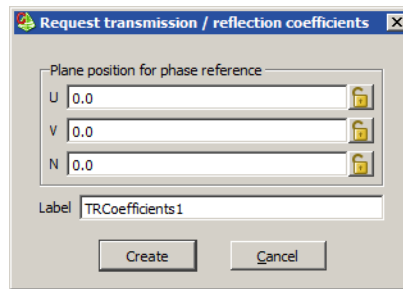
During the calculation of S-parameters, FEKO adds loads of the specified reference impedances at all the port locations. These loads normally remain in place after the S-parameter calculation. If *Restore loads after calculation* is checked the S-parameter loads are removed once the S-parameter calculation is complete. However, restoring the loads require repetition of a full matrix computation and LU decomposition step for the MoM. This is typically the most time consuming step in the analysis.

6.12.3 Transmission/reflection coefficients

The calculation of transmission/reflection coefficients for a plane wave interacting with a planar structure may be requested by selecting the *Request* tab and clicking on the *Transmission/reflection* button.

Figure 6-86: The *Transmission/reflection* button

The *Request transmission/reflection coefficients* dialog is shown in Figure 6-87.

Figure 6-87: The *Request transmission/reflection coefficients* dialog

Plane position for phase reference: The position of the plane wave in Cartesian coordinates are defined here.

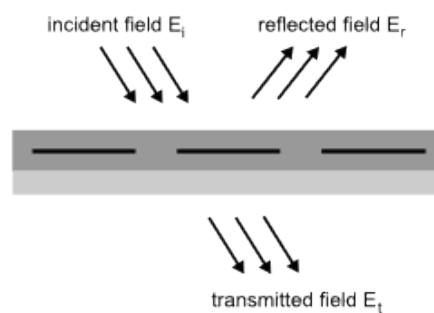


Figure 6-88: A plane wave interacting with a planar structure

The transmission coefficient is defined as

$$\tau = \frac{E_t}{E_i} \quad (6-13)$$

The reflection coefficient is given as

$$\rho = \frac{E_r}{E_i} \quad (6-14)$$

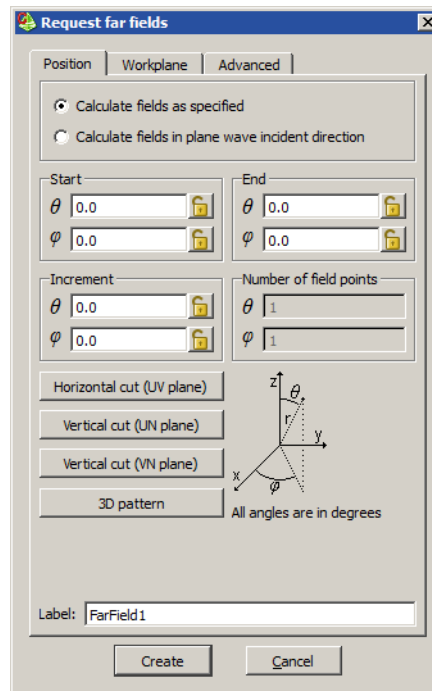
Note that only a single plane wave is allowed (i.e. no other sources are allowed in the model). The model must either contain a:

- Planar multilayer substrate without any other geometry/mesh in the model or
- 2D periodic boundary condition (PBC)

6.12.4 Far fields

The calculation of far fields is specified by adding a far field solution request. Select the *Request* tab and click on the *Far fields* button.

The *Request far fields* dialog is shown in Figure 6-90.

Figure 6-89: The *Far field* buttonFigure 6-90: The *Request far fields* dialog

The *Position* tab

On the *Position* tab, a selection can be made between two different far field calculation types.

Calculate fields as specified: This option provides for the extraction of general far field patterns. The *Start* and *End* angles and the *Increment* for each angular axis can be specified. (All these fields do accept point entry from the 3D view, though it may be counter-intuitive to enter 3D angles from a planar surface.) CADFEKO calculates and displays the number of field points. The actual end angle depends on the start angle, the number of points and the increment and may not coincide exactly with the specified end angle value. In models that contain plane wave sources, the direct contribution of these sources is always ignored. The preview shows a spherical “surface” of the specified range, with markers on the surface, indicating the actual calculation directions. Buttons are provided on the dialog that can be used to quickly define some commonly used pattern requests.

Calculate fields in plane wave incident direction: This option is used in conjunction with a plane wave excitation (see section 6.5.1), mostly to calculate monostatic radar cross section (RCS). If this option is selected, the model must contain a plane wave excitation (with single or multiple directions of incidence). If no plane wave excitations are defined, an error will be reported during the solution. No additional parameters are required in the far field request, as scattered fields are only calculated in the direction that the incident wave is coming from. This option must be selected if an RCS optimisation search is based on this far field request.

When extracting large numbers of far field samples for both axes, the impact on the solution time may be significant.

When the number of requested far field points is larger than 1 for both angular axes, FEKO computes (in addition to the far field values themselves) the integral of the Poynting vector (i.e. the total radiated power) through the spherical segment defined by the start and actual end angles in each angular direction.

A full ϕ -cut runs from 0° to 360° and a full θ -cut runs from 0° to 180° . In order to account for a potential ambiguity when specifying such a closed surface representing a full pattern in spherical coordinates, FEKO also computes the radiated power over a second sector that extends by half an angular increment beyond the specified start and end points in both angular directions.

For the case where the ϕ and θ start and end angles exactly overlap, the power computation in the first-sector correctly represents the radiated power in all directions. If, however, the first and last angular directions do not exactly overlap, but are separated by a single increment angle value, the power computed over the second sector represents the actual power over the full pattern. (Both radiated power values — with and without the overlap-correction — are computed and stored in the text *.out file below the far-field information. The actual ranges represented by each of the power values is also indicated. The value that represents the intended angular ranges should be taken — for example, when considering a full pattern, the value for which the ϕ range is exactly 360° and the θ range is exactly 180° should be used.)

The *Advanced* tab

Directivity/Gain: A selection may be made between the calculation of directivity or gain (when not calculating RCS). Note that both POSTFEKO and OPTFEKO can access both gain and directivity results independent of this setting — this setting controls only what is written to the *.out file.

Export fields to ASCII file (.ffe):* If this option is checked, the computed far fields are exported to an *.ffe file. This file can be used by other post-processors or as a source pattern for the radiation pattern point source (see section 6.5.3) or a receiving antenna (see section 6.12.8) element.

*Export fields to *.out file:* If this option is checked, the fields are included in the text output *.out file. This file is not used by POSTFEKO or OPTFEKO and is only necessary if the user would like to view or extract the relevant data from the text output directly.

Calculate only the scattered part of the field: This option indicates that the radiated contribution from impressed sources (such as electric and magnetic point sources) should (in addition to the contribution from plane wave sources) be ignored, yielding only the scattered fields. Normally this option should remain unchecked.

Only determine radiated far field power by integration: If this option is checked, the far fields — and the total radiated power — are calculated, but the field values are not written to the *.bof or *.out output files (fields are still written to the *.ffe file if this has been requested). This option should only be used if the individual field values are not required and the output files would otherwise become too large. If this option is selected, the far field values will not be accessible for viewing in POSTFEKO or for optimisation in OPTFEKO

Calculate spherical expansion mode coefficients: This can be checked to calculate these coefficients. The coefficients can be exported to a TICRA *.sph file. Note that the gain in GRASP will only be correct if the radiated power in FEKO has been set to 4π Watts. For more detail see the description of spherical modes in the FF card (see section 14.37). Spherical mode

coefficients can also be imported from a *.sph file in the spherical mode excitation (see section 6.5.4).

Calculate far field for an array of elements: If this option is checked, the far field can be calculated for an array of elements. Optional is to set the *Number of elements along vector 1* and *Number of elements along vector 2*.

6.12.5 Near fields

The calculation of near fields is requested by adding a near field solution request. Select the *Request* tab and click on the *Near fields* button.



Figure 6-91: The *Near field* button

The *Request near fields* dialog is shown in Figure 6-92.

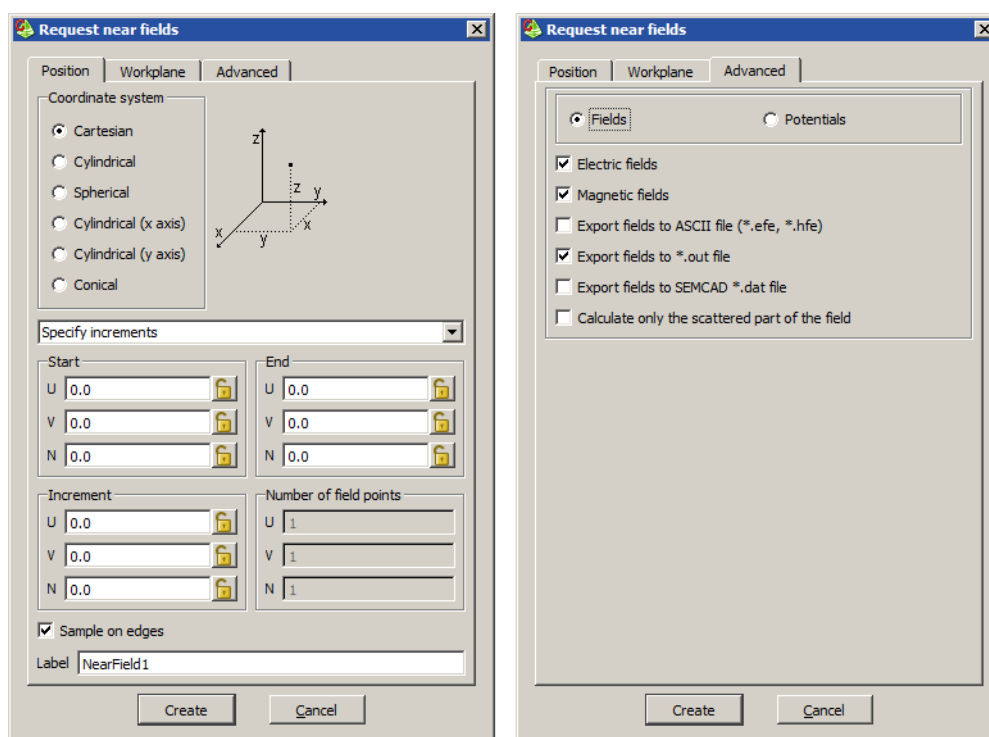


Figure 6-92: The *Position* and *Advanced* tabs of the *Near field request* dialog

The *Position* tab

On the *Position* tab, the *Start*, *Increment* and approximate *End* positions for each axis as well as the type of *Coordinate system*, must be specified.

Specify increments: A user specifies the *Start*, *End* and *Increment*. The *Number of field points* are then automatically calculated.

Specify number of points: A user specifies the *Start*, *End* and *Number of field points*. The *Increment* is then automatically calculated.

The actual end position depends on the start position, the number of samples and the increment and may not coincide exactly with the specified end position. The fields and units of the fields are interpreted according the coordinate system selection on the *coordinates* tab (distances in terms of the model unit and angles in degrees). CADFEKO dynamically calculates the number of sample points in each direction and displays a preview of in the 3D view. This preview shows the specified range as surfaces enclosing the specified volume, with markers at the actual sample positions. For finely sampled near field requests, where there are too many markers, the markers may not be displayed. (Note that using even moderate numbers in all three directions can quickly result in a large number of samples.)

The *Advanced* tab

On the *Advanced* tab, a selection may be made between the calculation of fields or potentials and special storage options may be set.

Field: The actual near field components are calculated and stored in the solution output. The electric and/or magnetic field components may be included.

Potential: This option may be used for advanced field analysis. Only one potential type may be included in a single near field request. The vector potential, or the gradient of the scalar potential for either the electric or magnetic field may be selected. The data from near field requests in which potentials are calculated cannot be viewed or processed in POSTFEKO.

The following options are common to both the Field and Potential near field request types.

Export fields to ASCII file (*.efe, *.hfe): Calculated electric fields/potentials are written to a *.efe file and magnetic fields/potentials to a *.hfe file. These are ASCII files that may be used by other post-processors.

Export fields to *.out file: If this option is checked, the fields are included in the human readable *.out file. (The *.out file is not required for viewing in POSTFEKO or for optimisation in OPTFEKO.)

Export fields to SEMCAD *.dat file: If this option is checked, the fields are written to a *.dat file.

Calculate only the scattered part of the field: If this option is checked, the radiated contribution from impressed sources (such as electric and magnetic point sources) is ignored, yielding only the scattered fields. Normally this option should remain unchecked.

6.12.6 SAR

Specific absorption rate (SAR) studies typically require the average absorption over a volume (the *Volume-average SAR*) or the maximum absorption in a 1 g or 10 g cube anywhere in a given volume (the *Spatial-peak SAR*). The calculation of these quantities can be requested by adding a SAR solution request. Select the *Request* tab and click on the *SAR* tab.

The *Request SAR* dialog is shown in Figure 6-94.



Figure 6-93: The SAR button

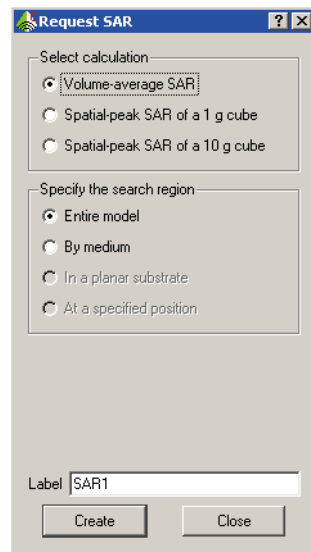


Figure 6-94: The SAR request dialog

Select calculation: The specific SAR related quantity that is required is defined here.

Specify the search region: The region (or volume) in which the SAR should be calculated is defined here.
The available search regions are:

- *Entire model* : All dielectric regions in the model will be considered, and a single average or peak SAR value will be computed.
- *By medium* : Only the set of regions with specific media types will be considered. All, or only a specific medium may be selected. For *All media* average or peak SAR is calculated separately for each medium, and thus it is not the same as selecting *Entire model*.
- *In a planar substrate* : A specific layer (layer 0 is the upper free space region, layer 1 the uppermost dielectric layer, etc.) or *All layers* may be included. This option cannot be used with volume averaging.
- *At a specified position* : The 1 g or 10 g cube SAR value may be calculated at a specific position. This option cannot be used with volume averaging.

6.12.7 Cable harness

- | The *Cable harness* request may be used to calculate the coupling into a cable near a large conducting surface (see section 4.6.4).

The results of a cable analysis calculation — the voltages, currents, impedance and power between the inner conductor and the shield — can be displayed in POSTFEKO. Note that even though FEKO calculates the coupling into the cables, the cable geometry is not taken into account in any other calculations. The presence of the cable therefore does not influence any other result — not even other cable analysis calculations.

6.12.8 Ideal receiving antenna

The *Ideal receiving antenna* provides a tool by which the power that would be received by an ideal antenna with a specified pattern can be calculated.

Select the *Request* tab and click on the *RX antenna* button.



Figure 6-95: The *RX antenna* button

The *Request ideal receiving antenna* dialog is shown in Figure 6-96.

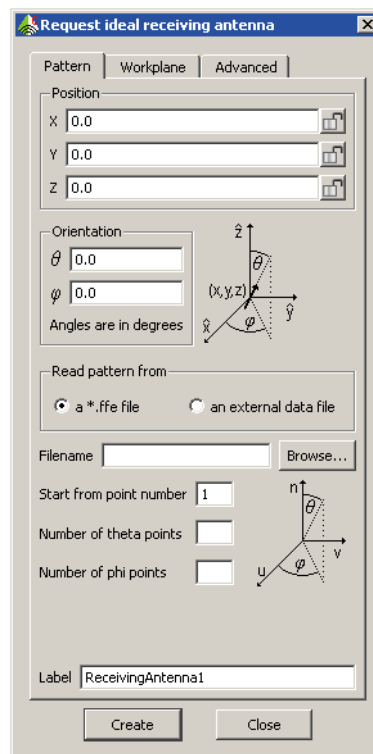


Figure 6-96: The ideal receiving antenna request dialog

The receiving antenna is reciprocal to the point source with a specified pattern (see section 6.5.3) and the parameters on the dialog are identical.

Note that the following assumptions must be kept in mind when using the ideal receiving antenna:

- The antenna is considered to be matched (i.e. no mismatch loss is taken into account)
- The antenna is assumed to exist only at one point
- The antenna and model are assumed to have no impact on each other during the solution phase (no coupling is taken into account)

On the *Advanced* tab the option is available to include only the scattered part of the field.

6.13 Error estimation

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. Requesting of error estimates will enable the user to find areas which might require local mesh refinement, (see section 4.8.4).

Select the *Request* tab and click on the *Error estimation* group.



Figure 6-97: The *Error estimation* button

A user may elect to request error estimates only on *All mesh elements*, *Only error estimates on triangles*, *Only error estimates on segments*, *Only error estimates on tetrahedra* or *Only error estimates on specified labels*.

The error estimate dialog is shown in Figure 6-98.

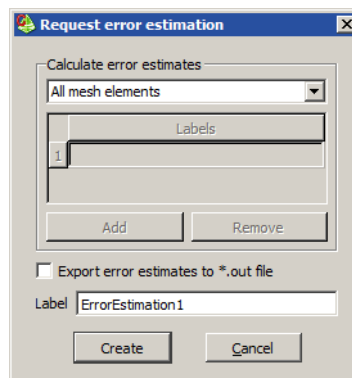


Figure 6-98: The *Request error estimation* dialog

*Export error estimates to *.out file:* In addition to the inclusion of the currents in the standard solution output (used by POSTFEKO), the error estimates are written to the text *.out file.

Adaptive mesh refinement may be applied by reading the error estimates from an earlier solution. Point refinement rules will be created in the areas where the errors are estimated to be the highest.

An *Adaptive refinement* may be added by selecting the *Mesh* tab and clicking on the *Adaptive refinement* button.



Figure 6-99: The *Adaptive refinement* button

6.14 Mesh manipulation

6.15 CEM validation of the CADFEKO model

CADFEKO provides a tool that performs a number of basic consistency checks on the model. These checks are largely related to electromagnetic modelling rules and limitations. As meshes may consist of millions of elements, CADFEKO does not perform continuous consistency validation of all of the mesh parameters, but requires that the user launch the validation tool manually to check the model. EM validation of the model is generally advisable and all indicated problems should be addressed before the FEKO solver is launched. Considerable amounts of time and frustration can be spared if modelling errors are identified as early as possible.

Select the *Solve/Run* tab and click on the *CEM validate* button.



Figure 6-100: The *CEM validate* button

As the electromagnetic validation for the various components are completed, the details are displayed on the dialog. The following components are checked: *Frequency*, *Geometry*, *Mesh* and *Solution*. As each component is validated, an icon is displayed giving the user information regarding the status of the completed EM validation. When selecting a received warning or error, a message detailing the warning/error is given in the *Error/Warning details of the selected item* window.

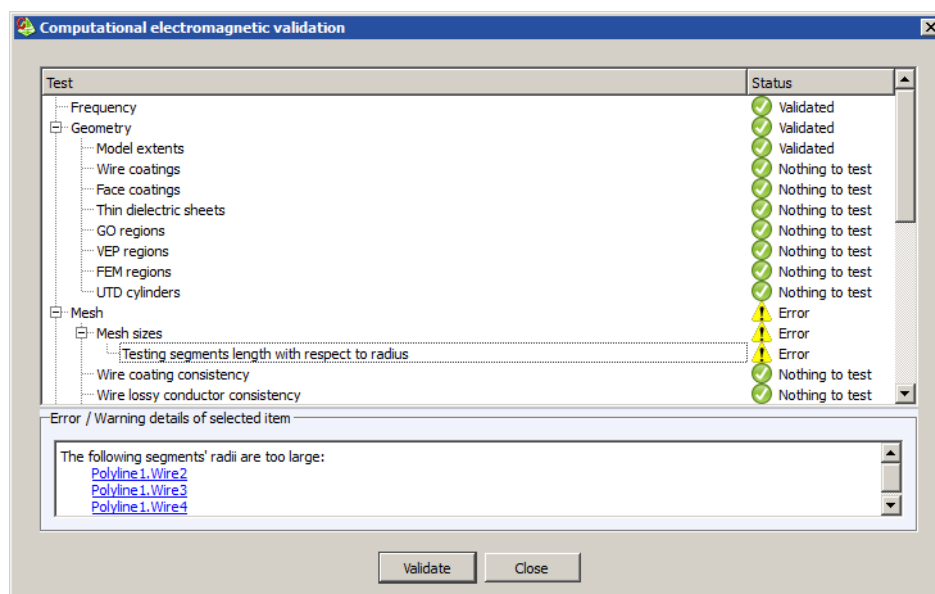


Figure 6-101: The *Computational electromagnetic validation* dialog

6.16 Running the FEKO solver components

The CADFEKO model is saved as a *.cfx file. In order to solve a model built in CADFEKO, two input files are required by the FEKO pre-processor. These are the *.cfm file (which contains the mesh information) and the *.pre file (which refers to the *.cfm file and contains

information regarding the solution process and settings). When saving the model, CADFEKO automatically exports the current mesh to the *.cfm file (if solution parameters are enabled (see section 6.18.1), the *.pre file is also updated). CADFEKO also exports the current mesh to the *.cfm (and *.pre) file each time a component is started from the *Run* menu.

If CADFEKO encounters any invalid settings, it writes error states to the *.pre file so that the user will get the appropriate error message, even when running PREFEKO from another component of the FEKO Suite.

Typically, once the model is set up in CADFEKO, PREFEKO is executed to create the FEKO input file (*.fek) from the *.cfm and *.pre files. The geometry, excitations and solution requests represented in this file can then be viewed and visually validated in POSTFEKO before running FEKO or OPTFEKO.

Once the FEKO solution is completed, the results (stored in the *.bof file) can be viewed in POSTFEKO, or in the text output file (*.out) .

6.16.1 Component parameters

The *Component parameters* item is situated on the *Solve/Run* tab. Click on the dialog launcher button to display the *Component launch options* dialog. This dialog provides for the specification of command line parameters for the various FEKO components. Note this dialog is the same for CADFEKO, POSTFEKO and EDITFEKO.



Figure 6-102: The *Dialog launcher* button

The *PREFEKO* tab

If *Treat errors as non-fatal...* is checked, PREFEKO will continue running after encountering an error. This is used to see all geometry modelling errors identified during the PREFEKO geometry validation at once. (PREFEKO can also generate additional debug output, but these options should only be activated to debug specific issues.)

Shown in Figure 6-103 is the *PREFEKO* tab on the *Component parameters* dialog. The following options are available: *Export of variables (names, values, comments)*, *Debug options* and *Advanced*.

The *Advanced* field allows manually typing the required command line options as would be done after the file name in a command shell. More detail can be found in the PREFEKO chapter (see section 17).

The *FEKO* tab

If *Only check the geometry* on the *FEKO* tab is checked, FEKO will perform all the geometry checks and exit before any computations commence. This option is usually used to check model validity before solving the model on a separate machine.

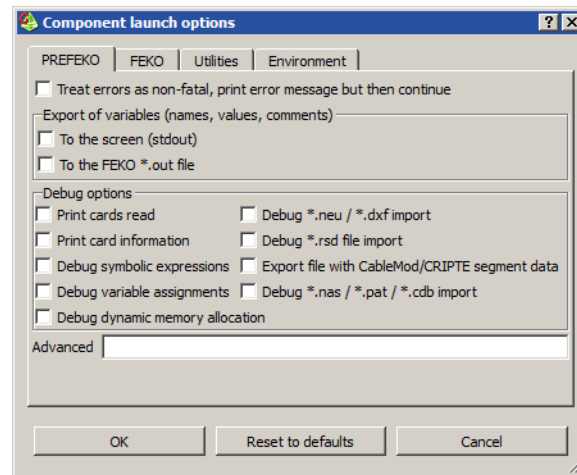


Figure 6-103: The *PREFEKO* tab on the *Component parameters* dialog

The user can also set priority of the FEKO run on this tab. If the priority is set to *Low* the run will take slightly longer, but the computer will still be responsive for other work. (Note that for parallel runs, all machines in the cluster operate at the speed of the slowest PC, so starting other CPU-intensive jobs on one of the PCs in a cluster is generally not recommended.)

Users may elect to utilise their NVIDIA GPUs based on the Compute Unified Architecture (CUDA) by checking the *Use GPU (graphical processing) for NVIDIA CUDA devices* checkbox.

GPU acceleration is supported for Windows and Linux platforms. Always ensure that the latest NVIDIA drivers are installed for the graphics card. More details regarding GPU acceleration and CUDA and a list of supported graphics cards is available at <http://www.feko.info/GPU>.

Note that enabling GPU acceleration is only applicable if a compatible NVIDIA GPU device is found. If no compatible NVIDIA GPU device is found, a warning will be issued during the computation run and the setting is reverted to the default state of no GPU acceleration.

The *Remote host* information is used for remote execution. The *Remote execution* is enabled on the *Solve/Run* tab. Additional information is also available in Section 18.2.1.

The *Remote execution* group deals with the parameters defining the settings used when launching FEKO on another machine remotely. First one specifies which machine to use in the *Remote host (hostname or IP address)* field and second the *Remote execution method* for the launching mechanism can be chosen. Here two options are available:

- *ssh/rsh*: This is the method using a remote shell (either RSH or SSH or similar) for launching the process. For copying of the files SCP (or similar) will be used. Thus the remote machine must be able to serve such connection attempts (i.e. a SSH daemon must be setup and running with public key authentication). This method can be used between different platforms.
- *MPI*: This is only supported between Windows machines (i.e. both machines must run a Windows operating system), since 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 then done by the MPI daemon which is installed already during the FEKO setup for parallel launching. Authentication is done by Windows internal mechanisms, so the remote machine must be able to authenticate the current user either against a domain or its local user database to grant access.

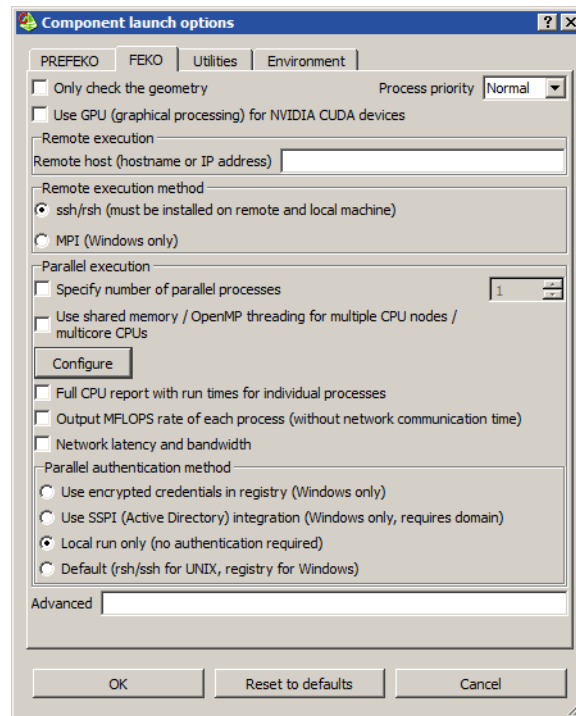


Figure 6-104: The *FEKO* tab on the *Component parameters* dialog

Additional information is also available in Section 18.2.3.

The options under *Parallel execution* are used for the parallel version of FEKO when *Run* → *Parallel FEKO execution* is selected on the *Run* menu.

The option *Specify number of parallel processes* along with the associated edit field specifies the total number of parallel processes to be launched. If this is not active, then all entries in the machines list (see below for the *Configure* button) with their specified number of processes will be used for launching. If this option is active then only the given number of processes will be started using the list from the beginning until the number is reached, regardless if there were more entries available.

If the *Use shared memory/OpenMP threading for multiple CPU nodes* is selected, the hybrid MPI/OpenMP parallelisation is utilised. It combines the benefits of OpenMP (implementation of multi-threading) and MPI (communications protocol used to program parallel computers). Processes are distributed over multiple machines/nodes by means of MPI. Multi-threading is then used on each node for subsequent parallelisation and memory saving.

Clicking *Configure* in this group allows specification of the machines to be used for the parallel solution and the number of processes to run on each. The order of the entries is of importance as this order is used upon starting the processes (i.e. the processes are started on the machines from top to bottom).

The options *Full CPU report with runtimes for individual processes*, *Output MFLOPS rate of each process (with network communication time)* and *Network latency and bandwidth* enable additional diagnostic tests and output and should be disabled for normal FEKO runs to not degrade performance.

In the *Parallel authentication method* group one selects the mechanism to be used for authenticating the parallel processes on the individual machines. Options are:

- *Use encrypted credentials in registry (Windows only)*: This option uses previously stored encrypted username and password from the Windows registry. The user has to save these credentials previous to starting any parallel computation using the tool *Update parallel credentials (MPIregister)* provided with the FEKO installation (*START* → *All Programs* → *FEKO* → *Suite x.y*). This is a per-user setting and has to be updated on each change of the user's password. If using remote-parallel launching, this must also be done on the remote host where then parallel FEKO will be started from.
- *Use SSPI (Active Directory) integration (Windows only, requires domain)*: If selecting this option all the machines must be member of a Windows (Active Directory) domain and also the user accounts must be domain accounts. Then the authentication will be carried out using internal functions of Windows without having to encrypt anything into the registry. Note that maybe additional (one time) configuration settings have to be done (by the domain administrator) to prepare the Windows domain for this kind of authentication requests (see the Intel MPI (User Authorization — Active Directory Setup) and/or MPICH2 (Runtime Environment — Security) documentation shipped in the directory `mpi\<mpi-version>\doc` of the FEKO installation directory for details).
- *Local run only (no authentication required)*: This option signals to FEKO that all processes run on the same local machine and thus no authentication is required at all. This is most commonly used if running parallel FEKO only locally on a multicore or multiprocessor machine.
- *Default (rsh/ssh for UNIX, registry for Windows)*: Selecting this option will always use the default authentication method for the target operating system. For UNIX systems the public key authentication of rsh/ssh will be used and for Windows the registry method is considered the default.

Additional information is also available in Section [18.2.2](#).

The *Advanced* field allows manually typing the command line options as would be done after the file name when launching FEKO from a command shell. The installation of remote launching as well as the parallel version is covered in the *FEKO Installation Guide*. More detail can be found in the FEKO section (see section [18](#)).

The *Utilities* tab

On the *Utilities* tab, the user can select if the temporary files created during OPTFEKO, TIMEFEKO and ADAPTFEKO runs should be deleted or kept (in the case of OPTFEKO, the files relating to the optimum model and solution are not considered temporary files and are not deleted), shown in Figure [6-105](#), TIMEFEKO and OPTFEKO.

The following options can be set on the *Utilities* tab: ADAPTFEKO, TIMEFEKO and OPTFEKO.

The *Restart analysis number* option for ADAPTFEKO can be used if the run was discontinued (and the temporary files were not deleted). The solution can be restarted at the number of the first unfinished model.

The *Restart from solver run* option for OPTFEKO can be used if the optimisation process was terminated or interrupted and no temporary files were deleted. The optimisation can be restarted

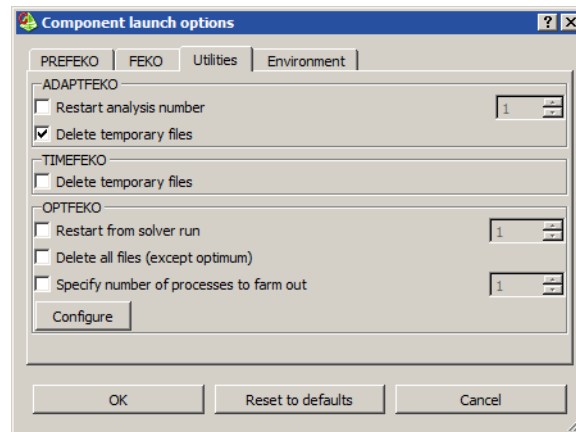


Figure 6-105: The *Utilities* tab on the *Component parameters* dialog

at the number of the last completed optimisation iteration. No changes whatsoever may be made to the model before restarting the optimisation process.

The *Farming out kernel runs* section provides a *Configure* button that launches the *Machines configuration* dialog. This can be used to set up the machines and processors that should be used for an optimisation run where farming out of the kernel solutions (see section 20.4) is used.

The *Environment* tab

The *Environment variables* field may be used to enter environment variables which can control the FEKO solution.

6.17 Additional solution configurations

The *Remote FEKO execution*, *Parallel FEKO execution* and *Farm out by OPTFEKO* can be toggled on or off by selecting the respective configuration on the *Solve/Run* tab.



Figure 6-106: The (a) *Remote* button, (b) the *Parallel* button (c) the *Farm out* button

If activated, a check mark will be displayed in front of the configuration method. When the combination of *Parallel FEKO execution* and *Farm out by OPTFEKO* are both selected, farming will be used during optimisation whilst using parallel FEKO runs.

6.18 Working with CADFEKO models in EDITFEKO

CADFEKO can control the entire solution configuration and it should generally not be necessary to work with the *.pre file directly. Some advanced features, however, are only available in the EDITFEKO interface and more flexible solution configurations can be realised by direct editing of the *.pre file.

Users may prefer to continue using EDITFEKO for existing models, particularly those with complex control settings. Even though CADFEKO can create new *.pre files, it cannot import settings or card-based geometry from an existing *.pre file. If a *.pre file created by CADFEKO is modified outside of CADFEKO, then CADFEKO will relinquish control of the *.pre file and it will have to be maintained outside of CADFEKO from that point onwards.

The rest of this section assumes that the user is already familiar with EDITFEKO (see section 11).

6.18.1 Disabling the CADFEKO solution configuration

As mentioned above, CADFEKO supports old models (or cases where the user wants to edit the *.pre file to create complex solutions) by working in a geometry only type mode. This is activated by selecting the *Solve/Run* tab and click on the *Enable solution* button.



Figure 6-107: The *Enable solution* button

Normally this item is disabled and all solution related operations are enabled. If a model is loaded or saved in CADFEKO, it checks if the *.pre file has been edited outside CADFEKO. If this is the case, the user can elect to ignore the changes made to the *.pre file or disable the solution control in CADFEKO.

If the CADFEKO solution is disabled, the *Enable solution* menu will be enabled and unchecked. All solution settings are unavailable and any existing settings are ignored when saving the model. (They are maintained and can be re-instated by enabling the solution again.) *Dielectric* media can be created, but their material parameters cannot be set.

Selecting *Enable solution* will re-enable all the solution settings and rename the existing *.pre file to name_orig_x.pre (x is a number chosen to ensure that no existing file is overwritten). After re-enabling the CADFEKO solution, all settings made exclusively in the *.pre file will be ignored, and only solution components and settings defined in CADFEKO will be used in the solution.

6.18.2 Setting units

If the solution configuration is disabled, FEKO expects all input to be in metres unless the SF card is used 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.

6.18.3 Referencing elements

In EDITFEKO, properties can be set on specific elements using their full labels (see section 4.4.4). 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...*). Of course, these names can be modified on the geometry or the mesh elements. See section 4.10.2 for more details.

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.Port1* while the remaining segments will have the label *Line1.Wire1*. For vertex ports, the associated segment is the shorter segment connected to the vertex — POSTFEKO can be used to check which segment was relabelled.

Since every region / face / edge has a unique label, a typical model may contain a large number of labels. Setting properties such as losses then requires a large number of cards. To simplify this, PREFEKO supports renaming multiple labels using the wild cards * and ? (where * expands to any string and ? to any character). For example, renaming *Wheel?.Face** to *Wheel* will rename the faces *Wheel1.Face1* and *Wheel2.Face17* (and all others matching this pattern, but not *Wheel10.Face1* or *Wheel1.Wire1*) to *Wheel*. They can then be referenced with a single name. After renaming these elements, the original names no longer exist, i.e. referencing items using those names will not find any elements. Care should, however, be taken not to rename labels — such as ports — which need to be unique.

6.18.4 Using variables and named points in EDITFEKO

When exporting the *.cfm file, CADFEKO includes all variables and named points with their current values (i.e. all expressions used in the definitions of variables and named points are evaluated and reduced to numerical values before inclusion in the *.cfm 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.

Note that all expressions used in the solution configuration in CADFEKO are evaluated and reduced to static numerical values before writing the *.pre file — hence changing these variables by redefining them after the IN card in EDITFEKO will not have any effect on expression-based solution configurations defined in CADFEKO.

6.18.5 Setting dielectric parameters

If the solution is disabled in CADFEKO, dielectric media may still be defined and applied to regions or mesh elements in CADFEKO.

The parameters of each additional dielectric must, however, be specified by the manual addition of a DI card — referencing the name of the dielectric specified in CADFEKO — to the control section of the *.pre file. As dielectric names are not hierarchical (like the label names) and the same dielectric can be applied to multiple, entirely separate regions.

7 Setting up optimisation in CADFEKO

CADFEKO provides a facility to allow the set up of an optimisation. This facility replaces the EDITFEKO-based optimisation approaches that are available in previous FEKO releases. The implementation is such that (though optimisation settings will have to be manually recreated in CADFEKO) all existing FEKO models can be used as part of an optimisation process.

In this section, setting up of optimisation and various optimisation-related options in the CADEFKO interface are considered. For further information regarding the optimisation work-flow, optimisation algorithms and the interpretation of the various optimisation-related settings, please refer to the description of OPTFEKO (see section 20). An example of the usage of the optimiser may be found in the *Getting started manual*.

It is important to note that adaptive or continuously sampled results (generated using ADAPTFEKO) cannot be used in an optimisation. Only results generated when using single or discretely sampled frequency settings may be used.

7.1 Defining optimisation searches

The first step in setting up an optimisation is the definition of an optimisation search. A search can be created in multiple ways:

1. Select the *Request* tab and click on the *Add search* button
2. Right-click on the *Optimisation* branch in the tree and select *Add search*.



Figure 7-1: The *Add search* button

When a new optimisation search is created, the *Add optimisation search* dialog will appear. In this dialog, the optimisation method and method-specific settings can be chosen for the optimisation search before the search is created. Once a search has been created, it is represented under the *Optimisation* branch of the model tree. In order to change the optimisation search settings, the *Modify optimisation search* dialog (as shown in Figure 7-2) may be opened by double-clicking on the relevant search in the *Optimisation* tree, or by right-clicking on the search in the *Optimisation* tree, and selecting *Properties* at any time after creation.

Multiple searches

Multiple searches may be defined in one model and are represented as individual branches in the *Optimisation* part of the model tree. Only one optimisation search may be activated at a time (if there is only one search defined in the model, then this search will always be active). The settings and options chosen in each search are completely independent, and only the settings specified in the active search will be saved to the *.opt and *.pfg files for use during an optimisation run.

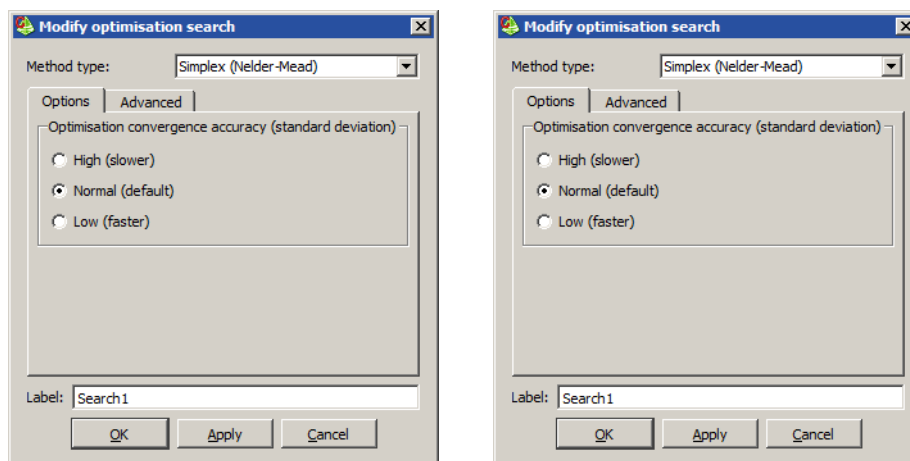


Figure 7-2: The optimisation search settings dialog

Figure 7-3: The *Activate* button

In order to activate a specific search, right-click on the relevant search in the tree and select *Activate optimisation search* or select the *Request* tab and click on the *Activate* button.

The active search is indicated by the addition of a green check-mark to the icon representation of the search in the tree.

7.1.1 Setting the optimisation method and stopping criteria

There are 5 methods that can be chosen for a specific optimisation search. These are:

Automatic: A method will be automatically chosen by the optimiser.

Simplex (Nelder-Mead): A gradient-based or ‘hill-climbing’ method (section 20.2.1).

Particle swarm optimisation (PSO): A swarm-based global search method (section 20.2.2).

Genetic algorithm (GA): An evolutionary global search method (section 20.2.3).

Grid search: This method searches over a pre-defined grid of parameter sets (section 20.2.4).

For the *Automatic*, *Simplex*, *PSO* and *GA* methods, two options that will influence the stopping criteria of the method may be chosen. The first option is *Specify the maximum number of solver runs*. If this option is activated, then the optimisation process will be terminated once the FEKO solver has been launched the specified number of times during the optimisation process (note that for the PSO and GA methods, the optimisation may terminate before the indicated number of solver runs when a full swarm or generation cannot be generated within the remaining number of allocated runs). The second termination option is the *Optimisation convergence accuracy (standard deviation)* option. This option allows the level of accuracy required for the optimisation process to be adjusted. There are three options, *High*, *Normal* and *Low*. Each of these options modifies the conditions under which the search algorithm will converge, and the effect

is dependant on the method chosen (for more information regarding the effects of this option on each method see section 20.2). The default accuracy for all methods is *Normal*.

For the PSO and GA optimisation methods, the seed value can be set for random generation (the *Advanced* tab of the *Modify optimisation search*, shown in Figure 7-2).

The following options are available:

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 can be entered as any number greater or equal to 1.

When an optimisation process is prematurely terminated due to the limitation placed on the number of solver runs, the optimum solution found up to that point, as well as all other information regarding the optimisation process will be available.

For the special case of the *Grid search* method, the termination criteria available for other methods are not relevant. If this method is chosen a single option is provided. The *Default number of points* setting specifies the default number of grid points that are to be used for each optimisation parameter in the predefined grid. This value will be used for each parameter for which the *Grid points* field in the *Optimisation parameters* dialog for that search is left empty (parameter settings are described in section 7.1.2).

7.1.2 Defining optimisation parameters

Parameters define the boundaries on variables of the model that can modified during an optimisation search. Select the *Request* tab and click on the *Parameters* button.



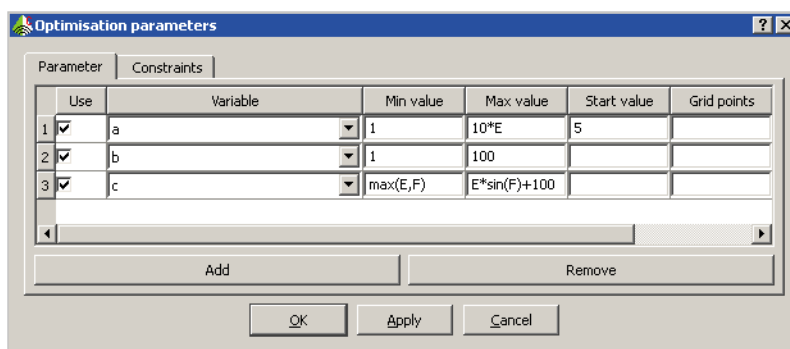
Figure 7-4: The *Parameters* button

The parameters are local to each optimisation search and are defined in the *Optimisation parameters* dialog (shown in Figure 7-5). This dialog is accessed by double-clicking on the *Parameters* entry in the tree under the relevant optimisation search. A valid optimisation search must have at least one parameter defined. (It is important to note that all settings, including the parameter selections and settings are local to each search.)

Using the *Add* and *Remove* buttons, parameter definition entries may be added or removed from the parameter list.

Parameter names

Each parameter name is chosen from a drop-down list containing all of the *Variables* defined in the CADFEKO model. Once selected, the parameter inherits the name of the variable and if the name of the variable is changed, the parameter name will be updated accordingly. (The definition of optimisation parameters based on variables that are only referred to in EDITFEKO card-based geometry and/or solution settings that have been manually added to the pre-file is possible. An identically named CADFEKO variable must, however, first be added to the CADFEKO model — information regarding the creation of variables in CADFEKO may be found in section 4.2.1.)

Figure 7-5: The *Optimisation parameters* dialog

Parameter boundaries

For each parameter a *Min value* and *Max value* must be specified. These may contain any valid CADFEKO expression given in Table 4-1 and may be a function of any other variables defined in the CADFEKO model provided that the expression does not imply a dependency between two optimisation parameters in the same search. Any dependency between parameters will result in an error when the settings are applied. The offending expression field/s will be highlighted and details regarding the dependency provided in the message window (see section 3.1). The evaluated maximum value expression must be larger than the evaluated expression for the minimum value.

In addition to the required minimum and maximum values for each parameter, a *Start value* and *Grid points* value may be set for each parameter. The *Grid points* value specifies the number of points to be used for that parameter when the Grid search method is used. If no value is chosen here, then the default value specified in the Grid search method settings (see section 7.1.1) will be used. The *Start value* specifies the first point that the optimiser should use for that parameter. This is particularly useful if some information is available regarding the approximate location of the optimum value for that parameter and the search can be started close to the optimal value. Particularly when the Randomised search methods (PSO (see section 20.2.2) and GA (see section 20.2.3)) are used, the starting value will have an effect on the optimisation process. If no start value is chosen for a given parameter, the value in the middle of the specified range will be chosen as the start value for that parameter. The specified starting point must lie in the range between the minimum and maximum values.

Constraints between optimisation parameters

The *Optimisation parameters* dialog provides a *Constraints* tab (shown in Figure 7-6). *Constraints* may be set on any optimisation parameter, defining dynamic boundaries for that parameter during the optimisation process.

Parameter and constraint deactivation

For each parameter in the parameter list or constraint in the parameter constraints list, there is a *Use* check-box that can be used to specify the inclusion of each specific parameter or constraint in the optimisation search process. If the *Use* check-box for a specific parameter or constraint is

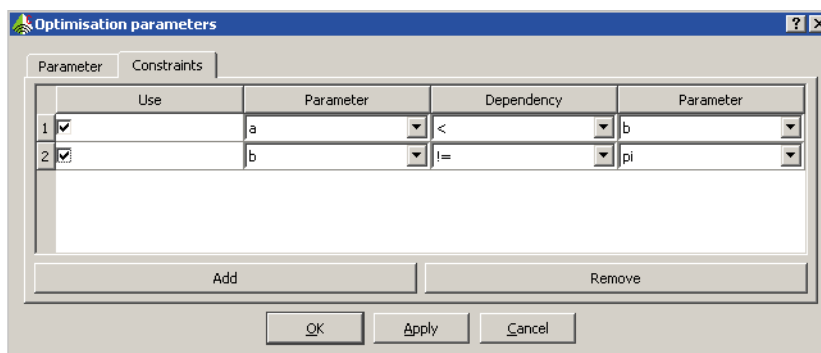


Figure 7-6: Setting *Constraints* on the *Optimisation parameters* dialog

un-checked, then that parameter or constraint will not be included in the *.opt or *.pfg files and will not influence the optimisation search. If a parameter is deactivated, the value of the variable as specified in the CADFEKO variables list will be used as if it were not defined as an optimisation parameter. (It is important to note that all parameter and constraint settings are local to each search. Deactivating a specific parameter or constraint in the parameter settings of one search will therefore not deactivate that parameter or constraint in any other search.)

7.1.3 Defining optimisation masks

An optimisation mask may be defined by selecting the *Request* tab and clicking on the *Add mask* button or by double clicking on *Masks* in the tree.

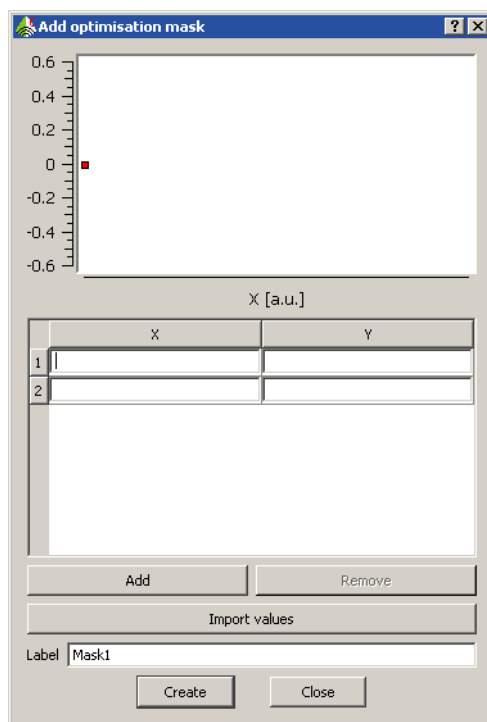


Figure 7-7: The *Add mask* button

The *Add optimisation mask* dialog (shown in Figure 7-8) allows the definition of the mask either by importing values from an external text file (tab-, comma- or space-delimited) or by manually entering the x and y coordinates of the mask points. No linkage is maintained to an external file from which mask coordinates may have been imported, and if the file changes, the points must be re-imported. When importing points from an external file, any existing coordinate definitions in the mask are over-written by the imported data. The import therefore always starts from the first coordinate point of the mask.

A graphical representation of the mask is shown. This is useful for validating that the mask data is in fact as intended, particularly when working with large numbers of data points imported from an external file.

The mask can be given a label that is used when referencing the mask in the optimisation objective of the optimisation goal(s). Note that masks are not specific to an optimisation search and may therefore be referenced from multiple optimisation searches without having to redefine the mask within each search.

Figure 7-8: The *Add optimisation mask* dialog

How masks are used for optimisation

An optimisation mask is a set of values that form a continuous line (or trace). The mask is used to determine the goal value for the FEKO simulations. All the points that are calculated by FEKO that satisfy the criteria for the goal type and name will be added to a long array of values and then compared to the mask. Three examples will be used to illustrate how masks are used during optimisation.

Example 1: If a user wants to optimise his far field pattern at a single frequency using a mask, the user needs to create the mask with the required shape and also create the far field request that should be compared to the mask. The first point in the far field calculation will map to the first point in the mask and the last point in the far field calculation will map to the last point in the mask. All other points of the far field will be compared to points in the mask (linear interpolation is used to ensure a continuous mask).

Example 2: If a user wants to optimise the antenna gain in a predefined direction to have a gain profile that can be described by a mask, the user needs to create the single point far field request and the mask. The gain at the first frequency will now map to the first point in the mask and the last frequency will map to the last point in the mask. The gain at the frequency values within the range will be compared to values in the mask using linear interpolation.

Example 3: Using a combination of the two examples above the user can create a complex optimisation requiring a predefined gain pattern that changes as a function of frequency. The user needs to create the multi-point far field request and create a mask such that the first point of the mask maps to the first point in the far field request at the first frequency and the last point in the mask maps to the last point of the far field request of the last frequency. Suppose the required far field pattern does not change over frequency, then the mask will have the same required far field pattern repeated several times (once for every frequency).

It is clear that very complex optimisations are possible using masks. Care should be taken to create the correct mask for the optimisation since the optimisation will not fail due to an incorrect mask, but the user will not get the optimum result that was expected.

7.1.4 Setting up optimisation Goals

Within each optimisation search, *Goals* must be defined that specify the desired state that the optimisation process should attempt to achieve by varying the specified parameters within the bounded region defined by the parameter definitions (see section 7.1.2). Though multiple Goals may be defined, a valid optimisation search must contain at least one Goal definition. Optimisation Goals are local to each search, and are created under the *Goals* branch of any optimisation search.

Optimisation with FEKO makes provision for multiple Goals in each search. These Goals can be related in various ways to each other in order to define the required result for the optimisation search. The definition of the set of Goals for an optimisation is very flexible, and there are many options available.

A Goal may be created in one of 2 ways:

1. Right-click on the *Goals* branch in the relevant search in the *Optimisation* tree and select the Goal type required
2. Select the *Request* tab and click on the *Add goal function* button.

Structure of an optimisation Goal

All optimisation Goals (irrespective of type) have the same basic structure. In this section the generic concepts and features of all Goals are considered.

Goals are divided into four basic parts.

Focus: The part of the FEKO solution that is to be considered for optimisation. The Focus is based on a quantity that is computed directly by the FEKO solver, and is uniquely identified based on the *Request label*.

Processing steps: A number of conversion steps or mathematical operations that are to be carried out on the Focus before the Goal is evaluated. Some processing steps are 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.

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.

Operator: The Operator indicates the desired relationship between the Focus and the Objective.

The different Goal types are differentiated only by the Focus (and the focus-specific options). The Operator and Objective options as well as many of the processing steps that may be chosen are identical for each Goal type.

Before considering the variations that are available for each Goal type we will consider the options and concepts that are common to the different Goals.

The Request label

The *Request label* for each Goal uniquely identifies the simulation results that are to be considered during the evaluation of the Goal. Every solution request and excitation that is included in the FEKO model can be assigned a label.

In the CADFEKO *Solution* tree, the name (label) of each calculation request and excitation that has been defined is displayed and can be changed by the user. When defining a Goal, the relevant label may be typed into the field. In addition to the text-entry option, the labels of the calculations or excitations defined in CADFEKO that may be referenced in a Goal of that specific type are included in the *Request type* drop-down list. If the *Request label* is selected from this list, then CADFEKO will maintain the reference even if the name of the calculation or excitation is changed in the *Solution* tree.

The labels of calculations and excitations defined directly in the *.pre file (and not in CADFEKO) may also be typed into the *Request label* field. The label of a calculation or excitation defined in EDITFEKO is indicated by a comment field (see section 13.1) added at the end of the line in which the card is defined (see section 11.4.1). No validation of the correct solution type or label validity is performed when a label is typed into the *Request label* field, and any changes must be manually maintained.

It is important to note that all data in the output that is of the correct type and with the indicated label will be included in the Goal evaluation, irrespective of frequency. When defining calculations and excitations in CADFEKO, it is not possible to have two entities of the same type with the same label. In EDITFEKO, however, there is no limitation in this regard. When solving the model at multiple frequencies, the data at all frequencies will be considered in the goal evaluation.

After the first optimisation iteration (during the goal evaluation phase), if no data of the correct type with the correct label is found in the FEKO output (or if the settings in the data request are such that the data relevant to the goal setup was not calculated) an error will be returned.

Focus processing options

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.

$$x \rightarrow x$$

Real/Imaginary/Magnitude/Phase: Allows selection of 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.

$$x \rightarrow \text{Re}(x)/\text{Im}(x)/\text{Phase}(x)/\text{Mag}(x)$$

<i>Unwrap:</i>	Unwraps a phase component — for a phase array, the whole array is considered in the unwrap process. This operator can only be applied directly after <i>phase</i> . $(x \rightarrow \text{unwrap}(x))$
<i>Absolute value:</i>	Takes the absolute value — for an array, the absolute value of each element is taken. $x \rightarrow x $
<i>Average/Minimum/Maximum:</i>	Finds the average, minimum or maximum value of an array — this has no effect on a single value. $x \rightarrow \text{ave}(x)/\text{min}(x)/\text{max}(x)$
<i>Normalise:</i>	Normalises to the largest value in an array — for a single value, '1' will be returned. $x \rightarrow \frac{x}{\text{max}(x)}$
<i>Log:</i>	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. $x \rightarrow \log_{10}(x)$
<i>Offset:</i>	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. $x \rightarrow x + n$
<i>Scale:</i>	Multiplies by a specified scale factor — for an array, each element of the array is multiplied by the scaling factor. $x \rightarrow nx$
<i>Exponent:</i>	Applies an exponent — for an array of values, the exponent of each value in the array is taken. $x \rightarrow x^n$
<i>Undefined:</i>	When a processing step is modified so that a previously chosen processing step becomes invalid, the processing step that is no longer valid reverts to an <i>Undefined</i> state. All <i>Undefined</i> steps must be deleted or redefined before the changes to the goal can be applied.

The Objective

The optimisation Objective may be defined in each Goal as a *Single value* or as a *mask*.

The *Single value* Objective This Objective is defined in the *Value* text-box. The error 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 have been applied, the Objective value will be compared to each of the array values separately, and the cumulative error will be extracted according to the Operator type by a summation of all of the errors.

The Mask Objective A 2D mask may be pre-defined (see section 7.1.3) 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 would typically be used when a quantity that varies with position, observation angle or frequency within one simulation result is to be optimised. The mask array need not be the same length as the computed data array to which it will be compared as the optimiser will use a piece-wise linear fitting on the mask array to determine the values for comparison at the correct points (points as calculated according to the solution set up).

The Operator

There are five Operator types that are common to all Goals.

1. *Equal* - Indicates that the processed Focus should be equal to the Objective.

$$error = \sum_{n=1}^N |Focus(n) - Objective| \quad (7-1)$$

2. *Greater than* - Indicates that the processed Focus should be greater than the Objective.

$$error = \sum_{n=1}^N \begin{cases} |Focus(n) - Objective| & \text{for } Focus < Objective \\ 0 & \text{for } Focus \geq Objective \end{cases} \quad (7-2)$$

3. *Less than* - Indicates that the processed Focus should be less than the Objective.

$$error = \sum_{n=1}^N \begin{cases} |Focus(n) - Objective| & \text{for } Focus > Objective \\ 0 & \text{for } Focus \leq Objective \end{cases} \quad (7-3)$$

4. *Maximise* - Indicates that the processed Focus should be maximized (no Objective is required for this Operator).
5. *Minimise* - Indicates that the processed Focus should be minimized (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 will be zero.

Figure 7-9: The *Impedance goal* button

7.2 Optimisation Goal types

7.2.1 The Impedance Goal

The *Impedance* Goal provides for optimisation based on factors relating to the impedance and admittance at any voltage or current source that is solved as part of the FEKO model. The Focus is identified based on the label of a *voltage source* or *current source* in CADFEKO or of a card-defined excitation of the type A1, A2, A3, A4, AF or AN in EDITFEKO.

Create impedance goal

Goal focus

Request label: VoltageSource1

Focus type: Input impedance

Reference impedance: 50

Focus processing steps

Expression: re(Focus)

Operation	Value
Real	

Add Remove

Goal operator

Operator type: Equal

Goal objective

Value:

Weight: 1.0

Label: ImpedanceGoal1

Create Close

Figure 7-10: The menu for the Impedance optimisation Goal

Focus type

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 consists 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 indicated reference impedance. (Note that 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 multiport model.)

Transmission coefficient: The transmission coefficient $(1 - \Gamma)$ is considered with respect to the indicated reference impedance.

VSWR: The Voltage Standing Wave Ratio $([1 + |\Gamma|]/[1 - |\Gamma|])$ for the observed input impedance is considered with respect to the indicated reference impedance.

Return losses: The return loss $(-20\log|\Gamma|)$ for the observed input impedance is considered with respect to the indicated reference impedance.

Current: The current flowing through the segment on which the selected voltage source is located. (Note that in order to use this optimisation goal, a port with an excitation (or an A1 card (see section 14.4)) must be applied to the segment of interest. The excitation should be set to zero magnitude, and referenced in the *voltage source label* field.)

Reference impedance

The impedance to be used during the calculation of the relevant Focus types can be indicated here. The impedance must be a single non-complex value and is local to each impedance goal (i.e. different reference impedances may be used for different impedance goals in the same optimisation search).

7.2.2 The Near field Goal



Figure 7-11: The *Near goal* button

The *Near field* Goal provides for optimisation based on factors relating to all near fields computed during of the FEKO solution. The Focus is identified based on the label of a *Near fields* request in CADFEKO or of an NF-card in EDITFEKO.

The *Near field* Goal dialog is shown in Figure 7-12. There are three *Near Field* Goal-specific options available in the choice of the Focus that are described below.

Field component

Electrical/Magnetic: The *Electrical* or *Magnetic* part of the near field must be chosen. If the data computed based on the referenced label does not include the chosen part of the near field, 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* selection list.

Figure 7-12: The menu for the Near field optimisation Goal

The coordinate system chosen here differs from the coordinate system chosen as part of the near field computation request, in that 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 request dialog is related to the positioning of the sample points for near field calculation. This distinction makes it possible 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* 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).

$$F_{\text{combined}} = \sqrt{|F_x|^2 + |F_y|^2 + |F_z|^2} \quad (7-4)$$

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 be performed.

7.2.3 The Far field Goal



Figure 7-13: The *Far field goal* button

The *Far field* Goal makes provision for optimisation based on factors relating to all far field quantities computed as part of the FEKO solution. The Focus result is identified based on the request label of a *Far fields* request (see section 6.12.4) in CADFEKO or an FF-card request (see section 14.37).

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.

Figure 7-14: The Far field optimisation Goal definition dialog

The *Far field* Goal dialog is shown in Figure 7-14. Goal-specific options for the *Far Field* Focus choice are described below.

Focus type

E-field: The *E-field* Focus type considers the radiated fields associated with specified 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 and Gain: With this Focus type, only the gain/directivity 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 the *Directivity/Gain* selection in the far field request.

Radar cross section (RCS): This Focus type is only valid for far field solutions that have been computed with an exclusively plane-wave excitation. The RCS Focus type delivers non-complex values (or an array of non-complex values) representing the derived RCS (see section 19-8) 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.

Polarisation

The *Polarisation* option allow specification of the component of the far field to be considered in the Goal.

Horizontal (Phi) / Vertical (Theta): These options allow specific selection of the θ - and ϕ -directed components of the far field. For the *Gain* and *Directivity* 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 (as defined in Equations 19-4 and 19-5). For the *Gain* and *Directivity* 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 (as defined in Equations 19-2 and 19-3). For the *Gain* and *Directivity* 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: The *Axial ratio* 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 (as described in section 19.6 and specifically Equation 19-22). 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.

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:

$$E_{\text{total}} = \sqrt{|E_{\theta}|^2 + |E_{\phi}|^2} \quad (7-5)$$

This value is representative of the power in the far field.

For the *Gain* and *Directivity* Focus types, the polarisation-independent quantities are considered. This is the only *Polarisation* option for RCS.

7.2.4 The S-parameter Goal

The *S-parameter* Goal dialog is shown in Figure 7-16. The *S-parameter*-specific options for the Focus choice are described below.



Figure 7-15: The *S*-matrix goal button

Figure 7-16: The dialog for the S-parameter optimisation Goal

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). It is important to note that if S_{nm} and S_{mn} are computed in an S-parameter solution, then both of these values will be considered in the Goal evaluation. If the consideration of only the coupling in one direction is required, the relevant port should be deactivated in the S-parameter calculation request.

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:

$$RL = -20 \log |S_{nn}| \quad (7-6)$$

Transmission coefficient: The transmission coefficient at the port/s will be considered in the optimisation. The transmission coefficient is calculated from:

$$\gamma = 1 - |S_{nn}| \quad (7-7)$$

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:

$$\text{VSWR} = \frac{1 + |S_{nn}|}{1 - |S_{nn}|} \quad (7-8)$$

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* (S_{mn}), then if the input port is specified as 2, only the values of S_{12} and S_{32} will be considered during the Goal evaluation.

Specify output port number (m): In a similar manner to the input port selection option, this option (when selected) 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* (S_{mn}), then by specifying the output port as 2, only the values of S_{21} and S_{23} will be considered during the Goal evaluation.

7.2.5 The SAR Goal



Figure 7-17: The SAR goal button

The SAR Goal dialog is shown in Figure 7-18. The SAR Focus delivers a non-complex value (or an array of non-complex values) based on the FEKO solution. Besides the general optimisation goal options (see section 7.1.4), there are currently no SAR goal-specific options.

7.2.6 The Power Goal

The Power goal allows the user to optimise 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. Besides the general optimisation Goal options (see section 7.1.4), there are currently no Power goal-specific options.

7.2.7 The Receiving antenna Goal

The receiving antenna goal allows the user to optimise receiving antenna efficiency, received power or power loss. Besides the general optimisation Goal options (see section 7.1.4), there are currently no Power goal-specific options.

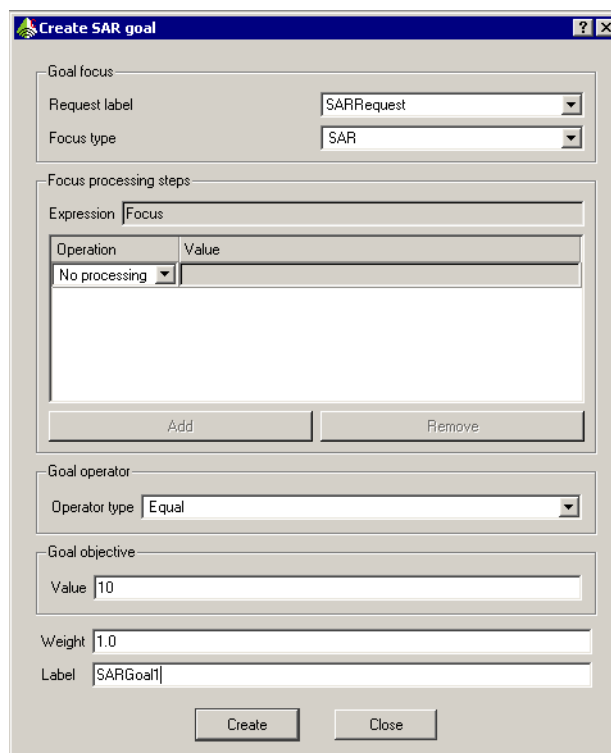


Figure 7-18: The dialog for the SAR optimisation Goal

Figure 7-19: The *Power goal* buttonFigure 7-20: The *RX antenna goal* buttonFigure 7-21: The *Transmission/reflection goal* button

7.2.8 The Transmission/reflection Goal

The *Transmission/reflection* Goal makes provision for optimisation based on factors relating to all transmission/reflection coefficients quantities computed as part of the FEKO solution. The Focus result is identified based on the request label of a *Transmission/Reflection coefficient* request (see section 6.12.3) in CADFEKO or a TR-card request (see section 14.62) in EDITFEKO.

The *Transmission/reflection* Goal dialog is shown in Figure 7-22. Goal-specific options for the *Transmission/reflection* Focus choice are described below.

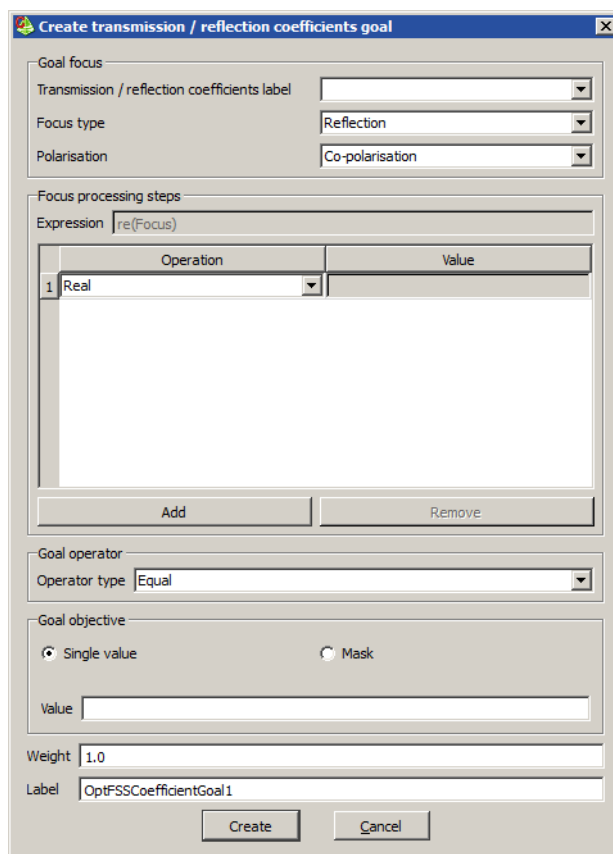


Figure 7-22: The Transmission/Reflection optimisation Goal definition dialog

Focus type

Transmission: The transmission coefficient is calculated as

$$\tau = \frac{E_t}{E_i} \quad (7-9)$$

Reflection: The reflection coefficient is calculated as

$$\rho = \frac{E_r}{E_i} \quad (7-10)$$

Polarisation

The co-polarisation and cross-polarisation is available for optimisation.

7.3 The global Goal: combining and weighting of multiple Goals

When multiple goals are included in a search, in addition to the individual goal definitions, the way in which the goals should be combined during the evaluation of the global goal must be defined. CADFEKO provides two mechanisms by which the evaluation of the global goal can be controlled.

7.3.1 Goal weighting

The first mechanism is a weighting that can be defined for each goal on the *Goal definition* dialog (see section 7.1.4). This weighting is used to modify the contribution of the goal's 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.

The weighting of each goal is shown in brackets next to the tree representation of the goal.

7.3.2 Goal combination tools

A goal combination tool is provided, which allows the extraction of a single error value from a set of goals. 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 a combination tool, goals in one search in the same tree level should be selected. The *Combine goals* tool can then be selected from the *Optimisation* toolbar or menu. The *Combine goals* dialog (shown in Figure 7-23) is launched in which the combination method can be chosen. Goals may be added to an existing combination by right-clicking on the combination in the tree, and selecting the type of goal to add. Goals can be removed from the combination by deleting them. If all goals in a combination are deleted, then the combination is automatically deleted as well. Goals can be copied out of a combination to the root of the goals tree by right-clicking on the goal and selecting *Copy*. The original goal can then be deleted.

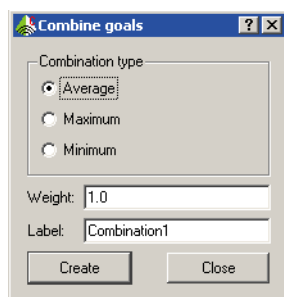


Figure 7-23: The goal combination tool dialog

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 will be returned, while if *Maximum* is chosen, the maximum error will be returned. Each combination can be assigned a weighting which 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.

7.4 Specifying special optimisation solver settings

Special options relating to OPTFEKO can be set by selecting the *Solve/Run* tab and clicking on the *dialog launcher* button, the *Utilities* tab under OPTFEKO.

Restart from solver run n: If this option is selected, then OPTFEKO will attempt to restart the optimisation process from the indicated iteration number (n). The optimisation can only be restarted if the temporary files have been kept during a previous optimisation run. If solution files are missing for a specific optimisation iteration, OPTFEKO will run the FEKO solver to recreate the missing files. If any changes have been made to the model, solution setup or optimisation setup, OPTFEKO will ignore all existing results, and re-compute all results as required.

Delete all files (except optimum): If this option is selected (default), then all of the temporary files will be 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 will be available along with all related simulation results. The optimum model and results are indicated by the addition of `_optimum` at the end of the file name/s. If this option is unchecked then no model or results files will be deleted during the optimisation process.

Farming out of optimisation solutions: This section allows the specification of the distributed computing system when farming out of the solutions during an optimisation is required. The *Configure* button launches a dialog in which the machines in the cluster as well as the number of processes that may be launched on each of the machines may be specified (this is identical to cluster configuration for parallel launching (see section 18.2.2)). Details regarding the use of farming during optimisation may be found in section 20.4.

Part III

Working with POSTFEKO

8 Introduction to POSTFEKO

POSTFEKO is used mainly for two purposes: to validate meshed geometry and to analyse results. Validation of mesh geometry is done so that users can confirm that their models are correct before starting a simulation. This is particularly useful when models are created using EDITFEKO, but is just as relevant for CADFEKO modelling. Analysis of results is the other primary function of POSTFEKO. Once a model has been simulated, POSTFEKO can be used to display and review the results. A variety of tools are available to help visualise data in a constructive manner.

Multiple models, with their geometry (in *.fek files) and results (in *.bof files), can be displayed in a single POSTFEKO project session. The displays are automatically updated each time the model and/or results are updated.

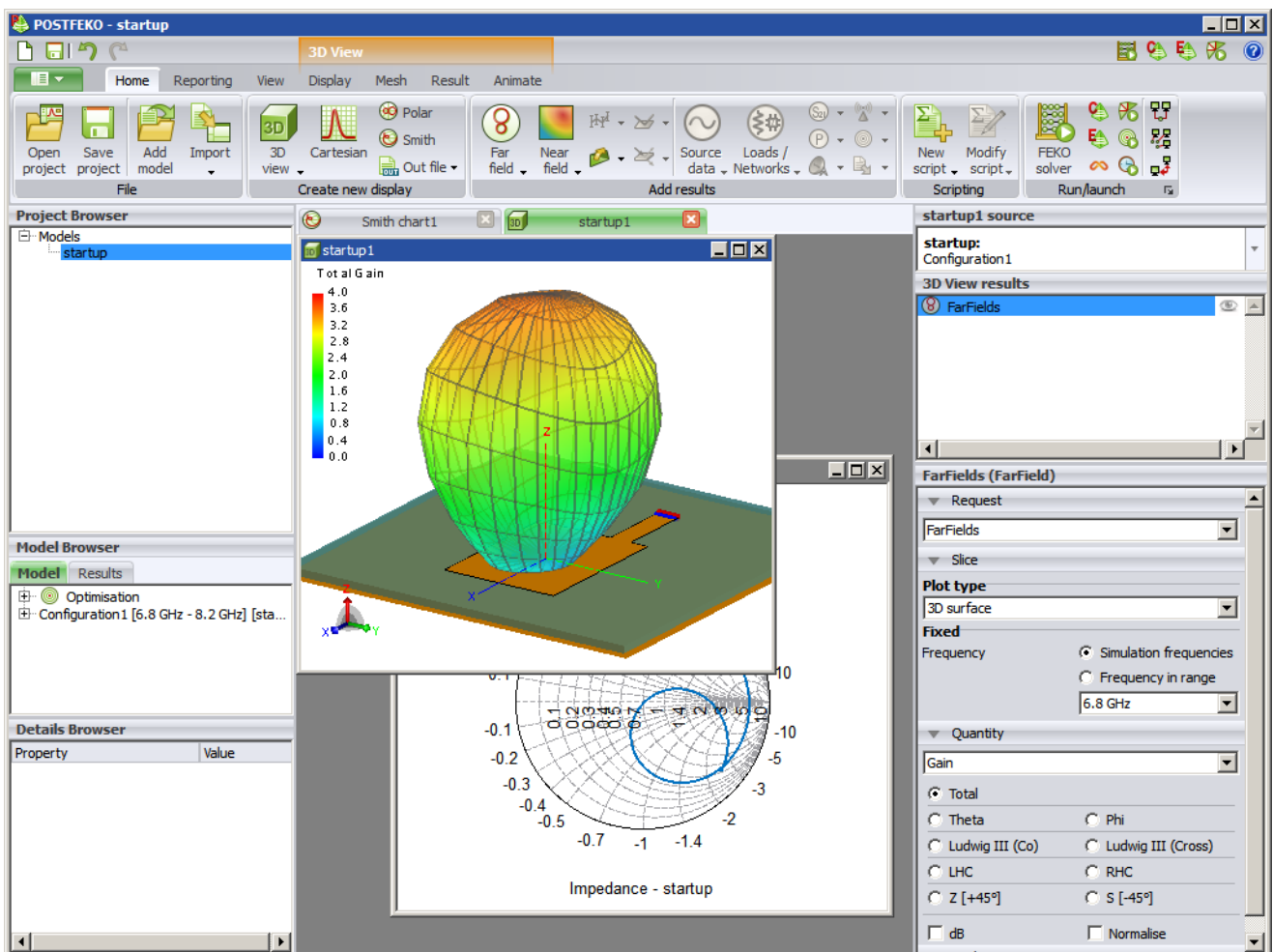


Figure 8-1: The main POSTFEKO display. Here the geometry and 3D results of a single model is displayed.

8.1 Environment

The POSTFEKO environment (shown in Figure 8-1) consists of a ribbon toolbar at the top, a viewing area in the middle, a project browser on the left and a control palette on the right-hand side. The ribbon is used to control views, their contents, and tools that are used for interpreting data. The viewing area can contain multiple 2D and 3D views, each with its own context-sensitive

ribbon tabs. The project browser (hidden by default, but can be shown by *View* → *Window* → *Project*) provides additional information about the project and advanced details about models and their contents. The result palette on the right-hand side allows the user to select the components of a result that should be displayed. Results shown in views can be tuned to provide the desired information by using this palette.

The ribbon is the main mechanism for manipulating views. Data is added to either 2D or 3D views by selecting an enabled result from the ribbon menu. All system controls are contained here and grouped for convenience. The different view types that may be created are *3D views*, *Cartesian graphs*, *Smith charts* and *polar graphs*. When any of these view types are in focus, the ribbon will adapt to provide options that are relevant to that view type and the contextual tabs belonging to the selected view are marked by a coloured bar above them. Whenever there are more items available than shown in the ribbon group, there is a small indicator at the right bottom of the ribbon group (besides the group title) which then opens a dialog window containing extended controls.

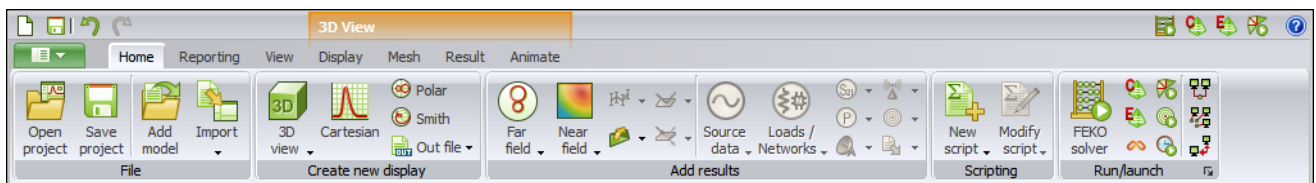


Figure 8-2: Example of the FEKO ribbon.

Views are tabbed in the viewing area by default, but may be rearranged by altering the window settings on the *View* tab. The context sensitive ribbon tabs give the user control over axis settings, titles, captions and a variety of other view specific settings.

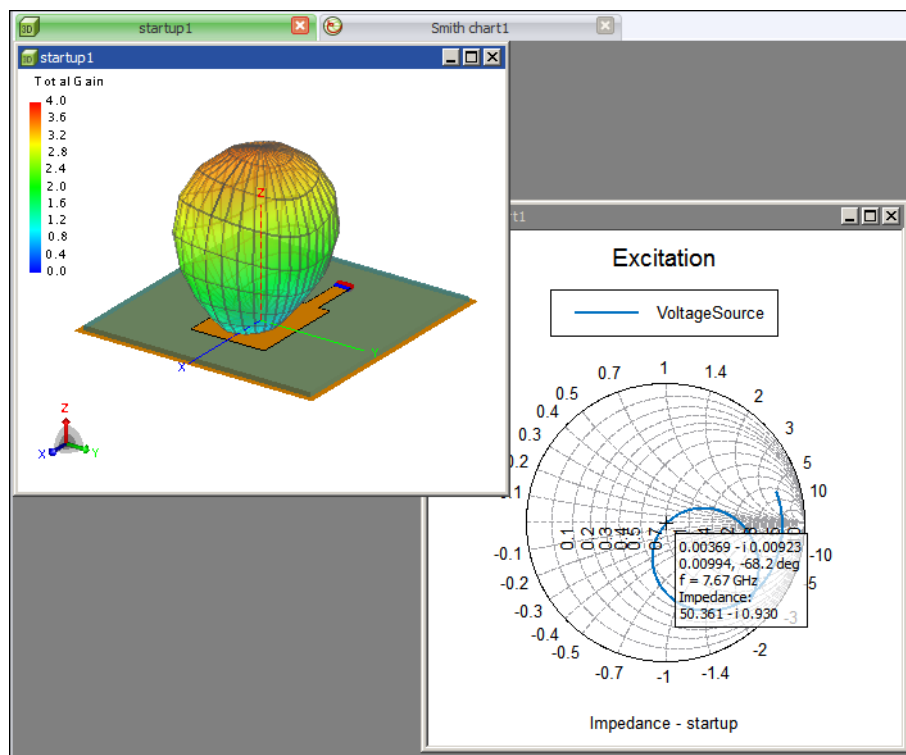


Figure 8-3: The POSTFEKO viewing area

On the right-hand side, the *result palette* is shown (see Figure 8-4). When a result is added to a view, it is populated in a list at the top of the panel. The currently selected solution is also displayed here. For a selected result, the slicing options are provided where the user can supply more specific information about the type of data that should be displayed. Only properties which are relevant for that result are populated in the palettes. The palette will then look different for every type of result that can be displayed.

The *Project browser* panel may be displayed to provide more control over the project session (see Figure 8-4). This *project browser* is not displayed by default. It provides additional information in a tree format about the current project, the models in the project and specific components of a model. The *Project Browser* panel lists all models that are loaded in the current project, as well as all stored or imported data. The *Model Browser* shows all properties of a given model, including details regarding all solution configurations, optimisation details and combined data. The *Details Browser* shows in-depth detail of any component selected in the model browser. It shows all relevant details regarding the entity settings (Note that no information about the calculated results are given here).

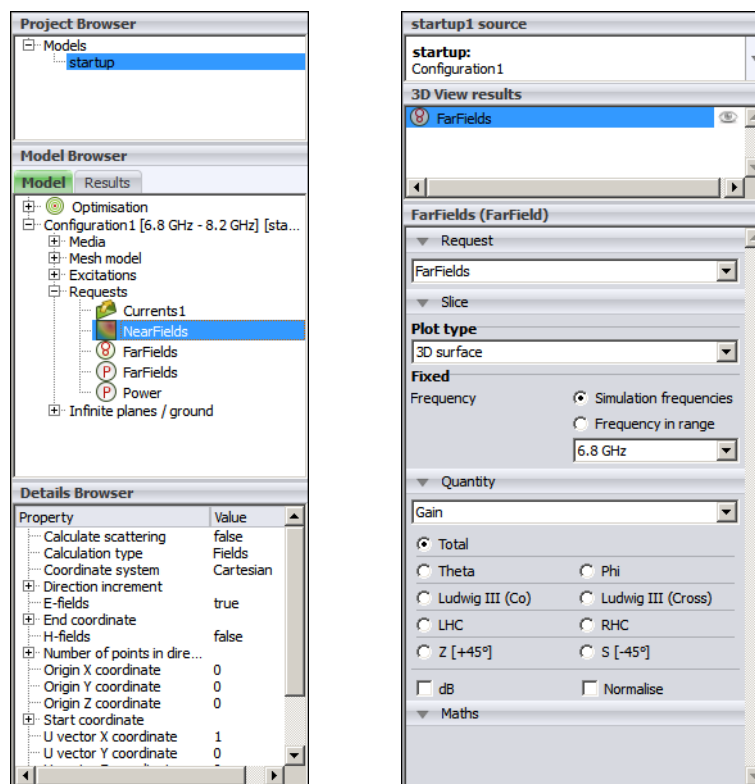


Figure 8-4: The POSTFEKO project browser (left) and quantity palette (right).

8.2 Start page

When a blank instance of POSTFEKO is opened (i.e. no models are loaded), the start page is displayed. It is also displayed when a project is cleared by pressing the *New project* button. The start page provides quick access to recent files, documentation and the FEKO website (<http://www.feko.info>). Additional help can also be found by pressing the *Help* button.



Figure 8-5: The start page.

8.3 Application menu

The application menu is accessed by clicking on the green application button in the top-left corner of POSTFEKO. It provides control over managing project sessions by creating new projects, opening existing projects, saving projects and adding models to a project session. Rendering options and updates settings can also be reached here, along with information about the current POSTFEKO version and third party libraries. The *Recent files* list provides quick access to files that have been used recently. Figure 8-6 depicts the application menu.

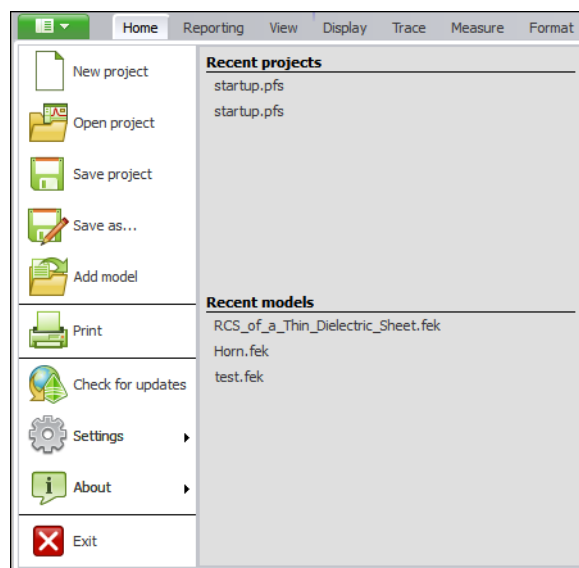


Figure 8-6: Application menu

The *Rendering options* are the same as those described in the section about rendering options (see section 3.1.5). These settings help configure how views are drawn and whether the main

calculations should be done in software or on the system's graphics card. Checking for updates ensures that the most recent version of FEKO is installed on a system. More information on configuring the updater can be found in the section about the FEKO updater (see section 23). All default settings for creating new views, adding data to graphs and other display settings can be modified under the default *Preferences* settings.

8.4 Ribbon tabs: tools and controls

The ribbon is divided into two classes, namely “default tabs” and “context-sensitive tabs”. Default tabs contain controls and tools that are applicable to all view types and provide a central point for controlling the application. For each view type, there also exist the context-sensitive tabs that contain all controls and tools that are only applicable to that view. In general, the way in which a user would like to interact with a 3D view is very different to how they would like to interact with a graph. Context-sensitive menus make it easier to find those tools for a selected view.

8.4.1 Default tabs

The default tabs include:

- *Home*: Groups the most common actions together for quick access. The *Home* tab is a good place to start for most basic operations, including: managing files and projects, adding new views, adding results to views and running suite applications.
- *Reporting*: Tools for exporting data, images and animations are provided. The generation of MS Word and MS PowerPoint presentations can also be managed from the *Reporting* tab.
- *View*: All camera position settings can be altered here. Windows can be tiled, cascaded, closed, etc. The visibility of the project browser and result palette can also be set. The viewing angle can be set to focus on any origin, distance and angles. Figure 8-7 shows the *Transform* view dialog that can be used to manually edit these properties.

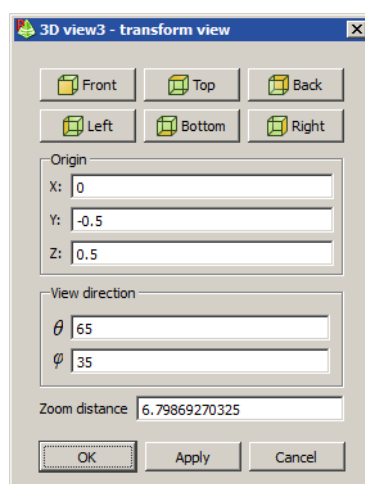


Figure 8-7: View transformation dialog.

8.4.2 Context-sensitive tabs

Context-sensitive tabs depend on the view type and are unique for each 3D view, Cartesian graph, polar graph and Smith chart. Depending on which view type is selected, a different set of operations is made available. This means that only those actions that are relevant to the current selection are available.

3D View context-sensitive tabs

The tabs that will appear when a 3D view is active are:

- *Display*: Affects the visualisation settings for a model. The visualisation of model entities can be set, including infinite planes, legends, symmetry, periodic boundary conditions, etc. Tools such as greyscale and cutplanes are also available. Views can also be duplicated or exported from this tab.
- *Mesh*: All settings pertaining to the mesh visualisation can be set here. This includes colouring and opacity options, as well as visibility settings for each mesh element type. Tools such as distance measurement, searching for mesh elements, highlighting certain solution methods or checking mesh connectivity are also provided here.
- *Result*: Settings regarding a selected result are provided. This includes visibility settings such as opacity, surface/grid visibility, colouring options, extrusion, etc. The requested points can also be displayed for a result, arrows can be displayed and all ray display settings are available. It is also possible to add new results to the view from the *Result* tab, duplicate the selected result or to store a local copy of the result in the current project.
- *Animate*: Provides access to animation settings, including the animation variable (phase, frequency, camera angle, etc.), rendering options and export settings.

2D View context-sensitive tabs

All 2D graphs (i.e. Cartesian graphs, polar graphs and Smith charts) have a *Display* and *Trace* tab. The tabs differ slightly between the different graph types, but their main function stay the same.

- *Display*: The *Display* tab allows the user to: duplicate views or generate equivalent copies of a graph in a different format, edit chart text (e.g. titles or axis labels), set greyscale, edit legend settings and modify axis and grid settings. In general, all settings pertaining to the view can be set in this tab.
- *Trace*: Duplications, new math traces and local copies can be managed here. Also, the rendering settings for the traces (i.e. the drawing order and sampling settings), independent axis scaling and units can be changed.
- *Measure*: Cursors can be managed here that display the value of the traces on a graph. Annotations can also be added to a trace from here, which update with the results and can give additional information about a trace (such as beamwidth, bandwidth, maxima, minima, etc.)

- *Format*: Provides access to font settings, colouring options (e.g. legend box or graph background colours), line settings, marker settings and legend drop shadow settings.

8.5 System operations

Along with the tools that manipulate displayed data, POSTFEKO also gives a set of tools that work on a system level. These tools include saving and loading projects, animating results or models in a 3D view, importing and exporting data (e.g. from measurements, calculations or other simulations) and to store data for later use.

8.5.1 Application Button and quick access toolbar

The green *application button* is the collection of some most used system operations for easier access and also contains recently used files, the options for printing, checking for updates, setting rendering options and showing help and version information as well as closing POSTFEKO. A detailed description follows in Section 8.3.

The *quick access toolbar* is located just above the application button and contains some operations such as (*New project*, *Save*, *Undo* and *Redo*) for one-click access without having to switch the ribbon context.

8.5.2 Save and load project sessions

Saving can be done by clicking on the application button and choosing either *Save* or *Save as*, depending on whether the file-name should be specified. The resulting project file will have a *.pfs extension and will store all settings and references to result files that were present at the time of save. There is also a quick save short-cut in the top-left corner above the application button.

To “load” could either mean loading a saved project, or to load a model into an existing project. The application button will provide both a *Load model* button as well as a list of recent files that were accessed by the user. The *Open project* button on the *Home* tab will restore a saved POSTFEKO project. Both models and projects can be accessed on the start page as well.

8.5.3 Preferences

In many cases, especially when it comes to styling, it is desired to always change a setting to something other than the default. A *Preferences* tool is provided that allows custom settings to be stored. This tool can be found under *Application menu* → *Settings* → *Preferences*. Figure 8-8 shows the dialog. A variety of options can be set, from default display settings to data sampling and presentation settings.

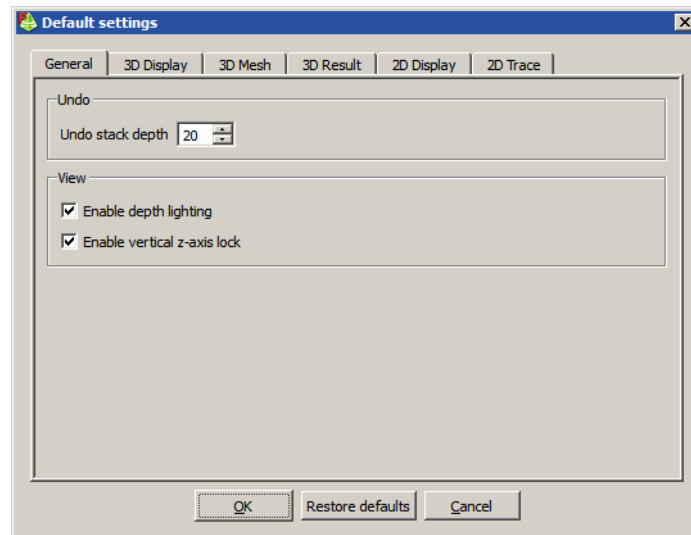


Figure 8-8: Default preferences dialog.

8.5.4 Animation

In 3D views, it is often useful to see how properties change over time. It is possible to animate a view or its results according to:

- Phase
- Frequency
- Phi rotate
- Theta rotate
- Theta & Phi rotate

For phase and frequency animations, results must be displayed that can be animated. For example, currents or near fields may be animated over phase and any result that was simulated over a continuous or discrete frequency range can be animated over frequency. Any 3D model may be animated over the theta or phi angles, since this is purely a camera view animation and does not require any results. Figure 8-9 shows the dialog where these settings are changed.

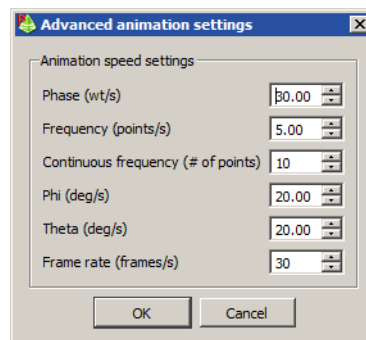


Figure 8-9: Animation settings dialog.

Animating by instantaneous phase gives insight as to how results change over time. The display shows the magnitude of a result at a given phase and increments this phase at a user-defined rate. Similarly, when multiple frequencies were simulated, the results at a certain frequency can be shown. In this case the frequency is incremented and the display of the results are varied accordingly.

8.5.5 Import and export features

Importing and exporting of data into POSTFEKO is also possible. Imports can be performed on most text-based data files, as well as data that is generated by FEKO, but not included in a model's results file. Calculated data can also be exported for external processing or for use in a different project (where the full set of results is not required).

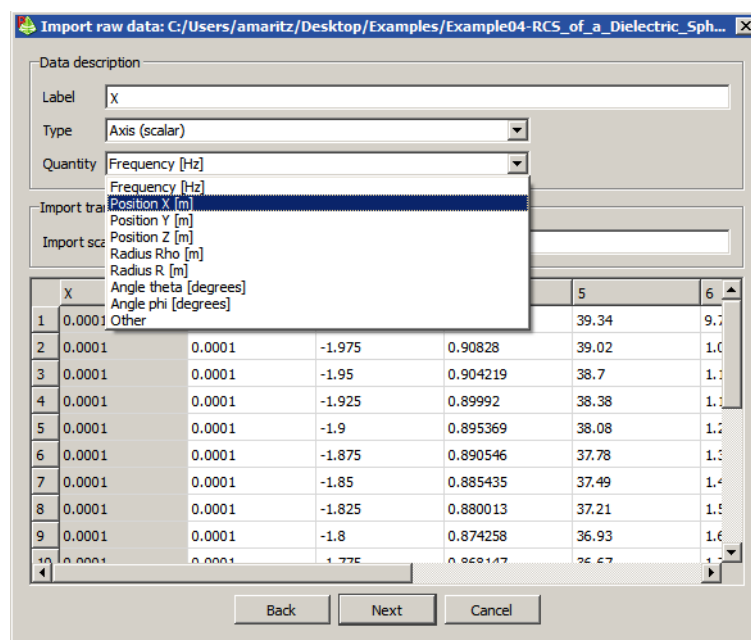


Figure 8-10: Dialog for specifying column details during an import.

Importing

Data that can be imported include

- POSTFEKO graph files (*.pfg files)
- FEKO field results files: electric fields (*.efe); magnetic fields (*.hfe) and far fields (*.ffe)
- Any ASCII based data file. For these files, the formatting of the data must be specified by the user in order for the import to be successful.

Data that was imported can be added to a 2D graph in the same manner as any result. There is an *Imports* button under the *Add results* group where imports can be accessed. The project browser contains an entry for each import under *Stored data*, where imports can be deleted from the project if they are no longer needed.

POSTFEKO will manage imports of familiar types by itself, without the need for the user to specify any further details. However, for the custom import of raw data, an import template must be defined. It is necessary to define how data columns are separated (i.e. with tabs, spaces, commas, etc.), which lines to read and whether column titles are present. The preview section will show where columns are separated and what their titles are.

Once the format is defined, the user must specify what type of data each column contains. The label can be specified, along with any scaling considerations (e.g. if the data is in dB instead of linear, or MHz instead of Hz, etc.). Figure 8-10 shows the import dialog where these properties are defined. Each column can be specified by what type of structure it has, along with more detail regarding the specific contents of the data:

- *Axis (scalar)*: If the column will be used as an independent axis on a 2D graph, this option should be chosen. Quantity options that are available include:
 - * Frequency, position, radius, angle or a user defined quantity.
- *Scalar*: Any scalar result type may be used. Quantity options that are available include:
 - * Far field, near field, voltage, current, power, specific absorption rate, impedance/admittance, scattering parameters, axial ratios, gain/directivity, radar cross section, voltage standing wave ratio, reflection coefficient, user defined quantities and several other typical data types.
- *Complex pair (Real + Imaginary)*: If two adjacent columns contain the real and imaginary components of a complex number, this option should be chosen.
- *Complex pair (Magnitude + Phase)*: If two adjacent columns contain the magnitude and phase (in degrees) of a complex number, this option should be chosen. Complex pairs can be classified as any of the above quantities that can be complex, such as:
 - * Far field, near field, voltage, current, impedance/admittance, scattering parameters, reflection coefficient, or a user defined quantity.

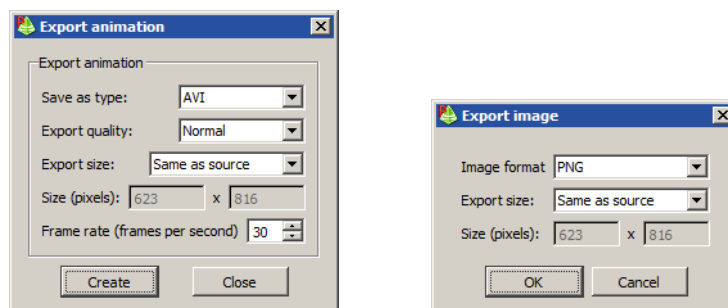


Figure 8-11: Export dialogs for animation (left) and image (right) exports.

Exporting

The export operation can refer to

- Exporting an image of the current view (2D or 3D)
- Exporting an animation in a movie file format (only 3D)

- Exporting data in a text format (only 2D)

When exporting an image of the current view, a visual copy of whatever the view contains will be stored. The user must specify the file location, file name, image format and image resolution.

The animation export operation works in a similar fashion to the image export. For animations the file location, file name, file format, export quality, video resolution and framerate needs to be specified.

For exporting data on 2D graphs, only the file location and file name is required. If, however, continuous data is being exported, the number of sample points must also be specified.

8.5.6 Storing data and using stored data

It is possible to store a copy of most calculated results in a project for later reference. A stored result will be unaffected by changes to the model, making it a useful reference point against future data. Stored data can be accessed in the same way as data from a file. When adding a result from the ribbon, a new entry for *Stored data* is created containing the stored result.

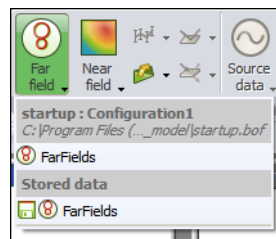


Figure 8-12: Accessing stored data from the ribbon.

In general, all results that can be plotted on a 2D graph can be stored. Results that cannot be stored include: cable paths, error estimates, imported data, rays, already stored data, currents and charges.

8.5.7 Math scripting using Lua in POSTFEKO

Scripting provides a method of creating custom results within the POSTFEKO environment. Any supported result type can be created using a script, imported from a file or modified from existing session results. These based on the Lua scripting language; a scripting language commonly embedded in applications. The Lua language is simple and extendible, making it well-suited to generating scripts for a wide variety of applications.

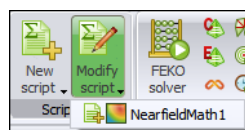


Figure 8-13: Using the ribbon to create or modify scripts.

All scripts are stored in the POSTFEKO session and can be modified or re-evaluated using the editor. See the section on “Using math scripts and Lua” (see section 9.2.9) for more information

on how the editor is to be used. Data generated by a script can be treated the same as any of the internal types, namely: near fields, far fields, sources, loads, networks, s-parameters, power, transmission/reflection coefficients or an arbitrarily defined format.

9 Using POSTFEKO

The POSTFEKO application can be launched from different suite components and in different ways, including a console command prompt, desktop icons, or from other suite applications. How this is done will be discussed, as well as what to do once the application starts up.

9.1 Launching POSTFEKO

POSTFEKO can be launched (i) from the command line in a console environment, (ii) by double clicking on the POSTFEKO icon, or (iii) by launching POSTFEKO from other suite components such as CADFEKO or EDITFEKO. If the application icon is used to launch POSTFEKO, then no models will be loaded and the start page will be shown. Launching POSTFEKO from other suite applications will automatically load the model into a new project.

The command line method give users a choice as to how they would like to launch POSTFEKO. If a model (or set of models) is specified, then it will be added to a new project; otherwise a blank project will be given. Command line arguments can be used to specify additional parameters when launching POSTFEKO. Arguments that may be used are listed in Table 9-1:

Table 9-1: Command line arguments for launching POSTFEKO

Argument	Description
--version	Displays the current version of POSTFEKO.
*.pfs, *.fek, *.bof *.pre, *.pfg, *.wfg	Files with these extensions may be loaded into POSTFEKO.

Launching POSTFEKO from a console with no command line arguments is the same as launching it from the desktop. No models will be loaded and a blank project will be presented. A list of arguments can be given and file formats with the extensions *.pfs, *.fek, *.bof, *.pre, *.pfg, *.wfg and *.out. Note that only one *.pfs file may be listed at a time and that the *.out file argument will only show a message indicating that *.out files cannot be loaded directly.

9.2 Typical user scenarios

Common user scenarios are presented to illustrate how to perform common operations. Scenarios include operations such as opening models, clearing projects, populating views, modifying the displayed data and using a range of POSTFEKO tools to interpret and verify the model and its results.

9.2.1 Managing projects/models

POSTFEKO can manage multiple models simultaneously in a given project. Models can be added to a project by clicking on the *Add model* button, using the start page (See section 8.2.), or by opening POSTFEKO with a model file (e.g. when launching POSTFEKO from CADFEKO).



To add files to a project, click on the *Add model* button. It is available in the *Home* → *File* group, as well as on the application menu. Models can also be added to a project from the start page from the recent files list.



Whether a single model or a complex project was created, the project can be saved for later use. Clicking on the *Save* or *Save as* buttons allow the user to specify a location and name for the current project. Projects that have been saved can later be opened again by using the start page or by clicking on the *Open project* button.



The *Open project* button provides access to any saved project. Such projects have a *.pfs extension and must be explicitly generated by the user. Project sessions can also be opened from the start page (if they have recently been created/opened) or from the application menu.

See also the options for launching POSTFEKO with multiple models or a project using the command line as described in [section 9.1](#).

9.2.2 Add results to a view

Results can be added to both 2D or 3D views. POSTFEKO will only enable the buttons for those results that are present in the current model or project. Clicking on a result button will provide a list of all results of that type that may be added to the current view. Note that a 3D view is always associated with a specific configuration for a single model. For graphs, however, any valid data for all the loaded models may be added.



Far fields: Adds far field results to a valid view.



Near fields: Adds near field results to a valid view.



Error estimates: Displays error estimates in a 3D view.



Currents: Adds currents to a valid view.



Rays: Adds rays to a 3D view.



Sources: Adds sources to a valid view.



Loads/Networks: Adds load/network data to a valid view.



S-Parameters: Adds scattering parameters (s-parameters) to a valid view.



Power: Adds power data to a valid view.



Imports and Scripts: Adds imported data or data generated by a script to any view on which the data is valid.



Transmission/reflection: Adds transmission/reflection coefficients to a valid view.



Optimisation: Adds optimisation data to a valid view.



Receiving antenna: Adds receiving antenna data to a valid view.



Standard absorption rate (SAR): Adds Standard Absorption Rate (SAR) data to a valid view.

3D views are able to display: far fields, near fields, error estimates, currents and rays. In cases where the SAR is calculated at a known location, the SAR may also be displayed.

Cartesian graphs may contain all data, *except* for: error estimates and currents on triangles. Polar graphs may contain data where an angle defines the independent axis. That is, any result that varies according to Theta (θ) or Phi (ϕ). Near and far fields are the only results that meet this criteria. Smith charts may contain complex source data, such as impedance or s-parameters. For all 2D graphs, it is possible to import data that meet the criteria for being added to a view. See Table 9-2 for a summary of the valid results on each graph type.

Table 9-2: Summary of which results may be plotted on various graphs.

Result type	Cartesian	Smith	Polar
Wire segment data (Charges/Currents/Error Estimates)	X		
Far Fields	X		X
Impedances	X	X	
Loads	X		
Near Fields	X		X
Networks	X		
Power	X		
Sources	X	X	
S-Parameters	X	X	
Transmission/reflection coefficients	X		
Optimisation results	X		

9.2.3 Convert graph types

It is possible to convert from one graph to another if the data is compatible with both. Table 9-2 summarises which result types may be plotted on which graphs. Note that there are some restrictions:

- Charges, currents and error estimates on mesh triangles cannot be plotted on 2D graphs



Figure 9-1: The (a) *Duplicate view* (b) *Cartesian copy* (c) *Polar copy* and (d) *Smith copy* buttons

- Only data that varies over angle can be plotted on a polar graph

In the *Display* → *Duplicate* group, each graph provides the option to duplicate itself (Figure 9-1 a), or to create a different type of graph (Figure 9-1 b, c, d). For instance, a Cartesian graph will provide the options to *Duplicate view* (Figure 9-1 a), create a *Polar copy* (Figure 9-1 c) or to create a *Smith copy* (Figure 9-1 d) that contains the same data. If the traces on the source graph are incompatible with the destination graph, an error will be given with a list of the traces that do not adhere to the requirements.

9.2.4 Using 2D graphs (Cartesian, Smith and polar)

There are three graph types available in POSTFEKO, namely the Cartesian graph, the polar graph or the Smith chart. Mostly, the same display options are available. However, for each graph, there may be some options that are specific to that graph. To see a list of what types of data may be displayed on which graphs, see Table 9-2. The *Trace* tab is common to all graphs and provides control over the styling of the plotted data.

The *Display* tab

The *Duplicate* group The *Duplicate* group provides the options to make a copy of the current graph (Figure 9-1 a). Simply duplicating the view will provide an identical copy of the current view, complete with all settings (excluding cursors). The functionality is also provided to derive a graph of a different type from the current graph (Figure 9-1 b, c, d). Section 9.2.3 contains a more detailed explanation of how to convert between different graph types.

The *Display* group Any text (other than the text for legend entries) can be set in the *Display* group with the *Chart text* button. Other options provided by the *Display* group include setting the view to *Greyscale* or to add a *Minor grid* (i.e. additional minor grid lines will be added between the existing grid lines). More information on changing the text on a graph can be found in Section 9.2.8.

The *Legend* group Legends can be added to either predefined positions, or by manually selecting a location using the tools provided in the *Legend* group. The legend entry for a selected trace can also be modified with the *Trace text* button. Section 9.2.6 further discusses the use of legends for both 2D and 3D views.

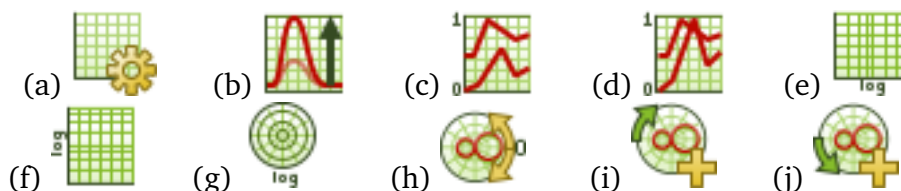


Figure 9-2: Axes group buttons

The Axes group The Axes group is available for Cartesian and polar graphs. Functions that are available here include setting the data ranges (Figure 9-2 a; see also Section 9.2.8), normalising the graph (Figure 9-2 b, c, d), changing an axis scaling to a logarithmic scale (Figure 9-2 e, f, g) or (in the case of polar graphs) changing the orientation of the graph (Figure 9-2 h, i). To normalise the data on a graph, select the *Normalise* (Figure 9-2 b) button. One of two normalisation modes can then be utilised:

- *Normalise to maximum of all traces* (Figure 9-2 c): This setting will look at all of the traces that are visible on a graph to determine the maximum value between them. All traces will then be normalised to that value, meaning that all of the traces will maintain their relative scaling to one another.
- *Normalise to maximum of individual traces* (Figure 9-2 d): This setting will look at each trace in isolation. The maximum value of that trace will be used for normalisation. The effect is that every trace will have an absolute maximum of 1.

Polar graphs are divided into 360° steps. The position of the 0° position can be set by using the *Orientation* (h) options. The orientation can be set to North, South, East or West. In addition, the direction in which the angle grows can be set to either a *Clockwise* (Figure 9-2 i) or *Counter-clockwise* (Figure 9-2 j) direction. The direction of growth is only enabled if the orientation has been set manually (Figure 9-2 h).

The Grid group The Smith chart has a unique group that allows the user to specify the type of Smith chart to use. The choice between an admittance and impedance Smith chart is provided.

The Trace tab

The *Trace* tab mainly controls the styling of the plotted data. Settings pertaining to the sampling of continuous data are also available.

The Manage group The *Manage* group provides the options to *Duplicate* the trace, to create a *New math* trace or to *Store a copy* of the currently selected data in the project session. Duplicating the view will result in another trace that contains the exact same settings as the current one. Math traces can be used to perform calculations on data sets or to provide purely mathematical reference curves. Storing a local copy of a data set saves the current state of the displayed data. This data can then be compared to later runs of the same model (or to different models) independently of how the models are changed.

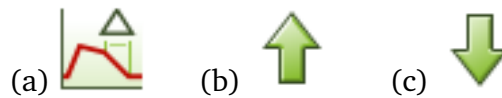


Figure 9-3: Rendering options for traces

The *Rendering* group *Rendering* options affect how the traces are drawn on the graphs. For continuous data, the sampling can be specified (Figure 9-3 a). The default rendering is automatically determined based on the sampled data. However, when the sampling is set, the amount of points can be reduced or increased to suit requirements. Raising (Figure 9-3 b) or lowering (Figure 9-3 c) a trace is purely a display feature. The top traces will be visible above lower traces.

The *Units* group For Cartesian and polar graphs, there is a *Units* group that provides control over the independent axis, as well as the displayed units for the traces that have been added to the graphs. The units which best fit the displayed data are automatically determined based on the range of the data. This *Auto* mode can be overwritten by manually specifying the desired display units.

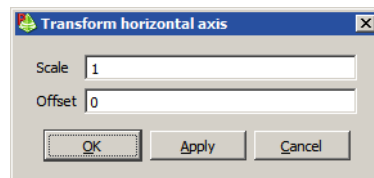


Figure 9-4: Transform the independent axis of a graph



Figure 9-4 shows the independent axis transformation dialog. The independent axis can be stretched or shrunk using the *Scale* unit and a constant offset can be given using the *Offset* unit.

The *Measure* tab

The measure tab helps with the reading off and interpretation of plotted results. Two classes of measurements are provided, namely “annotations” and “cursors”. Annotations can be added to a trace and highlight values of interest. The annotation will update along with the data and always display the value according to its definition. Cursors are more dynamic in the sense that they can be dragged around until they are placed at the desired positions. Cursors have the benefit of being able to read data off of several traces simultaneously, but suffer from the limitation that they cannot update along with the results.

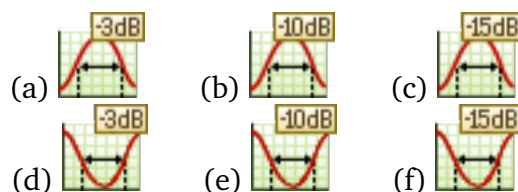


Figure 9-5: Source annotations group buttons

The *Source annotations* group Source annotations are used primarily for indicating bandwidths. Due to the varying definitions of “bandwidth” between industries and applications, several definitions of both transmission (Figure 9-5 (a)–(c)) and reflection (Figure 9-5 (d)–(f)) bandwidths are provided. The corresponding values for linear traces are supported. Reflection bandwidths are typically used for applications such as antenna problems, where transmission bandwidths are used more for filters or other multi-port problems.

Note that the -3 dB bandwidth refers to the half power bandwidth of an appropriate result, which is closer to -3.01029995663981 dB.



Figure 9-6: Far field annotations group buttons

The *Far field annotations* group Figure 9-6 show all annotations that are specific to far field results. These focus on the pattern data and provide various definitions of beamwidth and the sidelobe level. The half power beamwidth (Figure 9-6 (a)), first null beamwidth (Figure 9-6 (b)) and null to null beamwidths (Figure 9-6 (c)) are provided. The sidelobe level (Figure 9-6 (d)) is defined as the ratio between the maximum beam strength divided by the second largest radiated beam strength.

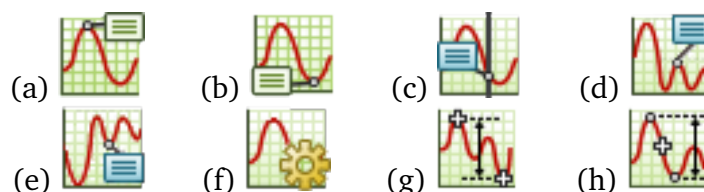


Figure 9-7: Custom annotations group buttons

The *Custom annotations* group The annotations defined so far refer to information that can be extracted from a trace with a very specific type of data. More generic definitions can be found under the *Custom annotations*. Three definition methods exist for defining annotations. All of the previous definitions are defined in terms of one of these methods:

Single point A single point of interest is isolated. Points that are typically of interest include maxima, minima or the value at a specific point. Figures 9-7 (a)–(f) are single point values. Selecting the *Other* definition method will show the dialog in Figure 9-8 (left).

Delta Two points are defined and the difference between the points is used. Values such as sidelobe level are defined using this method. Choosing to create a *Delta* definition will give the dialog in Figure 9-8 (centre).

Derived width A single point of interest is isolated. Two surrounding points are then derived from this value. Values such as bandwidth are defined using this method. Choosing to create a *Derived width* definition will give the dialog in Figure 9-8 (right).

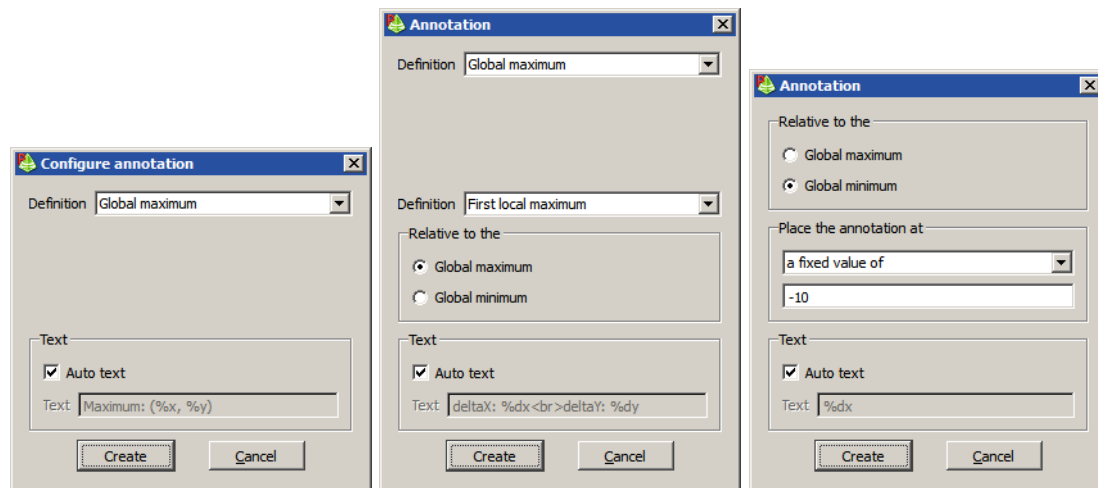


Figure 9-8: Defining a custom annotation using the three definition dialogs.

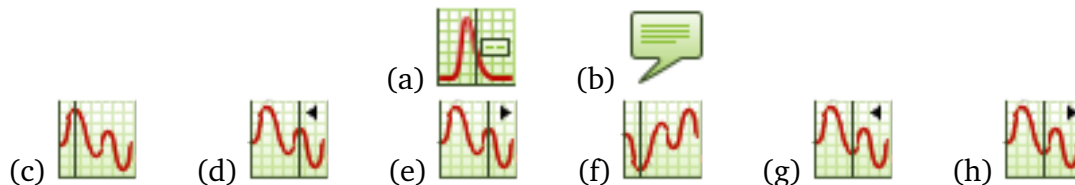


Figure 9-9: Measurement group buttons

The *Measurement* group Cursors are available to help read off information from a graph. The *Cursors* (Figure 9-9 a) button enables the cursors. Each graph type has a different type of cursor. Cartesian and polar graphs have two cursors. By enabling the *Cursor table* (Figure 9-9 b), a summary of the information is presented. This table contains both the data at the displayed points for all traces, but also the difference between the two points for all traces. Note that the Smith chart only has one cursor per trace.

If a cursor is moved outside of the visible region of a graph, a handle will appear in the corner of the graph so that the cursor can still be moved.

Several predefined positions for cursors are defined per trace. Using any of these buttons will make the currently selected cursor jump to the indicated location. The selected cursor is indicated by a solid line. The buttons in Figure 9-9 (c)–(e) will respectively jump to the global maximum, the next local maximum to the left and the next maximum to the right. Similarly, the buttons in Figure 9-9 (f)–(h) will jump to the corresponding minima.

The *Format* tab

The *Font, Colour and Effects* groups Settings that relate to the styling of a graph are provided on the *Format* tab. Fill colours, text colours, fonts, shading, text styling and general tools that pertain to the appearance are provided.

The *Line and Marker* groups Access to the formatting tools for the line and markers is also provided. For both lines and markers, the style, colour and weighting can be set. In the case of markers (Figure 9-11 a), there are three methods by which they can be drawn: markers can

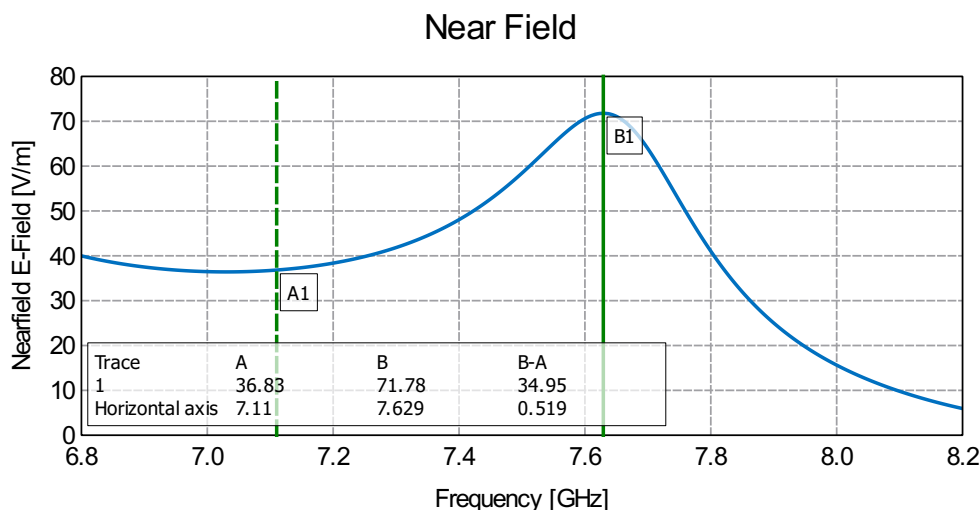


Figure 9-10: Cursors provide a means of quickly reading off data from a graph.



Figure 9-11: Marker placement settings

be drawn on the calculated frequencies (Figure 9-11 b), sparsely spaced markers (Figure 9-11 c) can be drawn at constant intervals and densely spaced markers (Figure 9-11 d) can be drawn at constant intervals. The latter two options are trace decorations that will always be visible in a view, irrespective of the zoom level.

9.2.5 Using 3D views

The 3D view is a powerful tool for getting a feel for how the electromagnetic properties of an environment behave. By setting the display properties of the meshed geometry, the theoretical model properties and the results, information can be presented in such a way as to aid understanding. For each category, a tab exists that groups similar features together and make finding features more intuitive.

The *Display* tab

The *Display* tab is used to change visibility settings of most of the model entities. Tools such as legends, cutplanes and axes are also available to help visualise the model and calculated results to help derive useful information. The *Display* tab is the main controlling tab for the 3D view. This means that the duplication and export of the views are also controlled from here.

The *Display* group



Figure 9-12: The (a) *Cutplanes* (b) *Greyscale* (c) *Boundary* and (d–f) *Annotation* buttons

Figure 9-12 shows the *Display* group, which contains buttons that enable greyscale (b) and the bounding box (c) visibility. The bounding box visibility applies to the geometry, but ignores the displayed results and non-mesh geometry. Cutplanes can be added by clicking on the *Cutplanes* button (a). Figure 9-13 shows the dialog that is presented.

Annotations in the 3D view can be shown or hidden by clicking on *Annotations* button (Figure 9-12 (d)). To select how the annotations must be shown, two options are available. The first is to highlight the selected element and to show an annotation (*Highlight and annotate* elements as shown in Figure 9-12 (e)). The second is to only highlight the element (*Highlight elements* as shown in Figure 9-12 (f)).

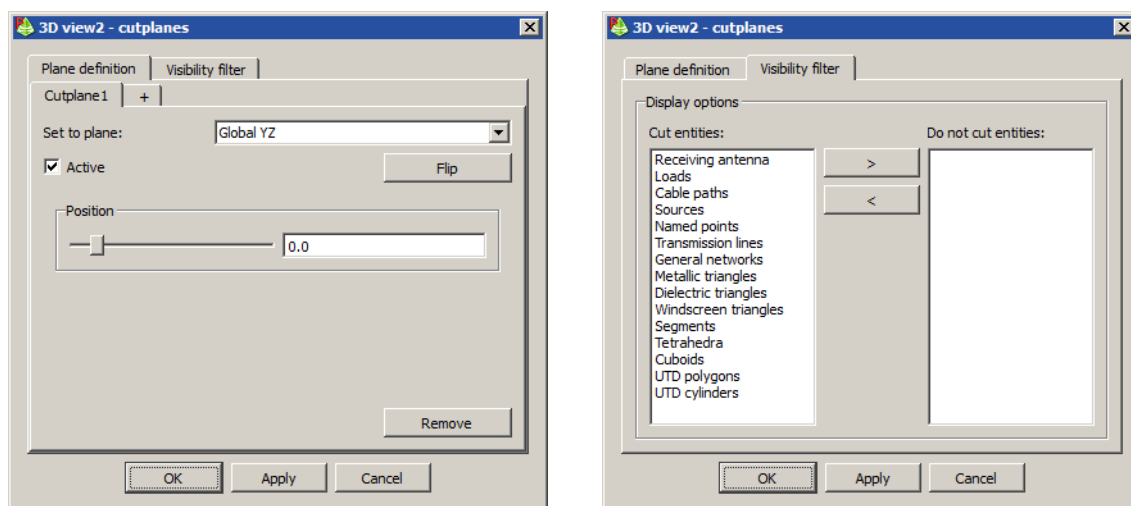


Figure 9-13: The (left) *Plane definition* and (right) *Visibility filter* tabs

On the plane definition tab, the location of the cut plane can be set by defining a flat plane. The *Flip* button alternates the normal direction of the plane, which in turn determines which side of the plane will be hidden. Multiple planes may exist and a given plane can be removed or deactivated on the tab for that plane. A global *Visibility filter* (shown on the right in Figure 9-13) provides control over which entities should be affected by the cutplane. By default, everything that can be visualised and is in the model will be affected. To change this, move the desired components over to the *Do not cut entities* list on the right and click *Apply*. Note that the visibility filter is shared between all cutplanes.



Figure 9-14: The (a) *Add legend to a position* and (b) *Individual range* buttons

The Legends group Figure 9-14 shows some of the buttons that are typically used to manipulate a legend. Legends can be added to any of the four corners in a 3D view. Each of the legend

buttons will have a list of entities that it can bind to, depending on what is currently being displayed. The ranges of the legends (and in effect the colouring of the results) can be set using the *Individual range* settings as well as the advanced scaling settings found in the advanced dialog. Legends are discussed in more detail in section 9.2.6.

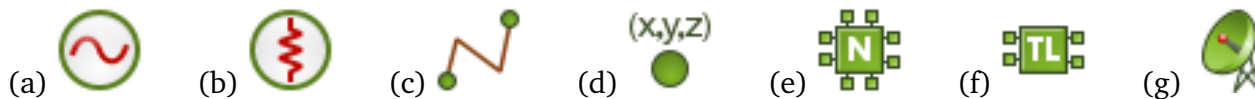


Figure 9-15: Display entities group buttons

The *Entities* group Each of the buttons in Figure 9-15 shows or hides all components of that type in the 3D view. By default, all entities except for named points are visible. Clicking on any enabled button will toggle that entity's visibility. The buttons in Figure 9-15 are: *Sources* (a), *Loads* (b), *Cables* (cable paths) (c), *Points* (named points) (d), *Networks* (general networks) (e), *TX line* (transmission lines) (f) and *RX antenna* (ideal receiving antennas) (g).

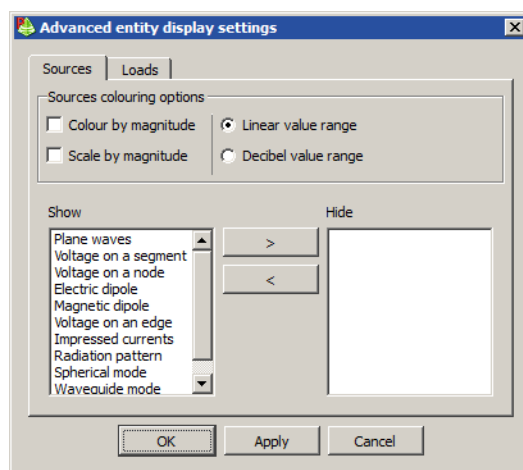


Figure 9-16: Advanced source and load display settings

The buttons in Figure 9-15 only provide a show/hide functionality. For more options regarding the visibility of sources and loads, the advanced dialog shown in Figure 9-16 can be used. For both sources and loads, a *Show* and *Hide* column is presented which contains all sources/loads present in the model. By placing entries in the *Show* column, sources/loads of that type will be shown *only if* the visibility for the entity type is enabled. Additional options are provided for source visualisation. Here, sources can be coloured and/or scaled by magnitude. This is often used in conjunction with aperture sources, electric dipoles, magnetic dipoles and impressed currents.



Figure 9-17: Display of (a) *Symmetry*, (b) *PBC* (Periodic Boundary Conditions) and (c) *Infinite planes*.

The *Method display* and *Infinite planes* groups Figure 9-17 shows the main display buttons for enabling the visualisation of symmetry planes, periodic boundary conditions (PBC) and infinite planes. As with the entity visualisation, these buttons merely enable/disable the visibility of

the method. For infinite planes, the opacity can also be set and the infinite layers may be shrunk to help improve rendering.

The Axes, Export and Duplicate groups are simple in function. Exports are discussed in more detail in section 8.5.5. The *Duplicate* view button does exactly what it says, in that an exact replica of the current view is created. The axis visualisation provides two main axis settings: show/hide the global axis (at the origin of the model) and show/hide the mini axis (in the bottom-left corner of the view). Axis tick marks can be shown on the global axis if it is displayed.

The Mesh tab

The *Mesh* tab also contains display settings, but since there are so many that apply specifically to the mesh, they are grouped together for convenience. The groups include settings for *Rendering*, *Opacity* and mesh *Visibility*, along with a set of tools to help inspect the model.

The Rendering group affects how the mesh is drawn in the 3D view. Features in this group are displayed in Figure 9-18. The “colour by” options display the mesh using different colours. Available options include colouring by: *Element face media* (Figure 9-19 a), *Model outline* (Figure 9-19 b), *Element region media* (Figure 9-19 c), *Element label* (Figure 9-19 d), *Element normal* (Figure 9-19 e) and *Element type* (Figure 9-19 f). The main “colour by” button (Figure 9-18 a) will then change and show the according icon that is selected for colouring.



Figure 9-18: Mesh rendering options

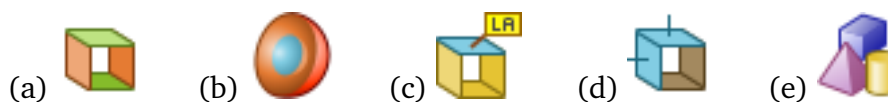


Figure 9-19: Mesh colouring options

Element normals can also be displayed using the *Normals* button (Figure 9-18 e), which displays a line indicating the normal direction. For segments, the radii can be artificially enlarged using the *Segment radius* button (Figure 9-18 b), which multiplies the defined radius by a factor between 1–10 times the original radius. Figure 9-20 shows the dialog for displaying the principal direction for anisotropic layers (Figure 9-18 d). Anisotropic layers are applied on a label. For each geometry part that is covered with an anisotropic dielectric layer, the option is made available to show the principal direction. Since multiple layers can be present, it is also necessary to specify which layer to show.

The Opacity group Figure 9-21 shows the opacity group that pertains to the opacity of mesh elements. Since windscreen and aperture triangles have an inherent opacity setting, they can be customised individually. *Mesh opacity* (a) applies to all visualised geometry and the effect on aperture (c) and windscreen triangles (b) are compounded. For a setting of 0%, the element will be purely transparent and will not be visualised. At 100%, no transparency will be applied.

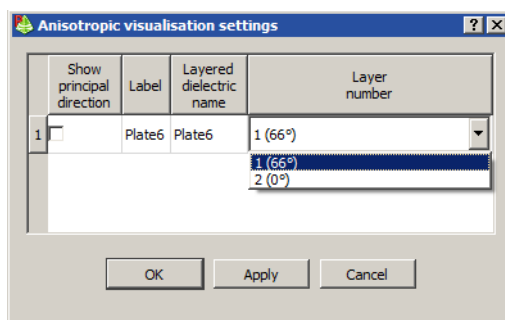


Figure 9-20: Anisotropic layer visualisation settings



Figure 9-21: Opacity settings

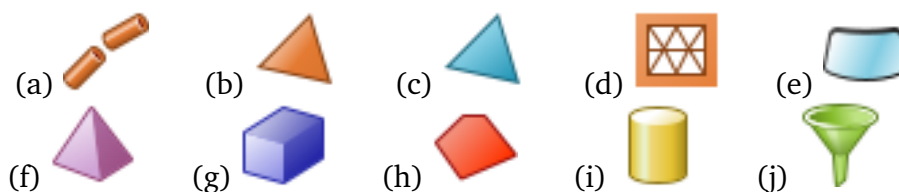


Figure 9-22: Mesh visibility settings

The Visibility group For most mesh elements, it is possible to individually set the visibility for the faces, edges and vertices of the triangles. Elements include: wire segments (Figure 9-22 a), metallic triangles (Figure 9-22 b), dielectric triangles (Figure 9-22 c), aperture triangles (Figure 9-22 d), windscreen triangles (Figure 9-22 e), tetrahedra (Figure 9-22 f), cuboids (Figure 9-22 g), UTD polygons (Figure 9-22 h) and UTD cylinders (Figure 9-22 i). For the some special cases, different settings are given. These cases are wire segments (surfaces, lines, vertices), tetrahedra (faces, edges, vertices, volume) and UTC cylinders (faces, edges). The visibility filter

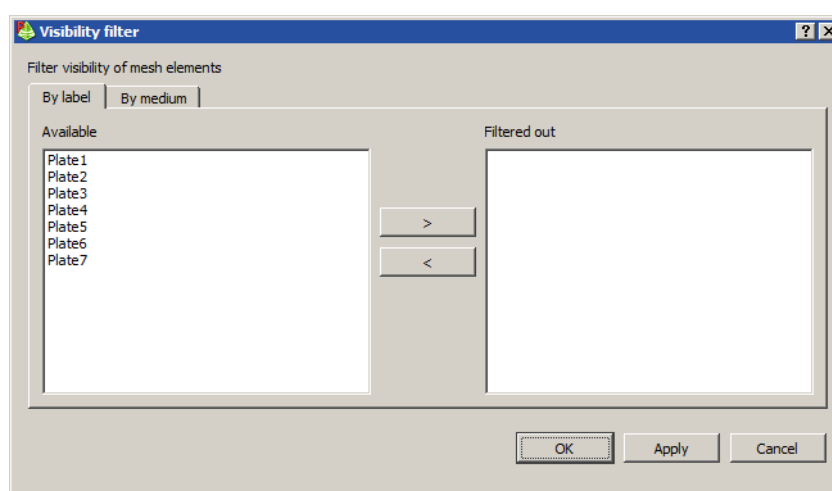


Figure 9-23: Mesh visibility filter

(Figure 9-23) provides some additional control over the visibility of mesh elements. The visibility filter is accessed through the associated button (Figure 9-22 j) and provides the functionality to filter out mesh regions with a specific label, or to filter out specific media.

The Tools group

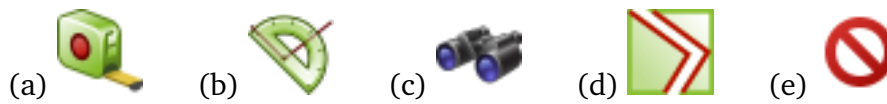


Figure 9-24: Mesh tools

The mesh tools group provides a set of tools that can be used to help identify specific parts of a geometry or to ascertain certain properties about the geometry. The *Measure distance* feature (Figure 9-24 a) places the 3D view into a mode that will measure the distance between any two selected points. To use the distance measurement mode, use the following procedure:

- Click on the *Measure distance* button to open the distance measurement tool.
- Hold down <Ctrl>+<Shift>
- Click to remember the first point.
- Repeat this process for the second point. The distance will be displayed in the tool.

The *Measure angle* tool works in much the same way, except that three points are required.

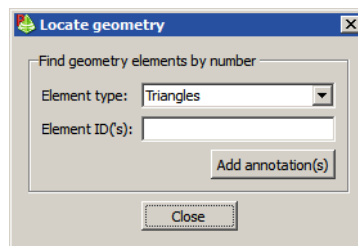


Figure 9-25: Find mesh elements

The *Find elements* tool (Figure 9-24 c) is used to identify mesh elements by their internal ID's. This is often used when kernel errors or warnings are given. An element type must be specified before entering the element ID in the dialog window (see Figure 9-25). Multiple ID's may be found at the same time by using a comma separated list. The 3D view will add a preview annotation to the specified elements. When the *Add annotation(s)* button is pressed, a permanent annotation is linked to that element(s), otherwise annotations will disappear once the dialog is closed or the selection is cleared.

Mesh connectivity (Figure 9-24 d) is a tool that indicates the free edges of a geometry by drawing a red line on the edge. In other words, mesh elements that do not connect to other elements are shown. Such elements typically appear at the edges of plates or at the ends of wires. When complicated or imported geometries are used, the mesh connectivity can show where the geometry does not line up correctly for an enclosed mesh. The mesh connectivity tool is a mode that can be either enabled or disabled on the ribbon.

Figure 9-26 shows the options that are available for mesh highlighting, if any is desired. The selected solution type will be highlighted in the 3D view by displaying a yellow grid over elements that are solved with that method. Solution types include: lossy metals (b), coatings (c), CFIE/M-FIE (d), EFIE (e), impedance sheets (f), PO (g), PO (Fock regions) (h), GO (i), UTD (j), FEM (k),

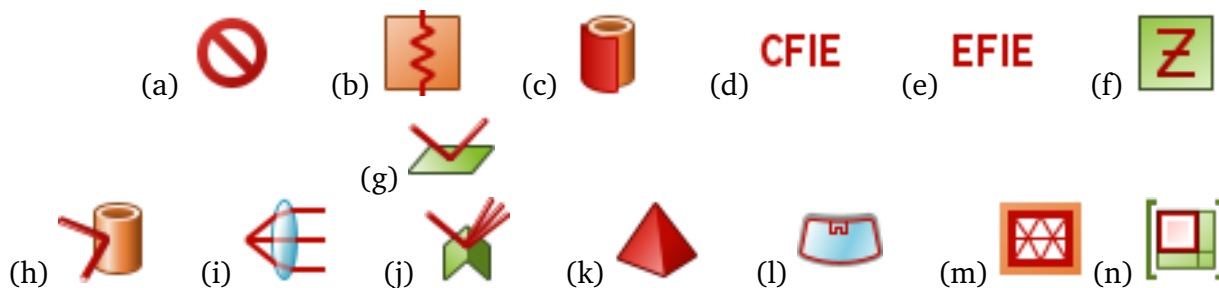


Figure 9-26: Mesh highlighting options

windscreen active elements (l), planar green's function aperture triangles (m) and numerical Green's function (n). The highlighting is activated by the associated button (Figure 9-24 e) which will then change and show the icon corresponding to the selected highlighting option. To deactivate, select the “none” (Figure 9-26 a) option.

The *Result* tab

The *Result* tab provides access to the visualisation setting of results. Determining which components of a result to display is done on the result palette; the *Result* tab only provides control over how that information is presented. To manage the display settings of a result, select it and note which buttons become active.

Display settings for far fields, near fields, currents and rays are available. In the cases of currents and near fields, arrows can be added to the view to help visualise the direction of the fields or currents. Rays will only be available if it was requested that they be stored prior to running the simulation.



Figure 9-27: Result management group

The *Manage* group Figure 9-27 shows the *Manage* group buttons on the *Result* tab. With these options, it is possible to make a duplicate of the selected result with all of its visualisation settings (Figure 9-27 a). It is also possible to store a local copy of a result. This result will be stored in the POSTFEKO session and becomes independent from any future changes made to the model. Note that a copy of the stored result is displayed in the view as soon as the copy is made.

The *Rendering* group Figure 9-28 shows the *Rendering* group buttons on the *Result* tab. The rendering group applies to far fields, near fields, currents and error estimates. In most cases, the rendering options will be useful for near and far fields. The *Grid* button (a) enables visibility of a mesh grid on top of the result, which helps give a sense of dimension to 3D results. When the *Surface* (b) option is deselected, the coloured surface of a result is hidden. *Discrete* (c) colouring removes the interpolated colouring of a surface and uses a predefined set of colours to represent

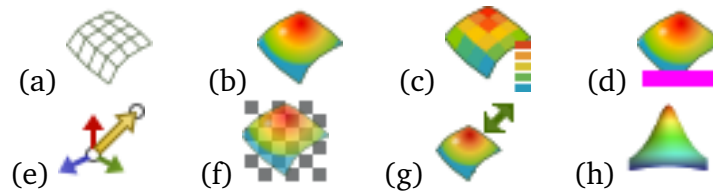


Figure 9-28: Result rendering group

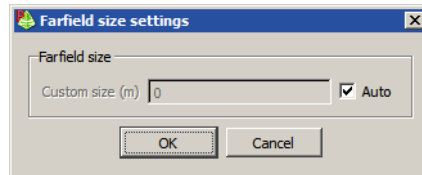


Figure 9-29: Advanced rendering dialog for far field size

the surface. The *Colour* option (d) applies to isometric surfaces (for 3D near fields). *Origin* (e) and *Size* (g) settings apply only to far fields. Note that the far field can be set manually in the advanced dialog shown in Figure 9-29. *Opacity* (Figure 9-28 f) sets the amount of transparency: For a setting of 0%, the element will be purely transparent and will not be visualised. At 100%, no transparency will be applied. *Extrusion* (Figure 9-28 h) applies to Cartesian near field surfaces that lie in a flat plane.



Figure 9-30: Result requests group

The Requests group Before a simulation is run, it is good practice to validate that data has been requested in the correct locations. Figure 9-30 shows the *Request points* (a–c), *Setting* (d) and *Boundary* (e) buttons which help ensure that near and far fields have been requested correctly. Request points will automatically be shown if no result data is present. Once data becomes available, the result data will be shown and the request points will be hidden. This behaviour corresponds to the *Auto request points display* (Figure 9-30 a). Request points can also be forced on or off by choosing *Display request points* (Figure 9-30 b) or *Don't display request points* (Figure 9-30 c), respectively. Note that the *Settings* button (b) is now enabled.

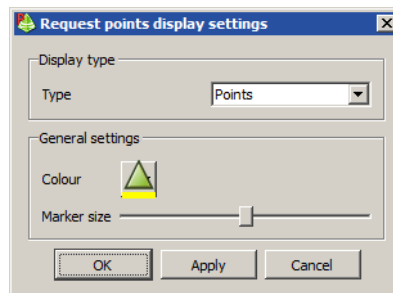


Figure 9-31: Request points display settings

The visualisation of the points can be set using the display settings dialog shown in Figure 9-31.

Here, the display type, marker colour and marker size for the requested points can be set. For near fields, the bounding box can also be shown by selecting the *Boundary* button (Figure 9-30 c).



Figure 9-32: Result contours group

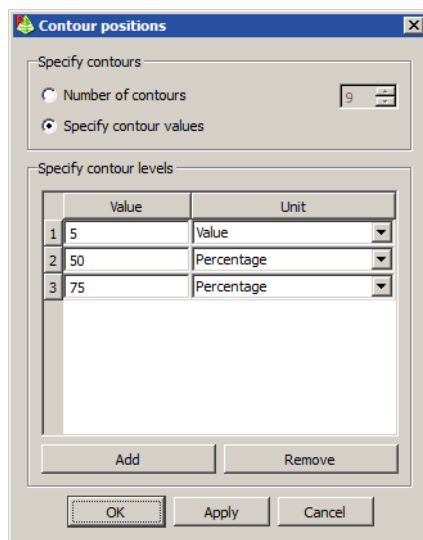


Figure 9-33: Manually specify contour positions

The Contours group Contour lines are lines that indicate a specific value on a result display. The *Show contours* button (Figure 9-32 a) toggles the visibility of the contour lines. The colour of the contours (Figure 9-32 c) can be set to any value, or the colour can be linked to the magnitude of the displayed value. Figure 9-33 shows the dialog used to specify the contour position accessed through the *Contour positions* button (Figure 9-32 b). An option is given to choose between the *Number of contours* and to *Specify the contour values*. The latter option allows for any number of contours that lie on user-specified positions. The positions can either be defined by its magnitude value, or by specifying a percentage of the value range.

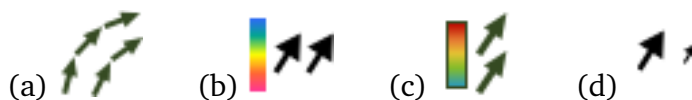


Figure 9-34: Result arrows group

The Arrows group Arrows indicate the direction in which a current flows or the direction in which a field is pointing. As these results change over time, arrows can only be plotted if the magnitude of such a result is shown for a specific phase value. Under *Quantity* in the result palette, *Instantaneous* should be selected and the phase should be specified. Once arrows are displayed by using the *Show arrows* button (Figure 9-34 a), the colour (b) and size (c, d) can be set with the remaining buttons. Note that the arrows are scaled by their magnitude by default.



Figure 9-35: Result rays group

The Rays group Different aspects of rays may be displayed independently from one another. *Ray lines* (Figure 9-35 a), *Ray numbers* (Figure 9-35 b), *Group numbers* (Figure 9-35 c) and *Intersections* (Figure 9-35 d) may be displayed in any combination by selecting the appropriate button. It is then also possible to colour the displayed ray lines by magnitude (Figure 9-35 f). The visibility threshold (Figure 9-35 e) can be set to only allow the higher magnitude rays to be displayed. This helps reduce clutter and only leave the most important rays in the view.

The Animate tab

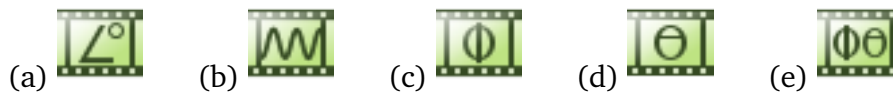


Figure 9-36: Animation types

Animation can be performed by varying a property over time. The properties that can be animated include: phase (Figure 9-36 a), frequency (Figure 9-36 b) and the camera angle (Figure 9-36 c–e). Phase and frequency animation requires a result to be present that varies over these parameters. Camera angle animation only requires a geometry to be present. The camera angle may be animated over only phi, only theta, or theta and phi simultaneously.

For frequency and phase animations, the results that will be affected must be added to the view and be made visible. For camera angle animations, only the geometry is required. Ensure that the correct animation type is chosen from the *Type* button on the *Animate* → *Settings* tab. By pushing the *Play* button, the animation will commence. The speed with which the animation variable changes can be set using the *Faster* and *Slower* buttons. Deselecting the *Legend* button will remove the description at the bottom of the screen.



Animation settings: The advanced animation settings allow for more precise definition regarding the speed and resolution with which a variable animates. Each property specifies how much the variable must change per second. For example, a *Phase* ($\omega t/s$) = 30.00° means that the phase will increment by 30° every second, which will produce a complete 360° loop in 12 seconds. The other properties work in much the same way.

Note that for continuous frequency models, it is necessary to break the continuous run into discrete steps. The combination of *Frequency (points/s)* and *Continuous frequency (# of points)* will then determine the sampling resolution and animation speed.



Export animation: Exporting an animation is also possible. Movies can be saved in *.avi, *.mov, or *.gif formats. 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. Setting the frame rate will affect how “smooth” the animation appears. Export works on the currently selected animation type and settings.

9.2.6 Using legends

A legend is a utility that explains what the data is that is being presented in a view. Legends can be placed on both 2D and 3D views. However, they function somewhat differently from one another, since the data which they represent are displayed differently.



Legend (2D): For 2D views, a legend is automatically added to the top of the graph (beneath the graph title). To change the location of a legend, click on the *Legend* button and select a better location from the list. It is also possible to remove the legend or to manually specify the location from here. The entries will reflect all displayed traces and their representations. Styling can be performed on the legend on the *Format* tab. The font settings, background colour, border line styling and drop shadow settings can be changed.



Trace text: Legend entries for each trace are automatically generated. This is generally sufficient for understanding what each trace represents. However, it is often desirable to manually specify the description of a trace. To do this, click on the *Trace text* button. The same option is available by right-clicking on a trace in the result palette. Figure 9-37 shows the dialog that can be used to modify the legend entry for a selected trace.

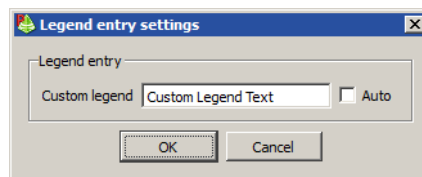


Figure 9-37: Chart text dialog

Manually setting the formatting and entries for the legend is only available for 2D legends. Specifying the legend location resizes the size of the graph to make room for the legend in that location. For example, a graph will be wider if the legend is placed on top than if it is placed to the right. When the overlay or manual positions are used, the graph is maximised and the legend should be placed such that it does not obscure important data.



Legend (3D): For 3D views, legends can be placed in any of the four corners of a view. Clicking on any of the four buttons provide a list of entities that the legend can link to. The title of the legend will give an indication of what data is being displayed and in what units. A colour key is also given to help interpret the different colours that a legend value can possess.

Setting the individual range of a 3D value legend is possible by clicking on the *Individual range* button. The value range is set to the minimum and maximum value of the displayed data by default. Here the range can be set manually for linear and logarithmic data. There are several settings that will affect the range limits. These can be accessed by clicking on the advanced dialog launcher in the corner of the *Legends* group. Figure 9-38 shows the dialog, followed by an explanation of the entries.

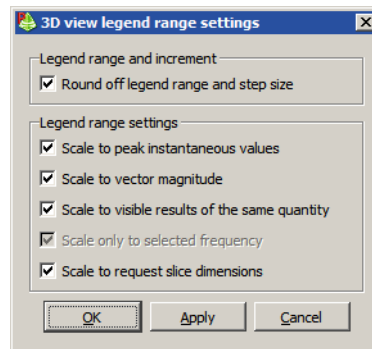


Figure 9-38: Advanced range limit settings

- *Round off legend range and step size*

Most data will have a large number of decimal values to accurately describe a given value. This results in a legend range that is difficult to read and interpret. To mitigate this, the values are rounded off and a range is selected that results in a legend that is more legible. The actual data range will always fall within the rounded off range.

- *Scale to peak instantaneous values*

This is used when the magnitude of a result along with the instantaneous phase is shown. The minimum and maximum value limits will remain constant for each phase step, meaning that the definitions for the colours are always the same. This makes it simpler to see how the magnitude changes over phase. When only the magnitudes at the given phase are of interest, this box can be deselected so that the range limits synchronise with the currently displayed data.

- *Scale to vector magnitude*

Often, especially with near field results, one wants to compare two components with each other. In these cases it is easier to compare relative magnitudes of the components if they are scaled to the same maximum. The total vector magnitude is used to scale all result components, so that a relatively small component is easily discernible from a larger one.

- *Scale to visible results of the same quantity*

It is possible to have several near field or far field requests in a model. Each request will have its own minimum and maximum value. However, POSTFEKO automatically scales the displayed data to keep all results that have been added into a view into account. This makes it clear what the values are relative to one another. Deselecting this option will render each result according to its own minimum and maximum value.

- *Scale only to selected frequency*

This option is only enabled when discrete frequency data is available in a model. Here the range limits are determined by the minimum and maximum values for all calculated frequency points. For example, seeing the change in gain for a far field over frequency is simpler when the field is scaled according to all calculated points.

- *Scale to request slice dimensions*

It is not always possible or desirable to plot all requested result data at the same time. If it is required that the range values are determined only by the displayed slice, this option

should be deselected. This is useful if only the range of data at a specific slicing is of interest.

All of the above-mentioned options are selected by default. Each setting is applied independently, meaning that a wide variety of combinations is available to help display the data in the desired manner.

9.2.7 Use the project browser

The project browser is an advanced tool that provides a project-wide, model-wide and detailed view of a project and its contents. It is divided into three sections, namely the project browser, the model browser and the details browser.



Project browser: The project browser can be accessed by enabling it on the *View* tab by the *Project* button. It can be hidden again in the same way, or by resizing the panel to zero width.

The *Project Browser* contains a tree of the models that have been loaded into the project. It also contains a tree for any stored and imported data that has been pulled into the project session. When selecting any of the contents of this window, the *Model Browser* becomes filled with data relevant to that entry. This is the only place where models may be removed from a project. Right-click on a model in the tree and select *Remove model* to remove all reference of it in the project.

When the *Model Browser* is populated, more detailed information about a model is given. It includes a summary of all solution configurations, as well as any data that could be combined over the configurations. Expanding the tree for a configuration shows further details on the media that was defined, the mesh, the solution entities that are present, result requests, infinite plane, etc. By selecting any of these components, the *Details Browser* is filled with further information that is relevant to that entry. If results have been calculated for the model, it is possible to add the data to the current view by right-clicking on the component and sending it to the active view.

The *Details Browser* summarises the information relevant to a select model component. For example, selecting a far field request in the *Model Browser* will give information about where the requested points are and what the request specifications are. Each entry is handled differently and shows a different set of information.

9.2.8 Manually specify axis properties, ranges and captions

On a 2D graph, it is often necessary to rename axes, titles or legend entries to better describe the data that is being presented. The axis ranges are set to automatically determine the data ranges and scale themselves accordingly. To manually specify these ranges is possible. Along with all the other styling options, a graph can be fully customised to professionally present data.



Chart text: The graph title, footer and axis labels are automatically derived from the contents of the graph. To change any of these entries, open the dialog by clicking on the *Chart text* button and deselect the *Auto* box. The customised entry will then be applied once *OK* or *Apply* is pressed. Figure 9-39 shows the dialog used to make these changes.

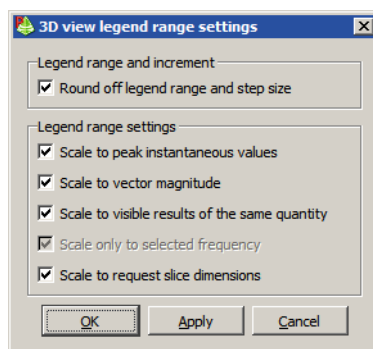


Figure 9-39: Chart text dialog



Axis settings: For polar and Cartesian graphs, the axis settings can be modified to have different ranges, grid spacing or numbering formats. To change any of these properties, ensure that the correct axis is being edited, deselect the box indicating that values will be determined automatically and enter the custom values. Note that the *zoom to extents* feature will now be applied to these ranges. Figure 9-40 shows the dialog for making these changes.

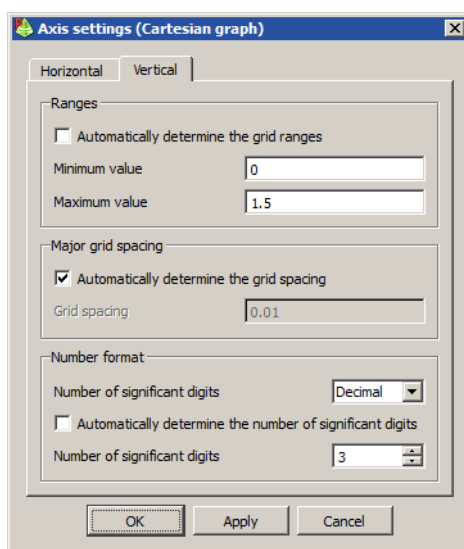


Figure 9-40: Axis settings dialog

For a 3D view, it is sometimes useful to clamp data between two values. This affects the colouring of the result, where blue will correspond to the minimum value and red to the maximum. This can help reveal changes in a result between two values that would otherwise be missed.



Individual range: Control is given over the result value ranges for both linear and logarithmic scaling (i.e. dB). For linear scaling, either the minimum and maximum values of the result can be used, or the range can be fixed to user-defined extents. For dB, the range limits are automatically defined by the minimum and maximum values of the data (*Automatic range limits*). The option is also available to use the fixed range values from the linear scale settings (*Use linear range limits (with auto clamping)*), although a dB-specific fixed range can also be specified (*Fixed range*). The range can also be specified by specifying the maximum dynamic range. Here the maximum value of the result data will be used and the minimum value will be the the maximum minus the value entered.

9.2.9 Using math traces

Any result plotted on a 2D graph provides the option of mathematically altering the trace by using a mathematical equation. However, it is also possible to create a math trace. These traces inherently contain no data and require other traces or mathematical equations to present information. The only difference between data traces using equations and math traces, is that math traces do not use the `self` keyword.

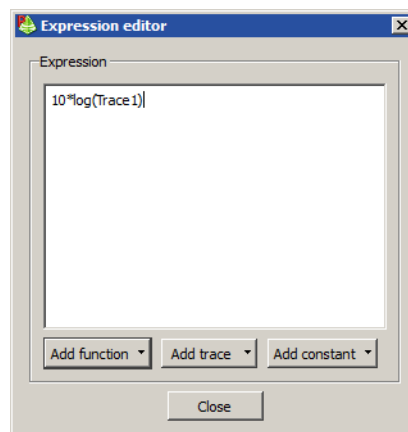


Figure 9-41: Expression editor for mathematical equations

Equations may be edited using the expression editor (Figure 9-41), which contains a list of all defined functions, traces and constants that may be used. Complex mathematics may be used in calculating results, but only scalar data can be displayed. If a trace cannot be displayed, then a warning icon should appear next to the trace in the results palette. Hovering over the icon will indicate the reason as to why the equation is not being shown.

9.2.10 Using math scripts

Scripting in POSTFEKO (see section 8.5.6) was introduced briefly in a different section. To summarise, a script is a mechanism with which an internal POSTFEKO result structure (or dataset) can be created or manipulated directly. In order to modify a script, the editor shown in Figure 9-42 is provided. Features such as syntax highlighting and code completion make editing scripts simpler and more efficient. Scripts are written in the Lua language.

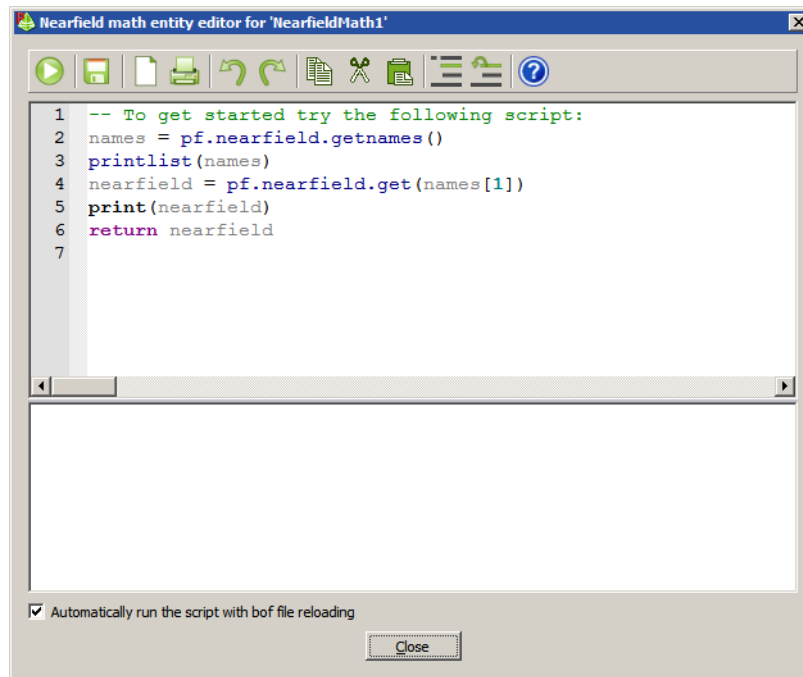


Figure 9-42: Editor for Lua scripts in POSTFEKO

Scripting allows for the manipulation and creation of data which can then be displayed in POSTFEKO as an internal data entity. For example, an existing near field can be pulled into a script, modified and returned. POSTFEKO can then use this data as though it was generated by the kernel. An alternative example is that a theoretical or measured pattern can be generated or imported into a script. This result can then be displayed and compared to simulated data.

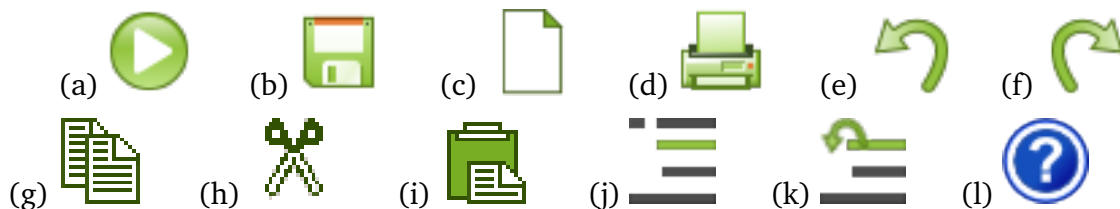


Figure 9-43: Editor buttons

The script will be executed when the *Run* button (Figure 9-43 a) is pressed. Saving a script (Figure 9-43 b) will keep the changes made to the script and also save the POSTFEKO session. The editor can be cleared using the *Clear all* button (Figure 9-43 c) or printed using the *Print* button (Figure 9-43 d). *Undo* and *Redo* is also available (Figure 9-43 e and f). The clipboard can be utilised through use of the *Copy*, *Cut* and *Paste* buttons (Figure 9-43 g, h and i). Block comments can also be managed using the *Comment* and *Uncomment* buttons (Figure 9-43 j and k). Additional help can be accessed using the *Help* button (Figure 9-43 l).

By default, a script will be re-evaluated once any of the result files in the session are reloaded. There are cases where this behaviour is not desired, such as when the script requires many system resources or is not dependent on the data being reloaded. If the automatic re-evaluation is disabled, the dialog must be opened and the script rerun manually for the results to update.

9.2.11 Generating reports

POSTFEKO is a useful tool to help analyse and present data in a useful format. Once this has been achieved, it is often desired 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. The data, images or animations can be exported, a *Quick report* can be generated, or a pre-defined template report can be populated.

Exporting and copying data, images and animations



Figure 9-44: Exporting and copying

Figure 9-44 shows all of the exporting and copying operations on data (Figure 9-44 a–b), images (Figure 9-44 c–e) and animations (Figure 9-44 f).

Quick reports

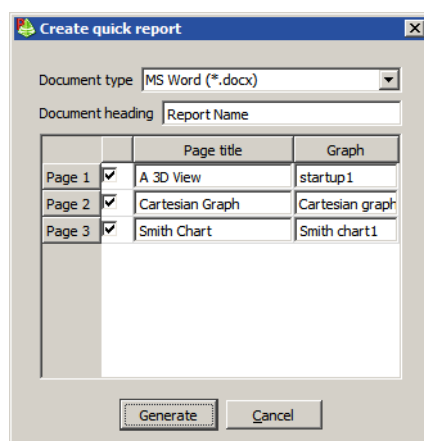


Figure 9-45: Generating quick reports



Quick report: Figure 9-45 shows the interface to the *Quick report* tool. A quick report is a report that contains all of the images and headers from a POSTFEKO session through a few quick clicks. Microsoft PowerPoint and Word documents may be generated, as well as a PDF (Portable Document Format) report. For the MS Office reports, MS Office 2003 or newer must be installed on the machine where the report is being generated.

These reports are a good starting point for a document and can be styled and extended externally as required. The PDF reports cannot be modified with most PDF viewers, but given the correct software can also be extended.

Generating a report from a template

When a MS PowerPoint or Word template is provided, images can be replaced by using the template reporting tool. This process involves three steps:

1. Create an MS PowerPoint or Word template document (i.e. a *.potx or *.dotx document).
2. Set up a new report using the *New from template* option.
3. Generate the report using the *Generate report* option.

The first step involves using MS PowerPoint or Word to create the template. Create a document in the appropriate application. Set up the document exactly as it should appear, with the correct styling, headers, images etc. POSTFEKO can add images to the document in places where place holders were inserted. To insert a place holder, insert a rectangle shape into the document. Add text to the document and identify them using tags in the following format:

`<feko:image:tag>`,

where tag can be replaced with any text string to help identify the place holder.



New from template: These tags can be read by POSTFEKO. The recognised tags will be listed and an opportunity will be presented to match each tag with one of the views currently open in the session (Figure 9-46). To start configuring a template report, click on the *New from template* option on the ribbon.

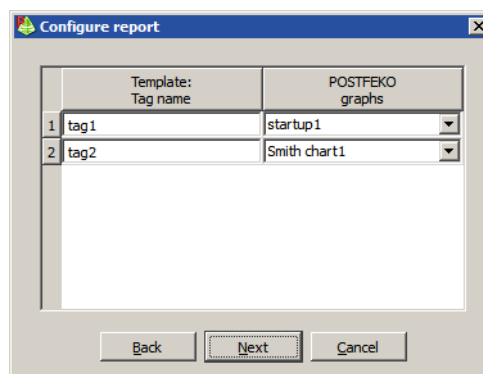


Figure 9-46: Configuring a template report.



Generate report: Once the report is configured, it will be listed in the *Project browser* under *Template reports*. Note that no report has been generated yet. The report has only been configured. To generate the report, click on *Generate report* and select the desired option from the menu. A prompt for the file location and name will appear, after which the report will be generated and saved to the chosen location.

For MS Office 2007 or newer, Word templates may also be constructed using “content controls”. This gives more control over the placement and formatting of the image. This is a more advanced feature and it will require activating the *Developer* tab in MS Word. Once the tab is active, add an image content control and set the *Tag* field to any value and POSTFEKO will treat it accordingly.

9.3 Errors and warnings

Errors and warnings are presented in a variety of ways depending on the source and severity of the notification. Notifications can be triggered by:

- changes made to model (*.fek) or result (*.bof) files from outside POSTFEKO;
- results and traces that are added to a view but do not have the necessary data for visualization;
- or a user action that either needs to be confirmed or is invalid.

In each case, there is a different mechanism for indicating the error/warning message. The mechanisms include icons next to results that are in error, a dialog indicating that a workflow error in POSTFEKO has occurred and then there is also a dialog that appears when changes are made externally to the model or result files.

9.3.1 Warning icon

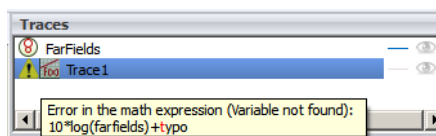


Figure 9-47: A warning icon on an erroneous result component.

The warning icon in Figure 9-47 shows that the far field cannot be displayed because of a missing results file. The tooltip appears when the mouse hovers over the icon and gives an indication as to why the result is missing. This type of notification occurs mainly when results cannot be plotted due to missing or corrupt data, or when an invalid mathematical equation is used.

9.3.2 POSTFEKO workflow error and warning dialog

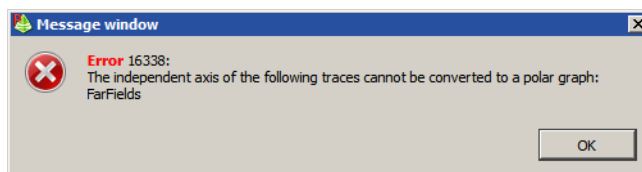


Figure 9-48: Error dialog for attempted invalid operations.

The error dialog in Figure 9-48 shows the second mechanism of informing the user of a problem. This type of dialog is commonly found when invalid actions are attempted. In the example shown, it was attempted to plot far field data on a Smith chart, which is not a valid operation.

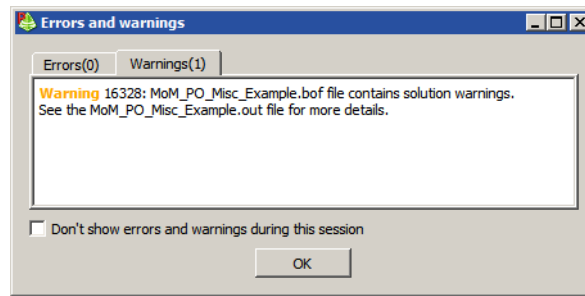


Figure 9-49: Dialog for logging background errors.

9.3.3 Error and warning logging dialog

This dialog appears when files are modified externally such that notifications are required. For example, if a simulation is run and warnings or errors have occurred, then the window in the dialog in Figure 9-49 is appended with the appropriate message. Errors and warnings are separated into two groups via a tab, so that they can be summarised separately. The same mechanism will be used, for example, when the result (*.bof) files contain errors or warnings or are out of synch with the model (*.fek) files.

When the *OK* button is clicked, the windows will be cleared. The next time an notification is logged, the dialog will appear again in a fresh dialog. By checking the box next to *Don't show errors and warnings during this session* and clicking the *OK* button, the window will be cleared and no more warnings or errors will be shown for the remainder of that project.

9.4 Getting Help

Additional information or explanations can be found in a variety of ways in the FEKO suite. A help browser can be launched from within the POSTFEKO application, digital copies of the user manual are included in the installation, examples are available on the FEKO website and a team of technical support staff can be contacted for more advanced help with specific problems or queries.

Figure 9-50 shows the home page of the Help browser. The browser contains information on how the FEKO Suite works as well as links to the digital copies of the manuals (in a *.pdf format), the FEKO website and contact details to get in touch with customer support.

The Help browser can be launched by clicking on the help button in the top right-hand corner of the POSTFEKO screen...



...or by clicking on the help button in the corner of the start page, or by pressing F1. Pressing F1 while a dialog is in focus will also jump to the relevant section in the Help browser.



Digital copies of all help documentation can be accessed from the start page, as well as links to the FEKO website (<http://www.feko.info>).

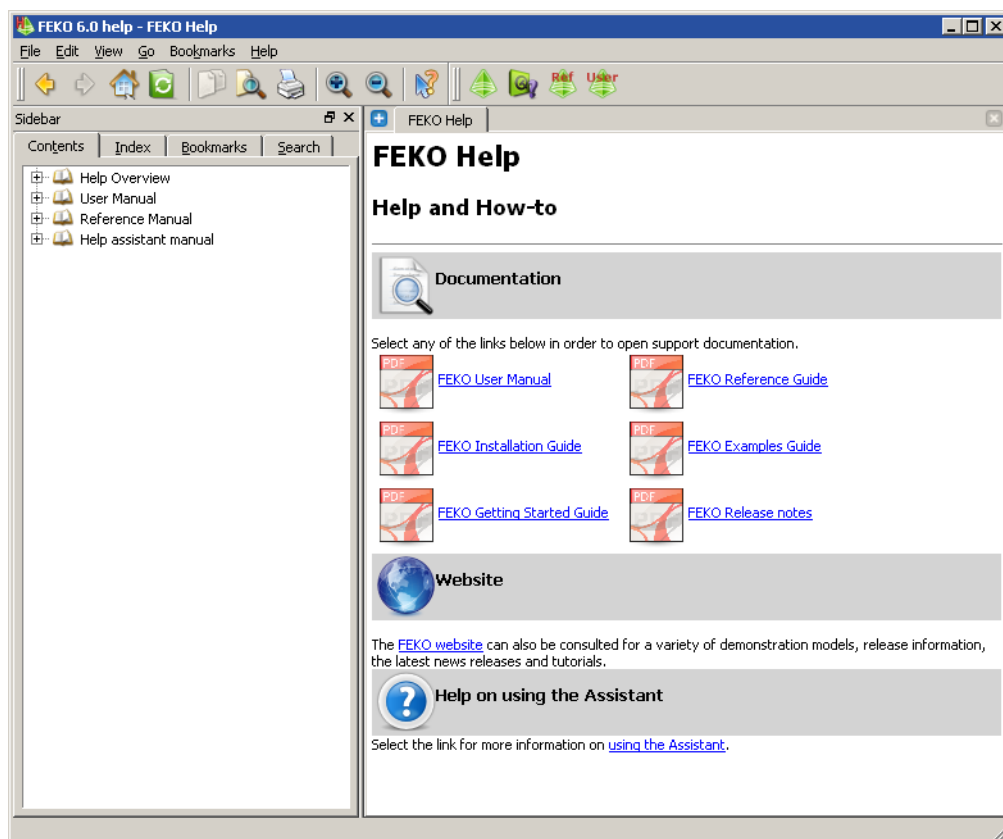


Figure 9-50: Home page of the Help browser.

10 Scripting guide for POSTFEKO

It is the intent of this chapter to provide a comprehensive guide to managing POSTFEKO results within a Lua environment. For a complete guide on the Lua language, we refer you to the official Lua Reference Manual (<http://www.lua.org/manual/5.1/>) and Programming introduction (<http://www.lua.org/pil/>). Another useful resource is the community maintained Wiki that is available at <http://lua-users.org/wiki/>.

The rest of the chapter deals with two principles:

- How to pull data into the scripting environment and how to return the results to a POSTFEKO session. The Lua language is used as a base with which to generate data that can be displayed in POSTFEKO, or to modify data that already exists in the session.
- How to use the language extensions that are made available within the Lua environment. Once a handle on a result has been obtained, it must be possible to modify that data. The tools that are available to analyse and modify results will be discussed.

10.1 The interface between POSTFEKO and the scripting environment

There are several keywords (other than the standard Lua keywords) that are recognised by the editor. These are:

Table 10-1: Global keywords recognised in the Lua editor.

<code>complex</code>	An object class that helps manage complex numbers.
<code>dataset</code>	A namespace that provides a set of tools for managing POSTFEKO data structures in the Lua environment. A <code>dataset</code> contains all of the information that is necessary for POSTFEKO to display and interpret information correctly.
<code>inspect</code>	A general function that displays the contents of any table that can be broken down into basic components. Output can be seen in the output window of the editor.
<code>pf</code>	The main interface between the Lua scripting environment and POSTFEKO. The <code>pf</code> namespace is used to pull results from a POSTFEKO session into the Lua environment for further processing.
<code>point</code>	An object class that helps manage points in 3D space.
<code>printlist</code>	A general function that prints out the contents of key/value pairs. Output can be seen in the output window of the editor.
<code>i, j</code>	Predefined constants for $\sqrt{-1}$

In addition, the `print` function is overloaded in the POSTFEKO editor environment. If a POSTFEKO dataset is given to the `print` function, a short summary of the structure of the dataset is displayed in the output window.

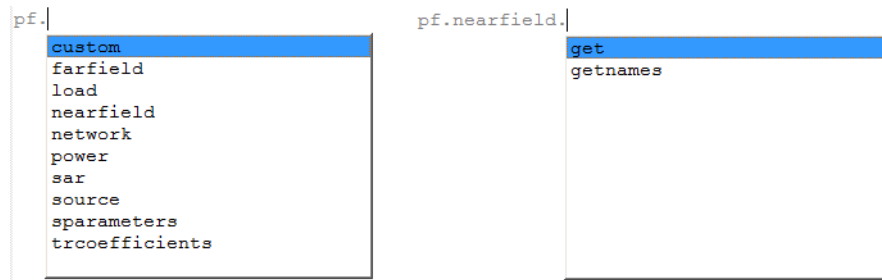


Figure 10-1: Auto-complete expansion of the pf namespace.

It may be useful to note that the auto-complete functionality of the editor will expand many of the keywords to reveal more operations. Figure 10-1 shows the expansion for the pf namespace, revealing a list of sub-categories to explore. Some of these may also be expanded, as seen in the second panel of the figure. The bottom level of the expansion is a function call (e.g. `pf.nearfield.getnames()`).

10.2 Pulling data into the scripting environment

A common use for the scripting functionality is to modify existing POSTFEKO results. As such, it is necessary to get a handle on a result (called a dataset) into the scripting environment. The following code will print a list of all near field results in a session. The output is also indicated.

```
nf_names = pf.nearfield.getnames()
printlist(nf_names)

1: 'Horn.Configuration1.NearField1'
2: 'startup.Configuration1.NearFields'
```

To get a handle on either of the near fields in the session, the key string must be used in conjunction with the get function. The key string is in the format:

```
"[Model].[Configuration].[Request Name]"
```

The following line of code will make the near field from the *Horn* model available for processing in the scripting environment under the variable name `nf`.

```
nf = pf.nearfield.get("Horn.Configuration1.NearField1")
```

This process can be applied to any result type. The result is a dataset, which will be explained in the section pertaining to dataset structures (see section 10.3).

10.3 The structure of a dataset

In order to modify or generate a dataset, it is important to understand how they are constructed. The structure of a dataset contains three types of data:

- axes refer to the independent axes along which data can be plotted in a view.
- quantities refer to the types of data that can be plotted along a specified axis.

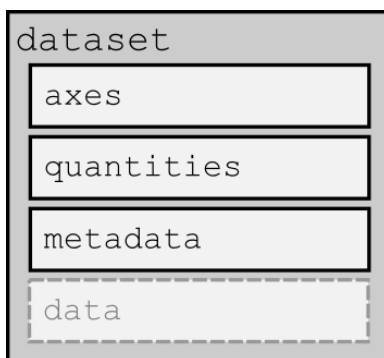


Figure 10-2: Visualisation of the dataset.

- `metadata` refers to additional data that pertains to a dataset that will affect how POSTFEKO interprets the dataset.

The additional *data* field refers to the actual values of the results. The indexing of the data is dependent on the structure of the axes and quantities that were defined. Since the *data* fields should not be accessed directly, it is not discussed here. Rather, the discussion of the proper access methods will follow (see section 10.4).

Table 10-2: Supported units.

Distance	m
	inch
	feet
	mile
	mil
Frequency	Hz
Time	s
	min
	hr
Mass	g
	t
Temperature	C
	K
	F
Current	A
Charge	C
Force	N
Potential Difference	V
Resistance	Ohm
Conductivity	S
Power	W
Capacitance	F
Inductance	H

Table 10-2 shows all of the base units currently supported by POSTFEKO. For all axes and quantities, the units need to be specified to help ensure that POSTFEKO knows how to manage the values. More complex units may be constructed out of the base unit from the table by using the operators “/” and “^”. For example, acceleration can be written as “m/s^2”.

Data can also consist of different types of data. Table 10-3 indicates the supported data types.

Table 10-3: Supported data types.

scalar	Any real valued data.
complex	Complex values; values containing real and imaginary components.
boolean	Either a true or false value.

10.3.1 Axes

An axis is a dimension along which data can be calculated. They typically contain the set of physical points where data is calculated, the frequency at which data is calculated, etc. A full list is given in Table 10-4. Any of these axes may be added to a dataset.

Table 10-4: Axis types for the axes field.

Position axes	x
	y
	z
	theta
	phi
	rho
	r
Frequency axis	frequency
Other axes	arbitrary
	index
	solution
	undefined
	time

As an example, we pull in a near field from a horn antenna and print out the different axes. Note that the near field specifies three spatial dimensions and a frequency dimension.

```
nf = pf.nearfield.get("Horn.Configuration1.NearField1")
for index, axis in pairs(nf.axes) do
    print("axis["..index.."]:",axis)
end

axis[1]: "frequency" axis with a single value at 1.645E+009Hz
axis[2]: "x" axis spanning from -0.26m to 0.26m with 21 values
axis[3]: "y" axis spanning from -0.2m to 0.2m with 21 values
axis[4]: "z" axis with a single value at 0.46m
```

Analysing the example, one can see that the near field was calculated at a single frequency and at a single height. The other two spatial dimensions form a rectangle, making it clear that the

near field is a flat surface at a fixed height and frequency. Each axis also indicates the unit in which the axis is measured. The spatial axes are measured in metre (m), while the frequency axis is measured in Hertz (Hz). All of the supported units are given in Table 10-2. Note also that the data type for all of the axes are scalar. In other words, all of the axis values in this instance contain real values.

10.3.2 Quantities

A quantity is a value that is calculated at each point for each axis in a dataset. More than one quantity can be calculated for any position on the axes; each one typically representing a different type or component of a result. For example, the E_x , E_y and E_z components of an electric near field would be stored as three separate complex quantities. All of these quantities are valid at the same physical position and frequency. Looking again at the near field example of a horn antenna, the structure of quantities are illustrated.

```
nf = pf.nearfield.get("Horn.Configuration1.NearField1")
inspect(nf.quantities)
```

```
[hfieldcomp1] [+]
    [unit] A/m
    [dataType] complex
[hfieldcomp2] [+]
    [unit] A/m
    [dataType] complex
[hfieldcomp3] [+]
    [unit] A/m
    [dataType] complex
[efieldcomp1] [+]
    [unit] V/m
    [dataType] complex
[efieldcomp3] [+]
    [unit] V/m
    [dataType] complex
[efieldcomp2] [+]
    [unit] V/m
    [dataType] complex
```

The quantities contained in this near field are the three components of both the electric and magnetic field. These values are specified at every dimension for the specified axes values. In order to specify a near field component, complex values must be used. The units are specified as $\frac{A}{m}$ for the electric near field components and $\frac{V}{m}$ for the magnetic components. If we were to inspect the entire dataset, a single data point would then contain six values.

```
[hfieldcomp1] -0.0016298448547029 + 0.0030584305657125i
[hfieldcomp2] 0.00019556510683489 + 5.7238782893743e-005i
[hfieldcomp3] 0.0023804214721832 - 0.00073565845419827i
[efieldcomp1] 0.33955688994892 - 0.15945928741613i
[efieldcomp2] -0.1306987855743 - 0.50674748783847i
[efieldcomp3] 0.10921303015844 - 0.13966547870835i
```

A dataset may contain any quantity. However, if the dataset is to be used as an internal type, the minimum quantities required for POSTFEKO to interpret the data correctly must be present. For instance, a near field must contain either a complete set of electric field components or a complete set of magnetic field components in order to be valid. This restriction only applies to a dataset that will be used as though it is a near field calculated by the FEKO kernel.

10.3.3 Metadata

Sometimes it is necessary to include additional data that is not relevant to a specific axis or quantity. This information generally tells POSTFEKO to treat the values in a slightly different way. For example, the x -, y - and z -axis is relative to a local coordinate system. The metadata fields can be used to specify the local origin and axis rotations so that the axes values are interpreted correctly.

10.4 Processing and modifying a dataset

A typical use for the scripting functionality is to modify existing results. The steps that will be followed in this case would typically be:

1. Create and simulate a model so that results are available.
2. Create a math script:
 - (a) Pull a single result or multiple results into the script.
 - (b) Perform processing on the results in the scripting environment.
 - (c) Store the results of the script in a dataset that is accessible in the POSTFEKO session.
3. Display and process the results.

These steps assume that a valid dataset is pulled in and being manipulated. Therefore, it is not necessary to create or modify any of the axes, quantities or metadata fields. The data stored in the dataset must simply be accessed and modified. However, the *data* block in a dataset cannot be accessed directly (see Figure 10-2). Instead, an indexing scheme is available with which to reach an element. Since individually indexing each element can become cumbersome, a library function is made available that iterates through all of the elements.

10.4.1 Indexing a single element in a dataset

This type of scheme is used when fine control is required when accessing a dataset. It is also the most intuitive method of accessing the elements. Each axis in a dataset contains a set number of values. By specifying the index of each value on an axis, the values can be accessed.

```
nearField[1][1][1][1].efieldcomp1 = nearField[1][1][1][1].efieldcomp1 * 2
```

The previous command will multiply the value of the E_x at the first frequency, at the first indexed point in space. By iterating through all of the axes, it is possible to selectively modify specific values based on the index of the axes. In the following example, all fields more than a 0.15m from the z -axis is set to 0.

```
-- Create the near field dataset
nf = pf.nearfield.get("Horn.Configuration1.NearField1")
-- Iterate through all of the axes and modify the field values
axis1 = nf.axes[1]; axis2 = nf.axes[2]; axis3 = nf.axes[3]; axis4 = nf.axes[4]
for freq = 1, #axis1 do
  for x = 1, #axis2 do
    for y = 1, #axis3 do
      for z = 1, #axis4 do
        if ( math.sqrt(axis2[x]^2 + axis3[y]^2) >= 0.15 ) then
          nf[freq][x][y][z].efieldcomp1 = 0 + 0*i
          nf[freq][x][y][z].efieldcomp2 = 0 + 0*i
          nf[freq][x][y][z].efieldcomp3 = 0 + 0*i
          nf[freq][x][y][z].hfieldcomp1 = 0 + 0*i
          nf[freq][x][y][z].hfieldcomp2 = 0 + 0*i
          nf[freq][x][y][z].hfieldcomp3 = 0 + 0*i
        end
      end
    end
  end
end
return nf
```

10.4.2 Iterating through all elements in a dataset

For the individual element style of indexing, it is necessary to manually iterate over each element. Expressions can become difficult to work with, so an alternative is provided. The library function `dataset.forallvalues` provides a simpler method for iterating through a dataset.

```
nf1 = pf.nearfield.get("Horn.Configuration1.NearField1")
nf2 = pf.nearfield.get("modNF_indiv")
nfAverage = pf.nearfield.get("modNF_indiv")

function average(index, target, source1, source2)
  target[index].efieldcomp1 = 0.5*(source1[index].efieldcomp1 + source2[index].efieldcomp1)
  target[index].efieldcomp2 = 0.5*(source1[index].efieldcomp2 + source2[index].efieldcomp2)
  target[index].efieldcomp3 = 0.5*(source1[index].efieldcomp3 + source2[index].efieldcomp3)
  target[index].hfieldcomp1 = 0.5*(source1[index].hfieldcomp1 + source2[index].hfieldcomp1)
  target[index].hfieldcomp2 = 0.5*(source1[index].hfieldcomp2 + source2[index].hfieldcomp2)
  target[index].hfieldcomp3 = 0.5*(source1[index].hfieldcomp3 + source2[index].hfieldcomp3)
end
dataset.forallvalues(average, nfAverage, nf1, nf2)
return nfAverage
```

Note that in this example, it was never necessary to define a loop. Instead, a function is written that explains what should happen to a given element. Any number of dataset sources can be provided as source inputs. The result of the calculation is then stored in the target dataset.

10.5 Creating a custom dataset

Another use for scripting is for when custom data objects are required. In these cases, the user would typically follow these steps:

1. Create a POSTFEKO session.
2. Create a math script. Note that the math script must not be of any specific type, otherwise POSTFEKO will attempt to validate the axes to make sure that the expected dataset structure is returned.
 - (a) Create a new dataset.

- (b) Add all of the desired axes and quantities fields. In this case any quantity can be defined, since POSTFEKO will not attempt to interpret them in any particular way. axes are still confined to the pre-defined values, where the arbitrary axis can be used for any unspecified axis type.
- (c) Perform processing on the results in the scripting environment.
- (d) Store the results of the script in a dataset that is accessible in the POSTFEKO session.

3. Display and process the results.

Consider the following example. Here the script looks at an existing near field. When the magnitude of the field at any point exceeds $5 \frac{V}{m}$, the custom dataset stores that value. For all other values it is zero.

```

nf1 = pf.nearfield.get("Horn.Configuration1.NearField1")
-- Create custom dataset
custom = dataset()
axisF = dataset.axis("frequency", "Hz", 1.645e9)
axisX = dataset.axis("x", "m", -0.26, 0.26, 21)
axisY = dataset.axis("y", "m", -0.2, 0.2, 21)
axisZ = dataset.axis("z", "m", 0.46)
dataset.addaxis(custom, axisF)
dataset.addaxis(custom, axisX)
dataset.addaxis(custom, axisY)
dataset.addaxis(custom, axisZ)
dataset.addquantity(custom, "above5V", "scalar", "V/m")

-- Populate the values
function process5Vthreshold(index, target, source1)
    local magEx = source1[index].efieldcomp1:magnitude()
    local magEy = source1[index].efieldcomp2:magnitude()
    local magEz = source1[index].efieldcomp3:magnitude()
    local totalE = math.sqrt(magEx^2 + magEy^2 + magEz^2)
    if (totalE >= 5) then
        target[index].above5V = totalE
    else
        target[index].above5V = 0
    end
end
dataset.forallvalues(process5Vthreshold, custom, nf1)

return custom

```

Here the dataset called *custom* has the same spatial axes of the source near field. An arbitrarily defined quantity was defined. For each position on every axis, a scalar value must be defined for the dataset to be valid. The dataset was created entirely within the script. The axes and quantities were added to the dataset and the data was populated. The result can also be plotted on a 3D or 2D view, similar to any internal data type.

10.6 Structures for internal dataset types

When creating a script that returns a pre-defined result type, it is necessary to add certain minimum fields so that POSTFEKO can process it correctly. For each type, the minimum values will be provided in the following sections.

10.6.1 Near field

Near field results must contain either a complete set of electric field components, a complete set of magnetic field components, or a complete set of both in order to be valid. For each different coordinate system, a different set of spatial axes are required. Table 10-5 will summarise the required fields.

Table 10-5: Required properties for near fields.

Property name	Property type	Unit	Description
frequency	axes	Hz	Every near field requires a valid frequency axis.
axis 1 axis 2 axis 3	axes		Every near field requires three independent spatial axes. Depending on the coordinate system, these axes may vary. Table 10-6 will give a breakdown of the required axes for each system.
efieldcomp1 efieldcomp2 efieldcomp3	quantities	V/m	Required quantity set for electric fields. All three complex values must be defined for a complete electric field definition.
hfieldcomp1 hfieldcomp2 hfieldcomp3	quantities	A/m	Required quantity set for magnetic fields. All three complex values must be defined for a complete magnetic field definition.
origin	metadata		The local origin for the workplane around which the far field is defined.
uvector	metadata		A point relative to the origin which indicates in which direction the $\hat{u}$ -vector.
vvector	metadata		A point relative to the origin which indicates in which direction the $\hat{v}$ -vector.
conical	metadata		A boolean flag indicating whether the near field is defined in a conical coordinate system. The value may be either true or false.
rho stepsize	metadata		A scalar value indicating the the increment with which the cone's rho axis increases. This effectively controls the angle of the cone for a conical system. Note that this field is only required if Conical is true.

The conical coordinate system is a special case, since it is technically not a complete coordinate system. As such, some additional information is required. For near fields defined in the conical coordinate system, the following must also be provided under the metadata fields:

1. `conical = true`. This flag indicates to POSTFEKO that the dataset fields must be interpreted differently than other coordinate systems.
2. `rho stepsize`. The step size indicates the intervals with which the rho axis grows. Note that the rho axis only contains a single value which corresponds to the starting point of the cone.

Table 10-6: Coordinate system axes.

<i>Coordinate system</i>	<i>axis 1</i>	<i>axis 2</i>	<i>axis 3</i>
Cartesian	x	y	z
Spherical	theta	phi	r
Cylindrical (x-axis)	rho	phi	x
Cylindrical (y-axis)	rho	phi	y
Cylindrical	rho	phi	z
Conical	rho	phi	z

10.6.2 Far field

Table 10-7: Required properties for far fields.

Property name	Property type	Unit	Description
frequency	axes	Hz	Every far field requires a valid frequency axis.
theta	axes	deg	Elevation angle component relative to the local workplane. The vertical position sits at $\theta = 0^\circ$.
phi	axes	deg	Azimuthal angle component relative to the local workplane.
incidenttheta	axes	deg	The theta direction from where an incident plane wave originates.
incidentphi	axes	deg	The phi direction from where an incident plane wave originates.
theta	quantities	V	This is a complex value indicating the theta component of the electric field in this theta direction, or E_θ .
phi	quantities	V	This is a complex value indicating the phi component of the electric field in this phi direction, or E_ϕ .
directivityfactor			
	quantities		A scaling factor that will scale the magnitude of a field value to the expected directivity in a given direction. The formula that is used is given below in Eq. 10-1.
gainfactor	quantities		A scaling factor that will scale the magnitude of a field value to the expected gain in a given direction. The formula that is used is given below in Eq. 10-2.
realisedgainfactor			
	quantities		A scaling factor that will scale the magnitude of a field value to the expected realised gain in a given direction. The formula that is used is given below in Eq. 10-3.
rsc	quantities		A scaling factor that will scale the magnitude of a field value to the expected radar cross section for a given observation direction.
origin	metadata		The local origin for the workplane around which the far field is defined.
uvector	metadata		A point relative to the origin which indicates in which direction the $\hat{u}$ -vector.
vvector	metadata		A point relative to the origin which indicates in which direction the $\hat{v}$ -vector.

Directivity

The formula used to calculate directivity is given. The directivity is a figure of merit indicating in which direction the most energy will be radiated. Directivity is the power density radiated in any direction versus an isotropic radiator which is radiating the same amount of energy.

$$D = \text{directivityfactor} \times [\text{Re}(E_\theta)^2 + \text{Im}(E_\theta)^2 + \text{Re}(E_\phi)^2 + \text{Im}(E_\phi)^2] \quad (10-1)$$

Gain

The formula used to calculate gain is given. Gain is calculated in the same way as directivity, except that the input power is used rather than the radiated power. In other words, system losses are taken into account.

$$G = \text{gainfactor} \times [\text{Re}(E_\theta)^2 + \text{Im}(E_\theta)^2 + \text{Re}(E_\phi)^2 + \text{Im}(E_\phi)^2] \quad (10-2)$$

Realised Gain

The formula used to calculate realised gain is given. Realised gain is calculated in the same way as gain, except that the power which is reflected back to the input port is taken into account. In other words, system losses and mismatch effects are included.

$$RG = \text{realisedgainfactor} \times [\text{Re}(E_\theta)^2 + \text{Im}(E_\theta)^2 + \text{Re}(E_\phi)^2 + \text{Im}(E_\phi)^2] \quad (10-3)$$

10.6.3 Source

Table 10-8: Required properties for sources.

<i>Property name</i>	<i>Property type</i>	<i>Unit</i>	<i>Description</i>
frequency	axes	Hz	Every source requires a valid frequency axis.
current	quantities	A	Currents are complex values that are measured in Ampere.
admittance	quantities	S	The reciprocal of impedance and is measured in Siemens and is a complex value.
power	quantities	W	The rate at which energy is expended. This is equal to the current times the voltage and is measured in Watts, which is a complex value.
type	quantities		A value must be given to help indicate what type of source the dataset represents. A value of 0 represents a voltage source, where a value of 1 represents a current source.
impedance	quantities	Ohm	The total opposition to current flow and the ratio of the voltage to the current. It is measured using the complex value Ohm.
mismatchloss	quantities		The fraction of power that is reflected to or transmitted to other ports; i.e. the amount of power that does not enter the system. This is a scalar value with no unit.
voltage	quantities	V	The potential difference over the port which is a complex value measured in Volts.

10.6.4 Load

Table 10-9: Required properties for loads.

<i>Property name</i>	<i>Property type</i>	<i>Unit</i>	<i>Description</i>
frequency	axes	Hz	Every loads dataset requires a valid frequency axis.
current	quantities	A	Currents are complex values that are measured in Ampere.
impedance	quantities	Ohm	The total opposition to current flow and the ratio of the voltage to the current. It is measured using the complex value Ohm.
power	quantities	W	The rate at which energy is expended. This is equal to the current times the voltage and is measured in Watts, which is a complex value.
voltage	quantities	V	The potential difference over the port which is a complex value measured in Volts.

10.6.5 Network

Table 10-10: Required properties for networks.

<i>Property name</i>	<i>Property type</i>	<i>Unit</i>	<i>Description</i>
frequency	axes	Hz	Every networks dataset requires a valid frequency axis.
arbitrary	axes		This is an axis that lists the port numbers for the network.
current	quantities	A	Currents are complex values that are measured in Ampere.
impedance	quantities	Ohm	The total opposition to current flow and the ratio of the voltage to the current. It is measured using the complex value Ohm.
power	quantities	W	The rate at which energy is expended. This is equal to the current times the voltage and is measured in Watts, which is a complex value.
voltage	quantities	V	The potential difference over the port which is a complex value measured in Volts.

10.6.6 S-Parameter

Table 10-11: Required properties for scattering matrices (s-parameters).

<i>Property name</i>	<i>Property type</i>	<i>Unit</i>	<i>Description</i>
frequency	axes	Hz	Every loads dataset requires a valid frequency axis.
arbitrary	axes		The string values here represent the scattering matrix element definitions in the form $S_{x,y}$, which can be read as “the ratio between the voltage wave measured at port x as a result of the voltage wave inserted at port y .”
portflag	quantities		A value indicating whether the port is active or passive (or both). If the value is 0, then the port is treated as a passive port. If the value is 1, then the port is considered to be adding and absorbing energy from the model.
load	quantities	Ohm	A complex component that is added to a model that has the property of being able to dissipate energy.
sparameter	quantities		Scattering parameters are complex values that relate the voltage wave measured at a port vs. the voltage wave that is inserted into port.

10.6.7 Power

Table 10-12: Required properties for power.

<i>Property name</i>	<i>Property type</i>	<i>Unit</i>	<i>Description</i>
frequency	axes	Hz	Every power dataset requires a valid frequency axis.
activepower	quantities	W	The amount of power that is being inserted into the system. This power could be radiated or absorbed by other ports.
efficiency	quantities		A percentage value indicating the relationship between the active power and the power provided to the system.
powerloss	quantities	W	The amount of power that is dissipated in the system. This could refer to any power lost due to lossy materials, etc.

10.6.8 Transmission and reflection coefficients

Table 10-13: Required properties for Transmission and Reflection coefficients.

<i>Property name</i>	<i>Property type</i>	<i>Unit</i>	<i>Description</i>
frequency	axes	Hz	Every transmission / reflection coefficient dataset requires a valid frequency axis.
CoPolarisedReflectionCoefficient			
	quantities		A complex value giving the ratio between the reflected wave vs. the incident wave. The co-polarised component is given and is defined in the direction of the polarisation angle of the incident plane wave.
CoPolarisedTransmissionCoefficient			
	quantities		A complex value giving the ratio between the transmitted wave vs. the incident wave. The co-polarised component is given and is defined in the direction of the polarisation angle of the incident plane wave.
CrossPolarisedReflectionCoefficient			
	quantities		A complex value giving the ratio between the reflected wave vs. the incident wave. The cross-polarised component is given and is defined in the direction of the polarisation angle of the incident plane wave.
CrossPolarisedTransmissionCoefficient			
	quantities		A complex value giving the ratio between the transmitted wave vs. the incident wave. The cross-polarised component is given and is defined in the direction of the polarisation angle of the incident plane wave.

10.6.9 Specific Absorption Rates (SAR)

Table 10-14: Required properties for Specific Absorption Rates (SAR).

Property name	Property type	Unit	Description
frequency	axes	Hz	Every specific absorption rate dataset requires a valid frequency axis.
hasAverageSAROverTotalVolume			
	quantities		This is a boolean value (i.e. true or false). If the flag is true, then averageSAROverTotalVolume must be specified.
averageSAROverTotalVolume			
	quantities	W/kg	Average SAR over the total volume of the model. This quantity is a scalar value.
hasAverageSAROverRequestedVolume			
	quantities		This is a boolean value (i.e. true or false). If the flag is true, then averageSAROverRequestedVolume must be specified.
averageSAROverRequestedVolume			
	quantities	W/kg	Average SAR over a specifically requested sub-volume of the model. This quantity is a scalar value.
hasPeakSAR			
	quantities		This is a boolean value (i.e. true or false). If the flag is true, then massOfPeakSARCube, airFractionOfPeakSARCube and peakSARInCube must be specified.
massOfPeakSARCube			
	quantities	kg	The mass of the SAR cube in kilograms. The value is a scalar and is typically 0.01 kg or 0.001 kg.
airFractionOfPeakSARCube			
	quantities		Air fraction in percent for this specific cube. The value is a scalar.
peakSARInCube			
	quantities	W/kg	Peak SAR in this sepcific cube. The value is a scalar.

Part IV

Working with EDITFEKO

11 Introduction to EDITFEKO

The editor EDITFEKO was designed to simplify the process of generating and editing *.pre (PREFEKO input) files. Though the *.pre file is a standard ASCII text file that may be created using any available text editor, EDITFEKO is a basic text editor with customised functionality.

Note that EDITFEKO was also used as an editor for the optimisation definition in a text *.opt file before the release of FEKO Suite 5.3. This function is no longer supported by EDITFEKO and all optimisation-related options should be defined in the optimisation tools provided in the CADFEKO interface (see section 7).

In this chapter, the general usage of the EDITFEKO tool, and the functions provided to aid in the creation and editing of specific cards in a *.pre file are discussed. More information regarding the usage of symbolic variables, advanced flow control using FOR loops and IF statements, adding termination conditions and printing information to the screen is discussed in the chapter dealing with PREFEKO (see section 17).

The model geometry and desired calculations are entered on separate lines of text in the *.pre file, referred to as cards. (For models built in CADFEKO the complete geometry is imported using one of these cards.) Each card may have a number of parameters which must appear in fixed positions. The arrangement of the cards is also important and controls the order of the steps that will be taken during simulation. (Some information regarding the card-order is found in the chapters describing the geometry cards (see section 13) and control cards (see section 14). More detailed information regarding the general structure of the *.pre file may be found in the chapter on advanced modelling and solution control (see section 12.2)).

The EDITFEKO interface consists of an editor area (see Figure 11-1) where more than one file may be opened — each in its own edit window. The toolbars at the top provide quick access to some standard functions. The status bar on the bottom of the window contains licence information, indicates when “Overwrite” is on, the line and column numbers at the current cursor position and also gives an indication of the file modification status. In the standard PREFEKO mode the button panel provides quick access to the card editor dialogs. When editing a TIMEFEKO *.tim file, EDITFEKO currently provides only a plain text editor with no special tools or functionality.

The program may be started from a command prompt by entering

```
editfeko example.pre
```

where the optional parameter (here example.pre) specifies the file to be opened. If the parameter does not contain an extension, the extension pre is added. Also, if such a file does not exist, it will be created. EDITFEKO can also be opened from inside CADFEKO and POSTFEKO.

If a file is opened in EDITFEKO a lock-file is created (with the name .filename.ext.lock) in the same directory as the file. If you open the file in another instance of EDITFEKO at the same time, it will give a warning (“A lock file already exists for this file. ... Are you sure you want to continue with this operation?”) with options to cancel or continue. Generally one should then click *Cancel* and switch to the other instance of EDITFEKO. One should only continue here if the lock file resulted after EDITFEKO was not closed properly (such as a power failure or system crash).

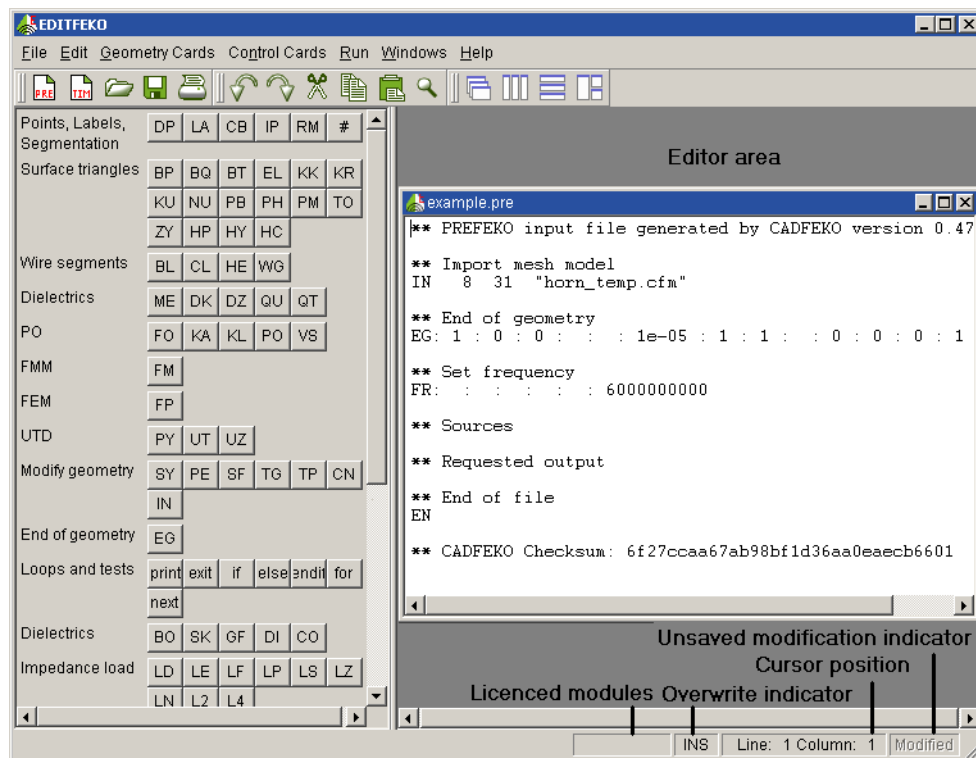


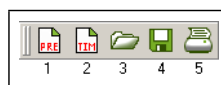
Figure 11-1: The EDITFEKO interface

11.1 Toolbars

The toolbars in EDITFEKO are usually docked at the top of the EDITFEKO window, but they can be dragged free or docked on any of the four sides. They can also be switched off by selecting *File* → *Preferences* from the main menu and selecting the *Appearance* tab.

Note that the splitter between the editor and the dialog panel may be dragged to resize the panel.

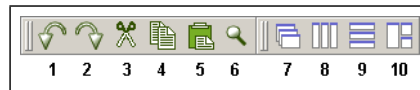
11.1.1 File control toolbar



Used for file and print control. The functions of the buttons are:

1. Create a new *.pre file and switch to PREFEKO mode.
2. Create a new *.tim file.
3. Open a file (and switch to the appropriate mode).
4. Save the current file.
5. Print current file.

11.1.2 The edit and window toolbars



The edit toolbar provides standard edit commands while the window toolbar allows positioning the edit windows in the editor area. The functions of the buttons are:

1. Undo. Multiple undo commands are supported.
2. Redo. When new text is added (by typing or adding a card) the redo chain is cleared and the new text forms the last entry in the undo chain.
3. Cut.
4. Copy.
5. Paste.
6. Search for text.
7. Cascade all edit windows.
8. Tile windows vertically.
9. Tile windows horizontally.
10. Arrange windows as best as possible.

11.2 Menu items in EDITFEKO

11.2.1 File menu

The *File* menu allows opening and closing files and is similar to standard Windows applications. Most of the options and the list of recently used files do not need any explanation.

The *New* sub menu item opens a sub menu where the user may choose *.pre for PREFEKO or *.tim for TIMEFEKO files. EDITFEKO will enter the appropriate mode depending on the selected option. The *Save* menu will ask for a file name if the file does not have one, otherwise it will save (without confirmation) using the current file name. Various short-cut keys are presented next to the respective menu items.

The file menu is also used to set user preferences (see section [11.3](#)).

11.2.2 Edit menu

This menu provides general editing commands. Note that multiple level undo/redo is allowed. Each change to the editor is recorded in the undo list. When the user clicks *Undo* the entry is moved to the redo list and the change to the editor is undone. When new text is added (by typing or adding a card) after a sequence of undo commands, the redo chain is cleared and the new text is recorded at the end of the undo list at that time.

It is possible to search the editor for specific text, case sensitive or not. The search always starts from the current cursor position. If it is desired to start the search from the top of the file, the cursor must be moved there (using <Ctrl><Home>). Note that there is also a fast card search. By right clicking on any button on the button panel, the next occurrence (starting from the current cursor position) of the associated card is found. This search is always downwards, and does not wrap back to the top of the file.

Note the *Goto line* option which is very useful to find lines reported by PREFEKO. (PREFEKO always reports the line number where errors occur.)

The *Comment* and *Uncomment* item allow the user to comment out blocks of text or remove the commenting indicators. This is often useful when debugging large files. The comments are inserted as ****** followed by a space and only removed if this sequence is present at the beginning of the line.

For example, in the lines

```
** ** Select the label
** LA    1
*** Next comment
```

the comment characters will not be removed from the third line using this function. It was also not generated by the *Comment* function. These commands are also available using the short-cut keys <Alt><C> (for *Comment*) and <Alt><U> (for *Uncomment*) or by right clicking in the editor and selecting from the resulting pop-up menu.

11.2.3 Accessing Geometry Control card definition panels

The *Geometry cards* and *Control cards* lists in the menu launch the card editor dialog panels of the various cards when editing a *.pre file. The relevant panel may also be launched by clicking on the appropriate buttons in the button panel. The cards in both the menu lists and on the button panel are grouped by functionality.

The card representing the settings defined in the card definition panel will be added in the line that the cursor was located when the panel was launched.

Pressing <F1> while the cursor is on a line containing a card definition will launch the relevant card-definition panel for the card defined on that line.

11.2.4 Run menu

This menu is used to launch the various FEKO components. If another GUI component is started from the *Run* menu, the relevant component of the currently active project is automatically opened in the launched UI component. Note, however, that CADFEKO does not automatically create a new model if a *.pre file with that name already exists. Hence if one runs CADFEKO from EDITFEKO on a model for which only a *.pre file exists, CADFEKO will open with a new nameless model. If this model is saved to the same name, it will overwrite the existing *.pre file. Hence one should generally create models in CADFEKO and then run EDITFEKO. Note that when running PREFEKO or FEKO, the current file is *saved without confirmation*.

Select *Run* → *Component parameters* to set the command line options for the various kernel components. See section [6.16.1](#) for more information.

It is possible to launch the solution on a remote machine. If the *Remote FEKO execution* (see section 18.2.3) item is checked, EDITFEKO launches the solution on the host specified in the *Remote execution* group of the *FEKO* tab of the *Component parameters* dialog.

If *Parallel FEKO execution* is checked, the solution is launched on a parallel cluster as defined in the *Parallel execution* group of the *FEKO* tab of the *Component parameters* dialog. Unless the parallel job is also a remote one, the local machine must be the first host name in the configuration. (See also section 18.2.2.)

If *Farm out by OPTFEKO* is checked, parallel optimisation are enabled. It is also possible to check both the *Parallel FEKO execution* and the *Farm out by OPTFEKO*. The combination of the two will result in farming during optimisations using parallel FEKO runs.

11.2.5 Window menu

This menu presents the standard Windows functionality to arrange or select the editor windows on the main editor area.

11.2.6 Help menu

In addition to providing access to information regarding the version of EDITFEKO (*About EDITFEKO*), this menu contains shortcuts to the EDITFEKO section of the FEKO help (*Help* → *FEKO help*) and the *FEKO User's Manual* and *Getting Started Manual* documents. These manuals are in PDF format and a PDF viewer must be configured on the *Preferences* dialog (see section 11.3).

Contextual help can also be accessed by pressing <F1> while EDITFEKO is open.

The *Check for updates* item opens the *Check for updates* dialog. This polls the EMSS web-site and informs the user if any updates have been added since the last check. (Since EDITFEKO does not know if the user also has licences for other platforms, it gives an update message regardless of which platform the update is intended for.) The update checker downloads one file containing the list of updates. It does not automatically download anything else and does not send any information to the web-site. If *Activate automatic update checking* is checked, EDITFEKO polls the web-site each time it is started more than seven days after last checking the web-site. (Note that this is off by default.) It is also possible to configure a proxy server if this is required to obtain access to the internet. Click the *Check* button to poll the web-site. The *Info* tab applies to the update checker, not the available update.

11.3 Preferences in EDITFEKO

Selecting *File* → *Preferences* opens the *Configure EDITFEKO* dialog. Various options are grouped under separate tabs as follows.

11.3.1 Appearance tab

Here the user can

- Switch off the top toolbar.
- Set the printer font.
- Set the size of the screen fonts.
- Set the preferred card format.

In the column based card format, the fields are located at fixed positions and cannot be longer than 5 (string and integer fields) or 10 (real fields) characters. In the colon separated format, the different fields in each card are separated with `:`-characters. This format allows arbitrary length entries. If EDITFEKO is using column based format, it will automatically switch to colon separated format if any of the input fields are longer than the allowed length.

11.3.2 Options tab

Here one may

- Elect to show superuser options if this is allowed by the licence. The superuser mode in FEKO is only used during program development and is not available to the general user. If one tries to edit an existing card containing superuser parameters, EDITFEKO will also prompt the user and switch to superuser mode if this is allowed. If the licence does not allow superuser mode, superuser options will be removed.
- Turn on auto-save and/or backups and specify the auto-save interval.
- Specify the PDF reader required for the manuals.

11.4 Inserting and editing input cards and variables

11.4.1 Generating input cards

Cards can be entered and edited directly in the editor window. In most cases the user would call one of the card editor dialogs by clicking on the appropriate button on the button panel or selecting from the *Geometry cards* and *Control cards* menus.

The buttons are grouped according to functionality namely

- Definition of node points, labels, segmentation parameters and variables (# button).
- Cards to generate surfaces which will be meshed into triangles elements.
- Cards generating wire segments.
- Cards for generating dielectric / magnetic volumes to be subdivided into cuboids, or specification of dielectric regions.
- Cards used in connection with the PO formulation.
- Cards to define the MLFMM and the FEM methods and parameters.

- Cards used in connection with the UTD formulation.
- Modifications to the geometry, such as translation, scaling and mirroring as well as the IN card which allows inclusion of an external file - typically a secondary *.pre file or geometry from an external file.
- The EG card signifying the end of the geometry. Generally the buttons above it should be used before the EG card and those below it should be after the EG card. Exceptions are the DP and IN cards and the FOR-NEXT and IF-ELSE blocks.
- Cards dealing with finite medium properties and finite grounds.
- Cards dealing with loading of wire segments.
- General solution cards, including Non radiating networks, transmission lines, and solution method parameters.
- Cards to specify the excitation type, frequency and power of the excitation.
- Cards used to control which data is written to the output files.
- Cards to specify where to calculate the fields, and other solution requests.
- The EN card which signifies the end of the input file.

A typical card editor dialog is shown below.

When a card editor dialog is selected from either the button panel or the menu, there is a button to *Add card*. This adds the card without closing this dialog. This is useful when defining a number of similar cards, for example, when specifying the first group of DP cards at the start of a model. Often only one or two parameters differ between cards. Clicking *OK* adds the card and closes the editor dialog. Pressing <Enter> has the same effect as clicking on the *OK* button. Similarly <Esc> closes the card editor dialog without adding the card as if the *Cancel* button has been clicked.

In the card editor dialogs, moving the mouse pointer over the labels may give more information of the specific input field. If the user is uncertain about the meaning or units of a given field, it is always advisable to move the mouse pointer to the label to determine if any additional information is available.

All the input fields in FEKO have a fixed anticipated length. In the card editor dialogs it is possible to enter longer strings than anticipated for the particular field. If longer strings or values that the maximum anticipated number of characters are entered, EDITFEKO will revert to the colon-delimited format (see section 11.3) in which there is no field-length limitation.

Editing an existing line

The editor dialogs can also be used to edit existing lines. The line at the current cursor position may be edited either by pressing <F1> or by right clicking at any position in the editor and selecting *Edit line at cursor* from the pop-up menu. Note that right clicking does not change the cursor position. When the selection (or highlighted part) in the editor spans more than one line, the line containing the cursor is edited. For multi-line cards the cursor can be anywhere in the card.

EDITFEKO processes each card as if it does not contain errors. For some of the fields where the user may select a number of options, the editor dialog's display will default to a "common" value if the input field is invalid. *Thus it is advisable to close the card editor with OK even when it was used just to check the validity of the line.*

If the user types, for example, PS and then presses <F1>, EDITFEKO may treat it either as a new card (with default options), or as an existing card with all options blank (zero). Setting all options to zero does not always make sense and EDITFEKO thus treats this case as a new card and uses the normal defaults. If the user wants to have a card with all options zero, he should type the card name followed by a space before pressing <F1>.

When editing existing lines, the *Add card* button is not available as the card is edited and written back to the editor.

Adding a comment label to an existing card

A comment or label field (see section 13.1) may be added to the end of the line representing a card after a '*' comment indicator. Label comments are automatically appended to all relevant cards in *.pre files that are generated by CADFEKO according to the labels in the CADFEKO model. The labels assigned to the EDITFEKO control cards are important in that they are used to uniquely identify the excitations and solution requests referred to in an optimisation setup (see section 7.1.4). In POSTFEKO the labels of solution request cards are also used in the legend text when plotting results from that card.

11.4.2 Variable editor

The # button on the top right corner of the button panel launches the card editor to edit variables. This is useful as it presents a list of functions and operations understood by FEKO and calculates the value of the variable as it would be evaluated by PREFEKO at this point.

The functions, variables or operations may be selected from the three drop-down lists and a special group for the "FILEREAD" function. The selected item is automatically placed at the current cursor position. Note that variables and functions are highlighted after insertion, but operations are not. If a function is selected while some part of the variable string is selected, the selected text will be inserted as the argument of the new function. Variables, operators and the "FILEREAD" function replace the currently selected text.

11.5 Important keystrokes

The arrow keys as well as <Pg Up> and <Pg Dn> behave in the normal fashion. The following keys may be different to some other applications. A block may be selected using the mouse,

Table 11-1: EDITFEKO cursor movements

<i>Key combination</i>	<i>Description</i>
<Ctrl><Left Arrow>	Move a word left
<Ctrl><Right Arrow>	Move a word right
<Ctrl><Page Up>	Move to top of visible page
<Ctrl><Page Down>	Move to bottom of visible page
<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

or pressing <Shift> and using the normal movement keys. If a block is selected, it will be overwritten when a key is pressed. The following list of short-cut keys are often used:

Table 11-2: EDITFEKO short-cut keys

<i>Short-cut key</i>	<i>Description</i>
<Ctrl><C> or <Ctrl><Ins>	Copy to clipboard
<Ctrl><X> or <Shift><Del>	Cut (delete) to clipboard
<Ctrl><V> or <Shift><Ins>	Paste (insert) from clipboard
<Ctrl><S>	Save
<Ctrl><A>	Save all files
<F1>	Edit line
<Alt><C>	Comment line(s)
<Alt><U>	Uncomment line(s)
<Ctrl><F>	Find
<F3>	Find next
<Ctrl><R>	Find and replace
<Alt><2>	Run PREFEKO
<Alt><3>	Run POSTFEKO
<Alt><4>	Run FEKO
<Alt><6>	Run OPTFEKO
<Alt><7>	Run TIMEFEKO

12 PREFEKO language and concepts

12.1 PREFEKO

PREFEKO is the FEKO Suite pre-processor. It is responsible for the creation of the `*.fek` input file used by the solution kernel. The input file for PREFEKO is the `(*.pre)` file that is edited using EDITFEKO (`*.pre` files can be edited in any text editor). For more information about launching PREFEKO from a command line see chapter 17.

This chapter presents the language and concepts required to create `*.pre` input files. The different input cards (geometry and solution control) are discussed in the section dealing with EDITFEKO geometry cards (see section 13) and section dealing with EDITFEKO control cards (see section 14) respectively.

12.2 Structure of the `*.pre` input file

The `*.pre` file is a text file that contains a list of high level commands. These commands are translated by the FEKO preprocessor (PREFEKO) to generate the input for the FEKO solver. The `*.pre` file commands are referred to as cards, and each card performs a specific function.

12.2.1 Order of the cards in the `*.pre` file

The cards can be categorised into geometry cards, control cards (which define the field parameters to calculate) and execution flow cards. PREFEKO processes these cards to generate the `*.fek` file.

When a model is saved in CADFEKO, CADFEKO writes a `*.pre` file that includes a card that indicates that the meshed CADFEKO model (stored in a `*.cfm` file) should be imported. The user may modify this card if required to change the import process. For example, the card may be changed so that structures on a symmetry plane are not imported — the symmetry plane can then be defined and an additional import card used to import the rest of the geometry. Applying symmetry in this way may provide a more efficient simulation approach for certain advanced cases.

Users are advised to use EDITFEKO to add and edit cards in the `*.pre` file. This will ensure that each card is formatted correctly and that the required parameters are limited to the correct values.

12.2.2 Format of cards in the `*.pre` file

The card format is described here mainly for users would like to generate FEKO input files automatically. FEKO supports two different formats for cards - a colon-separated and a column-based format. These formats may be mixed in a single input file. After the last column of a card (irrespective of the format) a comment may be added after the comment indicator (`***`). For control cards that define solution requests, this comment is used as a label for that card. The label of a card is used by OPTFEKO to uniquely identify specific results, and is indicated in POSTFEKO

to aid the identification of the solutions and blocks of interest when post-processing simulation results.

Column based format The basic form of this card format is shown below. The upper numbers indicate the columns. The name field (shown as ‘xx’) in columns 1 and 2 specifies the type of the card. This is followed by five integer parameters I_1 to I_5 (for some cards, these input fields may also contain strings 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	REAL

Colon separated format This is a less restrictive input format than the column-based format, where the individual integer and real parameters are separated by a colon character. Unlike the column-based format, when using this format, integer and real input fields are not restricted to 5 or 10 characters respectively. In this card format, the card name is still located in columns 1 and 2. The name is followed immediately in column 3 by a colon. The rest of the card has no spacing limitations. The lines shown below demonstrate valid syntax in this format.

```
DP: S1 : : : : : #x : #y : #z
BP: S2 : S2 : S3 : S4
```

All input and output parameters of FEKO are in SI units (e.g. lengths are in metres, potential in volts, etc.) All angles are in degrees. FEKO includes various scaling options (see the SF, TG and IN cards) so that dimensions may be entered in different units and scaled to metres.

The principal structure of the input file is shown below:

```
**  Comments at the start of the input file
...  Cards that define the geometry ** Comments
EG  End of the geometry
...  Control cards that define excitation, special solution op-
      tions and indicate which quantities should be calculated **
      Card labels / Comments
EN  End of the input file
```

The description in the section dealing with EDITFEKO geometry cards (see section 13) gives an overview of the cards related to the creation of geometry with detailed descriptions of the individual cards (these cards include the IN card, which is used to control the import of external meshes or CADFEKO models).

The description in the section dealing with EDITFEKO control cards (see section 14) gives an overview and description of the control cards used to define materials, solution methods, excitations and calculation requests for the model.

12.3 Guidelines for creating geometry in EDITFEKO

12.3.1 Meshing guidelines regarding connectivity

As discussed in section 2.2.3, 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, users must take care to conform to these rules when combining CADFEKO models with EDITFEKO scripting (e.g. attaching an antenna modelled with geometry cards on an aircraft meshed in CADFEKO), when creating the geometry solely in EDITFEKO, or when working with imported meshes.

Note that cuboidal volume elements (used to model volume dielectrics — the DK card (see section 13.10), the DZ card (see section 13.12) and the QU card (see section 13.40)) do not need to be connected in this way.

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. Thus, meshing the same line with the same mesh size will always give the same number of exactly equal divisions. 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 Figure 12-1, 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.

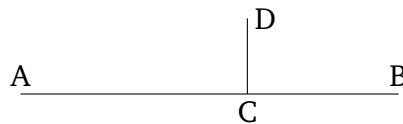


Figure 12-1: Example of a wire structure

Similarly, where two surfaces touch the common edge must be part of both surfaces. For example, the surface in Figure 12-2 should not be created as two rectangles ABFG and CDEF. If this is done, it is highly unlikely that there will be 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.

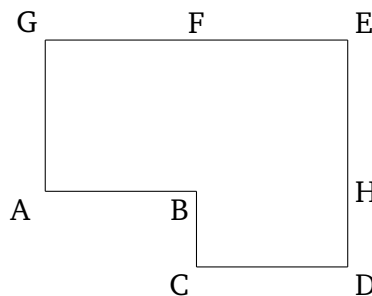


Figure 12-2: Example for a surface

A connection point between a segment and one or more triangles is only recognised when the beginning or the end of the segment lies on the vertex or vertices of the triangles. In Figure 12-3

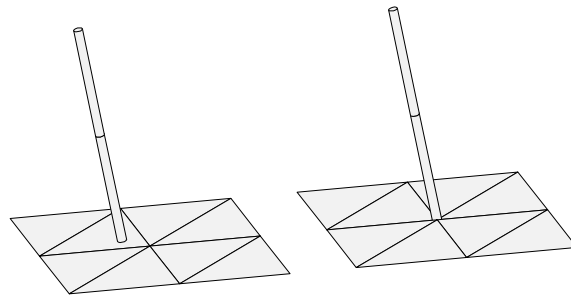


Figure 12-3: Incorrect (left) and correct (right) connection between a segment and triangles

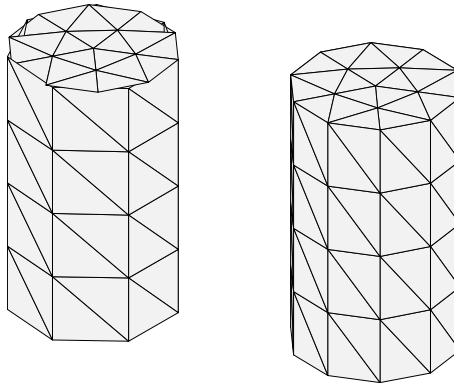


Figure 12-4: Incorrect (left) and correct (right) connection between curved edges

the left side is an incorrect and the right side a correct connection (here the segment is connected to six triangles).

When curved structures (circles, cylinders, spheres) 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. See the example in Figure 12-4.

12.3.2 Reducing GUI mesh sizes

For very large models (or at very high frequencies) it is possible that the user interface components cannot create and/or display the required mesh. To some extent, this problem can be overcome by creating a mesh of larger elements in CADFEKO and using the RM card in EDITFEKO to subdivide this further. The only restriction here is that 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.

12.4 Usage and concept of labels

In the scripting language, items are identified by their labels. These are then used to do operations or set electromagnetic properties on geometry. For instance in Figure 12-5 the label Feed can be used to define the feed segment (voltage source), or the label MainRef1 can be used to instruct FEKO to use physical optics for the main reflector.

Labels are set either directly in CADFEKO (see section 6.18.3), or by preceding geometry cards in EDITFEKO with the LA card (see section 13.29). When importing certain mesh formats by means of the IN card (see section 13.22) then labels can also be imported (for instance the NASTRAN property gets converted into a FEKO label). In principle, FEKO labels can be:

- A positive integer number (including zero).
- Any valid expression, like $3*\#i+2$, can be used. These expressions are evaluated, and the resultant numerical value is used as or 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 (see section 12.5)). Such variables at the end are evaluated and replaced by the corresponding numerical value (rounded to integer). Note that string labels are case insensitive in FEKO, i.e. labels Roof and ROOF are treated identically inside FEKO.

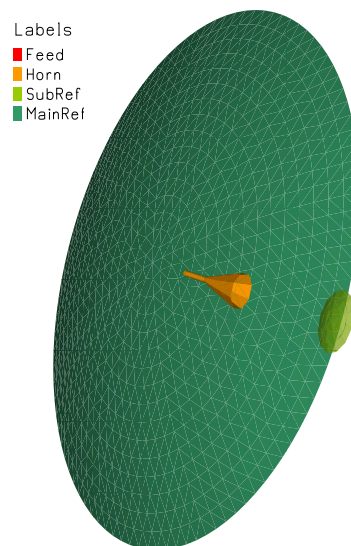


Figure 12-5: Example demonstrating the usage of labels (display of labels in colour with legend in POST-FEKO)

So for instance these labels are valid

```
23
5*#k+#j/2
LeftWing
Front_Door
Part#i
```

while these labels are not valid

```
Left+Wing    (invalid character '+')
-23          (negative integer)
Part_#i_#k   (two variables)
```

With the CB card, FEKO allows the user to change labels (e.g. after having imported some geometry). A powerful wild card globbing (expanding a non-specific label name containing a wildcard character into a set of specific labels) is supported (see section 13.6). At certain FEKO cards the user can specify label ranges and at other cards labels for created geometry can be derived from other labels (eg. when using symmetry - SY card), and thus it is important for users to understand

how the label algorithm works. The algorithm for this is that after having evaluated expressions or having replaced variables, a 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. See Table 12-1 for examples of labels are split into the base string and the associated number.

Table 12-1: Examples of splitting a label into its base string and associated number.

<i>Label name</i>	<i>Base string</i>	<i>Associated number</i>
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. See Table 12-2 for an example when incrementing a label by two.

Table 12-2: Label incrementing example (increment by two).

<i>Label name</i>	<i>Incremented label</i>
1	3
Roof	Roof
Part_17	Part_19

When using ranges of labels, then for the range specification to be valid the base strings must match, and the associated number must be in the correct range between the associated numbers of start and end label. So for instance the label Part_12 is in the range of the labels Part_10 ... Part_20, but the label Part_5 is outside this range. Using a range like Front ... Back is not valid (different base strings).

12.5 Symbolic variables

Instead of using numerical values in all the different cards, it is possible to use predefined variables. The name of a variable always consists of a #-sign followed by a string consisting of the characters a-z, A-Z, 0-9 and the special character _. The following are valid variable names: #height, #a, #STARTINGFREQUENCY, #a_1 or #P5_7f; while the following are not valid: #a?1 or #value2.1. There is no distinction between upper and lower case characters. For example, #a and #A are interpreted as the same variable.

Variables

It is important to note that in CADFEKO variables are used without the #-sign whereas PREFEKO requires the #-sign to distinguish between variables and functions (such as sin). When CADFEKO writes the variables to the *.cfm file, it prepends the #-sign 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 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 may be used when defining variables, so that direct calculations can be carried out. The variables have to be defined before they can be used in the respective cards. It is possible to use expressions like $2 * \#radius$ in the input fields subject to the maximum allowed length (10 characters for real values, 5 characters for integer values). For larger expressions additional variables have to be defined.

A variable is defined in the following way:

```
#2pi = 2*#pi
#vara = 1 + sqrt(2)
#varb = #vara * 2.3e-2 * (sin(#pi/6) + sin(rad(40))) + #vara^2)
#sum = #vara+#varb
```

Note that the # sign has to appear in the first column. Variables may also be defined from the command line as described in the previous section.

Arrays

In addition to plain variables, arrays with integer indices like $\#a[5]$ or, more complex,

```
#am_0[3*#i+ceil(#r[2])]
```

are also supported. The expression between the square brackets must evaluate to an integer number, which can also be negative. The implementation of using arrays is so that they do not need to be allocated, however they need to be initialised. So for instance after having used the instructions

```
!!for #i = 10 to 20
#array_a[#i] = 3*#i-10
#array_b[-#i] = 0
!!next
```

one will be able 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 $\#array\_b[0]$ will result in an error message that an undefined variable is used.

12.6 Built-in functions

Any predefined expression or variable as discussed above, can be used in conjunction with any of the following trigonometric-, Bessel-, miscellaneous-, coordinate- or data functions.

The following trigonometric functions as given in Table [12-3](#), are supported.

Table 12-3: Trigonometric functions

SIN	sine (argument in radians)
COS	cosine (argument in radians)
TAN	tangent (argument in radians)
COT	cotangent (argument in radians)
ARCSIN	arcsine (in radians)
ARCCOS	arccosine (in radians)
ARCTAN	arctangent (in radians)
ATAN2	this function has two arguments $\text{atan2}(\#y, \#x)$ it yields $\arctan(\#y/\#x)$ in the range $-\pi \dots \pi$
ARCCOT	arccotangent
SINH	hyperbolic sine
COSH	hyperbolic cosine
TANH	hyperbolic tangent

Table 12-4: Bessel functions

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

Table 12-5: Miscellaneous functions

SQRT	square root
LOG	logarithm to the base 10
LN	natural logarithm
EXP	exponential function
ABS	absolute value
DEG	convert radians into degrees
RAD	convert degrees into radians
STEP	step function, $\text{STEP}(x) = 0$ for $x \leq 0$ and $\text{STEP}(x) = 1$ 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 $\text{max}(\#a, \#b)$
MIN	returns the smallest of the two arguments — called as $\text{min}(\#a, \#b)$
FMOD	this function also has two arguments $\text{fmod}(\#a, \#b)$ 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 $\text{RANDOM}()$ 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 “ $\text{RANDOM}(-1)$ ” 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).

Table 12-6: Coordinate functions

X_COORD	This function returns the x coordinate of a point previously defined by a DP card, as discussed below.
Y_COORD	Returns the y coordinate of a previously defined point similar to the function X_COORD discussed below.
Z_COORD	Returns the z coordinate of a previously defined point similar to the function X_COORD discussed below.

The X_COORD, Y_COORD and Z_COORD, are used as follows: The name of the point, in quotation marks, is passed as an argument to the function, for example

```
DP    PNT01                1.234    0.4567    #z
#x = x_coord("PNT01")
```

will set the parameter #x equal to 1.234.

Data functions

The FILEREAD function reads a numerical value from an arbitrary ASCII file. The general syntax is

```
fileread("Filename", Line, Column)
```

and 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

```
#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
```

Logical & mathematical operators

The following mathematical operators can be used in conjunction with any predefined expression or variable:

()	brackets
+	addition
-	subtraction
*	multiplication
/	division
^	powers, for example 2^3=8

In addition to these functions, PREFEKO allows the use of logical operations. It supports the function NOT() — which returns TRUE if the argument is FALSE and FALSE when the argument is TRUE — and 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.

PREFEKO also supports a logical function DEFINED(#variable) which returns TRUE if a the variable #variable has been defined, and FALSE if not. This is useful in *.pre files used for OPTFEKO, TIMEFEKO or ADAPTFEKO runs. These 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 will be optimised with respect to the variable #a, one may use

```
!!if (not(defined(#a))) then
#a = 200.0e-3
!!endif
```

The order of precedence is (evaluated first in the order listed) is:

1. single number, expressions in brackets
2. function calls
3. + and - (when used as sign)
4. ^
5. * and /
6. + and -
7. >, <, >= and <=
8. = and <>
9. AND
10. OR

Some variables are predefined in PREFEKO, but may be overwritten by re-assignments. These are shown in Table 12-7.

Table 12-7: Pre-defined variable list.

Name	Value	Description
#pi	3.14159265358979...	The constant π
#eps0	$8.85418781761 \cdot 10^{-12}$	Dielectric constant ϵ_0 of free space
#mu0	$4 \pi \cdot 10^{-7}$	Permeability μ_0 of free space
#c0	$\frac{1}{\sqrt{\mu_0 \epsilon_0}}$	The speed of light in free space
#zf0	$\sqrt{\frac{\mu_0}{\epsilon_0}}$	The intrinsic impedance of free space
#true	1	Used for logical true
#false	0	Used for logical false

There are three other special variables #!x, #!y and #!z which are very 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
AO   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 (e.g. variable distance of the antenna in front of a reflector) an easy process, because only one variable needs to be changed. It also allows FOR loops and IF conditions.

12.7 FOR/NEXT loops

Some cards in FEKO implicitly use loops (such as when an FR card with multiple frequencies is used). This, however, does not always offer the flexibility which one may require, for example, to change the material parameters inside the loop. Another example would be the use of a loop to create a complex geometry.

For completely general loops, PREFEKO allows the construct

```

!!for #var = #start to #end step #delta

!!next

```

where a simple example would be


```

** Loop for the relative permittivity
!!for #eps_r = 1 to 5 step 0.5

** Set material parameters
GF    0                #eps_r    1

** Compute fields etc.
FE

** End of loop
!!next

```

The syntax requirements of FOR/NEXT loops are:

- 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, see the example below).
- 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 from 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.
- Several loops can be nested as shown in the example below.

A more complicated example, illustrating some of the points above, is as follows

```

#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
!!next

```

12.8 IF/ELSE/ENDIF constructs

This construct is used to allow different control cards under different conditions. The syntax requirements of IF/ELSE/ENDIF constructs are:

- The !! characters must be located in the first two columns of the line. This is followed by an arbitrary number of spaces, the expression to be evaluated and the keyword then (it is not case sensitive, THEN or Then are also accepted).

- 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` (again, 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.

Examples, illustrating some of the points above, are as follows

```
!!if #a > 5 then
...
!!endif
```

OR

```
#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
```

12.9 EXIT command

PREFEKO also supports the command `!!exit` to stop execution. This can be useful for checks like

```
!!if #a < 2*#b then
!! exit
!!endif
```

(see the various PRINT commands below to make this more informative).

12.10 PRINT commands

There are various commands to print strings (enclosed in double quotes) and floating point numbers to the screen and `*.out` file respectively:

- `!!print` will print some text to the screen.
- `!!print_warning` will also print some text to the screen, but will indicate that the given text is a warning, i.e. for console runs the string WARNING will be prepended, while for GUI runs the colour of the output will change according to a warning.

- `!!print_error` is similar to `!!print_warning`, but will handle an error message. Note that processing continues so that multi-line error messages can also be printed. To stop execution use the `!!exit` statement.
- `!!print_to_out` will write some text to the `*.out` file while the FEKO kernel is run.

These commands accept multiple arguments separated by commas. For example,

```
!!print "2*#b = ", 2*#b
!!if #a < 2*#b then
!!  print_error "The value of #a is too small:", #a, " (exiting now)"
!!  exit
!!endif
```

will print an error text and exit if the variable `#a` is smaller than two times variable `#b`. The line

```
!!print_to_out "This run was done with #b = ", #b
```

will print the value of `#b` to the `*.out` file at the position where it appears in the `*.pre` file.

12.11 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 particularly handy when a large number of points are required.

Single node names

When defining or using node names, simple variable names of the form `A#i` can be used to define the node. 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. The length of the node name string (before and after expansion) is still limited to 5 characters.

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
```

and use these either individually or inside another loop.

Node name arrays

Symbolic node name arrays can also be defined. These node arrays are particularly useful when creating polygonal surfaces (PY card and PM card) with many points since only the node name has to be specified instead of each individual point. Node name arrays allow expression of the form `A[2*#i+3]` to be used to index the array.

A symbolic node name array can be defined in a loop

```
!!for #k = 1 to 20
DP   P[2*#k+3]
!!next
```

and then later referenced only as `P`. Single node names can be referenced by indexing the array.

13 Geometry cards

The following table lists all input cards that are used to create the geometry (i.e. the cards that appear before the EG card in the *.pre file⁵). Most of these cards are processed by PREFEKO. For example, PREFEKO processes the BP card and writes the triangle elements to the *.fek file as input to FEKO.

Card	Type of Excitation
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
DD	
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
FM	set parameters related to the MLFMM
FO	defines a Fock region
FP	set 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
HY	creates a hyperboloid section
IN	reads an external <i>include</i> 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 segments, triangles, polygonal plates, etc.
ME	defines the medium
MB	defines a modal port boundary condition
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

⁵In general all the geometry cards must appear before the EG card. Exceptions are the IN card (when including *.pre files with control commands) and the DP and TP cards (when defining points for the AP card).

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	remeshing and adaptive mesh refinement
SF	enters a scaling factor, with which all dimensions are multiplied
SY	utilises symmetry in the construction of the geometry
TG	transformation (i.e. 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

13.1 ** card

The ** card is not a command, but defines a comment line. Everything that is found in this line is ignored by PREFEKO.

It is possible to add a comment to the end of an existing line or card. For example,

```
** Definition of parameters
#lambda = 1.0          ** Wavelength
#radius = #lambda/2    ** Cylinder radius
#height = 2*#lambda    ** Cylinder height
```

More information on the usage of comment lines and comments as card names may be found in the comment card description under the control cards (see section [14.1](#)).

13.2 BL card

This card is used to connect two points to form a line, which is then subdivided into wire segments.

BL - Specify a wire element

Start point of wire

End point of wire

☐ Set (tapered) wire radius

Radius at start point of wire

Radius at end point of wire

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:* If checked, the radius set at the previous IP card is overridden for the current wire. This setting does not affect segments created after this card. Both radius values are in m 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 ...:* The radius of the wire at the start point.
- Radius at end ...:* 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. Note that by using different radius values for the start and end points of the wire, a tapered wire can be created.



Figure 13-1: Sketch illustrating the use of the BL card

Examples of BL card usage:

The BL card can be used to create segmented wires as shown in Figures 13-2 and 13-3. In the first example the radius is specified with an IP card and in the second an exaggerated taper is specified at the BL card.



Figure 13-2: Example of a BL card from demo_BL1.pre



Figure 13-3: Example of a BL card with a tapered radius from demo_BL2.pre

13.3 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.

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 ...: Normally a parallelogram is segmented according to the edge length specified with the IP card. In some cases, e.g. 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 ($\hat{n}$) 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 (see section 13.37) — or with dielectrics — ME card (see section 13.31) or with the CFIE — CF card (see section 14.26).

Example of BP card usage:

Figures 13-4 and 13-5 show a plate with uniform meshing and a strip with non-uniform meshing respectively, both created using the BP card.

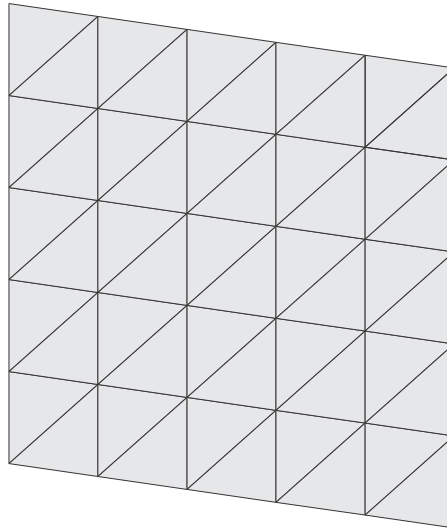


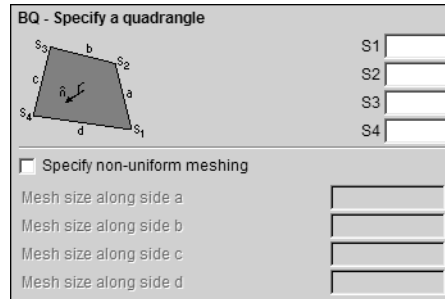
Figure 13-4: Example for the BP card from demo_BP1.pre



Figure 13-5: Example of a BP card with non-uniform meshing from demo_BP2.pre

13.4 BQ card

A mesh of surface triangles in the shape of a flat quadrangle can be created with this card. Models constructed using the BQ card can generally be simplified by using the PM card.



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 ...: Normally a quadrangle is segmented according to the edge length specified with the IP card. In some cases, e.g. when creating small microstrip lines, it may be desirable to use a finer segmentation in one direction. Check this item if finer segmentation is required along any edge. The mesh sizes are in m and are scaled by the SF card.

Mesh size along side a: Edge S1–S2.

Mesh size along side b: Edge S2–S3.

Mesh size along side c: Edge S3–S4.

Mesh size along side d: Edge S4–S1.

The points have to be predefined using DP cards prior to the BQ card and 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 BQ card can be used to create the mesh shown in Figure 13-6, or also for inhomogeneous meshing as shown in Figure 13-7.

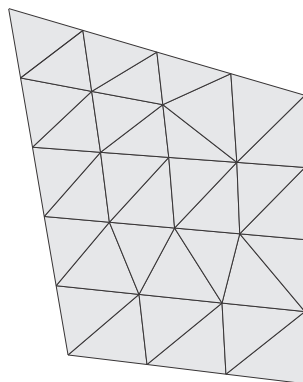


Figure 13-6: Example of a BQ card from demo_BQ1.pre

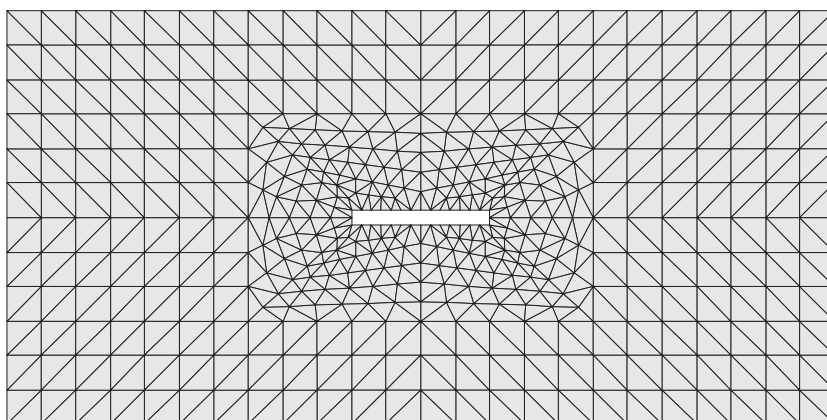
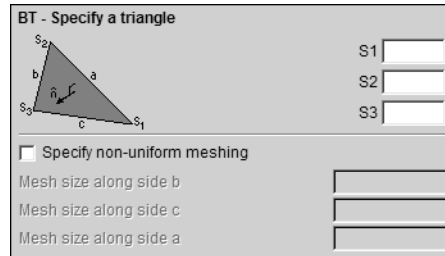


Figure 13-7: Example of a BQ card with an inhomogeneous segmentation from demo_BQ3.pre

13.5 BT card

A mesh of surface triangles in the shape of a flat triangle can be created with this card. In general, this card is replaced by the PM card.



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 ...: Normally a triangle is segmented according to the edge length specified with the IP card. In some cases it may be desirable to use a finer segmentation in one direction. Check this item if finer segmentation is required along any edge. The mesh sizes are in m and are scaled by the SF card.

Mesh size along side b: Edge S2–S3.

Mesh size along side c: Edge S3–S1.

Mesh size along side a: 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 meshes shown in Figures 13-8 and 13-9 were created with BT cards using uniform meshing and non-uniform meshing respectively.

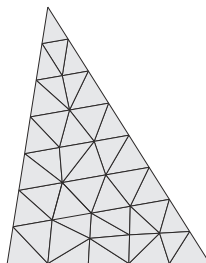


Figure 13-8: Example of a BT card from demo_BT1.pre

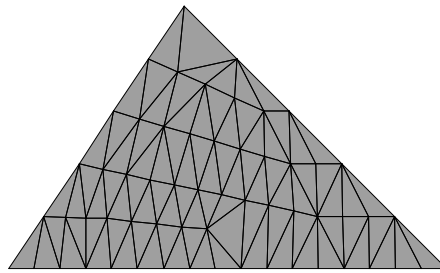


Figure 13-9: Example of a BT card with non-uniform meshing from demo_BT2.pre

13.6 CB card

This card is used to change or reassign the labels assigned to points, segments, triangles, cuboids, polygonal plates, tetrahedral elements, etc.

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) or transformations (TG card) or an imported geometry from CADFEKO, and for example, edges or wedges in the PO area are considered or any other properties shall be set by label (e.g. Skin effect). 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). So 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
```

(i.e. the old and new label entries 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 (i.e. a space or tab character). Comments line are allowed in these files and these 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 (i.e. the file `geometry.txt` would then do a proper mapping of the NASTRAN property to the part name in the original CAD program).

13.7 CL card

This card is used to create an arc consisting of wire segments.

CL - Arc of wire segments

S1

S2

S3

Subtended angle φ (degrees)

Maximum length of segments

☒ Set (tapered) wire radius

Radius at start point of wire

Radius at end point of wire

Scale second half axis with

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 ...:** 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 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 m 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 ...:** The radius of the wire at the start point.
- Radius at end ...:** The radius of the wire at the end point.
- Scale second half axis:** 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 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 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 field *Maximum length of segments*.

The radius of the arc is given by the distance between the points S1 and S3.

Examples of CL card usage:

The meshes shown in Figures 13-10 and 13-11 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.

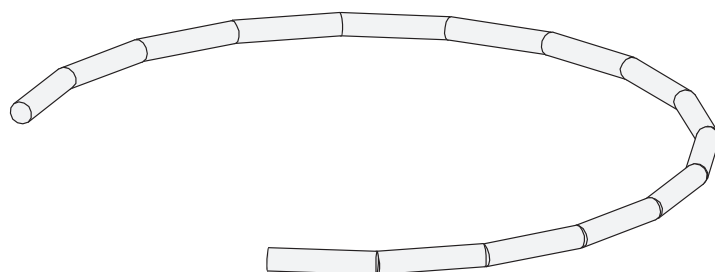


Figure 13-10: Example of a CL card from demo_CL1 .pre

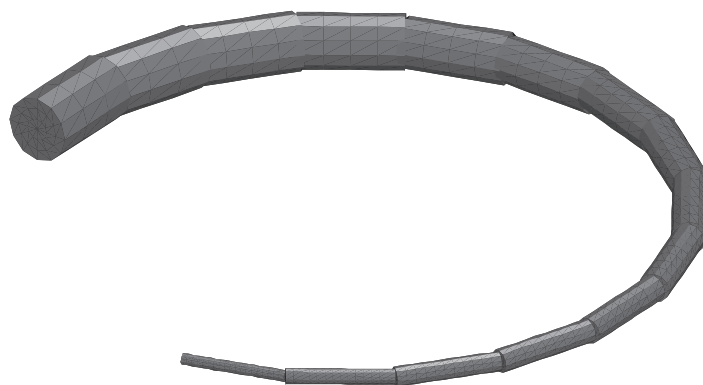
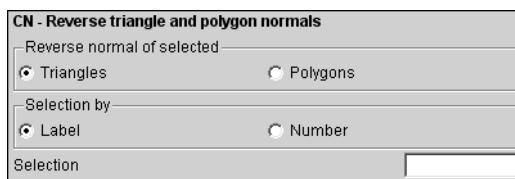


Figure 13-11: Example of a CL card with tapered wire radius from demo_CL2 .pre

13.8 CN card

This card is used to reverse the normal direction of previously created triangles or polygonal plates, for example after importing CAD data.



The image shows a dialog box titled "CN - Reverse triangle and polygon normals". It contains two groups of radio buttons. The first group, labeled "Reverse normal of selected", has "Triangles" selected (indicated by a filled circle) and "Polygons" unselected (indicated by an empty circle). The second group, labeled "Selection by", has "Label" selected (indicated by a filled circle) and "Number" unselected (indicated by an empty circle). At the bottom, there is a text field labeled "Selection" which is currently empty.

Parameters:

Reverse normal of ...: In this group, the user selects to reverse the normals of either *Triangles* or *Polygonal plates*.

Selection by: Here the user specifies whether the triangles / polygonal plates are identified by their *Label* or absolute element *Number*.

Selection: The label / element number of the triangles / polygonal plates that must have their normals reversed.

The normal direction is important in some cases, 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 as 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.

13.9 DD card

With this card a MoM model consisting of a dominant fixed static part and a smaller dynamic part, may be solved more efficient. The factorisation of the static interaction matrix is written to a *.ngf file and may then be reused to solve the MoM problem when a smaller dynamic part is modified.

DD - Domain decomposition

Domain name

Save / read NGF file

☒ No data files (normal execution)

☐ Save *.ngf file

☐ Read *.ngf file

☐ Read *.ngf file if it exists, else create it

Number of element labels

Labels of the elements

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 triangle 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 that only a single DD card is allowed in a model.

13.10 DK card

This card is used to create an eighth of a sphere, meshed into cuboidal elements, for solutions using the volume equivalence principle in the MoM. 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.

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 m) 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 here whether the sphere is dielectric or magnetic or both (this is always with respect to the environment, e.g. 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 (see section [13.40](#)).

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 the eighth of a sphere as shown in Figure [13-12](#).

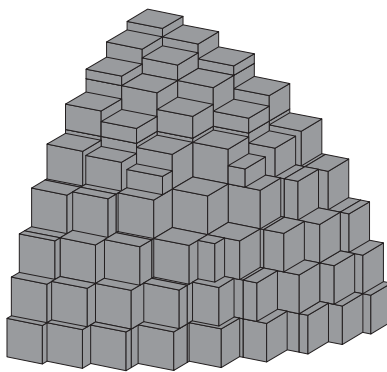


Figure 13-12: Example for the DK card from demo_DK1 .pre

13.11 DP card

With this card points in space are defined. These points are used to define the extent and orientation of other geometric entities and to locate excitations.

DP - Define an input point	
Point name	
X coordinate	
Y coordinate	
Z coordinate	
Nurb control point weight	

To avoid ambiguity each point is assigned a name (a 5 character string if column based format is used). In the other cards (e.g. BL card) the points are referred to by their names.

Parameters:

Point name: Name of the point.

X, Y, Z coordinate: Cartesian coordinates 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.

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.

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.

13.12 DZ card

This card is used to create a cylindrical shell, meshed into cuboidal elements, for solutions using the volume equivalence principle in the MoM. 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.

DZ - Specify a dielectric cylinder

S1

S2

S3

S4

The angle φ (degrees)

Maximum cuboid edge length on arc

Choose the medium

☒ Dielectric

☐ Magnetic

☐ Both dielectric and magnetic

☐ Old format (with medium parameters)

Parameters:

S1, S2: The start and end points, respectively, of the cylinder axis.

S3, S4: Points on the inside and outside, respectively, 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, e.g. 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 (see section 13.40).

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:

Figure 13-13 is an example of a cylindrical shell created with the DZ card.

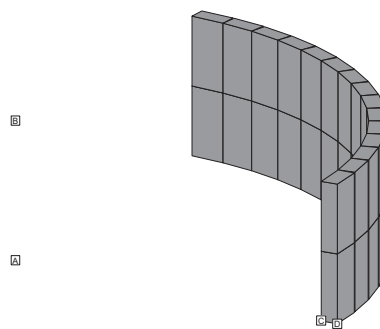


Figure 13-13: Example for the DZ card from demo_DZ1.pre

13.13 EG card

This card indicates the end of the geometrical input. It is essential that this card is used.

EG - End of the geometry input

Write geometrical output to

☐ FEKO output file (*.out)

☐ Nastran file (*.nas)

☐ STL file (*.stl)

Send to standard output

☐ Nothing

☒ Warnings, errors, progress messages

☐ Switch normal geometry checking off

Maximum identical distance

Solution accuracy

☐ Double precision

☒ Single precision

☐ Activate low frequency stabilisation for MoM

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 will be switched off (see the discussion below this table).

Solution accuracy: This parameter can be set to force FEKO to use single precision for the storage of some 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: If this item is checked, low frequency stabilisation for MoM is activated.

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 (wires, tetrahedra etc.) 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, some information is lost, for instance for wire elements the thickness (wire radius) cannot 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 (see section 12.4) just the associated number is used and exported as the NASTRAN property.

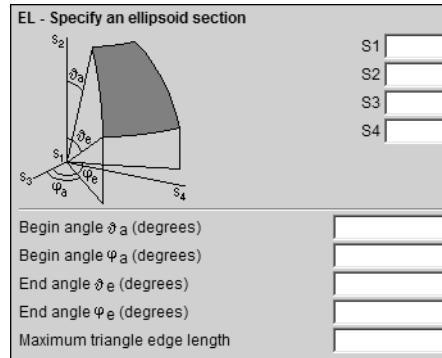
The *Maximum identical distance* is used to set the tolerance in the mesh. The mesh information is created by the program PREFEKO, and stored in a *.fek file, in which all the triangles and segments are described by their corner points. Due to rounding errors it is possible that, for example, end points 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 (see section 2.2.3). 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.

13.14 EL card

A mesh of surface triangles in the shape of an ellipsoidal section can be created with this card.



EL - Specify an ellipsoid section

S1

S2

S3

S4

Begin angle ϑ_a (degrees)

Begin angle φ_a (degrees)

End angle ϑ_e (degrees)

End angle φ_e (degrees)

Maximum triangle edge length

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, i.e. 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 in 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 on the surface of the ellipsoid is given as

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a \sin \vartheta \cos \varphi \\ b \sin \vartheta \sin \varphi \\ c \cos \vartheta \end{bmatrix} \quad (13-1)$$

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 13-14 is generated by using the EL card.

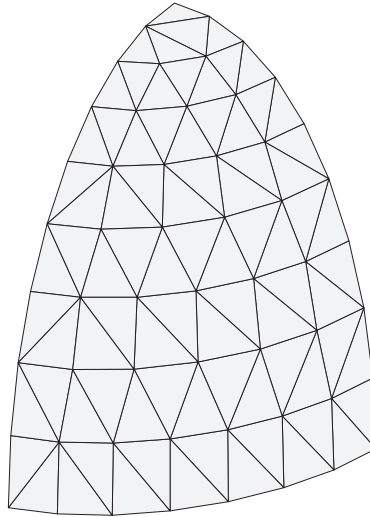


Figure 13-14: Example for the EL card from demo_EL.pre

13.15 FM card

This card is used to instruct the FEKO solver to calculate the solution using the Multilevel Fast Multipole Method (MLFMM) instead of the MoM on all structures in the simulation. A second method, the Adaptive Cross Approximation method is also available and is applicable to low frequency problems or when using a special Green's function.

Parameters:

Use MLFMM: The Multilevel Fast Multipole Method method is used to calculate the solution.

Use adaptive cross-approximation (ACA): The ACA method is used to calculate the solution. This method approximates the impedance matrix by constructing a sparse H-matrix (only a few selected elements are computed).

Box size at fines 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).

Near field calculation method: The MLFMM method can use a fast near field calculation method (this is the default), but under certain circumstances it is preferred to use the traditional integration method. The method to use can be selected here.

Far field calculation method: The MLFMM method can use a fast far field calculation method (this is the default), but under certain circumstances it is preferred to use the traditional integration method. The method to use can be selected here.

Multilevel Fast Multipole Method (MLFMM)

Currently, the MLFMM may not be used in conjunction with the Multi-layer Greens function.

Advantages of Adaptive Cross Approximation (ACA)

- ACA is done on the matrix level.
- No frequency breakdown like MLFMM.
- Not restricted to free space Green's function like MLFMM.
- Iterative solution, each matrix-vector multiplication is very fast.

Note that the ACA can currently only be used sequentially.

13.16 FO card

With this card an area is defined in which the surface current density is an approximation according to the Fock theory.

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 (e.g. the surface of the cylinder).

Axis start / Axis end point: These dialogs are only visible for Fock cylinders and should correspond to the start and end points 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 Method by Daniel Bouche or the Method by Louis N. Medgyesi–Mitschang.

Decouple with moment method: Check this item 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.

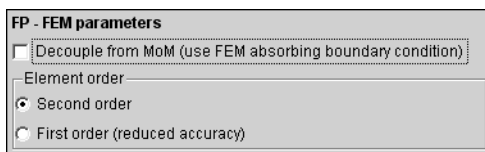
The radius of the cylinder or sphere does not have to be defined. It is determined by the distance to the metallic triangles, with the label specified in *Triangle labels*.

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, i.e. 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.

13.17 FP card

With this card options related to the FEM (*Finite Element Method*) can be set.



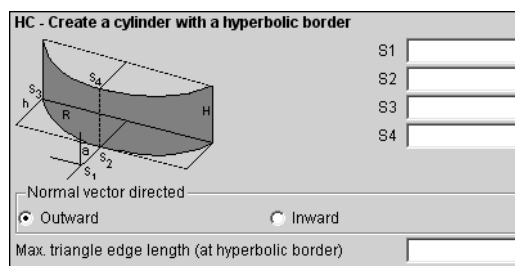
Parameters:

Decouple from MoM (use FEM absorbing boundary condition): Normally within the MoM / FEM hybrid method these two techniques are fully coupled, and also the surface of the FEM region is treated by the MoM. For certain applications when there is a larger separation between the MoM and the FEM regions (e.g. human body with a GSM base station antenna) this decoupling checkbox can be checked. Then similar to switching off the coupling for the MoM/PO or MoM/UTD hybrid methods, first the MoM region is solved for by neglecting the FEM domain, and this MoM solution is then used as impressed excitation for the FEM. Also the MoM is in this de-coupled case no longer used on the FEM surface, but rather an absorbing boundary condition is applied there. The advantage of this decoupling is a saving of memory and computation time.

Element order: When second order is selected, FEKO uses hierarchal 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. 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. Thus second order is the 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.

13.18 HC card

This card creates a cylinder with a hyperbolic border.



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 can be 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 13-15 is created using the HC card.

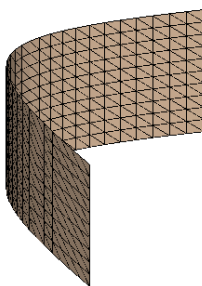


Figure 13-15: HC card examples — These examples are found in `demo_HC1.pre`.

13.19 HE card

With this card a helical coil, consisting of wire segments, can be created.

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 ...: Create connections from the two ends of the coil to the axis (at points S1 and S2). See also left side of Figure 13-16. If the connections are not generated, point S3 is a connection point. See also the right side of Figure 13-16.

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) 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 ...: The radius of the wire at the start of the coil.

Radius at end ...: The radius of the wire at the end point of the coil.

Scale second half axis: 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 shown in Figure 13-16 are created using the HE card.

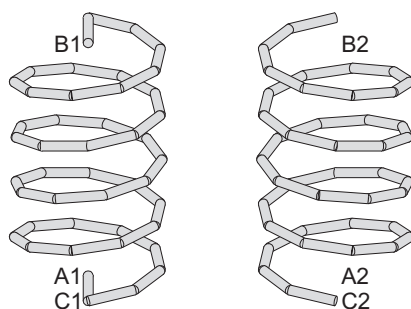
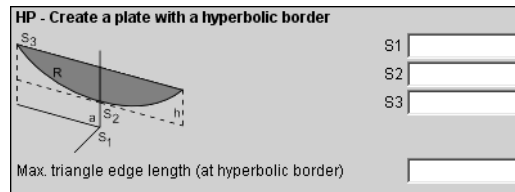


Figure 13-16: HE card examples — the coil on the right is coiled in the left handed direction. These examples are found in `demo_HE1.pre`

13.20 HP card

This card creates a plate with a hyperbolic border.



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 13-17 is created using the HP card.

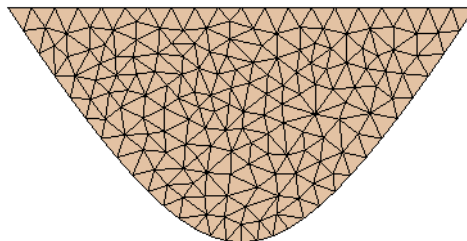
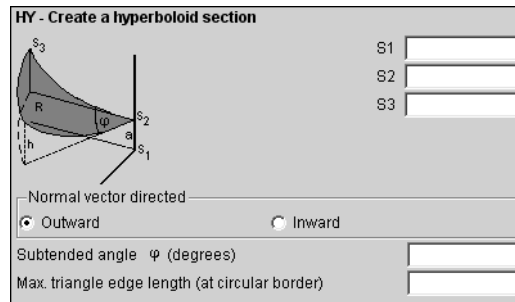


Figure 13-17: HP card example. This example is found in demo_HP1.pre

13.21 HY card

This card creates a hyperboloid section.



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. Thus 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 in Figure 13-18 is created using the HY card.

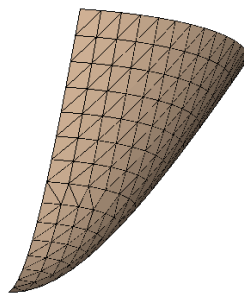


Figure 13-18: HY card example. This example is found in demo_HY1.pre

13.22 IN card

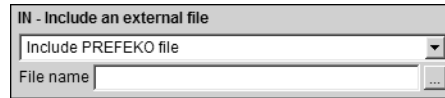
This card is used to include external files. These files may be other *.pre files (which are included as if they were part of the master file) or mesh data files containing wire segments, triangles, quadrangles, tetrahedral volume elements and/or polygonal plates (in FEMAP neutral, an ASCII format, NASTRAN, meshed AutoCAD *.dxf, NEC model, Concept model, STL, PATRAN neutral, ANSYS *.cdb, ABAQUS *.inp or GiD *.msh mesh files).

Some 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. (See below.)
- 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.

PREFEKO file

This option is used to include cards and commands (such as variable definitions etc.) from a separate file. One may, for example, create a single file with an antenna which is then imported into different environment models.



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 some 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 (i.e. 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.

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).

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.

Import special ASCII data file

This option is used to import meshes stored in the geometry data file as specified below.

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 meaning of the above is:

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 m (is scaled by the SF card)
y(i)	y coordinates of node i in m (is scaled by the SF card)
z(i)	z coordinates of node i in m (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 m
p1(m)	Number (index) of the first corner of polygon m
p2(m)	Number (index) of the second corner of polygon m
p3(m)	Number (index) of the third corner of polygon m
t1(m)	Number (index) of the first corner of tetrahedron m
t2(m)	Number (index) of the second corner of tetrahedron m
t3(m)	Number (index) of the third corner of tetrahedron m
t4(m)	Number (index) of the third corner of tetrahedron m
...	Number (index) of the additional corners of polygon m
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 13-19, 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
```

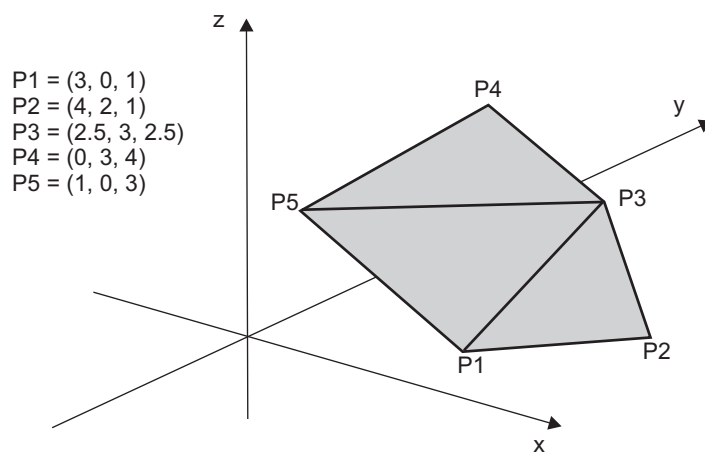


Figure 13-19: Example for IN card

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.

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 etc. 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 73 to 96 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 commas

```

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,,

```

Import AutoCAD DXF file

This card allows importing *.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:

Include segments: Same as above.

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: 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 keywords ENTITIES and ENDSEC.

The user can import points (i.e. 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 (i.e. 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 some 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 some 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.

Import NEC model file

PREFEKO also supports importing wire geometry from NEC⁶ 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.

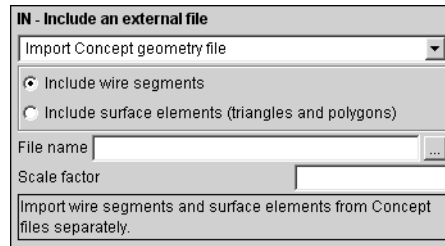
The label selection uses the NEC tags which are converted to FEKO labels. This applies to the tag when the element is defined. If the tag is modified after the inclusion (for example with the GM card) the elements with the modified tag are also included. The type selection parameter *x* is also supported, 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.

⁶G.J. Burke and A.J. Poggio, “Numerical Electromagnetics Code (NEC) — Method of Moments,” Lawrence Livermore Laboratory, January 1981.

Import Concept geometry file

With this option one may import CONCEPT⁷ geometry files.



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

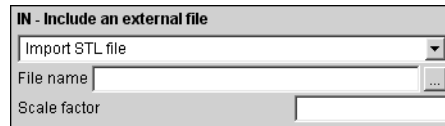
```
number_of_nodes    number_of_patches
x y z              [number_of_nodes times]
p1 p2 p3 p4        [number_of_patches times]
```

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.

⁷A MoM code developed at the University of Hamburg-Harburg, Germany.

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.



An example of an ASCII STL file is

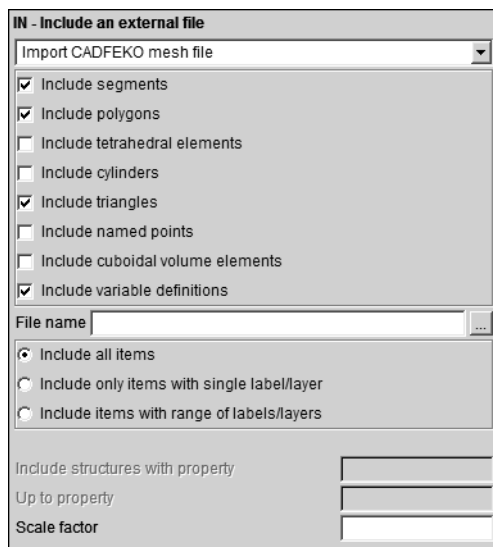
```
SOLID CATIA STL PRODUCT
  FACET NORMAL -4.602166E-01 -1.858978E-01 -8.681260E-01
    OUTER LOOP
      VERTEX 4.789964E-01 -8.440244E-01 2.878882E-01
      VERTEX 4.764872E-01 -8.439470E-01 2.892018E-01
      VERTEX 4.783065E-01 -8.414296E-01 2.876983E-01
    ENDLOOP
  ENDFACET
  FACET NORMAL -4.601843E-01 -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:

<http://www.ennex.com/~fabbers/StL.asp>

Import CADFEKO mesh file

CADFEKO exports the mesh to a *.cfm file which is imported by an IN card in the default *.pre 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.



IN - Include an external file

Import CADFEKO mesh file

- ☒ Include segments
- ☒ Include polygons
- ☐ Include tetrahedral elements
- ☐ Include cylinders
- ☒ Include triangles
- ☐ Include named points
- ☐ Include cuboidal volume elements
- ☒ Include variable definitions

File name

☒ Include all items

☐ Include only items with single label/layer

☐ Include items with range of labels/layers

Include structures with property

Up to property

Scale factor

Import PATRAN neutral file

PREFEKO also supports importing PATRAN files.

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.

Import ANSYS CDB file

PREFEKO also supports importing 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.

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 that points are not included by default.

Import ABAQUS mesh file

PREFEKO also supports importing mesh files from ABAQUS *.inp files. The various options for the card panel are as described globally or at the other import options:

IN - Include an external file

Include ABAQUS file

☒ Include segments

☒ Include quadrangles

☐ Include tetrahedral elements

☒ Include triangles

☐ Include named points

☐ Include only node points for imported triangles and/or wires

☐ For tetrahedral volume elements use the imported label also as medium index

File name

☒ Include all items

☐ Include only items with single label/layer

☐ Include items with range of labels/layers

(ABAQUS element sets convert to FEKO labels)

Include structures with property

Up to property

Scale factor

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, i.e. 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.

Import GiD mesh

PREFEKO also supports importing mesh files from GiD *.msh files.

The screenshot shows a dialog box titled "IN - Include an external file". It contains the following elements:

- A dropdown menu labeled "Import GiD file".
- A list of checkboxes for element types:
 - ☒ Include segments
 - ☒ Include polygons
 - ☒ Include quadrangles
 - ☐ Include tetrahedral elements
 - ☒ Include triangles
 - ☐ Include named points
 - ☐ Include cuboidal volume elements
 - ☐ Include only node points for imported triangles and/or wires
- A checkbox labeled "For tetrahedral volume elements use the imported label also as medium index".
- A text field labeled "File name" with a browse button (...).
- A group of radio buttons for inclusion criteria:
 - ☒ Include all items
 - ☐ Include only items with single label/layer
 - ☐ Include items with range of labels/layers
- A section titled "(GiD material types/layers convert to FEKO label)" with three input fields:
 - Include structures with property
 - Up to property
 - Scale factor

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.

13.23 IP card

With this card a number of parameters that define the meshing parameters or also the wire radius are set.

IP - Specify the segmentation parameters	
Radius of wire segment	
Maximum triangle edge length	
Maximum wire segment length	
Maximum cuboid edge length	
Maximum tetrahedral edge length	

Parameters:

Radius of wire segment: Segment radius in m (it is scaled by the SF card if used).

Maximum triangle edge length: Maximum edge length of triangular elements in m (it is scaled by the SF card).

Maximum wire segment length: Maximum segment length for wire segments 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 MoM) in m (it is scaled by the SF card).

Maximum tetrahedral edge length: Maximum edge length of tetrahedral volume elements (FEM) in m (it is scaled by the SF card).

The IP card only affects the commands following it, i.e. it 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 when different radii occur in the geometry. For any command (e.g. the BL card) the previous IP card is applicable.

Regarding the meshing, certain rules apply relating the element size (see section [2.2.2](#)) to the wavelength etc.

13.24 KA card

With this card two points are joined, which form the border of the PO area. On this edge, fringe wave currents are taken into account.

KA - Define a PO region connection edge	
Start point of edge	
End point of edge	
Label of triangles on the PO border	

Parameters:

Start point of edge: Start point of the edge.

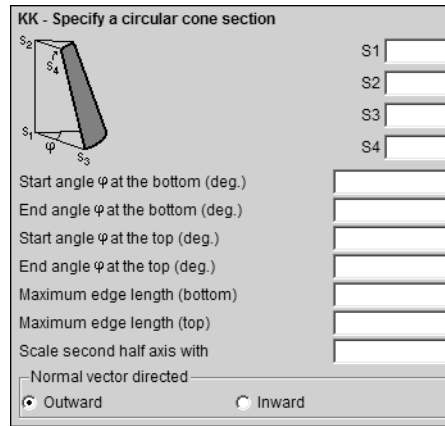
End point of edge: End point. The direction of an edge is arbitrary, i.e. it does not matter which edge point is chosen as the end or start point of the edge.

Label of triangles ...: Label of the PO triangles adjacent to the PO border, i.e. the edge correction current from this edge is applied to all triangles with this label.

Note that the surface must be flat, i.e. all triangles with the label specified here must lie in the same plane.

13.25 KK card

A mesh of surface triangles in the shape of a conical section can be created with this card. The cone can also be distorted to have an elliptical cross section. Cones or conical sections with included angles that vary from the top to the bottom, or that do not start in a specified plane, can also be created.



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 the cone has 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.

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 and a conical section where the start angle is not in the plane S2–S1–S3.

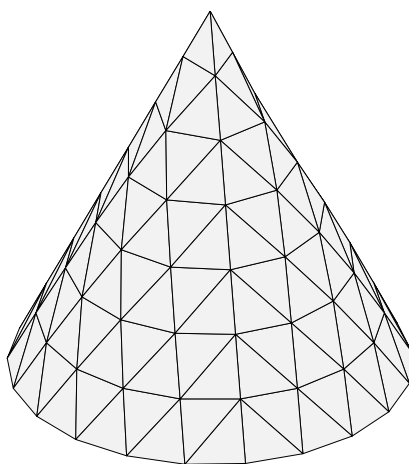


Figure 13-20: Example for the KK card from demo_KK1.pre

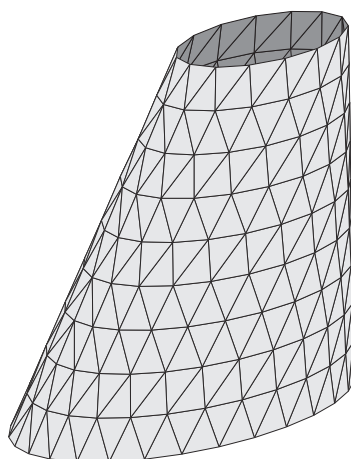


Figure 13-21: Example of a cone with an elliptical cross section from demo_KK2.pre

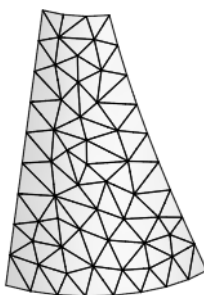


Figure 13-22: Example of a cone with different subtended angles at the top and at the bottom from demo_KK3.pre

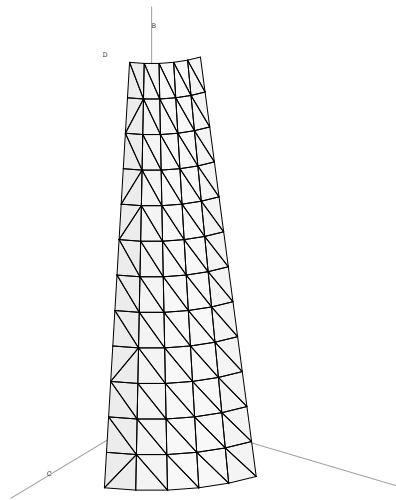
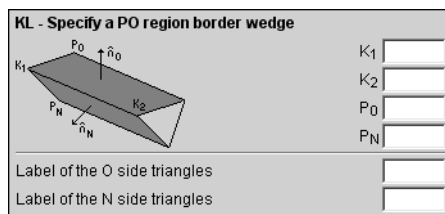


Figure 13-23: Example of a conical section where the start angle is not in the plane defined by S1, S1 and S3 (demo_KK4.pre)

13.26 KL card

This card defines a wedge for which correction terms are added to the PO currents on the two surfaces connected to it.



Parameters:

K_1 : The start point of the edge of the wedge.

K_2 : The end point of the edge of the wedge.

P_0 : A point on the O side of the wedge.

P_N : A point on the N side of the wedge.

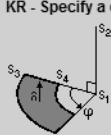
Label of the O side triangles: The label of the PO triangles that are adjacent to the wedge on the O side. This means that the corresponding correction term for the O side is assigned to the PO triangles that have this label.

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 that the wedge must be between flat surfaces, and that all triangles with the label specified here must lie in the same plane.

13.27 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 distort it to an elliptical region.



KR - Specify a circular section

S1

S2

S3

S4

The angle φ (degrees)

Maximum edge length (outer)

Maximum edge length (inner)

Scale second half axis with

Parameters:

- S1:** The centre point of the circle.
- S2:** A point that is situated 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 lie between S1 and S3.
- The angle φ :** 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, 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:** 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) is generated by sweeping the edge S3–S1 around the axis S1–S2 through φ degrees in the mathematically positive sense. For $\varphi = 360^\circ$ 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 KR card usage:

The following example meshes are all created using the KR card. Shown is a circular plate, a flat circular ring and a flat elliptical ring.

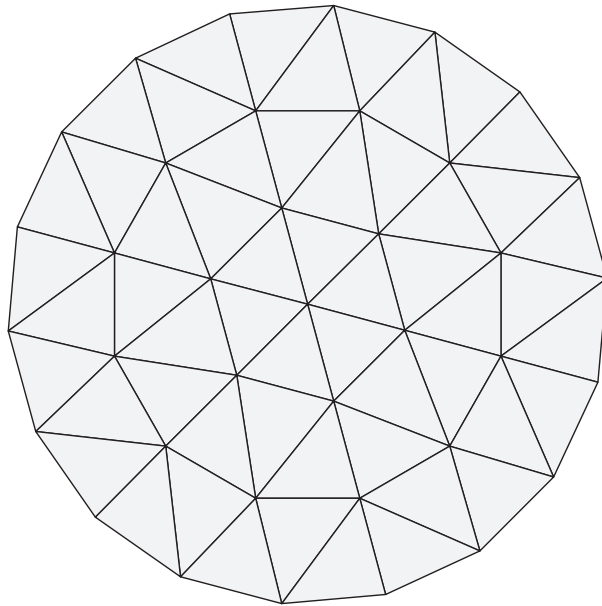


Figure 13-24: First example for the KR card from demo_KR1.pre

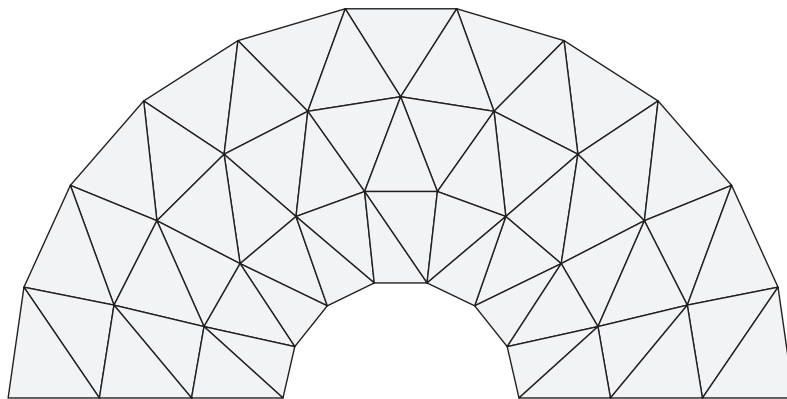


Figure 13-25: Second example for the KR card from demo_KR2.pre

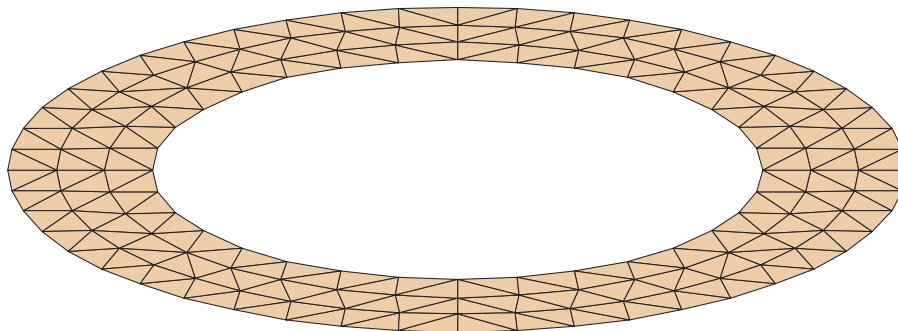
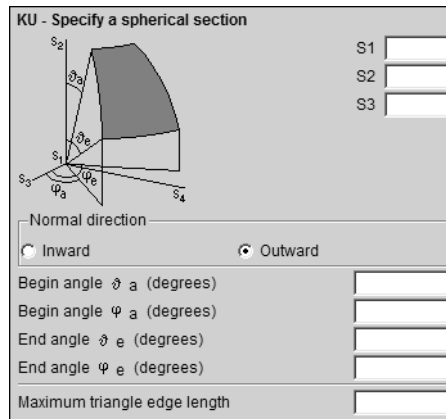


Figure 13-26: Example of an elliptical disk with a hole from demo_KR3.pre

13.28 KU card

This card creates a mesh of surface triangles in the shape of a spherical section.



Parameters:

- S1:** The centre of the sphere.
- S2:** A point that indicates the $\vartheta = 0^\circ$ direction in a spherical coordinate system. The distance between S1 and S2 is the radius of the sphere.
- S3:** A point that indicates the $\vartheta = 90^\circ$, $\varphi = 0^\circ$ 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 :** The start angle ϑ_a in degrees of the spherical segment.
- Begin angle φ_a :** The start angle φ_a in degrees of the spherical segment.
- Begin angle ϑ_e :** The end angle ϑ_e in degrees of the spherical segment.
- Begin angle φ_e :** The end angle φ_e in degrees of the spherical segment.
- Maximum triangle edge length:** 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 $\vartheta_a = \varphi_a = 0$, $\vartheta_e = 180^\circ$ and $\varphi_e = 360^\circ$.

An example of KU card usage:

The spherical segment shown in Figure 13-27 is generated using the KU card.

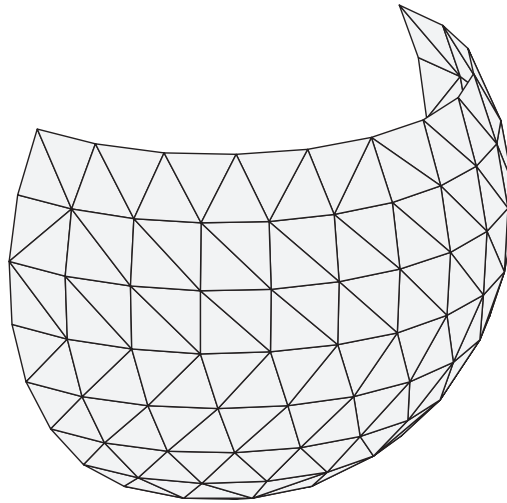


Figure 13-27: Example for the KU card from demo_KU1.pre

13.29 LA card

With this card, labels are assigned to segments, triangles, polygonal plates, cuboids, UTD cylinders and points.

LA - Assign labels
Label for following geometry

Parameters:

Label for following ...: The label assigned to all segments, triangles, etc. defined in cards following this one.

In order to select the position of the feed (Ax cards)⁸, 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, etc. is assigned a label. Other applications include the selective transformation or copying of parts of geometry (TG card), and outputting of currents on selected elements (OS card). Labels are also used to define triangles on which to apply physical optics (PO card (see section 13.37)).

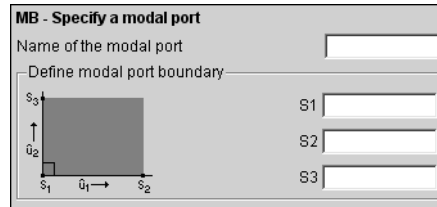
All elements, etc. that are created after the LA card (e.g. by a BP card), are assigned the value specified in the dialog as label. A different label is only assigned by a new LA card. All structures created before the first LA card (or if no LA card is present), is assigned the default label which is 0 (number zero).

Label names (see section 12.4) 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 (see section 12.4) and referenced using label ranges (see section 12.4).

⁸The definition Ax stands for any of the control cards A0, A1, A2, ...

13.30 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.



Parameters:

Name of the modal port: The label of the modal port.

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 region, situated on a planar surface.

The technology behind a modal port is a two dimensional FEM eigensolver which 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 memory requirement scales with N^2 , where N is the number of unknowns.

The user should take note that memory and time scaling may 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).

13.31 ME card

When solving the fields in dielectric objects by means of the MoM/MLFMM surface, volume current methods or by means of the FEM or VEP, this card must be used to distinguish the different media or to create segments and metallic triangles within the dielectric. Furthermore, this card is used to switch between the generation of metallic triangles and triangles that represent the surface of the dielectric. Another special case is when metallic triangles represent the surface of a dielectric object (e.g. a dielectric that has been coated with metal).

ME - Define a dielectric region

Type of triangles

- ☒ Metallic triangles in a homogeneous medium
- ☐ Triangles representing the surface of a dielectric region
- ☐ Metallic triangles representing the surface of a dielectric region
- ☐ Planar Green's function aperture triangles in a homogeneous medium

Type of tetrahedra

- ☒ Finite element method (FEM)
- ☐ Volume equivalence principle - dielectric
- ☐ Volume equivalence principle - magnetic
- ☐ Volume equivalence principle - dielectric and magnetic

Medium for triangles/segments

All segments and triangles following this card will lie in the specified medium.
 All tetrahedral and cuboidal volume elements following this card will have the properties of the specified medium.
 Note that the outer medium (extending to infinity) must always be medium 0.

Parameters:

Metallic triangles in a homogeneous medium: If this option is selected, then all the surface structures between this card and the next ME card are assumed to be fully contained inside the medium specified in the *Medium for triangles/segments* dialog.

Triangles representing the surface of a dielectric body: If this option is selected, then 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 medium on both sides of the triangles. The normal vector points from medium *A* to medium *B*. Thus if we have a dielectric body of medium DIELECTRIC, constructed so that all the triangle normals point outward, we set medium *A* to for instance DIELECTRIC and *B* to 0 (the number zero always represents the outer free space region).

Metallic triangles representing the surface of a dielectric body: If this option is selected, then 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 discussed above.

Planar Green's function aperture triangles in a homogeneous medium: If this option is selected, then all the surface structures between this card and the next ME card are assumed to be fully contained inside the medium specified in the *Medium for triangles/segments* dialog.

Finite element method (FEM): If this option is selected, the tetrahedral mesh elements will be solved with the FEM.

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 found. Triangles are treated differently — it depends upon 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 a ME card represent metallic structures in free space. This is also the case when an input file does not have a ME card.

When using the FEM and meshing structures into tetrahedral elements or when using the volume equivalence principle (VEP) in connection with the MoM/MLFMM and meshing into cuboidal volume elements, then the selection for type of triangle 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 ‘_’. See FEKO User Manual (section 12.4) for details, also with respect to ranges etc. 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 `example_04` (see the *Script Examples*, note that the normal vectors of the sphere point outwards from medium 1 — the dielectric — to medium 0 — free space). In addition `example_23` shows the more complex example of a dielectric cone (medium 1) mounted on top of a metallic cylinder shown in Figure 13-28. 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 basis 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

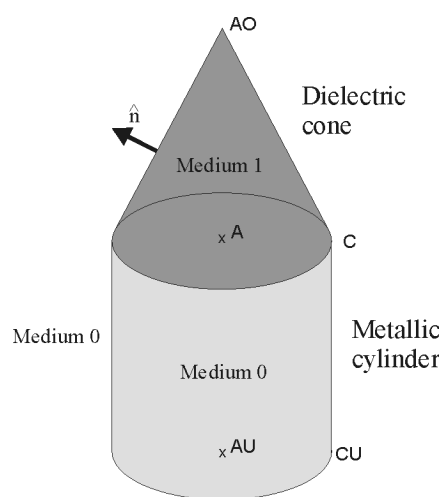


Figure 13-28: Example of a dielectric cone on top of a metallic cylinder, to demonstrate the use of the ME card.

13.32 NU card

Surface triangles representing a NURBS surface are created using this card.

NU - Specify a NURBS surface

4 Degree (p) of Bezier curve in u-direction

4 Degree (q) of Bezier curve in v-direction

Enter the control points:
 Rows correspond to the v-direction
 Columns correspond to the u-direction

	1	2	3	4	5
1					
2					
3					
4					
5					

Parameters:

Degree (p) of Bezier curve in u-direction: The degree of the Bézier curve in the $\hat{u}$ -direction.

Degree (q) of Bezier curve in v-direction: The degree of the Bézier curve in the $\hat{v}$ -direction.

Both p and q must be in the range from 1 to 4 where 1 is linear, 2 quadratic, and so on. 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 in Figure 13-29 and the linear-quadratic shape in Figure 13-30 are generated with NU cards using 4 and 6 control points respectively.

NURBS may also be used to generate flat surfaces with curved edges. The section of a circular plate with a square hole in Figure 13-31 is generated using a NU cards.

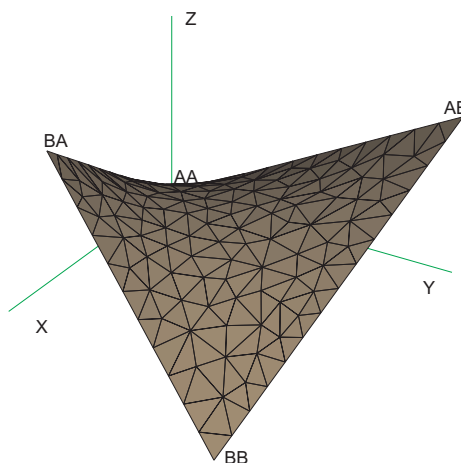


Figure 13-29: “Saddle point” example using the NU card from demo_NU1.pre

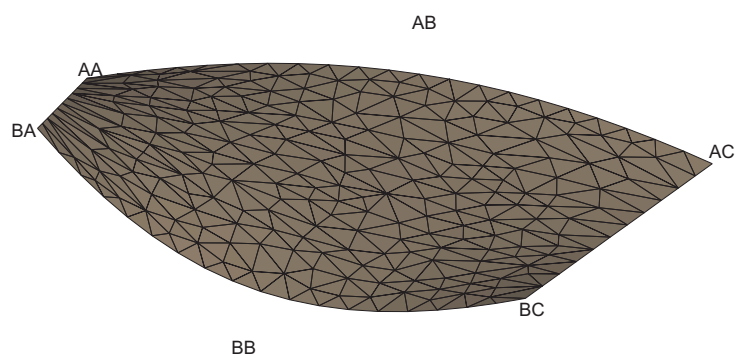


Figure 13-30: Linear-quadratic example using the NU card from demo_NU2.pre

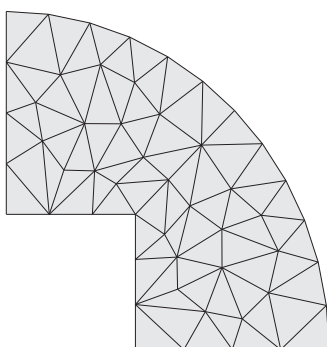
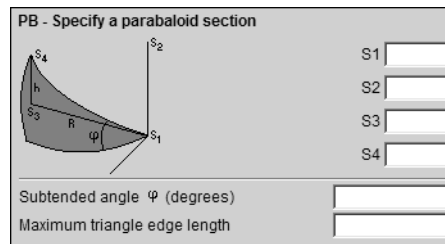


Figure 13-31: A flat surface with curved edges created with an NU card in demo_NU3.pre

13.33 PB card

This card can be used to generate a section of a parabolic reflector as shown in the figure on the card.



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:* 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:

$$f = \frac{R^2}{4h}. \quad (13-2)$$

Example of PB card usage:

The parabolic reflector as shown in Figure 13-32 can be generated by using the PB card.

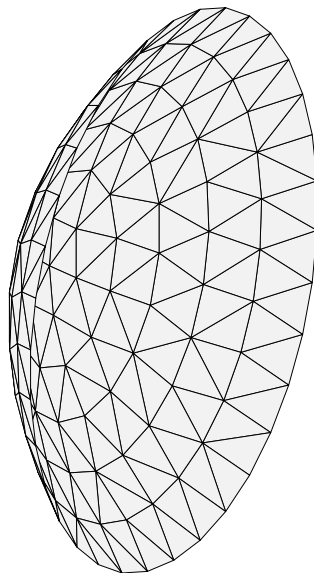
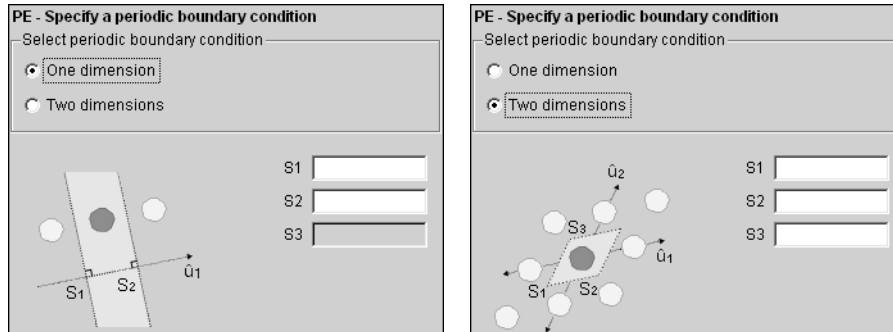


Figure 13-32: Example for the PB card from demo_PB1.pre

13.34 PE card

This card defines the unit cell for a periodic boundary condition (PBC) calculation. The phase change between the cells are specified using the PP card.



Parameters:

One dimension: One dimensional periodicity is used when the unit cell is repeated along a line (one dimension). See the image for a definition of the unit cell.

Two dimensions: Two dimensional periodicity is used when the unit cell is repeated to form a surface. See the image for a definition of the unit cell.

S_1 : Name of the point S_1 .

S_2 : Name of the point S_2 .

S_3 : Name of the point S_3 . (Only required for two dimensional periodicity.)

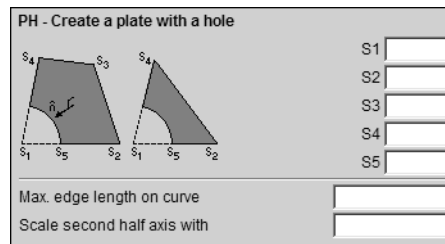
The lattice vectors ($\hat{u}_1$ and $\hat{u}_2$) do not have to be orthogonal. Geometry may also cross the periodic boundary as long as it lines up with the geometry edge on the opposite side of the unit cell. These features allow a large number of problems to be solved.

There are some restrictions on the current implementation of periodic boundary conditions. These are listed below:

- PBC is not available in conjunction with volume equivalence dielectrics (VEP).
- Triangles and wires are not allowed to cross, but are allowed to touch the the cell boundary.
- PBC can only 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, UTD or FEM techniques.

13.35 PH card

This 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.

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: 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 13-33. Note the extremely narrow triangles at the corners as mentioned above. Figures 13-34 and 13-35 show, respectively a quadrangular and a rectangular plate with the same elliptical hole. The triangular plate is obtained by leaving the field S3 empty.

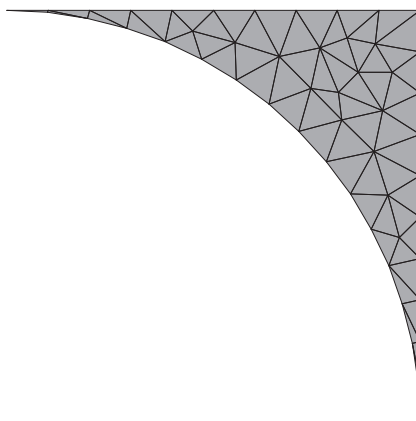


Figure 13-33: Example using the PH card from demo_PH1.pre

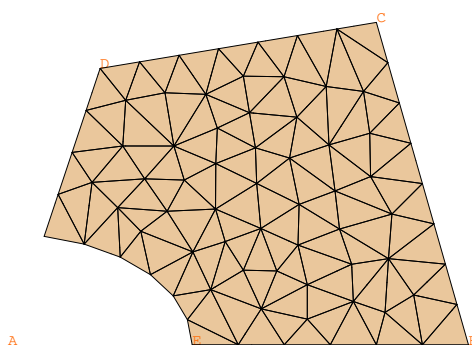


Figure 13-34: Example of a quadrangular plate with an elliptical hole (demo_PH2.pre)

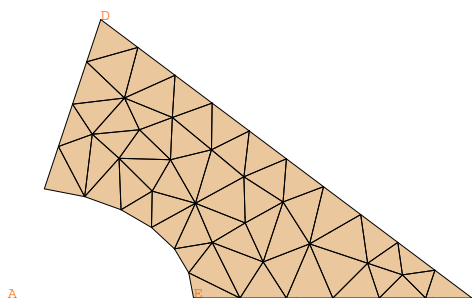


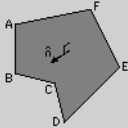
Figure 13-35: Example of a triangular plate with an elliptical hole (demo_PH3.pre)

13.36 PM card

A surface mesh of triangles in the shape of a polygon is created by using this card. This card also allows the specification of interior mesh points. This card should generally be used in favour of other cards that create flat surface meshes with straight edges.

PM - Polygon to mesh into triangles

☒ Specify points table/s
☐ Use point/variable arrays
☒ Specify non-uniform meshing
☒ Specify internal points



Corner points		Internal points		Edge length	
A		A		A - B	
B		B		B - C	
C		C		C - D (A)	
D		D		D - E (A)	
E		E		E - F (A)	
F		F		F - G (A)	
G		G		G - H (A)	
H		H		H - I (A)	
I		I		I - J (A)	
J		J		J - K (A)	
K		K		K - L (A)	
L		L		L - M (A)	
M		M		M - A	

Point-array Point-array Variable array

 Num of pts in array Num of pts in array

Note: Points are connected in the order that they are listed. The last specified point is connected back to point A. Concave corners are allowed.

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 ...: 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 13-36 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 13-37. Note that there are node points at Q1, Q2 and Q3.

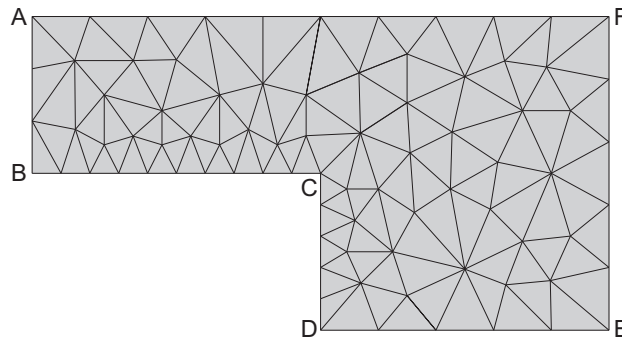


Figure 13-36: Example for the PM card from demo_PM1.pre

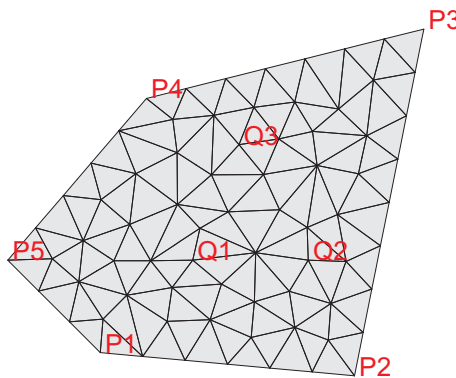


Figure 13-37: Example for the PM card with internal mesh points from demo_PM2.pre

13.37 PO card

With this card the application of the physical optics approximation is possible.

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 (see section 12.4).

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 speedup is always used. If geometrical symmetry is used, then this item should be checked to utilise symmetry. It is possible to e.g. 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.

Decouple with moment method: 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, which should lead to some saving in computational time and storage space, is especially useful when the PO region and the MoM is not directly adjacent.

Use large element PO formulation: When this item is checked, electrically large triangular patches for PO are allowed.

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 speedup — 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 FEKO 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, i.e. 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, i.e. 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 main 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, i.e. 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 (e.g. 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, i.e. 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 or MLFMM.
- The PO solution only or the MoM/PO hybrid solution is supported, but it is then required that the MoM and PO regions are decoupled.
- In order to get one wave propagation factor k_n in the PO region, sources (e.g. 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, i.e. near fields are inaccurate closer than $\lambda/20 \dots \lambda/10$ 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 PO region is present. When multiple reflections are present, ray-launching GO 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 (i.e. parallel ray tracing, parallel field computations etc.)
- Near field (E, H and potentials) computations as well as far field computations including RCS are allowed.
- All impressed sources may be used in conjunction with the large element PO.

13.38 PY card

This card defines (by specifying the corner points) a polygonal plate surface to which the UTD formulation is applied.

PY - Specify a polygonal plate

Specify corner nodes of the polygon.

☒ Specify a points table
☐ Use a points array

(Points are connected in the order that they are listed. Concave corners are allowed.)

	1
A	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	

Array name

Number of points in array

Parameters:

Specify points table: This option allows data entry in a table with a maximum of 26 points.

A: The first corner point of the polygon.

B: The second corner point of the polygon, etc.

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 the array: The number of elements, say 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 13-38. Note that this triangle is not meshed, as the result would be if the BT or PM cards had been used.

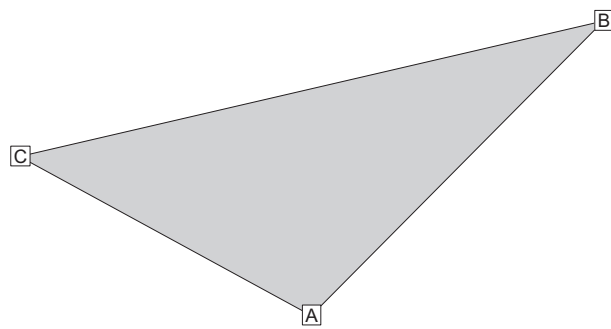
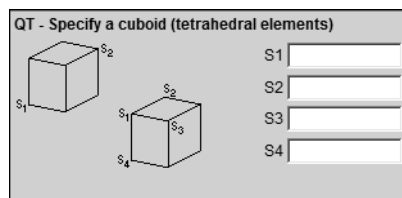


Figure 13-38: Example for the PY card from demo_PY1 .pre

13.39 QT card

This card is used to create a dielectric or magnetic cuboid, meshed into smaller tetrahedral volume elements (for solutions using the FEM or VEP). 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 tetrahedral elements.



Parameters:

- S1:** First corner of the cuboid.
- S2:** Opposite corner of the cuboid if aligned with the principle 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.

Note that when using the FEM, in the same model metallic structures are allowed (metallic surfaces also inside the FEM region or on the FEM boundary, wires only outside). But using dielectric bodies inside the MoM region at the same time is not supported.

Example of QT card usage:

The dielectric cuboid shown in Figure 13-39 is generated using a QT card.

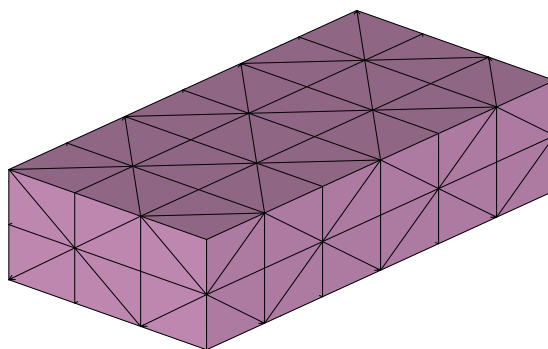
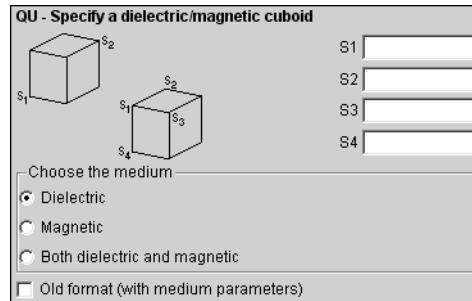


Figure 13-39: Example for the QT card from demo_QT1.pre

13.40 QU card

This card is used to create a dielectric or magnetic cuboid, meshed into smaller cuboidal volume elements, for solutions using the volume equivalence principle in the MoM. 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.



Parameters:

S1: First corner of the cuboid.

S2: Opposite corner of the cuboid if aligned with the principle 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, e.g. 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): Up to and including FEKO Suite 4.3 for cuboids the material parameters were specified directly at the QU card. This is the old card format. From FEKO Suite 5.0 onwards the concept of ME/DI cards is used to define the material by name and to set the material parameters. When checking this option, the panel layout will change to the old format so that the material parameters can be entered (depending then on the selection whether dielectric or magnetic cuboid). FEKO then uses a compatibility mode and creates artificial media with names QU_MED_xx (xx is an index). It is not recommended to use the old card format for new models. When working on old models and pressing F1 in EDITFEKO on an existing QU card, the old format panel will be opened automatically.

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 QU card usage:

The dielectric cuboid, shown in Figure 13-40 is generated using a QU card.

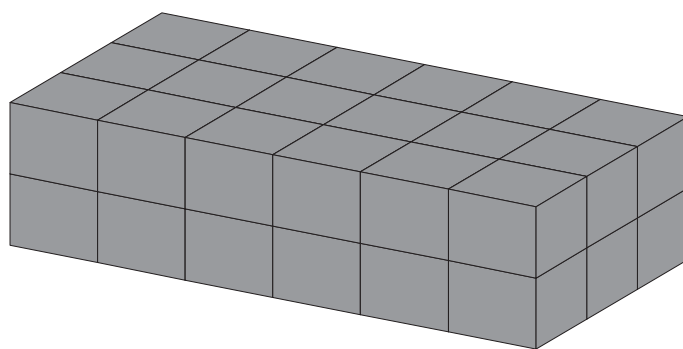


Figure 13-40: Example for the QU card from demo_QU1.pre

13.41 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 (e.g. direct creation in PREFEKO with cards, but also import from NASTRAN, FEMAP, PATRAN etc.) 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.

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.

Parameters:

Remove all existing ...: Clear all previously defined remeshing rules (i.e. 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.

Start a new remeshing rule: Set a new remeshing option (previously read RM cards will be discarded).

Add a remeshing rule: Add a remeshing rule to the already defined ones (i.e. 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 polygonal 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 e.g. 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 (i.e. triangles, wires, cuboidal volume elements etc.) 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.

When the actual remeshing is done, then for each created or imported mesh element (e.g. a triangular surface patch element) PREFEKO internally loops over all the RM cards which are active at this time, and determines the local mesh length for each RM card, and the smallest of these is then used for the actual mesh refinement of the specific mesh element under consideration.

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

$$s = s_1 + \frac{s_2 - s_1}{d_2 - d_1} (d - d_1) . \quad (13-3)$$

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, i.e. 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 (i.e. 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 13-41 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 13-42 a local mesh refinement with respect to two lines is used (i.e. two simultaneously active RM cards).

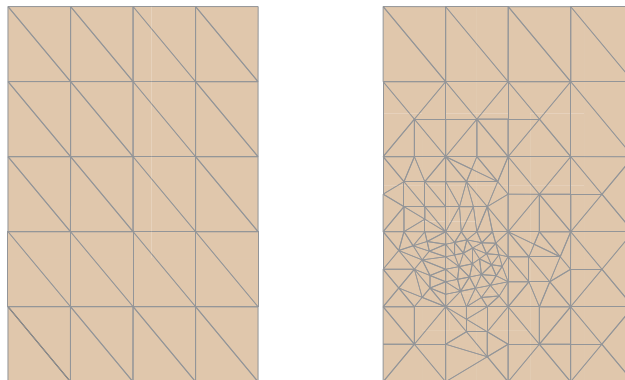


Figure 13-41: Example of local mesh refinement with respect to a point from demo_RM3.pre

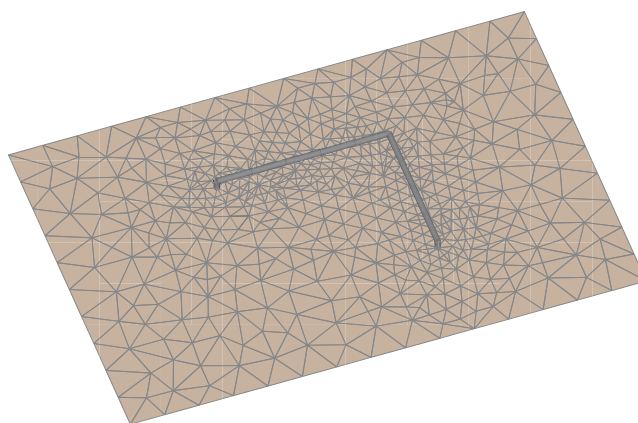


Figure 13-42: Example of local mesh refinement with respect to lines demo_RM5.pre

13.42 SF card

With this card the scaling of the geometric data is possible. This is useful for specifying models in a convenient unit (e.g. cm) and specifying a scaling factor once, since internally FEKO uses all dimension related values in metres.

SF - Scale all dimensions

☒ Modify all dimension related values

Multiply dimensions with

Note: Normally the "Modify all dimension related values" box should be checked. This box may be unchecked for compatibility with previous input files - in case certain values, such as source and field positions are not scaled. In both cases, all structural dimensions will be scaled (including coatings and dielectric layers).

Parameters:

Modify all dimension related values: If this item is checked all geometrical dimensions are scaled. If unchecked some coordinate values are not 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
- Radii of the all the layers when the Green's function for a homogeneous or layered dielectric sphere is used (see section [14.39](#))
- Thickness of the layers when the Green's function for a planar, multilayered substrate is used (see section [14.39](#))
- 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 (see section [14.59](#))
- Radius and thickness of a wire coating (see section [14.29](#))
- Coordinates of wedges and edges in the PO region

- Coordinates of the Fock region
- Transmission line length and end point coordinates (see section [14.61](#))
- The dimensions of the aperture used in the AP card and the amplitudes of the A5 and A6 dipoles which depend on the incremental areas.
- The variable *Maximum identical distance*, specified with the EG card (see section [13.13](#)), which controls whether two points are considered to lie at the same point in space

If it is checked all geometrical dimensions and coordinates are scaled. This includes, in addition to the parameters listed above, the following:

- Named points defined by the the DP card.
- The coordinates of the source point specified in the excitation cards A1, A2, A3, A4, 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.
- Radius of the approximated connecting segment in the A4 and L4 cards.
- Positions where the near field is calculated with the FE card.
- Offset in the near field calculation (see section [14.51](#)).

Note that if, for example, all data is specified in mm with the scaling set to 0.001, all input values are interpreted as mm. This also applies to the segmentation parameters (IP card) (see section [13.23](#)) and possible translations (TG card) (see section [13.44](#)).

13.43 SY card

Here symmetry can be used to generate the geometry and to reduce computation time.

Parameters:

Select symmetry for the plane $x = 0$: The type of symmetry, if any, in the yz -plane.

Select symmetry for the plane $y = 0$: The type of symmetry, if any, in the xz -plane.

Select symmetry for the plane $z = 0$: The type of symmetry, if any, in the xy -plane.

Label increment for the new structures: After they are mirrored, the labels of the new elements are incremented with the value specified in this field. Label 0 is, however, not incremented. The corresponding new elements will also have label 0. If this field is empty or set to 0, the labels are not incremented, i.e. the new elements will have the same label as the one 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. Furthermore, 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 once. A detailed description of the different types of symmetry (geometric, electric and magnetic symmetry) is given in FEKO User Manual (section [24.1](#)).

13.44 TG card

With this command the already entered geometric elements (triangles, segments, etc.) can be translated, rotated, mirrored and/or scaled. It is also possible to duplicate structures.

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: 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 (see section 12.4)). 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 α_x around the x axis in degrees.

Rotation around the y-axis: Angle of rotation α_y around the y axis in degrees.

Rotation around the z-axis: Angle of rotation α_z around the z axis in degrees.

Translation along the x-axis: Translation Δ_x in the x direction in m (scaled by SF card).

Translation along the y-axis: Translation Δ_y in the y direction in m (scaled by SF card).

Translation along the z-axis: Translation Δ_z in the z direction in m (scaled by SF card).

Mirror about 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 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 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, i.e. 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

$$\begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix} = \gamma M \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \gamma \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \end{bmatrix} \quad (13-4)$$

with the rotation matrix

$$M = \begin{bmatrix} \cos \alpha_y \cos \alpha_z & -\cos \alpha_y \sin \alpha_z & \sin \alpha_y \\ \cos \alpha_x \sin \alpha_z + \sin \alpha_x \sin \alpha_y \cos \alpha_z & \cos \alpha_x \cos \alpha_z - \sin \alpha_x \sin \alpha_y \sin \alpha_z & -\sin \alpha_x \cos \alpha_y \\ \sin \alpha_x \sin \alpha_z - \cos \alpha_x \sin \alpha_y \cos \alpha_z & \sin \alpha_x \cos \alpha_z + \cos \alpha_x \sin \alpha_y \sin \alpha_z & \cos \alpha_x \cos \alpha_y \end{bmatrix} \quad (13-5)$$

Multiplication by the rotation matrix M effectively rotates a point first by an angle α_z around the z axis, then by an angle α_y around the y axis and finally by an angle α_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 α_x around the x axis, then rotating α_y around the new y' axis and finally rotating α_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 (i.e. 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 some 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))

** Some 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.

13.45 TO card

Using this card a surface mesh in the form of a toroidal segment can be generated.

TO - Specify a torus section

S1

S2

S3

S4

The angle φ (degrees)

The angle α (degrees)

Max edge length, φ direction

Max edge length, α direction

Scale second half axis with

Normal vector directed

☒ Outward ☐ Inward

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 : The angle of rotation around the axis S1–S2.

The angle α : 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: 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 card usage

The toroidal segment, which is shown in Figure 13-43, 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 13-44.

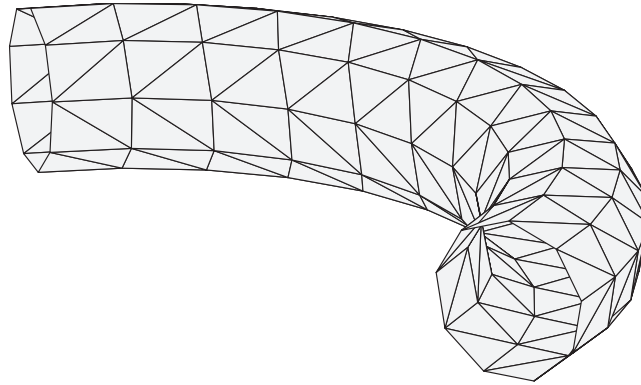


Figure 13-43: Example for the TO card from demo_T01.pre

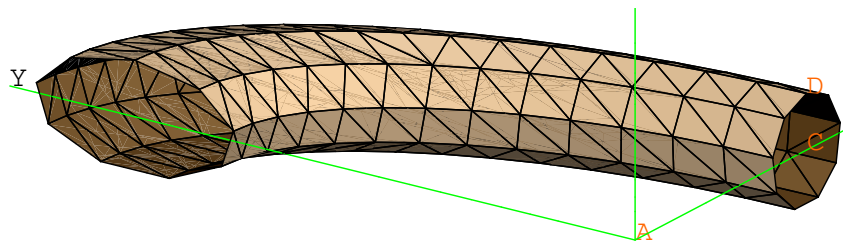


Figure 13-44: Example for the TO card with an elliptical cross section from demo_T02.pre. Note that it is stretched in the direction of the y axis, i.e. it is elliptical in the φ -plane. It is also elliptical in the α -plane when $\varphi=90^\circ$, but it is circular in the α -plane when $\varphi=0^\circ$.

13.46 TP card

With this card points (previously defined with the DP card) can be translated, rotated and/or scaled (relative to the origin).

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 (see section 12.4). 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 α_x around the x axis in degrees.

Rotation around the y-axis: Angle of rotation α_y around the y axis in degrees.

Rotation around the z-axis: Angle of rotation α_z around the z axis in degrees.

Translation along the x-axis: Translation Δ_x in the x direction. (All three translation distances are affected by the scaling factor set with the SF card.)

Translation along the y-axis: Translation Δ_y in the y direction.

Translation along the z-axis: Translation Δ_z in the z direction.

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: 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 $\gamma=1$).

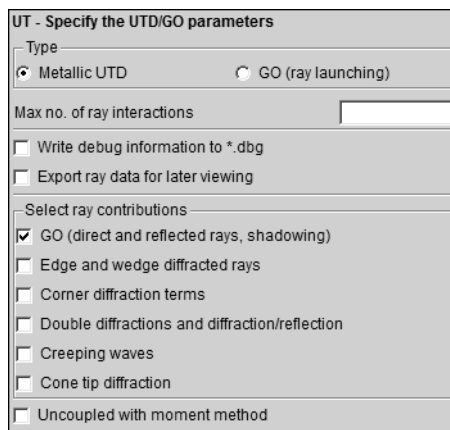
If a point is rotated around more than one axis with a single card, it is rotated first by an angle α_z around z axis, then by α_y around the y axis and finally by α_x around the x axis. A more detailed description of the transformation can be found in the description of the TG card (see section [13.44](#)).

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.

13.47 UT card

With this command the parameters for the Uniform geometric theory of diffraction (UTD) for polygonal plates and Geometrical optics (ray launching) (GO) are defined.

Uniform geometric theory of diffraction (UTD)



Parameters:

Type: The method (UTD/GO) that is to be applied can be chosen.

Max no. of ray interactions: This parameter gives the maximal number of ray-interactions (i.e. 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. This file contains large amounts of information and should only be used when debugging.

Export ray data for later viewing: When this item is checked the ray information is exported to the *.bof and to a special *.ray file, so that the ray paths can be displayed in POSTFEKO. The ray information can become very large, and thus it should only be exported if specific ray paths are to be examined.

The following abbreviations are used in the *.ray file and POSTFEKO:

- B:** Diffraction at an edge
- D:** Diffraction at a corner (of a wedge)
- E:** Diffraction at a corner (of an edge)
- K:** Diffraction at a wedge
- Q:** Source point
- R:** Reflection
- S:** Observation point
- C:** Creeping wave
- T:** Tip diffraction

Select ray contributions: Determines which ray contributions to take into account:

- *GO (direct and reflected rays, shadowing):* Geometrical optics (ray launching) (GO), i.e. direct and reflected rays and shadow regions are taken into account.
- *Edge and wedge diffracted rays:* 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 — must not be larger than specified in the *Max no. of ray interactions* field.)
- *Corner diffraction terms:* Corner diffraction.
- *Double diffractions and diffraction/reflection:* 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 are not close together.

Increasing the type and number of ray interactions increases accuracy but computation time as well. 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 perfectly conducting flat polygonal plates or a single cylinder allowed in the UTD region.
- No UTD and PO at the same time.

Geometrical optics (ray launching)(GO)

Parameters:

Type: The method (UTD/GO) that is to be applied can be chosen

Use geometrical optics (ray launching) on all surfaces with label: The label to which the geometrical optics (ray launching) should be applied

(optionally) up to label: The label up to which the GO should be applied. 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 (i.e. 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. This file contains large amounts of information and should only be used when debugging.

Export ray data for later viewing: When this item is checked the ray information is exported to the *.bof and to a special *.ray file, so that the ray paths can be displayed in POSTFEKO. The ray information can become very large, and thus it should only be exported if specific ray paths are to be examined.

Select ray contributions: Determines which ray contributions to take into account:

- *GO (direct and refl. rays, shadowing, diel. trans):* direct, reflected and transmitted rays are taken into account.

Ray launching settings: The specific settings for the ray launching used in the geometrical optics (ray launching) can be set here.

- *Automatic determination:* The ray launching parameters will automatically be determined during the solution process
- *Manually set angular / spatial increments* The angular / spatial increment angles for the ray launching are manually specified by the user.

Manually set angular / spatial increments

Angular increment (spherical ray launching): *Theta-direction (degrees):* Specific Theta increment angles can be specified for the ray launching procedure in the geometrical optics (ray launching) solution.

Phi-direction (degrees): Specific Phi increment angles can be specified from the ray launching procedure in the geometrical optics (ray launching) solution.

Spatial increment (parallel ray front): *u-direction (metres):* This parameter sets the distance between the individual rays in the parallel ray front in the u-direction.

v-direction (metres): This parameter sets the distance between the individual rays in the parallel ray front in the v-direction.

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 are not close together.

The main advantages of ray-launching GO:

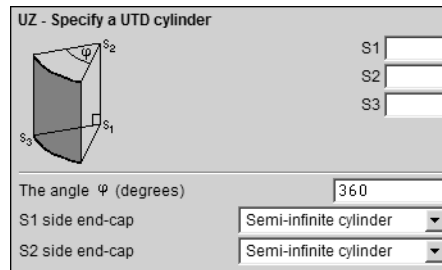
- Suitable for large arbitrarily shaped objects (also curved).
- Dielectrics may be modelled with the ray-launching GO method.
- Support of a plane wave as a excitation (required for the calculation of far field RCS).
- Multiple reflections more efficient than UTD or PO.

When no UT card is used, the following default values apply:

- *Max no. of ray interactions:* 3.
- *Write debug information to *.dbg:* unchecked.
- *Export UTD ray data for later viewing:* unchecked.
- *Select ray contributions includes.*
 - *GO (direct and reflected rays, shadowing).*
 - *Edge and wedge diffracted rays.*
 - *Corner diffraction terms.*

13.48 UZ card

With this card a cylinder is created for the UTD region.



Parameters:

- S1:* The start point of the cylinder axis.
- S2:* The end point of the cylinder axis.
- S3:* A point on the radius of the cylinder, the angle $S2-S1-S3$ must be 90° .
- The angle φ :* The angle of the cylinder segment. Currently only an angle of 360° is supported.
- 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.

The cylinder in Figure 13-45 was created using a UZ card. Note the absence of discretisation.

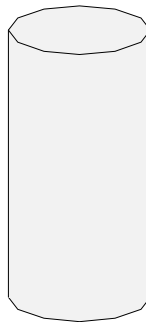


Figure 13-45: Example for UZ card from demo_UZ1.pre

13.49 VS card

This card specifies known visibility information (required when using physical optics with multiple reflections) to reduce the time required to calculate it.

To accurately compute multiple reflections the code 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 inform the code that certain triangles are hidden from each other and others are visible to 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 the field *triangles with label*). If checked, the card applies to all triangles with labels in the range from the value specified in the *triangles with label* field to that in the *to triangles with label* field.

Note that visibility is reciprocal, i.e. 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.

Consider the structure shown in Figure 13-46 consisting of four flat plates and a cylindrical section. The two plates lying at 45 degrees to the coordinate system (labelled 1 and 3), are half as wide as the plates with labels 0 and 2. Thus some triangles with label 2 are visible to some triangles with label 0, but not all.

We have to specify which triangles are visible / hidden from *all* triangles with label 0 first, then those visible from label 1 and so on. The VS cards for this example would be as follows:

- *Triangles labelled 0 are not visible from triangles with label 0.*

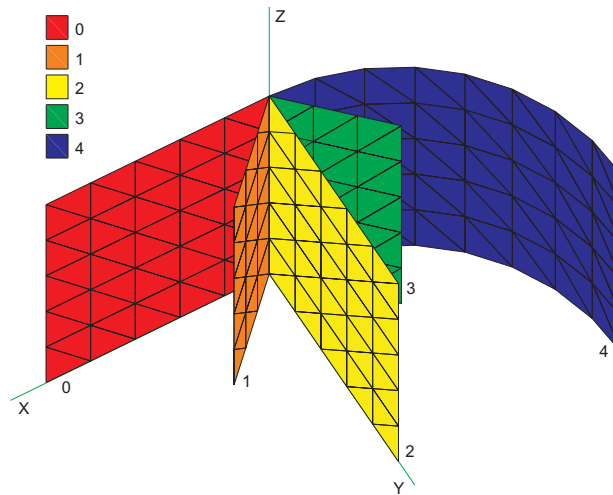


Figure 13-46: Structure used to demonstrate the use of VS cards from demo_VS1.pre

- 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.

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 some triangles with label 2 are visible from some triangles with label 0 while others are hidden, we cannot specify any information for this combination of layers. However, the plate with label 2 shadows all triangles with labels 3 and 4 and we may specify that these are hidden. This is done with the third VS card. Note that this card specifies a range of hidden labels.

Next we must 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 we have already specified the visibility between labels 0 and 1, we do not specify it 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,

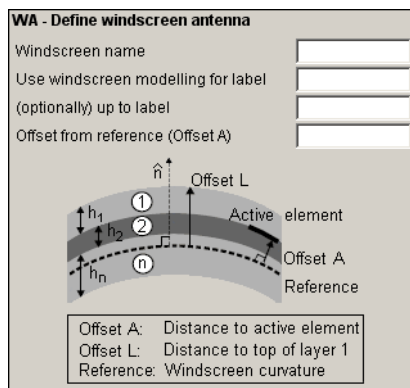
Next we look at label 2. As before we need not consider labels lower than 2. Also the label is hidden from itself as indicated by VS card number seven. Next we state that label 3 is visible, but we cannot specify anything about label 4 as only some of these triangles will be visible.

Similarly VS cards 9 and 10 states that label 3 is not visible to itself and fully visible to label 4.

Finally we must 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 some 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 that we cannot specify any information for label 4.

13.50 WA card

Use this card to define all active windscreen antenna 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).



Note that the the three cards WR (see section 13.52), WA (see section 13.50) and WD (see section 14.63) should be used together to create windscreen antenna models. These three cards respectively define the windscreen reference surface, windscreen active 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.

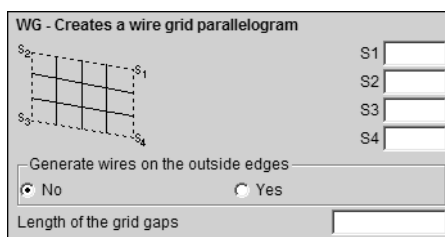
(optionally) up to label: End of the label range (optionally).

Offset from reference (Offset A): Offset of the specified label geometry w.r.t. the reference windscreen triangles.

The antenna elements will be limited to laying tangentially w.r.t. 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.

13.51 WG card

With this card a wire grid in the shape of a parallelogram can be generated.



Parameters:

S_1, S_2, S_3, S_4 : The four corners of the parallelogram in consecutive order.

Generate wires on the outside edges: If the *No* item is selected, only the wires inside the parallelogram are generated whereas if the *Yes* item is selected all the wires are generated. This option is important when two adjacent parallelograms are generated, as the segments along the sides must not be 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.

Example of WG card usage:

The wire grid seen in Figure 13-47 is generated using the WG card.

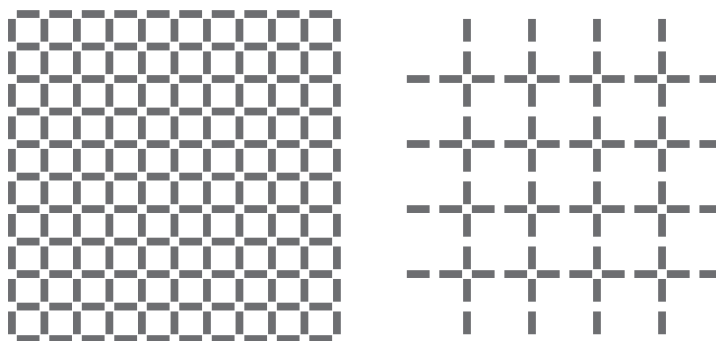


Figure 13-47: Examples of the WG card from demo_WG1.pre — the one on the right has a value of 2 in *Length of the grid gaps*.

13.52 WR card

Use this card to define the dielectric windscreen reference plane. Geometrically this surface is not part of the EM model and is used simply to determine the curvature factor between the two elements on the windscreen.

WR - Define windscreen reference	
Windscreen name	
Use as reference all surfaces with label	
(optionally) up to label	

Note that the the three cards WR (see section [13.52](#)), WA (see section [13.50](#)) and WD (see section [14.63](#)) should be used together to create windscreen antenna models. These three cards respectively define the windscreen reference surface, windscreen active elements (antenna) and the windscreen layered media definition.

Parameters:

Windscreen name: The name of the windscreen.

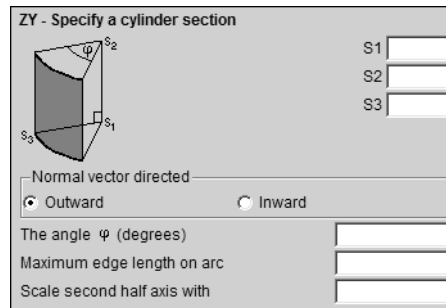
Use as reference all surfaces with label: The start of the label range.

(optionally) up to label: The end of the label range (optionally).

The windscreen reference triangles are defined by label and since this plane also forms the zero reference w.r.t. the defined windscreen antenna elements and glass layers, it should be noted that they should all have their normals pointing in the same direction.

13.53 ZY card

With this card a surface mesh in the form of a cylindrical segment can be generated.



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 : 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: 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 S1–S3. It is not recommended to generate elliptical cylinder 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 (i.e. the lines S1–S2 and S1–S3 are perpendicular), the segmented area (shaded in the figure of the cylinder) is obtained by rotating the point S3 around the axis S1–S2 through φ . For $\varphi=360^\circ$ a full cylinder is obtained.

An oblique cylinder (i.e. the circular or elliptical rim is not perpendicular to the axis) can also be created. Then S1–S2 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 S1, S2, S3, and parallel to the line S1–S3.

Examples of ZY card usage:

The cylindrical section (Figure 13-48), elliptical cylinder (Figure 13-49) and non-orthogonal cylinder (Figure 13-50) below were all created with ZY cards.

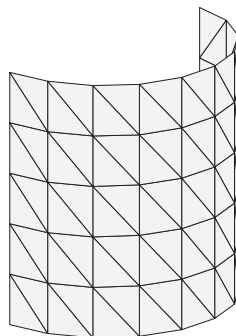


Figure 13-48: Example for the ZY card from demo_ZY1.pre

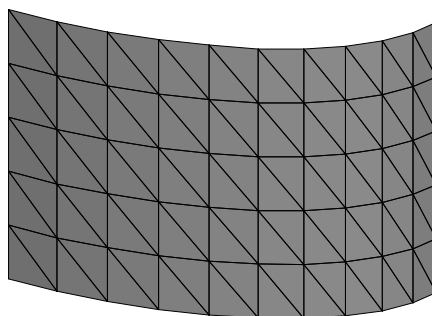


Figure 13-49: Example of an elliptical cylinder from demo_ZY2.pre

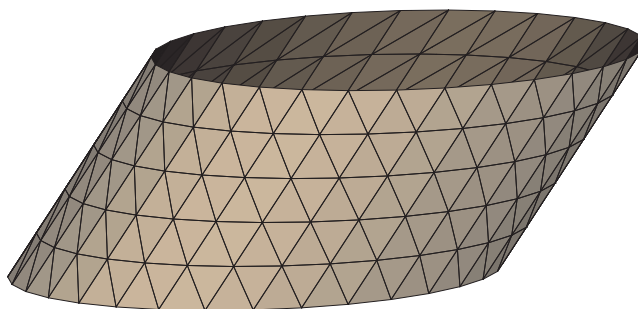


Figure 13-50: Example of a non-orthogonal elliptical cylinder from demo_ZY3.pre

14 Control cards

The following table summarises all the PREFEKO and FEKO control cards. These cards should not be used in the geometry section of the *.pre input file, i.e. they should only be used below the EG card. The control cards are used to specify, for example, the frequency and the type of excitation. They also determine the required calculations (such as the locations for near field calculations and the directions for far field calculations, etc.).

Card	Description
**	characters used to indicate a comment line
Ax	type of excitation (e.g. an incident plane wave or a voltage source)
AB	creates a modal port excitation
B0	through the use of the reflection factor coefficient a ground plane can be inserted
CA	defines a cable path section for the cable irradiation computation.
CD	defines a specific cable cross-section (single, coaxial, ribbon and bundle/multi-cable)
CF	set the type of integral equation for perfectly conducting metallic surfaces
CG	the algorithm used to solve the matrix equation is selected
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
CS	defines a cable path section and the centre/reference location to which a cable cross-section is applied
DA	creates additional files for the results
DI	enters the properties of the dielectric medium
DL	defines a layered medium
EE	calculates the error estimates
EN	indicates the end of the input file
FE	calculates the near fields
FF	calculates the far fields
FR	sets the frequencies at which the calculations are to be carried out
GF	sets the Green's functions
L2	defines a complex load on a vertex
L4	adds a load between a metallic triangle and the ground plane for the planar multilayer Green's function
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
LZ	defines a complex load on a segment
NW	define a linear non-radiating network
OF	offset i.e. displacement of the origin when calculating the near fields
OS	saves the surface currents in a file
PP	defines the phase for periodic boundary condition calculation
PS	sets general control parameters
PW	defines the radiating power of a transmitting antenna
RA	defines an ideal receiving antenna
SA	used to calculate SAR in dielectric media
SH	used to define solid and braided cable shields
SK	takes a 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	calculate reflection and transmission coefficients for an incident plane wave on a planar structure
WD	defines the dielectric properties of the windscreen glass layers

As mentioned above the control cards form the second part of the input file (see section 12.2). Control cards are processed line by line and only affect other cards and calculations specified below them in the input file. (Information specified in a control card is not available before that line is processed.) Any number of control cards can be used, but they should adhere to a basic sequence. Thus, for example, the frequency (FR card) and the type of excitation (Ax card) must be defined before the near fields can be calculated with an FE card.

In addition, a sensible order for the control cards can result in a considerable reduction in the computation time. For example, for an SK card the whole matrix has to be recalculated, while an Ax card only redefines the right hand side of the matrix. A summary of the dependencies is given in the table below: There are also other dependencies. If the matrix elements are recomputed

Action	For the cards
recalculates matrix elements	BO, CF, CO, DI, FR, GF, L4, LD, LE, LP, LS, LZ, SK, SP, TL
recalculates the right hand side	Ax, BO, DI, FR, GF
resolves the matrix equation	CG

then the matrix equation has to be solved again. The actual calculation is started by the CM, FE, FE, OS, RA, SA and SP cards. All other cards are read and the data stored.

When solving for a number of frequencies all the control cards following the FR card (up to, but excluding, the next FR card or EN card) are read into a buffer. For each frequency all these cards are read and processed. The computation time can be reduced significantly by using the cards in the correct order. If, for example, a structure needs to be investigated at two frequencies and

with two different excitations then the control cards can be organised in either of the following ways:

...

FR *FR card for the two frequencies*

Ax *first excitation*

...

Ax *second excitation*

...

EN *end of the input file*

or

...

FR *FR card for the two frequencies*

Ax *first excitation*

...

FR *FR card for the two frequencies*

Ax *second excitation*

...

EN *end of the input file*

In the first format, the two excitations are processed one after the other for each frequency. (The cards are executed in the order FR Ax ... Ax ... FR Ax ... Ax ... EN, where the second FR indicates execution of the frequency loop for the second specified frequency.) In the second format the first excitation is treated for both frequencies before treating the second excitation for both frequencies. (Here the cards are executed as FR Ax ... FR Ax ... FR Ax ... FR Ax ... EN.) For the computational requirements, one finds that in the first case the matrix elements have to be calculated twice (it has to be completely recalculated each time the frequency is changed) and in the second case four times.

The computation time is not only influenced by the control cards, but also by what has to be solved for. In the following example, the structure has to be solved at a number of frequencies and for the ideal (conducting) and non-ideal (conduction with losses) cases:

...

FR *FR card for multiple frequencies*

FE *calculation of the near fields*

SK *include skin effect*

FE *calculation of the near fields*

EN *end of the input file*

The three control cards FE, SK and FE are written to the buffer and are worked through in the loop for the different frequencies. At the first frequency the FE card initiates the field calculation. Because a SK card has not yet been read, ideal conductivity is assumed. Then the SK card is read and losses are taken into account when the second FE card is run. Thus the first frequency pass is finished. At the next frequency pass the cards FE, SK, and FE are read again, but the losses from the SK card are still active from the first pass. The SK card is thus useless and the two FE cards calculate the same things twice.

The correct input order is:

...

FR *FR card for multiple frequencies*

SK *skin effect switched off*

FE *calculation of the near fields*

SK *skin effect switched on*

FE *calculation of the near fields*

EN *end of the input file*

Then the four cards SK, FE, SK and FE are calculated for all the frequencies.

14.1 ** card

The comment lines (see section 13.1) can be used to add comment lines before or after the EG card.

The ‘**’ indicator may also be used to name the sources, loads, cables, transmission lines and a number of solution requests (e.g. near fields, far fields, currents, SAR, receiving antennas) after the EG card.

These names are then used by the FEKO kernel during the solution (e.g. for the screen process output or in the *.out file), but are also used for post-processing purposes.

Such names are also critical for any solution entities that are to be used or referred to in the optimisation setup (see section 7), as these labels are used to uniquely identify specific data in the output of the FEKO solution for consideration during the optimisation goal evaluation process.

The names of specific cards are set by adding a ** comment at the end of the line in which the card is defined (for multi-line cards, this comment should be added at the end of the first line of the card).

For example, with the card sequence

```
...  
A1  0    Feed                1      0    ** Top right array element  
  
FE   ....                  ** Cut plane 1 (through left mast)  
FE   ....                  ** Cut plane 2 (vertical through ground)
```

A voltage source is defined with the name Top right array element and two near field computation requests with the names Cut plane 1 (through left mast) and Cut plane 2 (vertical through ground). These names can then be referred to in the optimisation goal definitions (see section 7.1.4) and are used in POSTFEKO and in the FEKO *.out file.

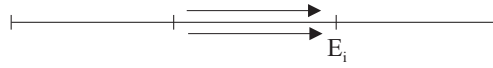
14.2 Ax Cards

This card defines the type of excitation as well as other parameters regarding the excitation. The following possibilities are available:

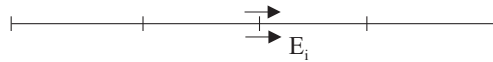
Card	Type of Excitation
A0	A linear polarised plane wave incident on the structure.
A1	Excitation by means of a voltage gap on a segment (i.e. impressed electric field strength along a segment).
A2	Excitation by means of a voltage gap at a node i.e. between two segments.
A3	Excitation by means of a magnetic ring current (TEM-frill) on a segment. Thus a coaxial feed can be modelled.
A4	Special vertical pin excitation, e.g. for a patch antenna on a planar substrate with a ground plane (coaxial probe excitation mode).
A5	An electric Hertzian dipole is used as excitation. The position and orientation in the space are arbitrary.
A6	A magnetic Hertzian dipole is used as excitation. The position and orientation in the space are arbitrary.
A7	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 (e.g. CRIPTE or CableMod) or by a PCB simulation tool (e.g. 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.
AK	Excitation by means of a voltage source connected to a radiating cable.
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.
AS	Excitation by means of impressed radiating spherical modes.
AV	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.

The different ways to realise a voltage source are summarised in Figures 14-1 and 14-2. The impressed electric field strength is indicated by $\vec{E}_i$.

A1 card: Voltage source on a segment



A2 card: Voltage source on a node between segments



A3 card: TEM-frill on a segment

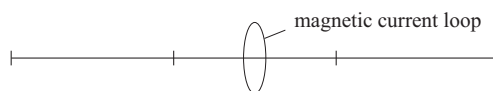
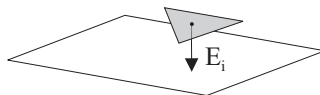
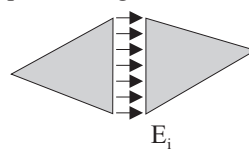


Figure 14-1: Possible ways to realise a voltage source on a wire segment

A4 card: Vertical pin approximation (dielectric substrate)



A7 card: Voltage gap on an edge



AE card: Voltage gap along edges

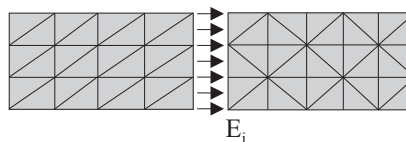


Figure 14-2: Possible ways to realise a voltage source in connection with triangles

More than one excitation is also allowed. Thus one may, for example, 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, A4 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* (EG card (see section [13.13](#))).

The excitations are described in detail in the following sections.

14.3 A0 card

This card realises excitation by a linearly polarised incident plane wave.

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 of the rotating options is selected, the *Ellipticity* must be specified below.
- Rotation about axis:* The rotation of the plane wave about the x, y 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 $\vec{E}_0$ of the incident field in $\frac{V}{m}$.
- Phase:* Phase of the field strength $\vec{E}_0$ of the incident field in degrees.
- Initial θ value:* Angle of incidence θ of the plane wave in degrees, see the figure in the card above.
- Initial φ value:* Angle of incidence φ of the plane wave in degrees, see the figure in the card above.
- Polarisation angle η :* See the figure in the card above. It is the angle — in a right-handed sense when viewing in the incident direction — from $-\hat{\theta}$ to $\vec{E}_0$.

<i>Increment in ϑ:</i>	If more than one direction of incidence is to be examined, ϑ is incremented by this value for each new angle of incidence.
<i>Increment in φ:</i>	If more than one direction of incidence is to be examined, φ is incremented by this value for each new angle of incidence.
<i>Ellipticity:</i>	This field is only applicable when elliptical rotation is selected under <i>Polarisation</i> above and gives the ellipticity of the rotating polarisation. It must be larger than 0 (linear polarisation) and smaller or equal to 1 (circular polarisation).

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 $\vec{E}_0$ are defined as indicated in the figure in the card above.

The electric field strength of the incident field is then given by

$$\vec{E}_i(\vec{r}) = [\vec{E}_0 + j\nu(\vec{E}_0 \times \hat{\beta}_0)] \cdot e^{-j\vec{\beta}_0 \cdot \vec{r}} \quad (14-1)$$

where ν 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

$$\vec{H}_i(\vec{r}) = \frac{1}{Z_F} \hat{\beta}_0 \times \vec{E}_i \quad (14-2)$$

with Z_F the wave impedance in the surrounding free space medium.

It should be noted that the incident power density (which is required, for example, for RCS computations) is given by

$$\vec{S}_i = \frac{1}{2} \frac{|\vec{E}_0|^2}{Z_F} (1 + \nu^2) . \quad (14-3)$$

It is possible to vary the direction of incidence. This is particularly useful when e.g. 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 e.g. the monostatic radar cross section is to be calculated for $\vartheta=90^\circ$ and $0^\circ \leq \varphi \leq 180^\circ$, the following command is used:

```

...
A0  0    1 181    1.0    0.0    90.0    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.

14.4 A1 card

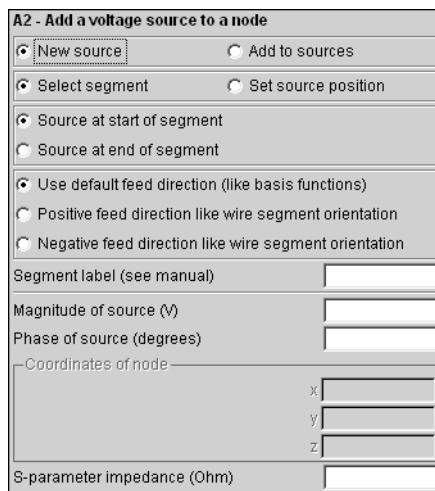
With this card a voltage source is placed on a segment.

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 segment:* When this item is selected, the *Apply source to last segment with label* field becomes active. Here one specifies the label of the segment to which the source must be applied. If more than one segment has this label, the source is applied to the last segment with this label. Alternatively, one may select the item *Set source position*, then 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 *Modify all dimension related values* is checked.
- 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.)
- Magnitude of source:* Magnitude of the voltage source in V.
- Phase of source:* Phase of the voltage source in degrees.
- S-parameter impedance:* The port impedance of this excitation is used in connection with S-parameter and realised gain calculations. If this field is empty or 0, the value specified at the SP card is used. (This value is only used if the S-parameters are requested with an SP card.)

14.5 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, etc.



A2 - Add a voltage source to a node

☒ New source ☐ Add to sources

☒ Select segment ☐ Set source position

☒ Source at start of segment
☐ Source at end of segment

☒ Use default feed direction (like basis functions)
☐ Positive feed direction like wire segment orientation
☐ Negative feed direction like wire segment orientation

Segment label (see manual)

Magnitude of source (V)

Phase of source (degrees)

Coordinates of node

x
y
z

S-parameter impedance (Ohm)

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 segment:* When this item is selected, then the *Source label* field becomes active. Here one specifies the label of the segment which shall be fed (either start or end point as determined by the corresponding check box). The excitation 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:* When this check box is activated, then 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 like the basis function defined over this node (see the discussion below under *Use default feed direction*).
- 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 like the basis function setup in FEKO. For wire / surface junctions (UTD plates, 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, then 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.

S-parameter impedance: The port impedance of this excitation is used in connection with S-parameter and realised gain calculations . If this field is empty or 0, the value specified at the SP card is used. (This value is only used if the S-parameters are requested with an SP card.)

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 that the vector direction of the feed is the direction of the current flow through the node. The internal EMF — electromagnetic force — of the impressed voltage is in the opposite direction.

14.6 A3 card

This card realises excitation by a magnetic ring current (TEM-frill) on a segment. This will give an accurate model of a coaxial feed, but requires both the inner and outer radii.

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 segment: When this item is selected, the *Apply source to last segment with label* field becomes active. Here one specifies 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. Alternatively, one may select the item *Set source position*, then the feed segment is determined by specifying the Cartesian coordinates in the *Segment centre* group. The position values are in m and are scaled by the SF card if *Modify all dimension related values* is checked.

Reverse feed orientation: By default, the vector of the excitation points in the direction of the segment - from the start point to the end point as the segment was created. When this option is checked, the orientation of the excitation is reversed.

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.

S-parameter impedance: The port impedance if this excitation is used in an S-parameter calculation. If this field is empty or 0, the value specified at the SP card is used. (This value is ignored if no SP card is used.)

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 and that the outer radius should be 2 to 3 times the size of the inner. If an impedance Z is desired, then the following relation can be used

$$Z \approx 60 \Omega \cdot \ln \left(\frac{\text{Outer conductor radius}}{\text{Inner conductor radius}} \right) \quad (14-4)$$

to determine the outer radius. For $Z = 50 \Omega$ the outer conductor radius should be equal to 2.3 times the inner conductor radius.

14.7 A4 card

This card creates a coaxial attachment feed approximation for use in connection with the Green's function for planar substrates with a metallic ground plane (GF card (see section 14.39)).

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:* When this item is selected, the *Source label* field becomes active. Here one specifies the label of the triangle to excite. The feed point is at the centroid of the triangle (see Figure 14-3). If there are more than one triangle with this label, the excitation is placed on the one with the highest element number. Alternatively, the user may select the item *Set source position*, and specify the Cartesian coordinates of the feed point in the *Coordinates of node*, group. (These values are in m and optionally scaled by the SF card.) FEKO will excite the triangle whose centroid is closest to the specified coordinate.
- Transform impedance to ground plane:* If unchecked the impedance is computed directly at the excitation point. If checked, an inductive approximation of the feed pin is used and the impedance is transformed to the ground plane.
- Magnitude of source:* Absolute value of the excitation current $|\vec{I}_0|$ in A. The positive current direction is the positive z direction.
- Phase of source:* Phase of the current $\vec{I}_0$ in degrees.
- Radius of the connection pin:* The radius of the coaxial probe feed pin in m. This value is optionally scaled by the SF card.
- S-parameter impedance:* The port impedance of this excitation is used in connection with S-parameter and realised gain calculations. If this field is empty or 0, the value specified at the SP card is used. (This value is only used if the S-parameters are requested with an SP card.)

An example of the excitation is shown in Figure 14-3. A typical application of the A4 card is given in `example_30b.pre` (*Scripting examples*). It is of course possible to discretise the vertical pin (into segments) and feed one of the segments with a voltage source (A1 card). The advantage of the A4 card is that there are no vertical currents, which results in substantially simpler Green's functions and a significant reduction in computing time.

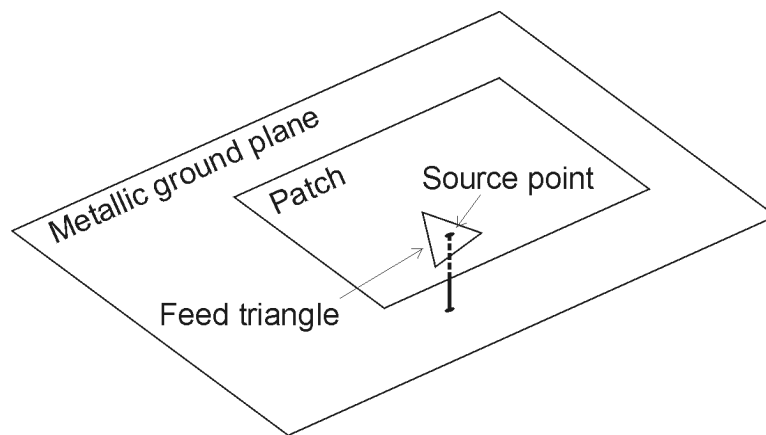


Figure 14-3: Excitation of a patch antenna with a vertical pin

14.8 A5 card

This card specifies excitation by an electric Hertzian dipole.

Parameters:

<i>New source:</i>	A new excitation is defined which replaces all previously defined excitations.
<i>Add to sources:</i>	A new excitation is defined which is added to the previously defined excitations.
<i>Magnitude of $I dl$:</i>	Absolute value of the complex amplitude $I \cdot l$ in Am.
<i>Phase of $I dl$:</i>	Phase of the complex amplitude $I \cdot l$ in degrees.
<i>x, y, z position:</i>	Coordinates of the position of the dipole in m. These values are optionally scaled by the SF card.
<i>ϑ angle:</i>	Orientation of the dipole in space — ϑ (in degrees) is the angle between the dipole and the z axis.
<i>φ angle:</i>	Orientation of the dipole in space — φ (in degrees) is the angle between the projection of the dipole onto the plane $z = 0$ and the x axis.

The dipole moment of the electric dipole is given by

$$p = \frac{I l}{j\omega} \quad (14-5)$$

The power radiated by the dipole in a free space environment is given by

$$P = \frac{\beta_0^2 \cdot Z_{F0} \cdot |I l|^2}{12\pi} = \frac{\omega^2 \mu_0^2 |I l|^2}{12\pi Z_{F0}} \quad (14-6)$$

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.

14.9 A6 card

This card specifies excitation by an elementary magnetic Hertzian dipole.

Parameters:

<i>New source:</i>	A new excitation is defined which replaces all previously defined excitations.
<i>Add to sources:</i>	A new excitation is defined which is added to the previously defined excitations.
<i>Electric ring current:</i>	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).
<i>Magnetic line current:</i>	Use the model of a magnetic line current (the moment $m = \frac{I_m l}{j\omega}$ where I_m is magnetic line current and l the length of the dipole).
<i>Magnitude of ...:</i>	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 .
<i>Phase:</i>	Phase of the complex amplitude in degrees.
<i>x, y, z position:</i>	Coordinates of the position of the dipole in m. These values are optionally scaled by the SF card.
<i>θ angle:</i>	The angle between the dipole and the z axis in degrees.
<i>φ angle:</i>	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

$$P = \frac{\beta_0^4 \cdot Z_{F0} \cdot |IA|^2}{12\pi} \quad (14-7)$$

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 aperture approximation — see the AP card) 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 $\vec{A}$, while the magnetic dipole model results in an electric vector potential $\vec{F}$ as well as a magnetic scalar potential Ψ .

14.10 A7 card

This card is used to specify a voltage source on an edge between two triangles or at a connection edge between a single triangle and a PEC ground plane or UTD plate. The AE card is substantially simpler to use and should be used for all new models. (The A7 card is supported only for compatibility with FEKO input files that were created before the AE card became available.)

Parameters:

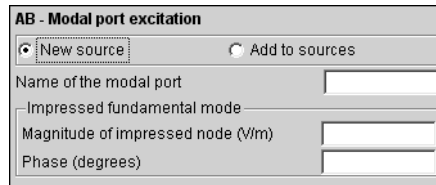
- New source:* A new excitation is defined which replaces all previously defined excitations.
- Add to sources:* A new excitation is defined which is added to the previously defined excitations.
- Select segment:* If this item is selected, a triangle with label specified in the *Element label* field is searched for. The excitation is placed on the edge that lies opposite to the first corner of the triangle. Once again the label must be unambiguous, i.e. if possible only one triangle must have this label. If there is more than one triangle with this label then only one will be fed. Alternatively, when selecting the item, *Set source position*, the feed edge is determined by specifying its Cartesian coordinates in the *Coordinates of edge centre*. These coordinates are in m and optionally scaled by the SF card. The edge must lie between two triangles or between a triangle and a ground plane or UTD plate.
- Magnitude of source:* Absolute value of the voltage source in V.
- Phase of source:* Phase of the voltage source in degrees.

If two triangles are connected to the edge, the basis function between these triangles is excited. The vector direction of the voltage source lies 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 certain special cases there may be only one triangle connected to the edge. If the edge lies 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.

14.11 AB card

This card is used to create a FEM modal excitation. The modal port will be excited with the fundamental mode of the associated, infinitely long guided wave structure of the modal port. If no excitation is defined, then the modal port will act as a passive port (sink) for fields incident on the port.

The image shows a dialog box titled "AB - Modal port excitation". It contains two radio buttons at the top: "New source" (which is selected) and "Add to sources". Below these are three input fields: "Name of the modal port", "Impressed fundamental mode", and "Phase (degrees)". The "Impressed fundamental mode" field is currently empty, while the others contain some text.

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.

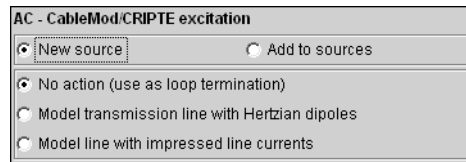
Name of the modal port: Label of the modal port.

Magnitude of source: Magnitude of the impressed node in V/m.

Phase (degrees): Phase of the impressed node in degrees.

14.12 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. Such a *.rsd file is created, for example, by the transmission line simulation programs CableMod or CRIPTE or by the PCB code PCBMod or by a current export with the OS card in FEKO. 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: 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 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 surface. (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. (See the description in these 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 x, y, or z direction. Standard FEKO units are used for these offsets (i.e. 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 degree, and the same definition as also used at the AR (see section 14.19) or TG (see section 13.44) cards applies.

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 at 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 at 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, ...) 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* 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 may look as follows

```
AC ...    ** Read the *.rsd file
FE ...    ** Calculate the near field
EN        ** End
```

However, if one wants to analyse, for example, a metal plate, 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

```
** 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
AO ...    ** Specify plane wave excitation
FE ...    ** Calculate the near field
OS ...    ** Output the currents
** End
EN
```

14.13 AE card

This card specifies an excitation at an edge between triangular surface elements similar to the A7 card, but the AE card has the advantage that the location of the feed point and the positive feed direction are substantially easier to specify. In addition it is possible to specify a feed edge which contains a number of triangle edges as shown in Figure 14-4.

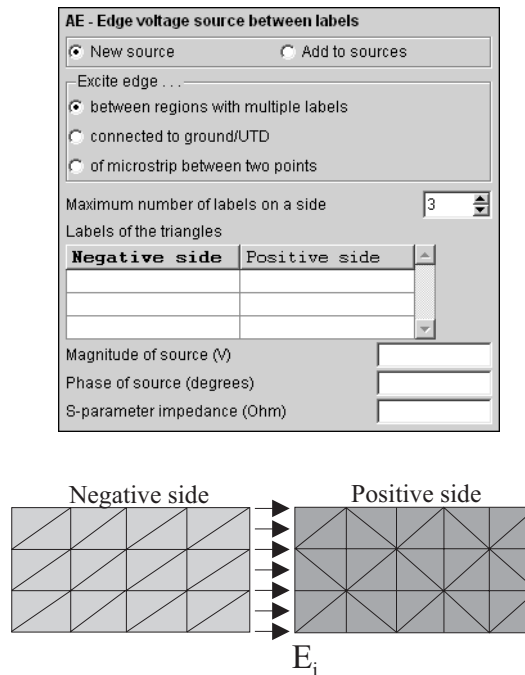


Figure 14-4: Example of the use of the AE card.

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:* This specifies how the edge is determined:
- *between regions with multiple labels:* 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* side (see Figure 14-4).
 - *connected to ground/UTD:* 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).
 - *of microstrip between two points:* 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.

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: Magnitude value of the voltage source in V.

Phase of source: Phase of the voltage source in degrees.

S-parameter impedance: The reference impedance to use for this excitation it is included in an S-parameter calculation requested by the SP card. If this field is empty or zero, the impedance specified at the SP card is used. (This value is only used if an SP card is applied to this source.)

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. One may, for example, 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 item, *between regions with multiple labels*, is selected, more than two surfaces may be connected to a feed edge. Examples for this are given in Figure 14-5 (one surface fed against three others), Figure 14-6 (two surfaces fed against two other surfaces), and in Figure 14-7 (permitted feed scenarios as indicated in the caption of this figure).

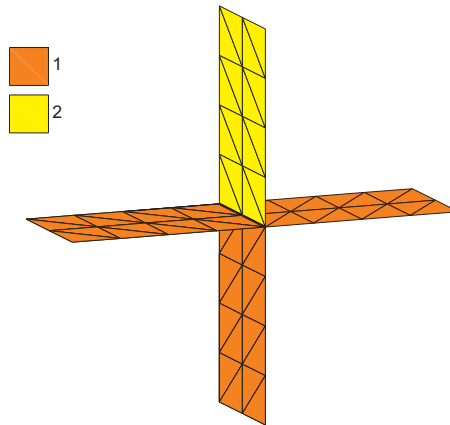


Figure 14-5: Example of a feed edge where more than one surface is connected on one side of the feed

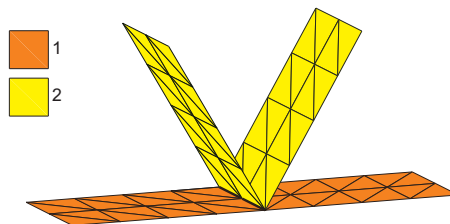


Figure 14-6: Example of a feed edge where more than one surface is connected on both sides (negative and positive) of the feed

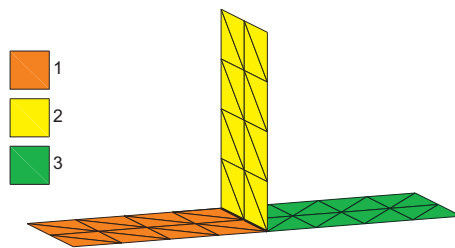


Figure 14-7: Example of a feed edge of three surfaces with different labels, where for instance label 2 could be fed against labels 1 and 3, but also label 1 could be fed against 2 and 3, or 3 could be fed against 1 and 2.

14.14 AF card

With this card an uniform electric current filament is impressed between two arbitrary points inside of the FEM region (i.e. it does not have to coincide with tetrahedral edges). This can be used to excite for instance a patch antenna. See also the example `DRA_finite` in the *Examples Guide*.

	Start	End
x coordinate		
y coordinate		
z coordinate		

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 of source: Amplitude in A of the impressed line current.

Phase of source: 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.)

S-parameter impedance (Ohm): The port impedance of this excitation is used in connection with S-parameter and realised gain calculations. If this field is empty or 0, the value specified at the SP card is used. (This value is only used if the S-parameters are requested with an SP card.)

The following restrictions apply when using the impressed current elements of 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, and this affects the accuracy of the computed input impedance of the source.

14.15 AI card

With this card an impressed current source is specified. The current varies linearly between the values at the start and end points, see Figure 14-8.

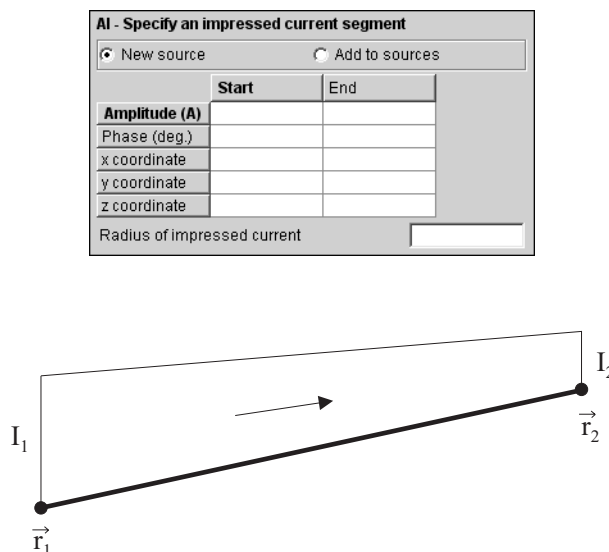


Figure 14-8: Impressed line current with a linear current distribution

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: Amplitude $|I_1|$ in A of the current at the *Start* point, $\vec{r}_1$, and *End* point, $\vec{r}_2$.

Phase: 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 in the FEKO model. (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 method, the current element must be in the free space medium, i.e. 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 which 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.

14.16 AK card

With this card a voltage source is applied to a radiating cable

AK - Add a voltage source to a radiating cable

☒ New source ☐ Add to sources

Positive

☐ Ground Connector label: Pin number:

Negative

☐ Ground Connector label: Pin number:

Magnitude of source (V):

Phase of source (degrees):

S-parameter impedance (Ohm):

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.

Ground: When this option is selected the cable connector pin is connected to ground (reference pin), denoted by *Pin 0*.

Connector label: The connector label uniquely identifies the connection of a cable path section.

Pin number: The pin number identifies the conductor associated with the cable path section as defined by the *Connector label* in the CS card (see section 14.30).

Magnitude of source: Magnitude of the voltage source in V.

Phase of source: Phase of the voltage source in degrees.

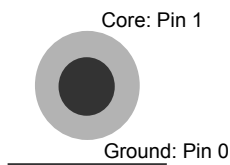
S-parameter impedance: The port impedance of this excitation is used in connection with S-parameter and realised gain calculations. If this field is empty or 0, the value specified at the SP card is used. (This value is only used if the S-parameters are requested with an SP card.)

Adding a voltage source to a cable requires knowledge of the conductor to cable connector pin relation. The following rules apply:

- *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 cross-section definition inside of a bundle

Note that loading (excitation) of a cable path is allowed only between connector pins within the same sub-circuit and its reference pin (i.e. loading between elements are not allowed over a shield of a cable). Pin numbering used for cable modelling can best be described by examples.

Example 1: An insulated single conductor above ground consisting of the following:

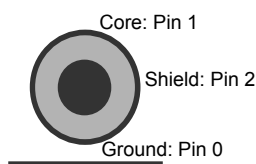


Sub-circuit 1: Single core (*Pin 1*) with geometry or meshed model as reference (*Pin 0*).

Example 2: Coaxial cable above ground consisting of the following:

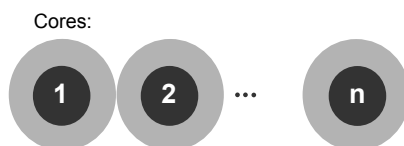
Sub-circuit 1: Outside of shield (*Pin 2*) with geometry or meshed model as reference (*Pin 0*).

Sub-circuit 2: Coaxial core (*Pin 1*) with the inside of shield (*Pin 2*) as reference.



Example 3: Ribbon cable with n cores consisting of the following:

Sub-circuit 1: Cores 1.. n (excluding *Pin i*) with *Pin i* as reference.



Example 4: Complex bundle (mixed) cable consisting 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

For example, loading (excitation) for sub-circuit 2 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.

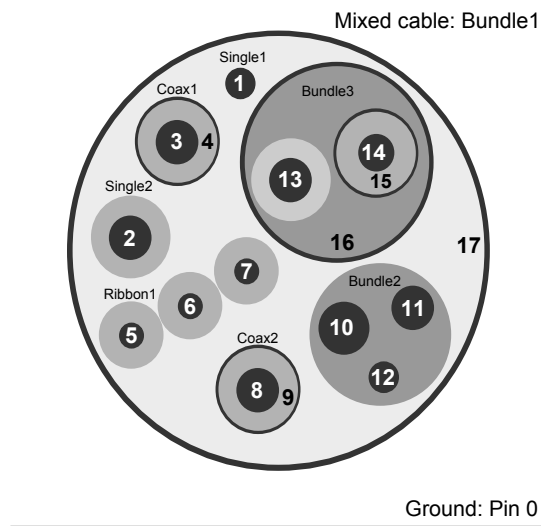


Table 14-2: Allowed loading between connector pins within the same sub-circuit and its reference pin

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}

14.17 AN card

With this card a voltage source can be added to any port of a general non-radiating network that does not have a connection to geometry.

AN - Add a voltage source to a network port

☒ New source ☐ Add to sources

Network name

Network port number

Magnitude of source (V)

Phase of source (degrees)

S-parameter impedance (Ohm)

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.
- Network name:* The network or transmission line name, with the network port number, uniquely identifies the connection terminals.
- Network port number:* The network port number, with the network/transmission line name, uniquely identifies the connection terminals.
- Magnitude of source:* Magnitude of the voltage source in V.
- Phase of source:* Phase of the voltage source in degrees.
- S-parameter impedance:* The port impedance of this excitation is used in connection with S-parameter and realised gain calculations. If this field is empty or 0, the value specified at the SP card is used. (This value is only used if the S-parameters are requested with an SP card.)

Adding of a voltage source to an internal network port (i.e. not connected to geometry) is completely defined using an 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.

14.18 AP card

With this card a planar, cylindrical or spherical aperture of measured or calculated field values is converted into an equivalent array of electric and magnetic dipoles. The card is processed by PREFEKO and replaced by A5 and A6 cards in the *.fek file.

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.

... field data ...: In this group one selects between *Load field data from *.efe/*.hfe file* (i.e. from *.efe or *.hfe files previously calculated with FEKO), *Load field data from ASCII text file* (the format of these files are documented below) or *The field data follows in the *.pre input file* (described below).

... aperture: 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 comment below).

Also sample along edges: This item is used to determine if dipoles should lie on the edges of the aperture or not (see Figures 14-9 to 14-12). When checked the outer dipoles lie on the edges, when unchecked the dipoles lie half an increment away from the edges. Dipoles should not lie 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.

S1, S2 and S3: These define the orientation of the aperture and should be clear from the figure on the dialog. For a planar aperture it defines 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 and x axes respectively. All angles are relative to these axes (see Figures 14-11 and 14-12).

E-field file name: The input file name from which the electric field data must be read. These may either be a *.efe 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. These may either be a *.efe or a text file (with units A/m).

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 aperture. 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 fields specify the number of field points along each of the two axes of the aperture.

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 $\vec{J}_s = \hat{n} \times \vec{H}$ and $\vec{M}_s = -\hat{n} \times \vec{E}$ 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 that all angles are read from the data, but no distance values.* Thus for planar apertures the positions are calculated entirely from the specified points (S1, S2 and S3). For cylindrical apertures S1 and S2 specify 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 lie 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.

Figures 14-9 and 14-10 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. When the *Also sample along edges* item is checked, the first point lies at S1 with the following points in the direction of S2 as shown by the indices in Figure 14-9. When it is unselected, the pattern is as shown in Figure 14-10. 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 14-11 shows the 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* item is checked, the samples extend up to the edges of the aperture, the points and the effective aperture are shown in figure (a). When it is unchecked, samples do not lie on the edges as shown in figure (b). Note that, when using identical input data as for the

case when the item is checked, the z positions of the samples changed, while in the φ direction the size of the effective aperture is increased by 5° on both sides.

Figure 14-12 shows 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 the *Also sample along edges* item is checked.

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

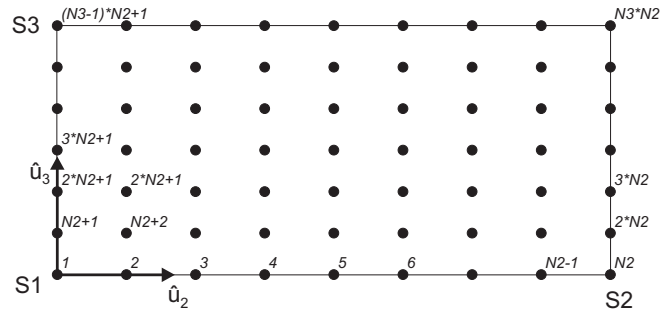


Figure 14-9: Location of the equivalent dipoles on a planar aperture where the *Also sample along edges* item is checked.

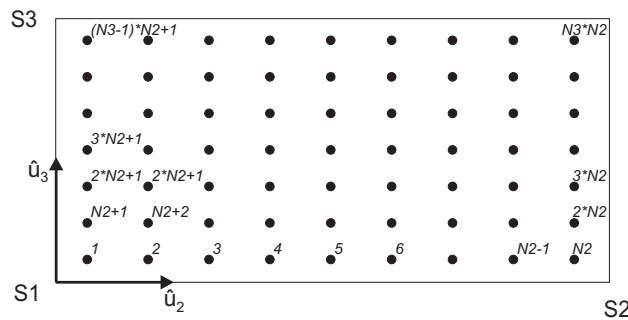


Figure 14-10: Location of the equivalent dipoles on a planar aperture where the *Also sample along edges* item is unchecked.

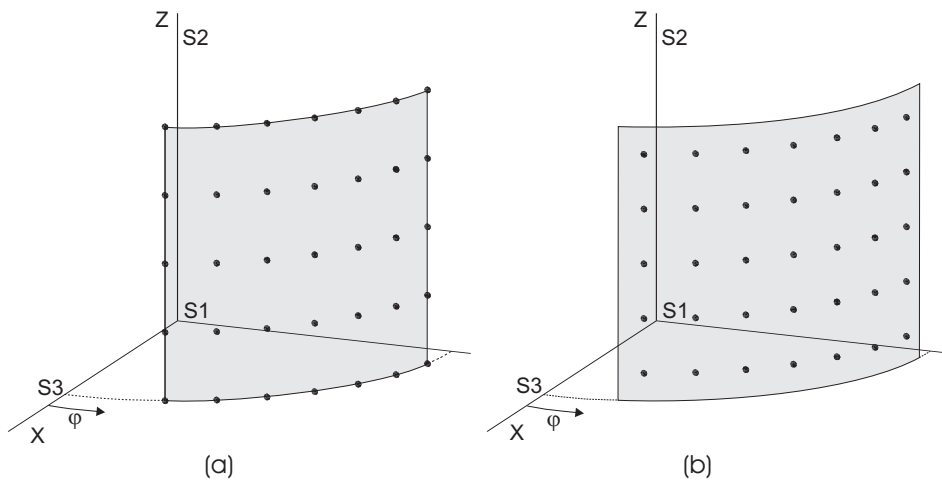


Figure 14-11: Location of the equivalent dipoles on a cylindrical aperture.

(a) *Also sample along edges* checked (b) *Also sample along edges* unchecked.

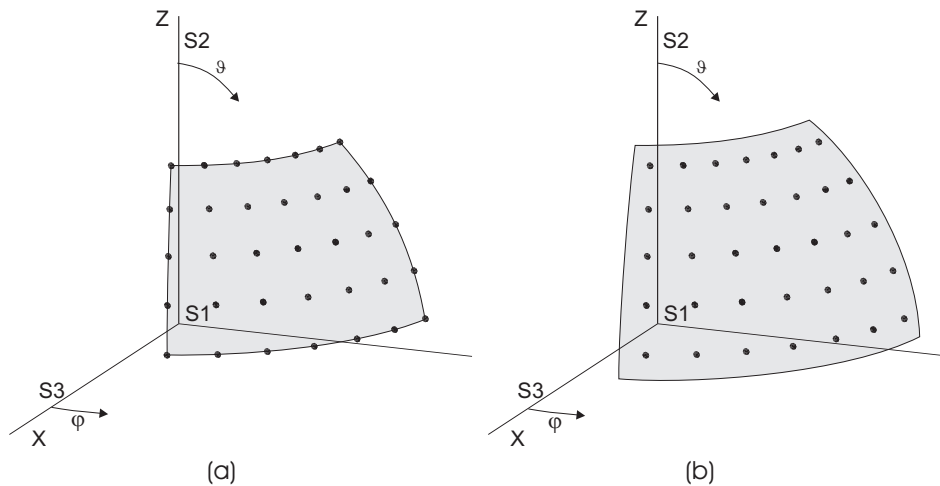


Figure 14-12: Location of the equivalent dipoles on a spherical aperture.

(a) *Also sample along edges checked* (b) *Also sample along edges unchecked*.

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 the *Also sample along edges* item is checked) are reduced by a factor of 2 and those on the corners by a factor of 4 so 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 to the exterior region, normally this is 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 and those outside retained.

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. Thus 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 (i.e. the structures should be electrically mirrored by using the SY card). Further, it should be remembered that 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. Note that FEKO takes this 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.

When the data is read from an ASCII format text file, each line in the file represents one point and the values are space delimited. For planar apertures, each line (point definition) must contain

the following four parameters: the absolute value and phase (in degrees) of the field component in the $\hat{u}_2$ direction followed by the absolute value and phase in the $\hat{u}_3$ direction (see the example below). It is required that the data is in a format in which the position first increments along the $\hat{u}_2$ direction. For cylindrical apertures each line (point definition) must contain the following five parameters: the angle φ (in degrees), then the absolute value and phase of the $\hat{\varphi}$ component, and then the absolute value and phase of the $\hat{z}$ component. For spherical apertures it must have six parameters: the angles ϑ and φ followed by the absolute value and phase of the $\hat{\vartheta}$ and $\hat{\varphi}$ components.

When the data is read from the *.pre input file, the data must be in the normal column based input format with the result that FOR loops etc. may be used. The four field components are the same as for the text data, and must be entered in columns 10 characters wide from columns 51 to 90. The angle φ must be in columns 40 to 49 and ϑ in columns 30 to 39 when they are required. 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 AP card usage:

As an example, consider an open ended X-band waveguide radiating through a hole in a large ground plane, as shown in Figure 14-13. Away from the aperture the plane $z = 0$ is perfectly conducting, i.e. the tangential electric field is zero, while the magnetic field is not — thus we will use electric symmetry.

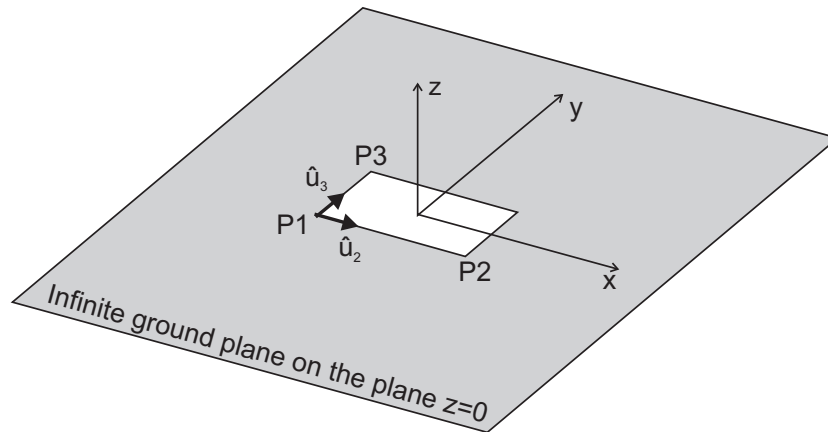


Figure 14-13: Example of an open waveguide as an implementation of the AP card

For this example the field is considered to be purely y directed (i.e. 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 (i.e. the $\hat{u}_2$ axis).

With *Number of points along $\hat{u}_2$* =5 and *Number of points along $\hat{u}_3$* =3 — in practice more points may be required — the data file 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  0  0

** 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"

...

```

This will generate nine x directed magnetic dipoles of the correct magnitude in the *.fek file.

14.19 AR card

With this card the radiation pattern of an antenna is used as an impressed source. The pattern is read from a data file or defined in the *.pre file below the AR card.

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, etc. 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 source in the UTD part which may 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.

Read pattern data from: In this group the user must select one of four options:

- *a *.ffe file:* Read the radiation pattern from an *.ffe format file (which may be created with the DA and FF cards).
- *an external data file:* Read the radiation pattern from an ASCII file (the format of this file is described below).
- *after this line in the *.pre file:* The radiation pattern is specified in the *.pre file following the AR card (the format is described below). With this option one can make use of the FOR loops to generate patterns from known functions.
- *use last pattern defined at previous AR card:* When using multiple AR cards (i.e. 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, then one can load the pattern just once and at subsequent AR cards use this

option. Then the last defined pattern will be used and memory can be saved (no need to store it again). Note that one is still able to set the antenna position, orientation, and amplitude and phase individually.

Source 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 of source: This parameter specifies a constant additional phase for the entire pattern.

Source position: In this group the x , y and z coordinates of the source point (i.e. the position where the antenna is placed) are entered in m. 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. We will refer to these fields as α_x , α_y and α_z in the rest of this discussion.

File name: The name of the *.ffe or ASCII input file.

Start from point number: This parameter is only relevant when the data is read from 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. This is the same as the line number if all blank lines, comment lines and header lines are stripped from the file.

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 a *.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

$$E_{\vartheta} = E_{\vartheta}^{FF} \cdot \frac{e^{-j\beta R}}{R} \quad (14-8)$$

or

$$E_{\varphi} = E_{\varphi}^{FF} \cdot \frac{e^{-j\beta R}}{R}, \quad (14-9)$$

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 , i.e. 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 $0^\circ \dots 180^\circ$ and they must be in ascending order, i.e. $\vartheta_{i+1} > \vartheta_i$. However, the angles do not have to be equidistantly spaced. (Thus, for example, for a highly directive antenna, a denser grid can be used close to the main beam direction.) The same applies to the angles φ_j where the permissible range is $0^\circ \dots 360^\circ$. For field angles outside the start and end values defined in the data (i.e. 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 $\vartheta_1 = 0^\circ$, $\vartheta_{\max} = 180^\circ$, $\varphi_1 = 0^\circ$ and $\varphi_{\max} = 360^\circ$.

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 α_z around the z axis, by α_y around the y axis and by α_x around the x axis. (The rotation is identical to the rotation executed by the TG card (see section 13.44) 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 $|E_{\vartheta}^{FF}|$ 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 value of the parameter *Read pattern data from*:

- *an *.ffe file (*.ffe file)*

With this option, the radiation pattern is read from a *.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, for example, 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				
FF	1	37	73	0	0	0	0	5	5

Note 37 points are used for ϑ and 73 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	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:

- The angle ϑ in degrees
- The angle φ in degrees
- Amplitude of the field strength E_{ϑ}^{FF} in V
- Phase of the field E_{ϑ}^{FF} in degrees
- Amplitude of the field strength E_{φ}^{FF} in V
- 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

ϑ_1	φ_1	$\cdots$
ϑ_2	φ_1	$\cdots$
ϑ_3	φ_1	$\cdots$
$\vdots$	$\vdots$	$\cdots$
ϑ_{I_4}	φ_1	$\cdots$
ϑ_1	φ_2	$\cdots$
ϑ_2	φ_2	$\cdots$
$\vdots$	$\vdots$	$\cdots$
ϑ_{I_4}	φ_{I_5}	$\cdots$

- *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 one may use the card dialog in EDITFEKO to generate a typical line.

We conclude this description with an example of a sector radiator. We want to realise an ideal sector radiator, which radiates 10 Watt horizontal polarisation in the angular region defined by $-70^\circ \leq \varphi \leq 70^\circ$ and $75^\circ \leq \vartheta \leq 105^\circ$. Since the angle range of the imported pattern must be positive, one may define separate sources for the regions $0^\circ \leq \varphi \leq 70^\circ$ and $290^\circ \leq \varphi \leq 360^\circ$. A more elegant solution is to define a single pattern in the range $0^\circ \leq \varphi \leq 140^\circ$ 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, i.e. E_{φ}^{FF} , is required):

```

** Application example for the AR card: Sector radiator

** No other structures considered
EG  1  0  0  0  0

** Set the frequency
FR  1  0          100.0e6

** Specified radiated power
PW  1          10.0

** Define the sector radiator
AR  0  3      2  2  1.0    0.0    0.0    0.0    0.0    0.0    0.0    -70
**      Theta  Phi      E_theta      E_Phi
      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    0.0    5.0    5.0

** 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 result is shown in Figure 14-14.

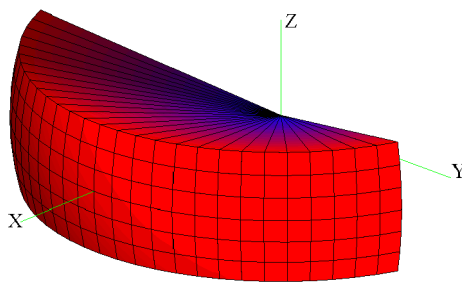


Figure 14-14: 3D radiation pattern of the sector radiator.

14.20 AS card

This card specifies an excitation by means of impressed spherical modes, which are either radiating (i.e. propagating in positive r direction to infinity, with r being the radius in a spherical coordinate system) or incident onto a structure (i.e. then propagating towards the origin $r = 0$).

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 (i.e. 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: This determines if the spherical waves propagate *Inward* (the model is illuminated with modes propagating towards $r = 0$, i.e. spherical Hankel function of the first kind $z_n^{(3)} = h_n^{(1)}$) or *Outward* (the modes radiate towards $r = \infty$, i.e. 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). Note that multiple spherical modes can be created as a single source. The imported spherical modes may be scaled and the phase given an offset by entering values for the *Scale magnitude by* and *Offset phase (deg)* fields.

- x, y, z position:* The coordinates of the origin $r = 0$ of the mode 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.
- |** *Rotation about axes:* The rotation of the spherical mode source about the x , y and z axes.
- 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, the user 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 $1, 2, \dots, \infty$ and m is the mode index in the azimuth direction φ . We do not distinguish between *even* and *odd* modes (with $\cos(m\varphi)$ and $\sin(m\varphi)$ angular dependencies), but rather use the angular dependency $e^{jm\varphi}$. Thus the index m can also be negative, but it must be in the range $-n \dots n$.
- 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

$$j = 2 [n(n+1) + m - 1] + s \quad (14-10)$$

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:* The phase of the complex amplitude of this spherical mode in degrees.

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_{smn}^c the electric and magnetic field strength is represented in a spherical coordinate system by

$$\vec{E}(r, \varphi, \vartheta) = \sum_{c=3}^4 \sum_{s=1}^2 \sum_{n=1}^{\infty} \sum_{m=-n}^n Q_{smn}^{(c)} \vec{F}_{smn}^{(c)}(r, \varphi, \vartheta) \quad (14-11)$$

$$\vec{H}(r, \varphi, \vartheta) = \frac{j}{Z_F} \sum_{c=3}^4 \sum_{s=1}^2 \sum_{n=1}^{\infty} \sum_{m=-n}^n Q_{smn}^{(c)} \vec{F}_{3-s,m,n}^{(c)}(r, \varphi, \vartheta) . \quad (14-12)$$

Here s , m and n are the mode indices with $s = 1$ indicating the TE-mode and $s = 2$ the TM-mode (see also the discussion above), 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 $F_{smn}^{(c)}$ are given by

$$\begin{aligned}\vec{F}_{1mn}^{(c)}(r, \vartheta, \varphi) &= \vec{M}_{mn}^{(c)} \\ &= \beta \sqrt{\frac{Z_F}{2\pi}} \frac{1}{\sqrt{n(n+1)}} \left(-\frac{m}{|m|}\right)^m \\ &\quad \cdot \left(0 \cdot \vec{e}_r \right. \\ &\quad \left. + z_n^{(c)}(\beta r) \frac{jm}{\sin \vartheta} \widehat{P_n^{|m|}}(\cos \vartheta) e^{jm\varphi} \vec{e}_\vartheta \right. \\ &\quad \left. - z_n^{(c)}(\beta r) \frac{\partial}{\partial \vartheta} \left\{ \widehat{P_n^{|m|}}(\cos \vartheta) \right\} e^{jm\varphi} \vec{e}_\varphi \right)\end{aligned}\quad (14-13)$$

and

$$\begin{aligned}\vec{F}_{2mn}^{(c)}(r, \vartheta, \varphi) &= \vec{N}_{mn}^{(c)} \\ &= \beta \sqrt{\frac{Z_F}{2\pi}} \frac{1}{\sqrt{n(n+1)}} \left(-\frac{m}{|m|}\right)^m \\ &\quad \cdot \left(\frac{n(n+1)}{\beta r} z_n^{(c)}(\beta r) \widehat{P_n^{|m|}}(\cos \vartheta) e^{jm\varphi} \vec{e}_r \right. \\ &\quad \left. + \frac{1}{\beta r} \frac{\partial}{\partial(\beta r)} \left\{ \beta r z_n^{(c)}(\beta r) \right\} \frac{\partial}{\partial \vartheta} \left\{ \widehat{P_n^{|m|}}(\cos \vartheta) \right\} e^{jm\varphi} \vec{e}_\vartheta \right. \\ &\quad \left. + \frac{1}{\beta r} \frac{\partial}{\partial(\beta r)} \left\{ \beta r z_n^{(c)}(\beta r) \right\} \frac{jm}{\sin \vartheta} \widehat{P_n^{|m|}}(\cos \vartheta) e^{jm\varphi} \vec{e}_\varphi \right)\end{aligned}\quad (14-14)$$

with the associated Legendre function

$$\widehat{P_n^{|m|}}(\cos \vartheta) = \sqrt{\frac{2n+1}{2} \frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \vartheta) \quad (14-15)$$

and the spherical Bessel functions

$$\begin{aligned}z_n^{(3)}(\beta r) &= h_n^{(1)}(\beta r) = j_n(\beta r) + j y_n(\beta r) \\ z_n^{(4)}(\beta r) &= h_n^{(2)}(\beta r) = j_n(\beta r) - j y_n(\beta r) .\end{aligned}\quad (14-16)$$

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 (e.g. 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, see earlier discussion).

Theoretically the index n runs in the range $1, 2, \dots, \infty$. For any practical application, one will have to consider a finite number of modes only, i.e. limit the range $n = 1 \dots N$. Some 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

$$N \approx \beta r_0 = 2\pi \frac{r_0}{\lambda} , \quad (14-17)$$

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

$$N \approx \beta r_0 + 10 \quad (14-18)$$

or

$$N \approx \beta r_0 + 3 \sqrt[3]{\beta r_0} . \quad (14-19)$$

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

$$J = 2 [N(N + 1) + N - 1] + 2 = 2N(N + 2) \quad (14-20)$$

for j (i.e. $j = 1 \dots J$ then). So for instance for an antenna with enclosing radius $r_0 = \frac{\lambda}{4}$ (then $\beta r_0 = 1.57$) 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 origin $r = 0$ of the modes can be set arbitrarily in space by means of the *Source position* field. However, no provision has been made to allow a rotation, i.e. in spherical coordinates the distance r of some point is measured with respect to the mode origin, but the angles ϑ and φ are always defined with respect to the global coordinate system. The reason why a rotation is not supported here for the AS card (as opposed to the A5 or A6 elementary dipoles) is that the modes inherently contain this radiation in an arbitrary direction. For instance, the TM-mode $n = 1$, $m = 0$ represents a z oriented electric dipole (i.e. vertical), while the TM-modes $n = 1$, $m = \pm 1$ represent horizontally aligned dipoles. An arbitrary pattern orientation can be obtained by a suitable superposition of different 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

$$P = 0.5 |\text{Magnitude of the mode}|^2 \quad (14-21)$$

(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 under *Propagation direction*) to originate from the *Source position*, i.e. the source is located in the medium where *Source position* is located, and its contribution will be zero in all other media. For inward propagating modes (when *Inward* is selected under *Propagation direction*) we assume them 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 $R \rightarrow \infty$ of the field strength with the factor

$$\frac{e^{-j\beta R}}{R} \quad (14-22)$$

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, i.e. the response of a receiving antenna to some incident mode. Then one looks at these quantities:

- Induced voltage or power at the antenna terminals for the network ports (i.e. 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 that 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  0  0

** Set the frequency
FR  1 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  2  10

** End
EN

```

The resulting pattern is shown in Figure 14-15. 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

```

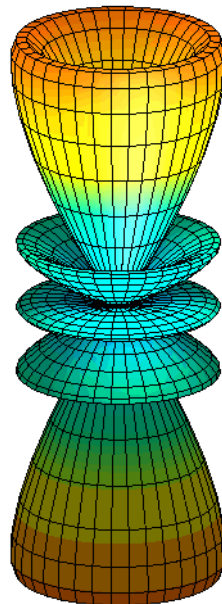



Figure 14-15: 3D radiation pattern of a spherical TE-mode with $n = 5$ and $m = 0$.

14.21 AV card

With this card an impressed current source is specified similar to that of the AI card, but with the AV card the end point makes electrical contact with a conducting surface as shown in Figure 14-16. 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.

AV - Impressed current connected to triangle

☒ New source ☐ Add to sources

☒ Specify end point coordinates below
☐ Connect to closest triangle(s) vertex

	Start	End
Amplitude (A)		
Phase (deg.)		
x coordinate		
y coordinate		
z coordinate		

Radius of impressed current

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 below: The coordinates of the end point $\vec{r}_2$ are known and 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(s) vertex: The coordinates of the end point $\vec{r}_2$ 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 $\vec{r}_1$ of the current element. This is then the end point $\vec{r}_2$. For both cases FEKO automatically searches for all the triangles making electrical contact with the end point.

Amplitude: Current amplitude (in A) at the *Start*, $\vec{r}_1$, and *End*, $\vec{r}_2$, points.

Phase: 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 surfaces:

- 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 $\vec{r}_2$ a continuous current model is used so that a point charge is not possible here.

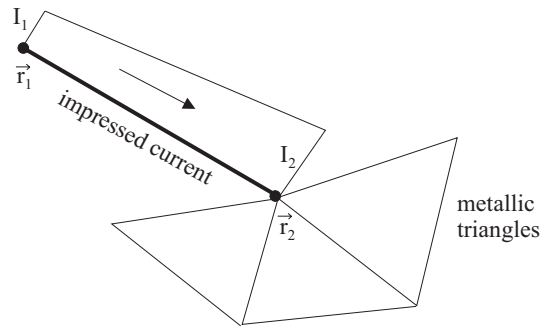


Figure 14-16: Impressed line current with a linear current distribution and electrical contact to conducting triangles

14.22 AW card

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.

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.

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 $\hat{u}_1$ (from S1 to S2) and $\hat{u}_2$ (from S1 to S3). The direction in which the mode is launched is given by $\hat{u}_3 = \hat{u}_1 \times \hat{u}_2$.

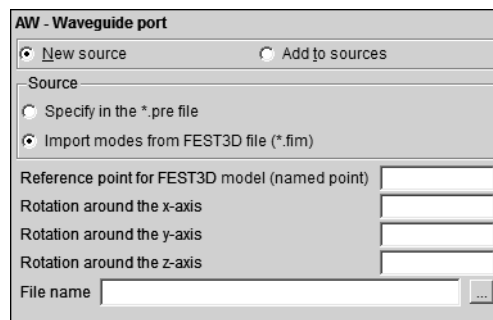
<i>Circular:</i>	A circular waveguide cross section is used. The point S1 denotes the centre of the circular port, and the point S2 specifies the radius and start point for the angular dependency. A further point S3 must be perpendicular above the centre of the circular plate, so that the direction from S1 to S3 indicates the direction in which the waveguide modes are launched.
<i>Coaxial:</i>	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.
<i>Excite fundamental mode:</i>	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) can not be specified since they are determined automatically.
<i>TE-mode:</i>	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.
<i>TM-mode:</i>	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.
<i>TEM-mode:</i>	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.
<i>Mode index m:</i>	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 $\hat{u}_1$ direction (i.e. 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.
<i>Mode index n:</i>	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 $\hat{u}_2$ direction (i.e. 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.
<i>Magnitude...:</i>	Absolute value of the complex amplitude of the impressed mode. For a TE-mode the unit is A/m, for a TM-mode the unit is 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 (e.g. as waveguide termination), and no wave is launched.
<i>Phase:</i>	The phase of the impressed mode in degrees.
<i>Rotation angle φ_0:</i>	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).
<i>Modal expansion:</i>	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: 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.

Import modes from FEST3D file (*.fim)

With this option, impressed forward travelling modes in all waveguides of a multi-port network can be imported from a *.fim file.



Parameters:

Position: 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 $\hat{u}_3$, 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 (see section 2.2.2). 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. 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. If this approach is followed, then waveguide feeds may be used. 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*, there `waveguide_step` (S-parameter computation for a waveguide discontinuity) and in the *Script Examples* `example_08` (horn antenna), and `example_34` (coaxial cable).

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.

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 $\hat{u}_1$ (distance between S1 and S2), and b the dimension of the waveguide port in $\hat{u}_2$ (distance between S1 and S3). The modal cutoff coefficient is the same for TM- and TE-modes, and is given by,

$$\beta_c = \left(\left(\frac{m\pi}{a} \right)^2 + \left(\frac{n\pi}{b} \right)^2 \right)^{1/2}. \quad (14-23)$$

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,

$$E_{u_3}^{\text{TM}}(u_1, u_2) = \hat{A} \sin \frac{m\pi u_1}{a} \sin \frac{n\pi u_2}{b} \quad (14-24)$$

for $m = 1, 2, \dots$, and $n = 1, 2, \dots$. $\hat{A}$ is the complex amplitude of the impressed mode in V/m. The remaining field components are,

$$E_{u_1}^{\text{TM}}(u_1, u_2) = Z_{\text{TM}} H_{u_2}^{\text{TM}}(u_1, u_2) = -\frac{j\beta}{\beta_c^2} \left(\frac{m\pi}{a} \right) \hat{A} \cos \frac{m\pi u_1}{a} \sin \frac{n\pi u_2}{b} \quad (14-25)$$

$$E_{u_2}^{\text{TM}}(u_1, u_2) = -Z_{\text{TM}} H_{u_1}^{\text{TM}}(u_1, u_2) = -\frac{j\beta}{\beta_c^2} \left(\frac{n\pi}{b} \right) \hat{A} \sin \frac{m\pi u_1}{a} \cos \frac{n\pi u_2}{b} \quad (14-26)$$

$$Z_{\text{TM}} = \frac{\beta}{\omega \epsilon} \quad (14-27)$$

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,

$$H_{u_3}^{\text{TE}}(u_1, u_2) = \hat{A} \cos \frac{m\pi u_1}{a} \cos \frac{n\pi u_2}{b} \quad (14-28)$$

for $m = 0, 1, 2, \dots$, and $n = 0, 1, 2, \dots$ (but not both m and n zero). $\hat{A}$ is the complex amplitude of the impressed mode in A/m. The remaining field components are,

$$E_{u_1}^{\text{TE}}(u_1, u_2) = Z_{\text{TE}} H_{u_2}^{\text{TE}}(u_1, u_2) = \frac{j\beta}{\beta_c^2} \left(\frac{n\pi}{b} \right) Z_{\text{TE}} \hat{A} \cos \frac{m\pi u_1}{a} \sin \frac{n\pi u_2}{b} \quad (14-29)$$

$$E_{u_2}^{\text{TE}}(u_1, u_2) = -Z_{\text{TE}} H_{u_1}^{\text{TE}}(u_1, u_2) = -\frac{j\beta}{\beta_c^2} \left(\frac{m\pi}{a} \right) Z_{\text{TE}} \hat{A} \sin \frac{m\pi u_1}{a} \cos \frac{n\pi u_2}{b} \quad (14-30)$$

$$Z_{\text{TE}} = \frac{\omega \mu}{\beta} \quad (14-31)$$

Circular waveguide expressions

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, i.e. for $r \leq a$, 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 $TM_{m,n}$ mode is expressed as,

$$E_z^{\text{TM}}(r, \varphi) = \hat{A} J_m(\beta_c r) \cos [m(\varphi - \varphi_0)] \quad (14-32)$$

for $m = 0, 1, 2, \dots$, and $n = 1, 2, \dots$. $\hat{A}$ is the complex amplitude of the impressed mode in 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,

$$E_r^{\text{TM}}(r, \varphi) = Z_{\text{TM}} H_\varphi^{\text{TM}}(r, \varphi) = -\frac{j\beta}{\beta_c} \hat{A} J'_m(\beta_c r) \cos [m(\varphi - \varphi_0)] \quad (14-33)$$

$$E_{\varphi}^{\text{TM}}(r, \varphi) = -Z_{\text{TM}} H_r^{\text{TM}}(r, \varphi) = \frac{j\beta m}{\beta_c^2 r} \hat{A} J_m(\beta_c r) \sin [m(\varphi - \varphi_0)] \quad (14-34)$$

$$Z_{\text{TM}} = \frac{\beta}{\omega \epsilon} \quad (14-35)$$

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,

$$H_z^{\text{TE}}(r, \varphi) = \hat{A} J_m(\beta_c r) \cos [m(\varphi - \varphi_0)] \quad (14-36)$$

for $m = 0, 1, 2, \dots$, and $n = 1, 2, \dots$. $\hat{A}$ is the complex amplitude of the impressed mode in 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,

$$E_r^{\text{TE}}(r, \varphi) = Z_{\text{TE}} H_{\varphi}^{\text{TE}}(r, \varphi) = \frac{j\beta m}{\beta_c^2 r} Z_{\text{TE}} \hat{A} J_m(\beta_c r) \sin [m(\varphi - \varphi_0)] \quad (14-37)$$

$$E_{\varphi}^{\text{TE}}(r, \varphi) = -Z_{\text{TE}} H_r^{\text{TE}}(r, \varphi) = \frac{j\beta}{\beta_c} Z_{\text{TE}} \hat{A} J'_m(\beta_c r) \cos [m(\varphi - \varphi_0)] \quad (14-38)$$

$$Z_{\text{TE}} = \frac{\omega \mu}{\beta} \quad (14-39)$$

Coaxial waveguide expressions

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 coaxial waveguide, i.e. for $r_i \leq r \leq r_o$, 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 functions 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 $\beta = \omega \sqrt{\mu \epsilon}$. The axial electric and magnetic field components are zero for a TEM-mode, and the transverse field components have a static field distribution,

$$E_r^{\text{TEM}}(r, \varphi) = Z_{\text{TEM}} H_{\varphi}^{\text{TEM}}(r, \varphi) = -\frac{\hat{A}}{r} \quad (14-40)$$

$$Z_{\text{TEM}} = \sqrt{\frac{\mu}{\epsilon}} \quad (14-41)$$

$\hat{A}$ 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 $TM_{m,n}$ mode is expressed as,

$$E_z^{\text{TM}}(r, \varphi) = \hat{A} \left(-\frac{J_m(\beta_c r)}{J_m(\beta_c r_o)} + \frac{N_m(\beta_c r)}{N_m(\beta_c r_o)} \right) \cos [m(\varphi - \varphi_0)] \quad (14-42)$$

for $m = 0, 1, 2, \dots$, and $n = 1, 2, \dots$. $\hat{A}$ is the complex amplitude of the impressed mode in 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,

$$\begin{aligned} E_r^{\text{TM}}(r, \varphi) &= Z_{\text{TM}} H_\varphi^{\text{TM}}(r, \varphi) \\ &= -\frac{j\beta}{\beta_c} \hat{A} \left(-\frac{J'_m(\beta_c r)}{J_m(\beta_c r_o)} + \frac{N'_m(\beta_c r)}{N_m(\beta_c r_o)} \right) \cos [m(\varphi - \varphi_0)] \end{aligned} \quad (14-43)$$

$$\begin{aligned} E_\varphi^{\text{TM}}(r, \varphi) &= -Z_{\text{TM}} H_r^{\text{TM}}(r, \varphi) \\ &= \frac{j\beta}{\beta_c^2 r} \hat{A} \left(-\frac{J_m(\beta_c r)}{J_m(\beta_c r_o)} + \frac{N_m(\beta_c r)}{N_m(\beta_c r_o)} \right) \sin [m(\varphi - \varphi_0)] \end{aligned} \quad (14-44)$$

$$Z_{\text{TM}} = \frac{\beta}{\omega \epsilon} \quad (14-45)$$

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,

$$H_z^{\text{TE}}(r, \varphi) = \hat{A} \left(-\frac{J_m(\beta_c r)}{J'_m(\beta_c r_o)} + \frac{N_m(\beta_c r)}{N'_m(\beta_c r_o)} \right) \cos [m(\varphi - \varphi_0)] \quad (14-46)$$

for $m = 0, 1, 2, \dots$, and $n = 1, 2, \dots$. $\hat{A}$ is the complex amplitude of the impressed mode in 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 two conductors must be zero at the inner and outer radii. The remaining field components are,

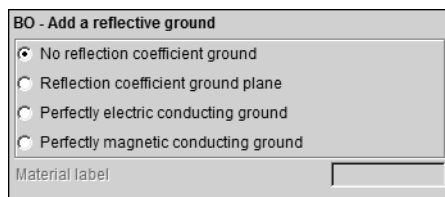
$$\begin{aligned} E_r^{\text{TE}}(r, \varphi) &= Z_{\text{TE}} H_\varphi^{\text{TE}}(r, \varphi) \\ &= \frac{j\beta m}{\beta_c^2 r} Z_{\text{TE}} \hat{A} \left(-\frac{J_m(\beta_c r)}{J'_m(\beta_c r_o)} + \frac{N_m(\beta_c r)}{N'_m(\beta_c r_o)} \right) \sin [m(\varphi - \varphi_0)] \end{aligned} \quad (14-47)$$

$$\begin{aligned} E_\varphi^{\text{TE}}(r, \varphi) &= -Z_{\text{TE}} H_r^{\text{TE}}(r, \varphi) \\ &= \frac{j\beta}{\beta_c} Z_{\text{TE}} \hat{A} \left(-\frac{J'_m(\beta_c r)}{J'_m(\beta_c r_o)} + \frac{N'_m(\beta_c r)}{N'_m(\beta_c r_o)} \right) \cos [m(\varphi - \varphi_0)] \end{aligned} \quad (14-48)$$

$$Z_{\text{TE}} = \frac{\omega \mu}{\beta} \quad (14-49)$$

14.23 BO card

With this card a ground plane (at $z = 0$) can be specified for all computations following the BO card. The reflection coefficient method is used.



Parameters:

No reflection coefficient ground: No ground plane (or use one of the other ground plane options — such as the GF card for an exact model of real ground, or the SY card for a perfect 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.

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 $z = 0$. The remaining parameters are also ignored in this case.

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, i.e. it is not possible to calculate the fields in the region $z < 0$. 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 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 $\frac{\lambda}{10}$ (FEKO 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, i.e. the hybrid MoM/UTD method cannot be used in the presence of a real ground.

14.24 CA card

This card is used to define a section of a shielded cable which is used for irradiation (i.e. computing induced currents and voltages at the cable terminals) due to external sources. Transmission line theory is applied, i.e. 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). 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).

CA - Cable path modelling for irradiation

☐ Remove all existing cable paths
☒ New cable path
☐ Add to existing cable paths

$Z_1^{(int)}$ $Z_2^{(int)}$
 $Z_1^{(ext)}$ $Z_2^{(ext)}$

Location

☒ Define points
 Start point (defines port 1)
 End point (defines port 2)
☐ Import from NASTRAN file
 File name
 NASTRAN segment property ID

Cable type

User defined cable

Cable properties

Magnitude of characteristic impedance (Ohm)
 Phase of characteristic impedance (degrees)
 Real part of propagation constant (1/m)
 Imag. part of propagation constant (1/m)
 Resistance of cable shield (Ohm/m)
 Magnitude of transfer impedance (Ohm/m)
 Phase of transfer impedance (degrees/m)
 Outer radius of cable shield

Port termination

Port 1

☒ Terminated ☐ Connected to another cable
 Real part of external impedance $Z_1^{(ext)}$ (Ohm)
 Imag. part of external impedance $Z_1^{(ext)}$ (Ohm)
 Real part of internal impedance $Z_1^{(int)}$ (Ohm)
 Imag. part of internal impedance $Z_1^{(int)}$ (Ohm)

Port 2

☐ Terminated ☒ Connected to another cable
 Real part of external impedance $Z_2^{(ext)}$ (Ohm)
 Imag. part of external impedance $Z_2^{(ext)}$ (Ohm)
 Real part of internal impedance $Z_2^{(int)}$ (Ohm)
 Imag. part of internal impedance $Z_2^{(int)}$ (Ohm)

Sampling point density

☒ Automatic determination
☐ Specify maximum separation distance

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 (i.e. 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 etc.).

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 *pre-defined cable from the internal FEKO cable database*:

If *User defined cable* is selected in the drop-down list, then 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 *pre-defined cable* type is selected from the drop-down list, then 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 (i.e. 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.

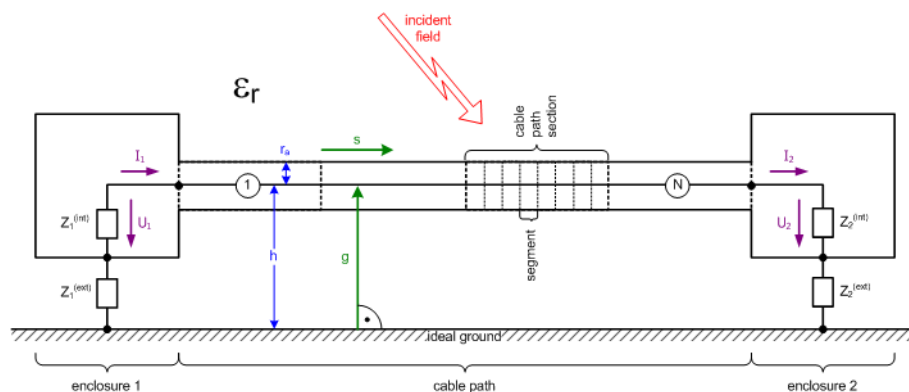


Figure 14-17: Complete scenario of a cable path

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, i.e. 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 (e.g. different cable positions, etc.) without repetition of a time-consuming solution of the whole model (e.g. MoM or MLFMM). Analysing another cable is similar to computing the near field at another point.

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 14-17. 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* $\vec{s}$ and the *ground vector* $\vec{g}$ must be (almost) perpendicular. This figure also shows the *cable path*, the *cable path sections* (1 ... N) and the *segments*.

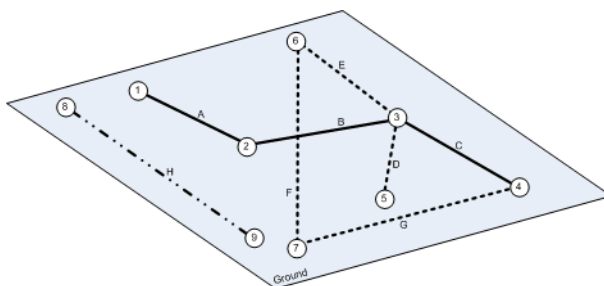


Figure 14-18: Possible ways to define cable paths

Figure 14-18 shows the setup of some cable paths. There are three cable paths in total (distinguished by the line style):

Table 14-3: Cable paths in figure 14-18.

<i>Path</i>	<i>Description</i>
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 (e.g. 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 14-18 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 path2)
CA Defining section F (continue cable path2)
CA Defining section G (end cable path2)
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, i.e. 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 (e.g. 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.

14.25 CD card

With this card a specific cable cross-section can be defined.

Single conductor

A single cable is defined by specifying the cable cross-section.

Parameters:

- PEC:** Select this option to set the core of the cable to PEC.
- Core material label:** The label of the metallic (as defined in the DI card) to be used for the core.
- Core radius:** The radius of the core is specified.
- Insulation material label:** The label of the material (as defined in the DI card) to be used as the insulation material.
- Thickness:** The thickness of the material to be used as insulator.

Coaxial cable

Three types of coaxial cables are supported: coaxial cables defined by characteristics, dimensions and an internally defined database of cables.

- Coaxial cables defined by cable properties.

- Shield label:** The label of the metallic (as defined in the DI card) to be used for the shield.
- Outer radius:** The outer radius of the shield.
- Magnitude of the characteristic impedance (Ohm):** The magnitude of the characteristic impedance of the cable.

Phase of the characteristic impedance (degrees): The phase of the characteristic impedance of the cable.

Real part of the propagation constant (m^{-1}): Specify the real part of the propagation constant.

Imaginary part of the propagation constant (m^{-1}): Specify the imaginary part of the propagation constant.

- Coaxial cables defined by the geometry dimensions

CD - Cable cross-section definition

Cross section label

Cable type

☐ Single conductor

☒ Coaxial cable

☐ Ribbon cable

☐ Bundle (mixed) cable

Coaxial cable dimensions

Core radius

Shield outer radius

Insulation layer thickness

① Insulation layer 1

② Insulation layer n

Shield

Shield label

Outer radius

Core

☐ PEC

Material label

Core radius

Number of insulation layers

	Material label	Thickness
1		

Shield label: The label of the metallic (as defined in the DI card) to be used for the shield.

Shield outer radius: The outer radius of the shield.

PEC: Select this option to set the core of the cable to PEC.

Material label: The label of the metallic (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.

- Internally defined database of cables. A pre-defined coaxial cable type can be selected from the list. Several commonly found shielded cable types are included in this list.

Ribbon cable

With this option one can define ribbon cables. Note that currently only round cores are supported.

Parameters:

CD - Cable cross-section definition

Cross section label

Cable type

☐ Single conductor

☐ Coaxial cable

☒ Ribbon cable

☐ Bundle (mixed) cable

Core radius

Insulation thickness

Core spacing

Core

☐ PEC

Material label

Core radius

Number of cores

Core spacing

Return path number

Insulation

Material label

Thickness

PEC: Select this option to set the core of the cable to PEC.

Material label: The label of the metallic (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.

Return path number: The number of the cable specified as the the return path.

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.

Bundle (mixed) cable

Bundles allow for the construction of complex multi-core cables based on existing cables. These cables are built from all types of cables already defined - single conductor, coaxial cable, ribbon cables and multi-cables. Note that currently only single conductor cables may be bundled together.

Parameters:

Number of cables: The number of cables which constitute the bundle cable.

Cable: The reference number of the cable.

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.

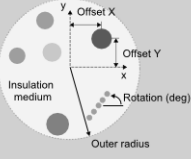
Shield label: The label of the metallic (as defined in the DI card) to be used as the shield.

CD - Cable cross-section definition

Cross section label

Cable type

- ☐ Single conductor
- ☐ Coaxial cable
- ☐ Ribbon cable
- ☒ Bundle (mixed) cable



Number of cables

Cable	Offset X	Offset Y
1		

Shield

Shield label

Insulation material label

Outer radius

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 cable.

14.26 CF card

Set the type of integral equation for perfectly conducting metallic surfaces.

CF - Type of integral equation

Type of integral equation for metallic surfaces

☒ EFIE

☐ CFIE

☐ Apply to all labels

☒ Apply to single label only

☐ Apply to label range

From label

to label

Factor for CFIE

Parameters:

Type of integral equation for metallic surfaces: Here one can choose the EFIE (*electric field integral equation*) and also 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, which should normally be used.

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, and 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 in conjunction with EFIE (on the same object) as long as all triangle normals 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 (i.e. from the closed field free region into the domain of interest where there are sources and fields shall be computed).
- Using symmetry in order to reduce the memory or run-time is not supported in FEKO (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 (i.e. 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 etc.).

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.

14.27 CG card

Here the method used to solve the matrix equation may be chosen. Normally the CG card should not be used. FEKO has been written to choose the optimal solution technique and also the corresponding pre-conditioners and options etc. for different types of problems automatically. These algorithms should work in all cases, but they might not be optimal for specific MLFMM or FEM configurations (which rely on iterative solvers). Thus for these solutions, advanced users might after consultation with FEKO technical support, apply the CG card in such special cases. But one should realise that convergence of the iterative techniques cannot be assured, and also when using an inappropriate preconditioner the memory requirement might be much higher than required. Also users should take care to reconsider any CG card settings in models that are derived from models containing the CG card.

Parameters:

In the first drop-down list, the matrix solution method is chosen:

- *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 (e.g. 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 make other CG card selections with respect to the pre-conditioning.
- *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, i.e. 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 on 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 FEM (no preconditioning).

Maximum number of iterations: The number of iterations limit for the iterative techniques.

Stopping criterium for residuum: Termination criterion for the normalised residue when using iterative methods. (Terminate with convergence 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. (Terminate with divergence when the normalised residue is larger than this value.)

Preconditioners: In this list the preconditioner is selected from:

- *Default preconditioner selection (recommended)* When this option is selected, then FEKO will automatically select a suitable preconditioner 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 (e.g. MLFMM or FEM). This option has regarding the preconditioner the same effect as not using a CG card, but still allows then to make other selections with respect to the solver (e.g. residuum).

- *No preconditioning*: No preconditioning is used.
- *Scaling the matrix A*: Scaling the matrix $[A]$, so that the elements on the main diagonal are all normalised to one.
- *Scaling the matrix $[A]^H[A]$* : Scaling the matrix $[A]^H[A]$, so that the elements on the main diagonal are all normalised to one.
- *Block-Jacobi preconditioning using inverses*: The inverses of the preconditioner are calculated and applied during every iteration step. For performance reasons *Block-Jacobi preconditioning using LU-decomposition* is recommended.
- *Neumann polynomial preconditioning*: Self explanatory.
- *Block-Jacobi preconditioning using LU-decomposition*: Block-Jacobi preconditioning where for each block a LU-decomposition is computed in advance, and during the iterations a fast backward substitution is applied.
- *Incomplete LU-decomposition*: Use an incomplete LU-decomposition of the matrix as a preconditioner.
- *Block-Jacobi preconditioning of MLFMM one-level-up*: Special preconditioner for the MLFMM, where additional information is included into the preconditioner.
- *LU decomposition of FEM matrix*: An LU decomposition of the FEM matrix is used as preconditioner.
- *ILUT decomposition of FEM matrix*: An incomplete LU decomposition with thresholding of the FEM matrix is used as preconditioner.
- *Multilevel ILUT/Diagonal decomposition of the FEM matrix*: Self explanatory.
- *Multilevel ILUT/ILUT decomposition of the FEM matrix*: Self explanatory.
- *Multilevel LU/Diagonal decomposition of the FEM matrix*: Self explanatory.
- *Sparse Approximate Inverse (SPAI) preconditioner*: Preconditioner which can be used in connection with the MLFMM.
- *Complete LU-decomposition*: Use a complete LU decomposition of the matrix (MLFMM/ACA) as a preconditioner.

Options for the BCG: This applies to the Biconjugate Gradient Method. Options are:

- *Fletcher's method*
- *Jacob's method*
- *Fletcher's method, pre-iteration using Fletcher's method*
- *Fletcher's method, pre-iteration using Jacob's method*
- *Jacob's method, pre-iteration using Fletcher's method*

Block size: The block size to be used for LU-decomposition with LAPACK as well as for the Block-Jacobi preconditioning. When nothing is specified (i.e. the input field is left empty), then appropriate standard values are used for LAPACK and the block preconditioners.

Threshold value for ILUT: This is the thresholding value used for the FEM in connection with the ILUT preconditioners.

Level-of-fill: This is used by the MLFMM during the iterative matrix solution in connection with the incomplete LU preconditioner. The range of this parameter is between 0 and 12. FEKO will choose the value for the best preconditioning, but if the size of the incomplete LU preconditioner is too large to fit into memory, it can be reduced by reducing the level-of-fill. It should be noted that a lower level-of-fill might result in a slower convergence or even divergence in the iterative solution.

Fill-in level per row: This is used by the FEM during the iterative matrix solution in connection with the incomplete LU preconditioners with thresholding. It sets a limit on the number of entries per row that will be included in the incomplete LU-decomposition of the preconditioner matrix.

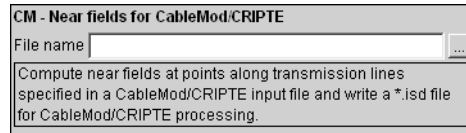
Stabilisation factor (FEM): This applies only to the incomplete LU preconditioners of the FEM and can be used to get better convergence for the FEM in critical cases (the value range is between 0 and 1).

Save/read preconditioner: For the incomplete LU preconditioners used with the FEM one has here the option to save run-time by computing the preconditioner only once and write to a *.pcr file and then for a subsequent run just read again from this file. Since the FEM pre-conditioners depend only on the FEM matrix even for combined MoM/FEM hybrid methods, this method is useful for instance when only the MoM part of a problem setup has changed.

Iterative solutions are used for instance for the MLFMM or the FEM. There might then be situations where during the iterations the residuum is decreasing but not reaching the specified stopping criterion. Instead of waiting very long until the maximum number of iterations has been 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 (e.g. 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).

14.28 CM card

This 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.)



CM - Near fields for CableMod/CRIPTE

File name ...

Compute near fields at points along transmission lines specified in a CableMod/CRIPTE input file and write a *.isd file for CableMod/CRIPTE processing.

The only parameter of this card is the file name of a *.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 near field 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.

14.29 CO card

This card specifies a dielectric or magnetic coating of wire segments or triangular surface elements. The coating applies to all calculations following the CO card.

Note that before a coating can be applied, the material properties for the coating need to be defined in the DI card. A coating with the specific material properties are then applied by referencing the label of the material in the CO card. A layered medium are defined by referencing the labels of the materials to be used for the individual layers in the DL card.

Parameters:

Label of elements to coat: All segments or triangles with this label are coated.

No coating: 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): Here the radius of the metallic wire is retained. The effect of the dielectric layer is accounted for by a volume polarisation current. The only restriction of this method is that the layer may not be magnetic in nature (i.e. the relative permeability μ_r as well as the magnetic loss tangent $\tan \delta_\mu$ 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. The layers may have different permittivity and permeability, but the total coating must be electrically (i.e. relative to the wavelength in the coating) 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. Here it is only required that the total coating must be geometrically thin, i.e. it must be thin relative to the triangle size (and thus also to the free space wavelength) as well as the radius of curvature of the surface.

<i>Material label:</i>	Label of the material (as specified in the DI card) which will be used as a coating.
<i>Layered dielectric label:</i>	Label of the layered dielectric medium (as specified in the DL card) which will be used as a coating..
<i>Thickness of ...:</i>	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 m and is scaled by the SF card.
<i>Wire radius:</i>	Wire radius ϱ of the metallic wire, without layers, in m (it is scaled by the SF card). This overrides the values specified with the IP card. This field is only applicable to wire coatings.

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 $\tan \delta = \frac{\sigma}{\omega \epsilon_r \epsilon_0}$) has to be identical to the loss tangent of the surrounding medium (specified with the EG card, usually free space)
- 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 (i.e. the relative permeability μ_r as well as the magnetic loss tangent $\tan \delta_\mu$ of the layer equal those of the surrounding medium) the option *Wire coating (Volume equivalence principle)* is recommended.

Note that for wire coatings, no surface triangles with the same label are allowed. Likewise for surface coatings, no segments with same label are allowed.

If *Dielectric / magnetic surface coating* (the electrically thick coating) is used, it must remain consistent for the whole FEKO run. (It cannot be on for one solution and off for the next.) It is, however, allowed to change the thickness and the medium parameters of the coating.

14.30 CS card

This card is used to define a path along which the cable runs thus also providing the centre/reference location to which a cable cross-section definition is applied.

A path may be specified using data points in the *.pre file or loading a cable path from a NAS-TRAN file.

CS - Define cable path section

☐ Remove all existing cable paths
☒ New cable path
☐ Add to existing cable paths

Path section label:

Cross-section definition label:

Cable type
☒ Irradiating
☐ Radiating

Solution method for outer cable problem (shield / external ground)
☒ Multiconductor transmission line (MTL)
☐ Method of moments (MoM), only for shielded cables

Path
☒ Specify points in *.pre file
☐ Import from Nastran file

Number of points:

	Path points
1	
2	

Connector labels
 Connector at start:
 Connector at end:

Sampling point density
☒ Automatic determination
☐ Specify maximum separation distance:

☐ Export cable parameters to *.out file

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 (i.e. the previously added ones 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.

Cable type: **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 in the cables will also be considered.

Solution method: **Multiconductor transmission line (MTL)** When this option is selected, the model will be solved with the multiconductor transmission line theory which is also hybridised with MoM or the MLFMM. The cable path should be within $\frac{\lambda}{5}$ of the conducting surface.

Hybrid MoM/MTL When this option is selected, the model will be solved by means of the hybrid 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 are also required to be able to import the cable path section.

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 they share a node, use the same cable connector and the same cable cross-section definition. Thus, after combining/reducing the number of cable path sections, each cable path consists of a section of cable with a unique cross-section. Note that there is no support for splitting of cable sections or combining of cable sections using loads.

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.

*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.

14.31 DA card

With this card some data like near fields or S-parameters can be exported to additional ASCII files. The card allows to switch this export on and off, and affects only cards for the computation of, for example, near fields or S-parameters that follow the DA card. By default no such export files are created.

Furthermore, normally FEKO writes all the results to the ASCII *.out file. For large datasets (e.g. full far field patterns at many frequency points) this *.out file can become very large. In order to display results in POSTFEKO only the binary output file (*.bof file) is required, and therefore to reduce the size of the *.out file it is possible to switch off certain results in that file (a header is still written so that one knows what type of computations were requested).

DA - Write additional data files

Write electric fields to—

☐ *.efe file

☒ *.out file

Write magnetic fields to—

☐ *.hfe file

☒ *.out file

Write far fields to—

☐ *.ffe file

☒ *.out file

Write currents/charges to—

☐ *.os/*.ol files

☒ *.out file

☐ Write residue of iterative solution to *.cgm file

☐ Write S-parameters to Touchstone *.snp file

☐ Write spherical wave expansion to TICRA *.sph file

☐ Write near fields to SEMCAD *.dat file

☒ Write error estimates to the *.out file

☐ Write generalised S-parameter matrix to FEST3D *.chr file

Parameters:

Write electric fields to: The electric field strength can be exported into a *.efe file. The output of this result type to the *.out file can also be deactivated here.

Write magnetic fields to: The magnetic field strength can be exported into a *.hfe file. The output of this result type to the *.out file can also be deactivated here.

Write far fields to: The far field can be exported into a *.ffe file. The output of this result type to the *.out file can also be deactivated here.

Write currents/charges to: The currents can be exported into a *.os/*.ol file. The output of this result type to the *.out file can also be deactivated here.

Write residue ...: The residue from the iterative algorithm used to solve the matrix equation is stored in a *.cgm file.

Write S-parameters ...: The S-parameters (see the SP card) are written to a file in Touchstone *.snp format (v1.0). The n here gives the number of ports.

Write spherical wave expansion ...: 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 that the far field computation with spherical wave expansion must be requested subsequently (FF card (see section 14.37)).

*Write near fields to SEMCAD *.dat file:* The near fields can be exported into a SEMCAD *.dat file. The output of this result type to the SEMCAD *.dat file can also be deactivated here.

*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.

More than one DA card is allowed in one input file. Thus, using the following sequence of control cards, with the appropriate options only certain blocks will be saved to the data files:

```
DA ...    ** Write near fields on
FE ...
DA ...    ** Write near fields off
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.

14.31.1 General File Format

The structure of the following file formats share a common structure and will be explained each in turn: *.efe, *.hfe, *.ffe, *.ol and *.os. The general structure can be described as follows:

Header Block The header block contains general information on the file. It includes information such as the file type, format, how and when the file was written, etc. 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 format is then “##Key: Value” for each header block line.

Comments Comments may be placed anywhere in the file. They are denoted by two asterisks (“**”), indicating that the rest of that line must be ignored.

Solution Block Any number of solution blocks (typically one per request per frequency) can now follow. A solution block can further be broken down into:

Solution Block Header This section contains information that describes the data block and includes information such as the frequency, coordinate system, the request name, column headers, etc. 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 (e.g. 1.23E-001).

The column headers are a part of the solution block header and must be presented 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"
```

Note that they differ from other solution block header lines in that they do not have a key / value pair format. Column header lines still start with a single hash (“#”), but is then followed by the column titles surrounded in quotation marks.

14.31.2 *.efe / *.hfe file: Electric / Magnetic near fields

The following fields are available in the *Header Block*:

Key	Required	Description
File Type	Yes	Describes the type of the file. For electric near fields, the value must be: Electric near field and for magnetic near fields: Magnetic near field
File Format	No	Denotes the file syntax version (e.g. "1.0"). If not present it defaults to version 1.0 (files pre-dating Suite 6.1). The initial new file format version will then be "2.0".
Source	No	Denotes the base filename of the source where this data comes from.
Date	No	Date of data export in format "YYYY-MM-DD hh:mm:ss" (i.e. 24-hour format)

The following fields are available in the *Solution Block Header*:

Key	Required	Description
Request Name	No	The explicit name given to that solution request (as denoted in the *.pre file). If none is specified, then during importing such files in POSTFEKO use a default name of "request_N" (where "_N" is replaced with a number for each unnamed request).
Frequency	Yes	Frequency in Hz for which the following data was measured / computed.
Coordinate System	No	Coordinate system in which the axes are defined: • Cartesian [default] • Cylindrical • Cylindrical (x axis) • Cylindrical (y axis) • Conical
Origin	No	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	No	Indicates a point on the <i>u</i> -axis relative to the Origin. If none is specified, it is assumed that the <i>u</i> -axis coincides with the <i>x</i> -axis.
v-Vector	No	Indicates a point on the <i>v</i> -axis relative to the Origin. If none is specified, it is assumed that the <i>v</i> -axis coincides with the <i>y</i> -axis.

No. of [\$\$\$] Samples	Yes	The number of samples in each axis direction. The [\$\$\$] term is replaced by: • X / U • Y / V • Z / N • Phi • Theta • Rho • Radius • Cuboid
Result Type	No	For electric near fields (*.efe), this specifies whether the output is: • Electric Field Values [default] • Magnetic Vector Potential • Gradient of Scalar Electric Potential For magnetic near fields (*.hfe), this specifies whether the output is: • Magnetic Field Values [default] • Electric Vector Potential • Gradient of Scalar Magnetic Potential
No. of Header Lines	No	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 the header lines (i.e. the column titles), the format in the column header format section (see section 14.31.1) should be used. The specific column headers for the different near field coordinate systems are also listed.

The default column headers will all have the same structure. The column headers will 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 is now given:

Cartesian:

```
"X" "Y" "Z" "Re($$$x)" "Im($$$x)" "Re($$$y)" "Im($$$y)" "Re($$$z)" "Im($$$z)"
```

Cylindrical (X axis):

```
"Rho" "Phi" "X" "Re($$$rho)" "Im($$$rho)" "Re($$$phi)" "Im($$$phi)" "Re($$$x)" "Im($$$x)"
```

Cylindrical (Y axis):

```
"Rho" "Phi" "Y" "Re($$$rho)" "Im($$$rho)" "Re($$$phi)" "Im($$$phi)" "Re($$$y)" "Im($$$y)"
```

Cylindrical:

```
"Rho" "Phi" "Z" "Re($$$rho)" "Im($$$rho)" "Re($$$phi)" "Im($$$phi)" "Re($$$z)" "Im($$$z)"
```

Spherical:

```
"Radius" "Theta" "Phi" "Re($$$r)" "Im($$$r)" "Re($$$theta)" "Im($$$theta)" "Re($$$phi)" "Im($$$phi)"
```

Conical:

```
"Rho" "Phi" "Z" "Re($$$rho)" "Im($$$rho)" "Re($$$phi)" "Im($$$phi)" "Re($$$z)" "Im($$$z)"
```

All of the coordinate system descriptions given contain a section of text containing \$\$\$\$. 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:

(<i>*.efe</i>)	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
(<i>*.hfe</i>)	Magnetic near field files
H	for magnetic field values
F	for electric vector potential values
grad(PHI)	for gradient of the scalar magnetic potential values

14.31.3 *.ffe file: Far fields

The following fields are available in the *Header Block*:

Key	Required	Description
File Type	Yes	Describes the type of the file. For far fields, the value must be: Far field.
File Format	No	Denotes the file syntax version (e.g. "1.0"). If not present it defaults to version 1.0 (files pre-dating Suite 6.1). The initial new file format version will then be "2.0".
Source	No	Denotes the base filename of the source where this data comes from.
Date	No	Date of data export in format "YYYY-MM-DD hh:mm:ss" (i.e. 24-hour format)

The following fields are available in the *Solution Block Header*:

Key	Required	Description
Request Name	No	The explicit name given to that solution request (as denoted in the *.pre file). If none is specified, then during importing such files in POSTFEKO use a default name of "request_N" (where "_N" is replaced with a number for each unnamed request).
Frequency	Yes	Frequency in Hz for which the following data was measured / computed.
Origin	No	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	No	Indicates a point on the <i>u</i> -axis relative to the Origin. If none is specified, it is assumed that the <i>u</i> -axis coincides with the <i>x</i> -axis.
v-Vector	No	Indicates a point on the <i>v</i> -axis relative to the Origin. If none is specified, it is assumed that the <i>v</i> -axis coincides with the <i>y</i> -axis.
No. of [\$\$\$] Samples	Yes	The number of samples in each axis direction. The [\$\$\$] term is replaced by: • Phi • Theta

Result Type	No	For far fields (*.ffe), this specifies whether the output is: • Gain [default] • Directivity • RCS • Unknown
Incident Wave Direction	Yes	This is a (Theta,Phi) pair indicating the angle where an infinite plane source is originating from. Note that this field is <i>only</i> required if the Result Type is RCS.
No. of Header Lines	No	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 the header lines (i.e. the column titles), the format in the column header format section (see section 14.31.1) should be used. The specific column headers for far fields are also listed following this table.

The default column headers will all have the same structure. The column headers will depend on the far field type that was defined, which can be found in the Result Type value. The following column headers are defined for each result type:

Gain:

```
"Theta" "Phi" "Re(Etheta)" "Im(Etheta)" "Re(Ephi)" "Im(Ephi)" "Gain(Theta)" "Gain(Phi)" "Gain(Total)"
```

Directivity:

```
"Theta" "Phi" "Re(Etheta)" "Im(Etheta)" "Re(Ephi)" "Im(Ephi)" "Directivity(Theta)" ...
... "Directivity(Phi)" "Directivity(Total)"
```

RCS (i.e. either a bistatic or monostatic radar cross-section problem):

```
"Theta" "Phi" "Re(Etheta)" "Im(Etheta)" "Re(Ephi)" "Im(Ephi)" "RCS(Theta)" "RCS(Phi)" "RCS(Total)"
```

Unknown (i.e. the results do not pertain to an antenna problem, nor to an RCS problem):

```
"Theta" "Phi" "Re(Etheta)" "Im(Etheta)" "Re(Ephi)" "Im(Ephi)"
```

14.31.4 *.ol file: Surface charge density

The following fields are available in the *Header Block*:

Key	Required	Description
File Type	Yes	Describes the type of the file. For surface charge density, the value must be: Charges.
File Format	No	Denotes the file syntax version (e.g. "1.0"). If not present it defaults to version 1.0 (files pre-dating Suite 6.1). The initial new file format version will then be "2.0".
Source	No	Denotes the base filename of the source where this data comes from.
Date	No	Date of data export in format "YYYY-MM-DD hh:mm:ss" (i.e. 24-hour format)

The following fields are available in the *Solution Block Header*:

Key	Required	Description
Request Name	No	The explicit name given to that solution request (as denoted in the *.pre file). If none is specified, then during importing such files in POSTFEKO use a default name of "request_N" (where "_N" is replaced with a number for each unnamed request).
Frequency	Yes	Frequency in Hz for which the following data was measured / computed.
No. of [\$\$\$] Samples	Yes	The number of mesh elements that follow. The [\$\$\$] term is replaced by: • Electric Charge Triangle • Segment Charge
No. of Header Lines	No	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 the header lines (i.e. the column titles), the format in the column header format section (see section 14.31.1) should be used. The specific column headers for far fields are also listed following this table.

For the column headers of charges, the only difference is an optional additional field that stores the total surface area of a triangle vs. the total length of a segment.

Triangles:

```
"Num" "X" "Y" "Z" "Re(Q)" "Im(Q)" [optional] "Surface Area"
```

Segments:

```
"Num" "X" "Y" "Z" "Re(Q)" "Im(Q)" [optional] "Length"
```

14.31.5 *.os file: Surface current density

The following fields are available in the *Header Block*:

Key	Required	Description
File Type	Yes	Describes the type of the file. For surface charge density, the value must be: Currents.
File Format	No	Denotes the file syntax version (e.g. "1.0"). If not present it defaults to version 1.0 (files pre-dating Suite 6.1). The initial new file format version will then be "2.0".
Source	No	Denotes the base filename of the source where this data comes from.
Date	No	Date of data export in format "YYYY-MM-DD hh:mm:ss" (i.e. 24-hour format)

The following fields are available in the *Solution Block Header*:

Key	Required	Description
Request Name	No	The explicit name given to that solution request (as denoted in the *.pre file). If none is specified, then during importing such files in POSTFEKO use a default name of "request_N" (where "_N" is replaced with a number for each unnamed request).
Frequency	Yes	Frequency in Hz for which the following data was measured / computed.
No. of [\$\$\$] Samples	Yes	The number of mesh elements that follow. The [\$\$\$] term is replaced by: • Electric Current Triangle • Magnetic Current Triangle • Segment Current
No. of Header Lines	No	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 the header lines (i.e. the column titles), the format in the column header format section (see section 14.31.1) should be used. The specific column headers for far fields are also listed following this table.

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)"
```

14.31.6 Other supported file formats

The structure of the remainder of the data files is described below:

- ***.cgm file** In this file the number of iterations is given and the resulting residue from the iterative solving process of the matrix equation.
- ***.snp file** The Touchstone S-parameter file name contains the number of ports in the model. The extension is *.s1p for a 1-port, *.s2p for a 2-port and so on. The file contains a header (following the # character) which 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:  f,  |S11|, ∠S11
2-port:  f,  |S11|, ∠S11, |S21|, ∠S21, |S12|, ∠S12, |S22|, ∠S22
3-port:  f,  |S11|, ∠S11, |S12|, ∠S12, |S13|, ∠S13
          |S21|, ∠S21, |S22|, ∠S22, |S23|, ∠S23
          |S31|, ∠S31, |S32|, ∠S32, |S33|, ∠S33
4-port:  f,  |S11|, ∠S11, |S12|, ∠S12, |S13|, ∠S13, |S14|, ∠S14
          |S21|, ∠S21, |S22|, ∠S22, |S23|, ∠S23, |S24|, ∠S24
          |S31|, ∠S31, |S32|, ∠S32, |S33|, ∠S33, |S34|, ∠S34
          |S41|, ∠S41, |S42|, ∠S42, |S43|, ∠S43, |S44|, ∠S44

```

where $|S_{11}|$ is the absolute value and $\angle S_{11}$ the phase (in degrees) of the given parameter. Note that 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.

- ***.sph file** This file is using the native SWE file format of TICRA as used in their code GRASP

14.32 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. The DI card is used for the MoM/MLFMM when using the surface current or volume current methods, or also for the FEM.

Parameters:

Material label: Label of the material to be defined.

Define dielectric medium: A dielectric medium can be defined by specifying frequency independent parameters, a frequency dependent dielectric model or loading points from a file.

Define metallic medium: A metallic medium can be defined by specifying frequency independent parameters or loading points from a file.

Define impedance sheet: An impedance sheet can be defined by specifying the frequency independent real and imaginary part or loading points from a file.

Source: The medium is defined by means of a wideband model or loading points from a file (see section 6.1.1).

Define dielectric medium

With this option a user specify the dielectric model used to define a dielectric medium.

DI - Set dielectric properties

Material label

Define medium

☒ Define dielectric medium

☐ Define metallic medium

☐ Define impedance sheet

Source

☒ Wideband model

☐ Load points from file (linear interpolation)

Dielectric modelling

Debye relaxation

$$\epsilon^* = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1 + j\frac{f}{f_r}}$$

Relative static permittivity ϵ_s

Relative high frequency permittivity ϵ_∞

Relaxation frequency f_r

Magnetic modelling

Constant (Frequency independent)

Relative permeability μ_r

Magnetic loss tangent $\tan\delta_\mu$

Mass density (kg/m³) ρ

Dielectric modelling: Constant (Frequency independent)

Relative permittivity Relative permittivity ϵ_r of the medium.

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 $\tan\delta = \frac{\sigma}{\omega\epsilon_r\epsilon_0}$ and have different frequency behaviour).

Conductivity Conductivity σ in $\frac{1}{\Omega\text{m}}$ of the medium.

Dielectric modelling: Debye relaxation

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.

Dielectric modelling: Cole-Cole

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 α of the medium.

Dielectric modelling: Havrillak-Negami

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 α of the medium.

Phase factor Phase factor β of the medium.

Dielectric modelling: Djordevic-Sarkar

Variation of relative permittivity Variation of the relative permittivity $\Delta\epsilon$ of the medium.

Relative high frequency permittivity Relative high frequency permittivity ϵ_∞ of the medium.

Conductivity Conductivity σ in $\frac{1}{\Omega\text{m}}$ of the medium.

Lower limit of angular frequency The lower limit of the angular frequency for the medium, ω_1 .

Upper limit of angular frequency The upper limit of the angular frequency for the medium, ω_2 .

Dielectric modelling: Specify points in the *.pre file.

Specify points in the *.pre file. The dielectric properties of the medium are specified in the *.pre file.

Dielectric modelling: Load points from file (linear interpolation)

A dielectric medium may also be defined by importing data points from file (see section 6.1.1).

Define magnetic medium

With this option a user specify the magnetic model used to define a dielectric or metallic medium.

DI - Set dielectric properties

Material label

Define medium

☐ Define dielectric medium

☒ Define metallic medium

☐ Define impedance sheet

Source

☒ Wideband model

☐ Load points from file (linear interpolation)

Magnetic modelling

Specify points in the *.pre file (linear interpolation)

Frequency

Number of points

Point 1	
Frequency	
Relative permeability	
Magnetic loss tangent	
Conductivity	

Magnetic modelling: Constant (Frequency independent)

Relative permeability Relative permeability μ_r of the medium.

Magnetic loss tangent Magnetic loss tangent $\tan \delta_\mu$ of the medium specified in *Set properties for medium number* (the complex permeability is then given by $\mu = \mu_0 \mu_r (1 - j \tan \delta_\mu)$).

Magnetic modelling: Constant (Frequency independent)

Non magnetic The relative permeability μ_r of the medium is set equal to 1 and the magnetic loss tangent $\tan \delta_\mu$ is set equal to 0.

Magnetic modelling: Specify points in the *.pre file

Specify points in the *.pre file. The magnetic properties of the medium are specified in the *.pre file.

Magnetic modelling: Load points from file (linear interpolation)

A magnetic medium may also be defined by importing data points from file (see section 6.1.1)

DI - Set dielectric properties

Material label

Define medium

☐ Define dielectric medium

☐ Define metallic medium

☒ Define impedance sheet

Source

☒ Wideband model

☐ Load points from file (linear interpolation)

Surface impedance (Ohm)

Constant (Frequency independent)

Real part

Imaginary part

Define impedance sheet

With this option an arbitrary user defined complex surface impedance $Z_{\square}$ can be used in FEKO. It must be noted that the impedance boundary condition for the MoM (also then for MLFMM etc.) has certain limitations regarding the range of validity. FEKO uses whatever the user has specified as surface impedance, and it is the responsibility of the user to ensure that the application of an IBC is still warranted for the specific configuration (dependent on the impedance value as such, but also frequency, radius of curvature of the structure etc.).

For surfaces the unit of $Z_{\square}$ is Ω . For wire structures, the value used by FEKO is in units of $\frac{\Omega}{m}$ and results from the surface impedance expression by dividing it by $\pi \varrho$ where ϱ represents the wire radius.

Real part: The real part of the surface impedance $Z_{\square}$ in Ω (for triangles) or in $\frac{\Omega}{m}$ for wires.

Imaginary part: The imaginary part of the surface impedance $Z_{\square}$ in Ω (for triangles) or in $\frac{\Omega}{m}$ for wires.

Load points from file (linear interpolation): An impedance sheet may also be defined by importing from file (see section [6.1.1](#)).

14.33 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.

Parameters:

Layered medium type: In this group the user must select one of two options:

- Isotropic layered medium
- Anisotropic layered medium

Isotropic layered medium

The screenshot shows the 'DL - Define layered medium properties' dialog box. The 'Layered dielectric label' field is empty. Under 'Layered medium type', the 'Isotropic layered medium' radio button is selected. The 'Number of layers' is set to 1. A table for 'Layer 1' is visible with columns for 'Thickness of this layer' and 'Material label (dielectric)'. The 'Material label (dielectric)' field is empty.

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 m (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

The screenshot shows the 'DL - Define layered medium properties' dialog box. The 'Layered dielectric label' field is empty. Under 'Layered medium type', the 'Anisotropic layered medium' radio button is selected. The 'Number of layers' is set to 1. A table for 'Layer 1' is visible with columns for 'Thickness of this layer', 'Angle of principle direction', 'Material in principle direction', and 'Material in orthogonal direction'. All four fields in the table are empty.

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.

14.34 EE card

This 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.

Parameters:

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 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.

14.35 EN card

This card indicates the end of the input file. It is essential and has no parameters.

14.36 FE card

This card controls the calculation of the near fields.

The dialog box is titled "FE - Calculate the near fields". It contains a dropdown menu labeled "Electric field values". Below it is a checkbox labeled "Calculate only the scattered part of the field". A section titled "Coordinate system" contains several radio buttons: "Cartesian" (selected), "Cylindrical (y-axis)", "Cylindrical", "Conical", "Spherical", and "Specified points". To the left of these buttons is a 3D coordinate system diagram with x, y, and z axes. Below the diagram are two groups of input fields. The first group, labeled "No. of points", has fields for x, y, and z. The second group, labeled "Increment", also has fields for x, y, and z.

Parameters:

First drop-down list: Select what to calculate:

- *No field calculation:* Field is not calculated.
- *Electric field values:* Calculate the electric field, $\vec{E}$.
- *Magnetic field values:* Calculate the magnetic field, $\vec{H}$.
- *Both electric field and magnetic field values:* Calculate both electric and magnetic fields.
- *Electric field and SAR values in cuboids:* The electric field and SAR values in the dielectric volume elements. For this option, no other parameters are required.
- *Magnetic vector potential:* Calculate the magnetic vector potential, $\vec{A}$.
- *Gradient of the scalar electric potential:* Calculate the gradient of the scalar electric potential, $\nabla \varphi$.
- *Electric vector potential:* Calculate the electric vector potential, $\vec{F}$.
- *Gradient of the scalar magnetic potential:* Calculate the gradient of the scalar magnetic potential, $\nabla \psi$.

Calculate only the scattered part of the field: When this item is checked, only the scattered part of the field / potential is written to the output file. Otherwise the total field / potential, the sum of the scattered and source contributions, are written to the output file. Note that depending on the used formulation in FEKO the region where the incident field as computed from the impressed sources is present might be different (for instance when using the surface equivalence formulation in the MoM to model dielectric bodies, then each source acts as incident field only in the medium where this source is located).

Coordinate system: In this dialog, the coordinate system for the calculation of the requested fields is specified.
If *Specified points* is selected here, additional dialogs for the *Number of field*

points and the *Coordinates* are shown. One must then enter the coordinates of each point.

For all other cases, additional dialogs for *Starting values*, *Increment* and *No. of points* are shown — each with three values for the three components of the respective coordinate system.

Use old output format: If this item is checked, the old format of the near field is used in the *.out 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).

Note that 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 selected (default behaviour and highly recommended) in the SF card — in this case coordinates must be in metre after scaling.

Potentials cannot be computed with the FE card if UTD or PO is used. Also only the free space Green's function is supported - not the Green's functions for layered spheres or multilayered planar media.

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. For a magnetic dipole (A6 card) the electric ring current model yields $\vec{A}$ and the magnetic current yields $\vec{F}$ and $\nabla \psi$; all the other potentials are zero.

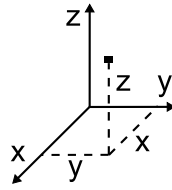
If one requests *.efe and/or *.hfe files with the DA card, then $\vec{A}$ and $\nabla \varphi$ are written to the *.efe file, while $\vec{F}$ and $\nabla \psi$ are written to the *.hfe file.

If a ground plane is used, a calculation of the near fields in the ground plane is not possible. The observation points in the area $z < 0$ are not taken into account.

It should be noted that the coordinates may have an offset (OF card). Thus the near field on the surface of a sphere can be calculated, with the centre of the sphere not being located at the origin of the coordinate system.

The different *Coordinate systems* are described on the following pages.

- *Cartesian* coordinates x, y, z

Figure 14-19: Field calculation in the *Cartesian* coordinate system

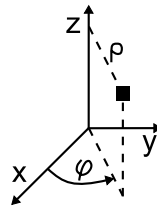
Observation Point:

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (14-50)$$

Unit vectors of the coordinate system:

$$\hat{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \hat{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (14-51)$$

- *Cylindrical* coordinates around z axis ρ, φ, z

Figure 14-20: Field calculation in the *Cylindrical* coordinate system

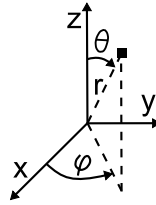
Observation Point:

$$\vec{r} = \begin{bmatrix} \rho \cos \varphi \\ \rho \sin \varphi \\ z \end{bmatrix} \quad (14-52)$$

Unit vectors of the coordinate system:

$$\hat{\rho} = \begin{bmatrix} \cos \varphi \\ \sin \varphi \\ 0 \end{bmatrix} \quad \hat{\varphi} = \begin{bmatrix} -\sin \varphi \\ \cos \varphi \\ 0 \end{bmatrix} \quad \hat{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (14-53)$$

- *Spherical coordinates* r, ϑ, φ

Figure 14-21: Field calculation in the *Spherical coordinate system*

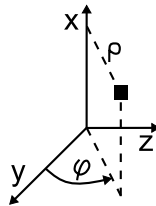
Observation Point:

$$\vec{r} = \begin{bmatrix} r \sin \vartheta \cos \varphi \\ r \sin \vartheta \sin \varphi \\ r \cos \vartheta \end{bmatrix} \quad (14-54)$$

Unit vectors of the coordinate system:

$$\hat{r} = \begin{bmatrix} \sin \vartheta \cos \varphi \\ \sin \vartheta \sin \varphi \\ \cos \vartheta \end{bmatrix} \quad \hat{\vartheta} = \begin{bmatrix} \cos \vartheta \cos \varphi \\ \cos \vartheta \sin \varphi \\ -\sin \vartheta \end{bmatrix} \quad \hat{\varphi} = \begin{bmatrix} -\sin \varphi \\ \cos \varphi \\ 0 \end{bmatrix} \quad (14-55)$$

- *Cylindrical coordinates* around the x axis r, φ, x

Figure 14-22: Field calculation in the *Cylindrical coordinate system* around the x axis

Observation Point:

$$\vec{r} = \begin{bmatrix} x \\ \rho \cos \varphi \\ \rho \sin \varphi \end{bmatrix} \quad (14-56)$$

Unit vectors of the coordinate system:

$$\hat{\rho} = \begin{bmatrix} 0 \\ \cos \varphi \\ \sin \varphi \end{bmatrix} \quad \hat{\varphi} = \begin{bmatrix} 0 \\ -\sin \varphi \\ \cos \varphi \end{bmatrix} \quad \hat{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (14-57)$$

- *Cylindrical* coordinates around the y axis r, φ, y

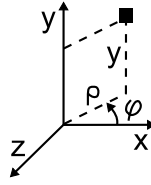


Figure 14-23: Field calculation in the *Cylindrical coordinate system* around the y axis

Observation Point:

$$\vec{r} = \begin{bmatrix} \rho \cos \varphi \\ y \\ -\rho \sin \varphi \end{bmatrix} \quad (14-58)$$

Unit vectors of the coordinate system:

$$\hat{\rho} = \begin{bmatrix} \cos \varphi \\ 0 \\ -\sin \varphi \end{bmatrix} \quad \hat{\varphi} = \begin{bmatrix} -\sin \varphi \\ 0 \\ -\cos \varphi \end{bmatrix} \quad \hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (14-59)$$

- *Conical* coordinates around the z axis φ, z

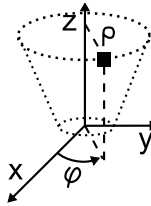


Figure 14-24: Field calculation in the *Conical coordinate system*

This option is similar to the field calculation in cylindrical coordinates around the z axis, where the radius r changes with the height z

$$r(z) = r_0 + \frac{\Delta r}{\Delta z} \cdot (z - z_0), \quad (14-60)$$

and z lies within the range $z_0 \dots z_0 + n_z \cdot \Delta z$.

Observation Point:

$$\vec{r} = \begin{bmatrix} \left(r_0 + \frac{\Delta r}{\Delta z} \cdot (z - z_0) \right) \cdot \cos \varphi \\ \left(r_0 + \frac{\Delta r}{\Delta z} \cdot (z - z_0) \right) \cdot \sin \varphi \\ z \end{bmatrix} \quad (14-61)$$

Unit vectors of the coordinate system:

$$\hat{r} = \begin{bmatrix} \cos \varphi \\ \sin \varphi \\ 0 \end{bmatrix} \quad \hat{\varphi} = \begin{bmatrix} -\sin \varphi \\ \cos \varphi \\ 0 \end{bmatrix} \quad \hat{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (14-62)$$

14.37 FF card

This card controls the calculation of the far fields in spherical coordinates.

Parameters:

No calculation: If this item is checked, no calculation is done.

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).

Only integrate field over area given below: The far field is calculated but it is not written to the output 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 one is just interested in the modal coefficients. 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. This option is only meaningful if only the scattered field is required. Normally one would not check this item so that the total field is calculated. This includes all source contributions except plane wave excitations.

Directivity / Gain: Select which quantity is required.

Number of θ points: The number of observation points in the θ direction. An empty field will be set to 1.

Number of φ points: The number of observation points in the φ direction. An empty field will be set to 1.

Initial θ : The angle θ_0 in degrees of the first observation point.

Initial φ : The angle φ_0 in degrees of the first observation point.

ϑ increment: Increment $\Delta\vartheta$ in degrees of the angle ϑ .

φ increment: Increment $\Delta\varphi$ in degrees of the angle φ .

Compute spherical mode coefficients: Check this item if the spherical mode coefficients of the far field should be calculated with the FF card. The *Maximum mode index N* should be specified if this option is chosen. More information on this option is supplied below.

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.

Information regarding the output format and derived quantities available from fields calculated at an FF card can be found in FEKO User Manual (section 19.6).

When calculating the monostatic radar cross section for a number of directions of incidence, the parameter *Fields calculated only in the incident direction* is necessary, otherwise *Fields calculated as specified below* can be used.

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:

- $\vartheta_0 - \frac{1}{2} \cdot \Delta\vartheta \leq \vartheta \leq \vartheta_0 + (N_\vartheta - \frac{1}{2}) \cdot \Delta\vartheta$ and $\varphi_0 - \frac{1}{2} \cdot \Delta\varphi \leq \varphi \leq \varphi_0 + (N_\varphi - \frac{1}{2}) \cdot \Delta\varphi$
- $\vartheta_0 \leq \vartheta \leq \vartheta_0 + (N_\vartheta - 1) \cdot \Delta\vartheta$ and $\varphi_0 \leq \varphi \leq \varphi_0 + (N_\varphi - 1) \cdot \Delta\varphi$

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 instance 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      #delta #delta
```

The output in the *.out file then reads for example

```
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 problem is symmetrical, it is not necessary to carry out the integration over the complete 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 a ground plane has been specified, the calculation of the far fields below the ground plane is not possible. Observation points with $z < 0$ will thus be ignored.

Spherical mode coefficients

can be computed while doing a far field computation. The calculation of these coefficients is based on the far field values and they are written to the FEKO output file.

Doing the integration requires a two-dimensional far field computation (i.e. both *Number of ϑ points* and *Number of φ points* larger than 1) over the whole sphere (i.e. $0 \leq \vartheta \leq 180^\circ$ and $0 \leq \varphi \leq 360^\circ$) and the angular increments should be chosen according to the desired accuracy and number of modes (i.e. a finer stepping is required for higher modes). It is suggested to do some initial convergence study (e.g. with increments 5° and then 1°) to see the sensitivity of the results.

Spherical modes have three indices s , m , and n with $s = 1$ for TE-modes, $s = 2$ for TM-modes, $m = -N \dots N$, and $n = 1 \dots N$ (see also more detailed description at the AS card (see section 14.20), including a one-dimensional compressed indexing scheme $j = 1 \dots J$ and the normalisation of the modes etc.). The input parameter *Maximum mode index* N determines the maximum mode index N , i.e. a total number of $J = 2N(N + 2)$ modes will be computed.

The modes origin is the same as for the far field computation in general (i.e. this is the global origin unless an OF card has been used to specify an offset). It should also just be mentioned that due to the nature of the far field (propagating towards $r = \infty$) all computed mode coefficients refer to spherical cylinder functions $z_n^{(c)}$ with type $c = 4$ (see more details at the AS card).

It shall be mentioned that when spherical modes are computed with the FF card, the DA card can also be used (must be in front of the FF card) in order to request that this spherical mode expansion is exported to an SWE file (extension *.sph) which can be imported into the computer code GRASP from TICRA. GRASP is a reflector antenna modelling code, and by means of this SWE file export one can for instance model a horn antenna as feed in FEKO and then export this feed structure and use in GRASP. Note that the gain in GRASP will only be correct if the radiated power in FEKO has been set to 4π Watts. The spherical mode expansion coefficients Q_{smn} as used by FEKO are described at the AS card (see section 14.20). In GRASP a slightly different convention is used:

- An additional factor $\frac{1}{\sqrt{8\pi}}$ is used
- The coefficients are conjugate complex (i.e. GRASP assumes an $e^{-i\omega t}$ time dependency)
- The index m is exchanged with $-m$ (since in FEKO the φ dependency is defined differently than in GRASP, $e^{jm\varphi}$ versus $e^{-jm\varphi}$)

These conversions are done automatically by FEKO when exporting the SWE file, so that this can readily be imported into GRASP.

14.38 FR card

This card sets the frequency/frequencies (in Hz), at which the solution will be obtained.

The solution can be done for a single frequency, a loop of discrete frequencies (linear or multiplicative stepping), 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: If this item is selected the following parameters are applicable:

- *Number of frequency points:* For a discrete frequency 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, i.e. the new frequency is the previous value plus the frequency increment. If *Multiplicative* is selected, then consecutive frequencies differ by a constant factor, i.e. 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. This, 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.

Continuous data: Select this item to use an adaptive frequency interpolation technique to obtain a continuous representation of the results in the given frequency band (see FEKO User Manual (section 21)). When using this feature, the remaining parameters have the following meaning:

- *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.
- *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.
- *Select quantities to include for adaptive frequency sampling:* The quantities selected will be included in the adaptive frequency sampling.
- *For CableMod/CRIPTE/PCBMod/Touchstone . . . , Number of discrete frequencies . . . :* 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 / ending frequency:* Defines the frequency range.
- *Min. frequency stepping:* 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:

$$\frac{1}{10000}(\text{Ending frequency} - \text{Starting frequency}) \quad (14-63)$$

If a discrete loop with more than 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):

$$\Delta_f = \frac{f_2 - f_1}{N_f - 1} \quad (14-64)$$

- for a multiplicative frequency scale (multiplicative increments):

$$fac = \left(\frac{f_2}{f_1} \right)^{\frac{1}{N_f - 1}} \quad (14-65)$$

where Δ_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 Δ_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.

14.39 GF card

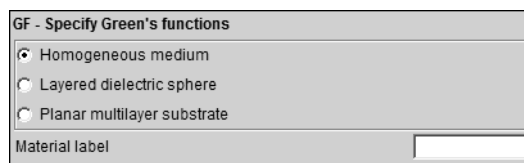
With this card the Green's function may be selected. The Green's function relates the fields in space to the sources present. The propagation space is usually free space, and FEKO uses the free space Green's function by default. For a few specific interaction environments, some of the complexities of the environments can be taken into account using special Green's functions. The Green's functions that FEKO supports are:

- *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 substrate can have metallic ground planes above and below it.

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.

The dialogs for each of the three cases above are discussed separately below.

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.

Metallic structures can have an arbitrary orientation (horizontal, vertical, and also diagonal). They can lie at an arbitrary position in one or more layers and can lie directly on the border of two layers (e.g. on the surface of a substrate). The only restriction is that no metallic segment or triangle may cross a boundary between layers, i.e. it must lie completely within one layer or at the boundary between layers.

If, for example, a metallic wire penetrates a multilayer substrate, the segmentation must be so that there is a node on each interface between layers. (See `example_31` in the *Script Examples*.)

The following is not possible with this Green's function:

- dielectric ground
- hybrid MoM/PO method
- hybrid MoM/UTD method
- dielectric bodies with the surface or volume equivalence principle

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 cellphone in front of a spherical shell model of the human head very efficiently.

Parameters:

Configuration list: The drop-down list allows selecting between *Single dielectric sphere* and a core with a number of layers. Note that whether metal structures are allowed, influences the number of layers that are allowed.

Allow metal structures inside sphere: When this item is checked metallic structures can be present in the inner parts of the sphere.

Use interpolation: When this item is checked interpolation (*.gfe and *.gfh files) is used to accelerate the computations.

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 m (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 outside.

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 (e.g. the hand of a user around a mobile phone), but the dielectric bodies must be outside the sphere.

An example of the use of the GF card for a sphere is given in example_15 (*Script Examples*).

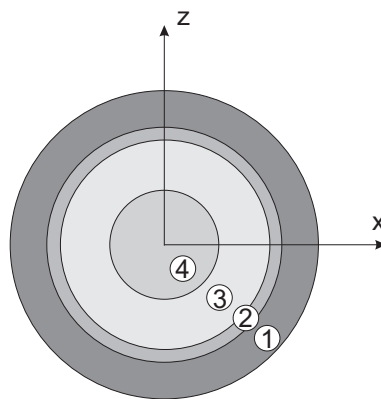


Figure 14-25: Example of a sphere consisting of 4 media (core and 3 layers) indicating the layer numbering.

Planar multilayer substrate

GF - Specify Green's functions

☐ Homogeneous medium
☐ Layered dielectric sphere
☒ Planar multilayer substrate

Diagram illustrating the planar multilayer substrate structure. The structure is defined by layers 0, 1, 2, ..., n, and infinity. The thickness of layer 1 is h_1 , and the thickness of layer 2 is h_2 . The z-axis is vertical, and the x-axis is horizontal.

Number of layers: 1

	Layer 0	Layer 1
Thickness	+infinite	-infinite
Material label		
Bottom ground plane	☐	☐

Z-value at the top of layer 1:

Advanced conducting ground plane definitions

Number of additional conducting ground planes: 0

Advanced ground planes at arbitrary z-values. These planes can be added at any point on the z-axis, irrespective of the layer definitions above.

z-value:

When this Green's function is selected, the EM interaction of a layered dielectric substrate located in the xy -plane is taken into account. The Green's function can also take the effect of a top and/or bottom ground plane 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 e.g. buried antennas.

Parameters:

Number of layers: Number of layers in the substrate (layer 0 — the upper half-space) is not included in the number.

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 layer.

Z-value at the top of layer 1: The value of the z coordinate at the transition between layer 0 and layer 1.

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.

As indicated in the figure of the card, layer 0 is the upper half-space, layer 1 is just beneath this, etc.

Metallic structures can have an arbitrary orientation (horizontal, vertical, and also diagonal). They can lie at an arbitrary position in one or more layers and can lie directly on the border of two layers (e.g. on the surface of a substrate). The only restriction is that no metallic segment or

triangle may cross a boundary between layers, i.e. it must lie completely within one layer or at the boundary between layers.

If, for example, a metallic wire penetrates a multilayer substrate, the segmentation must be so that there is a node on each interface between layers. (See `example_31` in the *Examples Guide*.)

The following is not possible with this Green's function:

- dielectric ground (BO Card)
- hybrid MoM/PO method
- hybrid MoM/UTD method
- hybrid MoM/FEM method
- dielectric bodies with the surface or volume equivalence principle

14.40 L2 card

The L2 card allows a load (complex impedance) to be placed on a node in the model.

L2 - Load node with a complex impedance

☒ Select segment ☐ Set load position

☒ Load at start of segment
☐ Load at end of segment

Segment label (see manual)

Real part of impedance (Ohm)

Imaginary part of impedance (Ohm)

Coordinates of node

x
y
z

Parameters:

Select segment: When this item is selected, then the *Segment label* field becomes active. Here one 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.

Source 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.

Real part of impedance: The real part of the complex impedance in Ω .

Imaginary part of impedance: The imaginary part of the complex impedance in Ω .

14.41 L4 card

This card can be used to add a load between a metallic triangle and the ground plane for the planar multilayer Green's function without having the requirement to model a vertical current element (analogous to the A4 excitation card).

Parameters:

Define a load at a coaxial attachment point: Define a load with the following parameters.

Remove all L4 type loads previously defined: This L4 card does not define a load, but rather all previously defined L4 loads are deleted. All the other input parameters of this card are ignored.

Select element: The label of the triangle to load. If there is more than one triangle with this label, the one with the highest element number is loaded.

Set source position: Alternatively, the user may specify the Cartesian coordinates x , y and z of the load point in the *Select source coordinates* group. The triangle with the centre point closest to this point is loaded.

Transform impedance to ground plane: The specified impedance refers to the metallic ground plane, and a transformation must be done to get the correct load impedance at the triangle.

Real part of impedance: Real part of the load impedance (in Ω).

Imaginary part: Imaginary part of the load impedance (in Ω).

Radius of the connection pin: The radius of the load pin in m.

The source coordinates (if entered) and the radius of the connection pin (if entered) is optionally scaled by the SF card.

If an L4 card is processed any existing L4 load on that triangle is replaced. For example, if two L4 cards are used directly below each other, the first specifying a 50 Ω and the second a 20 Ω load, the segment will be loaded with 20 Ω , not 70 Ω .

It must also be noted that if the L4 card is used in conjunction with an A4 card (impressed current source), the load impedance of the L4 card is placed parallel to the input impedance of the A4 source (it has no effect if it is in series with the current source), i.e. the resulting input admittance is the sum of input admittance without the L4 load, and the admittance added by the L4 card.

14.42 LC card

With the LC card, complex, series and parallel circuits can be applied between connector pins and also between a connector pin and ground can be defined.

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 number*. A load may also be removed between ground and a pin by checking the *PEC* checkbox.

Remove all cable loads previously defined: All previously defined LC type loads are removed.

Define/replace load between: A load can be placed between two connector pins by defining the *Connector label* and *Pin number*. A load may also be placed between ground and a pin by checking the *PEC* checkbox.

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.

For detailed conductor to cable connector pin relation, see the AK card (section [14.16](#)).

14.43 LD card

With this card it is possible to specify a distributed resistive, capacitive or inductive loading or even a series combination of these values for a segment.

LD - Distributed load	
Label of segments to load	
Resistance (Ohm/m)	
Inductance (H/m)	
Capacitance (F/m)	

Parameters:

Label of segments ...: 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

$$Z_s = \left(R' + j\omega L' + \frac{1}{j\omega C'} \right) \cdot l. \quad (14-66)$$

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 one. This card has no significance for surface elements, even when these are assigned the same label.

14.44 LE card

With the LE card an edge between surface triangles can be loaded with an impedance $Z = R + jX$, as shown in Figure 14-26.

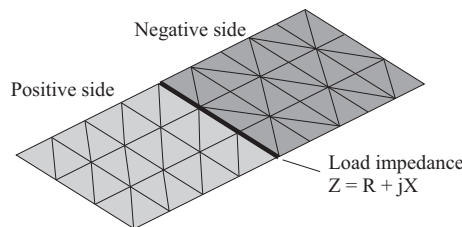


Figure 14-26: Application of the LE card

Parameters:

Load edge between regions with multiple labels: 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:* 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.
- *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: 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 the negative side of the load. Increasing this value will add rows to the table for the labels.
- *Negative side:* The labels of the triangles that are connected to the ground.

Load microstrip edge between two points: 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 below. 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.

Real part of impedance: Real part of the complex load impedance.

Imaginary part of impedance: Imaginary part of the complex load impedance.

See also the AE card for the excitation of such an edge. As shown in Figure 14-26 the edge can consist of several single edges between regions specified by labels (AE card (see section 14.13) provides a discussion of 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 (i.e. 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.

14.45 LF card

This card can be used to impress a complex impedance between two points inside a FEM mesh.

LF - Complex load impedance (FEM)		
Real part of impedance (Ohm)		
Imaginary part of impedance (Ohm)		
Load position		
	Start	End
x coordinate		
y coordinate		
z coordinate		

Parameters:

Real part of impedance: The real part of the complex impedance in Ω .

Imaginary part: The imaginary part of the complex impedance in Ω .

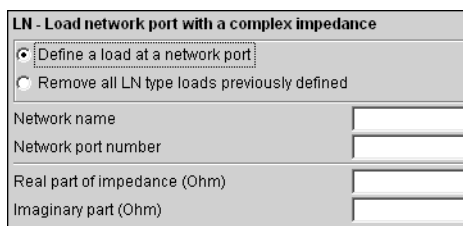
Load position: The Cartesian coordinates of the start and end points of the load.

The complex impedance is impressed between two points inside the FEM mesh. The line may be positioned arbitrarily inside the FEM mesh, i.e. 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.

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.

14.46 LN card

This card can be used to add a complex load to any non-radiating network port that is not connected to geometry (i.e. any non-radiating network of the type *Internal*).



LN - Load network port with a complex impedance

☒ Define a load at a network port
☐ Remove all LN type loads previously defined

Network name

Network port number

Real part of impedance (Ohm)

Imaginary part (Ohm)

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.

Network name: The network or transmission line name, with the network port number, uniquely identifies the connection terminals.

Network port number: The network port number, with the network or transmission line name, uniquely identifies the connection terminals.

Real part of impedance: The real part of the complex impedance in Ω .

Imaginary part: The imaginary part of the complex impedance in Ω .

14.47 LP card

This card can be used to assign discrete circuit elements (in parallel) to a segment. Figure 14-27 shows the parallel circuit that can be assigned to a segment.

LP - Load segment with parallel circuit	
Label of segments to load	
Resistor value (Ohm)	
Inductor value (H)	
Capacitor value (F)	

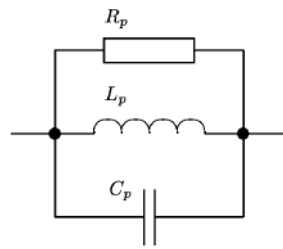


Figure 14-27: Sketch of the parallel circuit

Parameters:

Label of segments ...: All segments with this label are assigned the parallel 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 then given by

$$Z_p = \frac{1}{\frac{1}{R_p} + \frac{1}{j\omega L_p} + j\omega C_p} . \quad (14-67)$$

If the resistance is set to zero, then the resistance is interpreted as infinite, i.e. 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.

14.48 LS card

This card can be used to assign discrete circuit elements (in series) to a segment.

LS - Load segment with series circuit	
Label of segments to load	
Resistor value (Ohm)	
Inductor value (H)	
Capacitor value (F)	

Figure 14-28 shows this circuit that can be assigned to a segment.

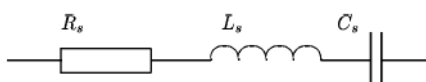


Figure 14-28: Sketch for the serial combination

Parameters:

Label of segments ...: 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

$$Z_s = R_s + j\omega L_s + \frac{1}{j\omega C_s} . \quad (14-68)$$

If a capacitance of zero is selected, it is interpreted as infinite capacitance, i.e. in the case of the series combination 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.

14.49 LZ card

This card can be used to assign a complex impedance to a segment.

LZ - Load segment with complex impedance	
Label of segments to load	
Real part of impedance (Ohm)	
Imaginary part (Ohm)	

Parameters:

Label of segments ...: All segments with this label are assigned the specified impedance.

Real part of impedance: The real part of the complex impedance in Ω .

Imaginary part: The imaginary part of the complex impedance in Ω .

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.

14.50 NW card

This card can be used to add a linear non-radiating network.

NW - Non-radiating general network

☐ Remove all existing networks
☒ New network
☐ Add to existing network

Network name:

Number of ports:

Port 1

Connected to:

Segment Label:

☐ Reverse connection orientation

Data type

☒ S-matrix
☐ Z-matrix
☐ Y-matrix
☐ SPICE *.cir file

Data source

☒ Load data from a Touchstone file
☐ The data follows in the *.pre file

File name:

Data matrix (Snm)

	Re(1)	Im(1)	Re(2)	Im(2)	Zref
1					
2					
3					
4					

The non-radiating general network allows the user to do combined analysis of electromagnetics with linear circuits (e.g. 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 problem solution can be found. The individual element solutions are without 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.

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 metre 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.

Data type: The network data can be specified S-parameters, Z-parameters, Y-parameters or SPICE *.cir file.

| Load data from Touchstone file: The network data can be loaded from a Touchstone file (in v1.0 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. A *.cir file is to be supplied containing the required SPICE circuit description. This description should include a subcircuit 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.

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 at each port.

It should be noted that it is not necessary to specify all possible connections. If, for example, NWName1.Port1 is connected to NWName2.Port1, it is not necessary to connect NWName2.Port1 to NWName1.Port1. The user should ensure that there is enough information to link all connected ports. If an internal port should be left open, then no connection should be entered.

14.51 OF card

This card specifies an offset for the origin of the coordinate system used for near and far field calculations. In addition it is possible to use only a part of the structure when calculating the fields (selected using labels).

OF - Options for field calculations

☐ Calculate near field or far field on in local coordinate system

☐ Use only some labels for field calculation

Start at label:

End at label:

Origin of offset coordinate:

x: y: z:

Rotation about the axes:

x: y: z:

Parameters:

Calculate near field on offset axis: Use the offset specified below as the origin of the coordinate system for field calculations.

Use only some labels for field calculation: 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 of the triangles lies in the specified range.)

Origin of offset coordinate: In this group the Cartesian coordinates of the transformed origin are specified. Each of x , y and z is scaled by the SF card if the SF card is used.

Rotation about the axis: The angle of rotation α_x around the x axis, the angle of rotation α_y around the y axis and the angle of rotation α_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 whose centre does not lie on 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.

14.52 OS card

With this card the currents on the surfaces and the segments can be extracted.

Parameters:

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.

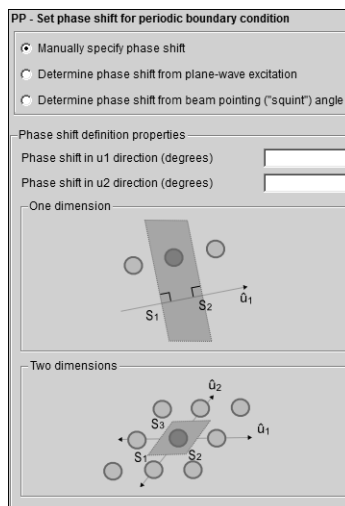
The options where a *.rsd file is written permit the creation of a *.rsd file for use with the transmission line simulation programs CableMod or CRIPTE or the PCB tool PCBMod. The currents along all or selected segments are exported to the *.rsd file (the file name without extension is the same as that 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) must 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, a *.os file will be created in addition to the currents written to the output file.

14.53 PP card

This card defines the phase shift of the excitation between one unit cell and the next when doing periodic boundary condition calculations. The unit cell for a periodic boundary condition calculation is specified using the PE card.



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 can not be specified, but is determined by the excitation.

Determine phase shift from beam pointing ("squin") angle: The phase shift is determined by specifying the theta and phi angle of the "squin" angle.

Phase shift in $\hat{u}_1$ direction: Phase shift in the first direction, $\hat{u}_1$.

Phase shift in $\hat{u}_2$ direction: Phase shift in the second direction, $\hat{u}_2$.

One dimension:

Theta angle (degrees): Orientation of the squint 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 - theta in degrees is the angle between the squint angle and the ($\hat{u}_1 = 0$) plane.

Phi angle (degrees): Orientation of the squint 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 (only one PE card).

14.54 PS card

This card is a general program control and can be used for instance to store the current expansion coefficients to a file or load them later again to speed up the solution.

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 etc.). Such elements can also be loaded again.

PS - Control data structure

Save / read matrix elements

- ☒ No data files (normal execution)
- ☐ Save matrix elements to *.mat file
- ☐ Read matrix elements from *.mat file
- ☐ Read *.mat file if it exists, else create it

Save / read LU decomposed matrix

- ☒ No data files (normal execution)
- ☐ Save LU-decomposition to *.lud file
- ☐ Read LU-decomposition from *.lud file
- ☐ Read *.lud file if it exists, else create it

Save / read currents

- ☒ No data files (normal execution)
- ☐ Save currents to *.str file
- ☐ Read currents from *.str file
- ☐ Read *.str file if it exists, else create it

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 subsequently with multiple different excitations (i.e. right-hand sides). Note that if you do this in one FEKO run (i.e. one *.pre file), then FEKO keeps the LU-decomposition automatically in memory. This option is useful if you want to solve for multiple excitations in different FEKO runs.

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.

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 stays the same (i.e. same geometry, same material parameters and loads, same frequency, same excitation), 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 (e.g. the direction of incidence of a plane wave). FEKO checks all this using checksums 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 (and then also only 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 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. It is advisable to place it right after the EG card.

14.55 PW card

When defining the excitation of an antenna, the source is normally specified as a complex voltage. The PW card allows the user to specify the radiated power or the source power instead (FEKO then just internally scales the whole solution to arrive at this desired power). In addition, it is possible to consider a mismatch between the antenna input impedance and the internal impedance of a voltage source or the characteristic impedance of a transmission line feed.

Parameters:

No scaling (use specified voltages): PW card is not activated, i.e. the specified value of the voltage source is used.

Total source power (no internal impedance): PW card is activated and all the currents are multiplied 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* field. Mismatch is not considered.

Total source power (internal impedance): All voltage sources are assumed to have an input impedance Z_i as specified by the parameters *Source impedance, real part* and *Source impedance, imag part*. The currents are scaled such that the total power supplied by the voltage sources equals P_0 as discussed below. The mismatch losses in the source impedance reduce the antenna gain.

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 , some 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 (i.e. elementary dipoles A5/A6, impressed current elements AI/AV, etc.) 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. When this item is checked, however, this mutual coupling is not considered. This is acceptable for sources which are relatively far away or when no accurate power values are required. (Since gain/directivity are based on power, they are then also possibly not very accurate.)

Source power: The total power P_0 in Watt, i.e. the total power supplied by all the voltage sources, or in the case of transmission lines, the total power of all forward travelling waves.

Details of the various possibilities with the use of the PW card are shown in Figure 14-29.

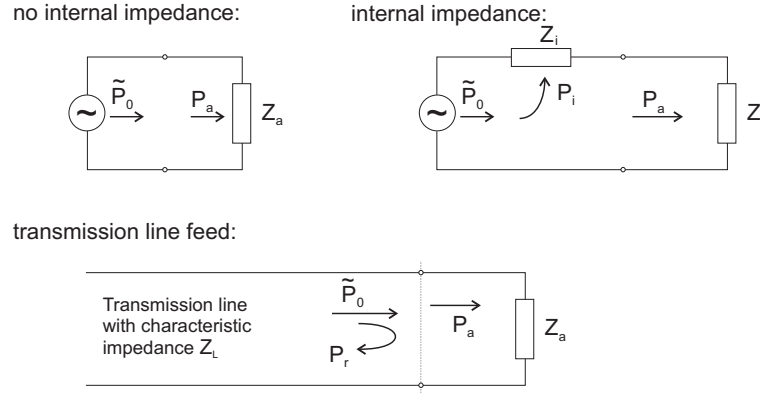


Figure 14-29: Possible applications of the PW card to determine the total power

The options *Total source power (internal impedance)* and *Total source power (transmission line feed)* are only allowed for voltage sources (the A1, A2, A3, A4, A7, AE and AN cards). For models containing other sources such as dipoles and impressed currents, the option *Total source power (no internal impedance)* should be used. For plane waves *No scaling (use specified voltages)* must be used.

The power equations for different cases are discussed below. Consider, in general, that there are N sources (such as in an array antenna) with open circuit voltages $U_{0,v}$ (before the scaling operation) where the parameter v lies in the range $1 \dots N$. At each source there is an antenna input impedance $Z_{a,v}$ (as calculated during the FEKO solution) to which power $P_{a,v}$ is transferred.

- *Total source power (no internal impedance)*

Using this option all the source power is delivered to the respective antennas, i.e.

$$\tilde{P}_{0,v} = P_{a,v} \quad (14-69)$$

as shown in Figure 14-29. To ensure that the total power is P_0 , the power must be scaled with the factor

$$s = \frac{P_0}{\sum_{v=1}^N \tilde{P}_{0,v}} = \frac{P_0}{\sum_{v=1}^N P_{a,v}} \quad (14-70)$$

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 source is considered as shown in Figure 14-29. Since the same current flows through 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

$$P_{i,v} = P_{a,v} \frac{\operatorname{Re} Z_i}{\operatorname{Re} Z_{a,v}} \quad (14-71)$$

and the scaling factor s (to scale the total power supplied by the sources to P_0) is

$$s = \frac{P_0}{\sum_{v=1}^N \tilde{P}_{0,v}} = \frac{P_0}{\sum_{v=1}^N (P_{a,v} + P_{i,v})} = \frac{P_0}{\sum_{v=1}^N P_{a,v} \left(1 + \frac{\operatorname{Re} Z_i}{\operatorname{Re} Z_{a,v}}\right)} \quad (14-72)$$

The combined loss caused by the mismatched antennas

$$P_{\text{loss}} = s \sum_{v=1}^N P_{i,v} = s \sum_{v=1}^N P_{a,v} \frac{\operatorname{Re} Z_i}{\operatorname{Re} Z_{a,v}} \quad (14-73)$$

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 $Z_{a,v}$) is considered to be excited by a transmission line with a complex characteristic impedance Z_L as shown in Figure 14-29. For most practical applications the transmission line will be lossless, resulting in a real characteristic wave impedance. For this lossless case the reflection factor

$$\varrho_v = \frac{Z_{a,v} - Z_L}{Z_{a,v} + Z_L} \quad (14-74)$$

is taken into account when calculating the incident power at the feed point.

The total incident power is given by

$$\tilde{P}_{0,v} = \frac{P_{a,v}}{1 - |\varrho_v|^2} \quad (14-75)$$

and the reflected power by

$$P_{r,v} = |\varrho_v|^2 \tilde{P}_{0,v} = P_{a,v} \frac{|\varrho_v|^2}{1 - |\varrho_v|^2} \quad (14-76)$$

To ensure that the total incident power is P_0 , the power is scaled with the factor

$$s = \frac{P_0}{\sum_{v=1}^N \tilde{P}_{0,v}} = \frac{P_0}{\sum_{v=1}^N \frac{P_{a,v}}{1 - |\varrho_v|^2}} \quad (14-77)$$

and the currents with the factor $\sqrt{s}$. As before the total reflected power

$$P_{\text{loss}} = s \sum_{v=1}^N P_{r,v} = s \sum_{v=1}^N P_{a,v} \frac{|\varrho_v|^2}{1 - |\varrho_v|^2} \quad (14-78)$$

reduces the gain of the antenna.

If we have a lossy transmission line, then forward and backward travelling waves can still be identified on such a transmission line, but a proper definition of power associated with such a wave is not possible, since due to the losses the power will constantly change along the length of the transmission line. It is also questionable whether using the PW card with a lossy transmission line makes any sense. But it has still been implemented, and then 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.

14.56 RA card

This card is used to define an ideal receiving antenna which is described by an impressed radiation pattern and which can be positioned anywhere in the model.

Parameters:

Read pattern data from: In this group the user must select one of four options:

- *a *.ffe file:* Read the radiation pattern from an *.ffe format file (which may be created with the DA and FF cards).
- *an external data file:* Read the radiation pattern from an ASCII file (the format of this file is described at the AR card (see section 14.19)).
- *after this line in the *.pre file:* The radiation pattern is specified in the *.pre file following the RA card (the format is described at the AR card (see section 14.19)). With this option one can make use of the FOR loops to generate patterns from known functions.
- *use last pattern defined at previous RA card:* When using multiple RA cards (i.e. 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, then one can load the pattern just once and at subsequent RA cards check this option here. Then the last defined pattern will be used and memory can be saved (no need to store it again). Note that one is still able to set the antenna position and orientation individually.

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.

Position: In this group the x , y and z coordinates of the receiving point (i.e. the position where the antenna is placed) are entered in m. 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. We will refer to these fields as α_x , α_y and α_z in the rest of this discussion.

File name: The name of the *.ffe or ASCII input file.

Start from point number: This parameter is only relevant when the data is read from 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. This is the same as the line number if all blank lines, comment lines and header lines are stripped from the file.

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 here at the RA card are identical to those of an impressed transmitting antenna (AR card (see section 14.19)). See also the example `antenna_coupling` in the *FEKO Examples Guide*.

The ideal receiving antenna is considered to be decoupled from the model (i.e. no change in the currents and charges 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). Checks are available inside of FEKO to ensure these assumptions.

FEKO computes the power received by this ideal antenna assuming a perfect match, i.e. conjugate complex 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 conjugate complex 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).

14.57 SA card

This card is used to control calculations of the specific absorption rate (SAR) in a dielectric medium.

SA - Calculate the average specific absorption rate (SAR)

Select calculation

☐ Calculate volume-average SAR

☒ Calculate spatial-peak SAR of a 1 g cube

☐ Calculate spatial-peak SAR of a 10 g cube

Specify the search region

☐ Entire region

☒ By medium

☐ By label (FEM only)

☐ At specified position

☐ By layer (Green's function)

☒ Peak SAR in all media

☐ Peak SAR in a single medium

☐ Peak SAR in a medium range

Include medium

up to medium

Parameters:

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, either by medium number or by label number or by 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 here 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' or 'by medium' or 'by layer' basis. The whole body average SAR is also calculated. Selecting the volume by label is only valid for the 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/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/label* and *up to medium/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 must 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 in FEKO, 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.

14.58 SH card

This card can be used to define solid or braided cable shields.

Solid (Schelkunoff)

Parameters:

PEC: Select this option to set the shield of the filaments to PEC.

Material label for metallic material: The label of the metallic (as defined in the DI card) to be used for the shield.

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: Define the thickness of the shield material.

Braided (Kley)

SH - Cable shield definition

Shield label

Shield type

☐ Solid (Schelkunoff)

☒ Braided (Kley)

☐ Define shield properties

Diagram illustrating the braided shield structure. Labels include: Filament, Aperture, ψ , Shield axis, nd , d , $m=8, n=5$, Carrier, and $2\pi(\text{Outer radius})$.

m: number of carriers
n: number of filaments in each carrier

Weave

Number of carriers m

Weave angle (degrees) ψ

Inside braid-fixing material label

Filaments

Number of filaments n

Diameter d

Metallic material

☐ PEC

Material label

Coating

Material label

Thickness of layer

Parameters:

Number of carriers (m): The number of the carriers in the braided shield.

Weave angle (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: The diameter of the filament.

PEC: Select this option to set the shield of the filaments to PEC.

Material label for metallic material: The label of the material (as defined in the DI card) to be used for the shield.

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

SH - Cable shield definition

Shield label

Shield type

☐ Solid (Schelkunoff)

☐ Braided (Kley)

☒ Define shield properties

☒ Define in *.pre file

☐ Load from file

Number of frequency points

	Frequency	Zt Magnitude	Zt Phase
1			
2			

☐ Metallic material

☐ PEC

Material label

Coating

Material label

Thickness of layer

Shield thickness

Parameters:

*Define in the *.pre file:* The frequency dependent shield parameters based on measured transfer impedance and admittance data can be specified in the *.pre file.

Load from file: The file format used when importing the shield properties from file will be described in the section *Load shield properties from file* below.

Frequency: The frequency at which the transfer impedance and admittance are specified in the *.pre file.

Magnitude of transfer impedance: The magnitude of the transfer impedance defined in the *.pre file.

Phase of transfer impedance: The phase of the transfer impedance defined in the *.pre file.

Magnitude of transfer admittance: The magnitude of the transfer admittance defined in the *.pre file.

Phase of transfer admittance: The phase of the transfer admittance in the *.pre file.

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.

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 shield properties from file Below is given an XML example containing fictitious measured data to show the file format for when importing measured cable 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>
```

14.59 SK card

This card is used to consider the Skin effect or ohmic losses or an arbitrary user defined impedance boundary condition on wire segments and surface elements. In addition, it can switch from metallic triangles to thin dielectric layers which may consist of multiple layers and which may be anisotropic. If no SK card has been defined, FEKO assumes ideal conductivity without any losses.

Skin effect and ohmic losses

Parameters:

Affect all structures with label: Affect all segments and triangles with this label.

Assume ideal conductivity: Ideal conductivity is assumed (also the default when there is no SK card for a given label). All other parameters are ignored.

High frequency approximation skin effect: High frequency approximation.

Static approximation skin effect: Static (ohmic losses) approximation.

Exact expression for the skin effect: Use the exact expression of the skin effect for wires and/or surfaces that is valid at arbitrary frequencies.

Triangles as thin isotropic dielectric sheet: Treat metallic triangles as thin isotropic dielectric layers (possibly consisting of multiple layers). Specify the label of the layered dielectric (as defined in the DL card) to be used as the medium for the thin isotropic dielectric layers.

Triangles as thin anisotropic dielectric sheet: Treat metallic triangles as thin anisotropic dielectric layers (possibly consisting of multiple layers). Specify the label of the layered dielectric (as defined in the DL card) to be used as the medium for the thin anisotropic dielectric layers.

User defined surface impedance: An arbitrary user defined complex surface impedance (as defined in the DI card) can be used by specifying the label of the surface impedance.

Thickness of elements: The thickness d of the surface elements in m (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).

Layered dielectric label: Label of the layered dielectric medium which will be used (as specified in the DL card).

Having both triangles and segments with the label *Affect all structures with label* should be avoided. Separate labels and a distinct 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 introduced for each radius.

It should be noted that the skin depth is given by $\delta_{skin} = \sqrt{\frac{2}{\omega\mu\sigma}}$, where the radial frequency $\omega = 2\pi f$ and the permeability $\mu = \mu_r\mu_0$.

- *Assume ideal conductivity:*

No further parameters.

- *High frequency approximation skin effect:*

The required parameters are μ_r , $\tan\delta_\mu$ (defined in the DI card) and σ , and for surfaces also the thickness d . A good conductivity is required, i.e. $\sigma \gg \omega\epsilon_0$.

- For wires a further condition requires that $\delta_{skin} < \varrho$ 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 condition $\delta_{skin} < \frac{d}{2}$ must be met. The surface impedance is given by $Z_s = \frac{1}{2} \sqrt{\frac{j\omega\mu}{\sigma}}$.

- *Static approximation skin effect:*

The required parameters are μ_r , $\tan\delta_\mu$ (defined in the DI card) and σ , and for surfaces also the thickness d . A good conductivity is required, i.e. $\sigma \gg \omega\epsilon_0$.

- For wires a further condition requires that $\delta_{skin} > \varrho$ where ϱ is the wire radius. The surface impedance is given by $Z'_s = \frac{1}{\pi\varrho^2} \frac{1}{\sigma}$.
- For metallic surfaces the condition $\delta_{skin} > \frac{d}{2}$ must be met. The surface impedance is given by $Z_s = \frac{1}{\sigma d}$.

- *Exact expression for the skin effect:*

The required parameters are μ_r , $\tan\delta_\mu$ (defined in the DI card) and σ , and for surfaces also the thickness d . A good conductivity is required, i.e. $\sigma \gg \omega\epsilon_0$.

- For wires with radius ϱ the surface impedance is given by

$$Z'_s = \frac{1-j}{2\pi\varrho\sigma\delta_{skin}} \frac{J_0\left[(1-j)\frac{\varrho}{\delta_{skin}}\right]}{J_1\left[(1-j)\frac{\varrho}{\delta_{skin}}\right]} \quad (14-79)$$

where J_0 and J_1 are Bessel functions.

- For metallic surfaces the surface impedance is given by

$$Z_s = \frac{1-j}{2\sigma\delta_{skin}} \frac{1}{\tan\left[(1-j)\frac{d}{2\delta_{skin}}\right]} \quad (14-80)$$

Examples are given in example_02 and example_33 in the *Script Examples*.

Triangles as a thin isotropic dielectric sheet

Parameters:

Layered dielectric label: Label of the layered dielectric medium which will be used (as specified in the DL card).

This option only makes sense for triangular surfaces, not for wires. The required parameters are d , μ_r , $\tan\delta_\mu$ and ε_r as well as σ or $\tan\delta$, so that $\mu = \mu_r\mu_0$ and the complex dielectric constant $\varepsilon = \varepsilon_r\varepsilon_0(1 - j\tan\delta) - j\frac{\sigma}{\omega}$. Normally 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, i.e. d must be small relative to the lateral dimensions. The mesh size is determined by the wavelength in the environment (i.e. in the medium ε_e, μ_e). Thus 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 $\mu = \mu_e$ and $\varepsilon \neq \varepsilon_e$. For a single layer, the card consists of only one line. The surface impedance, as used by FEKO, is then

$$Z_s = \frac{\beta}{2j\omega(\varepsilon - \varepsilon_e) \sin(\beta \frac{d}{2})} \quad (14-81)$$

where $\beta = \omega\sqrt{\varepsilon\mu}$ is the propagation constant. An example is given in example_32: RCS of a thin dielectric sheet in the *ScriptingExamples* provided with the FEKO installation. 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 $\mu = \mu_e$ or $\varepsilon \neq \varepsilon_e$. 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 a thin anisotropic dielectric sheet

This option is very similar to *Triangles as thin isotropic dielectric sheet*, but the layers are anisotropic. The principle direction in each layer is defined by the angle α (*Angle of princi-*

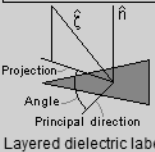
SK - Add a skin effect (finite conductivity)

Affect all structures with label

☐ Assume ideal conductivity
☐ High frequency approximation skin effect
☐ Static approximation skin effect
☐ Exact expression for the skin effect
☐ Triangles as thin isotropic dielectric sheet
☒ Triangles as thin anisotropic dielectric sheet
☐ User defined surface impedance

Reference direction ζ_x
 ζ_y
 ζ_z

The principal directions are determined by first projecting the reference direction into the plane of the triangle (along the normal direction). This vector is then rotated around the normal vector (in a mathematically positive sense) with the specified angle for each layer.



Layered dielectric label

ple direction field) relative to the projection of the vector ζ (*Reference direction* field) onto the plane of triangle (in the DI card). Here α is measured in the mathematically positive sense with respect to the normal vector of the triangle. (POSTFEKO can be used to display the fibre direction and visually check 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 principle direction.

Parameters:

Reference direction: The x , y and z components of the vector ζ (used to define the principle direction, see *Angle of principal direction*).

Layered dielectric label: Label of the layered dielectric medium which will be used (as specified in the DL card).

User defined surface impedance

With this option an arbitrary user defined complex surface impedance $Z_{\square}$ can be used in FEKO. The properties of the complex surface impedance are defined in the DI card and are used in the SK card by specifying the label of the surface impedance sheet.

SK - Add a skin effect (finite conductivity)

Affect all structures with label

☐ Assume ideal conductivity
☐ High frequency approximation skin effect
☐ Static approximation skin effect
☐ Exact expression for the skin effect
☐ Triangles as thin isotropic dielectric sheet
☐ Triangles as thin anisotropic dielectric sheet
☒ User defined surface impedance

Material label

Parameters:

Material label: Label of the material which will be used as the defined surface impedance (as specified in the DI card).

For instance, to model a solid dielectric object with relative permittivity ϵ_r and with conductivity σ at a specific angular frequency $\omega = 2\pi f$, one could use the surface impedance expression

$$Z_{\square} = \sqrt{\frac{\mu_0}{\epsilon_0 \epsilon_r - j \frac{\sigma}{\omega}}} . \quad (14-82)$$

14.60 SP card

This card is used to calculate the S-parameters for the active sources.

Parameters:

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 is backwards compatible to the behaviour of the SP card in FEKO Suite 5.1 and earlier: 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 (i.e. then 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 automatically added loads will be removed again, 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. This was the default behaviour of FEKO Suite 5.1 and earlier. See the discussion below regarding a potential run-time impact of restoring loads (re-computing the MoM matrix is required then).

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 Ω . Note that for waveguide sources (AW card) S-parameters are always related to the corresponding waveguide impedance.

The N ports must be defined before using the SP card. They are identified simply by defining excitation cards. Currently only A1, A2, A3, A4, 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 (i.e. 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 may 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 (Ohm)* 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 here are waveguide

ports (AW card) and modal ports (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 A4 sources it uses L4 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 will be overwritten, as controlled by the *Always add port impedance to existing loads* checkbox.

When execution continues after processing the SP card these loads will either be removed or kept, as controlled by the checkbox *Restore loads after calculation*. This makes a difference when for instance after the SP card the far field is computed by means of an 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 (i.e. far field with ports not loaded by the S-parameter reference impedances). The disadvantage of restoring the loads is that after the SP card processing the loads change and for instance 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 (i.e. the LU decomposed matrix can be re-used).

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 some 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 is not something special in FEKO, but is how S-parameters are defined (i.e. 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 the other way round (e.g. other feed segment 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 S11 (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 consist of a 1-port network and an SP card was requested and the reference impedance remains unchanged. 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 $n \geq 2$) while processing S-parameter request, source, power and impedance data is not written to *.out and *.bof files since this was misleading as the effect of port loads were included.

14.61 TL card

This card is used to connect a non-radiating transmission line between FEKO geometry or other general non-radiating networks or transmission lines. Loads and sources can also be connected on a transmission line terminal using the LN and AN cards respectively.

The screenshot shows the 'TL - Non-radiating transmission line' control card. It features several sections:

- Action:** Three radio buttons: 'Remove all existing transmission lines', 'New transmission line' (selected), and 'Add to existing transmission lines'.
- Network name:** A text input field.
- Input port:** A section with a 'Connected to:' dropdown menu showing 'Wire segment label' and a 'Segment Label' text input field.
- Orientation:** A checkbox labeled 'Reverse connection orientation'.
- Output port:** A section with a 'Connected to:' dropdown menu showing 'Edge connected to ground/UTD'.
- Labels:** A 'Maximum number of labels on a side' spinner set to 1, and a checkbox 'Meshed surface represents positive connection side'.
- Negative side:** A text area for defining the negative side.
- Advanced Options:** Two checkboxes: 'Cross input and output ports' and 'Calculate length from position'.
- Parameters:** A series of text input fields for: 'Transmission line length', 'Losses (dB/m)', 'Real part of Zo (Ohm)', 'Imaginary part of Zo (Ohm)', 'Real part of shunt Y at port 1', 'Imaginary part of shunt Y at port 1', 'Real part of shunt Y at port 2', and 'Imaginary part of shunt Y at port 2'.

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 metre 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.

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 for 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.

Losses: 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, i.e. if a scaling factor which reduces the length of the transmission line is added, the total loss through the line will be less. (The length is now less and the loss per distance remained the same.)

Real part of Z_0 : Real part of the characteristic impedance of the transmission line in Ohm.

Imaginary part of Z_0 : 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 the transmission line 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. To illustrate this, a transmission line port connected wire segment is discussed here, but this applies to connections at edges and nodes also. See Figure 14-31 for an illustration of loads and excitations when doing S-parameter calculations. Figure 14-30 is an illustration of loads and excitation for calculations that do not form part of a S-parameter calculation.

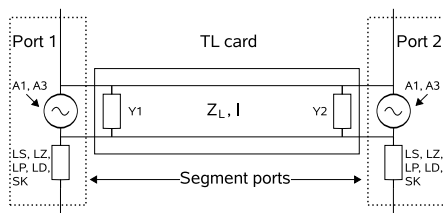


Figure 14-30: Normal load placement.

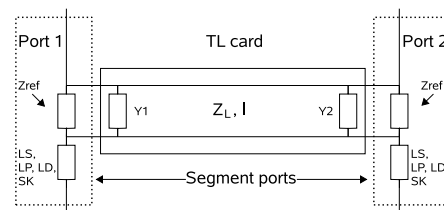


Figure 14-31: Load placement for S-parameter calculations.

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 (i.e. feeding the transmission line directly with an impressed voltage). Any other sources are in series with the port. If S-parameters are computed with respect to an excitation on a wire segment to which a TL card is connected, then the LZ load may not be used at this segment.

The wire segments for the two ports should be located in the same medium, so that the propagation constant of this medium can be taken for the transmission line. If the segments are in different media, then the medium of the segment at the input port is used. Note that the propagation constant and thus also the propagation loss of the transmission line is the same as that of the medium surrounding input 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, one needs to reduce the length of the transmission line.

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.

The TL card is used in `example_39` (*Scripting examples*) to create a log periodic antenna.

14.62 TR card

This card is used to calculate the transmission and reflection coefficients for a plane wave interacting with a planar structure.

TR - Calculate transmission and reflection coefficients for incident plane wave	
Plane position	
X coordinate	
Y coordinate	
Z coordinate	

Parameters:

X, Y, Z coordinate: Cartesian coordinates of the position of the plane wave in m (is scaled by the SF card).

The transmission coefficient is defined as

$$\tau = \frac{E_t}{E_i}. \quad (14-83)$$

The reflection coefficient is given as

$$\rho = \frac{E_r}{E_i}. \quad (14-84)$$

Figure 14-32 shows the incident, transmitted and reflected electric fields on a planar structure.

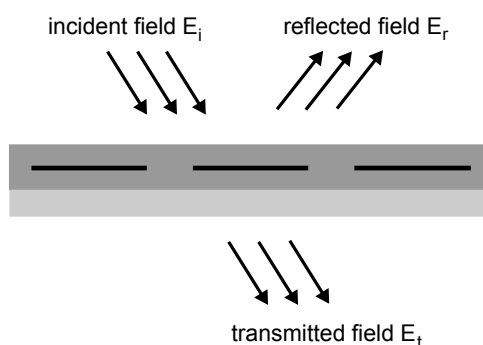


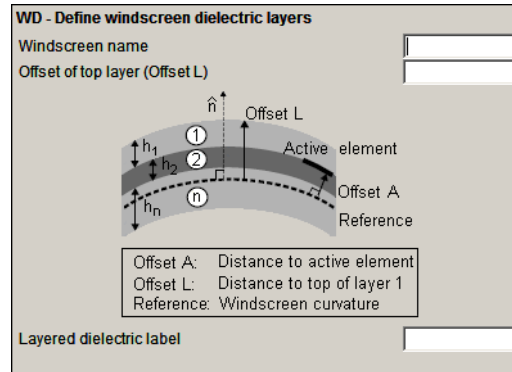
Figure 14-32: A plane wave interacting with a planar structure

Note that for a transmission and reflection request, only a single plane wave source is allowed (i.e. no other sources are allowed in the model). The model must either contain a:

- Planar multilayer substrate without any other geometry/mesh in the model or
- 2D periodic boundary condition (PBC).

14.63 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 w.r.t. the reference plane (in the direction of the reference plane normal).



Note that the the three cards WR (see section 13.52), WA (see section 13.50) and WD (see section 14.63) should be used together to create windscreen antenna models. These three cards respectively define the windscreen reference surface, windscreen active elements (antenna) and the windscreen layered media definition.

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 name: The label of the layered medium to be used, as defined in the DL card.

Part V

Working with SECFEKO

15 The program SECFEKO

15.1 Introduction to SECFEKO

The FEKO licence manager shows all the licences in the specified `secfeko.dat` file (for node locked licences) or connects to the floating licence servers and retrieves licence information. This allows viewing the active components / modules, limits and expiry dates of all licences in a clear human readable form. In addition — for floating licences — it allows viewing the available licences, selecting which licence to use, and managing the licences that are in use. See also the section on floating licences in the *FEKO Installation Guide*, in particular if the floating licence servers must be installed or activated.

The graphical licence manager is only available for MS Windows and Linux. For other platforms (but also for Windows and Linux) a similar functionality is provided by a command line utility `secfeko`. Please just run `secfeko` to see all the various options available, or refer to the Floating Licence section of the *FEKO Installation Guide* where some of the options are discussed.

The licence manager is started by typing `secfeko_gui` from a console window. Under MS Windows it may also be started by selecting *Licence manager* from the *FEKO* menu under the Windows *Start* menu.

15.2 Displaying licence information

Information regarding the licences are indicated by the following colours:

<i>green:</i>	Node locked licences are indicated by green. Note that FEKO will always use the first valid node locked licence. There should generally only be one valid licence on each machine.
<i>cyan:</i>	Floating licences are indicated in cyan.
<i>orange:</i>	Excluded licences are shown in orange, see section 15.3.3 .
<i>red:</i>	Expired licences are indicated in red. If only the <i>Maintenance and Support expiry date</i> is filled in red, it indicates a valid licence but without maintenance and support.

Figure [15-1](#) shows the licence manager with an excluded licence, floating licence, expired maintenance and support, expired licence and node locked licence.

The licence manager will, by default, open the file `%FEKO_HOME%/license/secfeko.dat` which is also used by all FEKO components. The user may, however, also uncheck *Default licence file* and browse for an alternative file, for example to check the information in a new `secfeko.dat` file before replacing an existing licence file. If *Local mode* is checked, floating licences are displayed without connecting to the server.

The main table displays — for each licence — the hostname (if available) where the licence is valid, the licence expiry date, the maintenance and support expiry date, the highest allowed version (older versions can usually be used with newer licence files, but not the other way around),

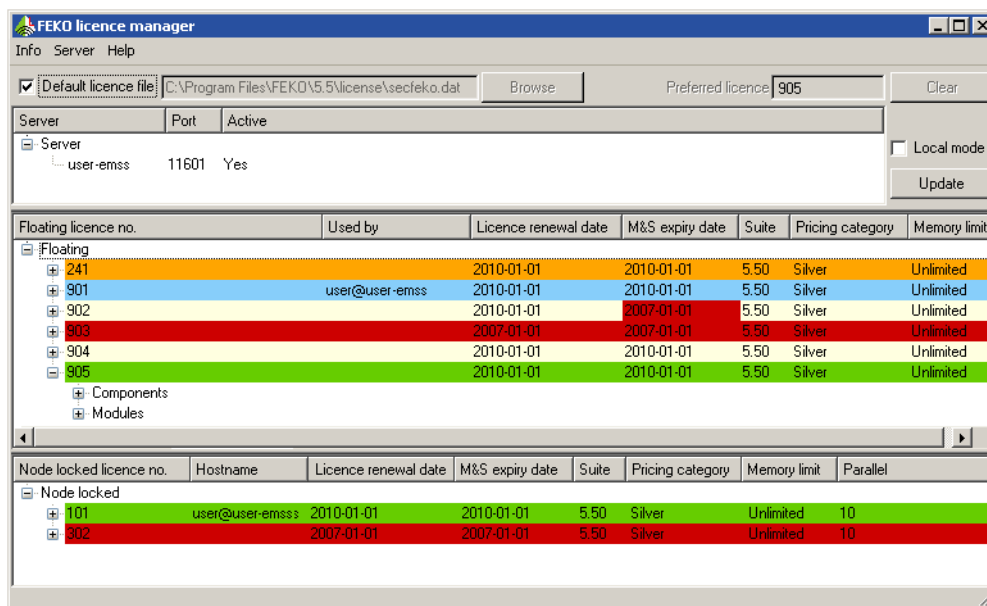


Figure 15-1: The licence manager showing an excluded, floating, expired maintenance and support, expired and node locked licences

the pricing category, the memory limit if there is one, and the number of parallel processes allowed if applicable.

Each licence also contains the following information, shown in Figure 15-2 (click on the icon to the left of the licence to expand / contract it):

- *Machine code*: Expanding this item shows the machine code of the machine where this licence is valid. (Only shown for node locked licences.)
- *Components*: This item lists all the components (for example, CADFEKO, FEKO, RUNFEKO, etc.) which are activated in this licence.
- *Modules*: The modules and extensions that are active.

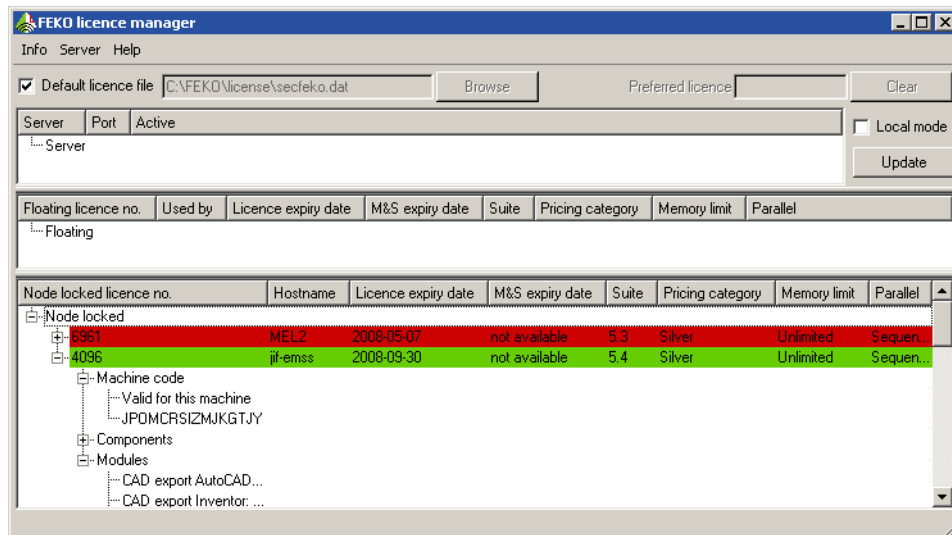


Figure 15-2: The licence manager displaying the *Machine code*.

15.3 Managing floating licences

Figure 15-3 shows the licence manager with a typical floating licence. If the `secfeko.dat` file contains one or more floating licence servers, then this information (server names and port numbers) is displayed at the top of the licence manager. This also includes the status of a floating licence server, whether it is active or not. See the discussion of installing and setting up floating licences in the *FEKO Installation Guide*. A quorum of licence servers must be active for the FEKO floating licence scheme to work (so if you have one server this must be active, but if you are running three redundant floating licence servers, then at least two of them must be active).

The *Update* button allows the user to update the state of the licences. (Note that floating licences are checked in and out as they are being used, thus this information is not static as is the case with node locked licences.)

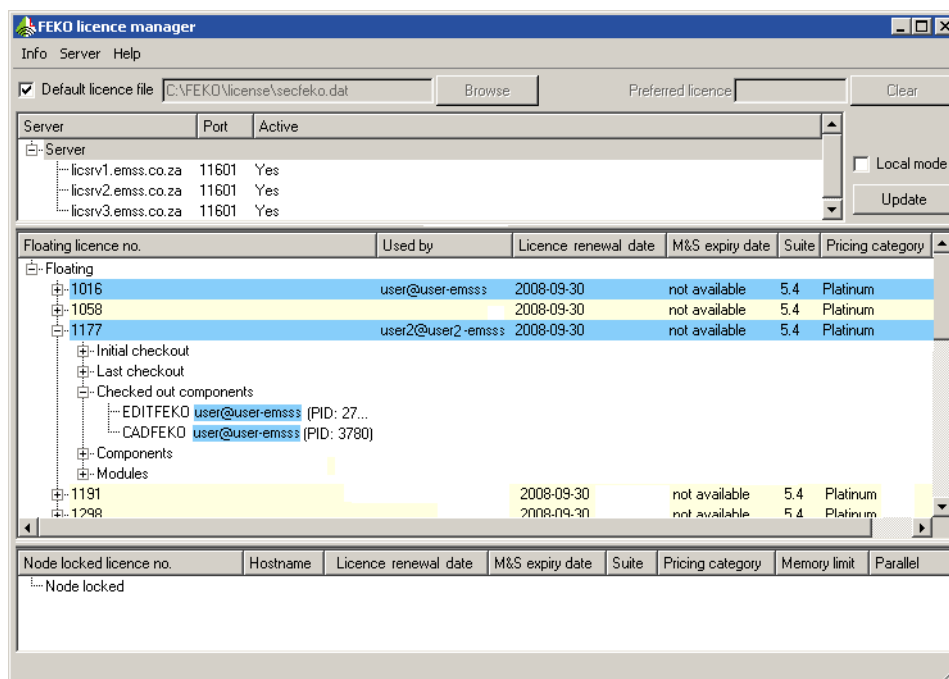


Figure 15-3: The licence manager with a typical licences

With the exception of the machine code of each licence, all the information displayed for node locked licences is also applicable to floating licences. For both floating and node-locked licences, a line with a green background indicates a preferred licence (see section 15.3.1) and a line with a light blue background indicates a licence that is in use.

For each licence that is in use, the table displays the user and hostname where this licence is used, and the *Parallel* column indicates how many solver instances are available. (If a user has a 16 node licence, he may start two concurrent 8 process parallel jobs, but not an 8 process job and a 10 process job.)

In addition, expanding the licence shows information on when the licence was checked out and what components are in use. The user and hostname to which each component has been checked out to, as well as the process ID (*PID*), is shown for each component. This allows each component process to be uniquely identified and located for administrative and tracking purposes. It is possible to manually check in a licence to make it available for a new user (see section 15.3.2).

15.3.1 Concept of a preferred licence

When a new licence is requested, the floating licence server will check out the first available licence. Where users have multiple licences they may not necessarily be identical. (Some licences may, for example, be memory limited GUI licences, or some CAD import module may be activated only in some licences.) In this case the first available licence may not be the desired one.

The licence server supports the setting of a preferred licence for both floating and node-locked licences. Right click on any licence and select *Mark as preferred* from the pop-up menu. The preferred licence is listed at the top right of the licence manager. The next time a new licence is requested, this preferred one will be used if it is still available. (The preferred licence can also be selected by clicking on the licence and selecting *Server* → *Set preferred licence* from the main menu.)

Click the *Clear* button or select *Server* → *Clear preferred licence* from the main menu to remove the preferred licence. When using a floating licence, since the server will just use the first available licence if the preferred licence is already in use, there is very little need for an individual user to clear his preferred licence. However, if your company has one full and a number of limited GUI licences you should not check out the full licence while just setting up your model — then nobody else can use this and you could possibly have used one of the GUI licences.

Finally, note that all licence requests from the same user and the same machine will use the same licence. Thus if you set a preferred licence while you have other FEKO components running (the licence manager does not require a licence), the new licence will only be used once you have closed all components and opened a new one.

15.3.2 Managing floating licence servers

While a licence is checked out, nobody else can use that licence. It may happen that a user has say CADFEKO open on his machine while he is on leave, or that the licence is urgently needed for another run and the user is not available to stop his run. The licence manager therefore allows checking in a licence so that it becomes available again. Right click on the licence and select *Check in licence* from the resulting pop-up menu, or click on the licence and select *Server* → *Check in licence* from the main menu. For security reasons, this check in operation is allowed only for users with administrative privileges (Windows) or the root user (UNIX) or the user who has initially checked out this particular licence.

If a licence has been checked in manually, all running components using that licence will fail the next time they check the licence server. In the case of the solver components (RUNFEKO, PREFEKO, FEKO, ADAPTFEKO, OPTFEKO and TIMEFEKO) the error is fatal and the entire solution will be lost. The GUI components (CADFEKO, POSTFEKO and EDITFEKO) will close after giving the user an option to save the current model. Thus it is usually preferred to only use this manual check-in feature when only GUI components are checked out for a particular licence, and if these GUI components cannot be closed directly.

In addition, the *Server* item on the main menu allows *Shutdown* and *Reset*. If multiple redundant floating licence servers are used, then both *Reset* and *Shutdown* apply to all the running and active servers. For security reasons, both these operations are restricted to users with administrative privileges (Windows) or the root user (UNIX) or the user account under which the licence server is running. Generally it should only be necessary to *Reset* the licence server when modifying the licence file (for example when obtaining a new licence file or when changing the TCP port). *Note that resetting the licence server will check in all licences*, so that all active user jobs will be terminated similar to when checking in individual licences. Users should not need the *Shutdown* option unless the floating licence server executable needs to be updated with a newer one. Note also that once the licence server has been shut down it must be restarted on the server itself — the licence manager cannot connect to the service if it is not running (for multiple redundant floating licence servers each must be restarted individually). Please check the *FEKO Installation Guide*, for further instructions on the floating licence server installation and startup.

15.3.3 Allow or deny usage of certain floating licences

Upon startup, the floating licence server SECFEKO will in addition to reading the licence file `secfeko.dat`, also read a configuration file `secfeko.cfg`. The configuration file defines the

EXCLUDE and the INCLUDE rules for the licences. The same strategy as used for searching for the licence file will be used to locate the configuration file, namely the %FEKO_HOME%\license, but if the environment variable FEKO_LICENSE_FILE is set, the floating licence manager will try to locate the configuration file from the specified directory. If the configuration file is not in the directory as set by the environment variable, the default setting will apply. By default (when the configuration file is unavailable or no rule is specified) a floating licence can be checked out by anybody, from any host.

An INCLUDE rule will result in only specified users or hosts to check out a particular licence. An EXCLUDE rule can be set that will enable all users or hosts except those specified by the rule to be able to check out the licence. Note that an INCLUDE and EXCLUDE rule is not allowed for the same licence, although multiple INCLUDE or multiple EXCLUDE rules are allowed for the same licence.

Syntax of configuration files

Comment lines are indicated by empty lines or lines starting with a # symbol. The INCLUDE statement is shown in the following example.

```
INCLUDE <licence_number> USER <username>
INCLUDE <licence_number> HOST <hostname or IP address>
```

The use of wildcards are also allowed in the hostname or IP address.

```
INCLUDE <licence_number> HOST node-*
```

The same rules apply when creating EXCLUDE rules.

```
EXCLUDE <licence_number> USER <username>
EXCLUDE <licence_number> HOST <hostname or IP address>
```

15.4 Determining the machine code and creating licence requests

The machine codes of the current machine can be displayed by selecting *Info* → *Machine code info* from the licence manager main menu. All licences that match any of these machine codes will be valid on this machine.

By selecting *Info* → *Create request file*, a file that contains all of the relevant information for the creation of a new licence for the machine will be generated. This *request* file is created in the license subdirectory of the FEKO installation and can be used to request a new licence file for the machine as described in the FEKO installation guide.

15.5 SECFEKO command line

The program SECFEKO can also be controlled from a command line. The command line options available are discussed in the following table.

-m	Print the FEKO machine codes for this machine.
-p	Print licence status summary in a table. Included in this summary report will be a line for each floating licence indicating their respective status.

- r Reset the floating licence servers (note that this will reset all FEKO licences and then reread the licence file and start the servers again).
- s Shutdown all floating licence servers (note that this will reset all FEKO licences, and the licence servers must be started manually again).
- slocal Shutdown the floating licence server running on this machine (note that the licence server must be started manually again).
- e xxx Set licence no. xxx to be the preferred licence to be checked out by you from this host (one can reset a previously selected licence by using the licence number 0, i.e. by using the option -e 0).
- c xxx Manually check in licence no. xxx into licence server (normally not required to be done only possibly after a power failure or similar).
- f xxx Instead of using the default FEKO licence file `secfeko.dat` in the default directory, we use an alternative licence file xxx.
- q Quiet mode, don't ask user for confirmation.
- version Print the version information and then exit.
- delay Special internal delay during version output.
- run-from-gui Special execution mode for the graphical user interface.
- local-mode For floating licences no connection will be made to the floating licence server, only the local licence file will be parsed (option only applicable with -p).

Part VI

Working with QUEUEFEKO

16 Introduction to QUEUEFEKO

QUEUEFEKO (and the GUI component QUEUEFEKO_GUI) facilitates the creation of a package which can be transported to remote cluster machines. These packages can be placed in an execution queue (such as PBS) and executed when time and other resources are available. All the 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 has been completed. Results can be viewed by copying the correct package back from the cluster and extracting the contents.

Packages can optionally be encrypted to allow secure transfer of models and results between the user's local machine and the remote cluster. Various options are available for the encryption of packages (see section 16.2.5).

16.1 QUEUEFEKO system overview

QUEUEFEKO_GUI is a graphical interface that allows users to create and extract packages. QUEUEFEKO is a console application that adds a package to an execution queue and takes care of all the management tasks required for the successful execution of the simulation. QUEUEFEKO_GUI is used on the user's local machine for package configuration, creation and extraction, while QUEUEFEKO runs on the remote cluster and oversees the simulation described in the package.

Figure 16-1 is a flowchart describing the general process of using QUEUEFEKO_GUI and QUEUEFEKO.

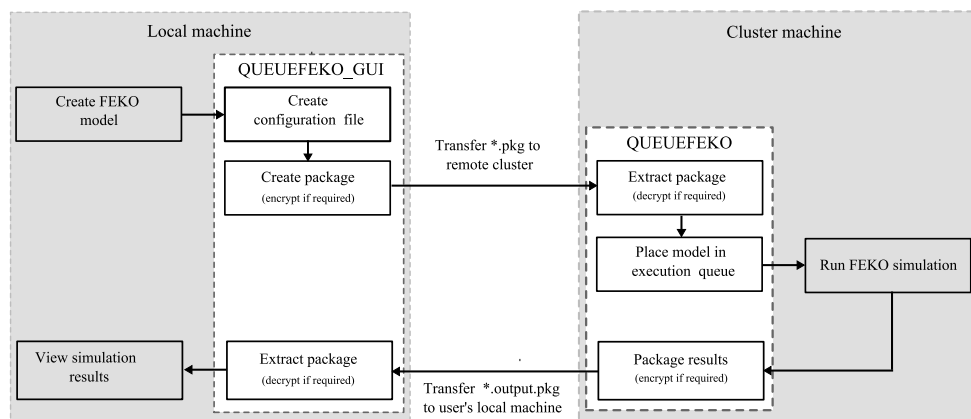


Figure 16-1: Remote execution scenario highlighting QUEUEFEKO's role.

16.2 Creating and extracting packages

The following steps should be followed in order to generate a new package for remote execution. Each step in the process is discussed in the sections that follow.

- Create a new configuration file (or edit an existing configuration file) (see section 16.2.1).
- Set the configuration options (see sections 16.2.2 to 16.2.5).

- Generate the package (see section 16.2.6).
- Add the package to the execution queue (see section 16.2.7).
- Extract the package containing the simulation results (see section 16.2.8).

16.2.1 Package configuration files

The QUEUEFEKO_GUI allows a user to easily create packages with similar settings by storing these settings in a package configuration file. The package configuration is not the package itself, but it contains all the information required to create a package. The QUEUEFEKO_GUI allows the user to access 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.

A new package configuration file can either be created by selecting *File→New configuration* or by opening an existing configuration file by selecting *File→Open configuration*.

When a new configuration is created, a dialog (Figure 16-2) is displayed containing the most often used settings from all three categories (files, FEKO options, cluster settings).

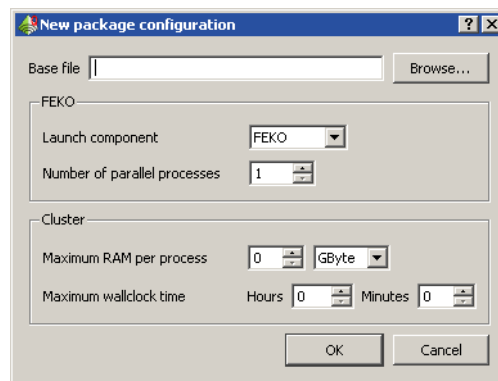


Figure 16-2: New package configuration dialog.

These settings can be adjusted after a new package configuration file is created by selecting the *FEKO* or *Cluster* tabs on the main window.

16.2.2 Including package files

The files required to perform the simulation are determined by reading the *.pre file. Files that are not required for the simulation (but still form part of the FEKO project) are not included in the package by default, but can be added manually. Unnecessary files are not included to minimise upload times to the remote cluster.

Package file selection is controlled by the items in the *Package→File selection* menu. The available options are:



Select a different *.pre file to be the new base file for this package.



Add additional files to the package.



Remove a file from the package.

Figure 16-3 shows a package where two of the files (`startup.pre` and `startup.cfm`) have been included since they are required for the simulation, but the file `network.s2p` has been added manually.

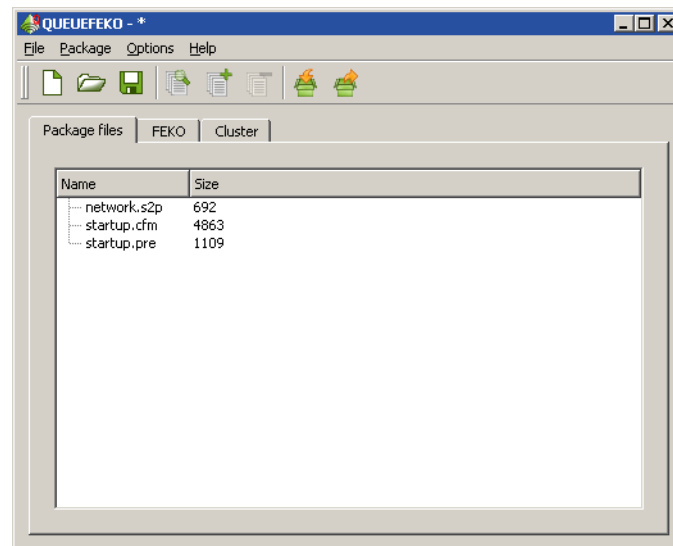


Figure 16-3: QUEUEFEKO_GUI showing the list of files that will be included in the package.

Note that all package files must be located either in the same directory as the base `*.pre` file, or in a subdirectory inside the base directory. This is required since the files are packaged with the directory structure intact. The directory containing the base `*.pre` file is considered the root directory of the package.

16.2.3 FEKO options

Start by selecting the FEKO component (Figure 16-4) that should be launched. Each of the components that form part of the simulation have options that can be set. The *Advanced* edit box on each of the component tabs allow additional command line parameters to be specified that are not available in the graphical interface.

The most important FEKO kernel option is to specify the number of processors that should be used.

16.2.4 CLUSTER options

The remote cluster options allow a user to define the specific job queue that should be used and the manner in which the user should be notified when the job starts and completes. Additionally, resources such as RAM and run time required for the simulation must also be specified. The wallclock time and RAM fields are required; all other fields are optional. Figure 16-5 shows the options available for the cluster.

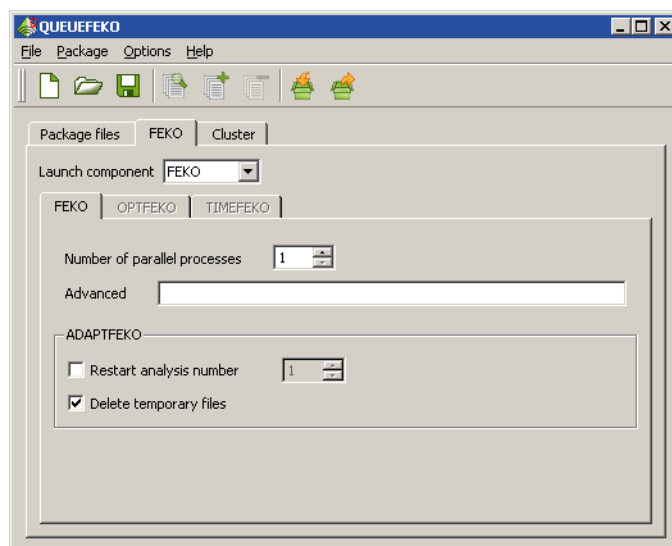


Figure 16-4: The FEKO component dialog.

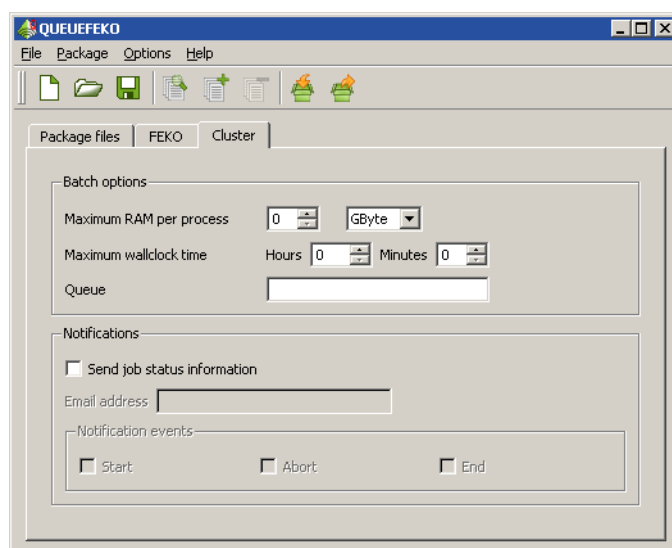


Figure 16-5: Options available for the remote cluster in QUEUEFEKO.

16.2.5 Using encryption (optional)

The package can be encrypted, allowing for secure transfer of models and results between the user's local machine and the remote cluster. Package encryption is only possible if the user has an encryption engine installed locally that is compatible with the encryption engine installed on the remote cluster, and also has the public key of the remote cluster.

The *Security* dialog shown in Figure 16-6 allows encryption to be enabled or disabled for the package, and allows selecting users who will be able to decrypt the input and output packages. It is important that the remote cluster be chosen as one of the users that will be able to decrypt the input package, otherwise QUEUEFEKO will not be able to extract and simulate the model. Also, remember to add yourself to the list of users that will be able to decrypt the result package!

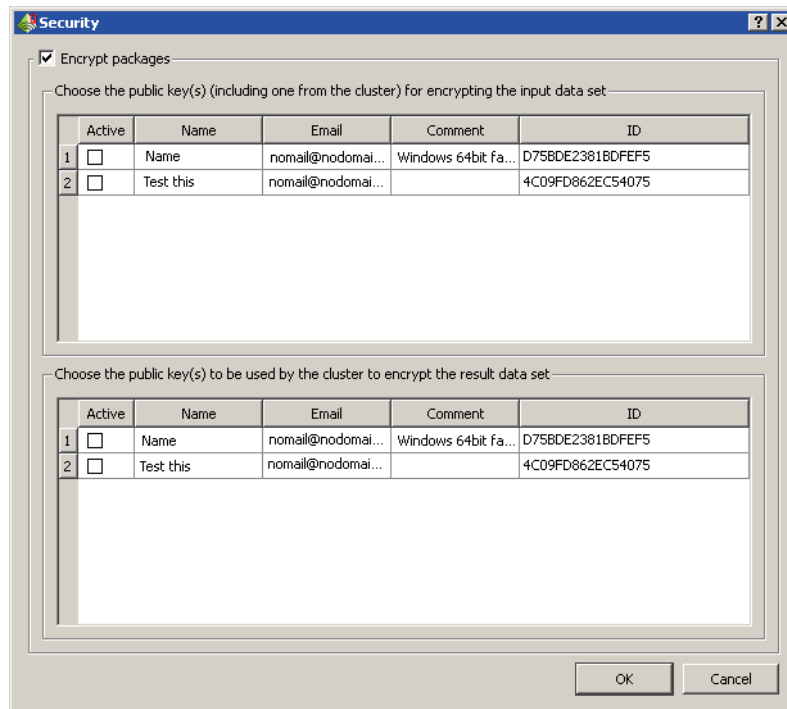


Figure 16-6: Security options available in QUEUEFEKO.

16.2.6 Generate package



A new package can be generated by selecting *Package*→*Generate package*.

The user will be prompted to enter the destination path of the newly created package. Packages are created with a .pkg extension.

16.2.7 Adding the package to the execution queue

The package is transferred to the remote cluster where it is placed in an execution queue. Users using the CrunchYard website can simply upload the package to the website and mark the package for execution. For other clusters the file may be copied and added to the PBS queue manually.

On the cluster machine, the package is added to the queue using the following command:

```
queuefeko mypackage.pkg
```

Simulation of the package will commence as specified by the user in the package configuration file.

16.2.8 Extract package

After the simulation has completed, a new output package (*.output.pkg) will be available for downloading from the remote cluster machine to the local machine. The package can be extracted using the QUEUEFEKO_GUI.



A package is extracted by selecting *Package*→*Extract package*, selecting the package to extract and selecting the location on the disk where the contents of the package should be extracted to. Once the package has been extracted, the entire directory contents as it was on the remote machine is made available on the local machine.

16.2.9 Decryption of package (if required)

The package will be decrypted using the first available local Private key. If it was protected by a passphrase, the user will be prompted to enter it. The typed characters are not saved, but sent directly to the encryption engine for verification.

16.2.10 View results

Results obtained from the remote cluster machine can now be viewed in POSTFEKO on the user's local machine as if the simulation was run locally.

16.3 Setting preferences

The *Preferences* dialog allows the user to specify the PDF viewer to be used when opening the *FEKO user manual*, as well as the encryption engine that should be used. Currently only GNU Privacy Guard is supported.

Part VII

Preprocessing and the FEKO solver

17 The preprocessor PREFEKO

17.1 Description

The surface of the structure to be analysed with the program FEKO, has to be subdivided into elementary surfaces. Wires have to be subdivided into segments. The mesh size is dependent on the wavelength in the medium surrounding the structure. The program PREFEKO can do all the meshing. It automatically generates the geometric data, in the form required by FEKO, from the data given by the user. The mesh density is controlled by a couple of parameters. PREFEKO also imports mesh geometry, for example constructed in CADFEKO, integrating it with the final FEKO input file.

This section describes the principle workings of the program. 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).

17.2 Running PREFEKO

If, for example, a file `example.pre` has been created using a text editor, PREFEKO is started using the following command:

```
prefeko example
```

After successful execution a file `example.fek`, is created. This is the input file for FEKO. The program 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 listed below:

- `--version` Print the version information and then exit.
- `--fek-format x` Write the `*.fek` file in the x^{th} file format.
- `-#var=value` Set a variable `#var` to the value `value`.
- `-d` Debug mode with some extra output (can be used to troubleshoot errors)
- `--ignore-errors` Error messages are non-fatal, i.e. 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 always the first one only.
- `--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 (see the next section) from the command line, for example calling PREFEKO with

```
prefeko filename -#variable1=value1 -#variable2=value2 ...
```

It is a good idea to use the `!!print_to_out` command to write these variables to the output file in order to keep a record of their values.

Note - prior to FEKO Suite 5.2, the syntax

```
prefeko filename #variable1=value1 #variable2=value2 ...
```

(i.e. without the minus sign) was supported, but this has been superseded by the new format to avoid shell escaping problems under UNIX where the `#` must be masked as `\#`. While the old syntax is still supported for backwards compatibility, users are advised to use the new syntax with the leading minus sign.

18 The FEKO solution kernel

18.1 Introduction

The program FEKO does the actual calculations involved in the solution. Input and output of the FEKO solver is based on files. During a solution, the status of the calculation phases is indicated, on screen.

18.2 Running the FEKO kernel

18.2.1 Running the sequential version

Normally users are advised 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 *Run* menu, by selecting *FEKO* (the shortcut key <Alt><4> can also be used).

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.pre` must be an existing input file. RUNFEKO executes PREFEKO if the `*.fek` file is missing or older than the `*.pre` file and then executes the appropriate FEKO solver.

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 section 18.2.2. Additional options for the remote launching of FEKO are found in section 18.2.3. In the GUI these settings are available through the *Run* menu → *Component parameters* → FEKO dialog (see also Section 6.16.1).

- `--execute-prefeko` Always execute PREFEKO — even if the existing `*.fek` file is newer than the `*.pre` file.
- `--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` file are older than the existing `*.fek` file.
- `--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.
- `--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 and 4 = high. If not specified, the default is 2. This option might not be available for specific systems or specific FEKO versions, then it is just ignored.

- use-gpu Execute FEKO using GPU acceleration.
- use-job-scheduler Run the parallel FEKO kernel within a queuing system (see the list of supported systems below) and obtain the number of parallel processes as well as the host list directly from the respective job scheduler.
- remote-use-mpi Activates the MPI method on Windows.
- prefeko-options ... All options following this, up to the next -xxx-options are passed to PREFEKO.
- feko-options ... All options following this, up to the next -xxx-options are passed to FEKO.
- adaptfeko-options ... All options following this, up to the next -xxx-options are passed to ADAPTFEKO.
- d Debug mode with some extra output (can be used to troubleshoot errors).

Various queuing systems are supported by FEKO under Windows and Linux when using Intel MPI (which is the default on these systems). The queuing systems that are currently supported are Windows CCS (compute cluster server) Torque 1.2.0 and higher, Altair PBS Pro 7.1 and higher, OpenPBS 2.3, Platform LSF 6.1 and higher, Parallelnavi NQS V2.0L10 and higher, Netbatch 6.x and higher.

The optional command line parameters for FEKO (specified after -feko-options) are listed below.

- check-only If this option is used, FEKO processes and checks the model, but does not start a solution. This is useful to for instance check an input file on a local PC before submitting it to a parallel computer.
- 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.

18.2.2 Running the parallel version

The parallel version of FEKO may be used on any system that is licenced to run multiple FEKO processes concurrently. If a system has a multiple-core CPU (e.g. a quad-core processor) then a sequential FEKO licence will allow a parallel solution with up to 4 parallel processes to be launched. For multiple-CPU systems (e.g. a system with 2 separate dual-core CPUs) a 2-CPU parallel FEKO licence will be required in order to run parallel solutions involving all 4 of the available cores.

In order to use the parallel version of FEKO from the GUI, one first has to configure the host-names and number of processes that will be used for each node. (This will initially be set up during installation, meaning that reconfiguration is only necessary if changes are made.) To do this configuration, open the *Run* menu → *Component parameters* (see also Section 6.16.1 for additional detailed information on the settings shown on that dialog). On the *FEKO* tab, under *Parallel execution* click the *Configure* button. In the dialog (shown in Figure 18-1), the host-names and number of processes to be started on each host must be entered. Usually one process per core available on each machine should be chosen. For example 2 processes for a dual-core machine. One may also use this to implement a crude load balancing system, running more

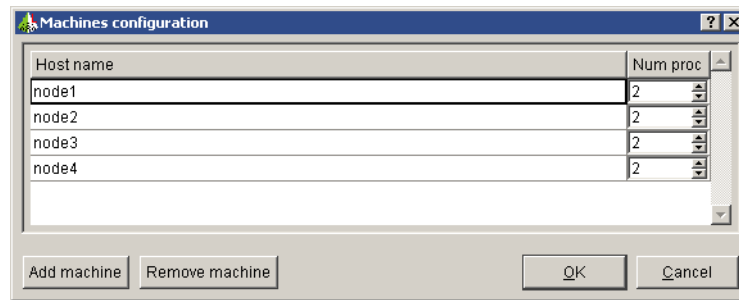


Figure 18-1: Dialog for the parallel host configuration

processes on hosts with faster CPUs or more memory. Nodes may be added or removed from the current cluster setup using the *Add machine* and *Remove machine* buttons respectively.

A special note for parallel Windows PC clusters: FEKO must be installed in 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.⁹

When the user clicks *OK* on this window, the hosts are saved to a file `machines.feko` in the directory specified by the environment variable `FEKO_USER_HOME`. This file is then used in the actual parallel process launching.

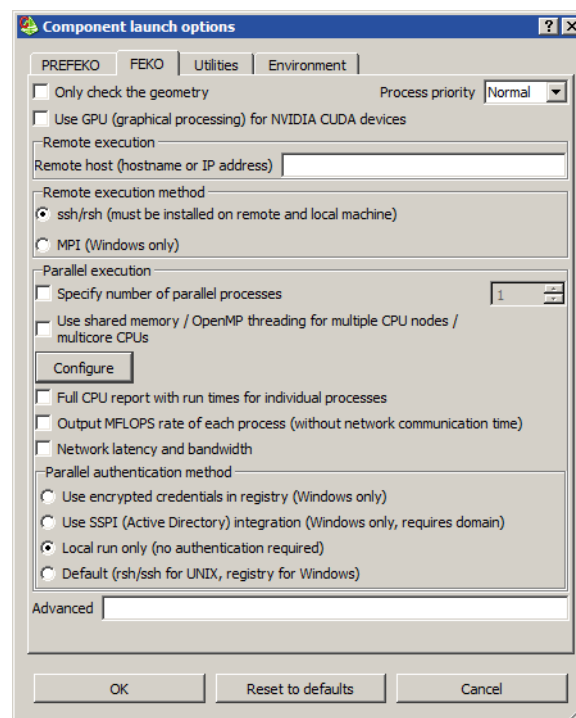


Figure 18-2: Dialog for setting FEKO solution options

On the *Run* → *Component parameters* → FEKO tab (shown in Figure 18-2), further settings with regard to the FEKO solution can be made. If the *Output MFLOPS rate ...* and *Network*

⁹It is possible to launch the job without including the local machine. The `*.fek` input file must then be located on the first PC in the list and the `*.out` and `*.bof` output files are created on this PC — both in the exact 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, or one can also use remote parallel launching (see section 18.2.3) where FEKO does this copying explicitly.

latency and bandwidth options are selected, FEKO will also print a table giving the performance of the various nodes. It is recommended that this is used during setup to ensure an optimum configuration. These checks are repeated each time FEKO calculates the solution. 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 the user to continue with other interactive work. However, all machines in the cluster operate at the speed of the slowest node, so starting other CPU-intensive jobs on one of the nodes in a cluster is generally not recommended.

After having configured the parallel FEKO version and after having set any possible special options, the parallel FEKO version can be run. To do this, *Run* → *Parallel FEKO execution* should be selected so that a check-mark is shown 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 (e.g. on a UNIX workstation), the parallel FEKO version is started with

```
runfeko example_08 -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 section 18.2.1, 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.
- use-openmp-threading** Enable shared-memory parallel communication between multiple processes residing on the same host.
- machines-file *machname*** The file *machname* is the machines file with the node names and the number of CPUs (see below).
- mpi-options ...** All options following this (if another **-xxx-options** parameter is used, all arguments before the second **-xxx-options** parameter) are passed to the MPI launcher (e.g. *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).
 - locallyonly** 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. ¹⁰
 - registry** Encrypt the credentials (username and password) into the registry. This option is available on Windows only.

¹⁰Note that maybe additional (one time) configuration settings have to be done (by the domain administrator) to prepare the Windows domain for this kind of authentication requests (see the Intel MPI (User Authorization — Active Directory Setup) and/or MPICH2 (Runtime Environment — Security) documentation shipped in the directory *mpi\<mpi-version>\doc* of the FEKO installation directory for details).

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
```

with names `host1` and `host2` (this is the output of the UNIX command `hostname`), and 4 and 8 processors respectively, the machines file will be

```
host1:4  
host2:8
```

With this machines file, the example with 6 processes given above will run with 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.

Such a machines file (the file `mpi/share/machines.feko` under the FEKO installation path `FEKO_HOME`) is automatically created during the installation of the parallel version of FEKO. By default FEKO uses this file. If a different distribution of the processes is required, this file can be manually edited - this is, however, strongly discouraged. The user should rather create a separate machines file with the syntax described above. The environment variable `FEKO_MACHFILE` can be used to force `RUNFEKO` to use this file instead of the default. The required commands (for the `sh` shell) assuming the desired machines file is, for example `machname`, are

```
FEKO_MACHFILE=./machname  
export FEKO_MACHFILE  
runfeko example_08 -np 6
```

Alternatively one may pass the name of the machines file as an argument to `RUNFEKO` on the command line like

```
runfeko example_08 -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` (see section 24.3) contains more information). For very special applications or experienced users it may be necessary to pass additional options to MPI. (In such a case, the appropriate MPI manuals should be considered.) These 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 that `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 the Intel MPI the environment variable `I_MPI_DEVICE` is quite important to control which device should be used (sockets or shared memory, or some RDMA device etc.). Such environment variables should preferably be set in the FEKO initialisation script `initfeko`. Check the comments there, and also

check the supplied MPI documentation (typically in the subdirectory `mpi/<mpi-version>/doc` in the FEKO installation).

If using the `--use-openmp-threading` option and the number of parallel processes are specified as 12, 2 MPI processes with 4 and 8 threads respectively will be used. The machines file will then be as follows:

```
host1:4  
host2:8
```

for the example machines used throughout this section.

18.2.3 Running on a remote host

The FEKO kernel can also be started on a remote host with automatic file transfer. For instance the user can run the FEKO user interface on a Windows PC, but start a sequential or parallel FEKO job directly from this user interface on some other 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, e.g. one can launch a remote job from a Windows PC on an 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 when doing remote launching from a Windows host to a Windows host. It is based on pure Windows commands and relies on a network share for copying files and uses the MPI daemon (as shipped with the FEKO Suite) 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*.

General settings and usage

After the setup, using remote launching is simple. On Windows and Linux, this remote launching facility can be used directly from within the GUI components CADFEKO, EDITFEKO or POSTFEKO. As described above for the parallel launching, open the *Component parameters* dialog (see also Section 6.16.1) from the *Run* menu. This dialog is shown in Figure 18-2. 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 the remote launching after it has been set up, select *Run* → *Remote FEKO execution* so that a check mark is shown next to the *Remote FEKO execution* option. Runs of the FEKO solver (either sequential or parallel if *Parallel FEKO execution* 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 command

```
runfeko example_08 --remote-host h
```

can be used, where the parameter `h` following `--remote-host` gives the hostname or the IP address of the remote host. This will automatically use the SSH based remote launching method. If one wants to use the MPI based method, then the additional option `--remote-use-mpi` must also be given as

```
runfeko example_08 --remote-host h --remote-use-mpi
```

This command line option of RUNFEKO may be combined with other options, for instance using

```
runfeko example_08 --remote-host h -np 4 --machines-file m
```

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 mentioned earlier, the remote launching facility has an automatic file transfer included, so it is not necessary 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 run. After the completion of the remote run, all files will be copied back to the client and this temporary subdirectory on the remote machine will be removed.

A final note on 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 (i.e. it will be copied over). So 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 (when launching from the GUI it will always be there, but from the command line this can be omitted, the parallel hosts are then found using some default mechanism (environment variable `FEKO_MACHFILE`, default location for the file `machines.feko` etc. More information can be found in section [18.2.2](#)). It is important to realise that these default options are used as set on the remote host where the parallel job is launched (this is e.g. the control node of a parallel cluster). Thus, FEKO will read the `machines.feko` file on the remote host, and not on the local host where jobs are launched.

19 Description of the output file of FEKO

The program FEKO writes all the results to an ASCII output file *.out as well as to a binary output file *.bof. The latter is used by POSTFEKO for the result extractions. The *.out file is human readable and contains all of the expected results.

In this section the various parts of the output file *.out are described.

19.1 Geometric data

First the geometric data is given (if it has not been suppressed at the EG card). 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				
	0	0.0000E+00	2.0000E-01	0.0000E+00					
	0	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				
	0	3.3333E-02	0.0000E+00	0.0000E+00					
	0	0.0000E+00	2.0000E-01	0.0000E+00					
		0.0000E+00	0.0000E+00	-1.0000E+00	3.3333E-03				

In the first column the number of the triangle is written. In the second column, the label is given followed by the medium in which the triangle is situated. A 0 means that it is in free space. The next three columns are the x, y and z coordinates 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 toward the triangle. Below the edge numbers the area of the triangle is given in m^2 .

Following this is an extract of the data for the edges between the triangle. Whenever two triangles have two common vertices, such an edge is generated.

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)								
no.	type	length/m	media		triangle no.		points of tr. ...	
					KORP	KORM	POIP	POIM
1	1	2.0276E-01	0	-1	1	2	1	1
2	1	2.0000E-01	0	-1	2	3	3	3
3	1	3.3333E-02	0	-1	2	7	2	2
information on symmetry								
					yz	xz	xy	status
					0	0	0	unknown
					1	0	0	0
					0	0	0	unknown

Each edge is assigned a consecutive number, which appears in the first column. The second column indicates the type of the edge. The length of the edge is given in the third column and the medium in which the edge is found appears in the fourth column. On an edge there are exactly two triangles. In the columns KORP and KORM the numbers of these two triangles are

given and the positive current direction is from the triangle KORP to the triangle KORM. In the column POIP the number of the corner point of the triangle KORP, which is opposite to the edge, is given. The same applies to the column POIM.

The next four columns contain information concerning the symmetry. In the column yz the number of the edge appears, corresponding to the plane $x=0$ (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 planes $y=0$ and $z=0$.

The last column with the heading STATUS has the following meaning: If unknown appears in it, the edge has an unknown status. The applicable coefficient of the current basis function cannot be determined from the symmetry, but has to be determined from the solution of the matrix equation. If 0 is present in the STATUS column, 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 whose coefficient is equal to (positive sign in the column STATUS) or the negative of (negative sign in the column STATUS) of the coefficient of the current basis functions. From symmetry the coefficient of the current triangle does not have to be determined.

The data of the dielectric triangles (surface current method) differ only slightly.

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	7.2441E-02	1	2	3	
	1	9.4044E-01	0.0000E+00	3.8954E-01					
	0	8.6886E-01	3.5989E-01	3.8954E-01					
		8.2033E-01	1.6317E-01	5.4812E-01					
2	0	9.4044E-01	0.0000E+00	3.8954E-01	7.8817E-02	4	5	6	
	1	1.0179E+00	0.0000E+00	0.0000E+00					
	0	9.4044E-01	3.8954E-01	0.0000E+00					
		9.6264E-01	1.9148E-01	1.9148E-01					

Two additional columns, POIA and POIE are provided, but they are not relevant anymore and should be ignored. The symmetry information is shown for the basis functions of both the equivalent electric or magnetic current densities.

For the edges the extract is

DATA OF THE DIELECTRIC EDGES (with MoM)												
no.	type	length/m	media	triangle no.		points of the triangle ...						
				KORP	KORM	POIP	POIM	POIA	POIE			
1	3	3.6694E-01	0	1	1	3	1	3	3	2		
2	3	5.1069E-01	0	1	1	4	2	1	1	3		
3	3	3.9718E-01	0	1	1	45	3	2	1	2		
				electr. info of symmetry				magnet. info of symmetry				
				yz	xz	xy	status	yz	xz	xy	status	
				40	75	141	unknown	40	75	141	unknown	
				41	76	142	unknown	41	76	142	unknown	
				42	-3	143		0	42	-3	143	unknown

In addition to the data that is given for the metallic triangles, the following columns are provided POIA, POIE, KNP and KNM. The column POIA contains the number of the corner point of the triangle in KORP, where the basis function for magnetic surface current begins and the column

POIM contains the number of the end point of the triangle where the basis function ends. The sizes KNP and KNM are the lengths when the vertices are connected to the middle of the opposite edge in the triangles KORP and KORM. The symmetry information is shown for the basis functions of both the equivalent electric or magnetic current densities.

The data for the segments follows 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	
	0	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
	0	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
	0	0.0000E+00	0.0000E+00	4.2857E-01	1.4286E-01	2.0000E-02

Here each segment is assigned a consecutive number. In the second column the label of the segment appears and below it the number of the medium in which it finds itself. A zero (0) means free space (vacuum). Then the coordinates of the start and end points of the segment follow. In the previous row the numbers of the nodes, that are adjacent, appear. A positive sign for the node number indicates that the positive current direction is defined away from the segment. When there is a negative number, then the positive direction is directed toward the segment. In the next row the length of the segment appears, followed by the radius.

For the data of the nodes between the segments a data table is given.

DATA OF THE NODES BETWEEN THE SEGMENTS						
No.	no. of segment		points of segm.		...	
	ISEGP	ISEGM	KNOP	KNOM		
1	1	2	2	1		
2	2	3	2	1		
3	3	4	2	1		
	info of symmetry					
		yz	xz	xy	status	
		0	0	5	unknown	
		0	0	6	unknown	
		0	0	7	unknown	

The consecutive numbers of nodes are given in the first column. Then the number ISEGP and ISEGM of the two connected segments follow. 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). The following four columns contain the data about the symmetry and are the same as for the metallic triangles described above.

If there are any connections between triangles and segments, then the following data is given.

GEOMETRIC DATA OF CONNECTIONS SEGMENTS - TRIANGLES									
no.	Data of triang.		data of segm.		angle	info of symmetry			
	DRENUM	DREPOI	SEGNUM	SEGPOI		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 number of the triangle at the connection point is given in the column DRENUM with the connecting vertex (1 to 3) in the column DREPOI. Likewise the connecting segment's number is given in the column SEGNUM 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 radians). The meaning of the symmetry information in the last four columns is the same as that of the metallic triangles given above.

If dielectric volume elements 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			
0	3.3333E-01	0.0000E+00	0.0000E+00			
Cube	0.0000E+00	3.3333E-01	0.0000E+00			
	0.0000E+00	0.0000E+00	3.3333E-01			
2	0.0000E+00	0.0000E+00	3.3333E-01			
0	3.3333E-01	0.0000E+00	3.3333E-01			
Cube	0.0000E+00	3.3333E-01	3.3333E-01			
	0.0000E+00	0.0000E+00	6.6667E-01			
3	0.0000E+00	0.0000E+00	6.6667E-01			
0	3.3333E-01	0.0000E+00	6.6667E-01			

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 to which from the reference point the first basis function is defined. Further, the third and fourth rows define the next two basis functions with respect to the reference point.

In each dielectric cuboid there are three basis functions, in each coordinate direction one. The data of these basis functions is given in the following format:

DATA OF THE BASIS FUNCTIONS FOR DIELECTRIC CUBOIDS						
No.	cuboidno.	direc.	Symmetry information			
			yz	xz	xy	status
1	1	1	28	55	109	unknown
2	2	1	29	56	110	unknown
3	3	1	30	57	111	unknown
4	4	1	31	58	112	unknown
5	5	1	32	59	113	unknown

In the first column the consecutive number of the basis function is given. The next column indicates the number of the cuboid. The column direction indicates the direction of the basis function in the respective cuboid. 1 indicates that e.g. the basis function is defined from the reference point to the second corner point. The last four columns contain information concerning the symmetry properties of the cuboid, where the structure and the meaning is the same as with the other basis functions.

For the FEM, the data of the tetrahedral volume elements is printed in a table like this:

DATA OF THE TETRAHEDRAL VOLUME ELEMENTS (FEM)										
no.	label	x1 in m	y1 in m	z1 in m	nodes					
	medium	x2 in m	y2 in m	z2 in m	faces					
		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	0.0000E+00	-1.3337E-02	1.3828E-03	1	2	3	4		
	air	0.0000E+00	-1.4000E-02	0.0000E+00	1	2	3	4		
		0.0000E+00	-1.2500E-02	0.0000E+00	1	2	3	4	5	6
		1.0157E-03	-1.2403E-02	1.1734E-03	3.5111E-10					
2	DRA1	0.0000E+00	-1.4000E-02	0.0000E+00	2	3	4	5		
	air	0.0000E+00	-1.2500E-02	0.0000E+00	4	5	6	7		
		1.0157E-03	-1.2403E-02	1.1734E-03	4	5	7	6	8	9
		2.7313E-03	-1.3731E-02	0.0000E+00	8.0125E-10					
3	DRA2	0.0000E+00	-1.2500E-02	0.0000E+00	3	4	5	6		
	air	1.0157E-03	-1.2403E-02	1.1734E-03	7	8	9	10		
		2.7313E-03	-1.3731E-02	0.0000E+00	6	8	10	9	11	12
		2.4386E-03	-1.2260E-02	0.0000E+00	7.1540E-10					

The consecutive numbers of the elements are given in the first column. The label and dielectric medium label of the element are given in column 2. Columns 3, 4 and 5 provide the x-, y- and z-coordinates of the vertices of the element. The numbers of each node, face and edge bounding the tetrahedral element follow in the last columns. Thereafter information follows regarding the number of basis functions.

DATA FOR MEMORY USAGE									
Number of metallic triangles:	0				max. triangles:MAXNDR	=		176	
Number of dielectric triangles:	176								
Number of diel. GO triangles:	0								
Number of FEM surface triangles:	0								
Number of metallic segments:	0				max. segments: MAXNSEG	=		0	
Number dielectr./magnet. cuboids:	0				max. cuboids: MAXNQUA	=		0	
Number of tetrahedral elements:	0				max. tetrah.: 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 reg.:MAXFOGE	=		0	
Number of polygonal surfaces:	0				max. surfaces: MAXPOLYF	=		0	
					max. corner p.:MAXPOLYP	=		0	
Number of UTD cylinders:	0								
Number of metallic edges (MoM):	0	unknown:	0		max. edges: MAXNKA	=		264	
	0	unknown:	0		(magnet.)				
Number of metallic edges (PO):	0	unknown:	0		(electr.)		0 (magnet.)		
Number of dielectric edges (MoM):	264	unknown:	66		(electr.)				
	264		66		(magnet.)				
Number of dielectric edges (PO):	0	unknown:	0		(electr.)		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 basis funct. for MoM:	528	unknown:	132		max. basisf. MAXNZEILE	=		528	
Number of basis funct. for PO:	0	unknown:	0		max. basisf. MAXNKAPO	=		0	

Here the data, e.g. the number of basis functions on the nodes between segments, can be extracted. It is also indicated how many have the status unknown, i.e. how many have to be determined by solving the matrix equation.

19.2 Excitation

The data here is structured depending on the means of excitation. A voltage source on a segment generates the following data block is generated:

EXCITATION BY VOLTAGE SOURCE AT SEGMENT

```

Name:                      VoltageSource1
Number of voltage source:  N =      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 deg.:            ARG(U0) = 0.00
Source at segment w. label: ULA = Line1.Wire1.Port1
Absolute number of segment: UNR =    11
Location of the excit. in m: X = 0.00000E+00
                           Y = 0.00000E+00
                           Z = 0.00000E+00
Positive feed direction:  X = 0.00000E+00
                           Y = 0.00000E+00
                           Z = 1.00000E+00

```

Similar information is provided for other voltage excitations (excitation on an edge, microstrip excitation, voltage source connected to a general network, etc).

If a waveguide excitation is used, the information in the output file has the following format:

EXCITATION BY IMPRESSED WAVEGUIDE MODE

```

Name:                      WaveguideExcitation1
Number of the excitation:  N =      1
Port label:                Union2.Face44
Frequency in Hz:           FREQ = 1.64500E+09
Wavelength in m:          LAMBDA = 1.82245E-01
Port type:                 Rectangular
Port dimensions:           width = 1.29600E-01 m
                           height = 6.48000E-02 m
Port reference location:   X = -6.48000E-02 m
                           Y = -3.24000E-02 m
                           Z = -3.02000E-01 m
Impressed mode:           TE 1 0
Direction of propagation: NX = 0.00000E+00
                           NY = 0.00000E+00
                           NZ = 1.00000E+00
Amplitude in A/m:         1.00000E+00
Phase in degrees:          0.00
Transmitted power in W:    1.13772E+00

```

Information regarding the mode expansion at the waveguide excitation follows the waveguide excitation information as follows:

MODE EXPANSION DATA OF WAVEGUIDE PORT

```

Port label:                Union2.Face44
Port type:                 Rectangular
Port width:                1.29600E-01 m
Port height:               6.48000E-02 m
Frequency:                 1.64500E+09 Hz

```

Mode indices		Cutoff freq.	Transverse wave impedance	...
m	n	in Hz	real part	imag. part
TE	0 1	2.31321E+09	0.00000E+00	3.81056E+02
TE	1 0	1.15661E+09	5.29795E+02	0.00000E+00
TE	1 1	2.58625E+09	0.00000E+00	3.10534E+02
TM	1 1	2.58625E+09	0.00000E+00	-4.57038E+02

Propagation factor		Description
real part	imag. part	
3.40852E+01	0.00000E+00	Evanescent
0.00000E+00	2.45159E+01	Propagating
4.18260E+01	0.00000E+00	Evanescent
4.18260E+01	0.00000E+00	Evanescent

Impressed (active)

For a FEM excitation (impressed current source) the following information is provided:

EXCITATION BY IMPRESSED CURRENT ELEMENT (FEM)

```

Name:                      CurrentSource1
Number of excitation:      N =          1
Frequency in Hz:           FREQ =  3.00000E+09
Wavelength in m:           LAMBDA =  9.99308E-02
Amplitude in A:            |I| =  1.00000E+00
Phase in deg.:             ARG(I) =          0.00
Element location in m:     X =          0.00000E+00      start point      end point
                           Y =          6.50000E-03      0.00000E+00
                           Z =          -5.00000E-04      6.50000E-03
Element length in m:       LEN =          5.00000E-04      -1.00000E-03

```

If an incident plane wave is used then the output file has the following format:

EXCITATION BY INCIDENT PLANE ELECTROMAGNETIC WAVE

```

Name:                      PlaneWave1
Number of excitation:      N =          1
Frequency in Hz:           FREQ =  1.00000E+08
Wavelength in m:           LAMBDA =  2.99792E+00
Direction of incidence:    THETA =  20.00  PHI =  50.00
Polarisation:              LINEAR
Axial ratio:               V =          0.0000
Dir. of polarisation:      ETA =  60.00
Direction of propag.:      BETA0X = -4.60764E-01
                           BETA0Y = -5.49117E-01
                           BETA0Z = -1.96945E+00
Field strength in V/m:     |EOX| =  9.65425E-01  ARG(EOX) = -180.00
(Phase in deg.)           |EOY| =  1.96747E-01  ARG(EOY) =   0.00
                           |EOZ| =  1.71010E-01  ARG(EOZ) =   0.00

```

The vector $\vec{\beta}$, whose components are given, is the vector which points in the direction of propagation. The vector $\vec{E}_0$ represents the direction of the electric field.

For an impressed aperture excitation, the following information is given:

EXCITATION BY AN APERTURE

```

Number of the aperture:    1
Frequency in Hz:           FREQ =  1.25000E+10
Wavelength in m:           LAMBDA =  2.39834E-02
No. of dipoles electr.:    1296
                           magnet.:  1296
Extent of the aperture:    X = -1.70096E-02 ... 1.34267E-01 m
                           Y = -7.56383E-02 ... 7.56383E-02 m
                           Z = -7.59272E-02 ... 7.59272E-02 m

```

No specific information regarding the magnitude and phase of the dipole elements that make up the excitation is given in the output.

Excitation by an impressed radiation pattern point source is shown in the output information as:

EXCITATION BY IMPRESSED RADIATION PATTERN

```

Name:                      RadiationPattern1
Number of the excitation:  N =      1
Frequency in Hz:          FREQ = 1.60000E+09
Wavelength in m:          LAMBDA = 1.87370E-01
Max. field strength * dist. in V: 5.90127E+01 * PWFAKTOR (see below)
Radiated power in W:      1.12011E+00 * PWFAKTOR**2 (see below)
Directivity of the antenna in dB: 17.148
Distance for far-field cond. in m: 1.96884E+00
Local co-ordinate system of the source:
  Number of grid points NTHETA = 37
                                NPHI  = 73
  Angular range THETA in deg.: 0.00 ... 180.00
                                PHI   in deg.: 0.00 ... 360.00
  Source position in m:      X = -2.16000E-01
                             Y = 0.00000E+00
                             Z = 0.00000E+00
  Rotation angle in deg. with respect to the global co-ordinate system
    around X-axis: 0.00000E+00
    around Y-axis: 0.00000E+00
    around Z-axis: 0.00000E+00

```

No specific information regarding the magnitude and phase of the impressed pattern is given.

For an impressed spherical mode excitation, the following information is written to the output:

EXCITATION BY IMPRESSED SPHERICAL MODE

```

Name:                      SphericalSource1
Number of the excitation:  N =      1
Number of modes:          MODES = 880
Frequency in Hz:          FREQ = 1.25000E+10
Wavelength in m:          LAMBDA = 2.39834E-02
Propagation direction:     C =      4 (outwards)
Mode location in m:       X = 0.00000E+00
                           Y = 0.00000E+00
                           Z = 0.00000E+00

```

mode indices				coefficient in sqrt(Watt)		rad. power(Watt)
J	S	M	N	magn.	phase	power
3	1	0	1	9.21068E-02	-88.00	4.24183E-03
4	2	0	1	7.28918E-18	-16.25	2.65661E-35
11	1	0	2	1.49455E-17	-89.63	1.11684E-34
...						

Point source type (Hertzian dipole) excitations result in the following information output:

EXCITATION BY HERTZIAN DIPOLE

```

Name:
Number of excitation:  N =      1
Frequency in Hz:       FREQ = 2.99792E+08
Wavelength in m:       LAMBDA = 1.00000E+00
Amplitude in Am:       |IL| = 1.00000E+00
Phase in deg.:         ARG(IL) = 0.00
Dipole location in m:  X = 2.00000E+00
                       Y = 0.00000E+00
                       Z = 0.00000E+00
Orientation of dipole:  THETA = 0.00
                       PHI   = 0.00

```

19.3 Currents and charges

The OS card can request the current distribution. The following data is given for each triangle

Current request with name: Currents1

VALUES OF THE CURRENT DENSITY VECTOR ON TRIANGLES in A/m (no averaging)

Triangle number	centre x/m	y/m	z/m	JX magn.	phase	JY magn.	phase	JZ magn.	phase	...
1	1.70187E+00	7.99633E-02	0.00000E+00	2.674E-02	98.24	2.687E-03	-74.34	0.000E+00	0.00	
2	1.81048E+00	1.96490E-01	0.00000E+00	6.015E-03	95.34	4.069E-03	96.46	0.000E+00	0.00	
3	3.94881E-01	8.42417E-02	0.00000E+00	1.456E-01	137.29	3.578E-03	80.18	0.000E+00	0.00	
4	1.97230E-01	8.42417E-02	0.00000E+00	2.087E-01	141.97	9.018E-02	141.94	0.000E+00	0.00	
5	9.92519E-02	2.03608E-01	0.00000E+00	9.135E-02	141.60	1.895E-01	141.57	0.000E+00	0.00	

Current magnitude in the
3 corner points

1.984E-02	2.979E-02	3.175E-02
1.912E-02	1.961E-02	8.007E-03
1.503E-01	1.438E-01	1.433E-01
1.433E-01	1.918E-01	1.216E+03
1.216E+03	1.683E-01	1.551E-01

At the position (x, y, z) the current density vector $\vec{J}$ in the complex form is given. The last three columns indicate the value for the surface current density in the three vertices of the triangles. (Note that the value of the current written in the *.out file will be affected if averaging of the currents is de-activated at the OS card. If averaging is requested, the average of the current at the vertices of all three adjacent triangles is shown). 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/m²

Triangle number	SIGMA magn.	phase
1	1.42700E-10	15.66
2	3.34062E-10	5.41
3	1.80384E-10	170.18
4	2.57440E-10	176.74
5	2.43734E-10	174.98

The current on the segments is written as

VALUES OF THE CURRENT IN THE SEGMENTS in A

Segment number	centre			IX		...
	x/m	y/m	z/m	magn.	phase	
1	0.00000E+00	0.00000E+00	9.60000E-01	0.000E+00	0.00	
2	0.00000E+00	0.00000E+00	8.70000E-01	0.000E+00	0.00	
3	0.00000E+00	0.00000E+00	7.70000E-01	0.000E+00	0.00	
4	0.00000E+00	0.00000E+00	6.70000E-01	0.000E+00	0.00	
5	0.00000E+00	0.00000E+00	5.70000E-01	0.000E+00	0.00	
			IY	IZ		
			magn.	phase	magn.	phase
			0.000E+00	0.00	2.202E-02	-38.72
			0.000E+00	0.00	6.907E-02	-38.55
			0.000E+00	0.00	1.172E-01	-38.30
			0.000E+00	0.00	1.614E-01	-38.03
			0.000E+00	0.00	2.009E-01	-37.73

With the associated charge

VALUES OF THE LINE CHARGE DENSITY ON SEGMENTS in As/m

Segment number	Q	
	magn.	phase
1	1.16913E-09	-128.72
2	1.06276E-09	-128.23
3	9.82679E-10	-127.64
4	8.92312E-10	-126.94
5	7.85966E-10	-126.07

For every voltage source the current at the feed point is determined and thus the impedance. The following is the result:

DATA OF THE VOLTAGE SOURCE NO. 1

		real part	imag. part	magn.	phase
Current	in A	2.4933E-01	-1.7696E-01	3.0574E-01	-35.37
Admitt.	in A/V	1.5541E-02	-1.1030E-02	1.9058E-02	-35.37
Impedance	in Ohm	4.2790E+01	3.0371E+01	5.2473E+01	35.37
Inductance	in H	6.4494E-08			

Power in Watt: 2.00000E+00

19.4 Finite conductivity

Firstly the block with the set of characteristics for the single labels is displayed:

DATA OF LABELS

Label Cuboid1.Face6: Skin = 3 Load = 0 Coating = 0
 Triangle 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: 5.81353E-03 m

All segments and triangles without a listed label are perfectly conducting

After the calculation of the currents the losses that result from finite conductivity are displayed.

POWER LOSS METAL (in Watt)

Label	in the segments				in the triangles
	skineffect	conc.load	distr.load	coating	
Cuboid1.Face6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.3348E-06
Cuboid1.Face1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	3.7427E-07
Cuboid1.Face3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0371E-06
Cuboid1.Face4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8.1218E-06
Cuboid1.Face5	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0371E-06
Cuboid1.Face2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.3348E-06
total	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.3240E-05

Total loss in the segments: 0.0000E+00 W

Total loss in the triangles: 1.3240E-05 W

Loss metal (total): 1.3240E-05 W

SUMMARY OF LOSSES

Metallic elements:	1.3240E-05 W
Dielectric (surface equiv. princ.):	0.0000E+00 W
Dielectric (volume equiv. princ.):	0.0000E+00 W
Dielectric (FEM region):	0.0000E+00 W
Mismatch at feed:	0.0000E+00 W
Non-radiating networks:	0.0000E+00 W
Backward power at passive waveguide ports:	0.0000E+00 W
Sum of all losses:	1.3240E-05 W
Efficiency of the antenna:	99.7266 %
(based on a total active power:	4.8418E-03 W)

In the first column the label is displayed, the lowest row displays the sum of losses in all labels.

19.5 Near fields and SAR

If near fields are calculated, the extract shown below is given for each solution request.

The position as well as the individual components of the electric and the magnetic field strength are given. This is — if not otherwise requested at the FE card — the total value of the field, i.e. the sum of the incident wave and the scattered field.

Near-field request with name: NearField1

VALUES OF THE MAGNETIC FIELD STRENGTH in A/m

(total field, incident and scattered)

The following offset is in effect for the co-ordinates:
X= 0.00000E+00 m Y= 0.00000E+00 m Z= 0.00000E+00 m

medium	LOCATION			EX		EY		...
	X/m	Y/m	Z/m	magn.	phase	magn.	phase	
0	0.00000E+00	0.00000E+00	-1.00000E+00	6.70088E+01	99.86	7.65636E-01	166.42	
0	1.00000E-01	0.00000E+00	-1.00000E+00	6.46235E+01	74.23	1.14589E+00	166.13	
0	2.00000E-01	0.00000E+00	-1.00000E+00	6.23014E+01	47.98	1.55289E+00	165.83	
0	3.00000E-01	0.00000E+00	-1.00000E+00	5.99908E+01	21.51	1.95743E+00	163.95	
				...		EZ		
						magn.	phase	
						6.89061E+01	126.74	
						7.17685E+01	98.76	
						7.35678E+01	70.70	
						7.41473E+01	42.30	

If the electric field inside dielectric cuboids is determined, then the value for the SAR (*specific absorption rate*) and the cuboid number are also given:

VALUES OF THE ELECTRIC FIELD STRENGTH in V/m

inside the dielectric cuboids

X/m	LOCATION		EX		EY		EZ		...
	Y/m	Z/m	magn.	phase	magn.	phase	magn.	phase	
0.050	0.050	0.050	5.776E+00	59.89	1.259E+01	-177.82	1.415E+02	-125.12	
0.050	0.050	0.150	2.192E+01	33.75	4.114E+00	122.93	1.640E+02	-130.45	
0.050	0.050	0.250	2.584E+01	31.18	3.420E+00	19.21	1.679E+02	-137.51	
0.050	0.050	0.350	2.625E+01	22.29	8.499E+00	-24.72	1.551E+02	-144.87	
				...		SAR		cuboid no.	
						in W/kg			
						0.000E+00		1	
						0.000E+00		2	
						0.000E+00		3	
						0.000E+00		4	

For specific SAR solution requests, the following output is shown (note that the extract shown below is representative for a volume-average SAR calculation — the output for other options like spatial peak SAR calculations will differ):

AVERAGE SPECIFIC ABSORPTION RATE in W/kg				
Average SAR for the entire domain:		6.47236E+00 W/kg		
AVERAGE SPECIFIC ABSORPTION RATE in W/kg				
medium	relative	conductivity	relative	...
	permittivity	in S/m	permeability	
air	1.00000	0.000000E+00	1.00000	
dome	9.50000	1.585526E-03	1.00000	
	magnetic	density	volume	SAR
	conductivity	in kg/m^3	in m^3	in W/kg
	0.000000E+00	1.000000E+03	1.567165E-06	0.000000E+00
	0.000000E+00	1.000000E+03	4.036295E-06	6.472361E+00

19.6 Far fields and receiving antenna

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 $e^{-j\beta R}/R$ not considered									
The following offset is in effect for the co-ordinates:									
X= 0.00000E+00 m Y= 0.00000E+00 m Z= 0.00000E+00 m									
LOCATION		ETHETA		EPhi		directivity in dB ...			
THETA	PHI	magn.	phase	magn.	phase	vert.	horiz.	total	
90.00	0.00	1.235E+00	168.98	0.000E+00	0.00	7.1722	-999.9999	7.1722	
90.00	2.00	1.233E+00	168.90	0.000E+00	0.00	7.1583	-999.9999	7.1583	
90.00	4.00	1.227E+00	168.65	0.000E+00	0.00	7.1166	-999.9999	7.1166	
POLARISATION									
						axial	r.	angle	direction
						0.0000	180.00		LINEAR
						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 4.88015E-03 W									
and on a power loss of 0.00000E+00 W									

For incident plane waves, the values that are displayed here are the values of the scattered field, i.e. the incident field is not taken into account. However, for any other sources (e.g. elementary Hertzian dipoles or impressed radiation pattern or transmission line), the fields radiated by the excitation are considered.

In the far field a complex field strength $\vec{E}_{\text{far}}$ is defined using the relation

$$\lim_{R \rightarrow \infty} \vec{E}(\vec{r}) = \frac{e^{-j\beta_0 R}}{R} \vec{E}_{\text{far}} \quad (19-1)$$

with a large distance $R = |\vec{r}|$ which tends to infinity (and which in the FEKO calculations is identical to infinity). Please note that the dimension of $\vec{E}_{\text{far}}$ is voltage due to this extra distance factor R .

In the *.out file the ϑ (vertical) and φ (horizontal) components of $\vec{E}_{\text{far}}$ are tabulated by magnitude and phase, i.e. $\vec{E}_{\text{far},\vartheta}$ and $\vec{E}_{\text{far},\varphi}$. With POSTFEKO also results for other polarisations can be extracted. The corresponding formulas used are then

$$\vec{E}_{\text{far},S} = \frac{1}{\sqrt{2}} (\vec{E}_{\text{far},\varphi} - \vec{E}_{\text{far},\vartheta}) \quad (19-2)$$

for S-polarisation,

$$\vec{E}_{\text{far},Z} = \frac{1}{\sqrt{2}} (\vec{E}_{\text{far},\varphi} + \vec{E}_{\text{far},\vartheta}) \quad (19-3)$$

for Z-polarisation,

$$\vec{E}_{\text{far},\text{LHC}} = \frac{1}{\sqrt{2}} (\vec{E}_{\text{far},\varphi} + j\vec{E}_{\text{far},\vartheta}) \quad (19-4)$$

for left-hand circular polarisation, and

$$\vec{E}_{\text{far},\text{RHC}} = \frac{1}{\sqrt{2}} (\vec{E}_{\text{far},\varphi} - j\vec{E}_{\text{far},\vartheta}) \quad (19-5)$$

for right-hand circular polarisation.

If the excitation is an incident wave, the results include the radar cross section. In the case of voltage sources, the gain or directivity is included (see the parameters of the FF card).

For the radar cross section, the incident plane wave with complex amplitude $\vec{E}_0$ carries a power density of

$$S_i = \frac{1}{2} \cdot \frac{|\vec{E}_0|^2}{Z_{F0}} \quad (19-6)$$

(Z_{F0} denotes the wave impedance of the surrounding medium) which gets scattered on the object and a wave is reflected with the scattered power density

$$S_s = \frac{1}{2} \cdot \frac{|E_\vartheta|^2 + |E_\varphi|^2}{Z_{F0}} . \quad (19-7)$$

The radar cross section (RCS) σ is then defined as

$$\sigma_{\text{Total}} = \lim_{R \rightarrow \infty} 4\pi R^2 \frac{S_s}{S_i} = \lim_{R \rightarrow \infty} 4\pi \frac{|RE_\vartheta|^2 + |RE_\varphi|^2}{|\vec{E}_0|^2} = 4\pi \frac{|E_{\text{far},\vartheta}|^2 + |E_{\text{far},\varphi}|^2}{|\vec{E}_0|^2} ; \quad (19-8)$$

$$\sigma_{\text{Horizontal}} = \lim_{R \rightarrow \infty} 4\pi \frac{|RE_\varphi|^2}{|\vec{E}_0|^2} = 4\pi \frac{|E_{\text{far},\varphi}|^2}{|\vec{E}_0|^2} ; \quad (19-9)$$

$$\sigma_{\text{Vertical}} = \lim_{R \rightarrow \infty} 4\pi \frac{|RE_\vartheta|^2}{|\vec{E}_0|^2} = 4\pi \frac{|E_{\text{far},\vartheta}|^2}{|\vec{E}_0|^2} . \quad (19-10)$$

For antenna and general radiation problems, as mentioned above, FEKO is computing either the gain or the directivity depending on the FF card setting (this applies to the values tabulated in the

*.out file only, in POSTFEKO any quantity can be selected). Let us assume that P_t is the source power and P_v are some losses in the structure (e.g. dielectric losses), then a power $P_r = P_t - P_v$ will be radiated. The directivity (as compared to an isotropic point source) is then defined as

$$D_{Total} = 4\pi R^2 \frac{S_s}{P_r} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\theta}|^2 + |E_{far,\varphi}|^2}{P_r} ; \quad (19-11)$$

$$D_{Horizontal} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\varphi}|^2}{P_r} ; \quad (19-12)$$

$$D_{Vertical} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\theta}|^2}{P_r} . \quad (19-13)$$

For the gain a similar definition is used, just now the source power P_t and not the radiated power P_r is acting as reference:

$$G_{Total} = 4\pi R^2 \frac{S_s}{P_t} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\theta}|^2 + |E_{far,\varphi}|^2}{P_t} ; \quad (19-14)$$

$$G_{Horizontal} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\varphi}|^2}{P_t} ; \quad (19-15)$$

$$G_{Vertical} = \frac{2\pi}{Z_{F0}} \cdot \frac{|E_{far,\theta}|^2}{P_t} . \quad (19-16)$$

Between gain and directivity there is the relation

$$\frac{G}{D} = \frac{P_r}{P_t} = \frac{P_t - P_v}{P_t} = \eta, \quad (19-17)$$

where η represents the antenna efficiency.

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 Figure 19-1. The coordinates are $\vec{e}_r$, $\vec{e}_\theta$ and $\vec{e}_\varphi$, and the view is in the direction of the propagation of the wave ($\vec{e}_r$).

In order to evaluate these quantities, let us define the magnitude and phase of the far field components as

$$E_{far,\theta} = A \cdot e^{j\alpha} \quad E_{far,\varphi} = B \cdot e^{j\beta} . \quad (19-18)$$

Using the abbreviation $\tau = \omega t - \beta_{0r}$ one finds the temporal field strength vector in space as

$$\vec{E}(\tau) = \frac{A}{r} \cdot \cos(\tau + \alpha) \cdot \vec{e}_\theta + \frac{B}{r} \cdot \cos(\tau + \beta) \cdot \vec{e}_\varphi . \quad (19-19)$$

This equation describes then the polarisation ellipse of Figure 19-1. One can find the minimum and maximum values of the field strength magnitude at these times

$$\tau_1 = -\frac{1}{2} \cdot \arctan \frac{A^2 \cdot \sin(2\alpha) + B^2 \cdot \sin(2\beta)}{A^2 \cdot \cos(2\alpha) + B^2 \cdot \cos(2\beta)} \quad (19-20)$$

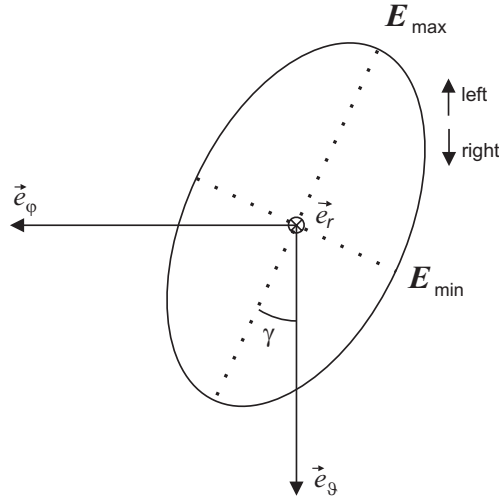


Figure 19-1: Elliptic polarisation in the far field

and

$$\tau_2 = \tau_1 + \frac{\pi}{2} . \quad (19-21)$$

Let $E_1 = |\vec{E}(\tau_1)|$ and $E_2 = |\vec{E}(\tau_2)|$ and assuming that we have $E_1 > E_2$, then according to Figure 19-1 we have $E_{max} = E_1$ and $E_{min} = E_2$.

The axial ratio (Minor/Major) is defined as

$$v = \frac{E_{min}}{E_{max}} = \frac{E_2}{E_1} \quad (19-22)$$

The axial ratio (Major/Minor) is defined as

$$v = \frac{E_{max}}{E_{min}} = \frac{E_1}{E_2} \quad (19-23)$$

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 $\vec{e}_\theta$ and can be computed using

$$\gamma = \arctan \frac{B \cdot \cos(\tau_1 + \beta)}{A \cdot \cos(\tau_1 + \alpha)} . \quad (19-24)$$

If the FF card is used with $N\theta \geq 2$ and $N\phi \geq 2$ the Poynting vector is integrated over the specified sector, see the detailed discussion given with the FF card (see section 14.37). The result is the radiated power and is given below the field values.

When analysing an antenna the source power (calculated from the input impedance) should equal the integral of the radiated power over the surface of a sphere. This may be used as a partial validation of the result. Note that power losses in dielectrics and finite conductivity should be taken into account separately.

The user may also elect to integrate the far field power without writing the field values to the output file (using the FF card with FFREQ = 3). FEKO then produces the output

VALUES OF THE SCATTERED ELECTRIC FIELD STRENGTH IN THE FAR FIELD in V
Factor $e^{-j\beta R}/R$ not considered

The following offset is in effect for the co-ordinates:
X= 0.00000E+00 m Y= 0.00000E+00 m Z= 0.00000E+00 m

Integration of the normal component of the Poynting vector in the angular
grid DTHETA = 2.00 deg. and DPHI = 1.00 deg. (1810 sample points)
angular range THETA angular range PHI radiated power
-181.00 .. 181.00 deg. -0.50 .. 9.50 deg. 2.82057E-04 Watt
-180.00 .. 180.00 deg. 0.00 .. 9.00 deg. 2.53851E-04 Watt

Polarisation dependent radiated power:

horizontal polarisation: 0.00000E+00 Watt (0.00 %)
vertical polarisation: 2.53851E-04 Watt (100.00 %)
S polarisation: 1.26925E-04 Watt (50.00 %)
Z polarisation: 1.26925E-04 Watt (50.00 %)
left hand circular pol.: 1.26925E-04 Watt (50.00 %)
right hand circular pol.: 1.26925E-04 Watt (50.00 %)

where the first line of total power is calculated assuming that each specified point lies at the centre of an incremental integration area. The effective area is therefore slightly larger than the area defined by the start and end angles. The second line gives the power integrated over an area defined by the start and end angles. For example, one may request an integration from $\varphi=0^\circ$ to $\varphi=350^\circ$ and $\vartheta=5^\circ$ to $\vartheta=175^\circ$ both in 10° increments in which case the first total will give the total power through the sphere. One may also request the integration from $\varphi=0^\circ$ to $\varphi=360^\circ$ and $\vartheta=0^\circ$ to $\vartheta=180^\circ$ in which case the second total will give the correct total power through the sphere.

The polarisation dependent power on the second block is calculated according to the effective area of the second line.

When using a receiving antenna, the received power and phase of the received signal is given as follows:

Ideal receiving antenna with name: ReceivingAntenna1

RECEIVED POWER IDEAL RECEIVING ANTENNA

Received power (ideal match assumed): 3.2871E-03 W

Relative phase of received signal: 9.0641E+01 deg.

19.7 S-parameters

If S-parameters have been requested with an SP card, FEKO prints different tables to the output file. The first lists the impedance at each port (all sources that are active when the SP card is processed are considered as ports).

S-PARAMETER REFERENCE IMPEDANCES AT PORTS

port	impedance in Ohm
1	5.00000E+01
2	1.00000E+02
3	5.00000E+01

Then the S-parameters are listed for each source as shown below. Note that sources whose amplitude are set to exactly zero are only used as sink ports, i.e. they are not excited and no such block is created. All the ports are loaded and FEKO therefore also writes this information to the

output file. The second data line below gives S_{21} or the coupling to port 2 when port 1 is excited. In the second block here under the first line gives S_{13} or the coupling into port 1 when port 3 is excited.

SCATTERING PARAMETERS							
ports				magnitude		phase	
sink	source	real part	imag. part	linear	in dB	in deg.	
S	1 3	-1.50775E-01	3.89893E-02	1.55735E-01	-16.1523	165.50	
S	2 3	7.64835E-01	3.34088E-01	8.34618E-01	-1.5702	23.60	
S	3 3	4.46365E-01	-1.47829E-01	4.70207E-01	-6.5542	-18.32	
Sum S ^2 of these S-parameters:				9.41935E-01	-0.2598		
...							
SCATTERING PARAMETERS							
ports				magnitude		phase	
sink	source	real part	imag. part	linear	in dB	in deg.	
S	1 1	1.50829E-01	-8.81429E-01	8.94240E-01	-0.9709	-80.29	
S	2 1	1.78833E-01	-1.48216E-01	2.32270E-01	-12.6802	-39.65	
S	3 1	-1.24083E-01	1.07973E-01	1.64484E-01	-15.6775	138.97	
Sum S ^2 of these S-parameters:				8.80670E-01	-0.5519		

19.8 Computation time and peak memory

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	0.750	0.750
Checking the geometry	0.500	0.500
Initialisation of the Greens function	0.000	0.000
Calcul. of coupling for PO/Fock	0.000	0.000
Calcul. of the FMM transfer function	0.688	0.687
Fourier transform of FMM basis funct.	3.453	3.453
Calcul. of matrix elements	214.187	214.187
Calcul. of right-hand side vector	0.016	0.016
Preconditioning system of linear eqns.	709.953	710.000
Solution of the system of linear eqns.	58.640	58.641
Determination of surface currents	0.000	0.000
Calcul. of impedances/powers/losses	1.578	1.578
Calcul. of averaged SAR values	0.000	0.000
Calcul. of power ideal receiving ant.	0.000	0.000
Calcul. of cable coupling	0.000	0.000
Calcul. of electric near field	3.688	3.687
Calcul. of magnetic near field	0.000	0.000
Calcul. of far field	0.000	0.000
other	3.688	3.641

total times:	997.141	997.140
(total times in hours:	0.277	0.277)

This table is followed by an output of the peak memory (main memory, excluding possible out-of-core files) usage which FEKO encountered during any solution phase:

Peak memory usage during the whole solution: 516.765 MByte

Part VIII

FEKO Utilities

20 The optimiser OPTFEKO

20.1 Introduction

Optimisation is the process of changing parameters in a model to achieve a specific predefined response or state. The parameters of the optimisation are usually associated with geometric dimensions, material properties or excitations/loadings. For example, the gain of a horn antenna may be maximised by varying the size of the horn aperture.

The component of the FEKO Suite that controls the optimisation process is OPTFEKO. OPTFEKO requires two components for successful execution. The first is a parametric FEKO model, and the second is an *.opt file specifying how this model is to be optimised. A parametric FEKO model may be built up in various ways, and will consist of at least a *.pre file, or a *.cfx file — or possibly both.

As discussed in *Setting up optimisation in CADFEKO* (see section 7), all options relating to optimisation are specified through the CADFEKO interface and is stored in the *.opt file. The parametric model may be prepared using CADFEKO (see section 4) and/or EDITFEKO (see section 11).

For completeness sake, we will give a brief summary of the concepts involved in setting up a specific optimisation in CADFEKO.

Optimisation is based on *Searches*, comprising a number of parts

- *Method* to be used for the search (including method settings regarding accuracy and stopping criteria)
- *Parameters* for the search
- *Goals* of the search

The *Method* defines how the optimisation search will be performed. The stopping criteria and accuracy define under what circumstances the search will be terminated. The optimisation *Parameters* define the range in which the search will be performed and the *Goals* specify the desired result of the optimisation process.

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 (the relevant case is chosen automatically based on the *Goal* definitions) using the optimisation algorithm indicated by the *Method* selection.

20.2 The optimisation method

During optimisation, new values for parameters are chosen based on an algorithm known as the optimisation method. OPTFEKO provides various optimisation methods, each with different characteristics that are suitable for a different set of problems. Specifying the most appropriate 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. In this section we will shortly discuss each of the methods available in OPTFEKO based on the algorithm that is used, the termination criteria as well as other general method-specific factors. We will also consider the text-log that is produced using each method.

20.2.1 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 in Figure 20-1

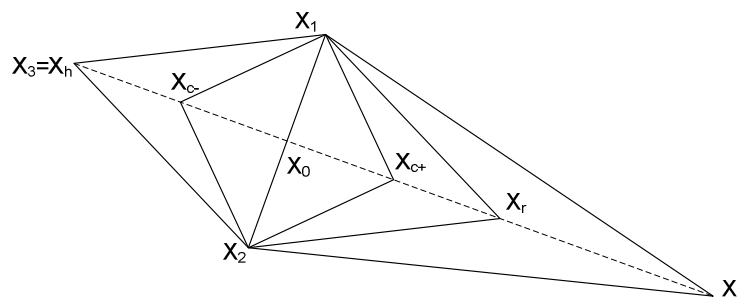


Figure 20-1: Reflection, Expansion and Contraction for the Simplex method

Reflection

Considering the diagram in Figure 20-1, 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 20-1 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 is 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.

The *Simplex* method can be summarised as shown in Table 20-1. (The $F(X)$ operator represents the evaluation of the fitness at the point X in the parameter space.)

Table 20-1: A summary of the Simplex operations

<i>Objective Function</i>	<i>Operation</i>
$F(X_r) < F(X_l)$	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

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

Failure during re-evaluation and meshing (in the CADFEKO batch meshing tool or in PREFEKO) for a specific set of parameters (including violation of boundaries placed on parameter values) is treated by writing an appropriate error message to the *.log file. A fictitious fitness value is assigned to such a parameter set and the simplex will continue. If too many consecutive parameter sets result in evaluation failures, or if failure of a parameter set occurs before the initial simplex has been established, OPTFEKO will terminate with an error.

The text log of the Simplex 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. When the Simplex (Nelder-Mead) method is used, the log file structure is as shown below.

Section 1: General information regarding the optimisation setup.

```
===== L O G - F I L E - O P T F E K O =====

Version: 13.22 of 2007-05-08
Date: 2007-06-06 16:45:43
File: test

OPTIMISATION WITH 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

```

20.2.2 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 been shown in certain instances to outperform other methods of optimisation like Genetic Algorithms (section 20.2.3).

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 (section 20.2.3), 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. When the PSO method is used, the log file structure is as shown below.

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 PSO method parameters.

```
===== 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: Information regarding the parameter values, goal values and PSO best aims at each iteration.

```
===== 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  7.732876627e-01  7.732876627e-01
  3  2.000000000e+00          2.267123373e-01  7.732876627e-01  7.732876627e-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.

```

===== 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

```

20.2.3 Genetic algorithm (GA)

Genetic algorithm (GA) optimisers are robust, stochastic search methods modelled on the Darwinian principles and concepts of natural selection and evolution. Like the PSO method (see section 20.2.2), GAs 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 (C), this needs to be converted into a generation size (A) and number of iterations (B), 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 Genetic 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. When the GA method is used, the log file structure is as shown below.

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

```

20.2.4 Grid search

This method is strictly speaking not an optimisation method. The optimisation parameters are linearly varied between their minimum and the maximum values in a pre-defined number of steps.

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 pre-defined 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 as shown in Equation 20-1.

$$N_{\text{solver runs}} = \prod_{i=1}^{N_{\text{parameters}}} N_{\text{points}}(i) \quad (20-1)$$

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. When the Grid search method is used, the log file structure is as shown below.

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

20.3 Sensitivity analysis

OPTFEKO computes an indication of the sensitivity of the goal function w.r.t. each parameter upon termination of an optimisation using the PSO, GA or Simplex methods, if and only if sufficient information is available. The computed 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.

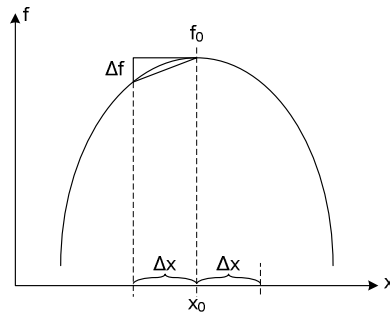


Figure 20-2: Sensitivity analysis of the goal function f w.r.t the parameter x .

Figure 20-2 shows an example goal function f that varies as a function of the parameter x . The sensitivity w.r.t. parameter x can be described by the following equation:

$$S(x) = \left. \frac{\partial f}{\partial x} \right|_{x_0 \pm \Delta x} \quad (20-2)$$

with Δx equal to 1

$$\Delta x = 0.01(x_{\max} - x_{\min}). \quad (20-3)$$

Solving Equation 20-2 directly, 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

$$S(x) \approx \left. \frac{\partial f}{\partial x} \right|_{x_0 \pm \Delta x} \approx \left. \frac{\partial^2 f}{\partial x^2} \right|_{x_0} \cdot \Delta x. \quad (20-4)$$

to finally give the sensitivity w.r.t. x as

$$S(x) \approx f''(x_0) \cdot 0.01(x_{\max} - x_{\min}). \quad (20-5)$$

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.

20.4 Running OPTFEKO

Once a parametric FEKO model and accompanying *.opt file have been created, the optimiser may be launched. The optimiser may be launched from any of the FEKO GUI components (*Run* → *OPTFEKO*, or using the <Alt><6> shortcut key). Optimisation options may be set in the *Run* → *Component parameters* → *Utilities* tab. When the optimiser is launched from the GUI, an execution window will open in which text information regarding the optimisation progress is displayed.

OPTFEKO may also be launched from the command line using the command:

```
optfeko dipole
```

Where *dipole* is the name of the optimisation project. On the command line the following parameters may be added:

- `--version` Output only the version information to the command line and terminate.
- `-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 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 etc.).
- `-np x` The number of processors to be used for farming out of the individual optimisation steps (see below).
- `--machines-file machname` The file *machname* is the machines file with the node names and the number of CPUs to be used for farming of the individual optimisation steps. This machines file is used for both farming as well as parallel execution when farming and parallel execution is used simultaneously. Some examples regarding this topic is discussed below.

- `--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 will be used in the launcher RUNFEKO for the FEKO kernel. For instance in order to use the parallel FEKO solver during the optimisation one can use the command

```
optfeko file --runfeko-options -np 2
```

or also

```
optfeko file --runfeko-options -np 2 --machines-file m
```

where *m* indicates the machines file. For a remote execution of the FEKO runs during the optimisation on another host, the suitable command would be

```
optfeko file --runfeko-options --remote-host hostname
```

Additional options for ADAPTFEKO and PREFEKO may also be included in the OPTFEKO command as part of the RUNFEKO options. These will be passed to the relevant component by RUNFEKO as needed. This allows for control of all of the FEKO components during the optimisation process. See section 18.2 for a list of all such RUNFEKO options.

Farming and advanced control

Farming out of the steps of an optimisation involves the concurrent solution of various optimisation steps on a number of available processors or hosts. Farming may not be used with the *Simplex* method.

When farming out the individual optimisation steps, the number of processes to start on each available host is specified in a so-called *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. For example, if two hosts are available with names *host1* and *host2* (this is the output of the UNIX command *hostname*), and 4 and 8 processors respectively, the machines file will be

```
host1:4
host2:8
```

Launching an optimisation run with 12 processors for farming using this machine file would cause the first 4 optimisation steps to be launched on *host1* and the next 8 steps to be launched on *host2*. The example command is running model *<x>* and farming out to *<n_farming>* processors listed in the machines file *<m>*.

```
optfeko <x> -np <n_farming> --machines-file <m>
```

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.

Some large problems may require that FEKO be run in parallel (simulation spread over more than one host or processor - not the same as farming.) This is possible by adding the number of cores that FEKO should use as a RUNFEKO option. The example command is running model *<x>*

and each solution is using `<n_parallel>` processors listed in the machines file `<m>`. Note the similarity and the differences between this command and the previous command.

```
optfeko <x> --runfeko-options -np <n_parallel> --machines-file <m>
```

It is also possible to use both the power of farming and parallel computing to optimally utilise resources. The example command shows how to run model `<x>` by farming to `<n_farming>` processors while running every solution on `<n_parallel>` processors. The available processors are listed in a single machines file `<m>`. OPTFEKO automatically creates machines files for every solution that is farmed out.

```
optfeko <x> -np <n_farming> --machines-file <m> --runfeko-options -np <n_parallel>
```

OPTFEKO output

Information on the optimisation process is stored in a log file with the extension `.log` — in the example above the file name will be `dipole.log`. The structure of this file is dependent on the optimisation method, and it is therefore discussed in the sections dealing with each of the method algorithms.

It should be noted that when using the remote launching facility (see section [18.2.3](#)) or farming out of optimisation steps, the actual optimisation is done on the local machine, only the FEKO kernel 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* button in the GUI process information window, or `<Ctrl><C>` in a command line. Confirmation of the interruption will be requested, and if given, the optimisation process will be stopped. 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.

21 The program ADAPTFEKO

21.1 Description

In examples with narrow resonances a fine frequency resolution is required to locate these resonances. If the frequency band is large, a very large number of analyses may be required when simple linear or multiplicative frequency stepping is used. ADAPTFEKO is used to overcome these problems. It uses adaptive frequency sampling and interpolation methods to automatically choose smaller frequency steps near resonances and larger steps where the results are relatively smooth.

For each frequency ADAPTFEKO creates a *.pre file and calls PREFEKO and FEKO. 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, ...).

21.2 Running ADAPTFEKO

ADAPTFEKO is started automatically by RUNFEKO (see section 18.2) if a *Continuous (interpolated) range* is selected in the CADFEKO *Solution frequency* setting or if the FR card (see section 14.38) defining the range of solution frequencies contains the adaptive frequency sampling option.

The syntax is:

```
runfeko filename
```

or

```
runfeko filename --adaptfeko-options [options]
```

where the optional argument options in the second line may be

- version Output only the version information to the command line and terminate.
- keep-files All solution files (*.pre, *.fek, *.out, etc.) 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.)

21.3 The *.pre input file for adaptive sampling

The *.pre file is created (manually in EDITFEKO or automatically by CADFEKO) as for linear or multiplicative stepping.

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 solution frequency-based variation (for example, adaptive meshing). One 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 (an example of this usage shown below).

```
** define a frequency variable if it is not already defined by an ADAPTFEKO run
!!if (not(defined(#adaptfreq))) then
#adaptfreq = 250.0E6
!!endif
```

Note that care must be taken when using adaptive meshing with ADAPTFEKO. Small discontinuities that may result from changes in the mesh can influence the convergence and accuracy of the adaptive sampled results dramatically.

22 The program TIMEFEKO

22.1 Description

With the program TIMEFEKO electromagnetic scattering problems can be solved in the time domain. It is based on the program FEKO (that does the relevant calculations in the frequency domain) and an FFT algorithm (that transforms the data to the time domain).

For the excitation a number of different pulses have been pre-defined and stored in the function library (see section 22.3.1).

The program TIMEFEKO is constructed in such a way that all the data in the output file (*.out) from FEKO is transformed, i.e. in the input file the cards have to be specified in the correct way to ensure that the correct data is transformed. Information on the correct card definitions is given in the following section.

The program TIMEFEKO uses two input files,¹¹ that have to be created by the user.

22.2 The *.pre input file for TIMEFEKO

The input file with extension *.pre is the normal input file for PREFEKO, in which the frequency *has* to be expressed i.t.o. the variable #freq. While the program TIMEFEKO is running, the frequency is constantly changed. TIMEFEKO generates new *.pre files in which the actual numerical value of the frequency is assigned. The user may use the DEFINED function to assign a numerical value to the frequency variable #freq in the *.pre file — this is useful for running PREFEKO and viewing the geometry in POSTFEKO (see the TIMEFEKO example in the *Examples Guide*).

The FR card *must* use the frequency variable #freq. The number of frequencies to be examined *must* be one (NFREQ=1).

Example:

```
** Set the frequency
FR      1                      #freq
```

In the input file (*.pre), all the desired output parameters in their respective cards must be set (for example, FF card, FE card, OS card, ...). Only the data in the output file (*.out) can be transformed.

It is also possible to vary the segmentation for certain applications (far field, near field) to save memory and computation time. To do this one may use the variable #freq (which is constantly changed by TIMEFEKO) in the definition of the segmentation parameters.

Example:

```
** Define some constants
#maxfreq = 250.0e+08
#minlambda = #c0/#maxfreq

** Define the edge length - note the use of #freq
#edgelen = (2.0-#freq/#maxfreq)*#minlamdba/4.0

** Set the segmentation parameters
IP                      #edgelen
```

¹¹The files have the same file name, but with *.pre and *.tim extensions.

22.3 The *.tim input file

In the file *.tim the pulse form, position and the point in time are assigned. This file consists of a number of sections which are optional at present.

- Set the pulse form with characteristic value and the time shift.
- Set the value of the highest frequency and the sampling points.
- Set the normalisation (normalise time to the speed of light).
- Set whether the output is written to the output file.
- Set the time points.

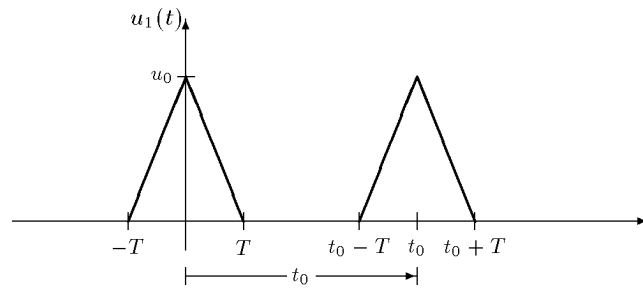
In the *.tim file, empty or comment lines (starting with **) are allowed. The parameters need not be entered in any particular column, but they have to be in the correct order.

The keywords used in the *.tim file exist in German and English (for example ANREGUNG and EXCITATION). Both versions of each keyword will be given in the discussion below.

22.3.1 Defining the pulse form

The assignment of the pulse form is necessary. Each pulse has a predefined name and has particular characteristic parameters. The parameters must be assigned absolute values and not normalised values. The amplitude factor u_0 is 1. It can be changed by using the AX card (in the file *.pre) with the appropriate amplitude value.

Time shifting t_0 : indicates the time (in seconds) that the pulses are delayed (see the shift in Figure 22-1). The time shift should be such that the excitation of the structure at $t = 0$ is approximately 0. Since both the time and frequency domain data are continuous this is not strictly required, but it simplifies working with the results.

Figure 22-1: Time function $u_1(t)$ shifted by t_0

The following pulses are available:

1. Gaussian pulse GAUSS / GAUSS (see Figure 22-2)

$$u_1(t) = u_0 e^{-a^2(t-t_0)^2} \quad (22-1)$$

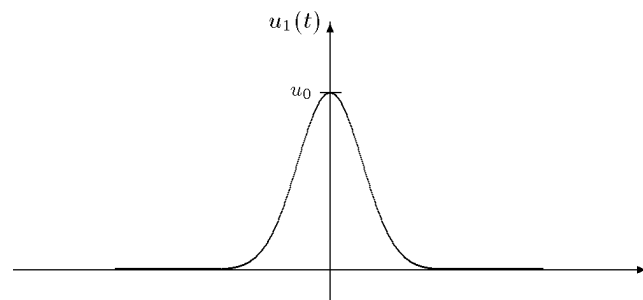


Figure 22-2: Gaussian pulse

Example:

```
** Pulse form
GAUSS
**      t0      Exponent a
      2.0E-08    3.0E+08
```

2. Triangular pulse DREIECK / TRIANGLE (see Figure 22-3)

$$u_1(t) = \begin{cases} u_0 \left(1 - \frac{|t - t_0|}{T} \right) & \text{for } |t - t_0| \leq T \\ 0 & \text{for otherwise} \end{cases} \quad (22-2)$$

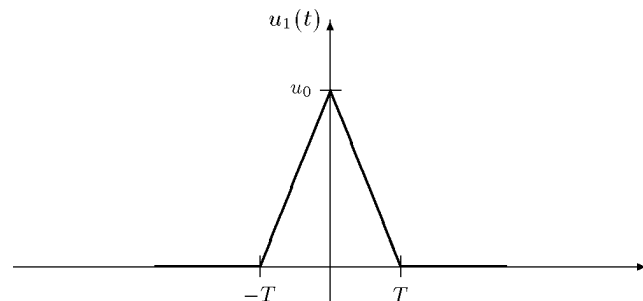


Figure 22-3: Triangular pulse

Example:

```
** Pulse form
TRIANGLE
**      t0      Impulse Duration T
      2.0E-08    1.0E-08
```

3. Double exponential pulse DEXP / DEXP (see Figure 22-4)

$$u_1(t) = \begin{cases} 0 & \text{for } t \leq t_0 \\ u_1 \left(1 - e^{-\frac{t-t_0}{\tau_1}} \right) & \text{for } t_0 \leq t \leq T + t_0 \\ u_2 e^{-\frac{t-t_0}{\tau_2}} & \text{for } t \geq T + t_0 \end{cases} \quad (22-3)$$

$$u_1 = \frac{u_0}{1 - e^{-\frac{T-t_0}{\tau_1}}} \quad u_2 = \frac{u_0}{e^{-\frac{T-t_0}{\tau_2}}} \quad (22-4)$$

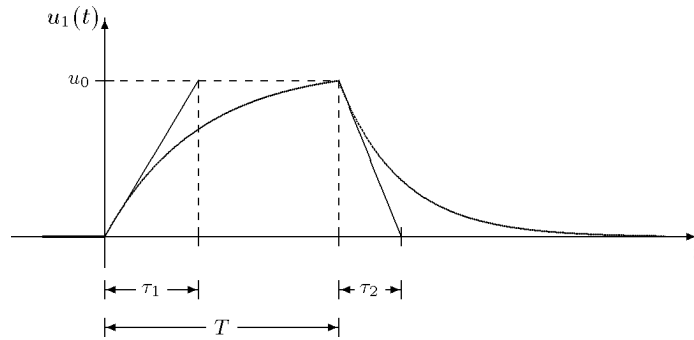


Figure 22-4: Double exponential pulse

Example:

```

** Pulse form
DEXP
**   t0      T      tau1      tau2
    0.0    10.0E-09  5.0E-9   10.0E-9

```

4. Ramp pulse RAMPE / RAMP (see Figure 22-5)

$$u_1(t) = \begin{cases} 0 & \text{for } t \leq t_1 + t_0 \\ u_0 \left(1 - \frac{|t - t_2 - t_0|}{\tau_1} \right) & \text{for } t_1 + t_0 \leq t \leq t_2 + t_0 \\ u_0 & \text{for } t_2 + t_0 \leq t \leq t_3 + t_0 \\ u_0 \left(1 - \frac{|t - t_3 - t_0|}{\tau_2} \right) & \text{for } t_3 + t_0 \leq t \leq t_4 + t_0 \\ 0 & \text{for } t \geq t_4 \end{cases} \quad (22-5)$$

$$t_1 = -\frac{T + \tau_1}{2} \quad t_2 = -\frac{T - \tau_1}{2} \quad t_3 = \frac{T - \tau_2}{2} \quad t_4 = \frac{T + \tau_2}{2} \quad (22-6)$$

Example:

```

** Pulse form
RAMP
**   t0      Impulse Duration T      tau1      tau2
    20.0E-09  15.0E-09  5.0E-09  10.0E-09

```

5. Double exponential impulse (second type) DBLEXP / DBLEXP (see Figure 22-6)

$$u_1(t) = \begin{cases} 0 & \text{for } t \leq t_0 \\ u_0 \left(e^{-\frac{t-t_0}{\tau_1}} - e^{-\frac{t-t_0}{\tau_2}} \right) & \text{for } t > t_0 \end{cases} \quad (22-7)$$

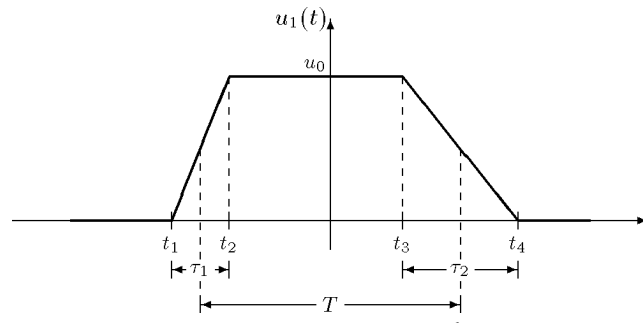


Figure 22-5: Ramp pulse

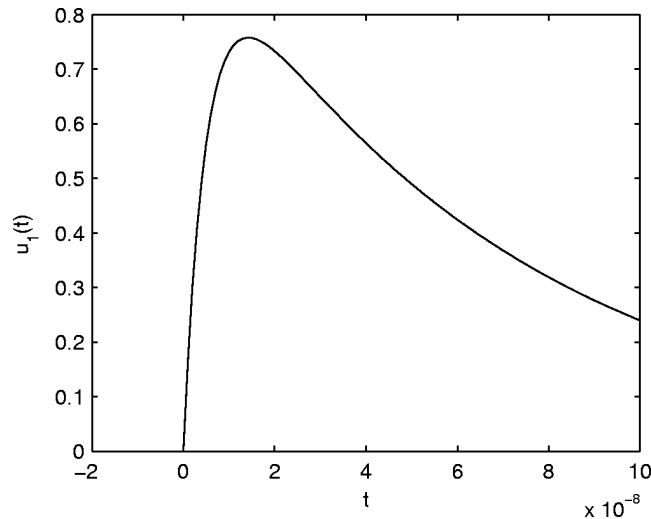


Figure 22-6: Double exponential impulse (second type)

Example:

```

** Impulse form
DBLEXP
** Time. t0      Parameter tau1  Parameter tau2
    20.0E-9      70.0E-9        5.0E-9

```

22.3.2 Definition of the frequency block

The upper frequency limit f_{max} and the number of frequency points N are defined in the frequency block FREQUENZ / FREQUENCY.

Example:

```

FREQUENCY
** Upper frequency limit      Number of frequency points
    250.0E+06                  34

```

The maximum frequency f_{max} should be large enough such that the whole spectrum of the exciting pulse is covered. For example, for the Gaussian pulse — GAUSS (see section 22.3.1), we find

$$f_{3dB} = \frac{a}{\pi} \sqrt{\ln \sqrt{2}} \approx 0.187 a \quad (22-8)$$

with a as defined in Equation 22-1. One should select approximately $f_{max} = 4 f_{3dB}$.

The number of frequency points N is then selected so that the total time T is long enough for the exciting pulse and all included currents, radiated fields, etc. to have decayed. Once the total time has been determined, the number of samples may be determined from the relation

$$\Delta f = \frac{1}{T} = \frac{f_{\max}}{(N-1)} \quad (22-9)$$

or

$$N = 1 + T f_{\max} \quad (22-10)$$

Finally, let P be the smallest power of 2 which is larger than or equal to $N - 1$. (For example: for $N=10$, $P=16$; for $N=33$, $P=64$; and for $N=50$, $P=64$.) Then the time stepping will be $\Delta t = \frac{T}{2P} = \frac{N-1}{2f_{\max}P}$. With the relation $\frac{P}{2} < N - 1 \leq P$ we get the bounds

$$\frac{1}{4f_{\max}} < \Delta t \leq \frac{1}{2f_{\max}} \quad (22-11)$$

22.3.3 Definitions of the normalisation

Using the keyword `NORM / NORM`, the time can be normalised to the speed of light in a vacuum c_0 . The normalised time then has a unit of `lm` (*light-metre*).

$$t_{\text{norm}} = t \cdot c_0 \quad (22-12)$$

Example:

```
** Normalising time with respect to the speed of light
NORM
```

22.3.4 Definition of the excitation output

The keyword `ANREGUNG / EXCITATION` indicates whether the time variation of the exciting pulse is to appear in the output file.

Example:

```
** Output the excitation pulse
EXCITATION
```

22.3.5 Definition of a time point

With the keyword `ZEITPUNKTE / POINTS_IN_TIME`, the near fields, surface currents and line currents can be calculated at certain points in time. Using time points has no effect on the calculation of source currents or far fields. If normalisation has been set (`NORM`), then the time points must also be given in normalised form.

Example:

```
** Specifying the time points -- time has been normalised
POINTS_IN_TIME
5.0 10.0 15.0 20.0
```

22.4 Running TIMEFEKO

Firstly, the input files `*.pre` and `*.tim` have to be created. During execution new input files (`*.pre`) are continuously generated by adding the string `_tim_` and a sequentially incremented number to the file name. If, for example, the input files `cube.pre` and `cube.tim` have been created, TIMEFEKO is executed with the command

```
timefeko cube
```

On the command line the following parameters may be added:

- `-a` Here an FFT is only performed on the already available data.
- `-r` The `*.pre` and `*.out` files are deleted after each iteration. This saves disk space.
- `--runfeko-options` After this option one can specify additional options which will be used in the launcher RUNFEKO for the FEKO kernel. For instance in order to use the parallel FEKO solver during the time domain analysis one can use the command

```
timefeko file --runfeko-options -np 2
```

or also

```
timefeko file --runfeko-options -np 2 --machines-file m
```

For a remote execution of the FEKO runs during the analysis on another host, the suitable command would be

```
timefeko file --remote-host hostname
```

See section [18.2](#) for a list of all such RUNFEKO options.

- `--version` Output only the version information to the command line and terminate.

It should be noted that when using the remote launching facility (see section [18.2.3](#)), the actual TIMEFEKO process is executed on the local machine, only the FEKO kernel runs (which are the time and memory consuming part) are done on the remote machine.

22.5 TIMEFEKO output

TIMEFEKO generates different `*.pre` files at the different frequencies. The `*.out` files for all these runs are available if TIMEFEKO was called without the `-r` option. These results are summarised — if they are requested in the `*.pre` file — in `*.ost` (triangle currents), `*.oss` (segment currents), `*.cur` (the currents on voltage sources), `*.far` (far fields), `*.nfe` (near electric fields) and `*.nfh` (near magnetic fields). In each case the results for each frequency is listed one after the other with frequencies separated by a line containing a single `#`-character.

The time domain results are given in the `*.aus` file for all outputs requested in the `*.pre` file. Currents are requested by the OS card. Note that requesting the current on a large number of structures will result in very large `*.aus` files.

23 The FEKO software updater

23.1 Introduction

With the FEKO GUI update utility or the command-line FEKO_UPDATE utility the latest FEKO updates can automatically be downloaded and installed. Users are given the option to download the latest FEKO update from a master or a local repository. The master repository is the FEKO website and requires internet access. The command-line updating utility enables scripted updates whilst the GUI version of the updater is interactive and allows the user to set preferences as well as checking for new updates.

Note that no updates will be made available for FEKO Lite installations. Other installations will require a valid M&S licence to download updates. FEKO Suite updates will only apply to the current FEKO suite installation.

No information is sent to the FEKO website during an update, a list containing the latest software is retrieved and compared to the currently installed components.

23.2 GUI update utility

The FEKO software updating utility is automatically launched every time one of the GUI components is launched, but it only checks for updates if the check interval time has elapsed (the default check interval is one week). When updates are available the user will be notified and also has an option to download and install the updates. It is also possible to force a *Check for update* from within CADFEKO, EDITFEKO and POSTFEKO. Select *Help* → *Check for updates*.

On the *Info* tab, information regarding the available updates will be displayed. Settings regarding the update schedule, update source and proxy settings are given on the *Settings* tab as shown in Figure 23-1.

The name of the GUI update utility is `feko_updchecker_gui` and allows the following options to be set as well as checking for updates and performing updates.

Schedule: The user may set the intervals for when the FEKO updater should check for updates. The option to automatically check for updates can also be disabled by unchecking the checkbox *Check for updates automatically* on the settings tab of the *FEKO update* dialog.

Update from: The web or a local repository may be set as the desired location from where the latest updates are to be downloaded from. The user has the option to download updates via the web or from a local repository. Using the local repository is recommended when the computer network or cluster does not have internet access due to security reasons or only limited bandwidth is available. In this case, the updates may be downloaded from the FEKO website by the system administrator and placed at a location accessible for the computer network or cluster. The *Local repository* can be set on the *Settings* tab of the *FEKO update* dialog. Select *Local repository* and enter the location of the updates.

When the *Web* option is selected, the updates will be downloaded from the FEKO website.

Proxy: If the web is used as the repository, the user can set the proxy server and authentication (if required).

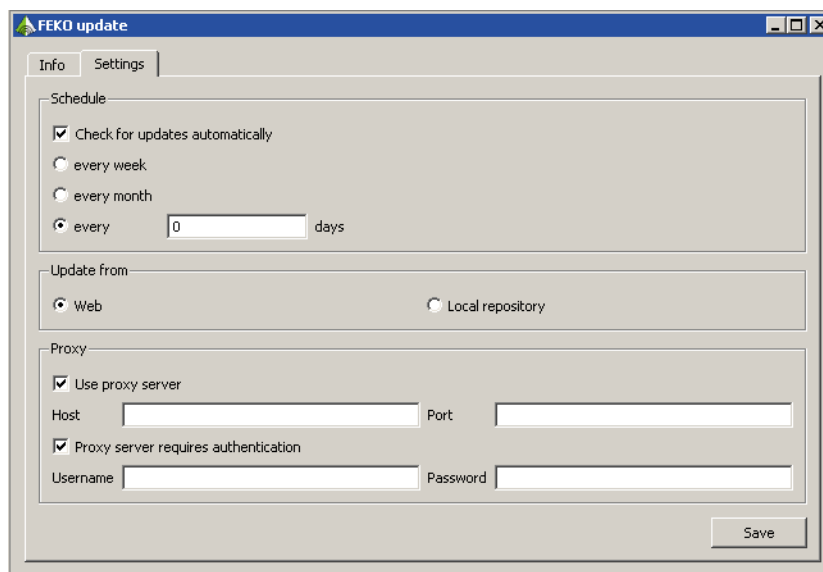


Figure 23-1: The *FEKO update* dialog with the *Settings* tab selected

23.3 Command-line update utility

FEKO_UPDATE is a command-line application that enables scripted updates. Launching the application without any arguments will list the possible argument options. In most cases `feko_update` will be launched as shown below to check for updates and then download and install any available updates.

```
feko_update --update
```

The optional arguments for `feko_update` are:

- `--check` [<host>[:port] [user [password]]] Check if updates are available from the master repository. 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:
- The date of the available update.
 - A list of the updated FEKO Suite components with their version numbers.
 - The total download size of the update.
 - The release notes for all updates for the Suite version.

If no updates are available, the user is informed and the application exits.

- `--update` [<host>[:port] [user [password]]] Check if updates are available from the master repository, then download and install all available updates. If the computer is behind a proxy server, the proxy server address and the login details can be optionally supplied. If updates are available, the following will occur:

- Print the same output as that of the `--check` option.
- Print each file which is being downloaded.
- Print each file which is being updated.
- Print a message stating that the update was successful and exit.

If no updates are available, the user is informed and the application exits.

- `--check-from <path>` Similar to the `--check` option, the updater checks if updates are available, but the update source is not the FEKO website. The update source is the local repository specified by `<path>`.
- `--update-from <path>` Similar to the `--update` option, the updater checks if updates are available and install the updates, but the update source is not the FEKO website. The update source is the local repository specified by `<path>`. The path must be an absolute file path which can point to a unmapped network share (Windows), mapped (mounted) network share or a directory on a local drive.
- `--version` Print the version information and then exit.

23.4 Proxy settings - advanced

The FEKO updater (GUI and command line versions) will use the system proxy by default. On Windows the proxy is the same as is used by Internet Explorer. The proxy can be specified manually or by using PAC file. On Linux the system proxy is defined by the environment variable `http_proxy`. If the the environment variable `http_proxy` is not defined, then no proxy will be used.

The parameter `--no-proxy` was added to bypass the system settings and use a direct connection.

23.5 Creating a local update repository

The following describes the steps required for the system administrator to create a local FEKO update repository. Note that the path for the local FEKO update repository must be an absolute file path which can point to a unmapped network share (Windows), mapped (mounted) network share or a directory on a local drive.

- If previous updates are still located at the required destination, first delete the old updates.
- Download the updates for the required platforms.
- Unzip the folders in the required destination.

If multiple platforms are downloaded, ensure that the various platform updates are located at the same destination. The zip file contains an entire directory structure which must be kept intact. When updates for multiple platforms are required, the directories must be merged.

If for example WIN64_AMD64 and WIN64_EM64T updates were downloaded to `C:\Updates`, unzipped and merged, the following directory tree will result:

```
C:\
|--Updates
    |--FEKO_EMSS
        |--SuiteXX.YY
            |--Win64_AMD64
            |--Win64_EM64T
```

In the above case set the local repository path to C:\Updates (or a network reachable equivalent UNC path or mapped drive). Always point the local repository path to the directory that contains the root item of the *.zip file.

Part IX

Appendix

24 Advanced modelling and solution control

24.1 Utilisation of symmetry

It is possible to reduce the calculation time and memory usage if symmetry is utilised. This can be done by using the SY card (see section 13.43), or by setting symmetry for the model in CADFEKO (see section 6.8).

Three coordinate planes, $x = 0$ (yz plane), $y = 0$ (xz plane) and $z = 0$ (xy plane) may be defined as planes of symmetry. There are three different types of symmetry. They are described below.

24.1.1 Geometric symmetry

With this type of symmetry the geometry of the modelled solid or part of the solid is symmetric about one or more coordinate planes. *The interaction between any two basis functions must be the same as that between their symmetrical counterparts. Everything which affects this must be symmetrical, i.e. loading, losses, Green's functions, etc.* The source, however, is not symmetric, thus a symmetric current distribution does not exist. This asymmetric current distribution leads to asymmetric electric and magnetic fields.

The body of a truck with an antenna placed at the front left hand side, will be used as an example. In the input file, half of the body is constructed (either the left or right side). The other half is then created with the SY command. Finally the antenna is placed in the correct position on one side.

A rectangular metallic plate, illuminated by an electromagnetic wave from a direction outside the principle planes, is another example. In this case a quarter of the plate is constructed and the rest is created using the SY card (see section 13.43) with geometric mirroring around two coordinate planes.

When defining symmetry in CADFEKO, symmetrical geometry must be created and the relevant symmetry set before meshing. In the case where a model is to be considered that is not entirely symmetrical (as in the first example above), the asymmetrical part/s of the model must be added by editing the pre file and cannot be defined in CADFEKO.

Geometric symmetry does not reduce the number of unknown coefficients in the current basis functions. Therefore there is no reduction in memory usage. There is, however, a reduction in computation time when the matrix elements are determined.

24.1.2 Electric symmetry

An electric symmetry plane is a plane which can be replaced by an ideal electrically conducting wall without changing the field distribution.

In Figure 24-1 an electric symmetry plane is displayed. The electric $\vec{E}$ field's tangential component disappears and the magnetic $\vec{H}$ field's normal component disappears. The electric current density $\vec{J}$ is anti-symmetric and the magnetic current density $\vec{M}$ is symmetric.

As with geometric symmetry, less computational time is required to calculate the matrix elements. The number of unknown coefficients of the current basis functions are also reduced, leading to linear equation system of a lower order. This leads to a further reduction in the computation time and requires less memory due to the reduction in the number of matrix elements.

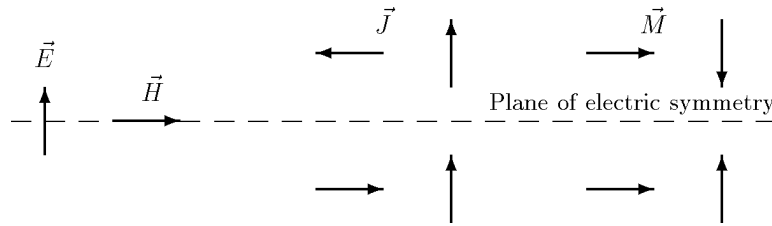


Figure 24-1: Electric symmetry plane

24.1.3 Magnetic symmetry

A magnetic symmetry plane is a plane which can be replaced by an ideal magnetically conducting wall without changing the field distribution.

In Figure 24-2 a magnetic symmetry plane is displayed.

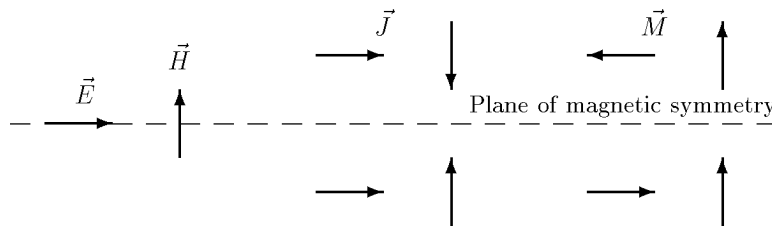


Figure 24-2: Magnetic symmetry plane

The electric $\vec{E}$ field's normal component and the magnetic $\vec{H}$ field's tangential disappears. The electric current density $\vec{J}$ is symmetric and the magnetic current density $\vec{M}$ is asymmetric.

When using magnetic symmetry there is a reduction in the computational time when determining the matrix elements. The order of the matrix equation is also reduced, which leads to a further reduction in the computational time and reduces the amount of memory needed to determine the matrix elements.

24.1.4 Example of the application of symmetry

In Figure 24-3 a dielectric sphere is shown with a linear polarised incident electromagnetic field. The full description of the problem is given in `example_04` in the *Script Examples*. Only the use of symmetry is described here.

The plane $z = 0$ (xy plane) is a plane of geometric symmetry, because the excitation does not have any symmetry in this plane. The plane $x = 0$ (yz plane) is a plane of electric symmetry, because the electric field is perpendicular to this plane and the magnetic field has a tangential component only. Similarly the electric field only has a tangential component in the $y = 0$ plane

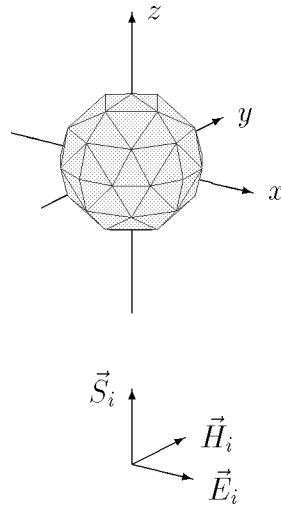


Figure 24-3: Dielectric sphere with incident field

Table 24-1: A summary of the Simplex operations

<i>Symmetry type</i>	<i>Number of unknowns</i>	<i>Memory usage</i>	<i>Solution time [s]</i>
Symmetry not used	1032	8.38 MByte	3.860
All 3 coordinate planes declared as planes of geometric symmetry	1032	8.41 MByte	2.015
Plane $x = 0$ declared as electric plane of symmetry, plane $y = 0$ as magnetic plane of symmetry and plane $z = 0$ as geometric plane of symmetry	258	2.36 MByte	1.094

and the magnetic field is perpendicular. This is thus a plane of magnetic symmetry. The reduction in time and resources is tabulated below ¹².

This example has relatively few unknowns. Most of the computational time is therefore used to determine the matrix elements in comparison to the time taken to solve the matrix equation. For applications with more unknowns, the reduction of unknowns could make a considerable difference in the time and memory required.

24.1.5 Special enforcement of symmetry: Even – odd method

The table in the previous section demonstrates the advantage of using electric and/or magnetic symmetry. For very large structures which have only geometrical symmetry, it may be worthwhile to consider two separate problems with electrical and magnetic symmetry as described below.

Figure 24-4 (a) shows the original problem. It consists of a dipole antenna with a passive wire below it. This is admittedly a very simple problem, normally this procedure would only be applied to much more complex structures.

The structure in Figure 24-4 is symmetric about the plane $z = 0$, but the excitation is asymmetric and thus only geometric symmetry can be applied in FEKO. This problem may be separated into

¹²The runtimes in this table are dependant on the platform on which the problem is run. These numbers are therefore only indicative of the relative effects on overall runtime.

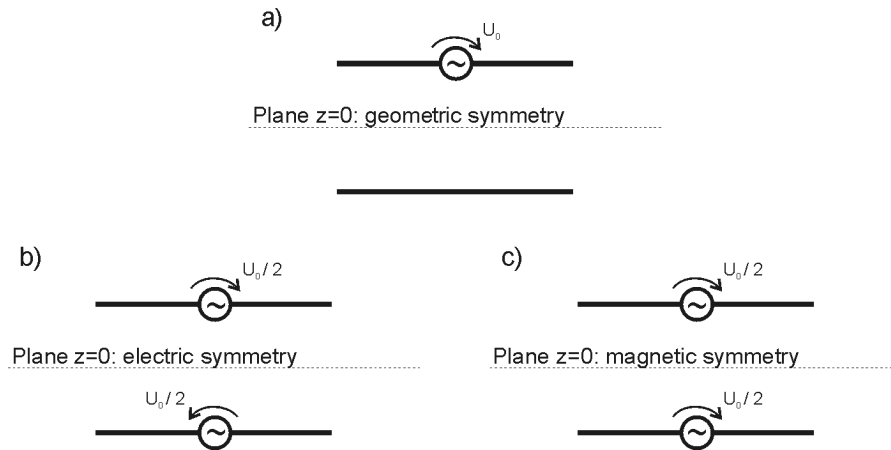


Figure 24-4: Separating a problem with geometrical symmetry into two problems with electric and magnetic symmetry respectively.

the two sub-problems shown in Figures 24-4 (b) and (c). These two problems can be solved using respectively electric and magnetic symmetry about the plane $z = 0$. Each of these problems require only half the number of unknowns required for case (a). Superposition of the problems (b) and (c) (this must unfortunately be done externally to FEKO, as such superposition is not currently catered for) yields a result to the original problem.

The solution of each of the sub-problems requires only half of the storage space required for case (a). For very large problems the solution time is dominated by the time required to solve the system of linear equations. In this case the two sub-problems each requires only one eighth of the time required for direct solution of case (a).

24.2 Dynamic memory management

24.2.1 Telling FEKO how much memory can be used

FEKO has the ability to manage the memory dynamically, i.e. the memory required for the geometry data and matrix elements etc. is determined and allocated at run time. When FEKO tries to allocate memory, in principle the operating system offers a certain address space, which might either be physically installed memory (i.e. RAM), but also virtual memory (system swap space swapped to the hard disk).

If FEKO starts to swap using virtual memory, then the whole solution process can be slowed down quite significantly, and this is not recommended. 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 which do not fit into the available RAM, but do fit into the RAM plus virtual memory, the user should tell FEKO what the actual physically installed memory is, less a certain reserve for the operating system.

Up to FEKO Suite 4.2 the variables `#maxalloc` or `#maxallocm` were used for this purpose:

`#maxallocm` This sets the maximum allocatable memory in MBytes that can be used 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. If this is not enough, the matrix will be saved to the hard disk or the program will be halted. For parallel versions of FEKO the memory limit for each process is determined by dividing the value of `#maxallocm` by the number of parallel processes on the same host. Note that if `#maxallocm` is used, it will have preference, and `#maxalloc` will be ignored if it is used in the same `*.pre` file.

`#maxalloc` Similar to `#maxallocm` but just specifies the memory in Bytes and not in MBytes.

These two variables `#maxalloc` and `#maxallocm` are still supported in FEKO Suite 4.2 and later for backwards compatibility reasons, however, their usage is strongly discouraged when using newer FKO versions.

In newer FEKO versions (Windows and Linux) the FEKO memory management is fully automatic and the usage of virtual memory (swap space) is avoided as far as possible. On some UNIX versions (such as HP-UX or SUN Solaris), an environment variable `FEKO_MAXALLOCM` is defined during the installation and set in the `initfeko` initialisation script (this specifies the physically installed RAM less an operating system reserve for a specific machine). The advantage then is that the FEKO `*.pre` files are machine independent, i.e. if you have two different computers with different memory, no variable `#maxallocm` has to be changed in the `*.pre` file as such, everything is either fully automatic (Windows or Linux) or defined on a per machine basis (other UNIX versions).

24.2.2 Variables that are automatically set correctly

Note that normally PREFEKO estimates the following variables correctly and they should only be declared in cases where there is an explicit error message stating that larger memory blocks are required. Under normal circumstances these variables should not be set, as they could have a negative impact on FEKO performance.

<code>#maxaeedges</code>	The maximum number of edges between triangles that may be excited with the AE card.
<code>#maxanr</code>	The maximum number of sources.
<code>#maxapo</code>	The size of the memory block that is used to save the coefficients in the physical optics approximation. For <code>#maxapo=0</code> the necessary amount will be dynamically allocated.
<code>#maxarang</code>	The maximum number of ϑ or φ angles used with the AR card (excitation by a point source with a specified radiation pattern).
<code>#maxarpat</code>	The maximum number of radiation pattern excitations (AR card) allowed simultaneously.
<code>#maxbsobnr</code>	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 <code>#maxbsobnr</code> is used in this case.
<code>#maxcolayer</code>	The maximum number of layers on a CO card which implements thin dielectric sheets.
<code>#maxdrnv</code>	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 <i>Specified points</i> .
#maxfoge	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.
#maxknor	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 (i.e. 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 moment method 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 so that 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).

24.3 Environment variables and registry keys

This section lists the environment variables that may be used to control the execution of FEKO. From Suite 5.5 the default location for these settings under Windows are in the registry and not as environment variables. The preferred way of changing any of these settings under Windows is by using `initfeko.bat`. The syntax for `initfeko.bat` is as follows:

-setreg	Write the environment values (as set in <code>initfeko.bat</code>) to the registry. Default values are defined in <code>initfeko.bt</code> directly by the initial installer and can be modified to match the users' needs. Note that the current environment/shell will not be changed.
-queryreg	Query the registry and set the values to the current environment if they are not already set.
-console	Opens a FEKO Console Window for running commands directly in the console. All required environment settings will be loaded (see above option "-queryreg") and applied to the current shell.
-d	Shows additional debug output while setting the environment.

These variables are also discussed in the *FEKO Installation Guide*, with respect to the script `initfeko` or the batch file `initfeko.bat` automatically created by the FEKO installation program. During the installation process, the environment variables are set correctly and the user does not generally need to set environment variables manually.

The environment variables can also be specified with registry keys that can be found in `HKLM\SOFTWARE\FEKO\major.minor\Environment (global)`

HKCU\SOFTWARE\FEKO\major.minor\Environment (current user)

where major.minor is the FEKO Suite (e.g. 5.5). The variables set in the environment have precedence over variables in the registry. Also note that the ones in HKCU have precedence over ones in the HKLM. Please note that the preferred way to change values in the registry is to use `initfeko.bat`. Changing the environment variables directly is strongly discouraged.

The following environment variables may be set:

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 some implementations of `mpirun`, `mpiiopt`, `mpprun` etc.

FEKO_HOME This variable is set to the FEKO installation path which contains the subdirectories such as `bin`, `doc`, `license` and, for the parallel version, `mpi`.
Note that it is not recommended to modify this environment variable.

FEKO_LICENSE_FILE 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` (see section 15.1).

FEKO_LITE If this environment variable is set to 1, then FEKO runs for 30 days in the unregistered FEKO LITE mode. FEKO is then restricted regarding the element sizes etc., but does not need a licence (see the *FEKO Installation Guide* for more details).

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. (More detail can be found in section 18.2.)

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 this computer. This environment variable is not needed or recommended for computers running Windows or Linux operating systems. On others the variable is set at installation time. The value of this variable should usually be set equal to the physical memory minus 70 MBytes for the operating system. In a few cases a lower limit may be required, and should be set here.

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 communication 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_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 etc. 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 (checks on the remote nodes, remote copying of files, remote execution of utilities etc.), 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 (remsh under HP-UX and rsh for all other UNIX platforms). A typical setup is then to use a `~/.rhosts` file (see detailed comments given during the installation). But this is not quite secure, and one 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 (e.g. rsh or remsh or ssh will be determined during the installation procedure, and the environment variable FEKO_RSH will be set to point to this executable. This can always be changed later (e.g. using rsh for the installation as root, but then ssh for the users using the parallel FEKO version or vice versa) by changing the value of the environment variable FEKO_RSH in the FEKO initialisation file `initfeko` or `initfeko.csh` accordingly. One can also set this on a user per user basis, then directly for instance in `~/.profile` after having executed `initfeko`. This environment variable will not be used on 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 (Windows) or a slash (UNIX). This is no longer required. For example, in UNIX it may be set with

```
set FEKO_TMPDIR=/tmp
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 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 suite 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 by the installation script) to one of the following options:

- 1: MPICH2 (all supported UNIX platforms, currently Linux on Intel, Alpha and SUN.)
- 2: HP-MPI for HP-UX
- 3: NEC MPI on NEC SX/5
- 4: SGI MPI on SGI¹³
- 5: CRAY/SGI MPI on CRAY T3E

¹³Support for SGI MIPS IRIX discontinued from Suite 5.4

- 6: ScaMPI (Linux on Intel/AMD and Alpha with SCI/Myrinet/Infiniband interfaces)
- 7: Compaq MPI
- 8: SCore (Linux on Intel with Myrinet interfaces)
- 9: GM (Linux on Intel with Myrinet interfaces)
- 10: ParaStation MPI
- 11: Intel MPI (Linux on Intel IA32, IA64, EM64T and on AMD64)
- 12: IBM Parallel Environment (PE)
- 13: MS MPI (Microsoft MPI)

FEKO_WHICH_SPICE_EXECUTABLE This variable specifies which SPICE solver is to be used. When left unset (default), FEKO will use the NGSPICE executable (included in FEKO installation). A user can set this variable to point to the full path and executable name of the preferred SPICE solver.

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.

MKL_SERIAL If this is set to YES, FEKO (and all other codes using the Intel MKL libraries) will run as a single threaded application (i.e. it will not utilise multiple CPUs) irrespective of the value of OMP_NUM_THREADS (see below).

OMP_NUM_THREADS Sequential FEKO versions for Windows or Linux PCs (Intel or compatible) can use multiple CPUs on one board for the LU-decomposition of the MoM matrix (the parallel FEKO version will have all phases of the solution in parallel). To use this, the environment variable OMP_NUM_THREADS must be set to the number of CPUs to use. (See also MKL_SERIAL above.)

25 Summary of files

The table below gives an overview of the different files and their respective functions. (STDOUT is the standard output - usually the screen; a *-symbol is used to symbolically indicate the file name.)

STDOUT	Usually the screen. This is where comments such as progress, warnings and errors are sent.
*._14	Page (temporary storage) file for the matrix in the sequential version of FEKO with out-of-core solution.
*._15	Page (temporary storage) file for the matrix in the sequential version of FEKO with out-of-core solution.
*._16	Page (temporary storage) file for the matrix in the sequential version of FEKO with out-of-core solution.
*._20	Page (temporary storage) file for the calculated coupling coefficients of the MoM/PO hybrid method during an out-of-core solution.
*._21	Page (temporary storage) file for the calculated coupling coefficients of the MoM/PO hybrid method during an out-of-core solution.
*.afo	Continuous frequency results created by ADAPTFEKO.
*.aus	Output file of TIMEFEKO.
*.bof	Binary version of the output file which is used for post processing.
*.cdb	ANSYS mesh file which can be imported with the IN card.
*.cfm	CADFEKO mesh file (exported file containing CADFEKO mesh and variables to be imported by PREFEKO).
*.cfs	Session file of CADFEKO.
*.cfx	Native CADFEKO model file (contains geometry, mesh, solution settings, optimisation settings etc.)
*.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.
*.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 (see section 14.31).
*.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.
*.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 (see section 13.22).)

*.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 (see section 14.31).
*.fek	Output file from PREFEKO — serves as the input file for FEKO.
*.fim	FEST3D file containing the waveguide modes which may be imported using the AW card.
*.ffe	File containing the far field data. This file is only generated on request by a DA card (see section 14.31).
*.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 (see section 14.31).
*.inc	Include file for PREFEKO.
*.inp	ABAQUS mesh file.
*.isd	Data file containing the field distribution calculated by FEKO for coupling with Cable-Mod, CRIPTE or PCBMod.
*.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 (see section 14.54).
*.mat	File in which the matrix elements of the linear equation system, are stored (only generated on request of a PS card (see section 14.54).
*.nas	NASTRAN geometry file that may be imported using the IN card.
*.neu	Geometrical data file which is exported by the program FEMAP
*.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 (see section 14.31).
*.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 (see section 14.31).

*.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.
*.ray	When using UTD or 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.
*.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).
*.sha	File storing shadowing information for the PO.
*.snp	Touchstone format (v1.0) S-parameter file as created by the DA card (see section 14.31). 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 (see section 14.31) or used to define a spherical mode excitation in a FEKO solution as part of an AS card).
*.str	File in which the coefficients of the basis functions are stored for reuse (generated on request from a PS card (see section 14.54).
*.tim	Input file for the program TIMEFEKO.
*.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.
*.wfg	Figure created and saved with GraphFEKO.
*.x_t/*x_b	Parasolid model file.

The files *.efe, *.hfe, *.ffe, *.os, and *.snp are redundant. All the information in these files is also available in human-readable format in the *.out file (if not explicitly switched off at the DA card). The format of these redundant files may lend itself more readily to further processing outside of POSTFEKO.

26 SPICE3f5 general structure, conventions and syntax

A SPICE circuit to be analysed is described by a set of element lines, which define the circuit topology and element values, and a set of control lines, which define the model parameters and the run controls.

Since the FEKO user is only presenting the circuit description (subset of a standard SPICE circuit), no run controls may be specified. The first line in the input file must be the title and the last line must be '.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 shown below:

```
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
```

Extract from the Berkeley SPICE manual regarding SPICE syntax will be presented in the sections that follow. 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.

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

General form:

```
.SUBCKT SUBNAM N1 <N2 N3...>
```

A circuit definition begins with a '.SUBCKT' line. 'SUBNAM' is the subcircuit name (needs to be the same name as the general network) and 'N1, N2...' are the external nodes, which cannot be zero. The group of element lines which immediately follows the '.SUBCKT' line define 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 (i.e. such models and definitions are not known outside the subcircuit definition). Also, any element nodes not included on the '.SUBCKT' line are strictly local, with the exception of 0 (ground) which is always global.

.ENDS line

General form:

```
.ENDS <SUBNAM>
```

The '.ENDS' line must be the last one for any subcircuit definition. 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 nested subcircuit definitions are being made.

.INCLUDE lines

General form:

```
.INCLUDE filename
```

Portions of circuit descriptions will often be reused in several input files, particularly with common models and subcircuits. In any SPICE input file, the '.INCLUDE' line may be used to copy some other file as if that second file appeared instead of the '.INCLUDE' line in the original file. There is no restriction on the file name imposed by SPICE beyond those imposed by the local operating system.

Resistors

General form:

```
RXXXXXXX N1 N2 VALUE
```

'N1' and 'N2' are the two element nodes. 'VALUE' is the resistance (in ohms) and may be positive or negative but not zero.

Capacitors

General form:

```
CXXXXXXX N+ N- VALUE
```

'N+' and 'N-' are the positive and negative element nodes, respectively. 'VALUE' is the capacitance in Farads.

Inductors

General form:

```
LXXXXXX N+ N- VALUE
```

‘N+’ and ‘N-’ are the positive and negative element nodes, respectively. VALUE is the inductance in Henries.

Coupled (Mutual) Inductors

General form:

```
KXXXXXX LYYYYYY LZZZZZZ VALUE
```

‘LYYYYYYY’ and ‘LZZZZZZ’ are the names of the two coupled inductors. ‘VALUE’ is the coefficient of coupling, K, which must be greater than zero and less than or equal to one.

Lossless transmission lines

General form:

```
TXXXXXX N1 N2 N3 N4 Z0=VALUE <TD=VALUE> <F=FREQ <NL=NRMLEN>
```

‘N1’ and ‘N2’ are the nodes at port one; ‘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 (as TD=10 ns, for example). 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, 0.25 is assumed (i.e. 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.

27 CADFEKO geometry faults

A list of possible geometry faults reported by CADFEKO is listed in the section that follows. A short description with a possible solution is also given for each fault.

Corrupt data structures: The solid modeller is in an inconsistent state. If this model was imported, either the translation failed or the original model contained errors. Please send the original model file (in the original format) to the FEKO support team.

Invalid or duplicate identifiers: This situation usually arises when old Parasolid models are imported.

Missing geometry: This situation only arises with imported models. The best solution is to select the faulty entity in the details tree and remove it. Note that this entity will not be visible in the 3D view. The geometry has to be recreated.

Invalid geometry: The fault is usually caused by scaled importing. The best solution is to select the faulty entity in the details tree and remove it. Note that this entity will not be visible in the 3D view.

Self-intersecting geometry or degenerate geometry: The fault is usually caused by importing and more specifically scaled importing. The best solution is to select the faulty entity in the details tree and remove it. The entity can then be recreated.

Geometry not G1-continuous: The fault is usually caused by importing and more specifically scaled importing. The best solution is to select the faulty entity in the details tree and remove it. The entity can then be recreated.

Open or non-periodic curve attached to ring edge: This fault can occur when a model is imported or the stitching tool has been used on the model. Please send the original model file to the FEKO support team.

Open or non-periodic nominal geometry attached to ring edge: This fault can occur when a model is imported or the stitching tool has been used on the model. Please send the original model file to the FEKO support team.

Vertex not on curve of edge: The fault is usually caused by importing and more specifically scaled importing. The best solution is to select the faulty entity and remove it. The entity can then be recreated.

Vertex not on nominal geometry: The fault is usually caused by importing and more specifically scaled importing. The best solution is to select the faulty entity and remove it. The entity can then be recreated.

Edge reversed: This fault only arises with imported models.

Nominal geometry in wrong direction: This situation arises with imported models.

SP-curves of edge not within tolerance: This fault should not arise. Please contact your FEKO support team.

SP-curves not within edge's tolerance of nominal geometry: This fault should not arise. Please contact your FEKO support team.

Vertices of edge touch: The fault is usually caused by importing and more specifically scaled importing. The best solution is to select the faulty entity and remove it. The entity can then be recreated. Note that it could be that the face has to be deleted to remove the faulty edge.

Faces incorrectly ordered at edge: The fault is usually caused by importing. It may be possible to explode the part and try to combine the entities again.

Vertex not on surface of face: The fault is usually caused by importing and more specifically scaled importing. The best solution is to select the faulty entity and remove it. The entity can then be recreated.

Edge not on surface of face: The fault is usually caused by importing. It may be possible to explode the part and try to combine the entities again.

Self-intersecting face (i.e. edge/edge inconsistency): The fault is usually caused by importing. It may be possible to explode the part and try to combine the entities again. See self intersecting geometry.

Edges incorrectly ordered at vertex: The fault is usually caused by importing. It may be possible to explode the part and try to combine the entities again.

Loops inconsistent: The fault is usually caused by importing and more specifically scaled importing. The best solution is to select the faulty entity and remove it. The entity can then be recreated.

Missing vertex at surface singularity: The fault is usually caused by importing and more specifically scaled importing. The best solution is to select the faulty entity and remove it. The entity can then be recreated.

Wire-frame edge/face inconsistency: This could be caused by unions involving wires and faced bodies. Please send the original model file to the FEKO support team.

Wire-frame edge/wire-frame edge inconsistency: This could be caused by unions involving wires and faced bodies. Please send the original model file to the FEKO support team.

Size-box violation: This fault should not arise. Please contact your FEKO support team. Please send the original model file to the FEKO support team.

Face-face inconsistency: The fault is usually caused by importing and more specifically scaled importing. The best solution is to select the faulty entity and remove it. The entity can then be recreated.

Body is inside out: This situation only arises with imported models.

Shells of region are inconsistent: This fault should not arise. Please contact your FEKO support team. Please send the original model file to the FEKO support team.

Regions of body are inconsistent: This fault should not arise. Please contact your FEKO support team. Please send the original model file to the FEKO support team.

Geometry/topology inconsistency in shell: This fault should not arise. Please contact your FEKO support team. Please send the original model file to the FEKO support team.

Acorn shell/shell inconsistency: This fault should not arise. Please contact your FEKO support team. Please send the original model file to the FEKO support team.

Unspecified checker failure or checker failure during face-face check: It should be possible to continue working, but the part may contain faults that can not be detected. Please contact your FEKO support team. Please send the original model file to the FEKO support team.

Non-printing character used in name of attribute definition: This situation only arises with imported models and it should be possible to continue working.

B-geometry has knots closer than the allowed precision: The fault is most probably caused by importing and more specifically scaled importing. The best solution is to select the faulty entity and remove it. The entity can then be recreated.

28 Copyright notices and acknowledgements

The copyright for FEKO lies with

Copyright 1998 – 2011: **EM Software & Systems-S.A. (Pty) Ltd**

However, FEKO uses also third-party libraries and components (both commercial and freely available ones). For some of the libraries the usage condition is that the corresponding copyright notices must be reproduced in the accompanying documentation. This shall be done here.

28.1 Copyright to Voronoi

The preprocessor PREFEKO uses part of the program voronoi to execute a Delaunay triangulation for some of the geometric cards. The copyright declaration for voronoi is as follows:

The author of this software is Steven Fortune. Copyright (c) 1994 by AT&T Bell Laboratories.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR AT&T MAKE ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

28.2 Copyright to Berkeley Spice 3f5

The FEKO kernel uses Berkeley Spice 3f5 to enable the integration of general SPICE networks within FEKO.

Copyright (c) 1985-1991 The Regents of the University of California.
All rights reserved.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

28.3 Copyright to SuperLU

The FEKO kernel is using for some parts the SuperLU library. The copyright declaration for SuperLU is as follows:

Copyright (c) 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy)

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- (1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- (3) Neither the name of Lawrence Berkeley National Laboratory, U.S. Dept. of Energy nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

28.4 Copyright of libcurl

The FEKO GUI makes use of libcurl to connect to the EMSS web-site to check for updates. The copyright declaration for libcurl is as follows:

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2005, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

28.5 Qwt project usage

For CADFEKO the figures in the mesh information dialogs are based, in part, on the work of the Qwt project (see <http://qwt.sf.net>).

28.6 HOOPS and Parasolid

POSTFEKO and CADFEKO use the HOOPS[®] 3D Application Framework from Tech Soft 3D for the 3D display and rendering. CADFEKO also uses the Parasolid[®] Kernel Modeller from UGS to represent the geometry.

28.7 MeshSim

CADFEKO uses MeshSim[™] from Simmetrix Inc. to mesh the geometry.

28.8 FFmpeg (export video files)

POSTFEKO utilises the FFmpeg application for the creation of animation videos (see <http://ffmpeg.org>) and is included in the FEKO distributions except FEKO LITE. EMSS-SA has obtained (as “video provider”) from MPEG-LA a MPEG-4 Visual Patent Portfolio License for commercial use. The included FFmpeg has been compiled to include only a limited subset of the video codecs.

28.9 Microsoft

The 2007 Microsoft Office Fluent User Interface is subject to protection under U.S. and international intellectual property laws and is used by EM Software & Systems - S.A. (Pty) Ltd under license from Microsoft.

28.10 Copyright of libXML2

The FEKO kernel is using for some parts the libXML2 toolkit. The copyright declaration is as follows:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Part X

Index

Index

- ** card, [13-3](#), [14-5](#)
- *.out file
 - basis functions, [19-5](#)
 - excitation, [19-5](#)
 - geometry, [19-1](#)
- *.pre file
 - geometry, [12-1](#)
 - solution control, [12-1](#)
 - structure, [12-1](#)
- 3D view, [3-1](#), [3-7](#)
 - point entry, [3-8](#), [3-10](#)
- A0 card, [14-9](#)
- A1 card, [14-12](#)
- A2 card, [14-13](#)
- A3 card, [14-15](#)
- A4 card, [14-17](#)
- A5 card, [14-19](#)
- A6 card, [14-20](#)
- A7 card, [14-21](#)
- AB card, [14-22](#)
- ABAQUS, [13-53](#)
- absorption, [6-65](#)
- AC card, [14-23](#)
- ACA, [13-27](#)
 - setting, [6-53](#)
- ADAPTFEKO, [21-1](#)
 - *.pre file for, [21-1](#)
 - running, [21-1](#)
- adaptive cross-approximation, [13-27](#)
- adaptive frequency sampling, [6-17](#), [21-1](#)
- advanced solution options, [6-70](#)
- AE card, [14-25](#)
- AF card, [14-28](#)
- AI card, [14-29](#)
- AK card, [14-31](#)
- align, [4-29](#)
- AN card, [14-34](#)
- analytical curve, [4-13](#)
- angle, [3-21](#)
- angle between points
 - CADFEKO, [3-21](#)
- angles
 - Kardan, [13-97](#)
- anisotropic layers, [9-12](#), [14-150](#)
- ANSYS, [13-52](#)
- AP card, [14-35](#)
- aperture, [14-35](#)
- AR card, [14-41](#)
- archive, [3-22](#)
- array sizes, [24-4](#), [24-5](#)
- AS card, [14-46](#)
- assemblies, [4-27](#)
- AutoCAD file, [13-44](#)
- AV card, [14-52](#)
- AW card, [14-54](#)
- axes
 - local, [4-7](#)
 - U, V, N, [4-6](#), [4-7](#)
- batch
 - mesh, [3-6](#)
- beam angle, [6-50](#)
- Bézier, [4-13](#)
- BL card, [13-4](#)
- BO card, [14-61](#)
- *.bof files, [12-1](#)
- Boolean operations, [4-20](#)
- border
 - PO correction edge, [13-56](#)
 - PO correction wedge, [13-61](#)
- BP card, [13-6](#)
- BQ card, [13-8](#)
- BT card, [13-10](#)
- CA card, [14-62](#)
- cable coupling, [14-62](#), [14-67](#), [14-80](#), [14-147](#)
- cable cross sections
 - bundle, [4-32](#)
 - coaxial, [4-32](#)
 - predefined, [4-32](#)
 - ribbon, [4-32](#)
 - single conductor, [4-32](#)
- cable harness, [4-36](#)
- cable irradiation, [14-62](#), [14-67](#), [14-80](#), [14-147](#)
- cable path, [4-34](#)
- cable shield
 - braided (Kley), [4-33](#)
 - define shield properties, [4-34](#)
 - solid (Schelkunhoff), [4-34](#)
- CableMod, [14-23](#), [14-77](#)
- cables, [4-31](#)
- CADFEKO, [3-1](#)
 - geometry and meshing, [4-1](#)
 - Optimisation, [7-1](#)
 - overview, [3-1](#)
 - setting solution, [6-1](#)
 - setting up optimisation, [7-1](#)

- calculating
 - antenna reception, [6-67](#)
 - error estimates, [6-68](#)
 - far fields, [6-61](#)
 - near fields, [6-64](#)
 - S-parameters, [6-59](#)
 - SAR, [6-65](#)
 - transmission/reflection coefficients, [6-60](#)
 - variables, [3-21](#)
- capacitance
 - loading, [14-121](#), [14-126](#), [14-127](#)
- card editing, [11-6](#)
- cascade
 - Network, [14-129](#)
- CB card, [13-12](#)
- CD card, [14-67](#)
- CDB file, [13-52](#)
- CF card, [14-71](#)
- CFIE, [14-71](#)
- *.cfm files, [6-69](#), [12-1](#)
- CG card, [14-73](#)
- check for updates, [3-5](#), [11-5](#)
- circular arcs, [4-13](#)
- circular cone, [13-57](#)
- circular dependencies, [4-2](#)
- circular disc, [13-62](#)
- circular hole, [13-76](#)
- CL card, [13-14](#)
- clash, [4-37](#)
- clear model, [3-4](#)
- click mouse, [3-7](#)
- cluster, [16-1](#)
- CM card, [14-77](#)
- CN card, [13-16](#)
- CO card, [6-13](#), [14-78](#)
- coarse segmentation, [13-55](#)
- coating of wires, [14-78](#)
- Coatings
 - viewing, [6-57](#)
- coatings, [2-7](#), [6-6](#)
 - faces, [6-12](#)
 - wires, [6-14](#)
- coaxial attachment approximation, [14-17](#)
- coaxial cable, [14-62](#)
- coil, [13-32](#)
- collapse, [4-28](#)
- collapse tree items, [3-15](#)
- colour
 - by coating medium, [3-12](#)
 - by face medium, [3-12](#)
 - by region medium, [3-12](#)
 - dielectrics, [6-1](#)
 - icon indicators, [6-1](#)
 - medium icons, [3-20](#)
- command line
 - mesh, [3-6](#)
 - parameters, [6-70](#)
- comments, [3-21](#), [13-3](#), [14-5](#)
- complexity, reducing, [4-23](#)
- component parameters, [6-70](#)
 - authentication, [6-72](#)
 - parallel execution, [6-72](#)
 - remote execution, [6-71](#)
 - SSPI, [6-72](#)
- CONCEPT file, [13-48](#)
- conducting faces
 - on dielectric, [6-12](#)
- conducting losses, [6-12](#)
- conductivity, [6-1](#), [6-5](#)
- cone primitives, [4-7](#)
- cones, [13-57](#)
- connection points
 - definition, [2-2](#)
- connectivity
 - rules for, [2-4](#)
- constants, [4-2](#)
- constructing solids, [6-11](#)
- contact us, [1-8](#)
- continuous frequency, [6-17](#)
- control cards, [12-1](#), [14-1](#)
 - **, [14-5](#)
 - A0, [14-9](#)
 - A1, [14-12](#)
 - A2, [14-13](#)
 - A3, [14-15](#)
 - A4, [14-17](#)
 - A5, [14-19](#)
 - A6, [14-20](#)
 - A7, [14-21](#)
 - AC, [14-23](#)
 - AE, [14-25](#)
 - AF, [14-28](#)
 - AI, [14-29](#)
 - AK, [14-31](#)
 - AN, [14-34](#)
 - AP, [14-35](#)
 - AR, [14-41](#)
 - AS, [14-46](#)
 - AV, [14-52](#)
 - AW, [14-54](#)
 - BO, [14-61](#)
 - CA, [14-62](#)
 - CD, [14-67](#)
 - CF, [14-71](#)

- CG, [14-73](#)
- CM, [14-77](#)
- CO, [14-78](#)
- CS, [14-80](#)
- DA, [14-82](#)
- DI, [14-93](#)
- DL, [14-97](#)
- EE, [14-99](#)
- EN, [14-100](#)
- FE, [14-101](#)
- FF, [14-106](#)
- FR, [14-109](#)
- GF, [14-112](#)
- L2, [14-118](#)
- L4, [14-119](#)
- LC, [14-120](#)
- LD, [14-121](#)
- LE, [14-122](#)
- LF, [14-124](#)
- LN, [14-125](#)
- LP, [14-126](#)
- LS, [14-127](#)
- LZ, [14-128](#)
- NW, [14-129](#)
- OE, [14-132](#)
- OS, [14-133](#)
- PS, [14-137](#)
- PW, [14-139](#)
- SH, [14-147](#)
- SK, [14-150](#)
- SP, [14-155](#)
- TL, [14-157](#)
- TR, [14-160](#)
- WD, [14-161](#)
- convergence accuracy, [6-18](#)
- copy, [4-27](#), [4-29](#)
 - geometry, [13-96](#)
 - multiple, [4-31](#)
 - original part, [4-28](#)
- copyright, [28-1](#)
- coupling, [14-155](#)
 - transmission line, [14-23](#), [14-77](#)
- create
 - geometry, [4-1](#), [4-4](#), [12-1](#)
 - solid regions, [6-11](#)
- creation history
 - removing, [4-28](#)
- CRIPTE, [14-23](#), [14-77](#)
- CS card, [14-80](#)
- cuboid primitives, [4-7](#)
- cuboidal volume elements, [13-18](#), [13-21](#), [13-86](#), [13-87](#)
- definition, [2-1](#)
- current sources, [6-43](#)
- currents
 - calculation request, [14-133](#)
 - line sources, [14-23](#), [14-28](#), [14-29](#), [14-52](#)
 - written to output files, [6-58](#)
- curves, [4-22](#)
 - arc, [13-14](#)
 - creating surfaces, [4-24](#)
- cutplanes, [3-11](#)
 - in POSTFEKO, [9-10](#)
 - through solids, [3-11](#)
- cylinder
 - UTD, [6-11](#)
- cylinder primitives, [4-7](#)
- cylinders, [13-114](#)
 - dielectric, [13-21](#)
 - UTD region, [6-11](#), [13-107](#)
- DA card, [14-82](#)
- DD card, [13-17](#)
- decouple
 - solutions, [6-54](#)
- decryption, [16-4](#)
- default face type, [6-12](#)
- delete
 - elements, [4-46](#)
 - faces and edges, [4-23](#), [4-24](#)
 - from archive, [3-22](#)
 - not while in use, [3-9](#)
- details tree, [3-15](#)
 - media display, [3-20](#)
- DI card, [14-93](#)
- dielectric, [2-6](#), [6-1](#), [13-68](#), [14-93](#), [14-97](#)
 - cuboid cylinder, [13-21](#)
 - cuboids, [13-18](#), [13-86](#), [13-87](#)
 - for mesh elements, [6-15](#)
 - losses, [6-1](#)
 - media, [6-1](#), [6-76](#)
 - sphere, [14-112](#)
 - thin sheets, [14-150](#)
- Dielectric sheets
 - viewing, [6-57](#)
- diffraction theory, [13-103](#)
- dimension, scaling, [13-93](#)
- dipole array aperture, [14-35](#)
- dipoles
 - point electric, [6-24](#)
 - point magnetic, [6-24](#)
- direction
 - normal vectors, [4-10](#), [4-23](#), [4-25](#), [4-26](#)
 - triangles, [4-43](#)

- disabled solution, [6-75](#)
- disc, [13-62](#)
- discrete elements, [14-124–14-128](#)
- display, *see* 3D display
 - items in CADFEKO, [3-12](#)
 - media, [6-17](#)
- distance between points, [3-2](#)
 - CADFEKO, [3-20](#)
- distorted mesh elements, [4-48](#)
- distributed load, [14-121](#)
- DK card, [13-18](#)
- DL card, [14-97](#)
- domain decomposition, [13-17](#)
- double precision, [6-52](#)
- DP card, [12-15](#), [13-20](#)
- dragging mouse, [3-7](#)
- duplicate elements, removing, [4-47](#)
- DXF file, [13-44](#)
- dynamic memory management, [24-4](#)
- DZ card, [13-21](#)
- edges, [3-15](#), [6-10](#)
 - definition, [2-2](#)
 - delete, [4-23](#)
 - length, [4-38](#), [4-48](#)
 - ports on, [6-34](#)
 - properties, [4-41](#)
- edit geometry, [4-19](#)
- EDITFEKO, [11-1](#)
 - card editor, [11-6](#)
 - edit menu, [11-3](#)
 - file menu, [11-3](#)
 - keystrokes, [11-9](#)
 - options, [11-5](#)
 - preferences, [11-5](#)
 - superuser mode, [11-6](#)
 - variables, [11-8](#)
 - with CADFEKO models, [6-74](#)
- EDITFEKO, program flow when using, [12-1](#)
- editor for model comments, [3-21](#)
- EE card, [14-99](#)
- efficiency, [14-139](#)
- EFIE, [14-71](#)
- EG card, [13-23](#)
- EL card, [13-25](#)
- electric dipoles, [6-24](#)
- electric fields
 - calculating, [14-101](#)
- elements, [13-10](#)
 - count, [4-48](#)
 - creation, *see* geometry cards
 - deleting, [4-46](#)
 - free edges, [4-49](#)
- ellipse, [4-10](#)
- ellipsoid, [13-25](#)
- elliptical arcs, [4-13](#)
- elliptical hole, [13-76](#)
- ELSE statement, [12-13](#)
- EN card, [14-100](#)
- encryption, [16-4](#)
- end of geometry, [13-23](#)
- end of input file, [14-100](#)
- environment of model, [6-46](#)
- environment variables, [24-7](#)
- equivalent aperture, [14-35](#)
- error estimation, [14-99](#)
- error messages, [3-5](#)
- excitations, *see* sources
 - current sources, [6-43](#)
 - FEM modal mode, [6-45](#)
 - Hertzian electric dipole, [6-24](#)
 - impressed aperture excitation, [6-29](#)
 - impressed current, [6-30](#)
 - impressed line current, [6-39](#)
 - impressed spherical mode, [6-27](#)
 - incident plane waves, [6-23](#)
 - magnetic dipole, [6-24](#)
 - radiation patterns, [6-25](#)
 - total power, [6-19](#)
 - voltage sources, [6-42](#)
 - waveguide mode, [6-44](#)
- EXIT command, [12-14](#)
- expand tree items, [3-15](#)
- explode, [4-22](#)
- export, [6-69](#)
 - POSTFEKO animations, [8-10](#)
 - POSTFEKO images, [8-10](#)
 - calculated current and charges, [6-58](#)
 - calculated far fields, [6-61](#)
 - calculated near fields, [6-64](#)
 - calculated S-parameters, [6-59](#)
 - geometry, [5-2](#)
 - Parasolid, [4-1](#), [5-2](#)
- expressions, [4-2](#)
 - functions, [4-2](#)
- extents, [4-1](#)
- extrude parts, [4-24](#)
- face displacement, [3-4](#)
- faces, [3-15](#), [4-10](#), [4-22](#), [6-10](#), *see* surfaces
 - coatings, [6-12](#)
 - conducting, [6-12](#)
 - default type, [6-12](#)
 - delete, [4-23](#)

- properties, [4-41](#), [6-12](#)
 - reverse normals, [4-26](#)
 - solution method, [6-57](#)
 - solution settings, [6-56](#)
- far fields, [6-61](#)
 - calculating, [14-106](#)
- FE card, [14-101](#)
- feed, *see* sources
- FEK file format, [17-1](#)
- *.fek files, [5-5](#), [12-1](#)
- FEKO, [18-1](#)
- FEKO_USER_HOME
 - environment variable, [24-7](#)
- FEKO_WRITE
 - environment variable, [24-7](#)
- FEM, [2-6](#), [13-30](#)
 - FEM modal port, [6-41](#)
 - impressed currents, [6-39](#)
 - Line port, [6-39](#)
 - reducing memory requirement, [2-3](#)
 - setting, [6-54](#), [6-55](#)
- FEM line port, [6-39](#)
- FEM modal
 - excitation, [6-45](#)
- FEM modal mode, [6-45](#)
- FEM modal port, [6-41](#)
- FEMAP, [13-38](#)
- FEST3D file, [14-82](#)
- FF card, [14-106](#)
- file format, [17-1](#)
- files, [14-82](#)
 - *.cfm, [6-69](#)
 - input, [12-1](#)
 - output, [19-1](#)
 - Parasolid, [5-2](#)
 - *.pre, [6-69](#)
 - summary of, [25-1](#)
 - used by components, [2-1](#), [12-1](#)
- finding
 - distorted elements, [4-48](#)
 - free mesh edges, [4-48](#)
 - free segment nodes, [4-48](#)
 - intersecting mesh elements, [4-48](#)
- flare primitives, [4-7](#)
- FM card, [13-27](#)
- FO card, [13-29](#)
- Fock area, [13-29](#)
- FOR/NEXT loops, [12-12](#)
- formulations
 - in FEKO, [1-1](#)
- FP card, [13-30](#)
- FR card, [14-109](#), [21-1](#)
- free edges, [4-37](#), [4-49](#)
- free mesh edges, [4-48](#)
- free segment nodes, [4-48](#)
- free space, [6-11](#)
- frequency, [6-17](#), [14-109](#)
 - adaptive sampling, [21-1](#)
- functions
 - in expressions, [4-2](#)
- general comments, [2-1](#)
- general networks, [6-20](#)
 - applying excitation, [6-22](#)
 - applying loads, [6-22](#)
 - connecting, [6-22](#)
 - creating, [6-21](#)
- General non-radiating networks
 - feeding, [14-34](#)
- geometrical optics, [13-103](#)
- geometry
 - align, [4-29](#)
 - CAD translation, [5-3](#)
 - creation, [4-1](#), [4-4](#), [12-1](#)
 - entering, [13-36](#)
 - extents, [4-1](#)
 - importing, [3-1](#), [5-2](#)
 - imprinting points on, [4-21](#)
 - in tree, [3-14](#)
 - mirror, [4-29](#)
 - modify, [4-19](#)
 - point entry, [3-10](#)
 - point import, [4-18](#)
 - ports on, [6-31](#)
 - re-evaluate tree, [4-28](#)
 - rotate, [4-29](#)
 - scale, [4-29](#)
 - show / hide parts, [3-12](#)
 - solids, [4-7](#)
 - surfaces, [4-10](#)
 - translate, [4-29](#)
 - validation, [4-37](#)
 - wires, [4-13](#)
- geometry cards, [12-1](#), [13-1](#)
 - **, [13-3](#)
 - BL, [13-4](#)
 - BP, [13-6](#)
 - BQ, [13-8](#)
 - BT, [13-10](#)
 - CB, [13-12](#)
 - CL, [13-14](#)
 - CN, [13-16](#)
 - DK, [13-18](#)
 - DP, [13-20](#)

DZ, [13-21](#)
 EG, [13-23](#)
 EL, [13-25](#)
 FM, [13-27](#)
 FO, [13-29](#)
 FP, [13-30](#)
 HC, [13-31](#)
 HE, [13-32](#)
 HP, [13-34](#)
 HY, [13-35](#)
 IN, [13-36](#)
 IP, [13-55](#)
 KA, [13-56](#)
 KK, [13-57](#)
 KL, [13-61](#)
 KR, [13-62](#)
 KU, [13-64](#)
 LA, [13-66](#)
 MB, [13-67](#)
 ME, [13-68](#)
 NU, [13-71](#)
 PB, [13-73](#)
 PH, [13-76](#)
 PM, [13-78](#)
 PO, [13-80](#)
 PY, [13-84](#)
 QT, [13-86](#)
 QU, [13-87](#)
 RM, [13-89](#)
 SF, [13-93](#)
 SY, [13-95](#)
 TG, [13-96](#)
 TO, [13-99](#)
 TP, [13-101](#)
 UT, [13-103](#)
 UZ, [13-107](#)
 VS, [13-108](#)
 WA, [13-111](#)
 WG, [13-112](#)
 WR, [13-113](#)
 ZY, [13-114](#)
 GF card, [14-112](#)
 GiD, [13-54](#)
 GO, [13-103](#)
 setting, [6-54](#)
 viewing, [6-57](#)
 GPU acceleration, [6-71](#)
 Green's functions, [2-7](#), [6-46](#), [14-112](#)
 grid, [4-6](#)
 ground planes, [6-46](#), [14-61](#)
 HC card, [13-31](#)
 HE card, [13-32](#)
 helices, [4-13](#), [13-32](#)
 help, [9-28](#)
 Hertzian electric dipole, [14-19](#)
 hide
 items behind cutplane, [3-11](#)
 single item, [3-12](#), [3-15](#)
 history
 removing, [4-28](#)
 hot-keys, [3-22](#)
 HP card, [13-34](#)
 HY card, [13-35](#)
 hyperbolic, [13-31](#), [13-34](#), [13-35](#)
 icons
 in tree, [3-15](#)
 ideal receiving antenna, [14-143](#)
 ideal transmission lines, [6-20](#)
 applying excitation, [6-22](#)
 applying loads, [6-22](#)
 connecting, [6-22](#)
 creating, [6-20](#)
 IF statement, [12-13](#)
 impedance, [14-124](#), [14-125](#), [14-128](#)
 loading, [6-45](#), [14-122](#), [14-124](#), [14-125](#)
 microstrip fed, [14-122](#)
 sheets, [6-7](#)
 impedance boundary condition, [14-150](#)
 import
 CAD formats, [5-3](#)
 geometry, [3-1](#), [5-2](#), [6-11](#), [13-36](#)
 meshes, [3-1](#), [5-5](#), [12-1](#), [13-36](#)
 points, [4-18](#)
 impressed aperture excitation, [6-29](#)
 impressed current, [6-30](#)
 impressed line current, [14-23](#), [14-28](#), [14-29](#), [14-52](#)
 in FEM region, [6-39](#)
 impressed spherical mode, [6-27](#)
 imprinting points, [4-21](#)
 IN card, [13-36](#)
 in use
 deleting items, [3-9](#)
 incident plane wave, [14-9](#)
 include files, [13-36](#)
 inductance
 loading, [14-121](#), [14-126](#), [14-127](#)
 infinite planes, [6-46](#)
 instances of ports, [6-31](#)
 intersect, [4-20](#)
 intersecting mesh elements, [4-48](#)
 IP card, [13-55](#)
 KA card, [13-56](#)

- Kardan angles, [13-97](#)
- KK card, [13-57](#)
- KL card, [13-61](#)
- KR card, [13-62](#)
- KU card, [13-64](#)
- L2 card, [14-118](#)
- L4 card, [14-119](#)
- LA card, [13-66](#)
- label selected calculation, [14-132](#)
- labels, [4-27](#), [12-4](#), [13-12](#), [13-66](#), [13-67](#)
 - changing, [4-45](#)
 - using, [4-45](#), [6-75](#)
- large element PO
 - setting, [6-56](#)
- large models, [12-4](#)
- layered dielectrics, [6-6](#)
- layered dielectrics (anisotropic), [6-7](#)
- layered media, [14-160](#)
- LC card, [14-120](#)
- LD card, [14-121](#)
- LE card, [14-122](#)
- length, [3-20](#)
- lens, [13-31](#), [13-34](#), [13-35](#)
- LF card, [14-124](#)
- licence, [15-1](#)
 - check in, [15-5](#)
 - configuration file, [15-6](#)
 - floating, [15-3](#)
 - include/exclude, [15-5](#)
 - information, [15-1](#)
 - preferred, [15-4](#)
 - server, [15-4](#)
- line, [4-13](#)
- line segments, *see* segments
- linear set of equations, [14-73](#)
- LN card, [14-125](#)
- loading, [14-62](#), [14-157](#)
 - an edge, [14-122](#)
 - attachment point, [14-119](#)
 - cable, [14-120](#)
 - distributed, [14-121](#)
 - FEM, [14-124](#)
 - impedance, [6-45](#), [14-128](#)
 - microstrip line, [14-122](#)
 - Network, [14-125](#)
 - parallel circuit, [6-45](#), [14-126](#)
 - series circuit, [6-45](#), [14-127](#)
- local mesh parameters, [4-41](#)
- lock point entry, [3-10](#)
- loft, [4-26](#)
- log files, [3-5](#)
- losses, [14-139](#)
 - conducting, [6-12](#)
 - on faces, [6-12](#)
 - on wires, [6-14](#)
- LP card, [14-126](#)
- LS card, [14-127](#)
- LZ card, [14-128](#)
- machine codes, [15-6](#)
- magnetic cuboids, [13-18](#)
- magnetic dipoles, [6-24](#), [14-20](#)
- magnetic fields
 - calculating, [14-101](#)
- magnetic ring current, [14-15](#)
- magnitude
 - peak convention, [6-19](#)
- make primitive, [4-28](#)
- manual solution control, [24-1](#)
- maxalloc(m), [24-4](#), [24-5](#)
- maximum constants, [24-4](#), [24-5](#)
- MB card, [13-67](#)
- ME card, [13-68](#)
- media
 - colour, [3-12](#), [6-1](#)
 - dielectric, [6-1](#), [13-68](#), [13-86](#), [13-87](#)
 - display, [6-17](#)
 - impedance sheets, [6-7](#)
 - in details tree, [3-20](#)
 - layered dielectrics, [6-6](#)
 - layered dielectrics (anisotropic), [6-7](#)
 - lists of, [6-1](#)
 - magnetic, [13-86](#), [13-87](#)
 - media library, [6-6](#)
 - metallic, [6-1](#), [6-5](#)
 - on imported meshes, [6-15](#)
 - on imported models, [6-10](#)
 - properties, [6-76](#)
 - setting, [6-10](#)
 - using, [6-1](#)
- memory
 - allocation, [24-4](#)
 - reducing FEM requirement, [2-3](#)
- menu
 - pop-up, [3-7](#), [3-15](#)
- merge
 - mesh, [4-47](#)
- merging
 - vertices, [4-46](#)
- mesh
 - batch, [3-6](#)
 - command line, [3-6](#)
 - merge, [4-47](#)

- union, [4-47](#)
- mesh refinement, [4-44](#)
- meshes, [4-38](#)
 - adaptive refinement, [6-68](#)
 - adding triangles, [4-43](#)
 - edit vertex, [4-42](#)
 - importing, [3-1](#), [5-5](#), [12-1](#)
 - information, [4-48](#)
 - local settings, [4-41](#)
 - point refinement, [4-44](#)
 - polyline refinement, [4-44](#)
 - ports, [6-31](#)
 - properties, [6-15](#)
 - refining, [13-89](#)
 - setting media, [6-15](#)
 - show / hide parts, [3-12](#)
 - small features, [4-40](#)
 - smoothing, [4-40](#)
 - tetrahedra, [4-40](#)
- meshing, [2-1](#)
 - non-uniform, [13-6](#), [13-8](#), [13-10](#)
 - rules, [2-2](#), [2-4](#), [12-3](#)
- message window, [3-1](#)
- messages, [3-5](#)
- metallic media, [6-5](#)
- microstrip
 - port, [6-36](#)
- microstrip port, [6-36](#)
- mirror, [4-29](#)
- mismatch loss, [6-19](#)
- MLFMM, [13-27](#)
 - setting, [6-53](#), [6-57](#)
- modal port, [13-67](#)
- model
 - creating, [3-4](#)
 - unit, [3-2](#), [4-1](#)
 - validation, [6-69](#)
- modelling
 - guidelines, [2-1](#)
 - rules, [2-1](#)
- modify geometry, [4-19](#)
- mouse interaction, [3-7](#)
 - point entry, [3-10](#)
- multilayer substrates
 - defining, [14-112](#)
- multiple reflections, [13-80](#)
- multiple selection, [3-9](#)
- N axis, [4-6](#), [4-7](#)
- named points, [4-4](#)
 - in *.pre file, [6-76](#)
- names, [4-27](#), [6-75](#)
- NASTRAN file, [13-42](#)
- near fields, [6-64](#)
 - calculating, [14-101](#)
 - calculation offset, [14-132](#)
- NEC file, [13-47](#)
- Network
 - cascade, [14-129](#)
 - loading, [14-125](#)
- networks, [14-62](#), [14-157](#)
- Neutral files, [13-38](#)
- new
 - model, [3-4](#)
- nodes
 - defining, [13-20](#)
 - definition, [2-2](#)
 - variable names, [12-15](#), [13-20](#)
- non-uniform mesh, [13-6](#), [13-8](#)
- normal vectors, [4-23](#), [4-25](#), [13-16](#)
 - reverse, [4-26](#), [4-43](#)
- notes, [3-21](#)
- NU card, [13-71](#)
- NURBS, [4-10](#)
- NURBS surfaces, [13-71](#)
- NW card, [14-129](#)
- OF card, [14-132](#)
- offset for near field calculation, [14-132](#)
- ohmic losses, [6-5](#), [6-12](#), [14-150](#)
- old models
 - re-evaluate, [4-28](#)
- open model, [3-4](#)
- OPTFEKO, [20-1](#)
 - Optimisation method, [20-1](#)
- Optimisation
 - CADFEKO setup, [7-1](#)
 - far field goal, [7-14](#)
 - goal combination tool, [7-19](#)
 - goal weighting, [7-19](#)
 - goals, [7-7](#)
 - impedance goal, [7-11](#)
 - masks, [7-5](#)
 - method, [7-2](#)
 - near field goal, [7-12](#)
 - parameters, [7-3](#)
 - s-parameter goal, [7-15](#)
 - SAR goal, [7-17](#)
 - search, [7-1](#)
 - stopping criteria, [7-2](#)
 - the global goal, [7-19](#)
- optimisation
 - farming, [20-13](#)
 - Genetic algorithm (GA), [20-8](#)

- Grid search, [20-10](#)
- Particle swarm (PSO), [20-5](#)
- running, [20-13](#)
- sensitivity analysis, [20-12](#)
- Simplex, [20-2](#)
- Optimisation search
 - Defining, [7-1](#)
 - Multiple, [7-1](#)
- options
 - EDITFEKO, [11-5](#)
 - rendering, [3-4](#)
 - solution components, [6-70](#)
- OS card, [14-133](#)
- *.out files, [12-1](#)
- out-of-core, [24-4](#)
- output file, [19-1](#)
- overlap, [4-37](#)
- overview, [1-1](#)
- package, [16-1](#)
- pan
 - model view, [3-7](#)
- paraboloid, [4-10](#), [13-73](#)
- parallel execution, [11-4](#)
- parallelogram, [13-6](#)
- parameters
 - points, [4-4](#)
 - segmentation, [13-55](#)
- parametric models, [3-1](#), [4-2](#)
- Parasolid, [5-2](#)
- parents, [3-14](#)
- parts, [3-14](#)
 - show / hide, [3-12](#)
- PATRAN file, [13-51](#)
- patterned point source, [6-25](#)
- PB card, [13-73](#)
- PCBMod, [14-23](#), [14-77](#)
- PDF viewer, [3-4](#)
- PE card, [13-75](#)
- peak magnitude, [6-19](#)
- Periodic boundary conditions, [6-50](#)
- permeability, [6-1](#)
- permittivity, [6-1](#)
- PH card, [13-76](#)
- physical optics, [13-80](#)
- planar, [6-56](#)
- planar substrate, [2-7](#), [14-112](#)
- plane wave, [14-160](#)
- plane wave excitation, [6-23](#), [14-9](#)
- plate with hole, [13-76](#)
- PM card, [13-78](#)
- PO
 - accelerated ray tracing, [6-56](#)
 - edge border, [13-56](#)
 - plane wave excitation, [6-56](#)
 - setting, [6-54](#), [6-56](#)
 - viewing, [6-57](#)
 - visibility, [13-108](#)
 - wedge border, [13-61](#)
- PO card, [13-80](#)
- point dipoles, [6-24](#)
- point entry, [3-10](#), [4-6](#)
 - snap mode, [3-8](#)
- point source with pattern, [6-25](#)
- points
 - angle, [3-21](#)
 - distance, [3-20](#)
 - geometry, [3-10](#)
 - imprinting, [4-21](#)
 - in *.pre file, [6-76](#)
 - named, [4-4](#)
 - names, *see* nodes
- polarisation
 - incident waves, [6-23](#)
- polygons, [4-10](#)
 - definition, [2-1](#)
 - meshed, [13-78](#)
 - normals, [4-43](#)
 - polygonal plates, [13-84](#)
 - UTD formulation, [13-103](#)
- polyline, [4-13](#)
- ports, [6-31](#)
 - edge, [6-34](#)
 - instances, [6-31](#)
 - licence server, [15-3](#)
 - microstrip, [6-36](#)
 - on wires, [6-33](#)
 - waveguide, [6-37](#)
- POSTFEKO, [8-1](#)
- power input, [6-19](#), [14-139](#)
- PP card, [14-135](#)
- *.pre files, [6-69](#), [12-1](#)
 - named points, [6-76](#)
 - variables, [6-76](#)
- precision, [6-52](#)
- preconditioners, [6-53](#), [6-54](#)
- PREFEKO, [17-1](#)
- preferences, [3-4](#)
 - EDITFEKO, [11-5](#)
- preferred licence, [15-4](#)
- preview mode, [3-10](#)
- primitive
 - collapsing part tree, [4-28](#)
- priority

- setting, [18-1](#)
- program execution control, [14-137](#)
- program flow, [2-1](#)
- project control, [3-22](#)
- project notes, [3-21](#)
- projection, [4-21](#)
- properties
 - medium, [6-11](#)
 - mesh, [4-41](#), [6-15](#)
 - modify, [4-19](#)
- PS card, [14-137](#)
- PW card, [14-139](#)
- PY card, [13-84](#)
- QT card, [13-86](#)
- QU card, [13-87](#)
- quadrangle, [13-8](#)
- quantity palette, [8-3](#)
- QUEUEFEKO
 - CLUSTER options, [16-3](#)
 - configuration files, [16-2](#)
 - creating packages, [16-1](#)
 - Decryption of package, [16-6](#)
 - Execution queue, [16-5](#)
 - Extract packages, [16-5](#)
 - extracting packages, [16-1](#)
 - FEKO options, [16-3](#)
 - Generate package, [16-5](#)
 - Including package files, [16-2](#)
 - Setting preferences, [16-6](#)
 - using encryption, [16-4](#)
- queuefeko, [16-1](#)
- RA card, [14-143](#)
- radiation
 - patterns, [14-106](#)
 - patterns as source, [6-25](#)
 - patterns as sources, [14-41](#)
- radius
 - segments, [4-38](#), [4-41](#)
- ray file, [6-54](#)
- RCS
 - radar cross section, [6-61](#)
- re-evaluate
 - the geometry, [4-28](#)
- re-use
 - solution, [6-52](#)
- real ground, [14-61](#), [14-112](#)
- receiving antenna, [6-67](#)
- rectangle, [4-10](#)
- redo
 - selection, [3-9](#)
- view operations, [3-13](#)
- referencing elements, [6-75](#)
- reflectors, [4-10](#)
- regions, [3-15](#), [6-10](#)
 - properties, [4-41](#), [6-11](#)
- registry keys, [24-7](#)
- relative
 - permeability, [6-1](#)
 - permittivity, [6-1](#)
- remeshing, [13-89](#)
 - CADFEKO models, [12-4](#)
- remote, [16-1](#)
- remote execution, [11-4](#)
- remove
 - duplicates, [4-47](#)
 - slivers, [4-46](#)
- renaming
 - geometry, [4-19](#)
 - mesh elements, [4-45](#)
- rendering options, [3-4](#)
- request
 - cable harness, [4-36](#)
 - antenna reception, [6-67](#)
 - current output, [6-58](#)
 - error estimations, [6-68](#)
 - far fields, [6-61](#)
 - near fields, [6-64](#)
 - S-parameters, [6-59](#)
 - SAR, [6-65](#)
 - transmission/reflection coefficients, [6-60](#)
- resistance
 - loading, [14-124](#)–[14-126](#)
- resistance loading, [14-121](#), [14-122](#), [14-127](#)
- restore
 - deleted faces / edges, [4-28](#)
- results, [6-57](#)
 - checking validity, [2-7](#)
 - continuous, [21-1](#)
 - time-domain, [22-1](#)
- reuse objects, [4-27](#)
- RM card, [13-89](#)
- rotate
 - geometry, [4-29](#), [13-96](#)
 - model view, [3-7](#)
 - parts, [4-24](#)
 - point, [13-101](#)
- ruled surface, [4-26](#)
- run
 - components, [6-69](#)
 - from GUI, [11-4](#)
- S-parameters, [6-59](#), [14-155](#)

- SA card, [14-145](#)
- SAR
 - specific absorption rate, [6-65](#), [14-145](#)
- save
 - as UNIX, [11-3](#)
 - model, [3-4](#)
- SBR, [13-103](#)
- scale, [4-29](#)
 - geometry, [13-93](#), [13-96](#)
 - Parasolid, [5-2](#)
 - units, [4-1](#), [6-75](#)
- scattering, [6-61](#)
- scripting, [2-1](#), [10-1](#)
- secfeko, [15-1](#)
- security, [16-4](#)
- segmentation
 - parameters, [13-55](#)
 - rules for, [2-2](#), [2-4](#), [12-3](#)
- segments, [13-4](#)
 - arc, [13-14](#)
 - coating, [14-78](#)
 - creation, *see* geometry cards
 - definition, [2-1](#)
 - helix, [13-32](#)
 - ports on, [6-33](#)
 - radius, [4-38](#), [4-41](#)
- selection
 - in 3D view, [3-8](#)
 - items, [3-8](#)
 - multiple, [3-9](#)
 - toolbar, [3-8](#)
- SEMCAD, [14-83](#)
- server
 - licence, [15-3](#)
- setting media, [6-10](#)
- SF card, [13-93](#)
- SH card, [14-147](#)
- shells, [6-11](#)
- shooting and bouncing rays, [13-103](#)
- short-cut keys, [3-22](#)
 - EDITFEKO, [11-9](#)
- show
 - single item, [3-12](#), [3-15](#)
- sides
 - of triangles, [4-43](#)
- simplify, [4-23](#)
- single precision, [6-52](#)
- single selection, [3-8](#)
- size
 - meshing rules, [2-2](#)
- SK card, [6-13](#), [14-150](#)
- skin depth, [2-2](#), [14-150](#)
- slivers
 - removing, [4-46](#)
- small features
 - remove, [4-40](#)
- snap mode, [3-2](#), [3-8](#)
- solid bodies, [4-7](#), [6-11](#)
 - versus shell bodies, [6-11](#), [6-76](#)
- solution, [11-4](#)
 - numerical Green's function, [6-51](#)
 - advanced control, [24-1](#)
 - component parameters, [6-70](#)
 - disabled, [6-75](#)
 - face settings, [6-56](#)
 - NGF, [6-51](#)
 - viewing, [6-57](#)
- solver
 - settings, [6-52](#)
- solving, [6-69](#)
- sources, [14-6](#)
 - current sources, [6-43](#)
 - FEM modal, [6-41](#)
 - Hertzian electric dipole, [6-24](#), [14-19](#)
 - impressed aperture excitation, [6-29](#)
 - impressed current, [6-30](#)
 - impressed line current, [6-39](#), [14-23](#), [14-28](#), [14-29](#), [14-52](#)
 - impressed spherical mode, [6-27](#)
 - incident plane wave, [14-9](#)
 - incident plane waves, [6-23](#)
 - magnetic dipole, [6-24](#), [14-20](#)
 - magnetic ring current, [14-15](#)
 - microstrip line, [14-25](#)
 - patch feed pin, [14-17](#)
 - radiation patterns, [6-25](#), [14-41](#)
 - spherical modes, [14-46](#)
 - total power, [6-19](#)
 - voltage on a node, [14-13](#)
 - voltage on a non-radiating network port, [14-34](#)
 - voltage on a radiating cable, [14-31](#)
 - voltage on a segment, [14-12](#)
 - voltage on an edge, [14-21](#), [14-25](#)
 - voltage sources, [6-42](#)
 - waveguide modes, [14-54](#)
- SP card, [14-155](#)
- specified vertex, [4-41](#)
- sphere primitives, [4-7](#)
- spheres, [13-64](#)
 - dielectric/magnetic, [13-18](#)
- spherical modes, [14-46](#)
- SPICE3f5
 - general structure, conventions and syntax, [26-1](#)
- spinning parts, [4-24](#)

- spirals, [4-13](#), [13-32](#)
- splines, [4-13](#)
- split parts, [4-22](#)
- squint angle, [6-50](#), [14-135](#)
- SSPI, [6-72](#)
- start page, [8-3](#)
- status bar
 - cut plane, [3-11](#)
 - distance, [3-2](#)
 - model unit, [3-2](#)
 - snap mode, [3-2](#)
- stitch, [4-24](#)
- STL file, [13-49](#)
- stripline
 - feeding, [14-25](#)
 - loading, [14-119](#)
- substrate, [14-112](#)
- subtract, [4-20](#)
- superuser mode
 - in EDITFEKO, [11-6](#)
- surface currents
 - distribution, [14-133](#)
- surface equivalence, [2-6](#)
- surface resistance, [6-7](#)
- surface triangles, *see* elements
- surfaces, [4-10](#), [4-22](#)
 - conducting, [6-12](#)
 - creating solids, [4-24](#)
- suspect, [4-19](#)
- sweeping parts, [4-24](#)
- SY card, [13-95](#)
- symbolic variables, [12-6](#)
- symmetry, [6-49](#), [13-95](#), [24-1](#)
 - electric, [24-1](#)
 - geometric, [24-1](#)
 - magnetic, [24-2](#)
- tab
 - Display options, [3-10](#)
- tetrahedral volume elements, [13-86](#)
 - definition, [2-1](#)
- TG card, [13-96](#)
- thin dielectric sheets, [2-7](#), [6-6](#), [6-12](#), [14-150](#)
- TIMEFEKO, [22-1](#)
 - *.pre file for, [22-1](#)
 - *.tim file for, [22-2](#)
 - output of, [22-7](#)
 - running, [22-7](#)
- TL card, [14-157](#)
- TO card, [13-99](#)
- tolerance, [4-46](#)
- toolbars, [3-1](#)
 - selection, [3-8](#)
- toroidal segment, [13-99](#)
- TP card, [13-101](#)
- TR card, [14-160](#)
- transform
 - view, [3-13](#)
- transformation, [4-29](#), [13-96](#)
 - of meshes, [4-31](#)
 - of point, [13-101](#)
- translation, [4-29](#), [13-96](#), [13-101](#)
- transmission line theory, [14-62](#)
- transmission lines, [14-157](#)
 - coupling, [14-23](#), [14-77](#)
- transmission/reflection coefficients, [6-60](#), [14-160](#)
- tree view, [3-1](#), [3-14](#)
 - icons, [3-15](#)
- triangles, *see* elements
 - back side, [4-43](#)
 - creating, [4-43](#)
 - normal side, [4-43](#)
 - normals, [4-43](#)
 - remove duplicates, [4-47](#)
- U axis, [4-6](#), [4-7](#)
- unconnected edges, [4-37](#), [4-49](#)
- undo
 - list depth, [3-4](#)
 - selection, [3-9](#)
 - view operations, [3-13](#)
- union, [4-20](#)
 - mesh, [4-47](#)
- units, [4-1](#), [6-75](#)
- updates, [3-5](#), [11-5](#)
- UT card, [13-103](#)
- UTD, [13-103](#)
 - cylinder, [6-11](#), [13-107](#)
 - polygonal plate, [13-84](#)
 - setting, [6-54](#)
 - viewing, [6-57](#)
- UZ card, [13-107](#)
- V axis, [4-6](#), [4-7](#)
- validation, [2-7](#), [6-69](#)
 - geometry, [4-37](#)
- variables, [3-21](#), [4-2](#)
 - environment, [24-7](#)
 - in *.pre file, [6-76](#)
 - in point names, [12-15](#)
 - memory allocation, [24-4](#)
 - predefined, [4-2](#), [24-4](#)
 - symbolic, [12-6](#)
- vertex

- loading, [14-118](#)
- vertices
 - definition, [2-1](#)
 - editing, [4-42](#)
 - merging, [4-46](#)
 - ports on, [6-33](#)
 - specified location, [4-41](#)
- view
 - 3D default, [3-10](#)
 - manipulation, [3-7](#)
 - transform, [3-13](#)
 - wire-frame, [3-10](#)
- visibility
 - PO region, [13-108](#)
 - view items, [3-13](#)
- voltage sources, [6-42](#)
 - on a node, [14-13](#)
 - on a non-radiating network port, [14-34](#)
 - on a port, [6-31](#)
 - on a radiating cable, [14-31](#)
 - on a segment, [14-12](#)
 - on an edge, [14-21](#), [14-25](#)
- volume equivalence, [2-6](#)
- VS card, [13-108](#)
- WA card, [13-111](#)
- waveguide
 - excitation, [6-44](#)
 - port, [6-37](#)
- waveguide mode, [6-44](#)
- waveguide port, [6-37](#), [14-54](#)
- WD card, [14-161](#)
- wedge as PO border, [13-61](#)
- WG card, [13-112](#)
- windscreen
 - antenna, [6-8](#)
 - dielectric, [6-8](#)
 - reference, [6-8](#)
 - WA, [13-111](#)
 - WD, [14-161](#)
 - WR, [13-113](#)
- wire-frame view, [3-10](#)
- wires, [6-10](#)
 - as objects, [4-13](#)
 - coatings, [6-14](#)
 - curved, [13-14](#)
 - grid, [13-112](#)
 - lossy, [6-14](#)
 - mesh into segments, [4-38](#)
 - ports on, [6-33](#)
 - properties, [6-14](#)
 - segments, *see* segments
- workplane, [4-6](#)
 - coordinates, [4-7](#)
- WR card, [13-113](#)
- zoom
 - model view, [3-7](#)
- ZY card, [13-114](#)