

Sonnet® User's Guide

Release 10



Cover: James Clerk Maxwell (1831-1879). A professor at Cambridge University, England, Maxwell established the interdependence of electricity and magnetism. In his classic treatise of 1873, he published the first unified theory of electricity and magnetism and founded the science of electromagnetism.

Sonnet® User's Guide

Printed: November 2004

Release 10

Sonnet Software, Inc.

100 Elwood Davis Road

North Syracuse, NY 13212

Phone: (315) 453-3096

Fax: (315) 451-1694

Technical Support: support@sonnetsoftware.com

Sales Information: sales@sonnetsoftware.com

www.sonnetsoftware.com

© Copyright 1989,1991,1993, 1995-2005 Sonnet Software, Inc. All Rights Reserved

Registration numbers: TX 2-723-907, TX 2-760-739

Copyright Notice

Reproduction of this document in whole or in part, without the prior express written authorization of Sonnet Software, Inc. is prohibited. Documentation and all authorized copies of documentation must remain solely in the possession of the customer at all times, and must remain at the software designated site. The customer shall not, under any circumstances, provide the documentation to any third party without prior written approval from Sonnet Software, Inc. This publication is subject to change at any time and without notice. Any suggestions for improvements in this publication or in the software it describes are welcome.

Trademarks

The program names ***xgeom***, ***emstatus***, ***emvu***, ***patvu***, ***dxfgéo***, ***ebridge***, ***emgraph***, ***gds***, ***emserver***, ***emclient*** and ***sonntawr*** (lower case bold italics), Lite, LitePlus, Level2 Basic, Level2 Silver, and Level3 Gold are trademarks of Sonnet Software, Inc.

Sonnet® and ***em***® are registered trademarks of Sonnet Software, Inc.

UNIX is a trademark of Unix Systems Labs.

Windows 95, Windows 98, Windows2000, Windows ME, Windows XP and Windows NT are trademarks of Microsoft, Inc.

X Window System is a trademark of the Massachusetts Institute of Technology.

AutoCAD and Drawing Interchange file (DXF) are trademarks of Auto Desk, Inc.

SPARCsystem Open Windows, SUN, SUN-4, SunOS, Solaris, SunView, and SPARCstation are trademarks of Sun Microsystems, Inc.

HP, HP-UX, Hewlett-Packard are trademarks of Hewlett-Packard Company.

ADS, Series IV, Touchstone, and Libra are trademarks of Agilent Technologies.

GDSII is a trademark of Calma Company.

FLEXlm is a registered trademark of Globetrotter Software, Inc.

OSF/Motif is a trademark of the Open Software Foundation.

IBM is a registered trademark of International Business Machines Corporation.

MS-DOS and Windows are registered trademarks of Microsoft Corporation.

Table of Contents

1	Introduction.....	17
	The Sonnet Design Suite	17
	The Analysis Engine, <i>em</i>	21
	A Simple Outline of the Theory	22
	The Sonnet Project.....	24
	Sonnet Applications	26
	<i>Em</i> Origins.....	28
	Sonnet User's Guide Layout	28
	Describing Menu Bar Accesses	29
	Invoking Sonnet.....	29
	UNIX	29
	Windows.....	30
2	What's New in Release 10	33
	Sonnet Lite.....	34
	New Features	34
	Changes	36
3	Important Options and Concepts	39
	De-embedding the Port Discontinuity	40
	Using Multi-Frequency Caching	43
	Using Symmetry	45
	Using the Memory Save Option.....	47
	Adjusting the Subsectioning	48
	Removing Parallel Subsections	49

4	Subsectioning	51
	Tips for Selecting A Good Cell Size	52
	Cell Size Calculator	55
	Viewing the Subsections	55
	Subsectioning and Simulation Error	56
	Changing the Subsectioning of a Polygon	57
	Default Subsectioning of a Polygon	57
	X Min and Y Min with Edge Mesh Off	60
	X Min and Y Min with Edge Mesh On	62
	Using X Max and Y Max for an Individual Polygon	64
	Using the Speed/Memory Control	65
	Setting the Maximum Subsection Size Parameter	68
	Defining the Subsectioning Frequency	68
	Conformal Mesh Subsectioning	69
	Conformal Mesh Subsectioning Control	71
5	Metalization and Dielectric Loss	73
	Metalization Loss	74
	Sonnet's Loss Model	74
	Problems In Determining Metal Loss	75
	Determining Good Input Values	76
	Creating Metal Types	77
	Metal Libraries	84
	Via Loss	89
	Setting Losses for the Box Top and Bottom (Ground Plane)	89
	Dielectric Loss	90
	Dielectric Layer Parameters	91
	Dielectric Layer Loss	92
	How to Create a New Dielectric Layer	92
	Dielectric Libraries	93

6	Ports	99
	Port Types	99
	Standard Box-Wall Ports	101
	Adding Standard Ports	101
	Special Port Numbering	102
	Ports with Duplicate Numbers	102
	Ports with Negative Numbers	102
	Changing Port Numbering	103
	Port Normalizing Impedances	104
	Changing Port Impedance	106
	Defining Reference Planes/Calibration Lengths	107
	Standard Ports	107
	Auto-ground Ports	109
	Deleting Ports	109
	Ungrounded-Internal Ports	110
	Via Ports	111
	Adding Via Ports	112
	Automatic-Grounded Ports	112
	Special Considerations for Auto-Grounded Ports	113
	Adding Auto-grounded Ports	114
	Creating Auto-grounded Ports: An Example	115
7	De-embedding	119
	Enabling the De-embedding Algorithm	121
	De-embedding Port Discontinuities	122
	Box-Wall Ports	123
	Shifting Reference Planes	124
	Single Feed Line	125
	Coupled Transmission Lines	126
	De-embedding Results	127
	De-embedding Error Codes	129

8	De-embedding Guidelines	131
	Calibration Standards	131
	Defining Reference Planes	132
	De-embedding Without Reference Planes	132
	Reference Plane Length Minimums	133
	Reference Plane Lengths at Multiples of a Half-Wavelength	134
	Reference Plane Lengths Greater than One Wavelength	134
	Non-Physical S-Parameters	135
	Box Resonances	138
	Higher Order Transmission Line Modes	139
9	Adaptive Band Synthesis (ABS)	141
	ABS Resolution	142
	Q-Factor Accuracy	143
	Running an Adaptive Sweep	143
	ABS Caching Level	145
	Multiple ABS Sweeps and Subsectioning	146
	Multi-Sweep Caching Scenarios	147
	Find Minimum and Find Maximum	149
	Parameter Sweep	149
	Analysis Issues	151
	Multiple Box Resonances	151
	De-embedding	152
	Transmission Line Parameters	152
	Current Density Data	153
	Ripple in ABS S-Parameters	153
	Output Files	153
	Viewing the Adaptive Response	154
10	Parameterization and Optimization of a Geometry Project	155
	Parameters	156
	Anchored Parameters	157

Symmetrical Parameters	159
Dependent Parameters	161
Circular Dependencies	163
Reference Planes	163
Parameter Sweep	164
Optimization	166
11 Conformal Mesh	169
Introduction	169
Use Conformal Meshing for Transmission Lines, Not Patches .	171
Applying Conformal Meshing	171
Conformal Meshing Rules	173
Memory Save Option	176
Using Conformal Meshing Effectively	177
Use Conformal Meshing for Non-Manhattan Polygons	177
Boundaries Should Be Vertical or Horizontal	179
Cell Size and Processing Time	179
Current Density Viewing	180
12 Netlist Project Analysis	183
Networks	184
Netlist Project Analyses	185
Creating a Netlist	186
Adding Networks	186
Adding Network Elements	187
Adding Modeled Elements	187
Adding Data File Elements	187
Adding Project Elements	188
Adding Parameters	188
Netlist Example Files	189
Cascading S-, Y- and Z-Parameter Data Files	189
A Network File with Geometry Project	191

Inserting Modeled Elements into a Geometry	194
Using Ungrounded-Internal Ports	197
13 Circuit Subdivision	201
Introduction	201
Circuit Subdivision in Sonnet	204
Choosing Subdivision Line Placement	207
Good and Bad Placements of Subdivision Lines	209
Subdivision Line Orientation	214
Setting Up Circuit Properties	216
Setting Up the Coarse Step Size Frequency Sweep	217
Subdividing Your Circuit	218
Analyzing Your Subdivided Circuit	218
14 Microwave Office Interface	221
System Requirements	222
Installation	222
Uninstall	223
Licensing	223
Microwave Office Interface Overview	223
Selecting Sonnet as your EM Simulator	226
Opening a New EM Structure	226
Selecting Sonnet for an Existing EM Structure	228
Editing in Microwave Office	230
Selecting Microwave Office as the EM Structure Editor	231
Changing the Fill Type	232
Controlling the Subsectioning (Meshing)	236
Controlling the Analysis Frequencies	237
Computing Current Density	241
Viewing the Subsectioning	245
Executing the Analysis	247
Response Data	248

Editing in Sonnet's Project Editor	249
Working Outside Microwave Office	252
Exporting Your EM Structure to Sonnet	252
Analyzing Outside Microwave Office	253
Importing a Sonnet Project	254
Translation Issues.	255
Sonnet Features Not Available in Microwave Office.	256
Metal Types	258
Port Mapping	260
Ports on Ground Plane	261
Port Termination and Excitation.	261
Off Grid Placement.	262
Coordinate System	262
3D Viewer Scaling	262
Meshing, Current Density Plots and Far Field Plots	263
Troubleshooting	263
Unlocking Microwave Office	263
Project Editor does not open due to a licensing problem	263
Coordinate System	264
15 The DXF and GDSII Translators	265
Converting your files to Sonnet	266
Layer Mapping.	266
DXF Layer Mapping.	267
GDSII Layer Mapping	270
Using Your Layout Program to Replace the Project Editor.	272
Tweaking the Circuit	272
Specifying Units for a DXF Translation	273
Error in Processing the DXF File.	273
Capabilities and Limitations of DXF Translation	274
Error in Processing the GDSII File	276
Partial Circuit Conversion in a GDSII Import	277

16 The Agilent Interface	279
Overview of the Agilent Interface.....	280
Translation Procedure	282
Substrate parameters	282
Box Settings	283
Special layers for special circuit elements.....	284
Special features of Selected \Rightarrow Geo.....	284
New and user defined instances.....	285
The Preference File.....	286
Translator preferences	287
Troubleshooting	294
<i>Ebridge</i> menu does not appear or is corrupted	295
Error in Processing the Layout.....	296
Error While Creating Hierarchy.....	296
Port Instances Do Not Translate	296
17 Using Diagonal Fill	297
A Coupled Open-Miter with Diagonal Fill.....	299
18 Vias and 3-D Structures	301
Introduction.....	301
Restrictions on Vias	302
Creating the Vias.....	302
Via Direction	302
Via Types	303
Via Posts	305
Adding a Via to Ground.....	306
Multi-layer Vias	308
Deleting Vias	310
Via Loss	311
Via Ports	312
Simple Via Example	314
A Conical Via	314

19 Thick Metal	317
Thick Metal Type	317
Creating a Thick Metal Polygon	319
Viewing Thick Metal in the Project Editor	322
Restrictions with Thick Metal Polygons	324
Modeling an Arbitrary Cross-Section	324
Thick Metal in the Current Density Viewer	327
20 Dielectric Bricks	331
Applications of Dielectric Bricks	333
Guidelines for Using Dielectric Bricks	333
Subsectioning Dielectric Bricks	333
Using Vias Inside a Dielectric Brick	334
Air Dielectric Bricks	334
Limitations of Dielectric Bricks	334
Diagonal Fill	334
Antennas and Radiation	334
The Agilent Translator, <i>ebridge</i>	335
Dielectric Brick Concepts	335
Creating a Dielectric Brick	335
Viewing Dielectric Bricks	335
Defining Dielectric Brick Materials	337
Changing Brick Materials	338
Z-Partitioning	339
21 Antennas and Radiation	343
Background	344
Modeling Infinite Arrays	344
Modeling an Open Environment	345
Validation Example	349
Far Field Viewer	351

Analysis Limitations	352
Spherical Coordinate System	352
Normalization	356
Polarization	357
References	357
22 SPICE Model Synthesis	359
PI Spice Model	362
Using The PI Model Spice Option	362
A Simple Microwave Example	366
Topology Used for PI Model Output	367
N-Coupled Line Option	368
Broadband SPICE Model	369
Class of Problems	370
Creating a Broadband Spice Model	371
Checking the Accuracy of the Broadband Spice Model	374
Improving the Accuracy of the Broadband Spice Model	377
23 Coplanar Waveguide and Balanced Ports	379
Ungrounded Coplanar	380
Different Types of Coplanar Waveguide	381
24 Package Resonances	385
Box Resonances	386
Runtime Warning Messages	386
Observations of Simulated Results	388
A Box Resonance Example	389
The Box Resonance Estimator	391
Box Resonances – Simple Removal	393
The Capability to Ask: What if?	395

25	Viewing Tangential Electric Fields	397
26	Accuracy Benchmarking	401
	An Exact Benchmark	402
	Residual Error Evaluation	404
	Using the Error Estimates	406
27	Time Required for Analysis	407
	The “Wall”	410
	Detailed Parameter Dependencies	411
Appendix I	<i>Em</i> Command Line for Batch	415
Appendix II	Sonnet References	419

Chapter 1 Introduction

The Sonnet Design Suite

The suite of Sonnet analysis tools is shown on page 20. Using these tools, Sonnet provides an open environment to many other design and layout programs. The following is a brief description of all of the Sonnet tools. Check with your system administrator to find out if you are licensed for these products:

Project Editor

The project editor is a user-friendly graphical interface that enables you to input your circuit geometry or circuit netlist for subsequent *em* analysis. If you have purchased the DXF and/or the GDSII translator, the translator interface is found in the project editor. You also set up analysis controls for your project in the project editor.

Program module: *xgeom*

Analysis Engine	<p><i>Em</i> is the electromagnetic analysis engine. It uses a modified method of moments analysis based on Maxwell's equations to perform a true three dimensional current analysis of predominantly planar structures. <i>Em</i> computes S, Y, or Z-parameters, transmission line parameters (Z_0, Eeff, VSWR, GMax, Zin, and the Loss Factor), and SPICE equivalent lumped element networks. Additionally, it creates files for further processing by the current density viewer and the far field viewer. <i>Em</i>'s circuit netlist capability cascades the results of electromagnetic analyses with lumped elements, ideal transmission line elements and external S-parameter data.</p> <p>Program module: <i>em</i></p>
Analysis Monitor	<p>The analysis monitor allows you to observe the on-going status of analyses being run by <i>em</i> as well as creating and editing batch files to provide a queue for <i>em</i> jobs.</p> <p>Program module: <i>emstatus</i></p>
Response Viewer	<p>The response viewer is the plotting tool. This program allows you to plot your response data from <i>em</i>, as well as other simulation tools, as a Cartesian graph or a Smith chart. You may also plot the results of an equation. In addition, the response viewer may generate Spice lumped models.</p> <p>Program module: <i>emgraph</i></p>
Current Density Viewer	<p>The current density viewer is a visualization tool which acts as a post-processor to <i>em</i> providing you with an immediate qualitative view of the electromagnetic interactions occurring within your circuit. The currents may also be displayed in 3D.</p> <p>Program module: <i>emvu</i></p>
Far Field Viewer	<p>The far field viewer is the radiation pattern computation and display program. It computes the far-field radiation pattern of radiating structures (such as patch antennas) using the current density information from <i>em</i> and displays the far-field radiation patterns in one of three formats: Cartesian plot, polar plot or surface plot.</p> <p>Program module: <i>patvu</i></p>

GDSII Translator

The GDSII translator provides bidirectional translation of GDSII layout files to/from the Sonnet project editor geometry format.

Program module: *gds*

DXF Translator

The DXF translator provides bidirectional translation of DXF layout files (such as from AutoCAD) to/from the Sonnet project editor geometry format.

Program module: *dxfgeo*

Agilent Interface

The Agilent Interface provides a seamless translation capability between Sonnet and Agilent EEsof Series IV and Agilent EEsof ADS. From within the Series IV or ADS Layout package you can directly create Sonnet geometry files. *Em* simulations can be invoked and the results incorporated into your design without leaving the Series IV or ADS environment.

Program module: *ebridge*

Microwave Office Interface

The Microwave Office Interface provides a seamless incorporation of Sonnet's world class EM simulation engine, *em*, into the Microwave Office environment using Microwave Office's EM Socket. You can take advantage of Sonnet's accuracy without having to learn the Sonnet interface. Although, for advanced users who wish to take advantage of powerful advanced features not presently supported in the integrated environment, the partnership of AWR and Sonnet has simplified the process of moving EM projects between Microwave Office and Sonnet

Program Module: *sonntawr*

Cadence Virtuoso Interface

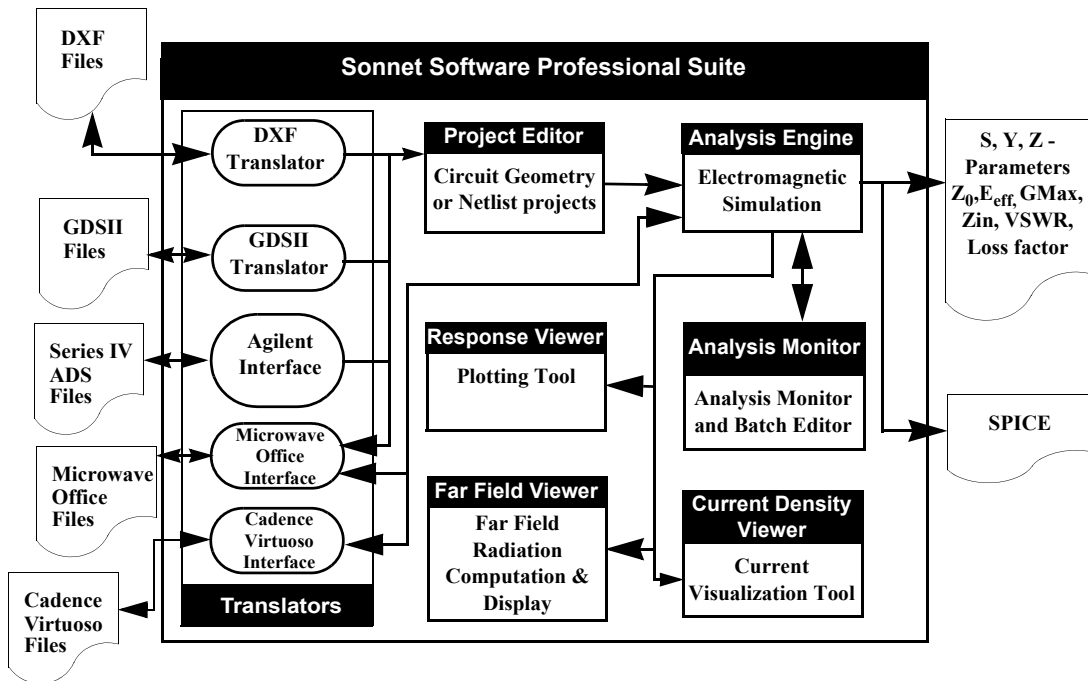
This Sonnet plug-in for the Cadence Virtuoso suite enables the RFIC designer to configure and run the EM analysis from a layout cell, extract accurate electrical models, and create a schematic symbol for Analog Artist and RFDE simulation.

Program Module: *sonntcds*

Broadband Spice Extractor

A new Broadband Spice extraction module is available that provides high-order Spice models. In order to create a Spice model which is valid across a broad band, the Sonnet broadband SPICE Extractor feature finds a rational polynomial which "fits" the S-Parameter data. This polynomial is used to generate the equivalent lumped element circuits which may be used as an input to either PSpice or

Spectre. Since the S-Parameters are fitted over a wide frequency band, the generated models can be used in circuit simulators for AC sweeps and transient simulations.



Em performs electromagnetic analysis [79, 80, 82] for arbitrary 3-D planar [54] (e.g., microstrip, coplanar, stripline, etc.) geometries, maintaining full accuracy at all frequencies. *Em* is a “full-wave” analysis in that it takes into account all possible coupling mechanisms. The analysis inherently includes dispersion, stray coupling, discontinuities, surface waves, moding, metalization loss, dielectric loss and radiation loss. In short, it is a complete electromagnetic analysis. Since *em* uses a surface meshing technique, i.e. it meshes only the surface of the circuit metalization, *em* can analyze predominately planar circuits much faster than volume meshing techniques.

The Analysis Engine, *em*

Em does a full three dimensional analysis that includes both 3-D fields and 3-D currents. This is in contrast to 2.5-D analyses which, while including full 3-D fields, allow only 2-D currents. Thus, a 2.5-D analysis does not allow vias or any other vertical current.

Em analyzes 3-D structures embedded in planar multilayered dielectric on an underlying fixed grid. For this class of circuits, *em* can use the FFT (Fast Fourier Transform) analysis technique to efficiently calculate the electromagnetic coupling on and between each dielectric surface. This provides *em* with its several orders of magnitude of speed increase over volume meshing and other non-FFT based surface meshing techniques.

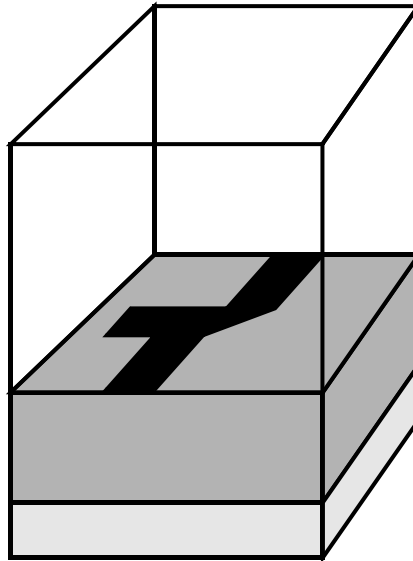
Em is a complete electromagnetic analysis; all electromagnetic effects, such as dispersion, loss, stray coupling, etc., are included. There are only two approximations used by *em*. First, the finite numerical precision inherent in digital computers. Second, *em* subdivides the metalization into small subsections made up of cells.

The cell size is an important factor in determining the accuracy. By using a smaller cell size, metal edges can be more accurately defined and the current distribution is better represented. The trade-off is increased execution time. A quantitative description of accuracy versus cell size is given in **Chapter 26, “Accuracy Benchmarking”** on page 401

The trade-off between execution time and accuracy increases the degrees of freedom available to the design engineer.

A Simple Outline of the Theory

Em performs an electromagnetic analysis of a microstrip, stripline, coplanar waveguide, or any other 3-D planar circuit by solving for the current distribution on the circuit metalization using the Method of Moments. The metalization is modeled as zero-thickness metal between dielectric layers.

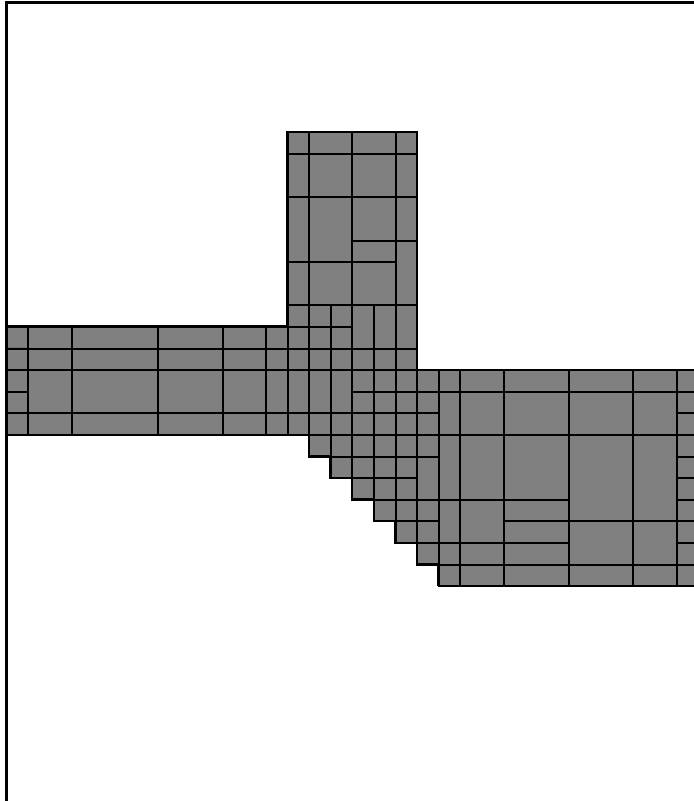


Em analyzes planar structures inside a shielding box. Port connections are usually made at the box sidewalls.

Subsectioning the Circuit

The analysis starts by subdividing the circuit metalization into small rectangular subsections. In an actual subsectioning, see page 23, small subsections are used only where needed. Otherwise, larger subsections are used since the analysis time is directly related to the number of subsections. Triangular subsections, which can be larger without sacrificing accuracy, can be used to fill in the diagonal “staircase” at the user’s discretion.

Em evaluates the electric field everywhere due to the current in a single subsection. ***Em*** then repeats the calculation for every subsection in the circuit, one at a time. In so doing, ***em*** effectively calculates the “coupling” between each possible pair of subsections in the circuit.



Em calculates the tangential electric field on all subsections, given current on one subsection. This figure shows the actual subsectioning for an example circuit.

Zero Voltage Across a Conductor

Each subsection generates an electric field everywhere on the surface of the substrate, but we know that the total tangential electric field must be zero on the surface of any lossless conductor. This is the boundary condition: no voltage allowed across a perfect conductor.

The problem is solved by assuming current on all subsections simultaneously. *Em* adjusts these currents so that the total tangential electric field, which is the sum of all the individual electric fields just calculated, goes to zero everywhere that there is a conductor. The currents that do this form the current distribution on the metalization. Once we have the currents, the S-parameters (or Y- or Z-) follow immediately.

If there is metalization loss, we modify the boundary condition. Rather than zero tangential electric field (zero voltage), we make the tangential electric field (the voltage on each subsection) proportional to the current in the subsection. The constant of proportionality is the metalization surface resistivity (in Ohms/square). In short, Ohm's Law.

The Sonnet Project

Each circuit, whether a netlist or a circuit geometry, is contained in a project. This project contains the specification of the circuit geometry or netlist, the analysis controls, and the analysis output data. What types of analysis data contained in the project depends on the types of analyses run on the project.

Each program in the Sonnet suite uses the same project. You open a new project in the project editor. The project editor supplies a graphical interface to define a circuit geometry or a netlist editor to input a netlist which is able to reference other projects.

If you have purchased a license for one of the Sonnet translators, Agilent, DXF or GDSII, you may input these files into the project editor and save the translated file as a Sonnet project.

Once the circuit is complete, the analysis controls are entered in the project editor. Then, the electromagnetic simulation is performed on your circuit using those analysis controls. Any data output from the analysis is stored in the project. This data may contain the results from multiple analyses, as long as the geometry remains consistent.

If the circuit is changed in the project editor, you are prompted to either delete the response data before saving or perform a Save As to another project name. In this way, the response data in a project is kept consistent with the geometry or netlist.

One exception to this rule is the current density data. If you generate current density data for a geometry with a parameter defined, then change the nominal value of a parameter in the circuit, you will receive an error message in the current density viewer since no current density data exists with the parameter set to the new nominal value. A change in the nominal value of a parameter is not a trigger for cleaning out the response data in a project.

The far field viewer inputs the current density data from the project and computes the far field radiation pattern. You may save these calculations to a “.pat” file, which can also be used as an input file for the far field viewer.

In addition, in your analysis setup in the project editor, you may define additional response files to be output separate from the project. The table below shows which formats are available.

Table 1 Optional Output Files

Format	Default File Extension	Descriptions
Touchstone	.s<n>p .s<mm>	Touchstone response file n = number of ports < 10. mm= number of ports > 10.
Databank	.s<n>p .s<mm>	Databank format frequency sorted response file. n = number of ports < 10. mm= number of ports > 10.
SCompact	.flp	SCompact format response file
Spreadsheet	.csv	Comma separated value data for a spreadsheet such as Excel.
Spectre	.scs	Cadence format Spice file - all models
PSpice	.lib	PSpice file - all models
Agilent MDF	.mdf	Agilent's Microwave Data Interchange Format

Sonnet Applications

Sonnet is appropriate for a wide range of 3-D planar structures. The via capability allows analysis of airbridges, wire bonds, spiral inductors, wafer probes and internal ports as well as for simple grounding.

It is appropriate to use Sonnet for:

- **Evaluation of specific discontinuities or groups of interacting**

discontinuities to assist in the design of a 3-D planar circuit. *Em* provides ultra-precise S-parameters for discontinuities allowing designers to work with confidence. *Em* can also quickly synthesize an equivalent lumped model for discontinuities. The lumped model can be used directly in circuit theory programs.

- **Design validation.** Using *em* for design validation effectively eliminates expensive design iterations (i.e., “tweak”, refabricate, etc.) of the passive, planar portion of a circuit. At present, because *em* makes no compromise in accuracy, performing an analysis of, say, an entire amplifier with a single electromagnetic analysis on a mid-range workstation or PC is difficult. However, validation of large portions of an amplifier design is reasonable.

If a circuit is designed from the start with electromagnetic analysis in mind, much larger circuits can be done. For example, the analysis works best with tightly packed, rectangular circuits, designed on a common dimension grid.

- **Microwave package evaluation.** It is important to assess how a circuit will operate in the package environment. *Em* analyzes a circuit inside a conducting box. If the box (acting as a dielectric loaded resonator) is resonant at a frequency where the circuits still have gain, poor performance results. *Em* provides an analysis of a package prior to fabrication. Resonances can then be dealt with on the computer rather than on the test bench.
- **Microstrip antennas.** The “top” of *em*’s box can be effectively removed. While radiation is outside of *em*’s primary thrust, a wide variety of microstrip antennas and radiating discontinuities can be evaluated.
- **High speed digital interconnect.** When an approximate model is not good enough, *em* can synthesize a SPICE lumped model including all delays and couplings. The lumped model is synthesized directly from electromagnetic data.

Sonnet is not appropriate for doing an initial design. Rather, the faster circuit theory simulators (which do not typically include stray coupling) should be used for the first cut. *Em* can then enhance the simulator performance by providing custom, ultra-precise discontinuity data and by validating large portions of the final circuit, including all stray interactions.

Sonnet is designed to work with your existing CAE software. Since the output data is in Touchstone or Compact format (at your discretion), **em** provides a seamless interface to your CAE tool.

***Em* Origins**

The technique used in **em** was developed at Syracuse University in 1986 by Rautio and Harrington [79, 80, 82]. It was originally developed as an extension of an analysis of planar waveguide probes [84]. The technique expresses the fields inside the box as a sum of waveguide modes and is thus closely related to the spectral domain approach.

The complete theory has been published in detail in peer reviewed journals. A full list of relevant papers is presented in Appendix II, "Sonnet References".

Sonnet User's Guide Layout

Chapter 2 details what's new in release 10 as well as major changes since release 9. Chapter 3 through Chapter 8 discuss important design considerations while using Sonnet. Many of the discussions contain examples to illustrate the point under consideration. Chapter 9 through Chapter 11 explain the details of analysis options such as Adaptive Band Synthesis (ABS), optimization and conformal mesh. Chapter 12 explains the use of the netlist editor. Chapter 13 explains the use of the circuit subdivision commands. Chapter 14 through Chapter 16 discusses the various Sonnet translators. This is followed by Chapter 17 through Chapter 27 in which advanced design topics are discussed. Appendix I discusses the use of the **em** command line for batch files. Appendix II provides a list of articles written by Sonnet authors on the theory and use of Sonnet.

Describing Menu Bar Accesses

In this manual, we describe accessing the menu bars of Sonnet programs using a "pointer" description to illustrate selecting the desired menu buttons. For instance, *File* \Rightarrow *Print* means to click on File in the menu bar and then select Print in the pull-down menu which appears.

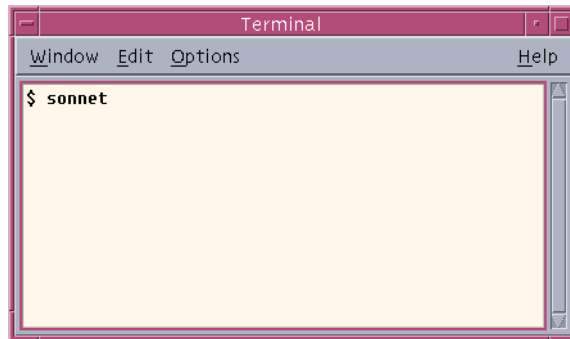
Invoking Sonnet

You use the Sonnet task bar, shown below, to access all the modules in the Sonnet *em* Suite. Opening the Sonnet task bar, for both Windows and UNIX systems, is detailed below.

UNIX

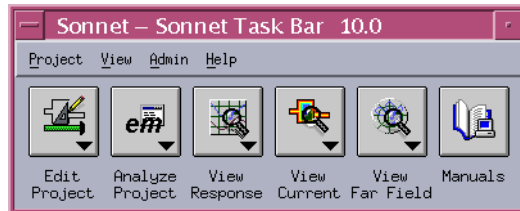
1 Open a terminal.

If you do not know how to do this, please see your system administrator.



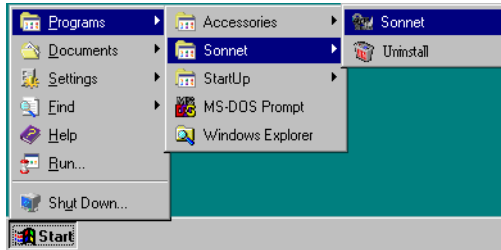
2 Enter "sonnet" at the prompt.

The Sonnet task bar appears on your display.

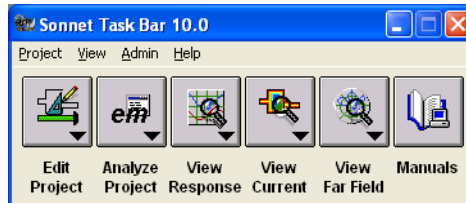


Windows

- 1 Select *Start* ⇒ *Programs* ⇒ *Sonnet* ⇒ *Sonnet* from the Windows desktop Start menu.









The Sonnet task bar appears on your display.



Once the Sonnet task bar is open, for UNIX or Windows systems, clicking on any given button opens the appropriate module. The table below shows which modules are invoked by each button.

Table 2: Sonnet Task Bar Buttons

Button	Button Name	Sonnet Program
	Edit Project	Project Editor
	Analyze Project	Analysis Monitor
	View Response	Response Viewer
	View Current	Current Density Viewer
	View Far Field	Far Field Viewer
	Manuals in PDF format	Adobe Acrobat

The translation programs for DXF and GDSII files are accessed through the project editor's File menu. For details on each program, please refer to the appropriate user's manual or online help.

Chapter 2

What's New in Release 10

This chapter summarizes new capabilities and changes in release 10 of Sonnet. If you are not yet familiar with *em*, you may want to just skim this chapter, skipping any terms that are unfamiliar. If you are an experienced user, this chapter merits detailed reading.

Sonnet User's Manuals are only updated with each full release. However, our online help is also available at our web site and starting with Release 10 will periodically be updated with new material. To access this help, go to www.sonnetsoftware.com/support and click on the "Knowledge Base" link for the most recent updates.

Sonnet Lite

If you are looking for what's new and changed in the Sonnet Lite release, please refer to the What's New topic in online help in either the Sonnet task bar or the project editor. For what's new in the full release, see the sections below.

New Features

Below is a summation of the major new features in release 10 of Sonnet. See below for more details. For changes from release 9, refer to [“Changes,” page 36](#).

Cadence Virtuoso Interface: A completely new Sonnet plug-in for the Cadence Virtuoso suite enables the RFIC designer to configure and run the EM analysis from a layout cell, extract accurate electrical models, and create a schematic symbol for Analog Artist and RFDE simulation. For more information, please refer to the **Cadence Virtuoso Interface** document included in PDF format in the Manuals interface. You may access the Manuals interface by clicking on the Manuals button on the Sonnet task bar. This is an optional feature for the Sonnet Professional Suite which must be purchased separately.

Available for SuSE Linux Platforms: This release is available for the Linux operating system SuSE Professional 9.0. It does not support the SuSE Enterprise Server or its client, SuSE Desktop 8.0. For installation instructions, please refer to the **Unix & Linux Installation Manual**.

Available on Redhat Linux platforms: This release is available for the Linux operating system Redhat Enterprise 3. It should run on Redhat 9.0 and 8.0, but has not been tested on those platforms. It will not run on Redhat 7.2 or 7.3. For installation instructions, please refer to the **Unix & Linux Installation Manual**.

NOTE:

You must purchase a Linux or UNIX license to operate Sonnet on a Linux platform. You may not use a PC license on a Linux system. See "Floating Licenses: UNIX, Linux and PC" on page 14 of the UNIX & Linux Installation Manual.

Broadband Spice Extraction: The new broadband Spice Extraction feature allows you to extract a distributed circuit model of a passive circuit which is valid over a broad band. The Broadband Spice Extractor feature generates models that can be used in Spice as a “black box” representing the broadband behavior of your circuit. This is an optional feature for the Sonnet Level2 Suites and above which must be purchased separately.

DC Point Analysis: The DC Frequency sweep analyzes your circuit at a low, non-zero frequency to supply response data at a DC point. This type of response data is necessary for some circuit analysis tools. The DC analysis frequency may be calculated automatically by *em*, or input manually by the user.

To calculate response data at a DC point, you use the DC frequency sweep which is available in the Frequency Sweep Entry dialog box in the project editor. To access this dialog box you select *Analysis* \Rightarrow *Setup* to open the Analysis Setup dialog box. Then, you select Frequency Sweep Combinations in the Analysis Setup dialog box and click on the Add button. See online help in the project editor for more details. This feature is only available as part of the Sonnet Professional Suite.

3D Viewer: This release introduces the new 3D viewer available in the project editor and the current density viewer. The viewer provides a three-dimensional view of your circuit. The view may be rotated in any direction so that the circuit may be observed at any angle. The 3D viewer is invoked by selecting *View* \Rightarrow *View 3D* from the main menu of the project editor or the current density viewer. You may animate the 3D view in the current density viewer. For more details about the 3D viewer, please refer to online help for the project editor and the current density viewer.

Graph Archives: This new feature allows you to create a graph archive file which includes all the source projects for the data used in the graph allowing you to re-create that graph on any computer by moving the graph archive file and unpacking it. For more details, select *File* \Rightarrow *Pack Graph* or *File* \Rightarrow *Unpack Graph* from the main menu of the response viewer. When the dialog box appears, click on the Help button to open online help.

Sending Projects to Support: There is a new command available in the project editor, *File* \Rightarrow *Send to Sonnet*, that provides an easy-to-use interface that allows you to send your projects to Sonnet quickly and easily. For more details, please see Send to Sonnet in online help.

Changes

Below is a summation of the major changes in release 10 of Sonnet. See below for more details. For new features in release 10, refer to [“New Features,” page 34](#)

Thick-Metal Conformal Mesh: In previous releases, conformal mesh polygons could not use the Thick Metal Model definition for loss. This limitation has been removed.

SPICE generation: Changes have been made in the way you generate Spice models and in the formats produced. You may now choose between N-Coupled Line Models, PI Models and Broadband Spice Models. Sonnet's Spice generation capability now also supports the Spectre format. You may create these files during the *em* analysis by specifying the file in the project editor or produce them post-analysis, using the new Output menu in the response viewer. For a detailed discussion of these options, please see **Chapter 22, “SPICE Model Synthesis”** on page 359.

Q-Factor Accuracy: A new advanced run option has been added to the Advanced Options dialog box in the project editor (Select *Analysis* \Rightarrow *Setup*, then click on the Advanced button in the Analysis Setup dialog box). Selecting this option forces a higher accuracy for ABS convergence by including the Q-factor of your analysis as a criterion for convergence. For more information, see **Chapter 9, “Adaptive Band Synthesis (ABS)”** on page 141.

Binary Boxes/Composite Boxes: In previous releases, using a binary box, where one or both dimensions of the substrate are 2N cells, provided a faster FFT (used during the matrix fill portion of the analysis) and thereby reduced the amount of time it took to analyze a circuit. In this release, the FFT has been improved for composite box sizes. A composite number is not a prime number, nor does it contain any large prime factors. For example, 1000 is a composite number because its largest prime factor is 5. But 998 is not a composite number because its largest

prime factor is 499. So a 998 by 998 box might take 2 to 3 times longer in the FFT calculation portion of the analysis than a 1000 by 1000 box. The FFT portion of the analysis is usually a small percentage of the total analysis time, unless you have a lot of layers or an especially large box. If either of these conditions are true, then it might be worthwhile to use a composite number in your box size.

New Output Menu in the Response Viewer: Instead of using an export command in the response viewer, you may now use the new Output menu to specify output files. You may create output files of your S, Y, and Z-parameters, generate SPICE models and output data formatted for a spreadsheet. For more details, please see online help on the Output menu.

Change in Project Editor Menu: The command *Analysis* \Rightarrow *Optional Files* command in the project editor is now *Analysis* \Rightarrow *Output Files*. You may specify output files of your S, Y, and Z-parameters, SPICE models and output data formatted for a spreadsheet. The specified output files will be generated as part of your *em* analysis. For more details, please see online help on the Output command.

Chapter 3

Important Options and Concepts

In the previous chapter, we described the basics of running Sonnet: what input files are required, how you set up and save a project, and what the output results mean. In this chapter we discuss the following techniques which are frequently used to obtain more accurate results and/or shorter analysis times:

- De-embedding the port discontinuity
- Using multi-frequency caching (MFC)
- Using symmetry
- Invoking single precision
- Adjusting the subsectioning
- Removing parallel subsections

De-embedding the Port Discontinuity

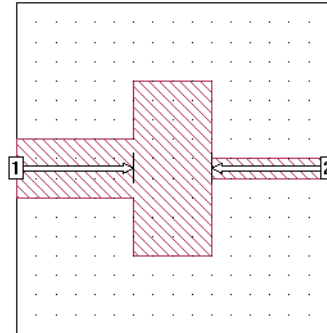
Each port in a circuit analyzed by *em* introduces a discontinuity into the analysis results. In addition, any feed transmission lines that might be present introduce phase shift, and possibly, impedance mismatch and loss. Depending upon the nature of your analysis, this may or may not be desirable. De-embedding is the process by which the port discontinuity and transmission line effects are removed from the analysis results.

The *em* de-embedding algorithm is described in detail in Chapter 7 and Chapter 8. To summarize, this algorithm performs the following analysis steps when enabled:

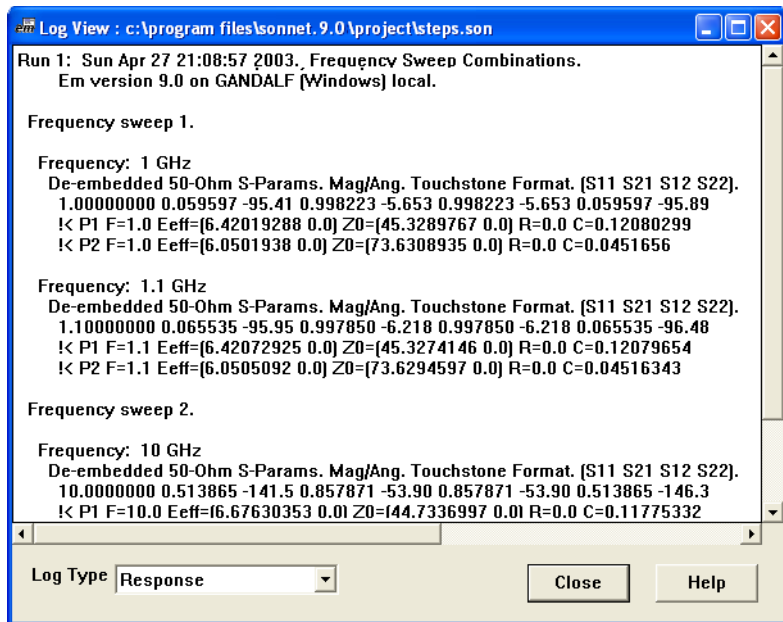
- 1 Calculates port discontinuities.
- 2 Removes effects of port discontinuities from analysis results.
- 3 Optionally shifts reference planes (removes effects of feed or port transmission lines from analysis results).
- 4 Calculates transmission line parameters E_{eff} and Z_0 .

Run *em* with de-embedding enabled whenever you do not want to include the effects of port discontinuities in your analysis results. In fact, the De-embed option is selected by default in the analysis controls. To enable de-embedding, click on

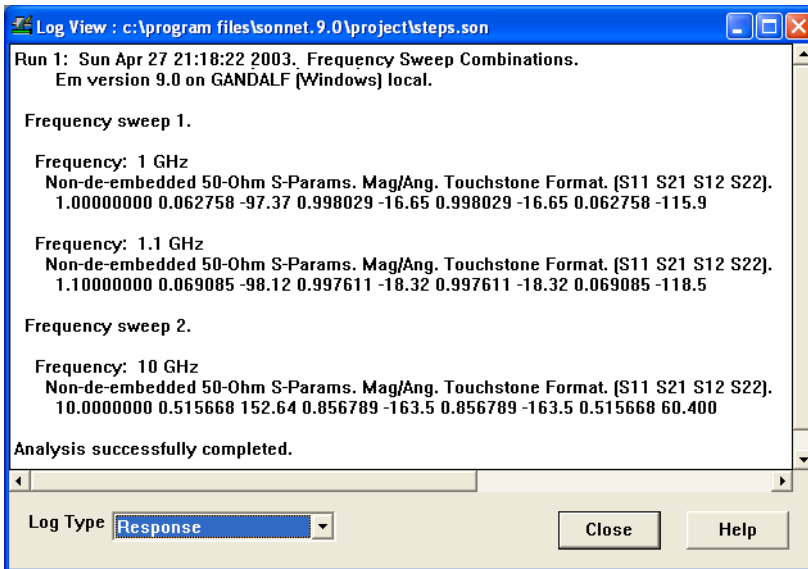
the Advanced button in the Analysis Setup dialog box in the project editor. In the Advanced options dialog box which appears on your display, click on the De-embed option checkbox.



The circuit “steps.son” is shown above. The de-embedded and non-de-embedded analysis results as they appear in the project log are shown below. Select *Project* ⇒ *View Log* ⇒ *Response Data* from the project editor’s main menu. The circuit is available in the Sonnet example files under [Steps](#). If you do not know how to obtain a Sonnet example, select *Help* ⇒ *Examples* from the menu of any Sonnet program, then click on the Instructions button.



The de-embedded results of a “steps.son” analysis shown in the project log.



The non-de-embedded results of a “steps.son” analysis shown in the project log.

Using Multi-Frequency Caching

Multi-frequency caching (MFC) is a technique that can dramatically reduce the *em* computation time when analyzing large circuits over many frequencies without any loss in accuracy. If you only need response data over a frequency band, an adaptive sweep (ABS) provides more data in much less time. See Chapter 9, “Adaptive Band Synthesis (ABS)” on page 141. However, an adaptive sweep only computes current density data or transmission line parameters (E_{eff} and Z_0) for the discrete data points but not for all the adaptive data. If you need to compute current density data or transmission line parameters at a large number of frequency points, the multi-frequency caching option can save on processing time.

This technique is enabled by checking the “Multi-Frequency Caching” run option in the Advanced Options dialog box in the project editor. Note that this checkbox is disabled if an Adaptive Sweep (ABS) is selected in the Analysis Setup dialog box.

MFC pre-computes frequency independent data and stores it on your hard disk. This data is later recalled while the simulation is performed. By storing and reusing the data in this manner, MFC significantly reduces the required matrix fill time. In standard *em* analyses, matrix fill is completely re-computed at each frequency. Thus, if you are analyzing at a large number of frequencies (greater than four), and if matrix fill time is appreciable for your circuit, enabling MFC will speed up your analysis. Since an analysis executes in either the same or less time, this option should always be used except when cache disk space is unavailable.

The MFC technique does require some time to pre-compute the frequency independent data. Because of this pre-computation time, MFC does not provide any advantage when you simulate at four or fewer frequencies. If you set up an analysis to run at four or fewer frequencies, and you check the “Multi-Frequency Caching” run option, *em* will automatically disable MFC for the simulation.

Circuits for which the matrix fill time is large will benefit the most from MFC. Characteristics of such circuits include:

- Circuits with a large number of cells in the X and Y dimensions.
- Circuits with a large number of metallization layers.
- Circuits with vias.
- Circuits with polygons using diagonal fill.
- Circuits with polygons for which XMIN/YMIN is greater than 1.

It is important to remember that MFC only reduces the matrix fill time. MFC has no affect on the matrix solve time.

MFC saves time by writing data to your hard disk and reusing that data later. Therefore, you need to have hard disk space available in order to use MFC. When *em* begins a simulation with MFC enabled, it displays the amount of disk space required, and where the data will be stored. In general, if you have disk space available, it is recommended that you always run with MFC enabled. MFC will not slow the simulation down.

If you wish to change the location where *em* will store the data, select *File* \Rightarrow *Preferences* from the project editor main menu, and enter a new disk location in the Cache Directory dialog box. You may also set a Cache Limit in the same preferences window. This limit sets the maximum amount of disk space that you

will allow **em** to use for MFC. If a particular simulation exceeds this limit, **em** will automatically disable MFC for that simulation. For details on how to set the location and limit the amount of memory, see “File - Preferences” in the project editor’s online help.

When MFC is enabled, **em** displays a message similar to the following:

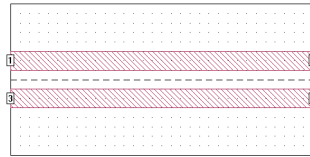
Multi-frequency caching enabled below <f_{mfc}> GHz.

This frequency is the maximum frequency at which the pre-computed data is valid and is used as a cutoff frequency. Above the cutoff frequency, it is not usually possible to compute frequency independent data. If your frequency analysis band is entirely below <f_{mfc}>, **em** runs with MFC enabled over your entire band. If your band is above <f_{mfc}>, **em** disables MFC for the entire simulation. If your band has frequencies both above and below <f_{mfc}>, **em** automatically uses MFC up to <f_{mfc}> and disables MFC for all frequencies greater than <f_{mfc}>.

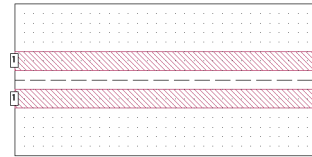
Using Symmetry

The microstrip circuit in “steps.son” (see the circuit on page 41) is symmetric about the horizontal center line. **Em** can take advantage of this, provided all ports are also on the center line or have an exact reflection. This second condition is important. If there is a port off the center line, **em** creates the port’s image on the other side of the center line and shorts the two together.

If you want to short two symmetrical ports together you should identically number the ports above and below the line of symmetry. Ports with the same number are electrically connected together. This may be useful for designers interested in the even-mode of a pair of coupled lines, as shown below.



This circuit cannot be run using symmetry.



This circuit may be run with symmetry, because the ports above and below the line of symmetry are the same number.

For the “steps.son” example file, your *em* analysis may take advantage of the symmetry setting. Open the project in the project editor and modify “steps.son” using *Circuit* \Rightarrow *Box* to open the Box Settings dialog box. When you click on the *Symmetry* checkbox, it toggles from *Off* to *On* and a dashed center line is drawn on the circuit. Visually check to make sure the circuit really is centered on the center line. If not, move the circuit until it is centered.

In electromagnetic terms, the symmetry option places a magnetic wall along the center line. In terms of performance, the number of subsections is reduced by nearly half, reducing the size of the matrix to one quarter of its original memory requirement, and decreasing matrix solve time by about a factor of eight. Thus, symmetry is a powerful tool in reducing both analysis time and hardware requirements. You will always get identical analysis results when using symmetry.

When finished, save the file under the new name “ste_sym.son”. You may want to add a comment or two indicating the modification as you store the file.

You may also obtain a copy of this file from the Sonnet examples under [Ste_sym](#). If you do not know how to obtain a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button. Be aware that this example is supplied with analysis response data so in order to see the results you will need to use the *Project* \Rightarrow *View Log* \Rightarrow *Response Data* command.

The output to the screen should be the same as that shown in the project log on page 42. If a non-zero user time was displayed in the previous example, the time should be about half that in this example. Note that the number of subsections is cut by almost half. While not particularly important here, the improvement in speed and memory use is very important for large circuits.

Using the Memory Save Option

Em normally works with double precision. In many cases, using the Memory Save option which uses a single precision matrix reduction is all that is needed. Keep in mind that the matrix elements are still calculated in double precision because of potential numerical difficulties. Once calculated, they are then truncated and stored in single precision format. To invoke a single precision matrix using the file “ste_sym.son”, do the following:

- 1 Open the “ste_sym.son” project in the project editor.
- 2 Select *Analysis* \Rightarrow *Setup* from the project editor main menu.
- 3 Click on the Memory Save checkbox in the Analysis Setup dialog box which appears.
- 4 The De-embed option is already on; therefore, click on the OK button to close the dialog box.
- 5 Click on the Analyze button on the tool bar to launch the analysis.

The results as shown in the project log are virtually identical to the results in the symmetry example.

By invoking single precision, you cut the memory required by a factor of two. This has a significant effect if your problem is large enough to cause *em* to swap out to disk. For these big problems, use of single precision can result in an order of magnitude increase in speed.

Unfortunately, compilers on certain other computers perform single precision arithmetic by first converting to double precision, performing the operation and then converting back to single precision. If such a machine is not memory swapping in double precision, there is no increase in speed in single precision. It may even result in a slightly slower analysis.

In general, use the memory saver option on large circuits. It is not needed on small circuits.

On occasion, the memory saver option can result in slightly different results. At extremely low frequencies, when the subsections size is on the order of 0.0001 wavelength or smaller, single precision may not be enough to allow the matrix to be accurately inverted. In this case, *em* issues a warning as corrupted results may be generated. If you are using the memory save option at very low frequencies and you notice non-physical results, eliminating the option and re-analyzing may remove the problem.

Adjusting the Subsectioning

The most direct way to reduce analysis time is to reduce the number of subsections. A reduction in number of subsections by half can result in one, or even two, orders of magnitude increase in speed.

The easiest way to reduce subsection count is to reduce the amount of metal to be analyzed. Any metal which is unlikely to carry significant current should be removed. Distant ground plane regions in coplanar waveguide, for example. Also, long, unneeded lengths of connecting transmission line provide no additional information in an analysis and should be removed.

A second way to reduce subsection count is to simply use a larger cell size. It can be surprising how little a larger cell size changes results on many circuits. Sonnet provides a Cell Size calculator in the project editor which calculates the optimal cell size based on the critical dimensions of your circuit which you specify. The Cell Size Calculator is accessed in the Box Settings dialog box which is opened when you select *Circuit* \Rightarrow *Box* from the project editor main menu.

Be careful not to reject viable options because of the dimensions being used. For example, if you want to analyze a line 7.0 microns wide, the obvious potential cell dimensions are 1 micron (for a seven cell wide line) and 7 microns (for a one cell wide line). There are other options, also. For example, the cell size can be specified to be $7/3 = 2.33333$ microns resulting in a three cell wide line.

Next, use your engineering judgement in determining the dimensions of a structure. If a 7.1 micron wide line is called for, and you use a 7 micron cell size, the residual error introduced by making the line 7 microns wide is probably less than the manufacturing tolerances.

However, there are circuits which have very fine geometries combined with very large overall structures. The fine geometry determines the cell size, requiring the overall circuit to use a large number of cells.

To address this problem, you may specify a minimum and maximum subsection size, in terms of cells, for each polygon. You use the parameters X Min, Y Min, X Max and Y Max to do this. Chapter 4, "Subsectioning," discusses this in more detail.

A third way to reduce the number of subsections is to cut the circuit into pieces and analyze each piece separately. For example, instead of analyzing an entire bandpass filter, analyze half of it. Then, if both halves are identical, just use the **em** circuit network capability, or a circuit theory program to cascade the half filter analysis with itself.

Removing Parallel Subsections

A transmission line connected to a port is subsectioned with both x and y directed subsections included in order to determine both x and y directed currents. However, in most cases, a majority of the current flows in the longitudinal direction and there is very little current flowing in the transverse direction.

For example, in a horizontal, x directed transmission line, most of the current is flowing in the x direction. There is very little transverse, y directed, current flowing. Without parallel subsection removal, all of the y directed subsections are still included in the analysis, though their current is almost zero. This wastes subsections, memory and time.

To remove the transverse subsections, in the project editor, select *Circuits* \Rightarrow *Parallel Subsections*. Then select a side and either specify a length in the text entry box or use the mouse to specify the length for which parallel subsections are to be excluded. The transverse subsections are parallel to the side of the box being referenced.

This option should be used carefully. Here are a few rules:

- Never remove subsections for lengths greater than $1/4$ to $1/2$ wavelength.
- Remove subsections only from transmission lines with very small transverse currents such as feed lines. For example, do not remove subsections from discontinuities, lines that change in width or regions where structures approach one another.
- Since this option affects all drawing levels in the project editor, check all levels for possible problems.

Chapter 4 Subsectioning

The Sonnet subsectioning is based on a uniform mesh indicated by the small dots in the project editor screen. The small dots are placed at the corners of a “cell”. One or more cells are automatically combined together to create subsections. Cells may be square or rectangular (any aspect ratio), but must be the same over your entire circuit. The cell size is specified in the project editor in the Box Settings dialog box which is opened by selecting *Circuit* \Rightarrow *Box*. The analysis solves for the current on each subsection. Since multiple cells are combined together into a single subsection, the number of subsections is usually considerably smaller than the number of cells. This is important because the analysis solves an $N \times N$ matrix where N is the number of subsections. A small reduction in the value of N results in a large reduction in analysis time and memory.

Care must be taken in combining the cells into subsections so that accuracy is not sacrificed. **Em** automatically places small subsections in critical areas where current density is changing rapidly, but allows larger subsections in less critical areas, where current density is smooth or changing slowly.

However, in some cases you may wish to modify the automatic algorithm because you want a faster, less accurate solution, or a slower, more accurate solution, than is provided by the automatic algorithm. Also, in some cases, you may have knowledge about your circuit that the software does not. For example, you may know that there is very little current on a certain area of your metal. Or you may have chosen a small cell size because you have a small dimension in your circuit, but do not need the accuracy of a small cell size in larger structures within your circuit. In these cases, you can change the method by which **em** combines cells into subsections.

This chapter explains how **em** combines cells into subsections and how you can control this process to obtain an analysis time or the level of accuracy you require. There is also a discussion of selecting the cell size and how that may affect the **em** analysis.

Conformal Mesh is a special case of subsectioning used to model polygons with long diagonal or curved edges. For more information on subsectioning when using conformal mesh, see "Conformal Mesh Subsectioning" on page 69.

Tips for Selecting A Good Cell Size

As you know, **em** subdivides the circuit into subsections which are made up of "cells," the building block in the project editor. The following discussion describes how to select a cell size. You may also use the Sonnet Cell Size Calculator which allows you to enter important dimensions to calculate the most efficient cell size which provides the required accuracy. To access the Cell Size calculator, click on the Cell Size Calculator button in the Box Settings dialog box, which is invoked when you select *Circuit* \Rightarrow *Box* from the project editor menu.



TIP

Select a cell size that is smaller than 1/20 of a wavelength.

Before calculating a cell size, it is important to calculate the wavelength at your highest frequency of analysis. An exact number is not important. If you know the approximate effective dielectric constant of your circuit, use this in the wavelength calculation; otherwise, use the highest dielectric constant in your structure.

Most circuits require that your cell size be smaller than $1/20$ of a wavelength. Larger cell sizes usually result in unacceptable errors due to incorrect modeling of the distributed effects across the cell. Cell sizes smaller than $\lambda/20$ may increase the accuracy slightly but usually increases the total number of subsections, which increases the analysis time and memory requirements.



TIP

When possible, round off dimensions of your circuit so that they have a larger common multiple.

Since your circuit geometry is snapped to the nearest cell, you must find a cell size such that all of the dimensions of the circuit are a multiple of this cell size. For example, if your circuit has dimensions of 30 microns, 40 microns and 60 microns, possible cell sizes are 10 microns, 5 microns, 2.5 microns, 2 microns, etc. Large cell sizes result in more efficient analyses, so choosing 10 microns is probably best.



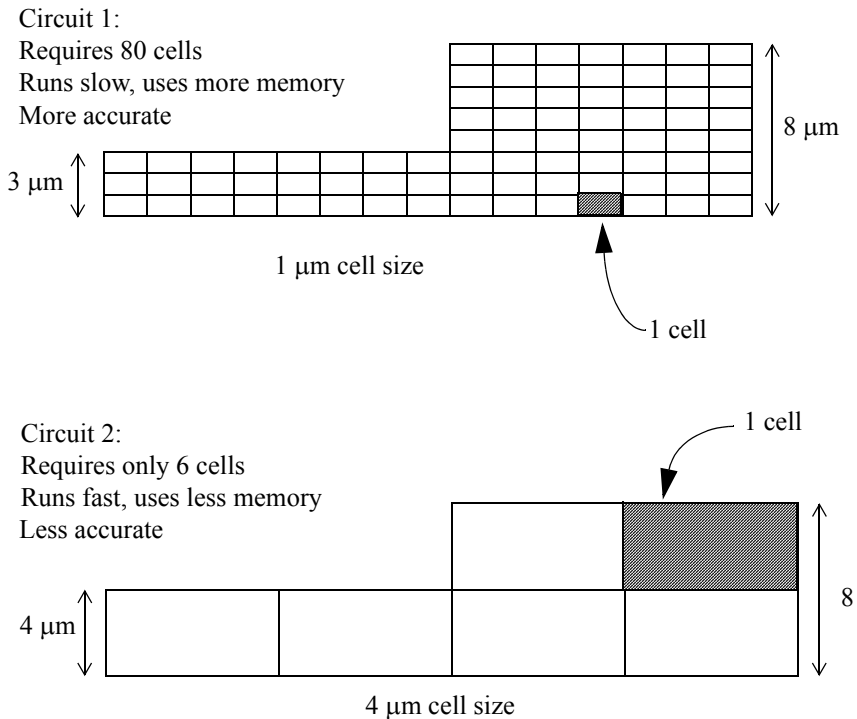
TIP

Calculate the X cell size and the Y cell size independently.

The X cell size and Y cell size do not have to be the same number. Calculate the X cell size based on just your dimensions in the X direction, and your Y cell size based on just your dimensions in the Y direction.

Sonnet User's Guide

For example, if you have a spiral inductor with widths of 3 microns and spacings of 8 microns, modify the 3 microns to 4 microns. You may now use 2 cells instead of 8, speeding up the analysis by several orders of magnitude with little impact on circuit performance. This concept is illustrated below.



Circuit 1 takes more time and memory to analyze than circuit 2 even though they have approximately the same amount of metal. This is because the dimensions in circuit 2 are divisible by 4, so a 4 μm cell size may be used. Circuit 1 requires a 1 μm cell size. Think about the sensitivity of your circuit to these dimensions and

your fabrication tolerances. If your circuit is not sensitive to a 1 micron change or can be made with only a +/- 1 micron tolerance, you can easily round off the 3 micron dimension in circuit 1 to the 4 micron dimension in circuit 2.

Cell Size Calculator

Sonnet also provides a cell size calculator which you may use to calculate the optimal cell size based on your critical circuit parameters. You access the Cell Size Calculator in the Box Settings dialog box (*Circuit* \Rightarrow *Box*). Using the Cell Size Calculator is detailed in Chapter 4, “Determining Cell Size” of the **Sonnet Tutorial**. A detailed discussion of all the entries in the cell size calculator may be found in online help.

Viewing the Subsections

You can see the subsections used by Sonnet by following the instructions below. Be aware that your dielectric layers must be defined and at least one port must be added to your circuit before you may use the Estimate Memory command.

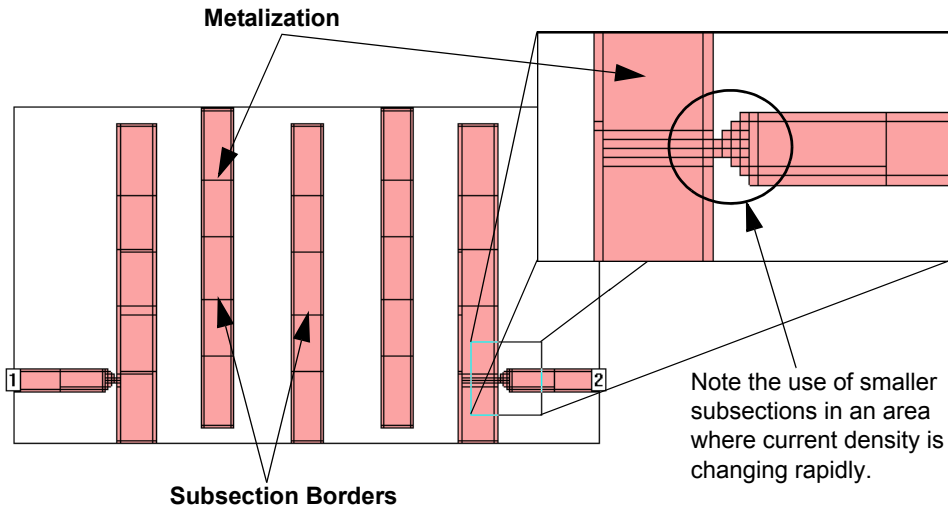
To view the subsectioning, do the following:

- 1 **From the project editor, select Analysis \Rightarrow Estimate Memory.**

This calculates the amount of memory required for your analysis.

2 Click the View Subsections button.

A picture of your circuit will appear. The metal will show up as red, and the subsection borders will show up as black lines as shown in the illustration below:



Subsectioning and Simulation Error

As discussed above, Sonnet uses a fixed resolution grid and discretely meshes a given metallization pattern based on that underlying grid. The edges of metal patterns in a design do not necessarily have to be aligned to the grid, even though Sonnet only simulates metal fill which is on the grid. Off grid metalization may be over or under filled depending on the degree of misalignment between grid and metal pattern. While misalignment gives the user visual feedback of one potential error source in a Sonnet simulation, it is important to keep in mind that every planar Method of Moments (MoM) simulation contains multiple sources of error.

Unlike Sonnet, most EM software vendors speak very little about error sources (see "Accuracy Benchmarking," page 401). The fact that Sonnet shows misalignment between the desired metal pattern and simulation grid does not necessarily imply that Sonnet simulations will be any less accurate than

competitive simulators that mesh using infinite resolution. With all simulation packages the user should investigate every potential error source (which will vary depending on the MoM technique used) and ensure a good converged data set is achieved.

Changing the Subsectioning of a Polygon

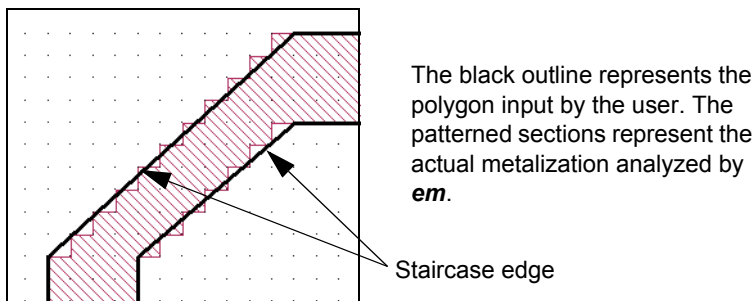
Sonnet allows you to control how cells are combined into subsections for each polygon. This is done using the parameters “X Min”, “Y Min”, “X Max” and “Y Max”. These parameters may be changed for each polygon, allowing you to have coarser resolution for some polygons and finer resolution for others. See “Modify - Metal Properties” in the project editor’s online help for information on how to change these parameters.

Before discussing how to make use of these parameters, we need to first understand *em*’s automatic subsectioning for a polygon when the parameters are set to their default settings.

Default Subsectioning of a Polygon

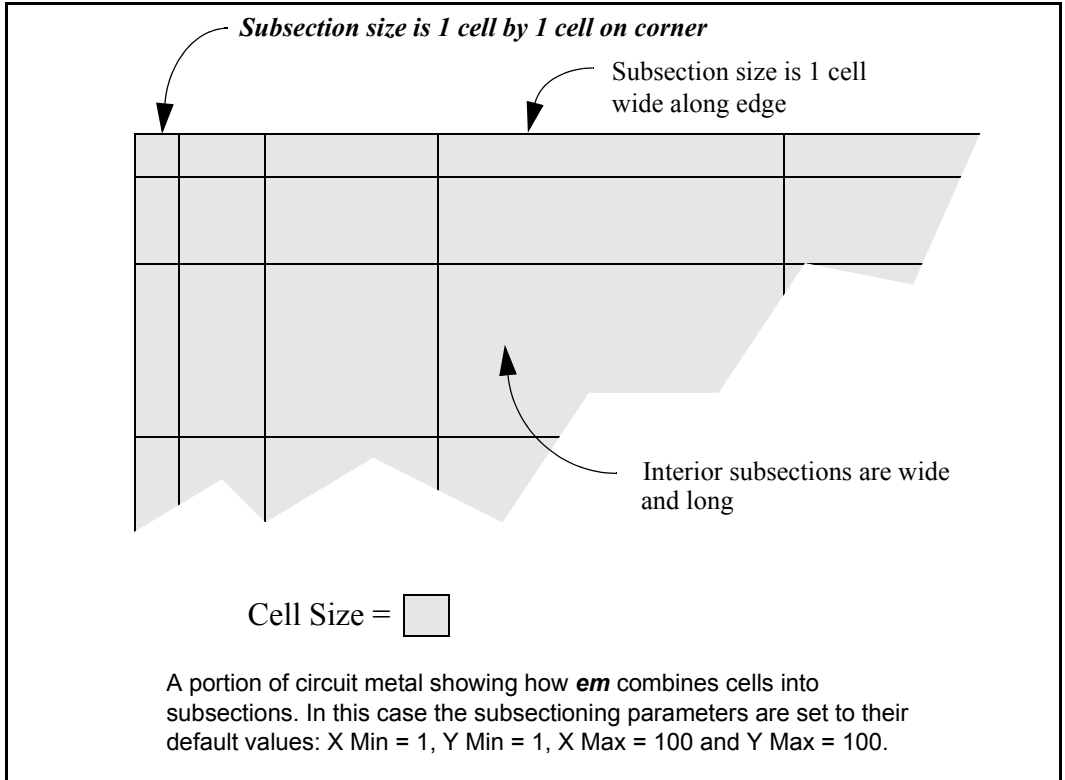
By default, Sonnet fills a polygon with “staircase” subsections. Other, more advanced fill types (diagonal and conformal) are covered in other chapters of this manual. For diagonal subsections, see **Chapter 17, “Using Diagonal Fill”** on page 297. For conformal mesh, see **Chapter 11, “Conformal Mesh”** on page 169. This chapter deals exclusively with staircase subsections.

This fill type is referred to as staircase because when using small rectangular subsections to approximate a diagonal edge, the actual metalization takes on the appearance of a staircase, as in the example shown below.



The default values for the subsectioning parameters are X Min = 1, Y Min = 1, X Max = 100 and Y Max = 100. These numbers specify the smallest and largest allowed dimensions of the subsections in a polygon. With X Min = 1, the smallest subsection in the X dimension is one cell. With X Max = 100, subsections are not allowed to go over 100 cells in length.

The illustration below shows how these default subsectioning parameters are used. Notice in the corner, the subsection size is just one cell. The current density changes most rapidly here, thus, the smallest possible subsection size is used.



As we go away from the corner, along the edge, the subsections become longer. For example, the next subsection is two cells long, the next one is four cells long, etc. If the edge is long enough, the subsection length increases until it reaches X Max (100) cells, or the maximum subsection size parameter, whichever comes first, and then remains at that length until it gets close to another corner, discontinuity, etc.

Notice, however, that no matter how long the edge subsection is, it is always one cell wide. This is because the current density changes very rapidly as we move from the edge toward the interior of the metal (this is called the edge singularity).

In order to allow an accurate representation of the very high edge current, the edge subsections are allowed to be only one cell wide. However, the current density becomes smooth as we approach the interior of the metal. Thus, wider subsections are allowed there. As before, the width goes from one cell to two cells, then four, etc.



TIP

If two polygons butt up against each other or have a small overlap, the modeling of the edge singularity will require a larger number of subsections at the boundary between the two polygons. Using the Merge command (*Edit* \Rightarrow *Merge Polygons*) to join the two polygons into one will reduce the number of required subsections and speed up your analysis.

Conversely, if you have an area of your circuit at which you desire greater accuracy, using the Divide (*Edit* \Rightarrow *Divide Polygons*) command at the point of interest to create two polygons, forces the analysis to use smaller subsections in order to model the edge singularities.

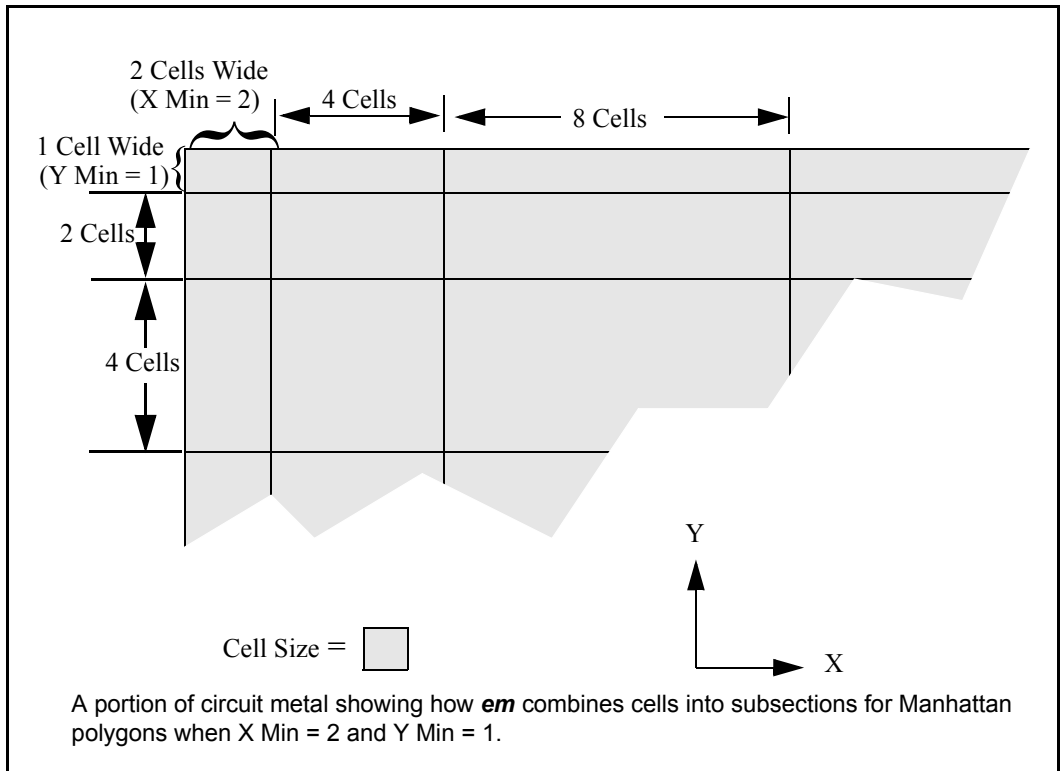
X Min and Y Min with Edge Mesh Off

Having the edge mesh option “on” is the default state for Sonnet projects; however, examining the case where edge mesh is “off” first makes understanding the concept easier. This part of the discussion only applies to Manhattan polygons, which is a polygon that has no diagonal edges. Turning edge mesh off for non-manhattan polygons has no effect.

On occasion, you may wish to change the default subsectioning for a given polygon. You can do this using the subsectioning parameters X Min, Y Min, X Max and Y Max.

For Manhattan polygons with edge mesh off, X Min and Y Min set the size of the edge subsections. By default, X Min and Y Min are 1. This means the edge subsections are 1 cell wide. When X Min is set to 2, the subsections along vertical edges are now 2 cells wide in the X direction (see the figure on page 61). This

reduces the number of subsections and reduces the matrix size for a faster analysis. However, accuracy may also be reduced due to the coarser modeling of the current density near the structure edge or a discontinuity.



If X min or Y min are greater than your polygon size, **em** uses subsections as large as possible to fill the polygon.

NOTE:

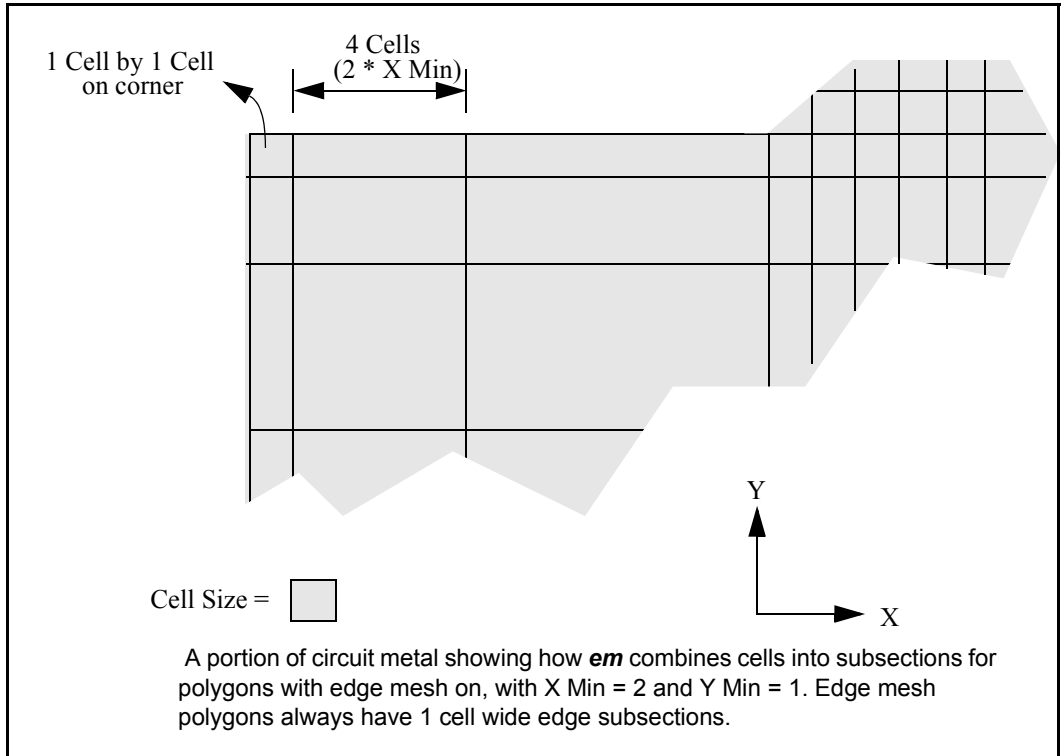
The subsection parameters, X Min, Y Min, X Max and Y Max are specified in *cells* (not mils, mm, microns, etc.). For example, X Min = 5 means that the minimum subsection size is 5 cells.

Although the X Min and Y Min parameters are very useful options, it is not a substitute for using a larger cell size. For example, a circuit with a cell size of 10 microns by 10 microns with X Min = 1 and Y Min = 1 runs faster than the same circuit with a cell size of 5 microns by 5 microns with X Min = 2 and Y Min = 2. Even though the total number of subsections for each circuit may be the same, *em* must spend extra time calculating the value for each subsection for the circuit with the smaller cell size.

X Min and Y Min with Edge Mesh On

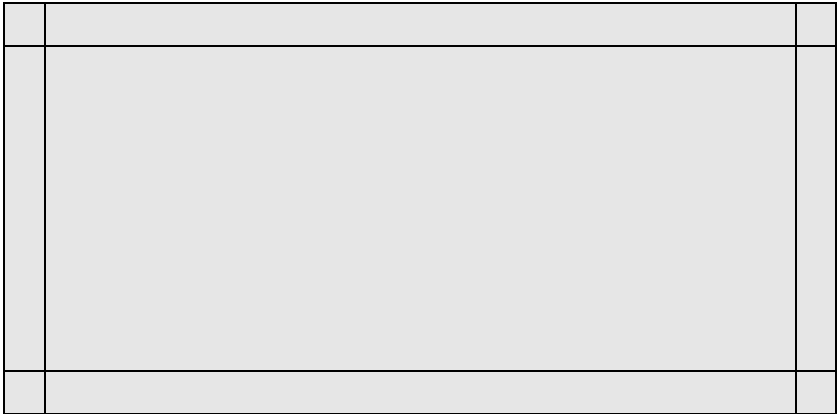
Having the edge mesh option “on” is the default state for Sonnet projects because it provides a more accurate analysis. Having edge mesh “on” for a polygon changes how the subsections on the very edge are handled. Starting from the left side of the previous example with edge mesh off, the subsections were 2 cells, 4

cells and 8 cells wide. With edge mesh on, the subsections for the same polygon would be 1 cell, 4 cells, and 8 cells as shown in the illustration below. Notice only the outermost edge is affected.



As mentioned in the previous section, the edge mesh setting only affects Manhattan polygons (i.e. those with no diagonal or curved edges). Edge mesh is always “on” for non-Manhattan polygons, regardless of the edge-mesh setting for that polygon.

When used in conjunction with large X Min or Y Min values, the edge mesh option can be very useful in reducing the number of subsections but still maintaining the edge singularity, as shown in a simple example below. This is very often a good compromise between accuracy and speed.



In the case pictured above, X Min and Y Min are set to be very large, and the frequency is low enough so that the Max. Subsection size parameter corresponds to a subsection size that is larger than the polygon.

Using X Max and Y Max for an Individual Polygon

You may control the maximum subsection size of individual polygons by using the X Max and Y Max parameters. For example, if X Max and Y Max are decreased to 1, then all subsections will be one cell. This results in a much larger number of subsections and a very large matrix which are the cause of increased analysis time. Thus, this should be done only on small circuits where extremely high accuracy is required or you need a really smooth current density plot.

NOTE:

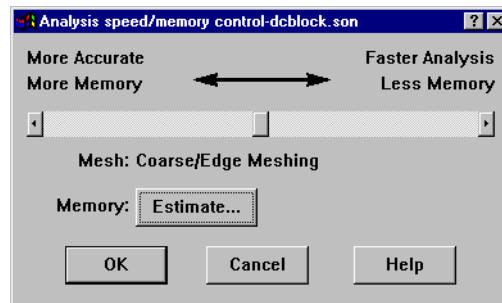
If the maximum subsection size specified by X Max or Y Max is larger than the size calculated by the Max. Subsection Size parameter, the Max. Subsection Size parameter takes priority. The Max. Subsection Size is discussed "Setting the Maximum Subsection Size Parameter" on page 68.

Using the Speed/Memory Control

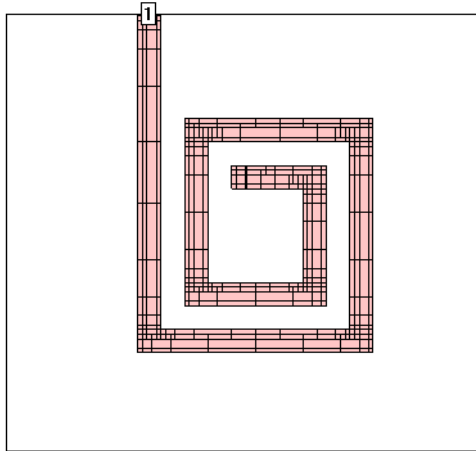
The Speed/Memory Control allows you to control the memory usage for an analysis by controlling the subsectioning of your circuit. The high memory settings produce a more accurate answer and usually increase processing time. Conversely, low memory settings run faster but do not yield as accurate an answer.

To access the Speed/Memory Control, follow the instructions below.

- 1 Select *Analysis* \Rightarrow *Setup* from the project editor main menu.
- 2 In the Analysis Setup dialog box which appears, click on the Speed/Memory button.
- 3 In the Analysis Speed/Memory Control dialog box which appears, select the desired setting.

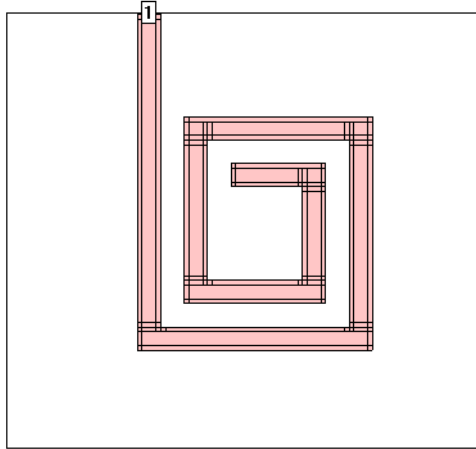


There are three settings for the Speed/Memory Control: Fine/Edge Meshing, Coarse/Edge Mesh, Coarse/No Edge Meshing. Fine/Edge Meshing is the default setting and is described in "Default Subsectioning of a Polygon" on page 57. An example is shown below. Again, note that this setting provides the most accurate answer but demands the highest memory and processing time.

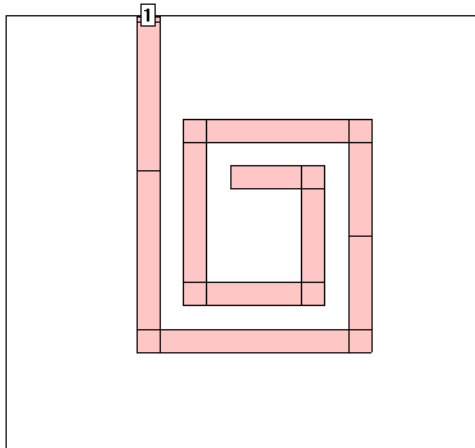


The second option is Coarse/Edge Mesh. This setting is often a good compromise between speed and accuracy. When this setting is used, the Xmin and Ymin of all polygons are set to a large number - typically, the value of 50 is used - and edge

mesh is on. Shown below is a typical circuit with this setting. Notice the edges of the polygons have small subsections, but the inner portions of the polygons have very large subsections because of the large Xmin and Ymin.



The last option is Coarse/No Edge Meshing. For this setting, all polygons are set to a large Xmin/Ymin and edge mesh is set to “off.” This yields the fastest analysis, but is also the least accurate. Shown below is the subsectioning of a typical circuit using this option.



Setting the Maximum Subsection Size Parameter

The parameter Max. Subsection Size allows the specification of a maximum subsection size, in terms of subsections per wavelength, where the wavelength is approximated at the beginning of the analysis. The highest analysis frequency is used in the calculation of the wavelength. This value is a global setting and is applied to the subsectioning of all polygons in your circuit.

The default of 20 subsections/ λ is fine for most work. This means that the maximum size of a subsection is 18 degrees at the highest frequency of analysis. Increasing this number decreases the maximum subsection size until the limit of 1 subsection = 1 cell is reached.

You might want to increase this parameter for a more accurate solution. For example, changing it from 20 to 40 decreases the size of the largest subsections by a factor of 2, resulting in a more accurate (but slower) solution. Keep in mind that using smaller subsections in non-critical areas may have very little effect on the accuracy of the analysis while increasing analysis time.

Another reason for using this parameter is when you require extremely smooth current distributions using for the current density viewer. With the default value of 20, large interior subsections may make the current distribution look “choppy”. Setting this value to a large number forces all subsections to be only 1 cell wide, providing smooth current distribution. Again, analysis time is impacted significantly.

The Max. Subsection Size parameter is specified in the Advanced Subsectioning Controls which are opened by selecting *Analysis* \Rightarrow *Advanced Subsectioning* from the project editor main menu.

Defining the Subsectioning Frequency

The subsectioning parameter, Max. Subsection Size, applies to the subsectioning of all polygons in your circuit, and is defined as subsections per wavelength. Normally, the highest analysis frequency is used to determine the wavelength. However, this may be changed by using the Subsectioning Frequency options in

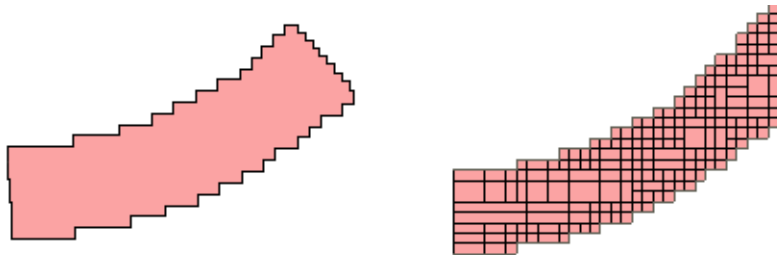
the Advanced Subsectioning Control dialog box in the project editor. This dialog box is opened by selecting *Analysis* \Rightarrow *Advanced Subsectioning* from the project editor main window. For details on what options are available to define the subsectioning frequency, click on the Help button in the Advanced Subsectioning Control dialog box.

The frequency defined by the selected option is now used to determine the maximum subsection size instead of the highest frequency of analysis. Thus, the same subsectioning can be used for several analyses which differ in the highest frequency being analyzed.

Conformal Mesh Subsectioning

Conformal meshing is a technique which can dramatically reduce the memory and time required for analysis of a circuit with diagonal or curved polygon edges. For a detailed discussion of conformal mesh and its rules of use, please refer to “Conformal Mesh,” page 169. Only the effect of conformal mesh on subsectioning is discussed in this chapter.

This technique groups together strings of cells following diagonal and curved metal contours to form long subsections along those contours. Whereas staircase fill results in numerous small X- and Y-directed subsections, conformal mesh results in a few long conformal subsections. The illustration below shows the actual metalization of a conformal section in close up alongside the same section using staircase fill.



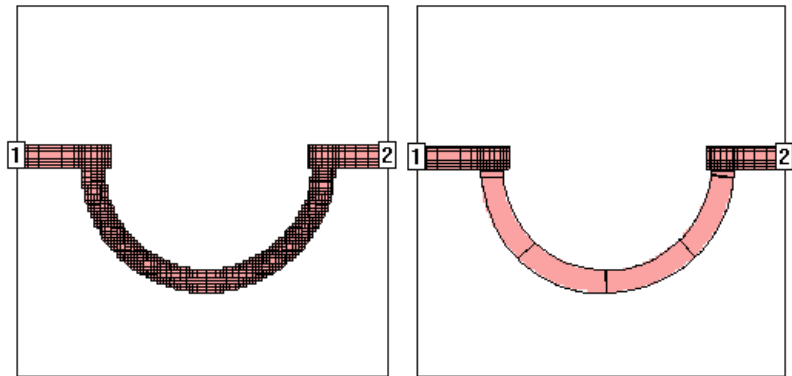
Conformal section

Staircase Fill

Conformal sections, like standard subsections, are comprised of cells, so that the actual metalization still shows a "jagged" edge when the polygon has a smooth edge. However, the sections can be much larger due to conformal meshing. These larger sections yield faster processing times with lower memory requirements for your analysis.

Standard subsectioning requires a lot of subsections to model the correct current distribution across the width of the line. Conformal subsections have this distribution built into the subsection. Sonnet conformal meshing automatically includes the high edge current in each conformal section. This patented Sonnet capability is unique. (U.S. Patent No. 6,163,762 issued December 19, 2000.)

An example of a circuit using both standard subsectioning and conformal mesh is shown below. The circuit shown at the left is displayed using standard subsectioning (staircase fill). Conformal meshing for the curved part of the circuit is shown on the right. Note that for the curved part of the geometry, conformal mesh uses substantially fewer subsections than the number used in the standard subsectioning.



Conformal Mesh Subsectioning Control

When you apply conformal mesh to a polygon, it is possible to limit the maximum length of a conformal section in order to provide a more accurate simulation. The default length of a conformal section is 1/20 of the wavelength at the subsectioning frequency. For more information on the subsectioning frequency, see “Defining the Subsectioning Frequency,” page 68.

To set the maximum length for a conformal section, do the following:

1 Select the desired polygon(s).

The selected polygons are highlighted.

2 Select *Modify* ⇒ *Metal Properties*

This opens the Metalization Properties dialog box.

3 Click on the Maximum Length checkbox in the Conformal Mesh Subsectioning Controls section of the dialog box.

This will enable the Length text entry box to the right. Note that this checkbox is only enabled when Conformal is chosen as the Fill Type.

4 Enter the desired Maximum Length in the text entry box.

Click on the OK button to close the dialog box and apply the changes.

For a more detailed discussion of Conformal Mesh, please refer to **Chapter 11, “Conformal Mesh”** on page 169 For a tutorial on Conformal Mesh, see **Chapter 4, “Conformal Mesh Tutorial”** on page 63 in the **Sonnet Supplemental Tutorials**.

Chapter 5 Metalization and Dielectric Loss

This chapter is composed of two parts: metalization loss and dielectric layer loss. For information on dielectric brick loss, see **Chapter 20, “Dielectric Bricks”** on page 331. Both the theoretical aspect of how Sonnet models loss and the practical how to's of assigning loss in your circuit are covered, including the use of metal and dielectric material libraries. The discussion of metalization loss begins below. For the discussion of dielectric loss, see "Dielectric Layer Parameters" on page 91.

There is also a paper available by the president and founder of Sonnet Software, Dr. James Rautio which contains a detailed discussion of metal losses. You may find this paper at www.sonnetsoftware.com/support/publications.asp.

Metalization Loss

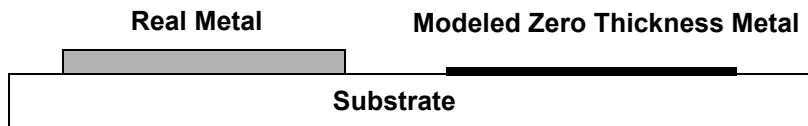
Metalization loss is specified in the project editor in the Metal Types dialog box which is opened by selecting *Circuit* \Rightarrow *Metal Types*. Losses may be assigned to circuit metal, top cover and ground plane. Sidewalls are always assumed to be perfect conductors.

A common misconception is that only one type of metalization is allowed on any given level. In fact, different metalizations (i.e., different losses) can be mixed together on any and all levels. For example, it is possible to have a thin film resistor next to a gold trace on the same level.

Sonnet allows you to use pre-defined metals, such as gold and copper, using the global library. The global library allows you to define your own metal types as well. There is also a local metal library which can be created for an individual or to share between users.

Sonnet's Loss Model

The Sonnet model of metal loss uses the concept of surface impedance, measured in Ohms/sq. This concept allows planar EM Simulators, such as Sonnet *em*, to model real 3-dimensional metal in two dimensions.



If you are unfamiliar with this concept, please refer to any classic textbook such as **Fields and Waves in Communication Electronics** by Simon Ramo, John R. Whinnery and Theodore Van Duzer, John Wiley & Sons, New York, 1965.

It is important to note that this technique models the loss of the true 3-dimensional metal fairly accurately, but does not model any change in field distribution due to the metal thickness. This approximation is valid if the metal thickness is small with respect to the width of the line, the separation between lines, and the

thickness of the dielectric. If the true 3-dimensional affect of the metal is important, then you should consider using the Thick Metal Model metal type as discussed in **Chapter 19, “Thick Metal”** on page 317.

Some electromagnetic analyses use a “perturbational” approach for loss. This means that they assume the current flowing everywhere is the same as the lossless case. This approximation works for low loss metals (good conductors). However for thin film resistors (high loss), the lossless current is not the same as the lossy current and a perturbational approach fails. *Em's* loss analysis is not perturbational. It works just as well for a 100 Ohms/square resistor as it does for a 0.004 Ohms/square good conductor. The Sonnet loss analysis also properly models the transition between electrically thin (low frequency) and electrically thick (high frequency) conductors. See reference [18] in "Sonnet References" on page 419 for a detailed description of the theory used by Sonnet. See reference [85] listed on page 426 for the equations actually used in the Sonnet model.

Another aspect of loss is that the surface impedance of a good conductor has an imaginary part which is equal to the real part. This reactive surface impedance is physically due to the increased surface inductance caused by the current being confined closer to the surface of the conductor. This surface reactance is included in the Sonnet loss model. The effect is small, but potentially significant in certain cases.

Keep in mind that a circuit running with lossless metal and dielectrics requires about one-half the amount of memory and runs about twice as fast. Therefore, the simplest approximation is to run a lossless simulation. This can be quite useful in the initial design phase.

Problems In Determining Metal Loss

Sonnet's loss model is very accurate if accurate values are used. In practice, however, there are many aspects of metal loss that cannot easily be accounted for. For example, surface roughness, metal purity, metal porosity, etc. cannot easily be measured and included in an all-encompassing loss model. In addition, most software programs, Sonnet included, do not allow you to enter all of the parameters that determine metal loss. Many users like to use the ideal values as a

starting point and add a little of their own “real-world” loss. But how much should be added to the ideal models? This question is not easily answered, but is addressed in the next section.

An additional loss problem exists with planar EM analysis tools such as Sonnet. The problem stems from the fact that planar EM tools treat the metal conductor as zero thickness. This means that there is no difference between the top of the conductor and the bottom of the conductor. In some circuits, stripline for example, the current is symmetrical with half of the current flowing on the top of the conductor and half flowing on the bottom of the conductor. The zero thickness model works well in these cases.

In other circuits, such as microstrip, the current can be unequally distributed, resulting in higher loss than the equivalent stripline circuit. If you know the ratio between the top and the bottom currents, you can obtain a better loss model. All planar solvers must either estimate this value in order to calculate metal loss, or the information must be input by the user. For this class of circuits, it is difficult for the user to know an exact value of the current ratio without obtaining measured data on the circuit. For these cases, assuming all the current flows on one side of the conductor gives a “worst case” loss result. This tends to compensate for the “best case” loss caused by ignoring the other aspects of loss (metal porosity, etc.) mentioned earlier.

Determining Good Input Values

The best method to determine proper loss values is to build and measure a simple structure of the desired metalization. The structure should be sensitive enough to loss so that measurement errors do not significantly affect the results. Then analyze the same structure on Sonnet and adjust the loss values until the calculated loss matches the measured loss. This may take several iterations before success, but then you can use these values for similar circuits. You are now effectively using measured values for the loss parameters.

Creating Metal Types

To assign loss to a polygon in Sonnet, you first define a metal type by inputting its loss parameters and then assign that metal type to the polygon drawn in your circuit. The previous section(s) described how to determine values for the Sonnet loss model. This section provides instructions for creating a metal type and discusses the loss models used in the Metal Editor dialog box.

The Metal Editor dialog box allows you to enter a loss definition for the metal type. There are five different methods for entering loss: Normal, Resistor, Rd./Ref, General and Sense. The different loss models are discussed below followed with a procedure for entering new metal types. The discussions assume simple, single conductor microstrip and stripline geometries where mentioned.

You do not need to read the details of each loss model. Instead, you should make yourself familiar with the loss models you are likely to use. Most users only need concern themselves with “Normal.” All the methods use the same loss model in Sonnet; which model you choose depends on the parameters you know about your real metal type.

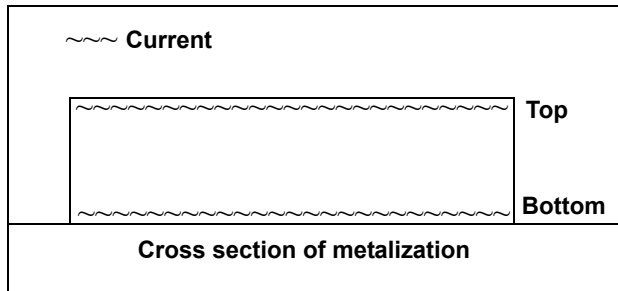
Normal

For the Normal metal type, you determine the loss using the bulk conductivity, the metal thickness and the current ratio.

NOTE:

Metal thickness is used only in calculating loss; it does not change the physical thickness of metalization in your circuit. The metalization in your circuit is still modeled as zero-thickness.

Sonnet models your circuit using zero thickness metal, but your real circuit possesses a metal thickness. At higher frequencies current flows in a thin skin around the edge of the metal, as pictured below. The current ratio is the ratio of the current flowing on the top of the metal to the current flowing on the bottom of the metal.



There are no sides in a zero thickness model; therefore when translating from these parameters, the current which flows on the sides is ignored. In some circuits, stripline for example, the current is symmetrical with half of the current flowing on the top of the metal and half flowing on the bottom of the metal. In this instance, the current ratio is 1. If you had twice as much current flowing on the top as on the bottom, the current ratio is 2. It is difficult to know an exact value of the current ratio without obtaining measured data on your circuit. All planar solvers must estimate this value in order to calculate metal loss; this particular model in Sonnet allows you to enter the value. Note that reciprocal values have the same effect; i.e., 0.5 results in the same loss as 2.0.

Resistor

To define a metal which you wish to use as a resistor, enter the DC resistance in ohms/sq in the Rdc text entry box which appears when you select this metal type. To use this loss model, the resistivity should be constant over the frequency range of interest.

Rdc/Rrf

This method allows you to enter two values: R_{DC} and R_{RF} . The first parameter, R_{DC} , determines loss at low frequency (where the conductor is much thinner than the skin depth). Surprisingly, other electromagnetic analyses often predict zero loss at low frequency because they assume R_{DC} is zero.

The second parameter is the skin effect coefficient, R_{RF} . *Em* multiplies this number by the square root of the frequency (in Hertz) to yield the Ohms/square value at high frequency.

The equations for R_{DC} and R_{RF} are:

$$R_{DC} = 1/(\sigma t)$$

$$R_{RF} = \text{Skin effect coefficient} = \sqrt{(\pi\mu)/\sigma}$$

where σ is the bulk conductivity in mhos/meter, t is the metalization thickness in meters, and $\mu = 4\pi \times 10^{-7}$ H/m. Typical values for R_{DC} and R_{RF} are 0.004 and $3e-7$. If you start getting very strange loss results, check R_{RF} , paying special attention to the exponent.

Em also properly models the transition between electrically thin (low frequency) and electrically thick (high frequency) conductors. The transition frequency from R_{DC} to R_{RF} is the square of R_{DC}/R_{RF} . At this frequency, and a relatively narrow band around it, both coefficients are important. If the skin effect coefficient (R_{RF}) is set to 0.0, then the value of R_{DC} is used over all frequencies. This is the usual case for resistors.

The above equation for R_{RF} assumes that all of the current travels on just one side of the conductor. This is a good approximation for some microstrip circuits. However, if the current really travels on both sides, this gives a pessimistic value for the loss. The equation should be modified for other structures. Stripline, for example, has current of equal amplitude on both the top and bottom of the conductor. In this case, you should divide the R_{RF} value by two, while maintaining R_{DC} .

As an example, the conductivity (σ) for copper is $5.8\text{E}+7$ Mhos/m, giving $R_{DC} = 0.006$ Ohms/square ($t = 3$ μm) and a microstrip $R_{RF} = 2.6\text{E}-7$. In reality, the bulk conductivity of copper, or any other given metal, may not equal the laboratory value, so the figures as calculated above are likely to be lower than actual results. The table below provides calculated results of commonly used metals using the equations above.

Properties of Commonly Used Metals

Metal	σ (S/M)	RDC (Ω/sq) $t = 1 \mu\text{M}$	RDC (Ω/sq) $t = 1 \text{ mil}$	RRF ($\Omega\text{Hz}^{-1/2}/\text{sq}$) "Skin Effect" (microstrip)
Aluminum	3.72e7	0.027	1.1e-3	3.3e-7
Brass	1.57e7	0.070	2.5e-3	5.0e-7
Copper	5.80e7	0.017	6.8e-4	2.6e-7
Gold	4.09e7	0.024	9.6e-4	3.1e-7
Nichrome	1.00e6	1.000	3.9e-2	2.0e-6
Silver	6.17e7	0.016	6.4e-4	2.5e-7
Tantalum	6.45e6	0.155	6.1e-3	7.8e-7
Tin	8.70e6	0.115	4.5e-3	6.7e-7

General

This loss model includes the metalization resistivity described above in R_{dc}/R_{rf} . The General loss definition also includes the metalization reactance, composed of the surface reactance, X_{dc} and the kinetic inductance, L_s .

Surface reactance, X_{dc} , is specified, in Ohms/square. **Em** uses the same reactance at all frequencies.

Until recently, the only surface resistivities of practical interest were pure real, i.e., pure loss. With the growing application of superconductors in high frequency work, surface reactance reaches significant levels. A superconductive effect known as “kinetic inductance” slows the velocity of the electrons with no loss of energy. This can be modeled as a surface inductance.

The effect of surface inductance is to make ϵ_{eff} larger, or the velocity of propagation slower. For normal conductors, ϵ_{eff} can never be larger than ϵ_{rel} . In a superconductor, this is no longer true. This unusual effect becomes significant for very thin substrates.

Surface inductance, L_s , is specified in the project editor in the Metal Types dialog box accessed by selecting *Circuits* \Rightarrow *Metal Types*. This parameter takes into account the surface reactance at higher frequencies.

There are three recommended approaches to obtaining a value for L_s . A first order approximation is to assume the metal is a perfect conductor.

$$R_{DC} = 0 \qquad R_{rf} = 0 \qquad L_s = 0$$

This model works well for moderate frequencies (less than 150 GHz) and moderate circuit dimensions which are much greater than the London depth of penetration.

The second approach is a model which is still valid at moderate frequencies, but includes effects due to kinetic inductance. The kinetic inductance is a function of temperature and can be approximated in the following manner:

$$R_{DC} = 0 \qquad R_{rf} = 0 \qquad L_s = \mu_0 \lambda_L(T)$$

where

$$\mu_0 = 4\pi(10^{-7}) \text{ H/m}$$

$$\lambda_L(T) = \lambda_0 / (\sqrt{1 - (T/T_c)^4}) \quad \text{London depth of penetration at temp.}$$

$$\lambda_0 = \text{London depth at } T = 0 \text{ meters}$$

$$T_c = \text{Critical (Transition) Temperature in degrees Kelvin}$$

The third model should be used to account for high frequency effects or effects due to small circuit dimensions. In these cases, the surface resistances proportionality to ω^2 begins to dominate and the following model is suggested. The resistivity is a function of frequency-squared, and Sonnet presently does not have a method to do this. Therefore, if you are analyzing over a broad band, you need to have a separate project for each frequency, using the following equations.¹

$$\begin{aligned} R_{rf} &= 0 \\ R_{DC} &= \frac{1}{2} \omega^2 \mu_0^2 (\lambda_L(T))^3 \sigma_N (\eta_n / (\eta_n + \eta_s)) \\ L_s &= \mu_0 \lambda_L(T) \end{aligned}$$

where

$$\omega = 2\pi f \text{ radians/sec}$$

$$\sigma_N = \text{Conductivity of the superconductor in its normal state (Mhos/m}^3\text{)}$$

$$\eta_n = \text{Normal state carrier density (1/m}^3\text{)}$$

$$\eta_s = \text{Superconducting state carrier density (1/m}^3\text{)}$$

$$\mu_0 \text{ and } \lambda_L(T) \text{ are as defined above}$$

1. Shen, Z. Y., "High-Temperature Superconducting Microwave Circuits," Boston, 1994, Artech House.

The Surface Impedance, Z_s , for superconductors is modeled in Sonnet using the following equation:

$$Z_s = R_{DC} + j \cdot (\omega L_s + X_{DC}) \quad \text{for } |L_s| > 0.0$$

Sense

Em solves for the current distribution; however, on occasion, you may want to view the fields, not the current. You do this with what is called a “sense metal”. The sense metal is a rectangular patch of conductor placed where you want to see the tangential electric field. (You cannot view the normal direction of the field with Sonnet.)

For further discussion of sense metal, see Chapter 25, “Viewing Tangential Electric Fields,” on page 397.

Thick Metal

For thick metal, you input two parameters: the bulk conductivity and the metal thickness. The loss is calculated in the same manner as for the Normal metal type, except that the thickness in this case represents a physical thickness which eliminates the need to enter the current ratio.

For a detailed discussion of thick metal, see Chapter 19, “Thick Metal” on page 324.

How to Create a Metal Type

To create a metal type, do the following:

- 1 Select Circuit \Rightarrow Metal Types from the main menu of the project editor.**

The Metal Types dialog box appears on your display.

- 2 Click on the Add button in the Metal Types dialog box.**

The Metal Editor dialog box appears on your display.

3 Select the desired loss model from the Type drop down list.

Most users should select Normal. The text entry boxes are updated with the inputs for the type selected.

4 Enter the loss parameters in the appropriate text entry boxes.

For a detailed discussion of these parameters, see the loss model discussions in the previous sections.

5 Enter the desired name for the metal type in the Name text entry box.

This name appears in the Metal Types dialog box and is used to identify the metal in your circuit. This step is optional.

6 Use the up and down arrows to select a pattern for the metal type.

Any polygons of this metal type that appear in your circuit are displayed with this pattern to identify the metal type.

7 Click on the OK button to close the Metal Editor and add the new metal type.

The new Metal Type appears in the Metal Types dialog box.

8 Click on the OK button in the Metal Types dialog box to close it and apply the changes.

This completes entering a new metal types.

Metal Libraries

There are two types of metal libraries available in Sonnet: global and local. The metal libraries contain standard definitions for metal types which may be used in your projects. There is no real difference between the global and local library. The names refer to how they are used. The global library would usually be used as a group wide library of standard metals for a group of designers.

There is a default global library supplied by Sonnet which contains definitions for some common metals. The default location for the global library is in <Sonnet Directory>/data/library where <Sonnet Directory> is the directory in which your Sonnet software is installed. You may choose to use this location or can save this library in another location.

The local library would usually be used as the user's own library of metal definitions. This library may be stored in a location of the user's choice. You use the Metal Editor dialog box to add, edit and delete entries to these libraries.

Using a Metal from a Metal Library

To use a metal type from the metal library in your circuit, do the following:

1 Select Circuit ⇒ Metal Types from the main menu of the project editor.

The Metal Types dialog box is opened on your display. This dialog box allows you to add, edit and delete metal types for your project.

2 Click on the Add button to add a metal type to your project.

This opens the Metal Editor dialog box which allows you to define a metal type or input a metal type from your library.



TIP

Clicking on the Library button in the Metal Types dialog box allows you edit the contents of your metal libraries but does not allow you to add metal types from the library to your project. This must be done in the Metal Editor dialog box.

3 Click on the Select Metal from Library button in the Metal Editor dialog box.

The Metal Library dialog box appears on your display.

- 4 Select My Library or Global Library by clicking on the appropriate radio button in the Metal Library dialog box.**

If you wish to select a new library file, click on the Specify button to the right of the radio button which opens a browse window which allows you to select a file. For instructions on setting up a library, see “Setting Up a Metal Library,” page 88.

Once a library is selected, a list of metal types in the library should appear in the dialog box.

- 5 Click on the desired metal type's entry to select it.**

The entry is highlighted in blue.

- 6 Click on OK in the Metal Library dialog box.**

This selects the metal type and closes the dialog box. The Metal Editor text entry boxes now contain the name and loss parameters for the library metal type you selected. Please note that any changes to these parameters does not affect the parameters of the metal library, only of the metal type you are creating.

- 7 If desired, you may change the display pattern for this metal type by clicking on the Up or Down arrows in the Pattern field.**

If the default field assigned to the metal is acceptable to you, you may skip this step.

- 8 Click on the OK button in the Metal Editor dialog box.**

The Metal Editor dialog box is closed and an entry for the metal type just selected from the library appears in the list in the Metal Types dialog box.

- 9 Click on the OK button in the Metal Types dialog box to close the box and apply the changes.**

This completes adding a metal from the metal libraries.

Editing a Metal Library

You use the Metal Library dialog box to add, edit or delete entries from your metal library. To edit a metal library, do the following:

1 Select Circuit \Rightarrow Metal Types from the main menu of the project editor.

The Metal Types dialog box is opened on your display. This dialog box allows you to add, edit and delete metal types for your project.

2 Click on the Library button to edit your metal libraries.

This opens the Metal Library dialog box which allows you to create metal libraries and to edit them.

3 Select My Library or Global Library by clicking on the appropriate radio button in the Metal Library dialog box.

If you wish to select a new library file, click on the Specify button to the right of the radio button browse for the desired file. For instructions on setting up a library, see “Setting Up a Metal Library,” page 88.

Once a library is selected, a list of metal types in the library should appear in the dialog box.

4 To add a metal type, click on the Add button.

The Metal Editor dialog box is opened.

5 Enter the desired name and loss parameters in the Metal Editor dialog box, then click on OK to add a new metal to the library.

This metal type is now available in the library to be used in any future projects. If you wish to use the metal type in your present project, you must add the library metal as a metal type to your project.

The thickness parameter is optional when defining a metal type in your library but is a required field if you add the metal type to your project to be used in your circuit.

6 To edit an existing metal, click on its entry to select it, then click on the Edit button.

The Metal Editor dialog box is opened with the present name and parameters for the metal type.

- 7 Make any desired changes, then click on the OK button.**

The Metal Editor dialog box is closed and any changes are updated in the metal type's entry line.

- 8 To delete a metal type from your metal library, click on the Metal Type to select it and click on the Delete button in the Metal Library dialog box.**

This deletes the metal type from the library. Note that if the metal type has been added to the project and is in use, deleting the metal from the library has no effect on the metal type being used in the project. The metal type will no longer be available in the library for other projects but will still exist in the present project.

This completes editing the metal library.

Setting Up a Metal Library

To create a new metal library, global or local, do the following:

- 1 Select Circuit ⇒ Metal Types from the main menu of the project editor.**

The Metal Types dialog box is opened on your display. This dialog box allows you to add, edit and delete metal types for your project.

- 2 Click on the Library button to edit your metal libraries.**

This opens the Metal Library dialog box which allows you to create metal libraries and to edit them.

- 3 Select the type of library you wish to create by clicking on the My Library or Global Library radio button in the Metal Library dialog box.**

Once a library is selected, a list of metal types in the library should appear in the dialog box.

- 4 To create a new library, click on the Specify button to the right of the radio button for the library you selected.**

A browse window appears on your display.

- 5 Use the browse window to specify the location and filename for the new metal library file.**

The default name for a new metal library is metal-library.txt. If you wish to name the library file something else, edit the filename entry.

- 6 Once the desired location is specified, click on the OK button in the browse window.**

The browse window is closed and a new metal library file with no entries is created at the location you just specified. Use the rest of the edit controls in the Metal Libraries dialog box to create entries for this metal library. For details on editing a metal library, see “Editing a Metal Library,” page 86.

Via Loss

The loss of a via is accomplished in much the same way as a metal polygon. The only difference is that a via is a 3D object and a polygon is only two-dimensional. Therefore, the Sonnet via loss model is only an approximation. In general, edge-vias use the loss of the polygon they are attached to, and via-polygons have their own loss properties. To assign loss to a via polygon, select the via-polygon, and choose *Modify* \Rightarrow *Via Properties*. In the Via Properties dialog box, select a metal type in the Via Loss drop list for the via polygon. For extremely precise loss analysis of vias, you should use the same measurement approach as discussed earlier for polygon loss.

Setting Losses for the Box Top and Bottom (Ground Plane)

You set the loss for the box top or bottom by assigning a metal type to the box top or bottom. The box top and bottom use the same metal types which are used for the metalization in your circuit, i.e., the polygons and via polygons. In addition, there are two special metal types available in Sonnet for the Top and Bottom metals: Free Space and Waveguide Load. See Chapter 21, “Antennas and Radiation” for a discussion of how and why you would use these types.

To assign a metal type to the Box Top or Bottom metal, select *Circuit* \Rightarrow *Box Settings* from the project editor main menu. In the Box Settings dialog box which appears on your display, select the desired metal type for the top metal from the Top Metal drop list and the desired metal type for the bottom metal from the Bottom Metal drop list.

Dielectric Loss

The dielectric loss calculations in Sonnet are virtually exact, given the substrate really has a frequency independent conductivity, and/or loss tangent. Our web site has a lossy conductivity benchmark you can perform on any electromagnetic solver (or measurement system). See Benchmarking on the Products section of our web site, www.sonnetsoftware.com.

The dielectric loss is calculated in Sonnet at the beginning stages of the analysis. The method Sonnet uses starts with the calculation of a sum of waveguide modes. The exact solution requires an infinite sum of modes, but Sonnet truncates this sum to some reasonable value (the truncation has never been a source of error). So, for each mode, if there is a lossy dielectric, the calculation involves complex numbers instead of just real numbers. This is NOT a discretized function - it is a continuous function. Therefore, the dielectric loss calculation can be thought of as exact (only limited to the precision of the machine).

A more reasonable source of error is in the assumption that the conductivity is constant with frequency. All real dielectrics have frequency-dependant loss (some smaller than others). Sonnet supplies you with two parameters (Loss Tan and Diel Cond) to control this frequency dependency. The equation Sonnet uses to calculate the TOTAL loss is given in "Dielectric Layer Loss," page 92. There are some dielectrics with more complicated frequency dependencies, but this equation works for most dielectrics. Of course, this requires that you know the frequency dependency of your dielectric. If you have a method of measuring the loss as a function of frequency (or published data which you can trust), and if it is constant over your range of frequencies, then dielectric loss is probably not a source of error. Be careful, however, of published loss data. Verify that the data is valid over your frequency range.

Dielectric Layer Parameters

You can set the dielectric constant and loss of a dielectric layer by changing the following parameters in the project editor by selecting *Circuits* \Rightarrow *Dielectric Layers*, then clicking on the Above, Below or Edit button in the Dielectric Layers dialog box. This opens the Dielectric Editor dialog box which allows you to edit the parameters below.

- **Erel:** The relative dielectric constant (ϵ_r). The ratio (ϵ/ϵ_0), where ϵ is the real part of the permittivity of the dielectric layer material, and ϵ_0 is the permittivity of free space. The ratio is dimensionless.
- **Dielectric Loss Tan:** The dielectric loss tangent. The ratio (ϵ''/ϵ'), where $\epsilon = \epsilon' - j\epsilon''$, and ϵ is the complex permittivity of the dielectric layer material. The ratio is dimensionless.
- **Diel. Cond:** The dielectric conductivity, σ , where σ is the bulk conductivity in siemens per meter.
- **Mrel:** The relative magnetic permeability (μ_r) of the dielectric layer material.
- **Magnetic Loss Tan:** The magnetic loss tangent of the dielectric layer material.

One last parameter that may be specified for a dielectric layer is the Z Partitioning. This value may be changed in the Z Partitions dialog box which is opened when you click on the Z Parts button in the Dielectric Layers dialog box.

- **Z-Partitions:** The z-partitioning parameter is the setting which controls the number of partitions which the dielectric layer is divided into in the z direction, for the dielectric layer. While this parameter is specified in the dielectric layer window, it only has an effect on the dielectric bricks on that layer. Changing this value for a particular layer will have absolutely no affect on the analysis if there are no bricks on the layer. If there are multiple bricks on the layer, the Z subsectioning for all of those bricks will be identical.

The more partitions (better resolution) used in the Z-dimension, the more accurate the analysis; however, analysis time and memory requirements also increase dramatically.

Dielectric Layer Loss

Em uses the above parameters to calculate the total effective $\tan\delta$ for the dielectric material as follows:

$$\tan\delta = (\text{Loss Tan}) + \frac{(\text{Diel Cond})}{\omega(\text{Erel})\epsilon_0}$$

where, ω is the radian frequency ($\omega = 2\pi f$, where f is frequency in hertz). Note that $\tan\delta$ has both a frequency-dependent term and a frequency-independent term.

The above equation for $\tan\delta$ can also be expressed in terms of conductivity as follows:

$$\text{Total Effective Cond} = (\text{Loss Tan})\omega(\text{Erel})\epsilon_0 + (\text{Diel Cond})$$

Both equations are equivalent. Each describes how *em* uses the input dielectric parameters to compute loss in the dielectric material.

See “Circuit \Rightarrow Dielectric Layers” in the project editor’s online help for information on setting these parameters.

How to Create a New Dielectric Layer

To create a new dielectric layer, do the following:

- 1 **Select Circuit \Rightarrow Dielectric Layers from the main menu of the project editor.**

The Dielectric Layers dialog box appears on your display.

- 2 **Click on one of the Add buttons, either Above or Below, in the Dielectric Layers dialog box.**

The Dielectric Editor dialog box appears on your display.

3 Enter the dielectric parameters in the appropriate text entry boxes.

For a detailed discussion of these parameters, see the loss model discussions in “Dielectric Layer Parameters,” page 91.

4 Enter the desired name for the dielectric material in the Name text entry box.

This optional name appears in the Dielectric Layers dialog box and is used to identify the dielectric material used in the layer in your circuit.

5 Enter the desired thickness for the dielectric layer in the Thickness text entry box.

This value may also be edited in the Dielectric Layers dialog box.

6 Click on the OK button to close the Dielectric Editor and add the new dielectric layer.

The new layer appears in the Dielectric Layers dialog box.

7 Click on the OK button in the Dielectric Layers dialog box to close it and apply the changes.

This completes entering a new dielectric layer.

Dielectric Libraries

The dielectric libraries contain standard definitions of dielectric materials which may be used for your dielectric layers. There are two types of dielectric libraries available in Sonnet: global and local. There is no real difference between the two libraries. The names refer to the way in which they are used.

NOTE:

The Dielectric Libraries materials may not be used for dielectric bricks but only for dielectric layers.

The global library would usually be used as a group wide library of standard dielectrics for a group of designers. There is a default global library supplied by Sonnet which contains definitions for dielectric materials. The default location

for the global library is in <Sonnet Directory>/data/library where <Sonnet Directory> is the directory in which your Sonnet software is installed. You may choose to use this location or can save this library in another location.

The local library would usually be used as the user's own library of dielectric material definitions. This library may be stored in a location of the user's choice. You use the Dielectric Editor dialog box to add, edit and delete entries to these libraries.

Using a Dielectric Material from a Dielectric Library

To use a dielectric material from the metal library in your circuit, do the following:

1 Select Circuit ⇒ Dielectric Layers from the main menu of the project editor.

The Dielectric Layers dialog box is opened on your display. This dialog box allows you to add, edit and delete dielectric layers for your project.

2 Click on an existing dielectric layer to select it.

There are always two default dielectric layers in a new geometry project.

3 Click on the Above or Below button to add a dielectric layer to your project.

This opens the Dielectric Editor dialog box which allows you to define a dielectric layer or input a dielectric material from your library to use for the dielectric layer.

You could also have clicked on the Edit button which would open the presently selected layer in the Dielectric Editor dialog box.

The dielectric layer will be added above or below the presently selected dielectric layer.



TIP

Clicking on the Library button in the Dielectric Layers dialog box allows you edit the contents of your dielectric libraries but does not allow you to add dielectric materials from the library to your project. This must be done in the Dielectric Editor dialog box.

- 4 Click on the Select Dielectric from Library button in the Metal Editor dialog box.**

The Dielectric Library dialog box appears on your display.

- 5 Select the My Library or Global library by clicking on the appropriate radio button in the Dielectric Library dialog box.**

If you wish to select a new library file, click on the Specify button to the right of the radio button which opens a browse window which allows you to select a file. For instructions on setting up a library, see “Setting Up a Dielectric Library,” page 97.

Once a library is selected, a list of dielectric materials in the library should appear in the dialog box.

- 6 Click on the desired dielectric material to select it.**

The entry is highlighted.

- 7 Click on OK in the Dielectric Library dialog box.**

This selects the dielectric material and closes the dialog box. The Dielectric Editor text entry boxes now contain the material name and loss parameters from the library dielectric you selected.

- 8 Enter the desired thickness for the dielectric layer in the Thickness text entry.**

You may also edit the thickness of the dielectric layer in the Dielectric Layers dialog box.

- 9 Enter the desired number of copies in the Number of Copies text entry box.**

You may enter multiple dielectric layers at one time by using this field. Each dielectric layer will be of the thickness specified here and use the same loss parameters.

- 10 Click on the OK button in the Dielectric Editor dialog box.**

The Dielectric Editor dialog box is closed and an entry for the dielectric layer just entered appears in the list in the Dielectric Layers dialog box.

- 11 Click on the OK button in the Dielectric Layers dialog box to close the box and apply the changes.**

This completes adding a dielectric layer using a dielectric material from the library.

Editing a Dielectric Library

You use the Dielectric Library dialog box to add, edit or delete entries from your dielectric library. To edit a dielectric library, do the following:

- 1 Select Circuit ⇒ Dielectric Layers from the main menu of the project editor.**

The Dielectric Layers dialog box is opened on your display. This dialog box allows you to add, edit and delete dielectric material for your project.

- 2 Click on the Library button to edit your dielectric libraries.**

This opens the Dielectric Library dialog box which allows you to create dielectric libraries and to edit them.

- 3 Select My Library or Global library by clicking on the appropriate radio button in the Dielectric Library dialog box.**

If you wish to select a new library file, click on the Specify button to the right of the radio button which opens a browse window which allows you to select a file. For instructions on setting up a library, see “Setting Up a Dielectric Library,” page 97.

Once a library is selected, a list of dielectric materials in the library should appear in the dialog box.

There is a default Global library provided by Sonnet that contains commonly used dielectric materials.

- 4 To add a dielectric material, click on the Add button and enter the desired name and loss parameters in the Dielectric Editor dialog box, then click on OK to add a new dielectric to the library.**

This dielectric material is now available in the library to be used in your project.

- 5 To edit an existing dielectric, click on its entry to select it, then click on the Edit button and make any desired changes, then click on the OK button.**

The Dielectric Editor dialog box is closed and any changes are updated in the dielectric material's entry line.

- 6 To delete a dielectric material from your dielectric library, click on the Dielectric Material to select it and click on the Delete button in the Dielectric Library dialog box.**

This deletes the dielectric material from the library. Note that if the dielectric material has been added to the project and is in use, deleting the dielectric from the library has no effect on the dielectric material being used in the project. The dielectric material will no longer be available in the library for other projects but will still exist in the present project.

This completes editing the dielectric library.

Setting Up a Dielectric Library

To create a new dielectric library, global or local, do the following:

- 1 Select Circuit \Rightarrow Dielectric Layers from the main menu of the project editor.**

The Dielectric Layers dialog box is opened on your display. This dialog box allows you to add, edit and delete dielectric materials for your project.

- 2 Click on the Library button to edit your dielectric libraries.**

This opens the Dielectric Library dialog box which allows you to create dielectric libraries and to edit them.

- 3 Select the type of library you wish to create by clicking on the My Library or Global Library radio button in the Dielectric Library dialog box.**

Once a library is selected, a list of dielectric materials in the library should appear in the dialog box.

- 4 To create a new library, click on the Specify button to the right of the radio button for the library you selected.**

A browse window appears on your display.

- 5 Use the browse window to specify the location and filename for the new dielectric library file.**

The default name for a new dielectric library is die-library.txt. If you wish to name the library file something else, edit the filename entry.

- 6 Once the desired location is specified, click on the OK button in the browse window.**

The browse window is closed and a new dielectric library file with no entries is created at the location you just specified. Use the rest of the edit controls in the Dielectric Libraries dialog box to create entries for this dielectric library. For details on editing a dielectric library, see “Editing a Metal Library,” page 86.

Chapter 6 Ports

This chapter describes the four different types of ports used in Sonnet, how they are modeled and how to place or delete them in your circuit.

The four types of ports used in Sonnet are the standard box-wall port, the ungrounded-internal port, the via port and the auto-grounded port.

Port Types

All ports in Sonnet are two-terminal devices. What distinguishes the port types is where the terminals are connected.

The most common type of port used is the standard box-wall port in which the first terminal is attached to a metal polygon and the second terminal is attached to the box wall (ground). This type of port can be de-embedded by *em*.

Occasionally, however, it is useful to attach the two terminals of a port between two adjacent polygons. This is the ungrounded-internal port. This type of port cannot be de-embedded by *em*. When analyzing multi-port circuits to find S-, Y- or Z-parameters, all of the ports in the circuit should be grounded. An ungrounded port can have a different ground reference from other ports in the circuit, which, in turn, can corrupt the results.

The third type of port is the via port. A via port has one terminal connected to a polygon on a given circuit level and the other terminal connected to a second polygon on a circuit level above the first polygon. *Em* cannot de-embed a via port. A via port is most commonly used when you wish to attach a port between two adjacent levels in your circuit or when you want a port to go up to the box cover rather than down to ground.

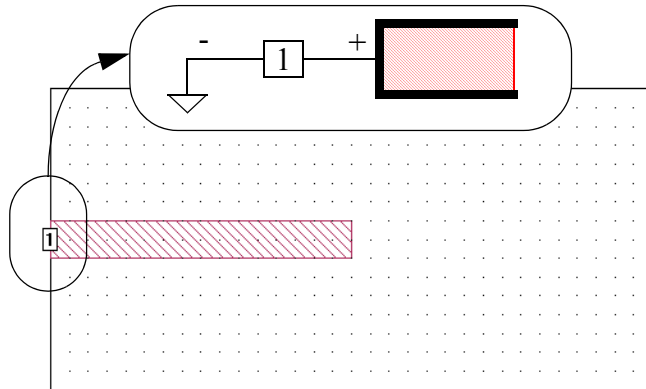
The last type of port is the auto-grounded port. The auto-grounded port is used in the interior of a circuit and has one terminal attached to the edge of a metal polygon and the other terminal attached to the ground plane through all intervening dielectric layers. *Em* can de-embed an auto-grounded port. For more details and the restrictions on using auto-grounded ports, see “Automatic-Grounded Ports,” page 112.

Ports, of any type, are not allowed on diagonal lines and are not allowed to overlap each other (just touching is permitted). The analysis engine tests for these error conditions.

Each port type is described in more detail in the sections that follow as well as issues pertaining to all port types, such as port numbering and impedance.

Standard Box-Wall Ports

A standard box-wall port is a grounded port, with one terminal attached to a polygon edge coincident with a box wall and the second terminal attached to ground. An example of a standard box-wall port is shown below. Standard box-wall ports can be de-embedded.



Adding Standard Ports

- 1 To add a standard port to a circuit, click on the *Add Port* button in the tool box, or select *Tools* \Rightarrow *Add Port* from the main menu of the project editor.
- 2 Then click on the polygon edge at the desired position for a port. A small box with a number in the center will appear on your circuit, indicating the position of the port.

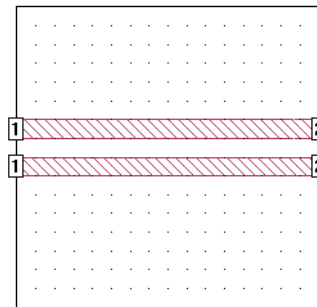
Ports are numbered automatically, in the order in which they are added to your circuit, starting at the number one. You may later choose to change a port number; see “Changing Port Numbering,” page 103, for details.

Special Port Numbering

All ports are assigned a number at the time they are created in the project editor. There is no limit on the number of ports and the number of ports has absolutely no impact on analysis time. By default, the ports are numbered by the order in which they are created (i.e. first port created is assigned the number 1, second port created is assigned the number 2, etc.). With this default method, all ports are positive and unique. However, there are some applications that require the ports to have duplicate, or even negative, numbers.

Ports with Duplicate Numbers

All ports with the same number, as pictured below, are electrically connected together. As many physical ports as desired may be given the same numeric label. Such ports are sometimes called “even-mode” or “push-push” ports and have many uses, including simulating thick metal or the even-mode response of a circuit. See , “Modeling an Arbitrary Cross-Section,” for an example of using “push-push” ports.

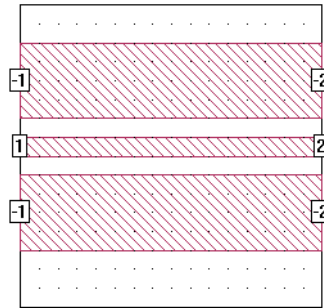


Ports with identical port numbers are electrically connected together.

Ports with Negative Numbers

Ports may also have negative numbers as shown in the figure on page 103. This feature can be used to redefine ground. Strictly speaking, *em* sums the total current going into all the positive ports with the same port number and sets that equal to the total current going out of all the ports with that same negative port number. For example, for a circuit with a +1 port and a -1 port, *em* sets current flowing into

port +1 to be equal to the current flowing out of port -1. Thus, they are sometimes called “balanced”, “push-pull” or “odd-mode” ports. Coplanar lines can be represented with balanced ports. See Chapter 23, “Coplanar Waveguide and Balanced Ports,” for an example of push-pull ports.



An example of push-pull ports.

Ports may be non-sequential, i.e., you may have only 2 ports, one labeled “1” and the other labeled “4”. The port order for the S,Y, or Z parameters will be listed in increasing numeric order. For the example of a two-port with ports labeled “1” and “4”, the output would be as follows: S11, S41, S14, S44.

Changing Port Numbering

- 1 Click on the port or ports to select it/them. This will enable the *Modify* menu option.
- 2 Select *Modify* \Rightarrow *Port Properties* to open the Port Attributes dialog box.
- 3 Now the port number can be changed by typing the desired number in the *Number* text entry box of the dialog box. Any nonzero integer, negative or positive, is valid. Note that if multiple ports are selected, “Mixed” will appear in the *Number* text entry box. Editing this field will apply the same number to all selected ports.

Port Normalizing Impedances

Whenever results are to be incorporated into a circuit theory based analysis program, the normalizing impedance for each port should be 50 ohms. In rare cases, S-parameters normalized to some other impedance are desired.

For example, you may want to see what the reflection coefficient of a structure is when port 2 is connected to a 1.0 pF capacitor in parallel with a 10 ohm resistor (a power FET input model), while the input is being driven with a source which has a 35 ohm internal impedance. In this case, normalize port 1 to 35 ohms and port 2 to 10 ohms plus 1.0 pF.

Frequently, pure electromagnetics are carried out using S-parameters normalized to the characteristic impedance of each connecting transmission line. These are often called “generalized” S-parameters, in spite of the fact that the line impedance is usually not specified, thus precluding precise conversion to 50 ohms for use in circuit theory software.

To understand the physical meaning of normalized S-parameters, recall that our standard 50 ohm S-parameters are measured by terminating all ports in 50 ohms and then measuring ratios of incident to reflected (or transmitted) wave amplitudes. If we were to use 60 ohm terminations, instead of 50 ohm terminations, the resulting measurements of traveling wave ratios would yield S-parameters normalized to 60 ohms. If we were then to take the S-parameters which had been measured using 60 ohm port terminations and use it in a circuit theory program (which expects you to use 50 ohm terminations), we would get incorrect results.

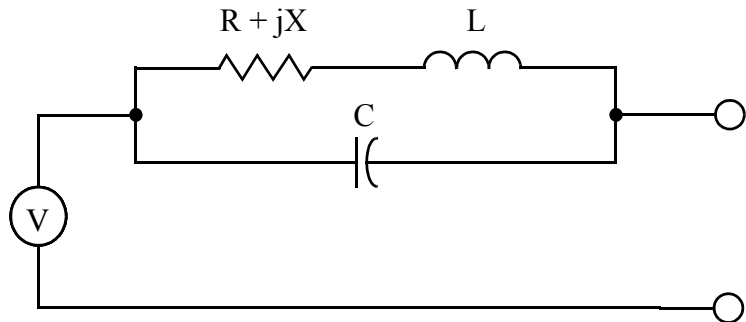
However, 60 ohm S-Parameters do have uses. Say you want to know what percentage of power is absorbed by a 60 ohm load terminating port 2 of a two port circuit. That is simply one minus the magnitude squared of S_{11} , using 60 ohm S-parameters. If you are interested in a different load, use a different normalizing impedance.

S-Parameters can be renormalized to any real impedance by using the circuit network capability of *em*, or a circuit theory based program, to cascade appropriate transformers on each port. This technique can not be used with a complex impedance, as is the case with a lossy line or complex load.

In *em*, the default normalizing impedance is 50 ohms. If you would like a different normalization, refer to the section “Changing Port Impedance,” page 106, on how to specify the normalizing impedance for each port.

The normalizing impedance is represented by four numbers. First is the real part in ohms. Next comes the reactive part in ohms. Third is the inductive part in nanohenries (nH). The last number is the capacitive part in picofarads (pF). The inductive and capacitive part modify only the reactive portion of the load, they are included so you do not have to manually re-calculate the reactive part at each frequency.

The resistance, reactance and inductance are all connected in series when the specific normalizing impedance is calculated at each frequency of analysis as illustrated below. The capacitance is connected in parallel with the result and then the final normalizing impedance at the frequency of analysis is calculated.



Equivalent circuit of an *em* port.

NOTE: **The normalizing impedances are ignored if Y- or Z-parameters are specified for output. Y- and Z-parameters are always normalized to one ohm.**

This capability should be used only by the most advanced users. If the above discussion is not clear, seek assistance. Under no circumstances should this capability be used on data which is to be incorporated in a circuit in a standard circuit theory program other than Sonnet. Many such programs assume S-parameters normalized to exactly 50 ohms.

For example, the 50 ohm S-parameters of a microstrip step junction look like a very small shunt capacitance and a very small series inductance. In other words, the reflection coefficient is almost zero and the transmission coefficient is almost unity with a couple degrees of phase. If, instead of 50 ohm S-parameters, we were to use S-parameters normalized to the characteristic impedances of the lines connecting to each of the two ports (this is common in electromagnetics), the S-parameters would look like a transformer between the two impedances. This causes many circuit theory programs, which are expecting 50 ohm S-parameters, to give grossly incorrect results.

Changing Port Impedance

There are two methods for changing the impedance of a port. If you wish to change the impedance of a given port, and do not need to see the impedance values of other ports, take the following steps using the project editor:

- 1 Click on the port or ports to select it/them. This will enable the *Modify* menu option.
- 2 Select *Modify* \Rightarrow *Port Properties* to open the Port Attributes dialog box.
- 3 The impedance values can be changed by typing the desired values in the *Resistance*, *Reactance*, *Inductance* and *Capacitance* text boxes in the dialog box. This changes the parameters on all ports selected and all ports with the same number as the ports selected.

If you wish to change the impedance of a given port, and wish to see the impedance values of other ports while doing so, proceed as follows:

- 1 Select *Circuit* \Rightarrow *Ports* from the main menu to open the Port Impedance dialog box.
- 2 Now the impedance values for any port can be changed by typing the desired values in the *Resistance*, *Reactance*, *Inductance* and *Capacitance* fields in the row labeled with the desired port number.



TIP

Note that the impedance of multiple ports may be changed at the same time through the first method by selecting multiple ports before selecting *Modify* \Rightarrow *Attributes*, and by the second method, by modifying all the desired port values while the Port Impedance dialog box is open.

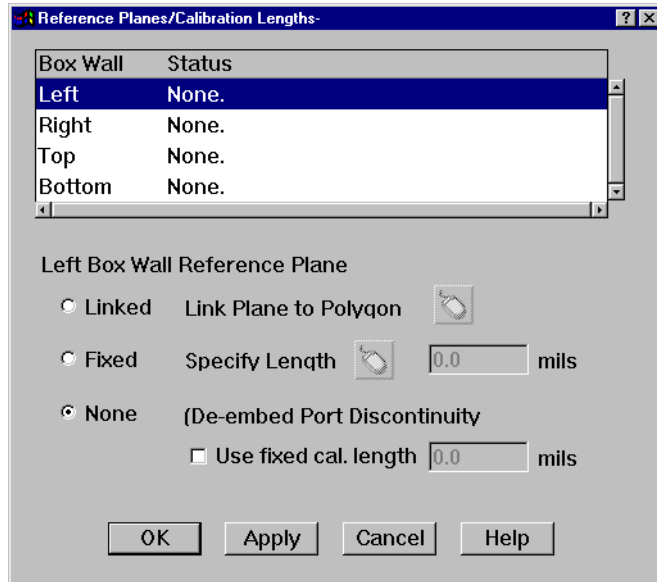
Defining Reference Planes/Calibration Lengths

Reference Planes or Calibration Lengths, which are mutually exclusive, can be set for most types of ports, but the method differs according to the port type. Reference planes cannot be set for via ports since *em* cannot de-embed them, or for ungrounded-internal ports since it would be uncertain which box wall was being referenced. Detailed below are the methods for standard ports and auto-grounded ports.

Standard Ports

For standard ports, which normally reside on the box wall, the Reference Plane or Calibration Length refers to the distance to the box wall. To set the Reference Planes/Calibration lengths, do the following:

- 1 Select *Circuit* \Rightarrow *Ref. Planes/Cal. Lengths* from the main menu. This will open the Reference Planes/Calibration Lengths dialog box.



- 2 Select the side from which the reference plane or calibration length is to extend (*Top, Left, Right, or Bottom*) in the box wall list.
- 3 Then select the type of reference plane, Linked or Fixed, from the radio buttons. If you wish to define a calibration length, click on the None radio button and then click on the Use fixed cal. length check box.
- 4 If you select a linked reference plane, click on the mouse button, then select a point on a polygon in your circuit to which you wish to link the reference plane. If you select a fixed reference plane, either enter the length in the text entry box or click on the mouse button to set the distance in your circuit. If you are defining a calibration length, enter the desired length in the text entry box.
- 5 To remove a reference plane or calibration length, click on the None radio button and do not check the Use fixed cal. length. The same reference plane or calibration length is used on all ports on all levels on that side of the box.

Auto-ground Ports

For autoground ports, the reference plane or calibration length is defined as the distance from a particular port. To set a reference plane or calibration length for an auto-grounded port, do the following;

- 1 Click on the port to select it. This will enable the *Modify* menu option.
- 2 Select *Modify* \Rightarrow *Port Attributes* to open the Port Attributes dialog box depicted on page 115.
- 3 Then enter the desired length in the *Ref. Plane or Cal. Length* field in the Autognd Port Data box. Alternatively, you may use the mouse for setting the reference plane by clicking the *Use Mouse* button and then clicking in your circuit at the desired location. The *Ref. Plane* entry will be updated to reflect your selection.
- 4 To remove a reference plane or calibration length, enter a value of zero in the respective field.



TIP

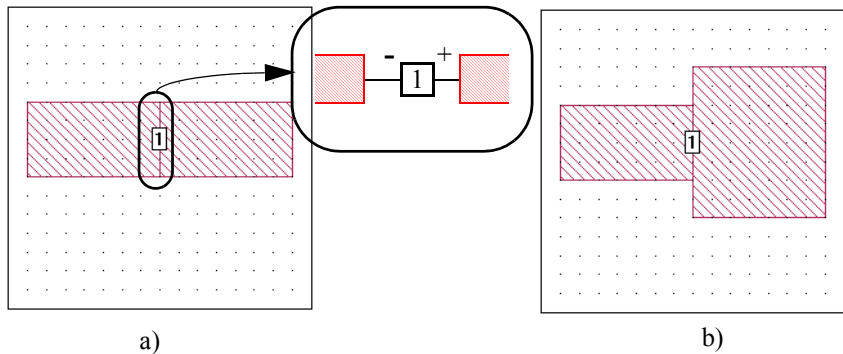
Changing a port to an autoground type and setting up a reference plane or calibration length for the port can be accomplished at the same time in the Port Attributes dialog box. It is also possible to set calibration lengths for multiple ports by selecting the desired ports, selecting *Modify* \Rightarrow *Port Properties* and inputting a value in the calibration length text entry box in the Port Attributes dialog box.

Deleting Ports

All types of ports are deleted in the same manner. Simply select the port, then perform a *Cut* or *Delete* operation, either through the tool bar or menu, to remove the port from the circuit.

Ungrounded-Internal Ports

A standard ungrounded-internal port is located in the interior of a circuit and has its two terminals connected between abutted metal polygons. An ungrounded-internal port is illustrated below. Note that internal ports have their negative terminals on the left or bottom side of the port depending on how the port is oriented. Ungrounded-internal ports can be de-embedded by *em*, however, you may not set a reference plane or calibration length.



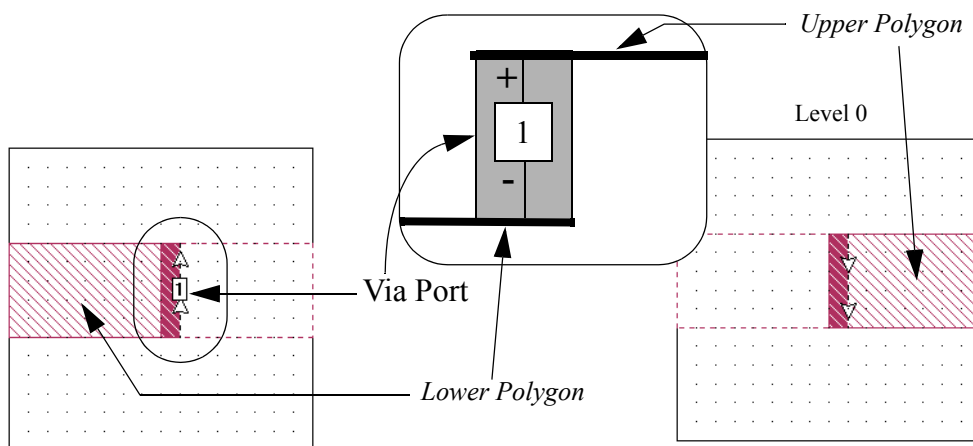
In part a of the figure, the ungrounded-internal port is attached between two polygons which have equal widths. This is not a necessary condition for ungrounded-internal ports. These ports can also be attached between polygons which are abutted, but have unequal widths, as shown in part b. The only difference between the two conditions is that de-embedding requires the use of more standards (and therefore more time) when the polygons have unequal widths.

Ports on the edge of a single polygon are not allowed; extreme care should be taken in interpreting the results since the port has no access to ground.

You add an ungrounded-internal port in the same manner that you add a standard box wall port; for details, see page 101.

Via Ports

A via port has one terminal connected to a polygon on a given circuit level and the other terminal connected to a second polygon on a circuit level above the first polygon. For more details on vias, Chapter 18, “Vias and 3-D Structures”. Thus, when ports are desired on the interior of a circuit, capture a via between two layers and add a port to the via polygon or edge via. An example of this port type on an edge via is shown below.



An example of a circuit with a standard via port. A side view of the enclosed area on the circuit is shown in the middle.

Em cannot de-embed via ports. However, in a circuit which contains a combination of via ports and other port types, the other port types can still be de-embedded. **Em** will automatically identify all of the other ports present in the circuit and de-embed them, but leave the via ports un-de-embedded.

The example file [Dual_patch](#) has an example of a via port used in a patch antenna. This example file can be found in the Sonnet example files.

In most cases where you need grounded ports, your first choice would be to use auto-grounded ports (as discussed in the next section), especially since it is possible to de-embed an autogrounded port. The two most common cases where

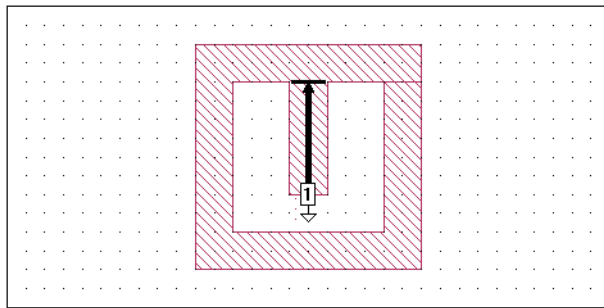
a via port would be used is when you wish to attach a port between two adjacent levels in your circuit or when you want a port to go up to the box cover rather than down to ground.

Adding Via Ports

- 1 To add a via port, you must first have two polygons on adjacent levels with a via connecting them. For more details on how to create vias, see Chapter 18, “Vias and 3-D Structures”.
- 2 Then click on the *Add Port* button in the tool box of the project editor, to put you in Add Port mode.
- 3 Then click on the edge via at the lower level polygon to add the port. An example is pictured on the previous page.

Automatic-Grounded Ports

An automatic-grounded port is a special type of port used in the interior of a circuit. This port type has one terminal attached to the edge of a metal polygon located inside the box and the other terminal attached to the ground plane through all intervening dielectric layers. An auto-grounded port with a reference plane shift is shown below.



In many circuits, the addition of auto-grounded ports has little influence on the total analysis time of the *em* job. However for some circuits, auto-grounded ports may require some extra overhead calculations, thus increasing the total analysis

time. Therefore, they should be used only when they provide an advantage over standard box-wall ports. Auto-grounded ports provide advantages over standard box-wall ports when:

- the layout of your circuit does not allow a direct path for a feed line to be connected between the port and the box wall, as in the figure above, or
- your circuit requires a large feed structure to reach the box wall. If all or part of your feed structure can be eliminated, using an auto-grounded port could reduce the total number of subsections in your circuit, thus decreasing the analysis time and/or memory requirements.

Auto-grounded ports are similar to via ports with the exception of the following characteristics:

- Via ports require you to manually create vias that extend upward through the dielectric to the edge of a metal polygon. This is not the case with auto-grounded ports. You simply place auto-grounded ports anywhere a grounded port is needed. *Em* automatically detects the presence of auto-grounded ports in the circuit and connects the port terminals appropriately.
- Auto-grounded ports connect directly through all dielectric layers to the ground plane. Via-ports allow the flexibility of connecting between any two adjacent dielectric layers.
- Auto-grounded ports are de-embedded when the de-embedding option is used, while via ports are not.
- Reference planes may be set with auto-grounded ports but cannot be set for via ports.

Special Considerations for Auto-Grounded Ports

Metal Under Auto-Grounded Ports

You cannot have metal directly beneath an auto-grounded port in a multi-layer circuit. Auto-grounded ports are two-terminal devices with one terminal connected to an edge of a metal polygon and the second terminal connected to the

ground plane. When *em* detects the presence of an auto-grounded port, it automatically connects the two terminals in this manner. This includes circuits which have multiple dielectric layers between the polygon and the ground plane. However, in order for *em* to accomplish this, there must be a direct path from the edge of the metal polygon to the ground plane. When an auto-grounded port is used in a circuit where there is more than one dielectric layer between the port and the ground plane, *em* checks to make sure that there is no metal directly beneath the auto-grounded port. If metal is found, *em* prints an error message and stops.

Edge of Metal Polygon is Lossless

Auto-grounded ports can attach to the edge of any metal polygon in the interior of a circuit. There are no restrictions on the loss parameters of the metal used in the polygon. However, along the edge of the metal polygon where the port is attached, *em* does force the cells to be lossless. For most circuits, this should have little or no effect on the results. If, however, the port is attached to a highly lossy metal polygon, such as a thin-film resistor, the edge cell(s) of that polygon will be made lossless, and the output results may be affected.

Require Short Distance to Ground

Auto-grounded ports require a short distance to the groundplane. The distance between the auto-grounded port and the substrate (groundplane) should be less than 1/8 wavelength.

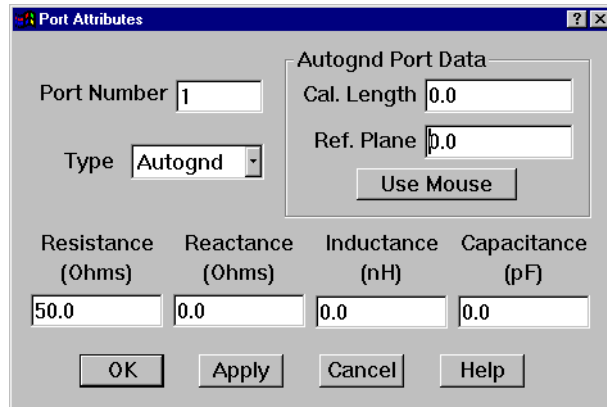
Auto-Grounded Ports on Box-Wall

Auto-grounded ports are designed to be used in the interior of a circuit. If an auto-grounded port is placed on a box-wall, it is treated as if it were a standard box-wall port.

Adding Auto-grounded Ports

- 1 To add an auto-grounded port, proceed as you would to add a standard port, as described on page 101.
- 2 Click on the port to select it. This will enable the *Modify* menu option.

- 3 Select *Modify* \Rightarrow *Port Properties* to open the Port Attributes dialog box which will resemble the figure below.
- 4 Now the port can be changed from a standard port to an auto-grounded port by choosing *Autognd* from the drop list in the *Type* field. The port should now be similar to the port shown on page 112.

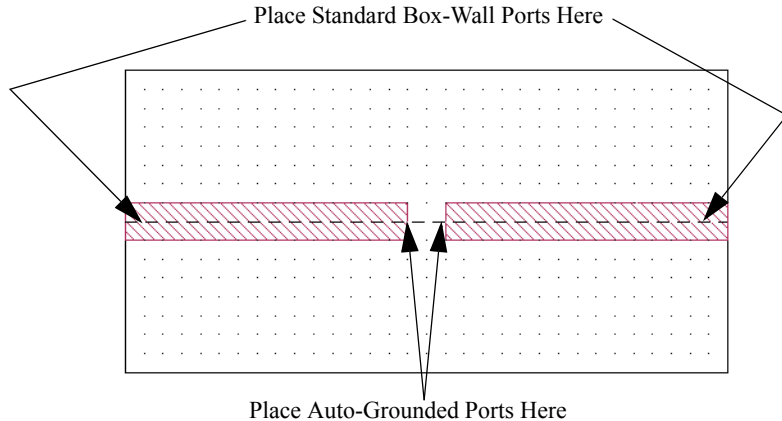


Creating Auto-grounded Ports: An Example

This next section details how to add auto-grounded ports to a circuit.

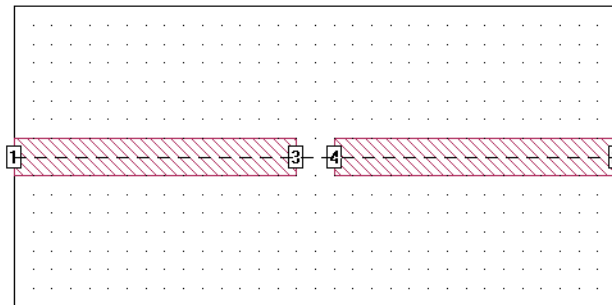
You add an auto-grounded port to a circuit in the same way that you add a “standard” box-wall or ungrounded-internal port. The only difference is that after adding the port, you must change its type to “auto-grounded.”

The procedure for adding auto-grounded ports to a circuit is shown below.



Assume that you start with the circuit pictured above. Detailed below are the steps you would use to create two “standard” box-wall ports and two “auto-grounded” ports at the locations indicated in the figure above.

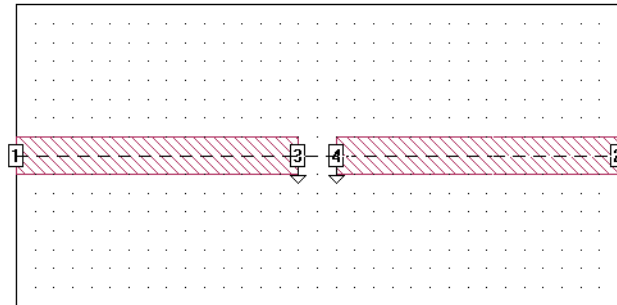
- 1 Select *(Shift)Tools* \Rightarrow *Add Port*.
- 2 Click with the mouse on each of the indicated polygon edges. The resulting circuit will be similar to that shown in on page 116. Note that all four ports are “standard” port types.



- 3 Press *ESC* to return to pointer mode.

- 4 Use the mouse to lasso ports #3 and #4. This action will highlight the ports you select.
- 5 Select *Modify* \Rightarrow *Port Properties*. This will open the Port Attributes dialog box shown in on page 115.
- 6 Now the ports can be changed from standard ports to auto-grounded ports by choosing *Autognd* from the drop list in the *Type* field.
- 7 Click on the *OK* button to exit the Port Attributes dialog box.

After converting ports #3 and #4 in this manner, the circuit will appear as shown below.



The circuit now has the two box-wall and two auto-grounded ports in the desired locations.

Chapter 7

De-embedding

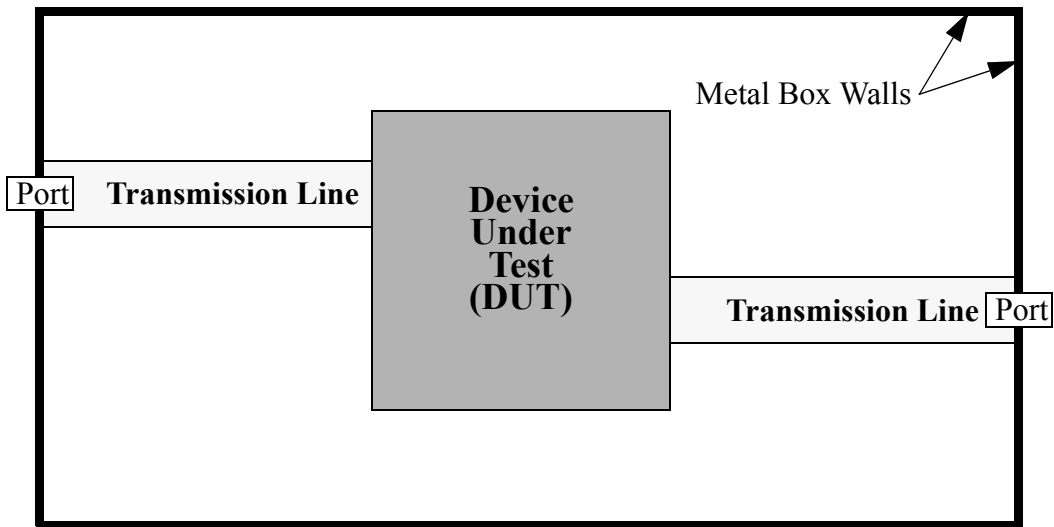
Each port in a circuit analyzed by *em* introduces a discontinuity into the analysis results. In addition, any transmission lines that might be present introduce phase shift, and possibly, impedance mismatch and loss. Depending upon the nature of your analysis, this may or may not be desirable. De-embedding is the process by which the port discontinuity and transmission line effects are removed from the analysis results.

The figure on page 120 illustrates the general layout of a circuit to be analyzed with *em*. The device under test (DUT), shown as a box in the figure, is the circuitry for which we wish to obtain analysis results. The DUT is located inside the metal box and is connected to one or more ports. The ports may be located on box walls, as in the figure, or in the interior of the metal box (see Chapter 6 for a description of port types available in *em*). Typically, transmission lines are necessary to connect the ports to the DUT.

When de-embedding is enabled, *em* performs the following sequence of steps:

- 1 Calculates port discontinuities.

- 2 Removes effects of port discontinuities from analysis results.
- 3 Optionally shifts reference planes (removes effects of feed transmission lines from analysis results).
- 4 Calculates transmission line parameters E_{eff} and Z_0 .



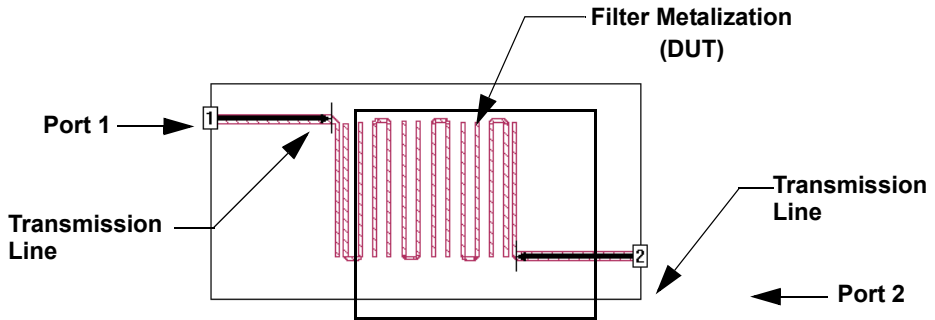
General layout of a circuit to be analyzed with **em**.

Upon completion of the de-embedding process, **em** outputs de-embedded S-parameter results, transmission line parameters and the calculated port discontinuities.

An abbreviated summary of the de-embedding algorithm used is presented in reference [70] and the complete theory is presented in reference [71] in **Appendix II, "Sonnet References"** on page 419.

Enabling the De-embedding Algorithm

To demonstrate de-embedding with *em*, we will analyze the filter shown below. This is an example of a hairpin filter with a passband of approximately 4.0 to 4.15 GHz. This circuit consists of eight sections making up the filter metalization, two ports and two transmission lines connecting the ports to the filter metalization. Reference planes have been defined for port 1 and port 2 at the left and right edges of the filter metalization, respectively. These reference planes instruct *em* to remove the effects of the transmission lines up to the filter metalization when de-embedding is enabled.



Port discontinuities and transmission lines at the upper left and lower right are removed from the *em* analysis results by enabling de-embedding.

NOTE:

Adding reference planes to a circuit in the project editor does not automatically enable de-embedding in *em*. However, the De-embed run option is set by default.

You select the de-embed option in the Advanced Options dialog box in the project editor. This run option is set by default. To open the Advanced Options dialog box, select *Analysis* \Rightarrow *Setup* from the project editor main menu. Then click on the Advanced button in the Analysis Setup dialog box which appears.

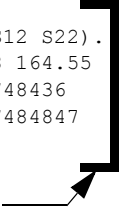
An analysis was performed on the filter starting at 3.95 GHz to 4.2 GHz in 0.002 GHz steps with the de-embedding option on.

As *em* performs the analysis, messages are output to the status section of the analysis monitor detailing the various tasks in the process. The actual response data is shown in the output window when the Response Data button is pressed. Below is the output response data for the filter as it appeared in the analysis monitor.

```
Run 1:  Sat Apr 07 15:18:18 2001.  Frequency Sweep.

Frequency:  3.95 GHz
De-embedded 50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
3.95000000 0.999998 149.22 0.001765 -113.1 0.001765 -113.1 0.999998 164.55
!< P1 F=3.95 Eeff=(2.93154529 0.0) Z0=(52.0732019 0.0) R=0.0 C=0.0748436
!< P2 F=3.95 Eeff=(2.93105773 0.0) Z0=(52.0687952 0.0) R=0.0 C=0.07484847
```

De-embedded S-parameter, transmission
parameter, and port discontinuity results



The analysis monitor displays the de-embedded S-parameter results along with the feed transmission line parameters (E_{eff} and Z_0) and calculated discontinuity (R and C) for each de-embedded port. “P1” and “P2” stand for “port 1” and “port 2”, respectively. A detailed discussion concerning the port discontinuities (R and C) is presented in the next section.

De-embedding Port Discontinuities

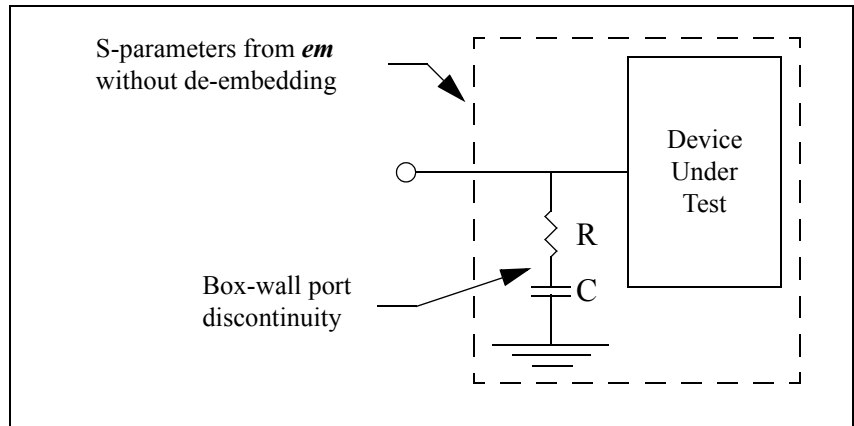
All ports in *em* introduce a discontinuity into the analysis results. Sometimes, this is desirable. For example, when analyzing a circuit fabricated with box walls, the effects introduced by a box-wall port discontinuity are real. Under this circumstance, the discontinuity should not be removed. However, in analyses where only the behavior of the DUT is of interest, all port discontinuities should be removed by de-embedding.

When enabled, *em*'s de-embedding algorithm automatically removes the discontinuities for box-wall, ungrounded-internal and auto-grounded ports (see Chapter 6 for a description of port types available in *em*). A via port is the only

type of port that cannot be de-embedded by *em*. The port discontinuity for box wall ports is described in the section that follows. The discontinuity for the other types of ports is similar in nature.

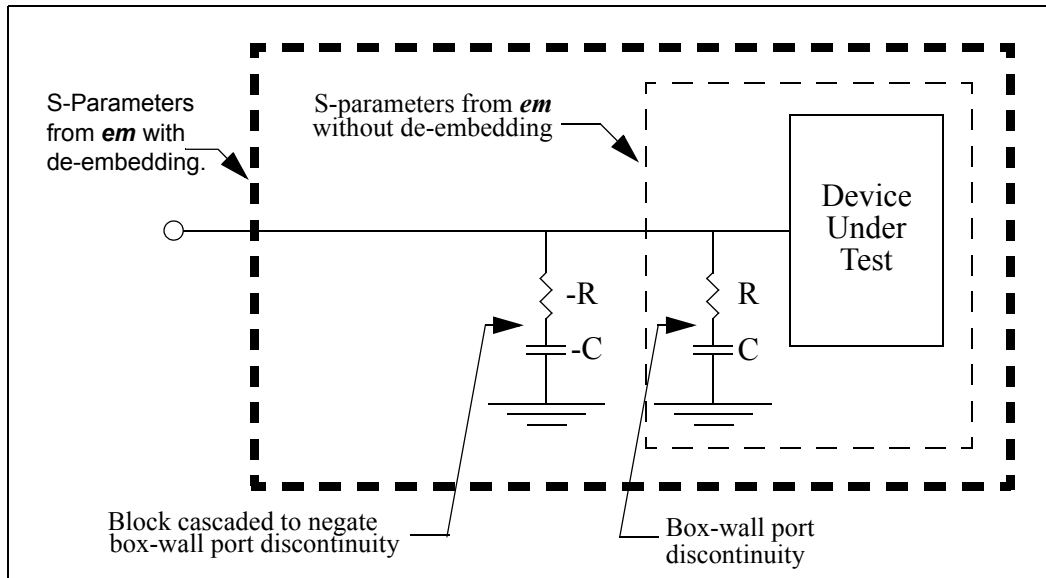
Box-Wall Ports

Box-wall ports have one port terminal connected to a polygon inside the metal box, and the second port terminal connected to ground (see the figure on page 101). The port discontinuity is modeled as a series resistor, R , and capacitor, C , shunted to ground as shown below. If the circuit being analyzed is completely lossless, the resistor value, R , is zero. Even with loss in the circuit, the capacitive reactance is normally very large compared to the resistance.



Port discontinuity associated with a box-wall port.

When enabled, *em*'s de-embedding feature automatically calculates the values of R and C for each box-wall port present in the circuit. The port discontinuities are then removed by cascading a negative R and C as illustrated above.



De-embedding automatically cancels the discontinuity associated with a box-wall port.

Shifting Reference Planes

Transmission lines are required in many circuits to connect ports to the device under test (DUT). If the length of a transmission line is more than a few degrees relative to a wavelength, unwanted phase (and possibly loss) will be added to the S-parameter results. If the impedance of the transmission line differs from the normalizing impedance of the S-parameters (usually 50 ohms), an additional error in the S-parameters results. Thus, if we are only interested in the behavior of the DUT, any “long” transmission lines connecting the ports to the DUT should be removed during de-embedding. The process of removing lengths of transmission line during de-embedding is known as “shifting reference planes”.

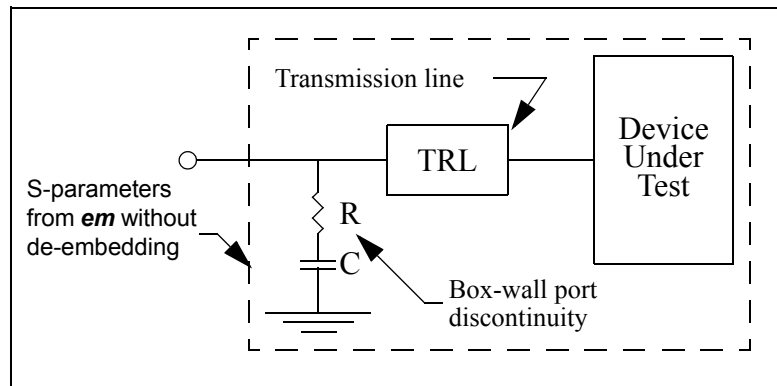
Reference planes may be specified in the project editor for box-wall and auto-grounded ports, but not ungrounded-internal ports. When *em* detects a reference plane, and de-embedding is enabled, it automatically builds and analyzes the calibration standards necessary to de-embed the port and shift the reference plane by the specified length.

NOTE:

Reference planes are not necessary for de-embedding. If you do not specify a reference plane in the project editor for a box-wall or auto-grounded port, the reference plane length defaults to zero. This means that *em* will not shift the reference plane for that port when de-embedding is enabled. However, *em* will remove the discontinuity for that port.

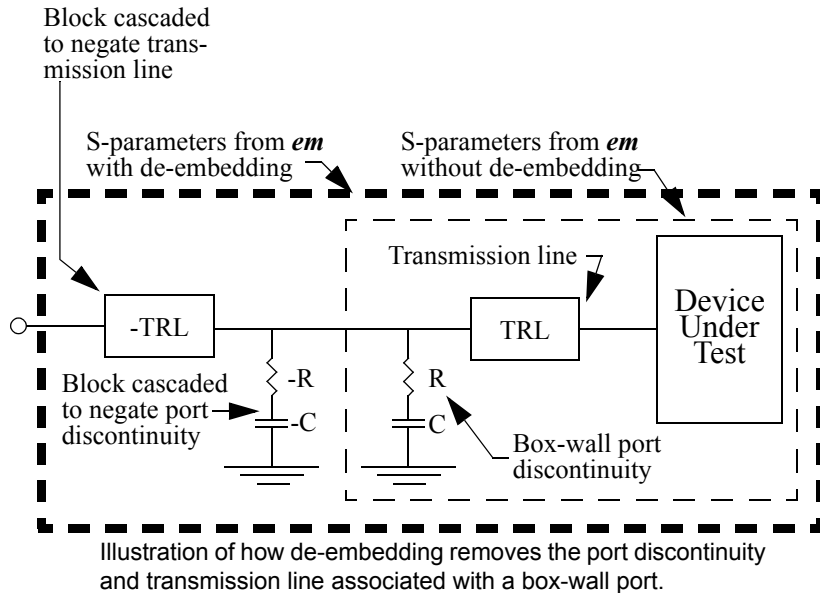
Single Feed Line

The figure below shows a circuit with a length of transmission line, TRL, inserted between a box-wall port and the device under test.



Port discontinuity and transmission line associated with a box-wall port.

When de-embedding is enabled, *em* removes the transmission line in a manner similar to that used to remove the port discontinuity. *em* calculates S-parameters for the TRL alone, and then cascades a “negative” TRL along with negative R and C as illustrated in the next figure.



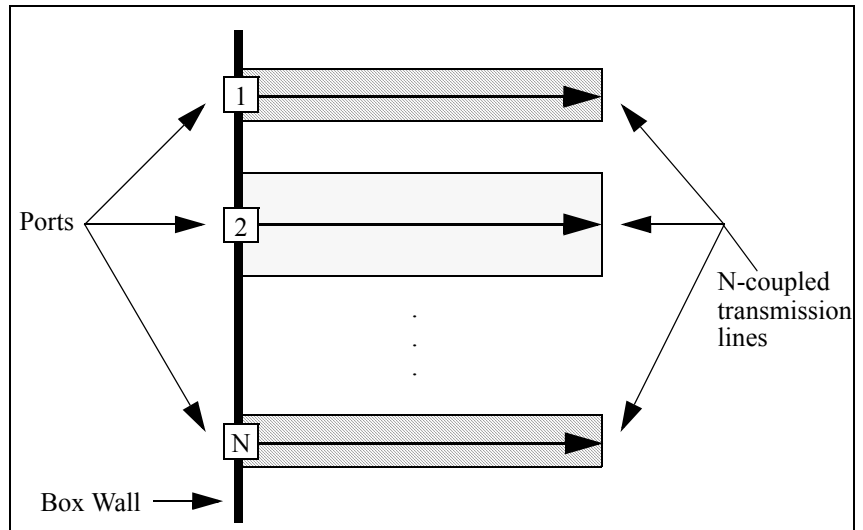
Coupled Transmission Lines

The two previous figures illustrated how the reference plane for a single transmission line attached to a box-wall port is shifted during de-embedding. In general, there may be multiple transmission lines on a given box wall on one or more circuit levels. This is illustrated in the next figure. In this situation, *em* shifts the reference plane an equal distance for all transmission lines on the given box wall. All coupling between the transmission lines is accounted for and removed.

NOTE:

When shifting a reference plane for coupled lines, *em* assumes the following:

- a) all coupled lines are either horizontal or vertical
- b) the width of each coupled line is constant
- c) the spacing between coupled lines is constant.



De-embedding shifts the reference plane an equal distance for all N-coupled transmission lines on a given box wall. The coupling between transmission lines is removed by the de-embedding process.

De-embedding Results

The listing below shows the de-embedded results obtained earlier in the chapter from the analysis of the example filter circuit (see page 121). This example illustrates the format of the de-embedded data is output in the analysis monitor and saved as part of your project. If you wish to also have a separate file containing

your response data, you may specify that one be output from an analysis using the *Analysis* \Rightarrow *Output Files* command in the project editor. See “Analysis - Output Files” in the project editor’s online help for details on specifying a file.

```
Run 1:  Wed Oct 11 18:38:10 2000.  Frequency Sweep.

Frequency:  10 GHz
De-embedded 50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
10.0000000 1.000000 -72.59 6.414e-4 17.050 6.414e-4 17.050 1.000000 -73.31
!< P1 F=10.0 Eeff=(6.45562325 0.0) Z0=(51.7880826 0.0) R=0.0 C=0.04163932
!< P2 F=10.0 Eeff=(6.47619184 0.0) Z0=(51.8822385 0.0) R=0.0 C=0.04165009

Analysis successfully completed.
```

Example showing format of results obtained when de-embedding is enabled in **em**.

You should notice the following about the results in above:

- The line which starts with “De-embedded” is a comment line which describes the analysis results on the line below. In this example, the results are de-embedded 50 ohm S-parameters in Touchstone magnitude/angle format.
- The second line gives the analysis results.
- The remaining two lines give de-embedding information for each port in the circuit. The various fields are defined as follows:

P#:	Port number.
F:	Frequency in units defined earlier in the results file.
Eeff:	Effective dielectric constant of the transmission line connected to the port.
Z0:	TEM equivalent characteristic impedance of the transmission line, in ohms.
R:	Equivalent series resistance of port discontinuity, in ohms.
C:	Equivalent series capacitance of port discontinuity, in pF.

De-embedding Error Codes

Please see “De-embedding Error Codes” in online help for explanations of the error code messages. To access online help, select *Help* \Rightarrow *Contents* from any Sonnet application.

Chapter 8

De-embedding Guidelines

The previous chapter describes the basics of de-embedding: what it is, how it is enabled, and what it does when enabled. This chapter presents guidelines for obtaining good de-embedded results.

Calibration Standards

In order to determine the port discontinuity as described on page 122, Sonnet must electromagnetically analyze several calibration standards which include the same port discontinuity as the primary circuit. For a single box-wall port, the calibration standards are two through lines which have the same geometry (width, dielectrics, distance to box wall, etc.) as the polygon which has the port attached to it. Sonnet then builds and analyzes two through lines based on this geometry. If there is more than one port on a box wall, then the calibration standard is a multiple-coupled line.

The lengths of these two through lines can determine the accuracy of the de-embedding (see below for a discussion of the problems which can occur with improper lengths). By default, Sonnet chooses calibration lengths for you. If the port contains a reference plane, then the first calibration length is the same length as the reference plane and the second length is double the first. If no reference plane exists, Sonnet chooses one for you.

If you are having trouble with de-embedding, you may want to change this calibration length using the following sections as a guide. If you are using reference planes, you can simply change your reference plane length, and the calibration lengths will change accordingly. If you are not using reference planes, then you can set the calibration length using *Circuit* \Rightarrow *Ref. Plane/ Cal Length*. This allows you to set the first calibration length. The second is always twice as long as the first.

Defining Reference Planes

Sonnet places very few restrictions on the reference planes which may be defined for a given circuit. This is done intentionally so as to provide maximum flexibility for all users.

However, there are some basic guidelines concerning reference planes that should almost always be followed. These guidelines are discussed below.

De-embedding Without Reference Planes

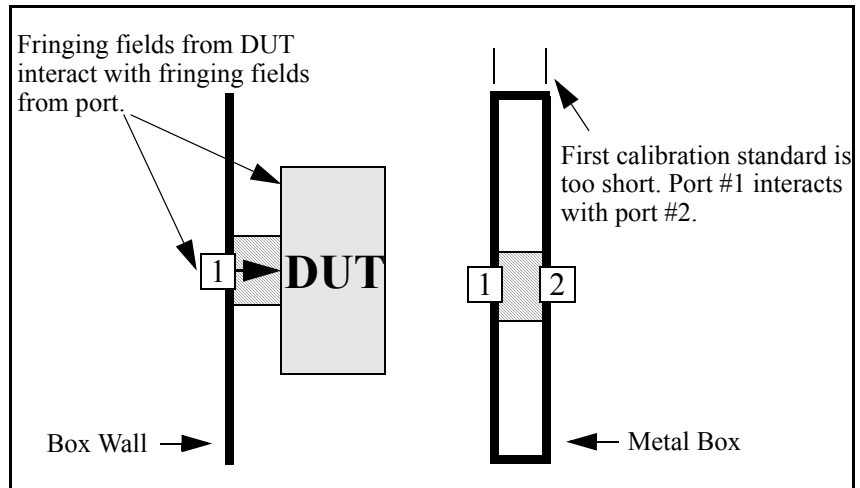
De-embedding does not require reference planes. Reference planes are optional for all box-wall and auto-grounded ports. If you do not specify a reference plane for a particular port in the project editor, **em** will assume a zero-length reference plane for that port. This means that de-embedding will remove the discontinuity associated with that particular port, but will not shift the reference plane for it.

As discussed in the next section, **em** may generate bad de-embedded results if you attempt to remove a very short (but greater than zero) reference plane. However, if you de-embed without a reference plane, **em** will not attempt to remove any length of transmission line at all. As a result, de-embedding without a reference

plane does not lead to any error. Therefore, we recommend that you de-embed without reference planes rather than specify very short, non-zero, reference plane lengths.

Reference Plane Length Minimums

If the reference plane or calibration standard is very short relative to the substrate thickness or the width of the transmission line, *em* may generate poor de-embedded results. This is due to one or both of the following reasons which are illustrated below:



Poor de-embedding results may be obtained when very short (but non-zero) reference plane lengths are used.

- **The port is too close to the device under test (DUT).** There are fringing fields associated with the port and separate fringing fields associated with the DUT. If the port and DUT are too close, the fringing fields interact. The de-embedding algorithm (which is virtually identical to algorithms used in de-embedding measured data) is based on circuit theory and cannot handle fringing field interaction. See reference [60] in Appendix II, "Sonnet References," for a detailed description of the problem.

- **The first calibration standard is too short.** In this situation, the discontinuity associated with port #1 interacts with the discontinuity associated with port #2. As a result, the first calibration standard does not “behave” like a transmission line and its S-parameters are invalid.

There is no precise rule as to how long a reference plane or calibration standard must be made in order to prevent the above effects from corrupting the de-embedded results. The required reference plane or calibration standard length is dependent upon the circuit geometry and the nature of the analysis. However, we recommend that you use reference plane or calibration standard lengths equal to or greater than one substrate thickness. This is sufficient for most types of analysis.

Reference Plane Lengths at Multiples of a Half-Wavelength

E_{eff} and Z_0 cannot be calculated when the length of the reference plane or calibration standard is an integral multiple of a half wavelength. For example, at an extremely low frequency the electrical length of the reference plane or calibration standard may be a fraction of a degree (i.e., zero half-wavelengths). In this case, the analysis is unable to accurately evaluate the electrical length and, especially, the characteristic impedance.

At some point as the length of the reference plane or calibration standard approaches a multiple of a half-wavelength, **em** is able to determine that the calculated values of E_{eff} and Z_0 are becoming corrupt. When this occurs, **em** outputs the error message “undefined: nl” in place of the E_{eff} and Z_0 values (see “De-embedding Error Codes” in online help). Note, however, that while **em** is unable to determine E_{eff} and Z_0 , the de-embedded S-parameter results are still perfectly valid.

Reference Plane Lengths Greater than One Wavelength

If the length of the reference plane or calibration standard is more than one wavelength, incorrect E_{eff} results might be seen. However, the S-parameters are still completely valid.

Em's calculation of E_{eff} is based on phase length. If the reference plane or calibration standard is, say, 365 degrees long, **em** first calculates E_{eff} based on a phase length of 5 degrees. However, **em** has some "smarts" built in. If a non-physical result is seen, **em** increases the calculated phase length by 360 degrees at a time until physical (i.e., $E_{\text{eff}} \geq 1.0$) results are obtained. This usually corrects the problem.

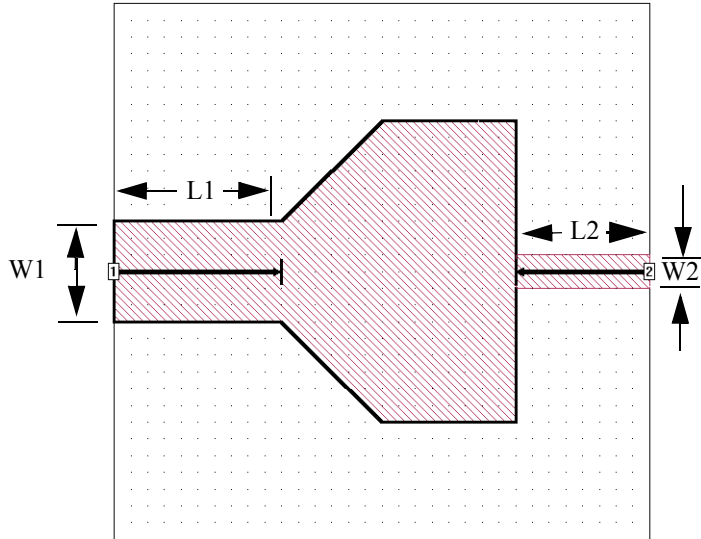
Thus, it takes a particularly long reference plane or calibration standard before the E_{eff} calculation fails. When it does fail, it suddenly jumps down to a value just above 1.0. Z_0 and the de-embedded S-parameter data still have full validity. This failure mode is rarely seen.

Non-Physical S-Parameters

Generally, reference planes should not be set in the project editor such that they extend beyond a discontinuity in the circuit. Doing so may result in non-physical S-parameters.

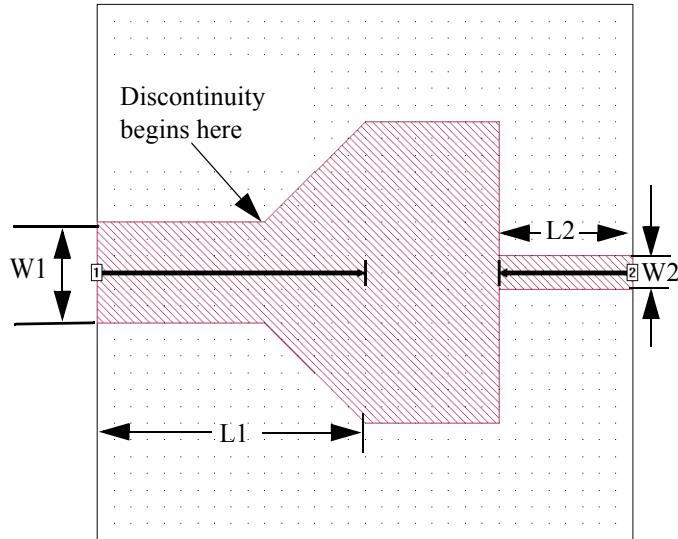
To illustrate this problem, consider the circuit shown below. In this circuit, the reference planes do not extend beyond any discontinuities. When de-embedding is enabled, the port #1 discontinuity is removed along with a transmission line of

width $W1$ and length $L1$. Similarly, the port #2 discontinuity is removed along with a transmission line of width $W2$ and length $L2$. The de-embedded result is a set of 2-port S-parameters for the block in the middle of the circuit.



Now, consider the figure on page 137. This circuit is identical to the circuit shown above except that the length of the reference plane originating on the left box wall has been increased. If *em* is run with de-embedding enabled on this circuit, it “removes” a length of transmission line equal to the specified reference plane

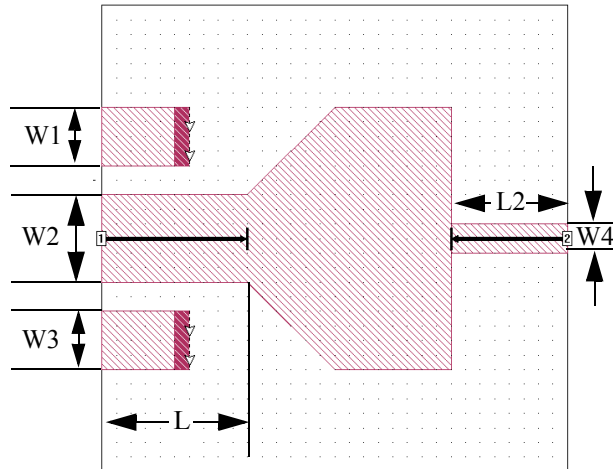
length. This occurs even though the actual port transmission line is shorter than the reference plane length. As a result, the de-embedded S-parameters are non-physical.



Example circuit for which non-physical S-parameters will be obtained when **em** is run with de-embedding enabled.

A second de-embedding example leading to non-physical S-parameter results is shown in the next figure. In this example, the circuit has two via pads on each side of the port transmission line. The via pads are grounded to the box wall.

When **em** is run with de-embedding enabled on this circuit, it “removes” three coupled transmission lines with a length equal to the reference plane length. Since the reference plane extends from the box wall beyond the vias, the de-embedded S-parameters are again non-physical.



Box Resonances

Because **em**'s de-embedding algorithm is based on circuit theory, it is unable to de-embed a structure contained inside a resonant cavity; a limit it shares with all de-embedding algorithms. Thus, whenever you wish to de-embed a circuit with box resonances, you must take the necessary steps to remove those box resonances. (See Chapter 24 for a detailed description on identifying and removing box resonances.) Note that if you do de-embed a circuit with box resonances, **em** may generate a “bd” de-embedding error code: see section “De-embedding error codes” in online help. This error code indicates that **em** has detected bad values for E_{eff} and Z_0 .

Higher Order Transmission Line Modes

De-embedding removes the port discontinuity and the connecting length of transmission line. The de-embedding assumes that there is only one mode propagating on the connecting transmission line, usually the fundamental quasi-TEM mode. If higher order modes are propagating, the de-embedded results are not valid. (The same is true for actual, physical, measurements.) If this is the case, we strongly recommend using a thinner substrate, unless, for some reason, multi-mode operation is desired.

Even when higher order microstrip modes are evanescent, there can still be problems. If the port is so close to the discontinuity of interest that their fringing (evanescent) fields interact, the de-embedding loses validity. Again, this is a problem which also arises in an actual physical measurement if the device to be de-embedded is too close to the fixture connector.

Chapter 9

Adaptive Band Synthesis (ABS)

The Adaptive Band Synthesis (ABS) technique provides a fine resolution response for a frequency band requiring only a small number of analysis points. *Em* performs a full analysis at a few points and uses the resulting internal, or cache, data to synthesize a fine resolution band.



TIP

This technique, in most cases, provides a considerable reduction in processing time.

Using the input frequency band, *em* first performs a full analysis of the circuit at the beginning and end frequencies. *Em* continues solving at discrete points, storing the full analysis data for each point. This process continues until enough internal, or cache, data is generated to synthesize a fine resolution response.

Once the frequency band response is synthesized, *em* outputs approximately 300 data points for the frequency band. These data points are a combination of the discrete analysis points and synthesized points. This combined data is referred to as adaptive data.

Em dedicates the bulk of the analysis time for an ABS analysis in calculating the response data at the discrete data points. Once the adaptive band synthesis is complete, calculating the adaptive data for the entire band uses a relatively small percentage of the processing time.

ABS Resolution

The ABS resolution is the value in frequency units between adaptive data points in your response output from an adaptive sweep. Normally, the resolution in an adaptive sweep is provided by *em* such that around 300 data points are output for a frequency band. It is possible for you to override this setting and use a coarser or finer resolution for your frequency band.

Entering a manual value to be used for ABS is done in the Advanced Options dialog box which is opened when you click on the Advanced button in the Analysis Setup dialog box. The Analysis Setup dialog box is opened when you select *Analysis* \Rightarrow *Setup* from the project editor menu. You enter the resolution by clicking on the Manual radio button in the ABS Resolution section of the Advanced Options dialog box and entering the desired resolution in the adjacent text entry box. For details on these dialog boxes, please refer to online help for the project editor.

There are several things to be aware of when using the manual setting for the ABS resolution. Coarse resolution does not speed things up. Once a rational polynomial is found to “fit” the solution, calculating the adaptive data uses very little processing time. A really coarse resolution could produce bad results by not allowing the ABS algorithm to analyze at the needed discrete frequencies. Fine resolution does not slow down the analysis unless the number of frequency points in the band is above approximately 1000 - 3000 points. A step size resulting in at least 50 points and less than 2000 points is recommended.

Q-Factor Accuracy

There is a Q-Factor analysis run option available in the Advanced Options dialog box in the project editor (Select *Analysis* \Rightarrow *Setup*, then click on the Advanced button in the Analysis Setup dialog box). Selecting this option forces a higher accuracy for ABS convergence by including the Q-factor of your analysis as a criterion for convergence. This is done to insure high accuracy in the Q-Factor result when ABS is used. The Q-Factor is defined as follows:

$$|(\text{imag } Y_{nn})/(\text{real } Y_{nn})|$$

The result is higher accuracy from the ABS sweep, but, this accuracy comes at the cost of requiring more discrete frequencies to be analyzed before conversion is reached.

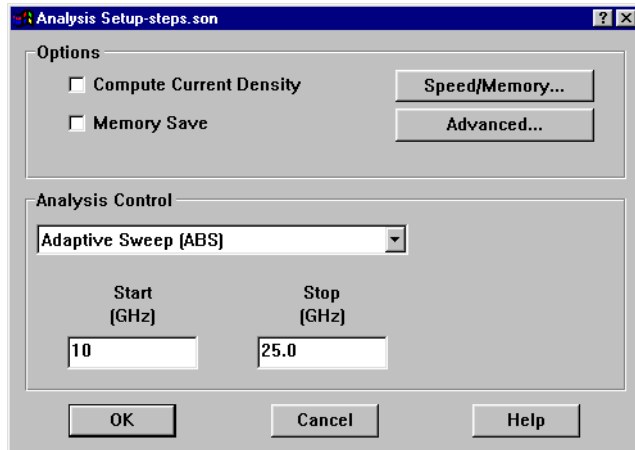
Running an Adaptive Sweep

To run an analysis using the Adaptive Band Synthesis technique, you do the following:

- 1 **Open your project in the project editor.**

2 Select Analysis \Rightarrow Setup from the main menu of the project editor.

The Analysis Setup dialog box appears on your display with an adaptive sweep already selected since Adaptive Sweep is the default for analysis control.



3 Select Adaptive Sweep (ABS) from the Analysis Control drop list if necessary.

This selects the ABS technique for the analysis.

4 Enter the desired frequency band in the Start and Stop text entry boxes.

This defines the frequency band on which you wish to perform the ABS analysis. The step size is automatically set by *em* during the analysis. See page 142 for a description of how the ABS resolution is determined.

5 Click on the OK button to close the dialog box and apply the changes.

6 Save the project by selecting *File* \Rightarrow *Save* from the menu or by clicking on the Save button on the tool bar.

You need to save the file before analyzing it.

- 7 **Select *Project* \Rightarrow *Analyze* from the menu or click on the *Analyze* button on the tool bar.**

Em performs an adaptive sweep on your project. The analysis monitor appears on your display, and indicates the progress of the adaptive sweep.

The Adaptive Sweep is also available within the Frequency Sweep Combinations analysis controls. This allows you to mix adaptive sweeps with other types of sweeps. For more details, see the “Frequency Sweep Combinations” topic in online help in the project editor.

ABS Caching Level

There are three levels of ABS caching available: None, Stop/Restart, and Multi-Sweep plus Stop/Restart. The options for ABS caching level are found in the Advanced Options dialog box. To access the Advanced Options dialog box, select *Analysis* \Rightarrow *Setup* from the project editor main menu, then click on the Advanced button in the Analysis Setup dialog box which appears. The default is Stop/Restart.

Stop/Restart retains the cache data while the analysis is proceeding. Once the adaptive data has been calculated, the cache data is deleted from the project. This setting provides for the circumstance in which the analysis is stopped or interrupted before the adaptive data is synthesized; you will not lose the internal data produced so far.

Multi-Sweep with Stop/Restart retains all calculated cache data in your project for every analysis job run. In addition, cache data is calculated and saved for even non-ABS types of sweeps. This option can reduce processing time on subsequent ABS analyses of your project but increases project size on non-ABS sweeps. In order for the cache data to remain useful there are also subsectioning issues of which you must be aware. For a detailed discussion of the Multi-Sweep cache option, please refer to “Multiple ABS Sweeps and Subsectioning” on page 146.

The third setting for ABS caching level is None. In this setting cache data is not retained. This option should only be selected if you have constraints on disk space.



WARNING

If you select None for the ABS caching level, and an ABS sweep is stopped before the adaptive data has been calculated, you will have to start the analysis over from the beginning. Any processing time invested in the analysis is lost.

Multiple ABS Sweeps and Subsectioning

If you will need to run multiple ABS sweeps on a project, it is important to set your ABS caching to Multi-sweep to avoid having to re-calculate your caching data each time you analyze your circuit. But be aware that in order to maintain the validity of the caching data, the subsectioning of the circuit must remain the same. To control the subsectioning you must use the Advanced Subsectioning Controls which you open by selecting *Analysis* \Rightarrow *Advanced Subsectioning* from the main menu of the project editor.



TIP

The most efficient way to obtain response data for your circuit is to run a single ABS sweep over the entire desired frequency band.

The analysis engine, *em*, uses the subsectioning frequency to calculate the wavelength which is used in setting the Maximum subsection size. The default setting used to determine the subsectioning frequency is to use the highest frequency from the present analysis job. If you perform multiple sweeps over different frequency bands then the cache data from one run will be invalid for the next, since the subsectioning frequency would be different. In order to avoid this you should select the **Previous Analysis Only** option which will use the highest frequency from all previous analysis jobs run on the project. In this case, you should analyze the frequency band with the highest upper limit first and take care

to ensure that the subsectioning frequency being used provides accurate subsectioning for your circuit. For details on subsectioning, see Chapter 4, “Subsectioning”.

Another way to keep the subsectioning frequency consistent is to select the Use Fixed Frequency option for the subsectioning frequency and enter the desired frequency. This ensures that all analysis runs on the project will use the same subsectioning frequency. Again, care should be taken that the subsectioning frequency entered provides the desired accuracy.

Multi-Sweep Caching Scenarios

The analysis engine always attempts to use any existing data in the project which is consistent with the present analysis. Described below are some common scenarios describing ABS analyses when the ABS caching level is set to Multi-Sweep with Stop/Restart and how data consistency is maintained.

Higher or Lower Resolution over the Same Frequency Band: You are running an ABS analysis over the same band as a previous ABS analysis but with higher or lower resolution; an example is shown below. In order for the caching data to be valid for the second analysis, your Advanced Subsectioning controls must be set such that the subsectioning frequency is the same for both runs. If the subsectioning frequency remains the same, the second analysis will usually not require any re-analysis and the results should be provided very quickly. The only exception would be if the difference between the resolutions is unusually high.

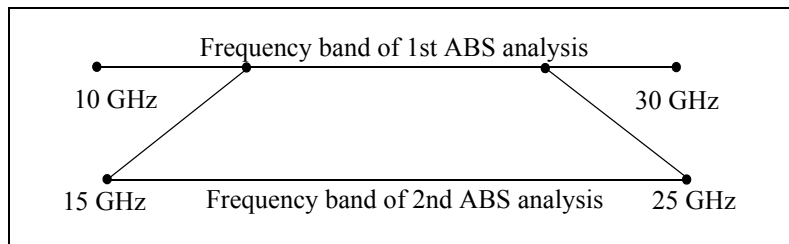
Frequency Band 10 - 40 GHz

1st ABS analysis: 10, 10.1, 10.2, 10.3 ... 39.8, 39.9, 40

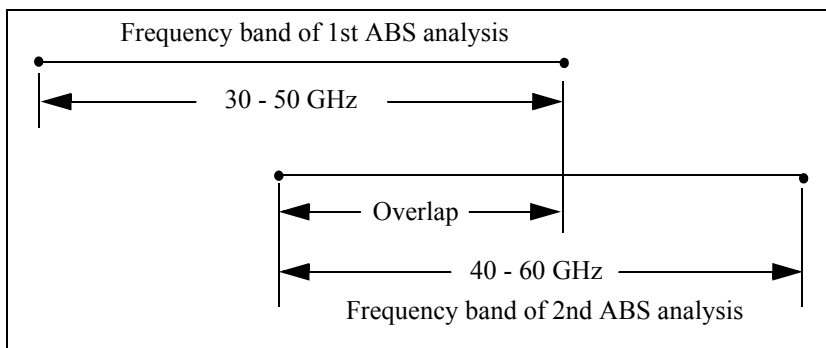
2nd ABS analysis: 10, 10.05, 10.1, 10.15, 10.2 ... 39.85, 39.9, 39.95, 40

Zoom In: You are running an ABS analysis over a narrower band than the previous ABS analysis of the project, as shown in the diagram below. This provides higher resolution over the narrower band since the ABS analysis defaults to approximately 300 data points. In order for the caching data to be valid for the second analysis, your Advanced Subsectioning controls must be set such that the

subsectioning frequency is the same for both runs. If the subsectioning frequency remains the same, the second analysis will not require any re-analysis and the results should be provided very quickly.



Extending the Band: You are running an ABS analysis which overlaps a previous ABS analysis of the project, pictured in the diagram below. The caching data for the overlap between the two analyses will be reused although some calculation may need to be done in the extension of the frequency band where it does not overlap with a previous analysis. In order for the caching data to be valid for the second analysis, your Advanced Subsectioning controls must be set such that the subsectioning frequency is the same for both runs.



Accuracy Assurance: If you wish to check a particular data point in an ABS analysis and wish to ensure that a full calculation is done at a particular frequency point, you should select a Linear Sweep. This analysis will calculate caching data if Multi-sweep is selected for ABS caching data, but will not use the caching data in producing analysis results.

Find Minimum and Find Maximum

Find Minimum determines the frequency where the circuit response reaches a minimum. Find Maximum determines the frequency where the circuit response reaches a maximum. You enter a starting and ending frequency in the Start and Stop text entry boxes, respectively and select the parameter for which you wish to determine the minimum or maximum value. **Em** performs an ABS analysis for the frequency band, then uses the adaptive data to determine the frequency where the response reaches a minimum or a maximum.

The Find Minimum and Find Maximum commands are both available in the Frequency Sweep Combinations analysis controls. For more details, see the “Frequency Sweep Combinations” topic in online help in the project editor.

Parameter Sweep

You may choose either a linear sweep or an adaptive sweep for a parameter sweep. Selecting an adaptive sweep for a parameter sweep is done in the Parameter Sweep Entry dialog box. For more information about parameter sweeps, please see “Parameter Sweep,” page 164.

The following example assumes that you have already defined the parameter “Width” in your circuit. For more information on inputting a parameter, please refer to “Parameters,” page 156.

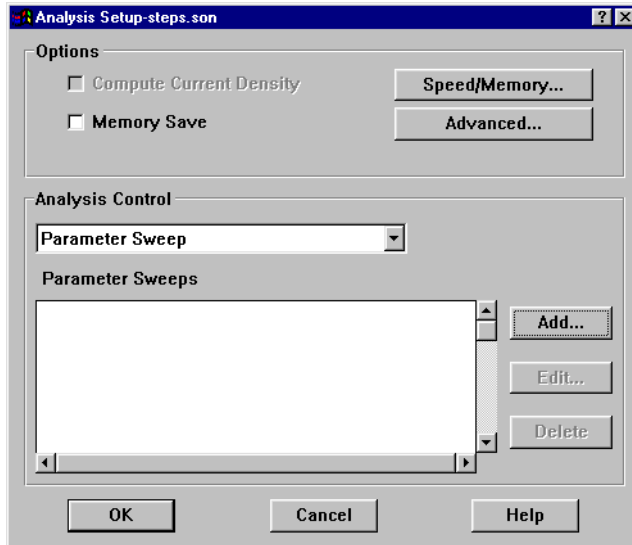
To select ABS for a parameter sweep, do the following:

- 1 **Select Analysis ⇒ Setup from the project editor main menu.**

The Analysis Setup dialog box appears on your display.

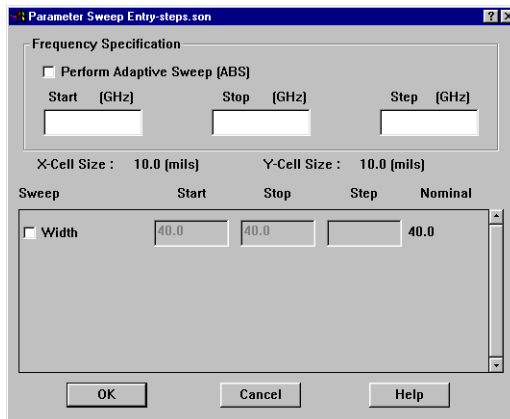
2 Select Parameter Sweep from the Analysis Control drop list.

The dialog box is updated to allow you to specify the parameter sweep.



3 Click on the Add button to the right of the Parameter Sweep list box.

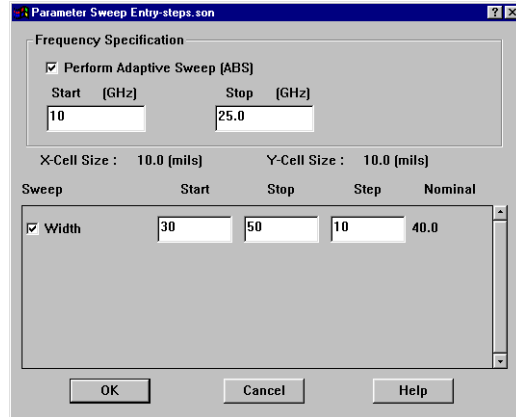
The Parameter Sweep Entry dialog box appears on your display. The default frequency specification is a linear sweep.



- 4 Click on the Perform Adaptive Sweep (ABS) checkbox to select an adaptive frequency sweep.

The dialog box changes so that there are only Start and Stop text entry boxes.

- 5 Enter the frequency band for the ABS in the Start and Stop text entry boxes.



The image shows a dialog box titled "Parameter Sweep Entry-steps.son". It has a "Frequency Specification" section with a checked checkbox "Perform Adaptive Sweep (ABS)". Below this are two text entry boxes: "Start [GHz]" with the value "10" and "Stop [GHz]" with the value "25.0". Below these are two more text entry boxes: "X-Cell Size : 10.0 (mils)" and "Y-Cell Size : 10.0 (mils)". Below these are four more text entry boxes: "Sweep" with a checked checkbox "Width", "Start" with the value "30", "Stop" with the value "50", "Step" with the value "10", and "Nominal" with the value "40.0". At the bottom are three buttons: "OK", "Cancel", and "Help".

This completes setting up an ABS frequency sweep for the parameter sweep. You would also need to select the parameters which you want to use in the parameter sweep and enter their data ranges before closing this dialog box.

Analysis Issues

There are several issues you should be aware of before using the ABS technique. These are covered below.

Multiple Box Resonances

You should be aware that circuits with multiple box resonances make it difficult for an ABS analysis to converge. The frequency band for an adaptive sweep should not contain multiple box resonances. If multiple box resonances are present the number of discrete full analysis points goes up dramatically and

synthesis of the data becomes very difficult. If you do not know how to identify box resonances, see Chapter 24, "Package Resonances" for a detailed discussion of box resonances.

De-embedding

Adaptive data, resulting from an ABS analysis, is either de-embedded or non-de-embedded. With other analysis types, when the de-embedding option is enabled (default), then both de-embedded and non-de-embedded response data is calculated and available for display and output. This is not true for an adaptive sweep.

In an adaptive sweep, if you run with de-embedding enabled, de-embedded data is available for the whole band. Non-de-embedded data is available only for the discrete data points at which full analyses were performed while synthesizing the response.

If you wish to have non-de-embedded data for the whole frequency band, you must perform an adaptive sweep with the de-embed option disabled. Select *Analysis* \Rightarrow *Setup* from the main menu to open the Analysis Setup dialog box, then click on the Advanced button to open the Advanced Options dialog box. Click on the De-embed checkbox to disable de-embedding. For details on these dialog boxes, please refer to online help for the project editor.

For more information about de-embedding, see Chapter 7, "De-embedding" and Chapter 8, "De-embedding Guidelines".

Transmission Line Parameters

As part of the de-embedding process, *em* also calculates the transmission line parameters, Z_0 and E_{eff} . You should be aware that when running an ABS analysis these parameters are only calculated for the discrete data points at which a full analysis is run. If you need the transmission line parameters at more data points, analyze the circuit using a non-ABS analysis.

Current Density Data

Current density data is calculated for your circuit when the Compute Current Density option is enabled in the Analysis Setup dialog box. For non-ABS sweeps, current density data is calculated for all the response data. For an adaptive sweep, the current density data is only calculated for the discrete data points, therefore, your plot in the current density viewer shows a coarse resolution of your frequency band.

If you wish to calculate the current density data at more points in your band, run a non-ABS sweep for the points in question with the Compute Current Density option enabled.

For more information about the Compute Current Density option, see the help topic “Analysis \Rightarrow Setup” in online help for the project editor.

Ripple in ABS S-Parameters

Please note that when the value of the S-parameters is close to 1 (0 dB) over the entire band you may have small ripples or oscillations in the S-parameter values. This is due to the rational fitting model having too many degrees of freedom when trying to fit a straight line. If this is a problem, it is recommended that you analyze the frequency band in which this occurs with another type of sweep.

Output Files

You specify additional output files in the Output Files dialog box which appears on your display when you select *Analysis \Rightarrow Output Files* from the project editor menu. You click on the appropriate button to open the corresponding file entry dialog box. Each entry dialog box has an option pertinent to ABS.

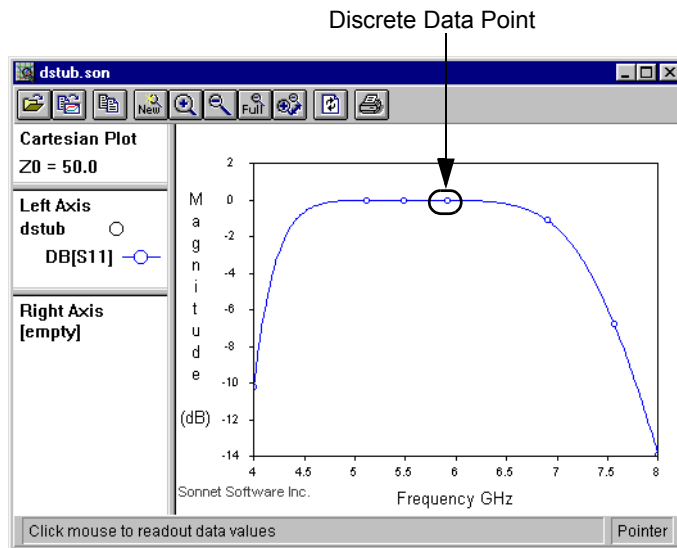
Response File

When you specify an optional output file for your project, you may select which type of data to output from an adaptive sweep. The data selection is controlled by the Include Adaptive Data checkbox in the File Entry dialog box. If this checkbox is selected, which is the default, then all the adaptive data from an ABS analysis

is included in the output file. If this checkbox is cleared, then only the data for the discrete data points is included in the output file. We recommend leaving this box checked.

Viewing the Adaptive Response

When viewing an adaptive response (ABS) in the response viewer there are several things of which you need to be aware. The adaptive data is plotted as a line. A symbol indicating a data point only appears at the discrete frequencies at which a full analysis was executed as shown in the picture below.



When exporting data, you may choose to output only the discrete frequencies or the complete response data for the ABS analysis. To output only the discrete frequency data, unselect the Include Adaptive Data checkbox in the Export Data dialog box in the response viewer. For details, see online help in the response viewer.

Chapter 10 Parameterization and Optimization of a Geometry Project

It is often necessary to perform several iterations of a circuit design in order to meet specifications. Part of those iterations involves changing the dimensions or attributes of some part of your circuit. Parameterization and optimization can be used to make this task more efficient. Rather than create multiple circuits each with different lengths of key components, you may select dimensions of your circuit and define them as a parameter. In the analysis, you automatically vary the value of the parameter rather than setting up a separate circuit file for each value. Parameterizing your circuit also provides you with a way to quickly and easily change dimensions in the project editor.

For a tutorial which details how to add parameters to your circuit and perform an optimization, please refer to Chapter 2, “Parameter Sweep and Optimization Tutorial” in the **Sonnet Supplemental Tutorials**.

The analysis engine, *em*, adjusts the value of a parameter in one of two ways. The first is a parameter sweep in which *em* sweeps the parameter values through a user defined range. The second method is optimization in which the analysis engine controls the parameter value, within a user defined range, in an attempt to reach a user defined goal. Both parameterization and optimization of a geometry may be performed over a range of analysis frequencies.

The first step in performing a parameter sweep or optimization is defining the parameters in the project editor.

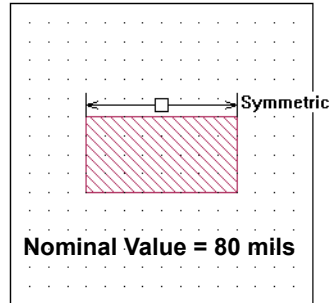
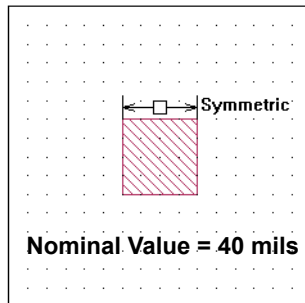
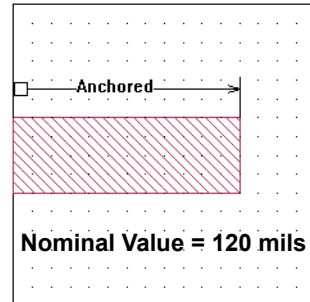
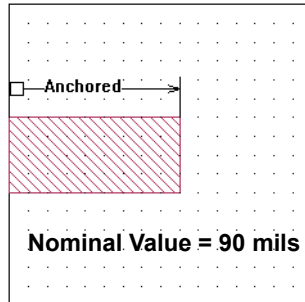
Parameters

Parameters are user defined circuit attributes that allow the analysis engine to modify the circuit in order to perform parameter sweeps and optimization. Parameters also provide a quick way for the user to change dimensions in the project editor. For example, the length of a transmission line can be assigned the parameter "L". To change the length of the transmission line, you edit the value for "L."

In order to set up parameters, you must first input your circuit in the project editor. You select dimensions and define them as parameters, which allows you to vary those dimensions within an analysis. The initial value of the parameter is the length that appears in your circuit. This is the nominal value of the parameter. If you change the nominal value then the circuit is redrawn with that length.

Chapter 10 Parameterization and Optimization of a Geometry Project

There are two types of parameters: anchored and symmetric. An anchored parameter allows you to fix one end of a parameter then vary its length extending from that point. A symmetric parameter allows you to fix the center point of a parameter and vary the distance it extends on each side. The two parameter types are described in detail in the following sections.



Anchored Parameters

An anchored parameter is one which extends from a fixed point, the anchor, to a adjustable point set. It is comprised of three things: the anchor, the reference point and the adjustable point set. When defining your parameter, you perform the following steps:

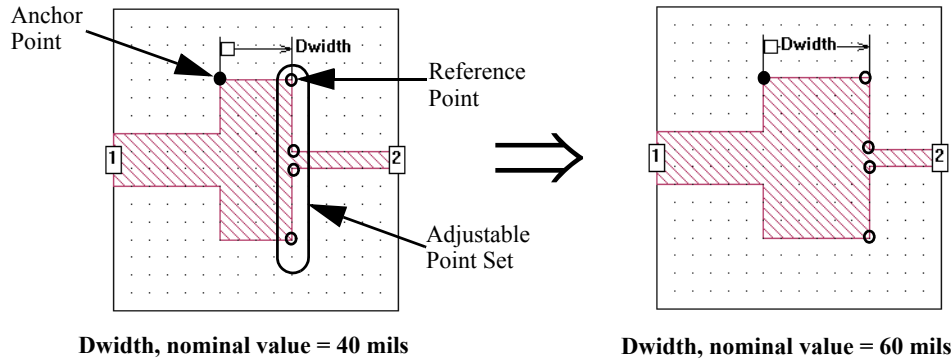
- You select the anchor first. This is the fixed starting point for the parameter.
- You select the reference point. The reference is the first point in the

adjustable point set. The distance from the anchor point to the reference point is the value of the parameter. When the value of the parameter is changed, the anchor retains the same position, but the reference point moves to a new position.

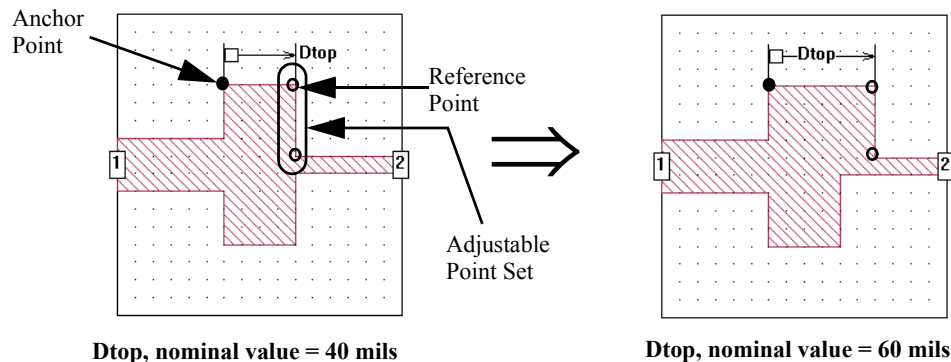
- Third, you select any additional points in your circuit you wish to move when the reference point moves. As the value of the parameter is varied, the reference point, as well as the rest of the adjustable point set, is moved accordingly.

Note that the parameter is always defined as the distance between the anchor and the reference point in either the X direction or the Y direction, never as a diagonal distance between them.

Two examples of anchored parameters, each at two different nominal values, are illustrated below.



Notice that although the top and bottom examples have identical anchor and reference points and starting and ending nominal values, that the resulting polygon on the top differs from that on the bottom due to a different adjustable point set (the point set is highlighted by the oval).



Symmetrical Parameters

A symmetric parameter is one which extends from a fixed middle point to two reference points and their respective adjustable point sets. It is comprised of four things: the first reference point, the first reference point's adjustable point set, the second reference point and the second reference point's adjustable point set. The fixed middle point is defined as the midpoint between the two reference points

when the parameter is defined; the user does not define an anchor point for a symmetric parameter. When defining your parameter, you perform the following steps:

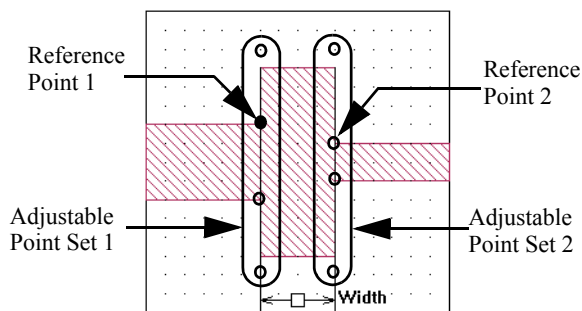
- Select the first reference point.
- Select the adjustable point set that moves with the first reference point.
- Select the second reference point. The value of the parameter is the distance between the two reference points.
- Select the adjustable point set that moves with the second reference point.

When the value is changed, each point set moves one half the distance of the difference between the present value and the previous value out from the middle point.

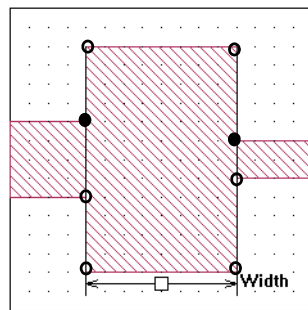
Note that the parameter is always defined as the distance between the reference points in either the X direction or the Y direction, never as a diagonal distance between them.

Chapter 10 Parameterization and Optimization of a Geometry Project

Two examples of symmetric parameters, each at two different nominal values, are illustrated below.

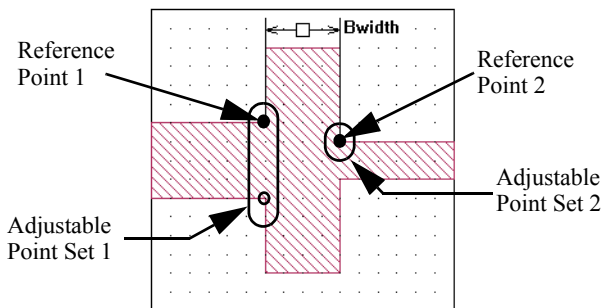


Width, nominal value = 40 mils

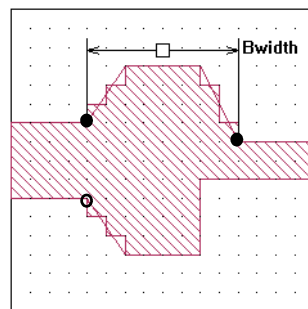


Width, nominal value = 80 mils

Notice that although the top and bottom examples have identical reference points and starting and ending nominal values, that the resulting polygon on the top differs from that on the bottom due to a different adjustable point set (the point set is highlighted by the oval).



Bwidth, nominal value = 40 mils

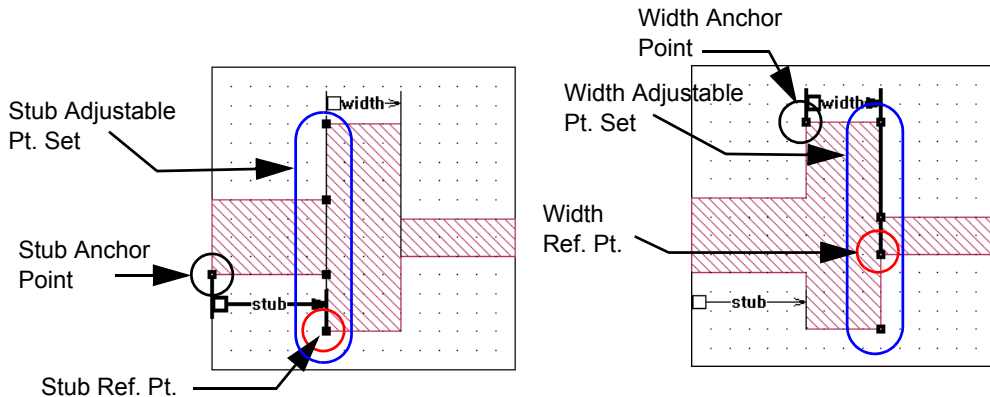


Bwidth, nominal value = 80 mils

Dependent Parameters

One parameter is dependent upon another if the anchor point and/or the reference point(s) for the second parameter are part of an adjustable point set for the first parameter. You need to be aware of dependent parameters so that you can take into consideration the complete impact on your circuit when the value of the

primary parameter is changed. When the primary parameter is changed, a dependent parameter is adjusted, i.e., the anchor point or reference point is moved, along with the primary parameter on which they depend. A picture of a dependent parameter is shown below with the Anchor and Reference points highlighted as well as the point sets.



The parameter “width” is dependent on the parameter “stub”. Note that the anchor point for the parameter “width”, shown on the right, is part of the adjustable point set for the parameter “stub”, shown on the left.

You may use the Select Dependents command in the project editor to determine if there are any dependent parameters. With the project open in the project editor, do the following:

- 1 Right-click on a parameter in your circuit.**

The parameter name and its point set is highlighted and a pop-up menu appears on your display.

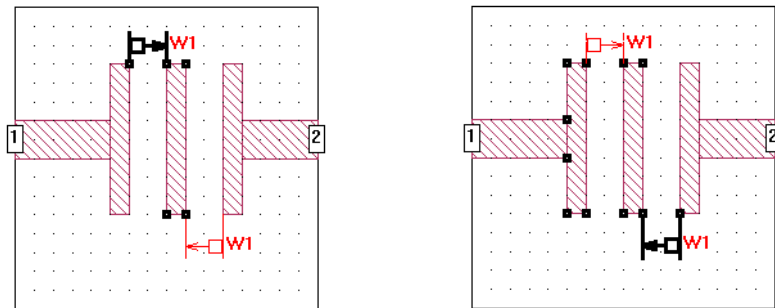
- 2 Select “Select Dependents” from the pop-up menu.**

If any dependent parameters are present in your geometry, the dependent parameter is highlighted.

Circular Dependencies

Care should be taken when adding parameters to your circuit that they do not form a circular dependency. A circular dependency is formed when two parameters are dependent on each other. This can happen for two parameters or multiple parameters. In the case of multiple parameters, the dependency extends from the first parameter through all the parameters until the first parameter is dependent upon the last.

If the project editor detects a circular dependency, all the involved parameters are shown in red and their nominal values are fixed. An example of two parameters in a circular dependency is shown below.



A circular dependency is an error condition and must be corrected before you can analyze your circuit. As long as the circular dependency exists, you are not able to use a value for the parameter other than the nominal value which makes the parameter useless. You need to redefine one of the parameters such that it is no longer dependent on the other.

Reference Planes

Special care should be taken with reference planes when parameterizing your circuit. If a reference plane is linked to a point on your circuit and that point is selected as part of an adjustable point set, the reference plane changes length in response to changes in the parameter length.

Parameter Sweep

Once you have parameterized your circuit, you may use the parameters to perform a parameter sweep. A parameter sweep allows you to run through a set of analyses, with different parameter values, all in one step. This allows you to see how changes in your circuit affect the response of your circuit.

Setting up a parameter sweep consists of three parts: specifying analysis frequencies, choosing which parameters to vary and specifying data ranges for the chosen parameters.

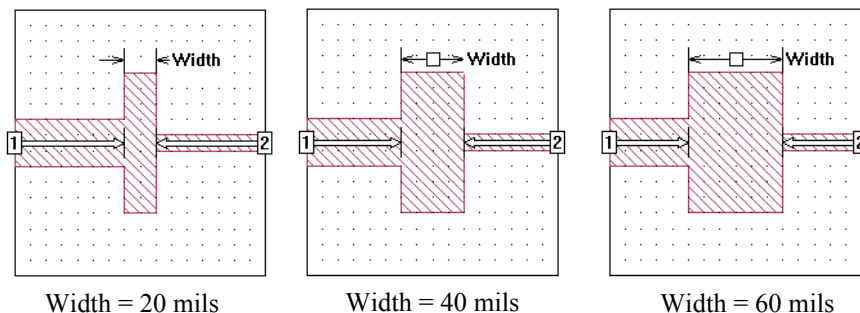
The analysis frequencies for a parameter sweep are defined as a linear sweep consisting of a starting frequency, an ending frequency and the interval between analysis frequencies. This frequency set is used for each parameter value.

You may select one or multiple parameters when running a parameter sweep. For each parameter that you select you must specify a starting value, an ending value and the interval between parameter values.

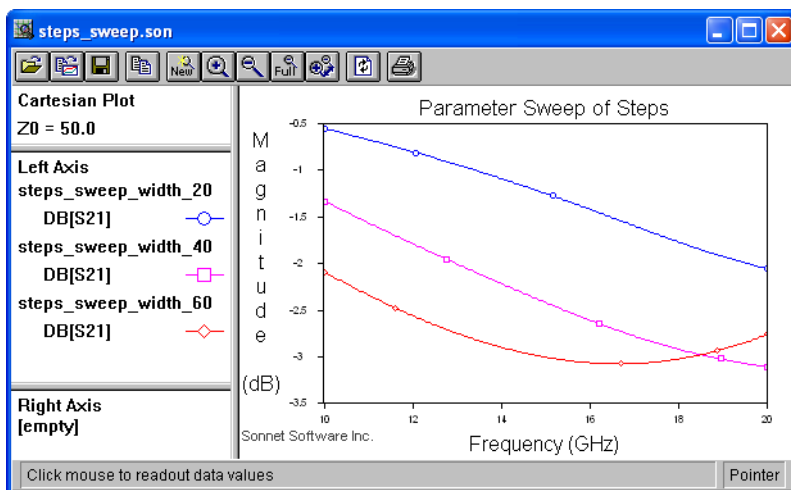
You set up the parameter sweep in the Analysis Setup dialog box in the project editor. To access this dialog box, select *Analysis* \Rightarrow *Setup* from the main menu of the project editor. For details on using this dialog box, please refer to online help.

Chapter 10 Parameterization and Optimization of a Geometry Project

For example, the graphics below illustrate a parameter sweep of the circuit “steps” with a single parameter “Width” defined. The parameter sweep starts with a value of 20 mils for width and increases in steps of 20 until the parameter’s value is 60. *Em* automatically performs an analysis at each specified frequency for each circuit shown below when the parameter sweep is executed.



In the case of this sweep, an ABS analysis from 10 - 20 GHz was performed. The response data for the parameterization is shown in the graph below.



Since an analysis of the circuit at each combination of parameter values is executed for each specified analysis frequency, care should be taken when choosing data ranges. The higher the number of analysis frequencies and parameter values, the higher the number of analyses that must be computed by *em*. The number of combinations specified is displayed in the project editor.

Optimization

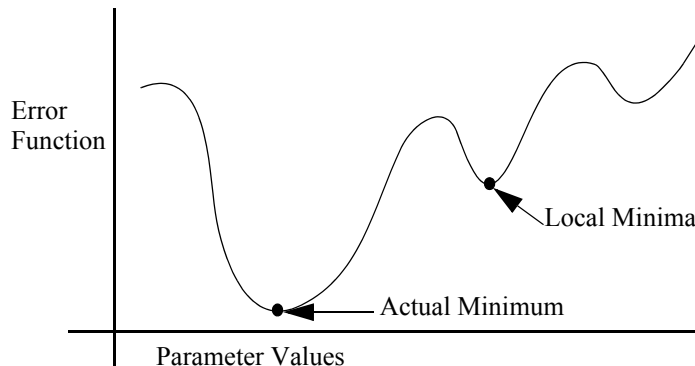
You may also use parameters to perform an optimization on your circuit. An optimization allows you to specify goals - the desired response of your circuit - and the data range for the parameter(s) over which you seek the response. The software, using a conjugate gradient method, iterates through multiple parameter values, searching for the best set which meets your desired goals.

The conjugate gradient optimizer begins by analyzing the circuit at the nominal parameter values. It then perturbs each parameter individually, while holding the others fixed at their nominal values, to determine the gradient of the error function for that parameter. Once it has perturbed each parameter, it then performs a line search in the direction of decreasing error function for all parameters. After some iterations on the line search, the optimizer again calculates the gradients for all parameters by perturbing them from their present “best” values. Following this, a new line search is performed. This continues until one of three conditions are met: 1) the error goes to zero 2) the error after the present line search is no better than the error from the previous line search 3) the maximum number of iterations is reached. When one of these three conditions is met, the optimizer halts.

Setting up an optimization consists of five parts:

- Specifying optimization frequencies
- Specifying your goals
- Choosing which parameters to vary
- Specifying data ranges for the chosen parameters
- Specifying the maximum number of iterations

Care should be taken when setting the nominal values for the parameters to be optimized. The optimizer starts at the nominal values and converges to the minima which is closest to those nominal values. Thus, it is highly recommended that you perform some pre-analysis prior to doing the optimization to ensure that the nominal values are in the right value range when the optimizer is started. Otherwise, the optimizer may converge to a local minima for which the error is not the minimum achievable value, as pictured below.



You specify a goal by identifying a particular measurement and what value you desire it to be. For example $S_{11} < -20$ dB. Keep in mind that the goals you specify may not be possible to satisfy. **Em** finds the solution with the least error.

You may also specify a goal by equating a measurement in one network to a measurement in another network or file. For example, you may set S_{11} for network “Model” equal to S_{11} for network “Measured.” Likewise, you may equate S_{11} for network “Model” to S_{11} for data file “meas.s2p.”

You may select one or multiple parameters to optimize. For each parameter that you select you must specify minimum and maximum bounds. The analysis limits the parameters to values within the specified bounds.

You specify the number of iterations. For each iteration, **em** selects a value for each of the parameters included in the optimization, then analyzes the circuit at each frequency specified in the goals. Depending on the complexity of the circuit, the number of analysis frequencies and the number of parameter combinations, an

optimization may take a significant amount of processing time. The number of iterations provides a measure of control over the process. Note that the number of iterations is a maximum. An optimization can stop after fewer iterations if the optimization goal is achieved or it finds a minima (finds no improvement in the error in further iterations).

Once the optimization is complete, the user has a choice of accepting the optimal values for the parameters resulting from the *em* analysis. Note that if the results of the optimization are accepted- used as the nominal values for the parameters- the actual metalization in the project editor is the closest approximation which fits the present grid settings. As a matter of fact, *em* analyzes “snapped” circuits and interpolates to produce responses for circuits which do not exactly fit the grid. For more information about the grid, see Chapter 4, “Subsectioning”.

Chapter 11

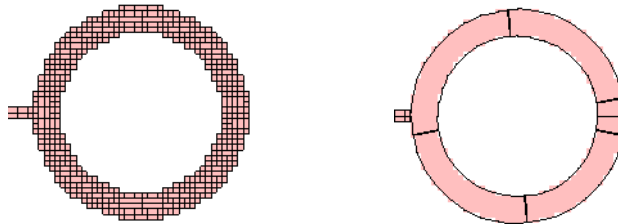
Conformal Mesh

Introduction

Analyzing circuits which have non-rectangular polygons can require extensive memory and processing time since the number of subsections needed to model the non-rectangular shapes is significantly higher than the number of subsections required for a rectangular polygon. Conformal meshing is a technique which can dramatically reduce the memory and time required for analysis of a circuit with diagonal or curved polygon edges.

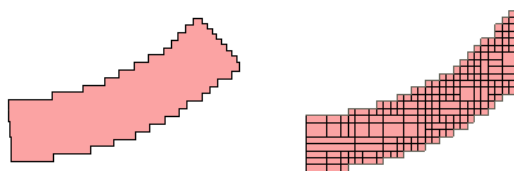
This technique groups together strings of cells following diagonal and curved metal contours. Whereas staircase fill results in numerous small X- and Y-directed subsections, conformal mesh results in a few long conformal subsections. Fewer subsections yields faster processing times with lower memory requirements for your analysis.

In older meshing techniques, large non-rectangular subsections did not include the high concentration of current on the edge of the lines required by Maxwell's equations. The results could significantly under-estimate loss and inductance. In contrast, the Sonnet conformal meshing automatically includes the high edge current in each conformal section. In conformal meshing, Sonnet can achieve the speed of using large subsections, and at the same time enjoy the accuracy of using small cells. This patented¹ Sonnet capability is unique.



Pictured above is a typical circuit which would be appropriate for Conformal Meshing. The left picture shows the rectangular subsections created by using staircase fill. This results in approximately 800 subsections (unknowns). The right picture shows the conformal sections created by using conformal fill, resulting in only about 130 subsections. Note that each conformal section shown represents multiple subsections.

Conformal sections, like standard subsections, are comprised of cells, so that the actual metalization still shows a “jagged” edge when the polygon has a smooth edge, as pictured below. However, the sections can be much larger due to conformal meshing. You may now make the underlying grid sufficiently small to accurately resolve challenging circuit dimensions without incurring excessive memory and analysis time requirements.



¹U.S. Patent No. 6,163,762 issued December 19, 2000.

Conformal meshing should be used in places where it will reduce subsection count. For rectangular polygons with no diagonal or curved edges, it is more efficient to use rectangular subsections (default). However, if a polygon contains a curved edge, conformal meshing provides a quicker analysis.

For a discussion on subsectioning when using Conformal Mesh, see "Conformal Mesh Subsectioning" on page 69.

Use Conformal Meshing for Transmission Lines, Not Patches

Conformal meshing assumes most of the current is flowing parallel to the edge of the conformal subsection. This works well for transmission lines. However, this is usually not accurate for geometries like patch antennas. For large areas of metal in both x and y directions, high current can flow parallel to the X axis edges, and parallel to the Y-axis edges at the same time. Conformal meshing can include only one of these currents. Thus, conformal meshing should only be used for transmission line geometries, which have a “line width” that is small compared to wavelength.

Applying Conformal Meshing

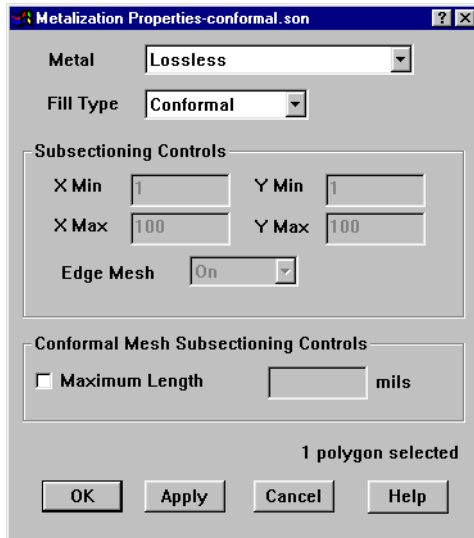
Conformal Meshing is applied as a property of a metal polygon. To use conformal meshing for a polygon, do the following:

- 1 Select the desired polygon(s) by clicking or lassoing.**

The selected polygons are highlighted.

2 Select Modify ⇒ Metal Properties from the main menu.

The Metal Properties dialog box appears on your display.

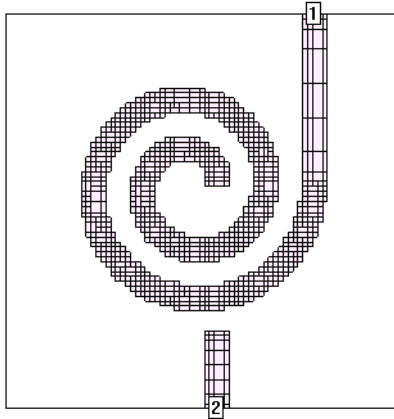


3 Select "Conformal" in the Fill Type drop list.

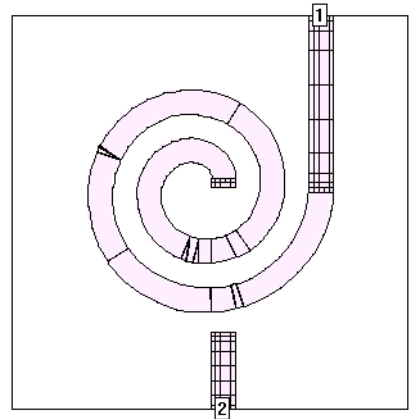
4 Click on the OK button to apply the changes and close the dialog box.

The polygon does not appear any different in the circuit. To see the difference, you need to use the *Analysis ⇒ Estimate Memory* command. When the Estimate Memory dialog box appears, click on the View Subsections button. Shown below is the subsectioning for the same spiral inductor. The circuit on the left uses rectangular subsections and the one on the right uses conformal subsections. Note

that the rectangular subsectioning uses a much higher number of subsections for the spiral inductor than does the conformal meshing. Rectangular subsectioning was used for the feed lines in both cases.



Spiral inductor with rectangular subsections. (Default)



Spiral inductor with conformal subsections.

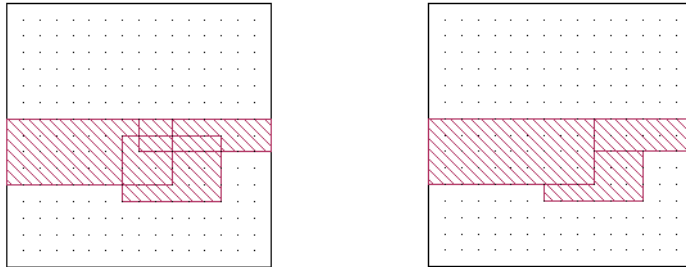
If you chose Conformal meshing, then the subsectioning controls in the Metalization Properties dialog box - Xmin, Ymin, XMax, YMax and Edge Mesh - are disabled and ignored.

Conformal Meshing Rules

Since conformal meshing is a new feature in the analysis engine, not all conditions which may affect accuracy or processing time are automatically identified in the project editor. Below are some basic rules for using conformal meshing you may follow to prevent causing an error in the analysis engine, *em*.

Rule 1: Polygon Overlap

Polygons should be drawn or moved in your circuit such that there is no overlap between polygons if any one of the polygons is using conformal meshing. It is possible for two polygons to overlap and not cause an error condition, but the most conservative use would be no overlaps. See the illustration below.

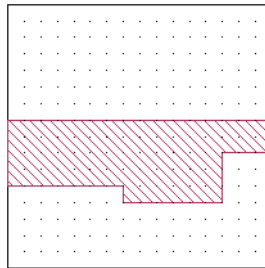


The circuit on the left has three overlapping polygons and the polygon on the bottom is using conformal meshing. This would cause *em* to issue an error message and stop running. The circuit shown on the right has no overlap between polygons and would not cause any errors.



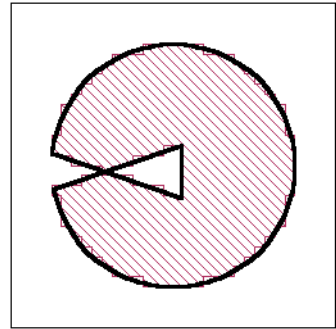
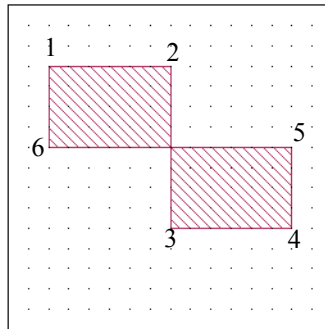
TIP

To maintain the same metal in your circuit without any overlap, use the *Edit* \Rightarrow *Merge Polygons* command on polygons which use the same metal type. Using the Merge command on the example above, in which all three polygons are the same metal type, is shown below.



Rule 2: Figure Eight Polygons

A conformal mesh polygon should not wrap back around itself; in other words, its vertices should not form a figure eight. This will result in an error message being issued and **em** will stop the analysis job. Two examples of this type of polygon are shown below. In the polygon shown on the left, the vertices have been labelled in the order in which they were added.



Rule 3: Adjacent Polygons Should Have No Gap

Any polygon which is adjacent to another polygon using conformal meshing should have its edges exactly touching with no gap existing between the two polygons. Extremely tiny gaps are automatically removed, but should be avoided. Tiny gaps can be easily avoided by using any of the following methods:

- Using a snap grid while drawing your circuit. (Tools \Rightarrow Snap Setup)
- Creating a larger polygon, then dividing the polygon (Edit \Rightarrow Divide Polygons) and applying conformal meshing to one of the resulting polygons. The resultant polygons are adjacent with no space in between.
- Adding a small polygon which bridges the gap and overlaps the two polygons on either side of the gap, then using the Merge Polygon command (Edit \Rightarrow Merge Polygons).

- Snapping the existing polygons to the grid. (Modify \Rightarrow Snap To command using the Preserve shape and spacing option).

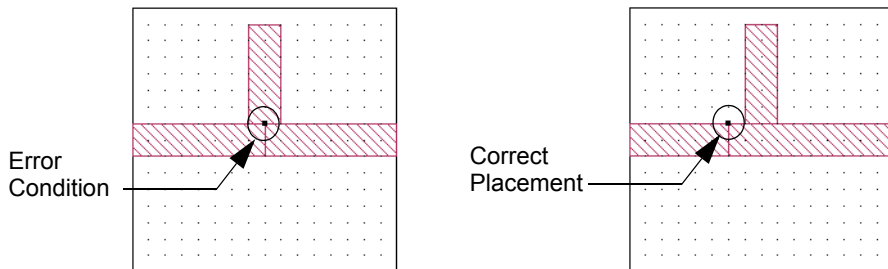


WARNING

If you are snapping a circuit with curved edges, use the Preserve shape and spacing option in the Snap Objects dialog box. Otherwise, curved edges can become distorted and are difficult to restore in the project editor.

Rule 4: Adjacent Polygons Should Not Have an Interior Vertex

When three polygons are adjacent, a vertex where two polygons meet should not occur between two vertices of the third polygon. See the illustration below.



Memory Save Option

It is recommended that memory save not be enabled when your circuit has polygons using conformal mesh fill. This is because conformal mesh subsections are sensitive to precision error. Since using the memory save option involves reducing the required memory at the expense of increasing precision error, its use may lead to noisy S-parameter results for circuits with conformal mesh fill.

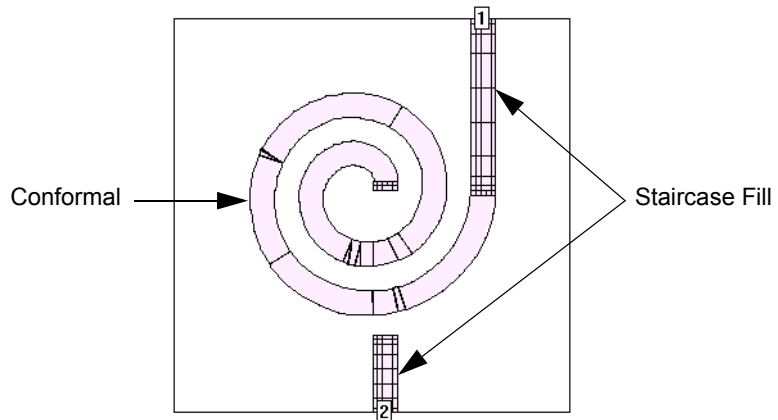
Using Conformal Meshing Effectively

This section discusses some guidelines to use in order to get the most improvement in processing time and memory use and the most accurate results when using conformal meshing. Following these guidelines will help you to use conformal meshing in the most efficient manner.

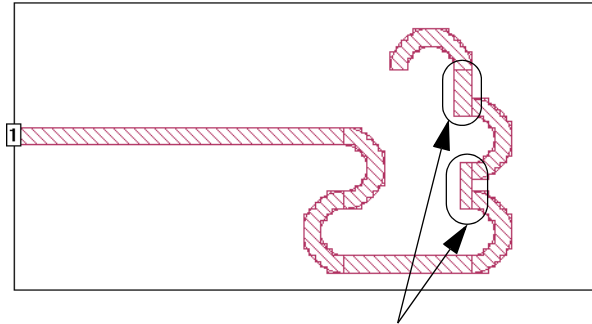
Use Conformal Meshing for Non-Manhattan Polygons

Conformal meshing should be used for non-Manhattan polygons. Manhattan polygons are polygons which only have vertical and horizontal edges, no diagonals or curves. For these types of polygons, rectangular subsections are more efficient.

You should look at your geometry and, if necessary, divide it up into Manhattan and non-Manhattan polygons using the Edit \Rightarrow Divide Polygon. Then set the Manhattan polygons to staircase fill and the non-Manhattan polygons to Conformal fill. For example, the spiral conductor shown below contains Manhattan sections in the feed lines and non-Manhattan sections in the circular spiral. It should be divided up such that the feedlines are represented by polygons set to staircase fill, and the circular spiral is another polygon set to Conformal fill.



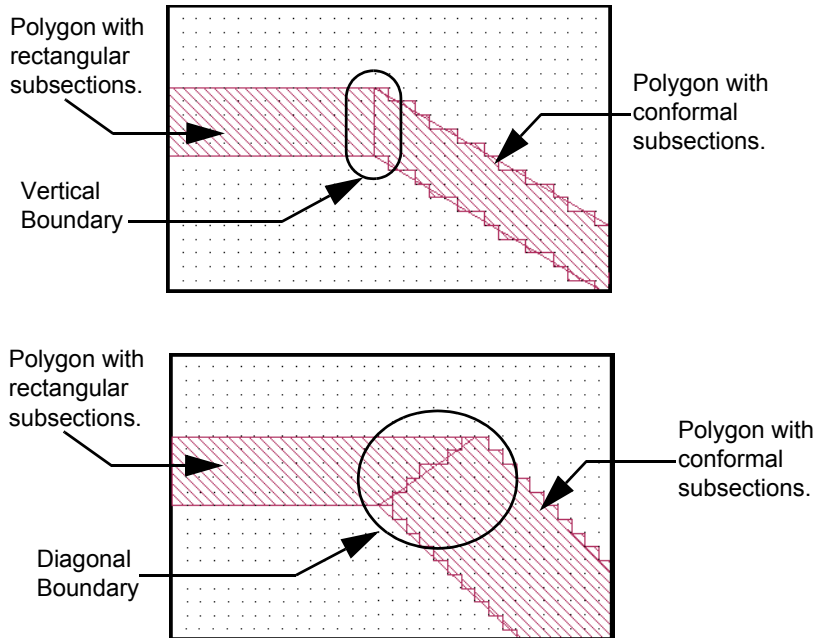
The exception to this rule is when relatively small Manhattan polygons are between conformal mesh polygons. In that case, the inefficiency of switching so frequently between staircase and conformal mesh outweighs the gain of using Manhattan polygons. In that case, conformal mesh should be applied to all the polygons. An example is shown below.



Normally, these polygons would use staircase fill, but because they are relatively small areas and in between polygons on which you would use conformal mesh, it is more efficient to apply conformal meshing to these Manhattan polygons.

Boundaries Should Be Vertical or Horizontal

For the most efficient results, the boundaries between polygons using conformal meshing and rectangular subsectioning should be vertical or horizontal as shown in the first picture below. Diagonal boundaries, as shown in the second picture, make the analysis less efficient.



Cell Size and Processing Time

Care should be taken when choosing your cell size when using conformal mesh. Many users, especially experienced Sonnet users, will estimate processing time based on the amount of memory required to analyze a circuit. The amount of memory used for conformal mesh can be deceptive. Using a smaller cell size in a circuit which uses conformal mesh may not increase the required memory but will have a noticeable effect on processing time. The significant factor in determining processing time with conformal meshing is the number of metalized cells needed

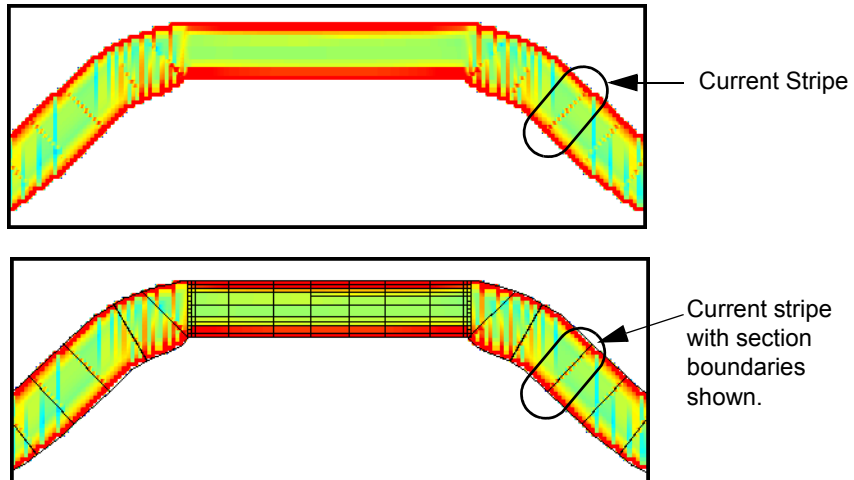
to construct a conformal section. The number of conformal mesh cells displayed as the result of the Estimate Memory command may be more reliably used as a guideline.

Current Density Viewing

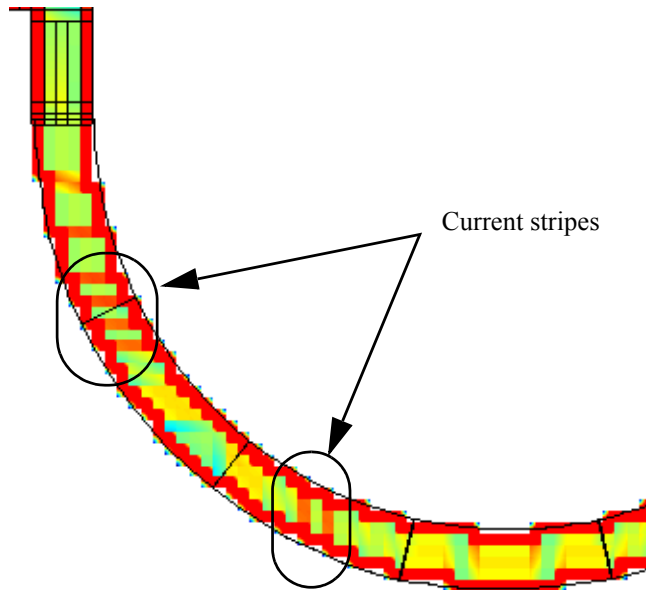
You may view the current of circuits using conformal mesh just like any other circuit. However, the current density of conformal mesh polygons might show unusual “striping”. These stripes do not represent real current, but are a by product of the conformal meshing algorithm.

There are two types of current striping:

- 1 A single stripe of current can appear on the junction between two conformal sections as shown below.



- 2 Horizontal or vertical stripes may appear within a curved conformal section producing a “ripple” effect as shown below.



For a tutorial on using conformal meshing please see Chapter 4, “Conformal Mesh Tutorial” in the **Sonnet Supplemental Tutorials**.

Chapter 12

Netlist Project Analysis

Netlist projects provide you with a powerful circuit analysis tool. Examples of ways in which the netlist may be used include:

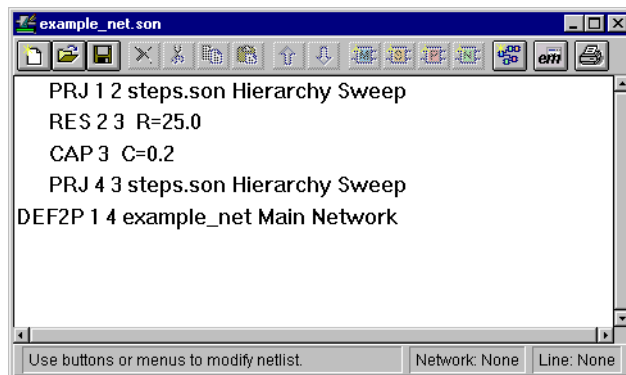
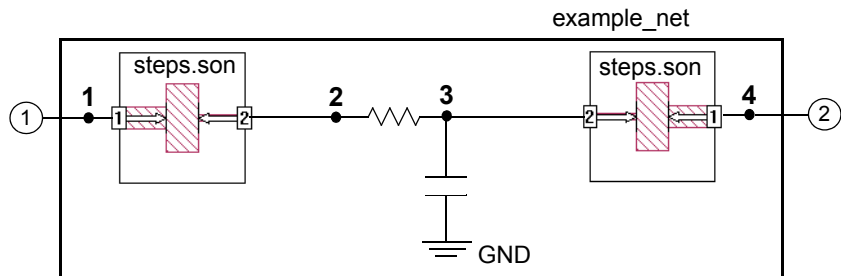
- ***Cascading Sonnet projects:*** You can analyze and combine multiple projects using previously existing data for the subprojects if it is available. This is particularly useful when analyzing large, complex circuits which require circuit subdivision for an *em* analysis. When analyzing a netlist project, *em* will automatically interpolate between frequencies if there are differences between the frequency sweeps used in the subprojects. It is also possible to impose the same frequency sweep on all the subprojects in a netlist. For more information about circuit subdivision, see Chapter 13, "Circuit Subdivision".
- ***Cascading S-, Y- and Z- parameter data files.*** You can read and combine multiple sets of S-, Y- and Z-parameter data files. This is particularly useful if you wish to combine results from another vendor's software for use in an

analysis by *em*. When analyzing a netlist project, *em* will automatically interpolate between frequencies if there are differences in the frequencies between the data files.

- ***Inserting modeled elements into a circuit.*** Modeled elements, such as resistors, capacitors, inductors and transmission lines, can be combined with geometry subprojects and S-, Y- and Z-parameter data files.

Networks

A netlist project contains a netlist which consists of one or more networks with elements connected together. The netlist provides a map in which the ports of individual elements in the netlist are connected to the ports of other elements by the use of nodes. Nodes represent a connection between netlist elements.



The picture above shows the network represented by the netlist shown in the project editor below it. The nodes are represented by the numbered black dots. The geometry project, steps.son, is connected between nodes 1 and 2 with node 1 corresponding to Port 1 in the geometry project and node 2 corresponding to Port 2. A resistor is connected between nodes 2 and 3. A capacitor is connected between node 3 and ground. The project steps.son is also connected between nodes 3 and 4, with Port 1 corresponding to node 4 and Port 2 corresponding to node 3. Port 1 of the network example_net corresponds to node 1 and port 2 of the network corresponds to node 4.

A netlist project is simply a list of these elements, as you can see in the netlist pictured above. Notice that the first number after the name of the element is the network node which corresponds to port 1 of the element, the second number is the network node which corresponds to port 2 of the element, and so on, for all the ports in an element.

Netlist Project Analyses

The sequence of steps for a netlist project analysis may be summarized as follows:

- 1 You input the netlist using the project editor in netlist mode. The project editor allows you to create and edit networks, network elements, project elements, modeled elements, and data file elements in your netlist. You also input the analysis controls which may include defining parameters in the netlist.
- 2 **Em** reads the netlist project which contains circuit and analysis control information. This includes S-, Y- and Z-parameter data files, modeled elements, geometry subprojects (project elements) and network elements.
- 3 **Em** uses the analysis controls input as part of the netlist project to run each electromagnetic analysis invoked by the network file. It is possible to configure the analysis controls in such a way that geometry subprojects are analyzed using their own analysis controls. Netlist subprojects always inherit their analysis controls from the present netlist.
- 4 Once the analysis of geometry subprojects is complete, **em** performs the circuit analysis specified in the netlist.

- 5 **Em** combines the electromagnetic results with the circuit results to obtain the desired output results.

Note that the above sequence of steps is generalized for analyses which include both electromagnetic and circuit analysis. In cases where the overall analysis is restricted to either electromagnetic analysis or circuit analysis, some of the steps are omitted.

Creating a Netlist

You create a netlist using the project editor. To create a new netlist, select *File* \Rightarrow *New Netlist* from the main menu of the project editor. The project editor tool bar and menus change for the netlist editor, to allow you to add elements and networks to your netlist.

The initial netlist file contains a default two-port network named Net. The last network in the netlist is the main network. The main network is the network whose solution you are solving for in this netlist. When you analyze the netlist, the response data produced in the analysis is for the main network.

You can edit the name and attributes of this network including the number of ports by double-clicking on the entry. This is true of all entries made in the netlist; you must double-click on them to open the dialog box which allows you to edit the entry or select the item, then select the *Tools* \Rightarrow *Modify* command from the main menu.

Adding Networks



To add a network to your netlist, select *Tools* \Rightarrow *Add Network* from the project editor main menu or click on the Add Network button on the tool bar. The Edit Network dialog box appears on your display. Using this dialog box, you can define the number of ports in the network, port terminations, and assign netlist node numbers to the network ports. When you add a network, it is always added as the main network for the netlist. The pre-existing network is moved up in the file.

There are four types of elements available to include in a network: network, modeled, data file and project.

Adding Network Elements



The first type of element is the network element. A network element allows you to use another previously defined network as part of the present network. This is useful if you want to duplicate a network multiple times.

To add a network element to your netlist, click on the network to which you wish to add it, then select *Tools* \Rightarrow *Add Network Element* from the main menu or click on the Add Network Element button on the project editor tool bar. The dialog box is similar to the Edit Network dialog box except you may not change the name. You must select which network you wish to use by selecting the name of an existing network from a drop list. Only networks defined above your present network are available.

Adding Modeled Elements



A modeled element is a resistor, capacitor, inductor, ideal transmission line or physical transmission line. To add a modeled element to your netlist, select *Tools* \Rightarrow *Add Modeled Element* from the main menu or click on the Add Modeled Element button on the project editor tool bar. The Edit Element dialog box appears on your display. You select the type of modeled element you wish to add from a drop list then enter the parameters for the element in text entry boxes provided in the bottom of the dialog box. If parameters are defined in your netlist, then a parameter may be entered as the value for the modelled element's parameters. A text entry box appears for each port and ground for the element. The desired node number of the network for each port should be entered here.

Adding Data File Elements



A data file element is a data file which contains response data which you wish to include in your network. The response data can be an optional output file from a Sonnet analysis or data in Touchstone or Super Compact formats such as measured data. To add a data file element to your netlist, select *Tools* \Rightarrow *Add Data File Element* from the main menu or click on the Add Data File Element button

on the project editor tool bar. A browse window appears which allows you to select the data file you wish to include in your network. Once the file is read, the Edit Element dialog box appears which allows you to define the netlist node numbers for the ports.

Adding Project Elements



A project element allows you to use another Sonnet project as part of your network. The project may be a geometry or netlist project. To add a project element to your netlist, select *Tools* \Rightarrow *Add Project Element* from the main menu or click on the Add Project Element button on the project editor tool bar. The Edit Element dialog box appears on your display. A browse window appears which allows you to select the project you wish to include in your network. Once the project is read, the Edit Element dialog box appears which allows you to define the netlist node numbers for the ports. If the project contains parameters, you may also enter values for these here. If parameters are defined in your netlist, then a parameter may be entered as the value for the modelled element's parameters. This feature can be used to force two parameters from two different projects to use the same value.

Adding Parameters

If you wish to add parameters to your netlist in order to perform parameter sweeps or optimizations, you do so by selecting *Circuit* \Rightarrow *Parameters* from the project editor main menu. The Parameters dialog box appears, which allows you to enter a parameter name and nominal value for each parameter you wish to create. Once a parameter exists in your netlist project, the parameter may be entered as the value of a parameter of a netlist element.



TIP

If you add parameters to a subproject in a netlist, the parameters are not automatically displayed in the netlist. You must save the main netlist and re-open it to display the parameters and make them available for editing.

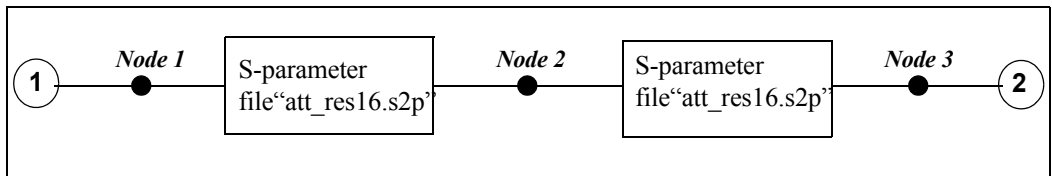
Netlist Example Files

All of the example files used in this chapter are available in the [Att](#) example in the Sonnet examples. You should copy the entire folder into your working directory if you wish to execute the examples. For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button.

Cascading S-, Y- and Z-Parameter Data Files

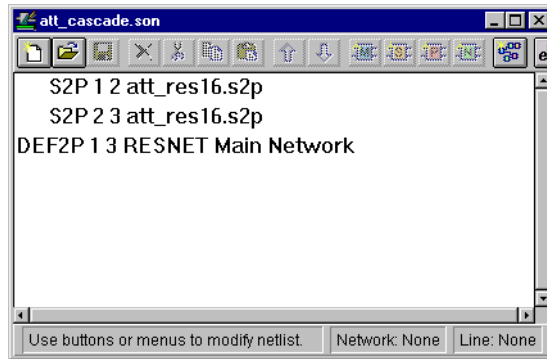
A particularly useful feature provided by a netlist project is the ability to cascade multiple S-, Y- and Z-parameter data files. There are no restrictions on the file formats which may be cascaded. For example, you can cascade *em* Z-parameter data in Touchstone format with measured S-parameter data in Super-Compact format. In addition, *em* can analyze at frequencies which are not included in the data files. *Em* automatically interpolates if there are any differences between the requested frequency points and those in the data files.

A good example of a cascading operation is the project att_cascade.son, which is included in the Att example for this chapter. A schematic representation of the two-port circuit is shown below. This circuit consists of two identical thin film resistors connected in series. The S-parameters from the geometry project analysis on the thin-film resistor are used as a data file element in the netlist. The desired output network is the series combination of resistors. The S-Parameter data file, att_res16.s2p, as well as the geometry project, att_res16.son, used to generate the data file, are included in the examples for this chapter.



The two-port S-parameters contained in file "att_res16.s2p" are cascaded to obtain an overall set of two-port S-parameters.

The netlist, att_cascade.son, for the circuit is pictured below.



The main network, Resnet, has two ports indicated by the “2” in DEF2P. Port 1 corresponds to node 1 in the network. Port 2 corresponds to node 3. There are two data file elements in the network. The first entry is the response file att_res16.s2p with 2 ports. Port 1 corresponds to node 1 of the network, which, as mentioned above, is port 1 of the whole circuit. Port 2 for the data file corresponds to node 2 of the network. The other entry is also for the data file att_res16.s2p except that port 1 of the data file goes to node 2 of the network which means that port 1 of the second data file is connected to port 2 of the first data file. Port 2 of the second data file corresponds to node 3 of the network. Node 3 of the network is Port 2 of the network “RESNET”.

The S-Parameters for an analysis of the netlist are shown below.

```
Frequency: 200 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
200.000000 0.250782 -5.309 0.748778 -6.263 0.748778 -6.263 0.250782 -5.309

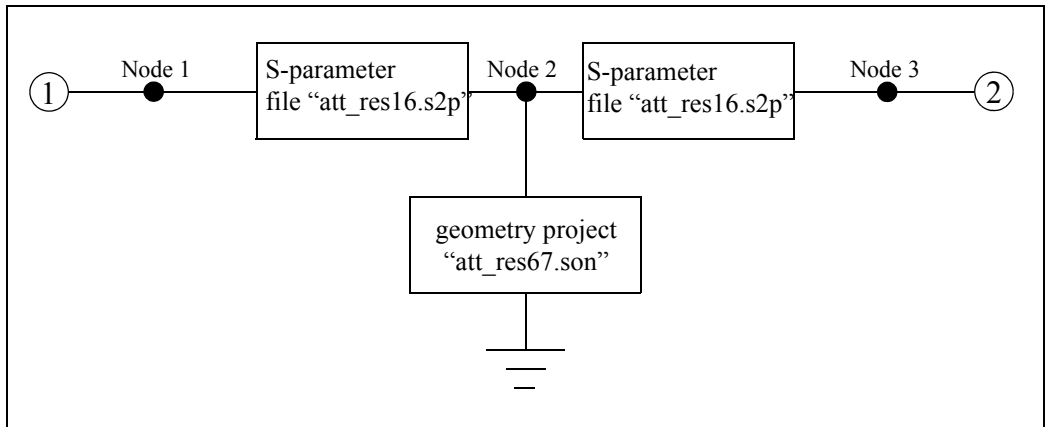
Frequency: 300 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
300.000000 0.250310 -7.963 0.748702 -9.395 0.748702 -9.395 0.250310 -7.963

Frequency: 400 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
400.000000 0.249650 -10.62 0.748595 -12.53 0.748595 -12.53 0.249650 -10.62
```

A Network File with Geometry Project

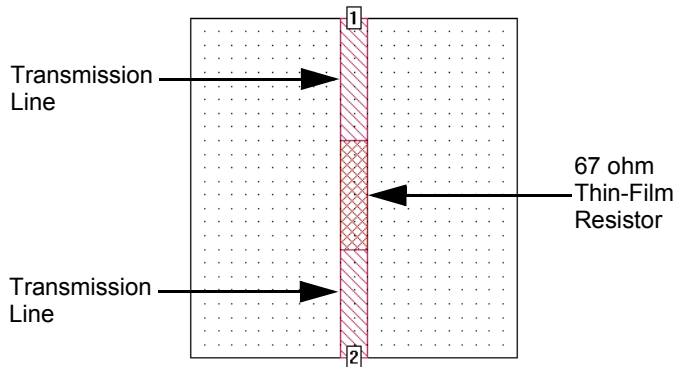
The next example demonstrates a netlist project analysis which invokes a geometry project analysis in conjunction with using previously generated data.

To demonstrate a netlist with a geometry project, the two-port T-attenuator shown below will be analyzed.

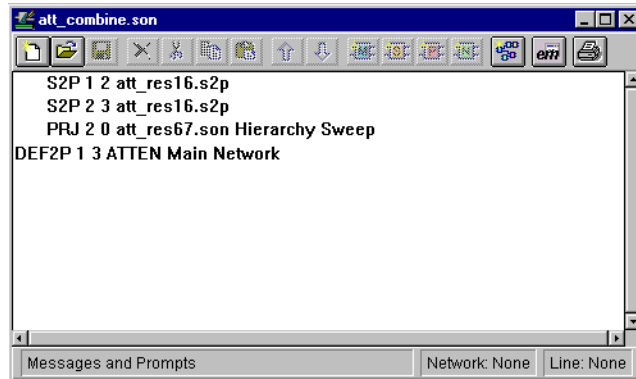


The two-port T-attenuator will be analyzed with **em** to demonstrate a combined electromagnetic/circuit analysis.

Pictured below is the geometry project “att_res67.son”, which is a 67 ohm thin-film resistor. This project is read by *em* and analyzed during the netlist analysis. The results of the project analysis are used to compute the results for the netlist.



The netlist project, att_combine.son is shown below. The project att_combine.son is available as part of the Att example for this chapter.



The primary distinction between the netlist shown above and the previous netlist is that this netlist contains an instruction to perform a project analysis. The PRJ keyword instructs *em* to run an electromagnetic analysis on the project “att_res67.son” using the analysis controls from the netlist. The analysis control use is indicated by “Hierarchy Sweep” in the PRJ statement. When control is set to “Hierarchy Sweep”, *em* automatically analyzes the subproject at the same frequency sweep and run options as the netlist.

During the analysis, *em* performs the following steps:

- 1 Reads S-parameter data from the file “att_res16.s2p”.
- 2 Performs an electromagnetic analysis of the geometry project “att_res67.son”, a 67 ohm thin-film resistor.
- 3 Combines the S-parameter results from the electromagnetic analysis with the S-parameter results from “att_res16.s2p” to obtain an overall set of S-parameters for the T-attenuator.



TIP

Before executing a PRJ statement, *em* checks for the existence of data at the specified control frequencies. If the data already exists, and the project has not changed since the data was generated, *em* does not execute an electromagnetic analysis, but uses the available data.

The listing below shows the output of the netlist analysis, as it appears in the analysis monitor, which contains the overall set of S-parameters for the T-attenuator.

```
Frequency: 200 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
200.000000 0.008924 67.700 0.500516 -5.758 0.500516 -5.758 0.008924 67.700

Frequency: 300 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
300.000000 0.013072 68.918 0.501160 -8.647 0.501160 -8.647 0.013072 68.918

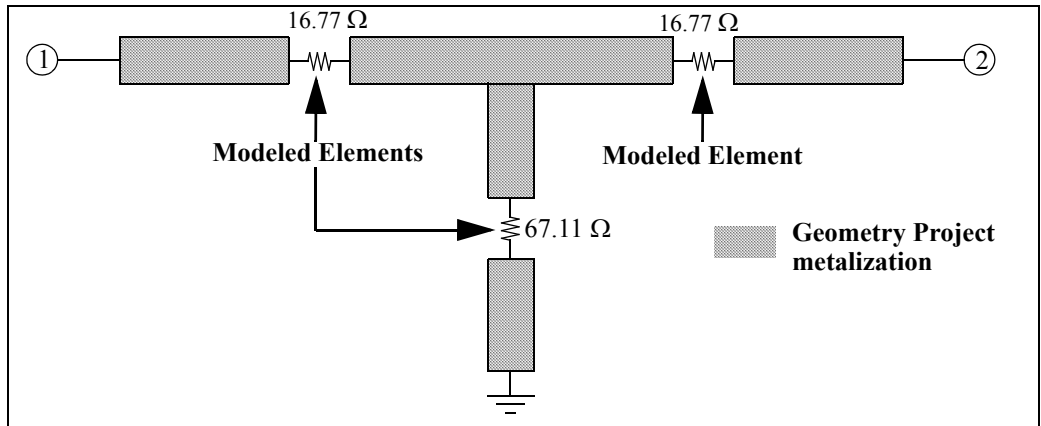
Frequency: 400 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
400.000000 0.017177 67.763 0.502055 -11.54 0.502055 -11.54 0.017177 67.763
```

Inserting Modeled Elements into a Geometry

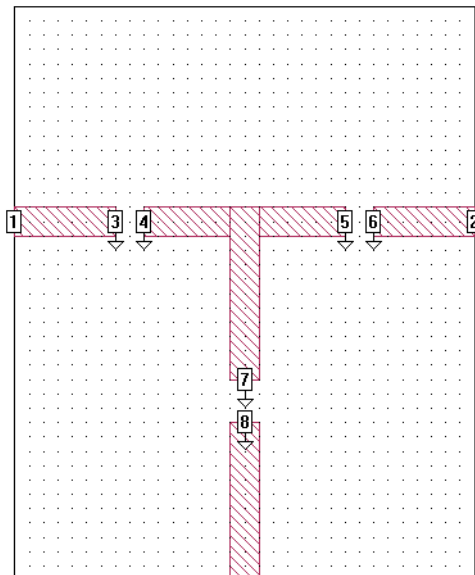
Another very useful feature of the netlist project is the ability to insert modeled elements into a geometry project after an electromagnetic analysis has been performed on that circuit. A modeled element is an ideal element such as a resistor, inductor, capacitor or transmission line, which has a closed-form solution. No electromagnetic analyses are performed on modeled elements.

To demonstrate the use of modeled elements, we will again analyze the T attenuator. However, instead of the attenuator being the result of connecting the results of electromagnetic analyses as shown previously in the chapter, in this case, the geometry project, `att_lgeo.son`, has the full attenuator with cutouts where the modeled elements need to be inserted. The three resistors will not be analyzed as part of the geometry project, but will be inserted as modeled elements in the netlist. The figure below shows the circuit layout with the modeled resistor elements. A geometry project for the transmission line structures are created first. A netlist project will then be used to insert the three resistors and calculate two-port S-parameters for the overall circuit.

To accomplish this task, it is necessary to create a geometry project with the transmission line structure and three “holes” where modeled elements will eventually be inserted. The figure on page 196 shows such a geometry project. Here, pairs of auto-grounded ports have been placed on the edges of each modeled element “hole”. When the modeled elements are inserted later on, each is connected across the corresponding pair of auto-grounded ports. Note that under certain conditions, ungrounded-internal ports can be used instead of auto-grounded ports. See “Using Ungrounded-Internal Ports,” page 197, for details.

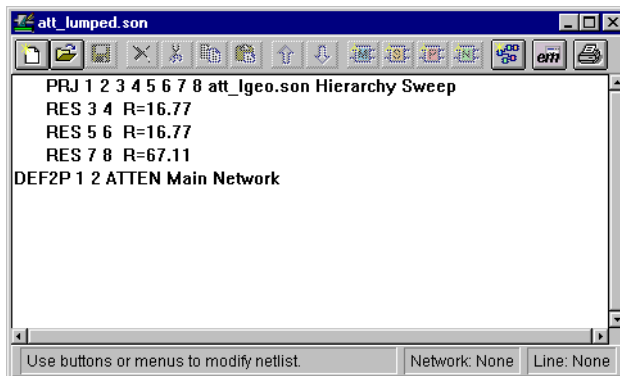


The two-port T attenuator will be re-analyzed to demonstrate the use of modeled elements.



The geometry file “att_lgeo.son” contains three sets of auto-grounded ports placed at locations where modeled elements will eventually be inserted. This file is available as part of the Att example used for this chapter.

Below is the netlist, att_lumped.son, that will be used for this example.



The netlist above instructs *em* to perform the following steps:

- 1 Perform an electromagnetic analysis on the geometry file “att_lgeo.son” using the Frequency sweep and run options defined for this netlist. Note that according to the PRJ line, Ports 1-8 correspond to nodes 1-8 respectively in the main network, atten. The node numbers are listed after the PRJ keyword in the order of ports in the circuit.
- 2 Insert a 16.77 ohm resistor between nodes 3 and 4 which is the equivalent of inserting the resistor between autogrounded ports 3 and 4 in the geometry project.
- 3 Insert a 16.77 ohm resistor between nodes 5 and 6.
- 4 Insert a 67.11 ohm resistor between nodes 7 and 8.
- 5 Calculate an overall set of S-parameters for the T attenuator.

The two projects, att_lgeo.son and att_lumped.son are available in the Att example for this chapter.

The listing below is the analysis output as it appears in the analysis monitor. Note that these results are similar to the results given above for distributed elements.

```
Frequency: 200 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
200.000000 0.007889 66.619 0.500390 -4.888 0.500390 -4.888 0.007889 66.619

Frequency: 300 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
300.000000 0.011495 69.396 0.500788 -7.336 0.500788 -7.336 0.011495 69.396

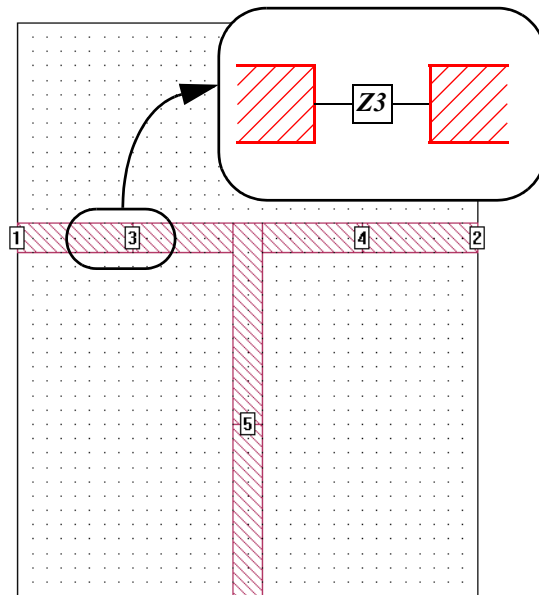
Frequency: 400 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
400.000000 0.015119 69.443 0.501342 -9.787 0.501342 -9.787 0.015119 69.443
```

Using Ungrounded-Internal Ports

In the example presented above, a pair of auto-grounded ports was placed at each location in the *em* circuit layout where a modeled element would eventually be inserted. It is also possible to perform the same analysis using ungrounded-

internal ports, because each resistor in this example is a series modeled element without access to ground. Any time access to ground is not required for a modeled element, you can replace the pair of auto-grounded ports with a single ungrounded-internal port.

The figure below shows a geometry project for the T attenuator with ungrounded-internal ports at each modeled element location. Note that the gaps between polygons at these locations have been removed. This is because you must attach ungrounded-internal ports between two abutted polygons. This slightly impacts the overall performance of the attenuator.



The geometry project "att_lgeo2.son" uses ungrounded-internal ports at locations where modeled elements will eventually be inserted.

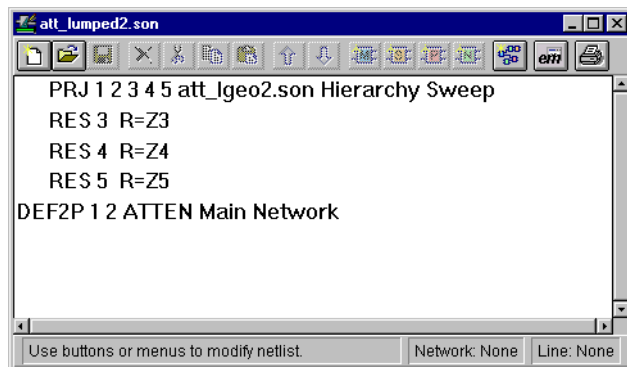
The network file shown below connects the desired resistors across the ungrounded-internal ports of the network shown on page 198. Since ungrounded-internal ports do not have access to ground, only a single node is specified when connecting an element across them.



WARNING

Ungrounded-internal ports have one terminal connected to an edge of a polygon and the second terminal connected to an abutted edge of a second polygon. Ungrounded-internal ports do not have access to ground. Therefore, only 1-port elements or 1-port networks may be connected across ungrounded-internal ports. Resistors, capacitors, and inductors are technically one-port elements and therefore, may be inserted in place of an ungrounded-internal port in a netlist.

The netlist for this circuit, att_lumped2.son, is shown below. Both the geometry project, att_lgeo2.son, and this netlist are available in the Att example provided for this chapter.



An important feature to notice in this netlist is the use of parameters. Three parameters, Z3, Z4 and Z5 have been defined in the netlist project and their values used for the three resistor modeled elements. Parameters are defined in a netlist by selecting *Circuit* \Rightarrow *Parameters* from the main menu, then entering the parameter name and nominal value in the Parameters dialog box which appears. Z3 and Z4 are equal to 16.77 and Z5 is equal to 67.11.

The listing below shows the S-parameter results obtained from the analysis with ungrounded-internal ports. These results are very similar, but not identical, to the results for auto-grounded ports. The differences are primarily due to the change in the gap size between polygons at the points where lumped elements are inserted.

```
Frequency: 200 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
200.000000 0.009217 68.496 0.500482 -5.785 0.500482 -5.785 0.009217 68.496

Frequency: 300 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
300.000000 0.013510 70.114 0.500994 -8.683 0.500994 -8.683 0.013510 70.114

Frequency: 400 MHz
50-Ohm S-Params. Mag/Ang. Touchstone Format. (S11 S21 S12 S22).
400.000000 0.017788 69.364 0.501707 -11.59 0.501707 -11.59 0.017788 69.364
```


Chapter 13 Circuit Subdivision

Introduction

Sonnet provides the capability to take a large circuit and split it into any number of smaller projects, then connect the results in a netlist project to produce a response for the whole circuit. This method can significantly reduce the required processing time and memory necessary to analyze the circuit while still obtaining an accurate answer.

The number of subsections in a circuit is one of the most important factors in determining processing time since the matrix solve time is proportional to N^3 . To illustrate how circuit subdivision reduces processing time, consider two subprojects each with half as many subsections as the source project. The total matrix solve time is now four times faster:

$$2(N/2)^3 = N^3/4$$

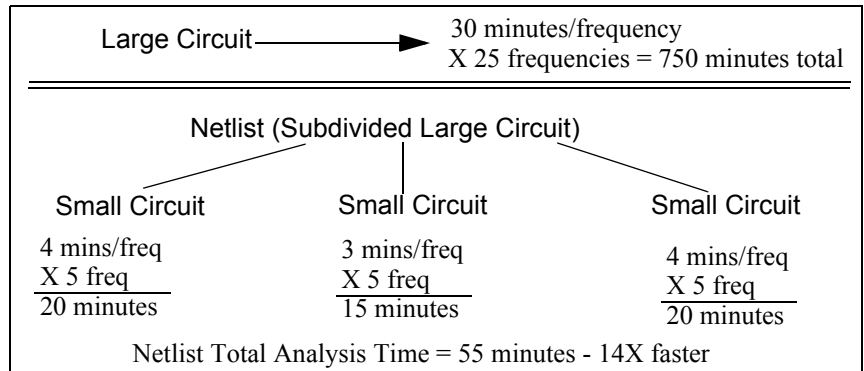
Circuit subdivision allows you to take advantage of this technique by breaking your circuit into smaller parts with fewer subsections, hence, requiring less processing time and memory to analyze. The trade off is that you introduce some error into the analysis. However, by subdividing the circuit appropriately you can minimize the error while still obtaining the reduction in processing time.

The circuit should be split where there is no coupling across the subdivision line. Areas where significant coupling occurs must be contained within a subproject. In this way, all the significant coupling in the circuit is accounted for. If care is taken when subdividing the circuit, the accuracy of the results is very high.

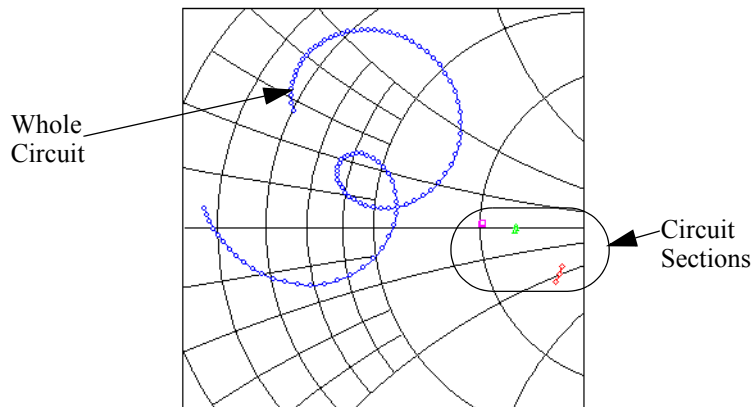
Circuit subdivision is not appropriate for every design, but in the cases of large circuits (5-10 minutes processing time per frequency) where it is applicable, you can obtain marked increases in processing efficiency.

Another advantage of circuit subdivision is the use of frequency interpolation in the master netlist analysis. A netlist is used to connect the response data of the subprojects of the circuit to simulate the full circuit. If the subprojects are chosen in such a way that their response data does not vary significantly over the frequency band, very few frequency points need to be calculated for the subproject. So not only do the smaller files require less time and memory because of their smaller size, but you can also analyze these smaller circuits at fewer frequency points. Interpolating in the netlist file requires much less processing than calculating data for a frequency point in a geometry project.

Shown below is an example showing the typical advantages of using this approach.



When the netlist analysis is performed, *em* will interpolate to provide simulation data at frequencies not specified in the subprojects. Each subproject should be analyzed at the same minimum and maximum frequency as the overall analysis and at enough points in between to provide for reasonable interpolation of data at frequencies which fall between these values. As you can see from the Smith chart below, while you need many frequency points to obtain reasonable response data for the whole circuit, you need far fewer frequency points to obtain accurate data for the smaller pieces of the whole circuit, whose response data does not vary appreciably.

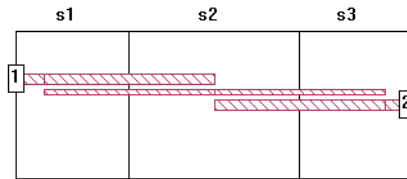


Be aware, however, that in some cases, you may need the added precision of analyzing all the pieces at the same resolution of the frequency band. Interpolation is best used when the response of a subproject varies little over the frequency band and the analysis time of the subproject is appreciable.

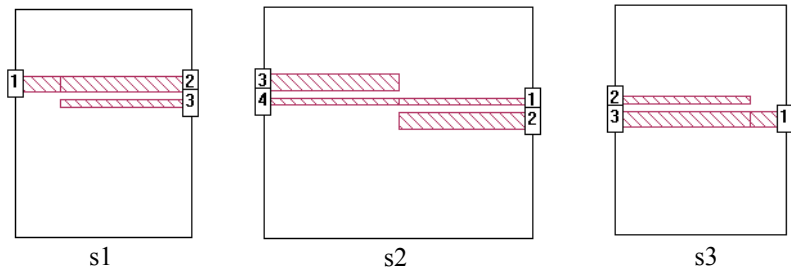
Circuit Subdivision in Sonnet

Circuit subdivision in Sonnet allows you to insert subdivision lines in your geometry in the project editor. These subdivision lines create the sections from which the subdivide command makes geometry subprojects. When you select the

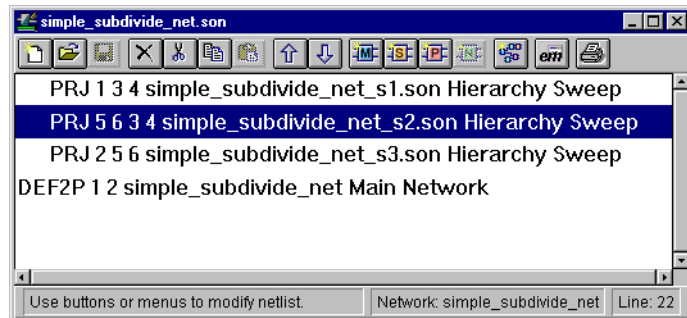
subdivide command, the software creates a main netlist file and the geometry subprojects. The main netlist connects the subprojects so that the response data for the netlist may be substituted for the response data of the source project.



Source Circuit with Subdivision Lines Added



Generated Subprojects



Generated Main Netlist

You should also be aware that if your main circuit contains any parameters or dimensions, they are removed during the subdivision process. After the subprojects are created, you may enter parameters in any of the geometries. In fact, it is possible to run optimizations on the main netlist project using a parameter in one of the subprojects.

Performing circuit subdivision as a method of analysis should, in general, be done as follows:

- 1 You should input as many of the circuit properties as possible before subdividing. Dielectric layers, dielectric brick and metal types, grid size, top cover height, etc. are inherited by the created geometry subprojects.
- 2 Decide where to subdivide your circuit. This step often requires expertise and experience to avoid splitting the circuit at a junction where there is coupling across the subdivision line.
- 3 Create the subdivision lines in the project editor. These lines are used to create the subprojects. A geometry project is created for each segment of your circuit. These geometry projects contain significantly smaller geometries that may be analyzed faster using less memory.
- 4 If you plan to take advantage of the netlist interpolation feature, set up the analysis frequency controls as a coarse resolution of the entire desired frequency band in the project before subdividing. Both the netlist and geometry subprojects all inherit these frequency specifications. Entering these frequency controls now in the Analysis Setup dialog box saves having to enter them in each individual subproject.
- 5 Subdivide the circuit in the project editor to create the subprojects and netlist project which connects the individual subprojects in a network equivalent to the circuit as a whole. Ports and reference planes are added to the subprojects as needed to connect to the larger circuit.
- 6 Edit the subprojects to fine tune the geometries, if needed. Possible adjustments would include the use of a binary box, adjusting the grid size, setting z-partitions for bricks, changing the frequency sweep specification, and adding parameters.



TIP

If you add parameters to a subproject in a netlist, the parameters are not automatically displayed in the netlist. You must save the main netlist and re-open it to display the parameters and make them available for editing.

- 7 Set up the analysis controls in the netlist to use the complete set of desired analysis frequencies if the subprojects are already set to analyze the coarse frequency sweep. When the analysis is performed on the master netlist project, *em* interpolates between the frequency points in the subprojects, saving processing time.

You may also use a Hierarchy Sweep in which the frequency band set up in the master netlist is imposed on the analyses of all the subprojects. This is useful when additional accuracy is needed in the data and you do not wish to use interpolation. This is accomplished by setting the Hierarchy Sweep option in the Analysis Setup dialog box in the project editor.

- 8 Analyze the netlist project. The data response for the netlist project provides analysis results that may be used for the whole circuit.
- 9 It is often a good idea after analyses are complete on the resultant subprojects to check the response data to verify that data was calculated for enough frequency points to provide accurate interpolated data.

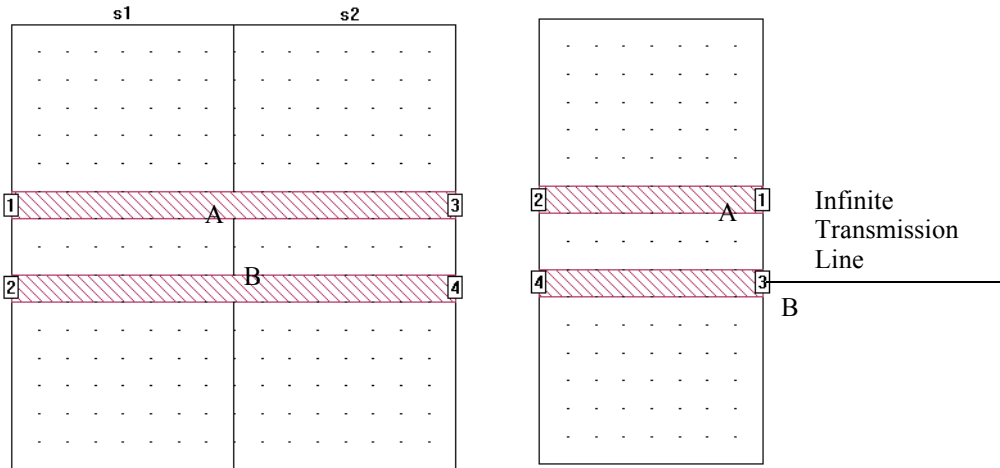
Since it is possible for a netlist project to include a netlist subproject, it is possible to use “double” subdivision. After subdividing your initial circuit, you then may use subdivision on one of the resulting geometry subprojects. In this case, you would need to change the name on the appropriate PRJ line from the old geometry subproject to the new netlist subproject.

Choosing Subdivision Line Placement

As mentioned above, the difficult part in using circuit subdivision is to decide where to place your subdivision lines to split the circuit. The subdivision lines should be placed between polygons which have negligible coupling. Places on the circuit where a high degree of coupling or rapidly varying currents are present should be kept within an individual subproject.

The de-embedding of the port discontinuity in Sonnet is done by essentially modeling infinitely long transmission lines at the port. This allows transmission lines to be subdivided with very little loss of accuracy. This includes microstrip lines, stripline, and coupled lines including coplanar. This point is illustrated below.

The circuit shown below, on the left, consists of a coupled transmission line. This is too simple a circuit to require subdivision but is very useful in demonstrating the principle. When subdivided, the circuit is split into two subprojects both of which would resemble the circuit shown on the right. Since the port discontinuity is modeled as an infinite transmission line when the port is de-embedded the coupling between points A and B is accounted for.

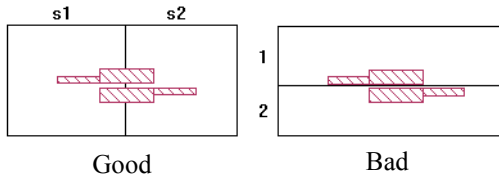


It is important to avoid areas where there is coupling across the subdivision line. Subdivision lines should not split any diagonal polygon edges. Illustrated below are good placements and bad placements of subdivision lines.

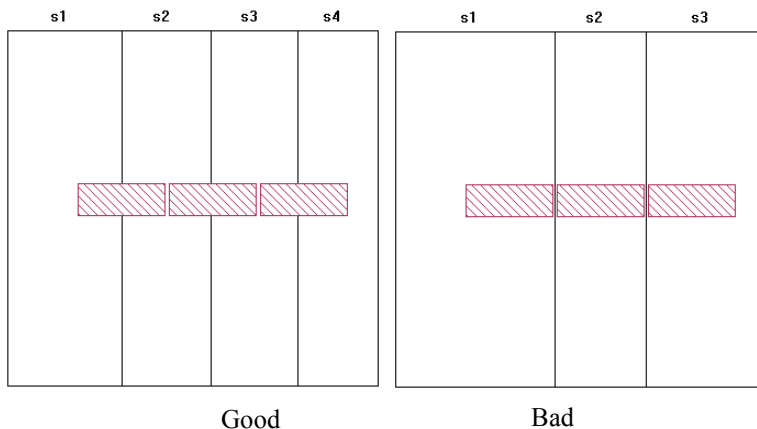
Good and Bad Placements of Subdivision Lines

This section contains a series of illustrations each showing the good placement of a subdivision line in a circuit and its counterpart showing a bad (and in some cases illegal) placement of a subdivision line. Setting a subdivision line perpendicular to one or more transmission lines provides a good general guideline for line placement.

The first example is a pair of coupled lines. As explained above, when you split coupled lines as on the left, very little loss of accuracy results. However, on the right, you have split the coupled pair along the axis where significant interaction takes place. The subprojects have no way to account for this coupling and will produce bad data.

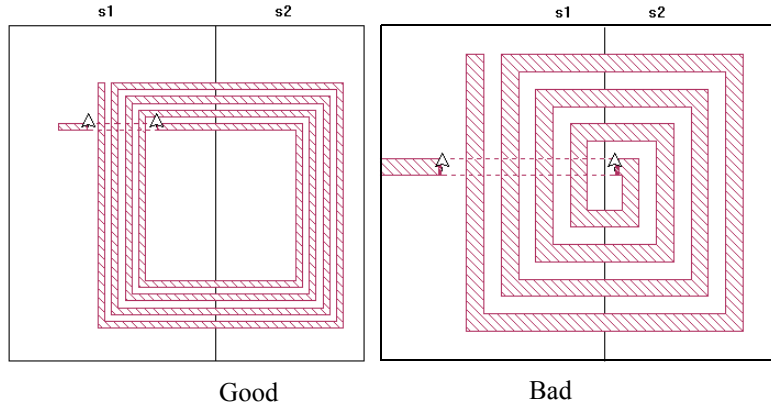


The second example shows how to split a series of resonators. In this type of structure, there is strong coupling at the gaps between adjacent resonators. The example on the left is good since the subdivision lines do not prevent this inter-resonator coupling. The example on the right is incorrect since the resulting subprojects do not contain the inter-resonator coupling.

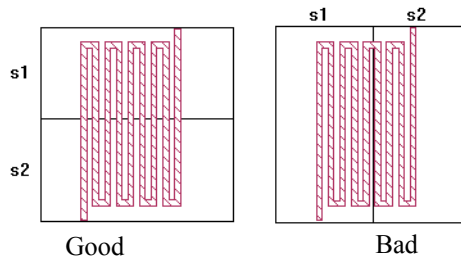


The third example shows a square spiral. The example on the left is a good placement since the location where the spiral is divided is essentially a group of coupled transmission lines and the subdivision line is perpendicular to those lines.

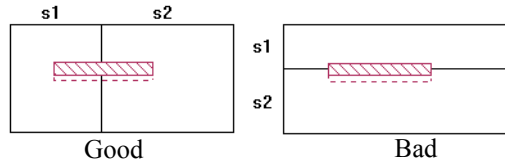
Here, the left side of the spiral is sufficiently far from the right side so that coupling is negligible. The example on the right is bad because the lines on the left side of the spiral do couple strongly with the lines on the right side.



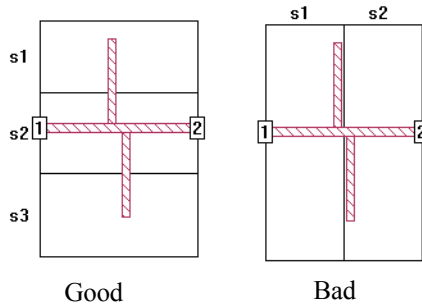
The meander line on the left is split in such a way that Sonnet provides an accurate answer since the bends on the top are far enough away from the bends on the bottom that coupling between them is negligible. The example on the right provides an inaccurate result because the coupling between two close transmission lines is eliminated by the subdivision.



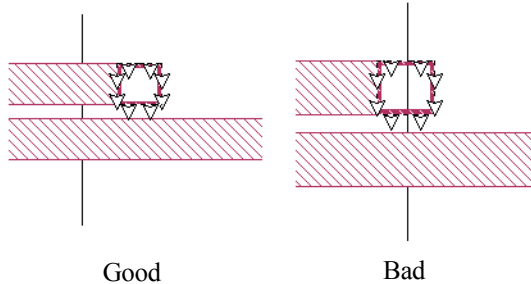
The circuit shown below has coupled transmission lines on two different layers. Once again, it is correct to place a subdivision line perpendicular to the transmissions lines, but not parallel to them. Subdivision is valid for multi-layer structures as long as the coupling across the subdivider is negligible.



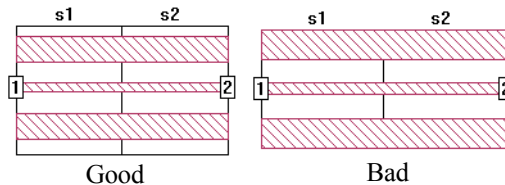
In the double stub circuit shown on the left, the subdivision lines split the polygon perpendicular to the direction of current flow and far from any discontinuities. The circuit on the right however, shows the subdivision line splitting the bases of the two stubs which may be coupled.



The subdivision line shown in the circuit on the right is wrong since the circuit is split in the middle of a via between layers. In general, subdivision lines should never be placed on top of discontinuities, such as vias. The subdivision line on the left is the correct placement.

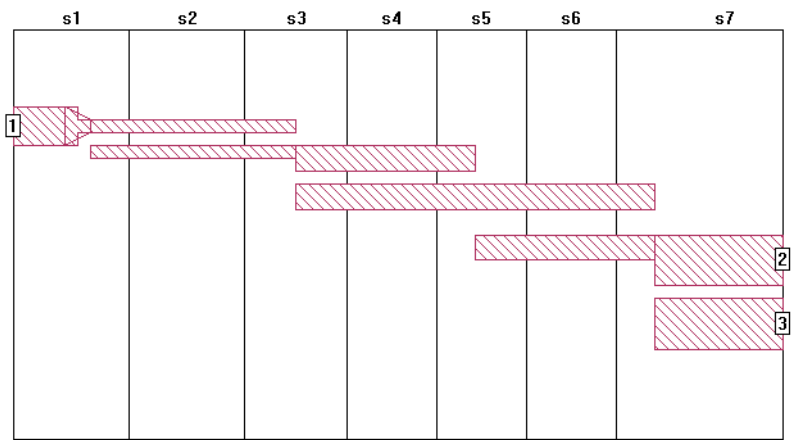


The subdivision line shown in the circuit on the right splits a polygon at the box wall which is an illegal placement for a subdivision line. It is illegal to subdivide polygons grounded to the box walls since such polygons do not behave like transmission lines. Also, the new ports added during the subdivide would be shorted to the boxwall. The circuit on the left is correct since there is no contact between the top and bottom polygons with the top and bottom box wall.

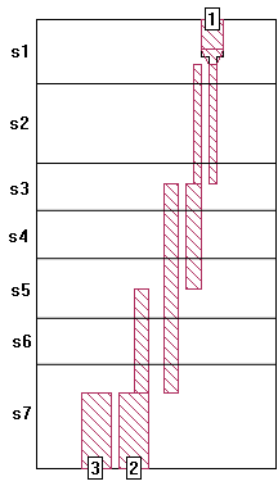


Subdivision Line Orientation

Subdividers may split the circuit on a horizontal axis or a vertical axis, but you may not mix orientation. Choosing the direction in which you split your circuit is dependent upon the structure of your circuit. Shown below is a typical circuit in which you would use the vertical orientation and another example in which you would use the horizontal orientation.



Example of Vertical Subdividers



Example of Horizontal Subdividers

You may use both orientations by using double subdivision mentioned earlier. The first time you subdivide your main circuit you choose an orientation for your subdivision lines. Then use circuit subdivision on the resulting geometry subprojects, this time using the opposite orientation for your subdivision lines.

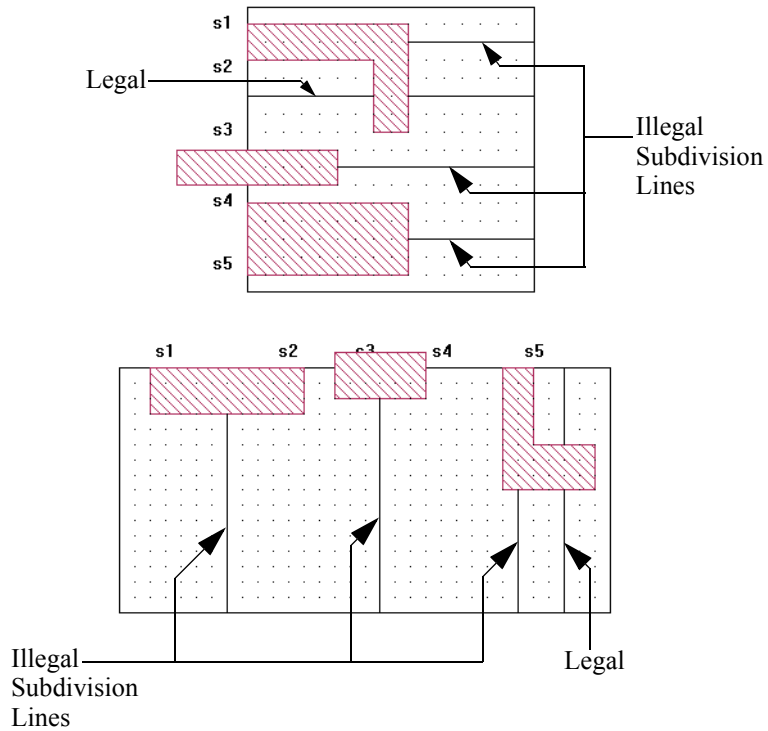
Before adding subdividers to your geometry project, you should ensure that specification of your circuit is complete. Subprojects created when you execute the subdivision inherit their properties from the source project. Such properties as cell size, metal types, properties of the dielectric layers, dielectric bricks, metal levels, etc. are all used in the resultant subprojects.

When you place a subdivider in your circuit, a line representing the subdivider appears in the horizontal or vertical plane running through the point at which you clicked. The resultant sections of the circuit are automatically labeled. Subdivision sections are labeled from left to right, or top to bottom, depending upon orientation. These labels are always sequential and are non-editable.

Once a subdivider has been added to your circuit, you may edit the subdivider as you would any other object in your geometry. You may click on the subdivider and move it. You may also control the display and selection of the subdivider lines and labels in the Object Visibility dialog box and the Selection Filter dialog box.

The following are illegal conditions for subdivision lines:

- May not be off grid.
- Should not be placed where there is coupling across the subdivision line.
- May not be colinear with polygon edges.
- May not split a diagonal polygon edge.
- May not split a port.
- May not be below the line of symmetry.
- May not split a polygon at a box wall. See the picture below.



Once you have completed adding all the desired subdividers to your circuit, you must save the project before performing the subdivision.

Setting Up Circuit Properties

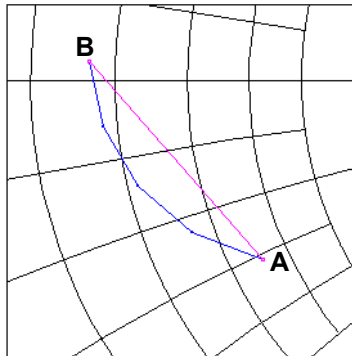
Since the geometry subprojects created by the subdivide inherit their properties from the source project, you should complete entering all the desired attributes for your circuit before performing the subdivide. This includes such things as defining the dielectric layers (which includes the height of the box top), top and bottom box metals, metal and dielectric brick materials, cell size and box size. This saves the effort of having to enter these values in each of the subprojects.

Setting Up the Coarse Step Size Frequency Sweep

If you plan to use interpolation to obtain response data when analyzing the master netlist project, you should input the coarse frequency sweep at which you wish to analyze the geometry subprojects before subdividing the circuit. These frequencies should cover the same frequency range as the analysis frequencies for the whole circuit but use a coarser step size. The subprojects should be analyzed at the same minimum and maximum frequency as the overall analysis and at enough points in between to provide for reasonable interpolation of the response.

By inputting the coarse frequency sweep prior to subdivision, the master netlist and geometry subprojects created by the subdivide command will all inherit the analysis setup. After subdividing, you will need to enter the desired finer frequency step size in the master netlist project before analyzing it. In addition, you will need to turn off Hierarchy Sweep.

The figure below shows a Smith chart with a circuit analyzed at five frequency points, next to the same circuit analyzed at only two points. As you can see, using only two data points would result in more interpolating error than using five data points. Whether or not two data points is acceptable depends upon the proximity of points A and B. If A and B are very close, then two data points are sufficient. If A and B are far away, then five or more data points should be used.



It is always a good idea to check the Smith chart for the response data of your subprojects to ensure that you have chosen enough frequency points at which to calculate data so that any interpolated data is reasonably accurate.

Subdividing Your Circuit

The actual subdividing of your circuit into separate geometry subprojects and a master netlist project is performed by the software. You enter the desired names for the master netlist and geometry subprojects. You may also automatically add feedlines of lossless metal to any ports generated in the subprojects.

Feedlines should be added when discontinuities contained in sections of your source circuit need to be moved away from the boxwall to prevent interaction between the boxwalls and the discontinuity. The use of feedlines are optional; if you choose to add a feedline, you may use the suggested length calculated by the software or input your own value. By default, the software creates feedlines using the suggested length.

When the subdivide is executed, Sonnet creates a geometry subproject for each section of the circuit in which you placed the subdividers. It also creates a master netlist that connects the geometry subprojects together to produce an equivalent circuit for the original geometry project that you subdivided.

Each of the geometry subprojects uses the properties of the original circuit: cell size, dielectric layers, dielectric and metal materials, analysis setup, etc. Therefore, all the geometry subprojects contain the same analysis setup with the same analysis frequencies specified.

Analyzing Your Subdivided Circuit

To obtain the desired response data, edit the analysis setup for the master netlist so that all of the desired analysis frequencies are specified. Each of the geometry subprojects are set up with the coarser resolution of analysis frequencies. When the netlist is analyzed, *em* runs the geometry project analyses first to produce response data for each part of the network. Then the analysis of the whole network is executed. Em interpolates to produce data for frequency points in between those available from the analysis of the geometry subprojects.

If properly subdivided, the results of the netlist analysis should provide an accurate solution for your difficult to handle circuit using fewer resources. The use of circuit subdivision is demonstrated in Chapter 3, "Circuit Subdivision Tutorial" in the **Sonnet Supplemental Tutorials**.

Chapter 14 Microwave Office Interface

Sonnet's Microwave Office Interface (MOI) provides a completely integrated "solver on request" interface between AWR's Microwave Office 6.0 or higher and Sonnet software. The interface allows you to stay completely in the Microwave Office environment using Sonnet as your EM analysis engine, or you may choose to edit your circuits in the Sonnet environment before running the EM analysis. Either way your results are easily integrated back into the Microwave Office environment. A third option allows you to export and import Sonnet projects to be used as EM Structures in your Microwave Office project.

NOTE:

The Microwave Office Interface only supports geometry projects. It is not possible to translate a netlist project from Sonnet.

This manual assumes that you are familiar with the basics of using both Sonnet and Microwave Office. If this is not true, we recommend referring to the appropriate documentation for whichever program you need to learn. If you are new to Sonnet, we suggest performing the tutorials in the Sonnet Tutorial manual, available as part of the hardcopy manuals set or in PDF format through the Sonnet task bar.

System Requirements

The interface is available for Windows 2000, Windows XP. The interface may operate on Linux Red Hat Enterprise 3.0 systems but has not been tested for this platform. For the AWR Microwave Office Interface, AWR's Microwave Office 6.0 or above is required.

Installation

The Microwave Office Interface is automatically installed when you perform your Sonnet installation. However, if the installation of the Microwave Office Interface does not complete successfully, you may install the Interface by doing the following:

- 1 Open the Sonnet task bar.**

If you do not know how to open the Sonnet task bar, please refer to “Invoking Sonnet,” page 15 in the **Sonnet Tutorial**.

- 2 Select Admin ⇒ MWOffice Interface ⇒ Install from the main menu on the task bar.**

The Microwave Office Interface is installed and this command is disabled.

NOTE:

If the Microwave Office Interface was successfully installed as part of the Sonnet installation, this command is disabled.

Uninstall

If you wish to uninstall the Microwave Office Interface without uninstalling all of Sonnet, do the following:

1 Open the Sonnet task bar.

If you do not know how to open the Sonnet task bar, please refer to “Invoking Sonnet,” page 15 in the **Sonnet Tutorial**.

2 Select Admin ⇒ MWOffice Interface ⇒ Uninstall from the main menu on the task bar.

The Microwave Office Interface is uninstalled. Once the uninstall is complete, the only item on the *Admin ⇒ Microwave Office* menu that is enabled is the Install command. Sonnet will no longer be available as a choice for EM Simulator in Microwave Office. All EM Structures in Microwave Office designs will be converted to EMSight, Microwave Office’s EM simulation engine.

Licensing

The Microwave Office Interface is available as part of Sonnet Lite and Sonnet LitePlus. It may be purchased as an option for Sonnet Level2 Basic or above. Please see your system administrator if you are unsure of the availability of this feature.

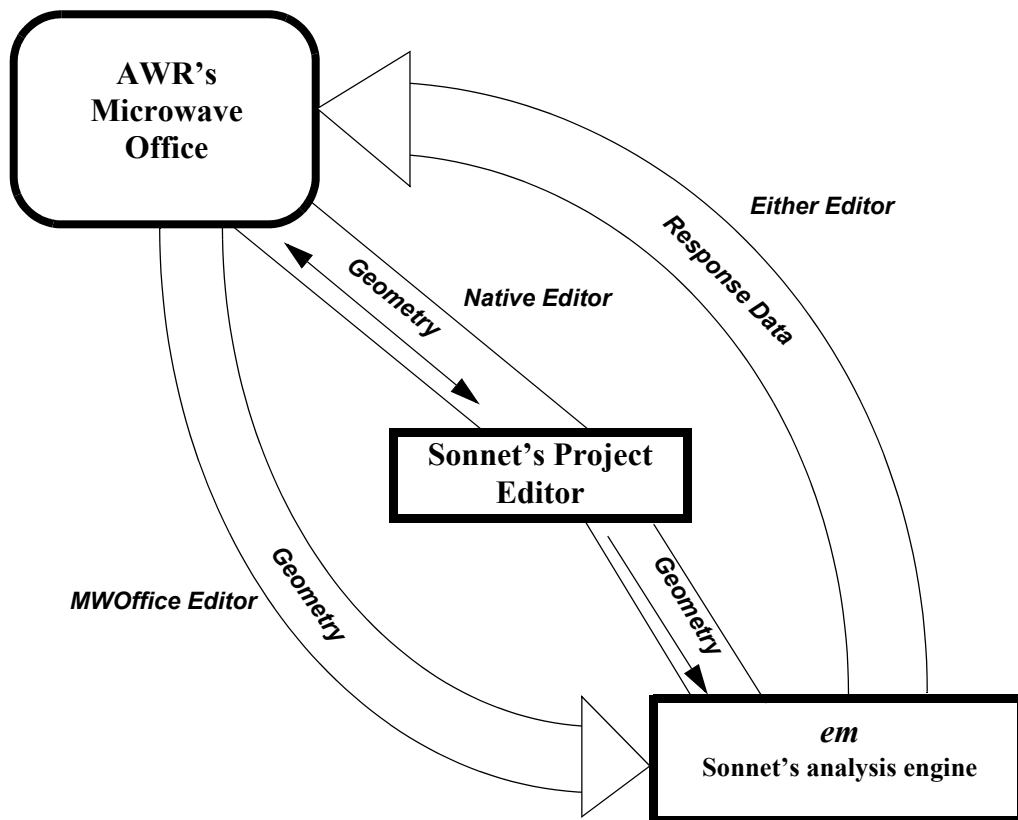
Microwave Office Interface Overview

You may use the Microwave Interface in a number of ways:

- You may edit your EM Structure, or circuit, in Microwave Office and run the Sonnet analysis engine, *em*, when you wish to simulate the circuit.
- You may invoke Sonnet’s project editor to edit your geometry, and run the analysis from the Microwave Office environment.
- You may export your EM Structure to a Sonnet project, work on the analysis in the Sonnet environment and import the Sonnet project to

Microwave Office when your analysis is complete.

Shown below is an overview of the Microwave Office (MWO) Interface and how the programs and projects that are used relate to one another



The Microwave Office Interface uses the EM Socket in AWR's Microwave Office to provide Sonnet's analysis engine, *em*, as the "solver on request". You may choose to edit your EM Structure in Microwave Office or in Sonnet's project editor (Native Editor in AWR).

When you edit your EM Structure in Microwave Office, the geometry is sent to Sonnet for analysis and only the analysis results are sent back to Microwave Office.

If you choose the Native Editor, Sonnet's project editor, to edit your geometry, the EM Structure is sent to Sonnet, changes are made in Sonnet and the geometry is sent back to Microwave Office and stored as part of the Microwave Office project. When an analysis is executed in Microwave Office, the Sonnet project is sent to Sonnet for analysis and analysis results are sent back when the analysis is complete.

Selecting Sonnet as your EM Simulator

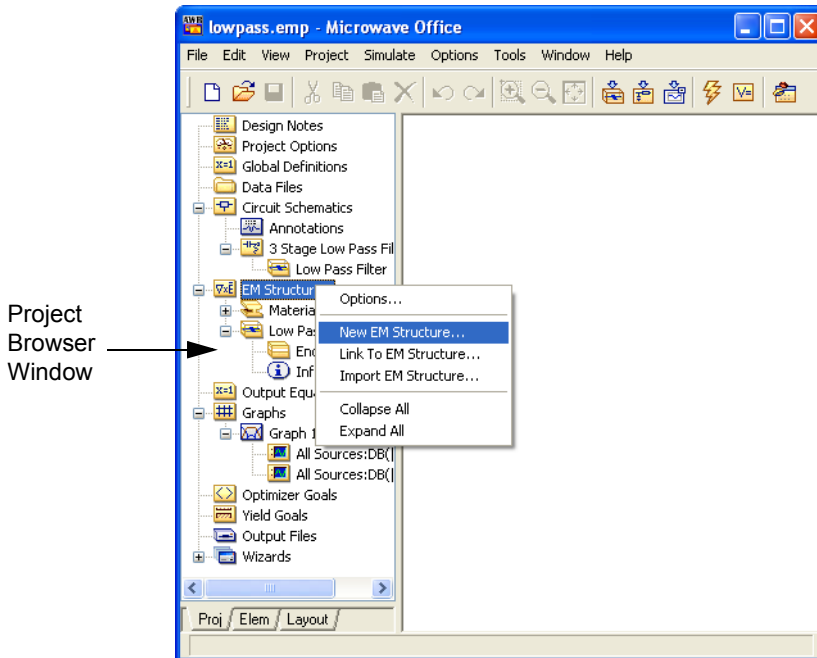
You may select Sonnet as your EM Simulator using one of two methods:

Opening a New EM Structure

When you open a New EM Structure in Microwave Office, you are prompted as to which analysis engine you wish to use. In order to select Sonnet do the following:

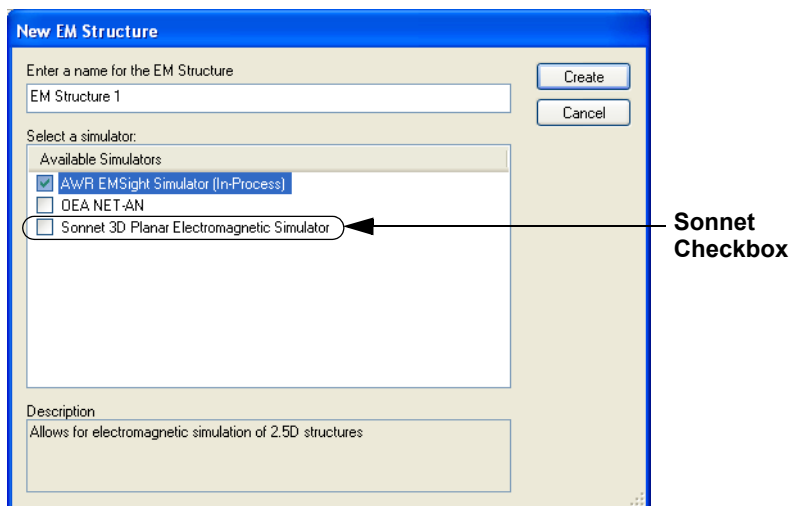
1 Open your Microwave Office Project in Microwave Office.

The Microwave Office window with the project browser window on the left hand side, appears on your display.



- 2 Right-click on EM Structures in the project browser and select “New EM Structure” from the pop-up window which appears.

The New EM Structure dialog box appears on your display.



- 3 Click on the “Sonnet 3D Planar Electromagnetic Simulator” checkbox.

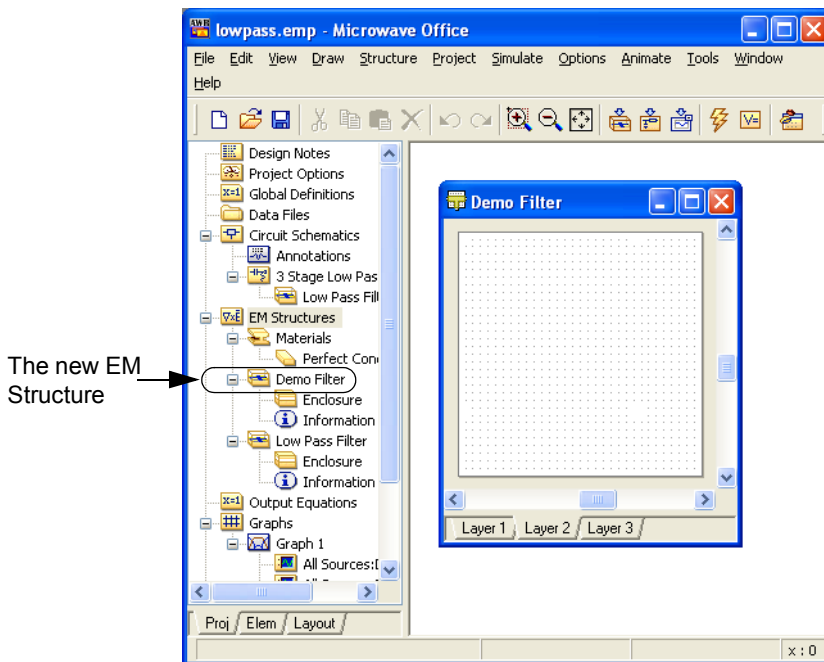
This select Sonnet’s *em* as your analysis engine for electromagnetic simulations within Microwave Office.

- 4 Enter the desired name for the new EM Structure in the Name text entry box at the top of the dialog box.

This will identify the EM Structure in the Microwave Office project.

- 5 Click on the Create button to create the new EM Structure and close the dialog box.

The new structure will appear in the project browser and a blank substrate appears in the Microwave office window as shown below. The new structure in this example is “Demo Filter.”



Selecting Sonnet for an Existing EM Structure

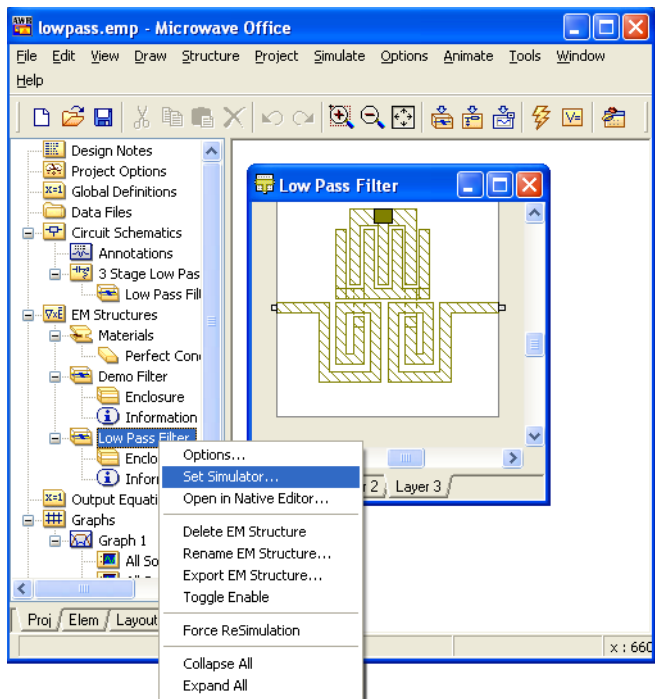
To select Sonnet as the EM Simulator for an existing EM Structure in your Microwave Office project, do the following:

- 1 Open your Microwave Office Project in Microwave Office.

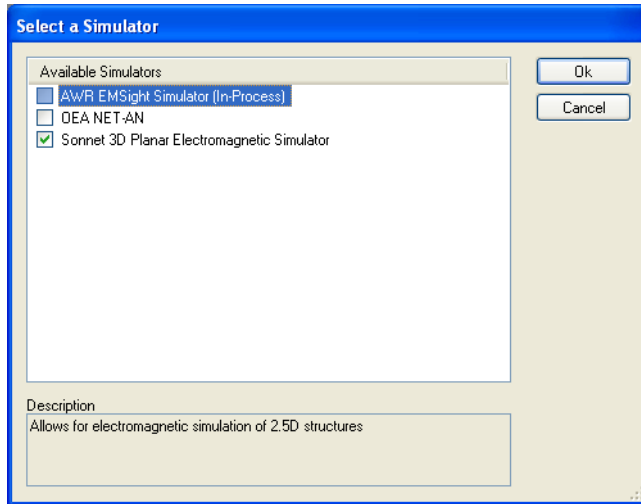
The Microwave Office window with the project browser window on the left hand side, appears on your display.

- 2 Right-click on the desired EM Structure in the project browser and select “Set Simulator” from the pop-up menu which appears.

The Select a Simulator dialog box appears on your display.



- 3 Select the “Sonnet 3D Planar Electromagnetic Simulator” checkbox.



- 4 Click on the OK button to apply the changes and close the dialog box.

This completes selecting Sonnet as the EM Simulator.

Editing in Microwave Office

When you select Sonnet as your analysis engine, you have a choice as to where you will edit the EM Structure: in Microwave Office or in Sonnet's project editor, known as the Native Editor. This next section discusses editing your structure in the Microwave Office environment.

You edit your EM Structure in Microwave Office when your structure does not use any advanced features unique to Sonnet such as thick metal, dielectric bricks or autogrounded ports. (For a complete list of features available only in Sonnet, see "Translation Issues" on page 255).

When you use Microwave Office as your editor, you may use the analysis controls provided by the Microwave Office interface in the Microwave Office menus and dialog boxes to control the analysis frequencies and run options. When you

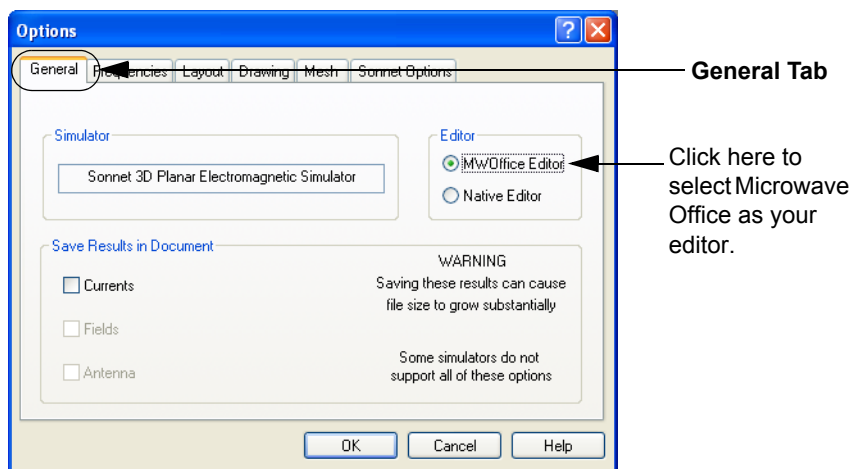
request an analysis, the geometry specification for the EM Structure entered in Microwave Office is passed to Sonnet and only the resulting analysis data is returned to Microwave Office.

Selecting Microwave Office as the EM Structure Editor

Microwave Office is the default editor for all new EM Structures. If you have previously selected Sonnet as your editor (see "Editing in Sonnet's Project Editor" on page 249), you may switch the editor back by doing the following:

- 1 **Right-click on the desired EM Structure in the Microwave Office project browser and select "Options" from the pop-up menu which appears.**

The Options property pages appear on your display and should appear similar to the illustration below.



- 2 **Click on the MWO Office Editor radio button to select Microwave Office as your EM Structure editor.**

All editing and changes to your EM Structure will be made in the Microwave Office editor. When you double-click on the EM Structure, the circuit is opened in the right hand pane of the Microwave Office window where you may edit it. This limits you to using features that are available in both Microwave Office and Sonnet. Please see "Translation Issues" on page 255 for a list of features which do not translate between Microwave Office and Sonnet.

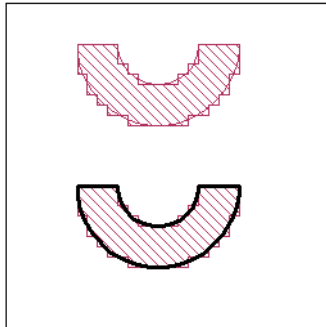


TIP

Double-clicking on an EM Structure in the project browser opens whichever editor is selected, Microwave Office or the Native Editor.

Changing the Fill Type

You may control the fill type used in Sonnet for metal polygons. The default fill type for Sonnet is staircase fill. In this case, small cells are used to approximate curved or diagonal edges as shown below.



The two polygons are identical. Notice that the dark outline on the bottom polygon indicates the polygon input by the user in Sonnet. The upper polygon shows the actual metalization analyzed by *em*. As you can see, a “staircase” is used to approximate a curved edge.

The conformal mesh fill type and the diagonal fill type are used to more accurately model curved and diagonal edges respectively. In Sonnet's editor, you may apply these fill types on a per polygon basis; however, when editing in Microwave Office, you may apply these fill types on a global, local or per polygon basis. For a detailed discussion on fill types and how they affect subsectioning in Sonnet, please refer to **Chapter 4, “Subsectioning”** on page 51.

Setting Sonnet Fill Type Globally

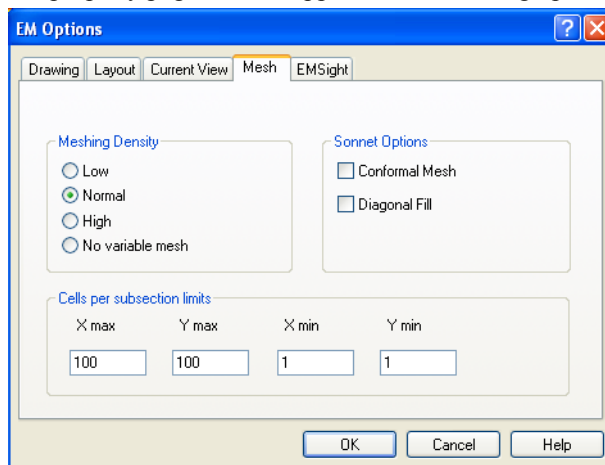
To apply the Conformal or Diagonal Fill option to all existing and new EM Structures in your Microwave Office project, do the following:

- 1 **Right-click on the “EM Structures” entry in the project browser and select “Options” from the pop-up menu which appears.**

The EM Options property pages appear on your display.

- 2 **Click on the Mesh tab in the property pages.**

The property pages should appear similar to the graphic below.



- 3 **Click on the desired checkbox, Conformal Mesh or Diagonal Fill, to use that cell fill type.**

This will become the default for all metalization in all EM Structures in your Microwave Office project. This setting may be changed locally for a particular EM Structure (see “Setting Sonnet Fill Type Locally,” page 234) or even for an individual polygon (see “Setting Sonnet Fill Type for a Single Polygon,” page 235).

- 4 **Click on the OK button to apply the changes and close the Property Pages.**

The fill type will default to your selection whenever a new EM Structure is created in this Microwave Office project.

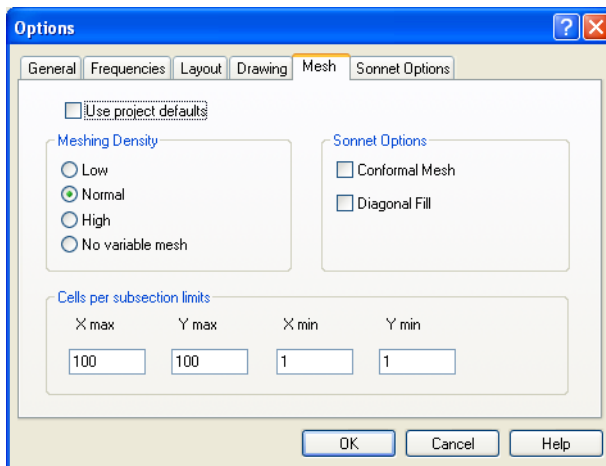
Setting Sonnet Fill Type Locally

- 1 **Right-click on the desired EM Structure name in the project browser and select “Options” from the pop-up menu which appears.**

The Options property pages appear on your display.

- 2 **Click on the Mesh tab in the property pages.**

The property pages should appear similar to the graphic below.



- 3 **If the Project Defaults checkbox is selected, click on it to clear it.**
When this checkbox is selected, then the Global settings are used. Clearing this checkbox allows you to make selections for this particular EM Structure.
- 4 **Click on the desired checkbox, Conformal Mesh or Diagonal Fill, to use that cell fill type for all the metalization in this EM Structure.**
If you wish to only use the mesh for a particular polygon, see “Setting Sonnet Fill Type for a Single Polygon” below.
- 5 **Click on the OK button to apply the changes and close the Property Pages.**

Setting Sonnet Fill Type for a Single Polygon

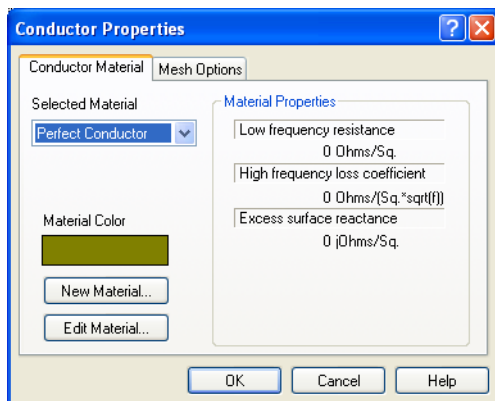
To change the mesh properties for a single polygon, do the following:

- 1 **Click on the desired polygon to select it.**

The vertices of the polygon will be highlighted to indicate selection.

- 2 **Right-click and select “Mesh/Material Properties” from the pop-up menu which appears.**

The Conductor Properties dialog box appears on your display.

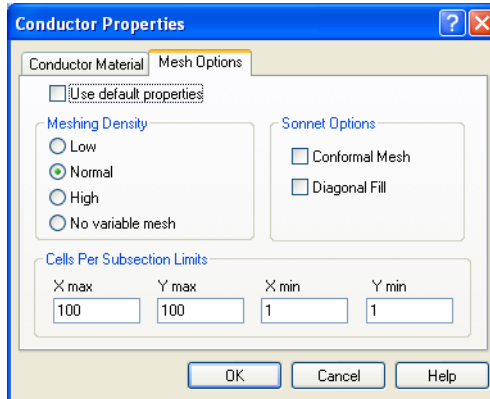


- 3 **Click on the Mesh Options tab in the Conductor Properties dialog box.**

The Mesh options are displayed.

- 4 If the Use default properties checkbox is selected, click on it to disable it.**

This enables all the settings and allows you to override the default settings for this EM Structure.



- 5 Click on the desired checkbox, Conformal Mesh or Diagonal Fill, to use that cell fill type for the selected polygon.**

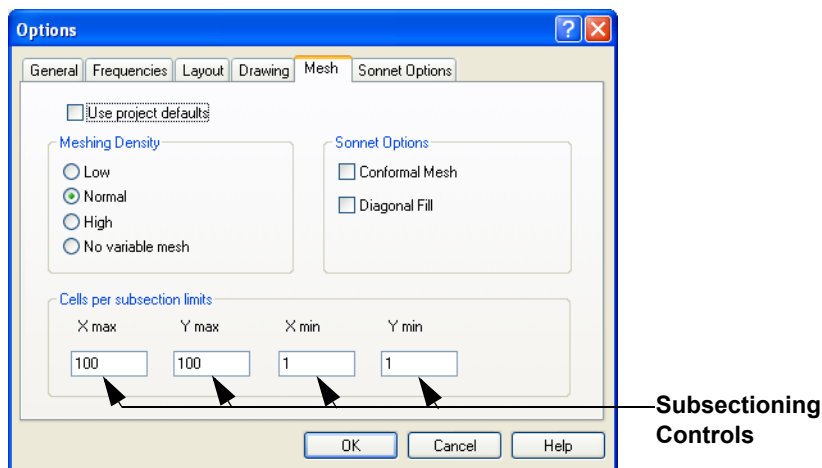
This applies the cell fill type to the selected polygon.

- 6 Click on the OK button to apply the changes and close the Conductor Property Pages.**

Controlling the Subsectioning (Meshing)

Sonnet allows you to control how cells are combined into subsections for each polygon. This is done using the parameters "X Min", "Y Min", "X Max" and "Y Max." These parameters may be changed for each polygon, allowing you to have

coarser resolution for some polygons and finer resolution for others. These parameters are available on the Mesh tab of the Options property pages in Microwave Office as shown below.



You may set these parameters globally, locally, or for a particular polygon as was shown in the “Changing the Fill Type” section above. For a detailed discussion of these parameters and how they affect subsectioning in Sonnet, please refer to “Changing the Subsectioning of a Polygon” on page 57.

Controlling the Analysis Frequencies

You may control the range of analysis frequencies as well as the type of frequency sweep used for your Sonnet analysis from the Options property pages in Microwave Office. You are limited to either analyzing at the points requested by Microwave Office or performing an Adaptive Band Sweep on a full or partial band.

Sonnet's Adaptive Band Synthesis (ABS) performs a fine resolution analysis of a specified frequency band. **Em** analyzes the circuit at the beginning and end frequencies. Using an iterative process, **em** then analyzes at other discrete frequencies and determines a rational polynomial fit to the S-parameter data within the frequency band. Once a rational polynomial fit is achieved with an acceptable error, the frequency response across the specified bandwidth is

calculated. If your Microwave Office simulation requires more than 4 or 5 frequency points, an Adaptive Sweep using ABS is the most efficient way to obtain the desired results.



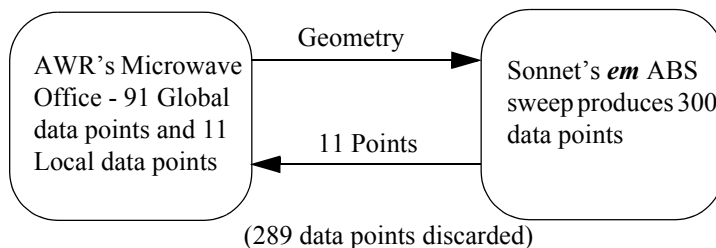
TIP

When using an ABS sweep, *em* produces the simulation data for 300 points in approximately the same amount of time as 10 data points.

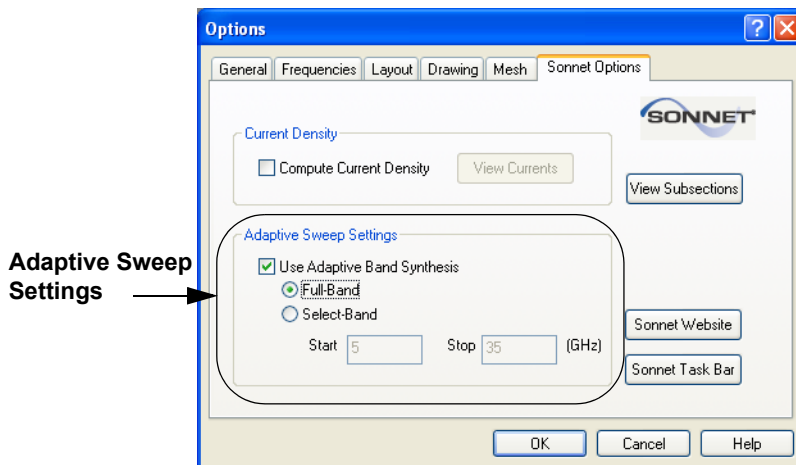
You will usually want to avoid using the Local frequency controls in Microwave Office, especially when the local controls request fewer data points than the Global Frequencies. Microwave Office uses the data obtained in the Local Frequencies to interpolate the Global frequency data. Since an ABS sweep in Sonnet can produce approximately 300 data points from just a few discrete data points, it is more efficient and accurate to only specify frequencies in the Global controls and use the data from Sonnet's analysis.

For example, assume the Global frequency control in Microwave Office is set to do an analysis from 1 to 10 GHz in steps of 0.1 GHz. This would require the analysis of the EM Structure at 91 frequency points. Also assume that the Local controls specify only 11 frequencies from 1 to 10 GHz and you specify an ABS sweep on the Full Band which in this case is 1 GHz to 10 GHz. When the *Simulate* \Rightarrow *Analyze* command is selected in Microwave Office, the Sonnet analysis engine, *em*, performs an ABS sweep from 1 GHz to 10 GHz producing approximately 300 data points across this band. The only data which is returned, however, are the 11 frequency points from 1 to 10 GHz in steps of 1 GHz as shown in the illustration below. Microwave Office would interpolate between these points to produce the

Global Frequency points. Therefore, the most efficient way to analyze your EM Structure is to specify all the desired frequencies in your Global controls in Microwave Office and specify an ABS sweep in Sonnet's *em*.



You use the Adaptive Sweep Settings on the Sonnet Options tab of the Options property pages to specify the frequency controls for your Sonnet analysis. To open this page, right-click on the EM Structure and select “Options” from the pop-up menu, then click on the Sonnet Options tab in the property pages which appear. The settings and their use are described below:



Use Adaptive Band Synthesis: Select this checkbox to run an ABS sweep in Sonnet when the EM Structure is analyzed. Note that although an ABS sweep produces approximately 300 data points, only those data points requested by

Microwave Office are returned from Sonnet. This checkbox is selected by default. If you do not want to run an ABS sweep, but want only to analyze at the required points set in Microwave Office, clear this checkbox.

Full-Band: Select this radio button, when you wish to perform the ABS sweep on the Full Band required by Microwave Office. This band is defined by the lowest and highest frequencies requested by Microwave Office. This radio button is only enabled when the Use Adaptive Band Synthesis checkbox is selected. This is the default setting for an ABS sweep. When this checkbox is selected, then the Analysis Control in the Analysis Setup dialog box (*Analysis* \Rightarrow *Setup*) in Sonnet's project editor is MWOFFICE Full-Band ABS.

Select-Band: Select this radio button when you wish to perform an ABS sweep over a different frequency band than the one required by Microwave Office. This may be part of the full band, overlap the full band or be an entirely different band. This radio button is only enabled when the Use Adaptive Band Synthesis checkbox is selected. When this checkbox is selected, then the Analysis Control in the Analysis Setup dialog box (*Analysis* \Rightarrow *Setup*) in Sonnet's project editor is Adaptive Sweep (ABS) and the frequency band is the one entered in this dialog box.

Note that if Microwave Office has requested frequencies out of this select band for which data does not already exist, **em** will have to run a full analysis at each requested data point which lies outside the select band. This may significantly increase processing time in the EM simulation.

Start: Enter the lowest frequency of the desired frequency band here. These dialog boxes are only enabled when a Select-Band Adaptive Band Synthesis is selected. If a Full-Band synthesis is selected, the lowest frequency requested from Microwave Office appears here, but may not be changed. The units being used appear to the right.

Stop: Enter the highest frequency of the desired frequency band here. These dialog boxes are only enabled when a Select-Band Adaptive Band Synthesis is selected. If a Full-Band synthesis is selected, the highest frequency requested from Microwave Office appears here, but may not be changed. The units being used appear to the right.

Computing Current Density

There is a run option available in a Sonnet analysis which allows you to compute current density data as part of the simulation. Selecting the Compute Current Density checkbox turns on this run option. Note that if you are running an Adaptive Band Sweep, current density data is only calculated for the discrete data points, not for all of the adaptive data.

NOTE:

NOTE: The current density data produced by Sonnet must be viewed using Sonnet's current density viewer. The current density data generated by the analysis engine, *em*, is not compatible with the Microwave Office framework. All selections in the Animate menu in Microwave Office are disabled when Sonnet is selected as the simulator.

Selecting the Run Option

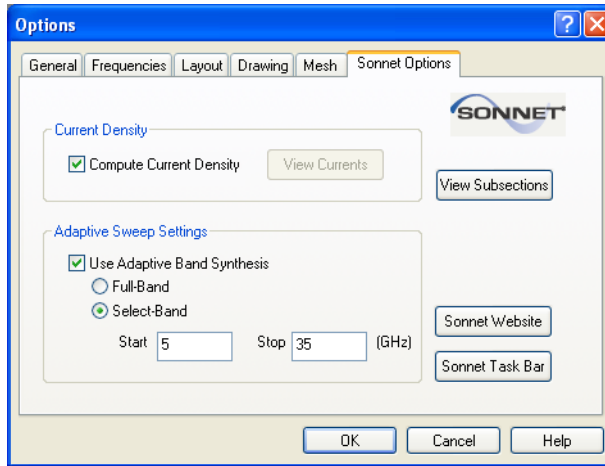
To calculate current density data as part of your Sonnet analysis, do the following:

- 1 **Right-click on the desired EM Structure (for which Sonnet has been selected as the simulator) and select Options from the pop-up menu which appears.**

The Options property pages appear on your display.

- 2 Click on the Sonnet Options tab in the property pages.

This displays the Sonnet options, as pictured below.



- 3 Click on the Compute Current Density checkbox.

When the Sonnet analysis is executed, current density data will be calculated for all discrete data points.

- 4 Click on OK to close the dialog box and apply the changes.

Viewing the Current Density Data

Once the analysis is complete and current density data is available, the View Currents button is enabled. Clicking on this button opens Sonnet's current density viewer, which allows you to view the current density data and animate the view as a function of time or frequency.

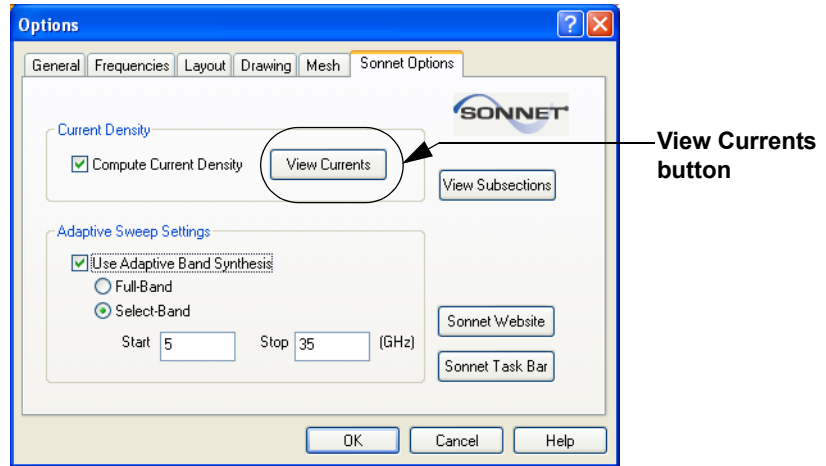
To view the current density data, do the following:

- 1 Right-click on the desired EM Structure (for which current density data has been created) and select Options from the pop-up menu which appears.

The Options property pages appear on your display.

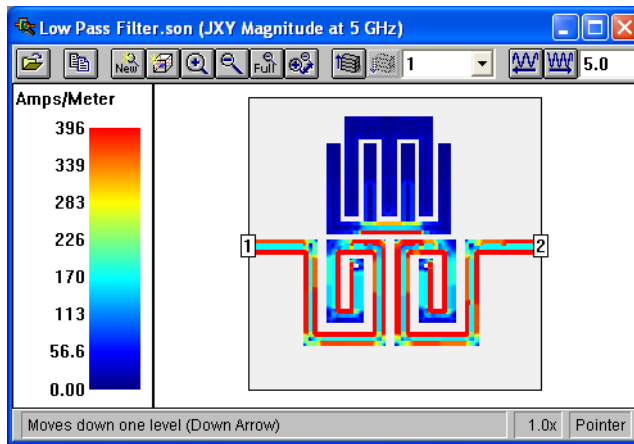
2 Click on the Sonnet Options tab in the property pages.

This displays the Sonnet options, as pictured below.



3 Click on the View Currents button.

This launches Sonnet's current density viewer which allows you to animate your current density data as a function of frequency or time. For more information on using the current density viewer, please refer to Sonnet's online help. Using the current density viewer is also demonstrated in Sonnet's first tutorial in the **Sonnet Tutorial**.



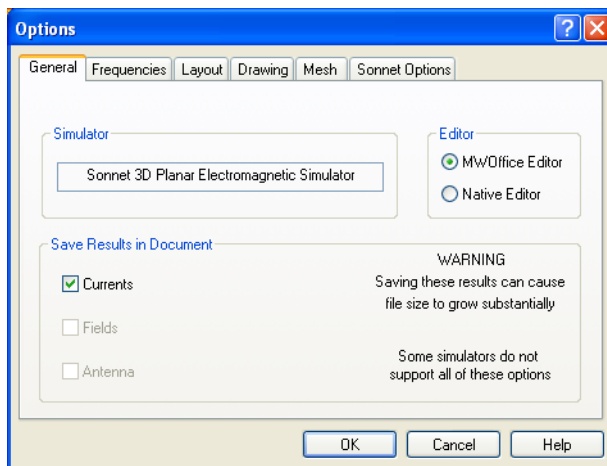
Saving the Current Density Data

You may choose to save your current density data from Sonnet as part of your Microwave Office project. To do so, perform the following:

- 1 **Right-click on the desired EM Structure and select "Options" from the pop-up menu which appears.**

The Options property pages appear on your display.

- 2 Click on the General tab of the property pages.
- 3 Click on the Currents checkbox to save the data.



When this checkbox is selected, any current density data returned from Sonnet after an analysis is stored as part of the Microwave Office project.

Viewing the Subsectioning

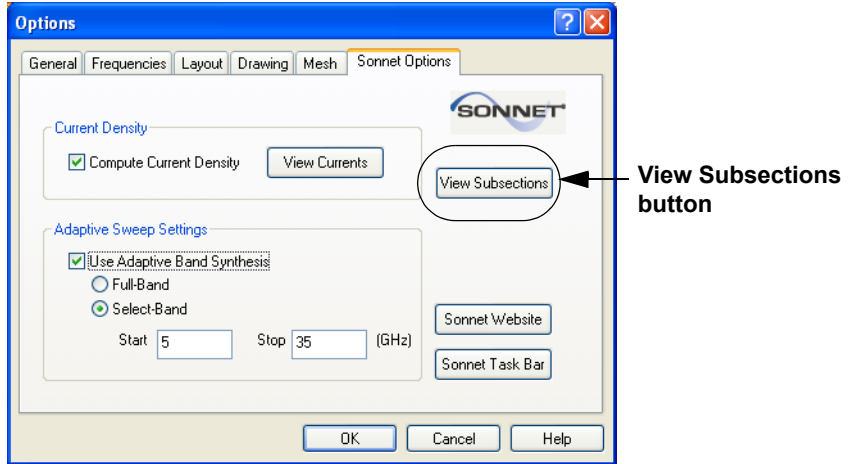
It is possible to view the subsectioning, or meshing, used by Sonnet when analyzing your circuit before you run a complete analysis. To do so, perform the following:

- 1 Right-click on the desired EM Structure and select Options from the pop-up menu which appears.

The Options property pages appear on your display.

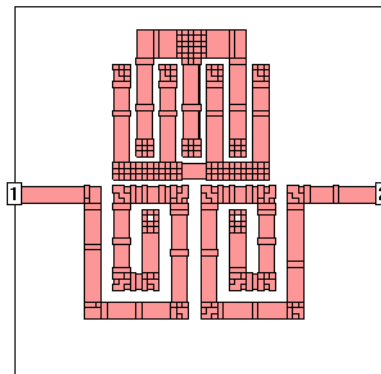
- 2 Click on the Sonnet Options tab in the property pages.

This displays the Sonnet options, as pictured below.



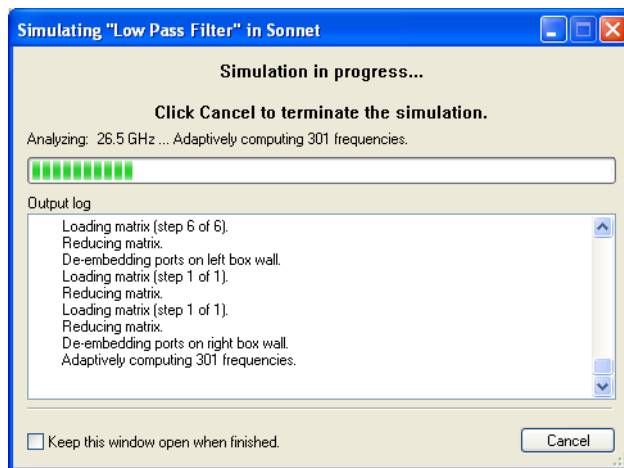
- 3 Click on the View Subsections button.

This launches Sonnet's subsection viewer which functions in a manner very similar to the current density viewer. The subsectioning for a circuit is pictured below. Note that the red indicates metalization and the black outlines are the subsections. (Color has been changed in diagram for clarity).



Executing the Analysis

When you select *Simulate* \Rightarrow *Analyze* from the main menu of the Microwave Office window, the Sonnet will be launched to analyze the EM Structure. A progress window detailing the Sonnet analysis appears on your display.



The window closes when the simulation is complete unless the “Keep this window open when finished” checkbox is selected. Be aware that Microwave Office is locked while the Sonnet analysis is running and you may not make any changes to your project at that time. If the Sonnet analysis is a significant one requiring a large amount of processing time, Microwave Office may be locked for some time. To avoid this, you may export your EM Structure to a Sonnet project. This allows you to make changes in the EM Structure and run analyses without locking up the Microwave Office project. However, this does increase the overhead on keeping your EM Structure in sync with the rest of the project and is more prone to human error. See "Working Outside Microwave Office" on page 252 for more details about the Import/Export options.

Response Data

When you use Sonnet as your analysis engine, the only electromagnetic simulation data which may be viewed in Microwave Office is the port parameter data (S, Y, Z, ABCD, etc.). Meshing, current density plots, and far field radiation pattern plots are not available in Microwave Office. When you choose Sonnet as your simulation engine, the Animation menu in Microwave Office is disabled. You must use Sonnet to view currents or far-field radiation patterns.

You may use the Graph function in Microwave Office to create plots of port parameter data including equations which use this data.

To view current density plots or far field radiation patterns, you must calculate current density data in your analysis. Since the format of current density data differs in Microwave Office and Sonnet, you must use Sonnet modules to view the current density data produced by Sonnet. See “Viewing the Current Density Data” on page 242.

Sonnet also uses current density data to calculate far field radiation patterns. If you have created current density data in a Sonnet analysis, you may use Sonnet's far field viewer to plot the patterns. To open Sonnet's far field viewer, open the EM Structure in the Native Editor, Sonnet's project editor, once the analysis is complete. Then, select *Project* \Rightarrow *View Far Field* from the project editor's main menu. Refer to online help for this program for instructions on how to view your data.

Editing in Sonnet's Project Editor

When you select Sonnet as your analysis engine, you have a choice as to where you will edit the EM Structure: in Microwave Office or in what Microwave Office refers to as the Native Editor which in this case is Sonnet's project editor. This next section discusses editing your structure in the Native Editor, Sonnet's project editor.



WARNING

If you choose to change the editor from Native Editor to Microwave Office Editor, any Sonnet specific changes in your geometry will be permanently deleted. If you once again choose Native Editor, you will have to re-enter those geometry changes.

You edit your EM Structure in Sonnet's project editor when you wish to take advantage of features unique to Sonnet such as thick metal, dielectric bricks or autogrounded ports. (For a complete list of features available only in Sonnet, see "Translation Issues" on page 255). Editing your EM Structure in Sonnet's project editor requires that you are already conversant with Sonnet software or you will have to use Sonnet documentation to gain familiarity with the Sonnet environment.

When you choose to use the Native Editor (Sonnet's project editor), the Sonnet project which contains the circuit specification is stored as part of the Microwave Office project file (.emp). When you open the EM Structure in the project editor or when you request an analysis, the geometry specification for the EM Structure is passed to Sonnet. When you have completed making changes to the circuit, saved the changes and exited Sonnet, the circuit specification is sent back to Microwave Office to be stored as part of the Microwave Office project. If an analysis has been run, the results are also stored along with the Sonnet project.

NOTE:

If you wish to store current density data from a Sonnet analysis as part of your project, you must select the “Currents” checkbox in the General tab in the Options dialog box in Microwave Office. See “Saving the Current Density Data,” page 244.

To choose the Native Editor option and edit the EM Structure in Sonnet's project editor, right-click on the EM Structure and select Open in Native Editor from the pop-up menu which appears on your display.

This launches a session of Sonnet's graphical editor. Once the EM Structure is opened in Sonnet, the Microwave Office project is locked until the Sonnet project editor is closed.

NOTE:

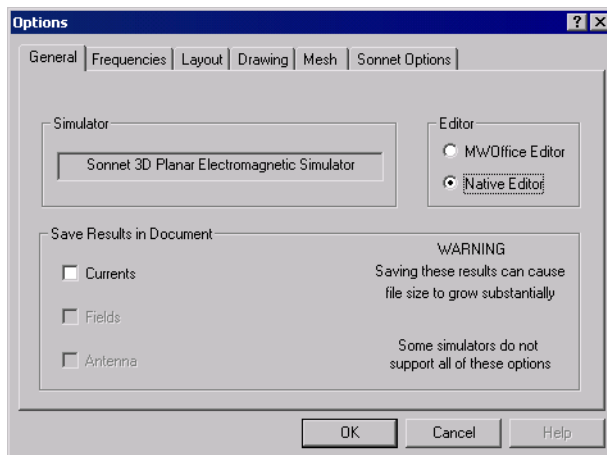
If a crash or other unexpected interruption happens to the project editor, the program may terminate without unlocking MWOffice. In those cases, it may be necessary to manually unlock MWOffice which will appear to be hung. For instructions on unlocking MWOffice, see “Unlocking Microwave Office,” page 263.



WARNING

When you close the Sonnet project editor after making changes to your EM Structure, if you are asked if you wish to make Sonnet your default editor, answer “Yes” to this question. Doing so ensures that any changes you make while in the Sonnet project editor are not lost.

You may also select the Sonnet editor for your EM Structure by selecting the Native Editor radio button in the General tab of the Options page for the EM Structure in Microwave Office. If you do this, you are not prompted when you end the session in Sonnet and your changes are saved automatically.



Editing in Sonnet has the advantage of allowing the user to use all capabilities of the Sonnet engine. The user may simulate the circuit, view all types of response data including current density data and far field patterns before returning to the Microwave Office environment with the analysis results.

It is important to remember when you are editing your EM Structure in Sonnet, the view in Microwave Office may not be accurate. When using Sonnet as your editor, your EM Structure may contain objects or features not available in Microwave Office; therefore, the circuit when viewed in Microwave Office, may not be accurately depicted. Opening the EM Structure in Sonnet's project editor provides an accurate depiction of your circuit.

If you open your EM Structure in Microwave Office when the Native Editor is selected, you may not make changes to the structure; the circuit is opened in Read Only mode.

You may analyze the circuit in Sonnet before returning to Microwave Office or the analysis can be run from Microwave Office. But be aware if you run the analysis from Sonnet then you must still analyze in Microwave Office. The analysis of the EM Structure will be very fast since the results are already available, but the analyze command in Microwave Office will trigger the delivery of the desired response data from Sonnet.



TIP

If your Microwave Office project has multiple EM Structures, it is easier to execute the simulations in Microwave Office to avoid executing multiple analyze commands in Sonnet and still having to analyze the complete circuit in Microwave Office.

You should also be aware of translation issues in going from the Microwave Office software to Sonnet software and vice versa. These translation issues are discussed in “Translation Issues,” page 255.

Working Outside Microwave Office

When you edit your circuit by using the Sonnet Native Editor from Microwave Office, then Microwave Office is locked while Sonnet's project editor is open. This does not present a problem for relatively small EM Structures which do not require long processing times; however, for a large circuit with significant processing time the lockout of Microwave Office may present a problem and exporting your EM Structure allows you to run a Sonnet analysis at the same time that you are using the Microwave Office framework. However, there is considerably more overhead in the translation and in keeping the EM Structure in Microwave Office synchronized with the Sonnet project.

Exporting Your EM Structure to Sonnet

The first way to work outside of the Microwave Office framework is to export your EM Structure to a Sonnet project by doing the following:

- 1 Open your EM Structure in the Native Editor.**

Your EM Structure is opened in Sonnet's project editor. Microwave Office is locked until the project editor is closed.

- 2 Select *File* ⇒ *MWOffice* ⇒ *Save As Sonnet Project* from the main menu of the Sonnet project editor.**

A browse window appears.

- 3 Select a name and location for the Sonnet project you are creating from your EM Structure.**

Your EM Structure is saved as a Sonnet project at the specified location.

- 4 Exit the project editor by selecting *File* ⇒ *Exit* from the main menu.**

This closed the project editor and unlocks Microwave Office.

You would then edit and analyze your circuit in Sonnet until the circuit satisfies your design criteria. Once this is completed, you would import the Sonnet project to an EM Structure in Microwave Office by using the Load Sonnet Project command in the project editor. For details, see "Importing a Sonnet Project" on page 254.

Analyzing Outside Microwave Office

The second method for working outside Microwave Office while performing your EM analysis is to use the Analyze Outside MWOffice command in Sonnet's project editor. This command allows you to launch an analysis from the project editor in such a way that once the analysis is running, you may exit the project editor which unlocks Microwave Office. During this command, you specify a temporary Sonnet project file for your EM Structure. Once the analysis is complete, you will need to import the temporary Sonnet project into Microwave Office.

To analyze outside Microwave Office, do the following:

- 1 Open your EM Structure in the Native Editor.**

Sonnet's project editor is opened with your EM Structure shown.

- 2 Select File ⇒ MWOOffice ⇒ Analyze Outside MWOOffice from the main menu of the project editor.**

If you have not turned off the message in the Hints Preferences, then a message explaining the command appears on your display.

- 3 Click on the OK button to close the message.**

A Browse window appears on your display which allows you to select a temporary location for the Sonnet project to which the EM Structure is exported and the analysis will be performed on.

- 4 Select a name and location of the Sonnet project to which you wish to save the EM Structure.**

This saves the translated EM Structure presently open in the project editor as a Sonnet project and launches the EM analysis.

- 5 Once the analysis monitor appears on your display, indicating that the analysis is started, you may exit out of the project editor by selecting File ⇒ Exit.**
- 6 The project editor is closed and Microwave Office is unlocked.**

When the analysis is complete in Sonnet, you will need to import the temporary project and its analysis results. For details on how to do this, see "Importing a Sonnet Project" on page 254.

Importing a Sonnet Project

Both methods discussed above, exporting a project and analyzing outside Microwave office, require you to explicitly perform updates of your EM Structure by opening your EM Structure in the project editor, then selecting the Load command. Once you close the project editor, the Sonnet project contents are sent back to Microwave Office, including any analysis results.

To import a Sonnet project into Microwave Office, perform the following:

- 1 Open the Microwave Office from which the original EM Structure came.**

- 2 Right-click on the EM Structure and select "Open in Native Editor" from the pop-up menu which appears.**

The project editor is opened on the selected EM Structure.

- 3 Select File ⇒ MWOffice ⇒ Load Sonnet Project from the main menu of the project editor.**
- 4 When the Browse window appears, select the Sonnet Project which you wish to import into Microwave Office.**

The Sonnet project is imported into Microwave Office when you exit out of the project editor. Be aware, however, that the response data is imported only when an analysis is executed in Microwave Office.

Translation Issues

There are features in both Microwave Office and Sonnet which do not translate and several translation issues you should be aware of before using the Microwave Office Interface. These are all discussed in the sections following.

Sonnet Features Not Available in Microwave Office

These features are handled differently depending whether you are using Microwave Office to edit your EM Structure or Sonnet's project editor (Native Editor). The table below lists the Sonnet features not available in Microwave Office and how the feature is handled with each editor.



WARNING

When a conversion or deletion occurs when you are editing in Microwave Office, the change is permanent and is not restored if you open the EM Structure in the project editor.

Sonnet Feature	Edit in Microwave Office	Edit in Sonnet's Project Editor
Thick Metal metal type	Converted to thin metal.	Kept in circuit but displayed as thin metal in Microwave Office
Dimensions	Deleted from geometry	Kept in circuit but not displayed in Microwave Office
Subdividers	Deleted from geometry	Kept in circuit but not displayed in Microwave Office
Parameters	Deleted from geometry	Kept in circuit but not displayed in Microwave Office
Parallel Subsections	Deleted from geometry	Kept in circuit but not displayed in Microwave Office
Linked Reference Planes	Converted to a fixed length reference plane	Displayed in Microwave Office as fixed length reference planes
Calibration Lengths	Deleted from geometry	Kept in circuit but not displayed in Microwave Office

Sonnet Feature	Edit in Microwave Office	Edit in Sonnet's Project Editor
Edge Vias	Deleted from geometry	Kept in circuit but not displayed in Microwave Office
Circular and Polygon Vias	Converted to rectangular vias	Kept in circuit but displayed as rectangular via in MWOOffice.
Symmetry	Deleted from geometry	Kept in circuit but not displayed in Microwave Office
Dielectric Bricks	Deleted from geometry	Kept in circuit but not displayed in Microwave Office
Relative magnetic permeability (Mrel) and magnetic loss tangent (Mag Loss Tan) for Dielectric Layers	Mrel set equal to 1 and Mag Loss Tan set equal to 0	Kept in circuit but not displayed in Microwave Office
Dielectric Layer Names	Deleted from geometry	Kept in circuit but not displayed in Microwave Office
Capacitance and Inductance in Port definitions	Capacitance and Inductance set equal to 0	Kept in circuit but not displayed in Microwave Office
Inductance (L_s) for metals	Set to a value of zero.	Kept in circuit but not displayed in Microwave Office
Comments	Deleted from geometry	Kept in circuit but not available in Microwave Office
Metal Types	See "Metal Types" on page 258	See "Metal Types" on page 258
Port Mapping	See "Port Mapping" on page 260	See "Port Mapping" on page 260
Ports on Ground Plane	See "Ports on Ground Plane" on page 261	See "Ports on Ground Plane" on page 261

Sonnet Feature	Edit in Microwave Office	Edit in Sonnet's Project Editor
Port Termination and Excitation	See "Port Termination and Excitation" on page 261	See "Port Termination and Excitation" on page 261
Off Grid Placement	See "Off Grid Placement" on page 262	See "Off Grid Placement" on page 262
Coordinate System	See "Coordinate System" on page 262	See "Coordinate System" on page 262

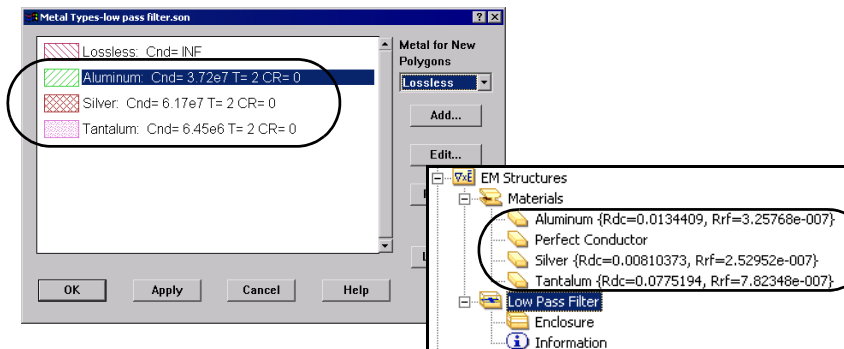
Metal Types

Models

There are six different metal type models available in Sonnet: Normal, Resistor, Rdc/Rrf, General, Sense Metal and Thick Metal Model. Each model has a different set of parameters. When your Sonnet project is imported into Microwave Office the metal types are converted to an equivalent model using the parameters Rdc, Rrf and Xdc (DC resistance, skin effect and DC reactance, respectively) such that the model yields the same loss characteristics in Microwave Office as are calculated in the Sonnet analysis. This is the equivalent in Sonnet of the General metal type with the inductance, Ls, always set to zero.

If you are using the Native Editor, Sonnet's project editor, the metal types retain their definition in the Sonnet project, but when they are displayed in Microwave Office, they will have their parameters converted to the three parameters, Rdc, Rrf and Xdc. The parameters being used in Sonnet are not displayed. The metal type appears using the same name in the Microwave Office project browser under Materials in the EM Structure folder and under metal types in Sonnet's project editor.

Shown below is the Metal Types dialog box in Sonnet with three additional metals defined and the same materials represented in Microwave Office. Note that the three metals use the Normal model in Sonnet which is converted to the Rdc, Rrf, and Xdc parameters for Microwave Office.



Loss

If you define a metal type in Sonnet which is not lossless, the metal type is added to the Materials list in Microwave Office when the EM Structure is brought back into Microwave Office. If you subsequently change the loss definition for that metal type in Sonnet, one of two behaviors will occur:

- If you change the definition of a metal type such that it still has loss but different parameters than the original definition, a new material is created in Microwave Office whose name is the original material name with an "X" added to the end of the name. For example, if you have a material named Gold, whose loss parameters you modify in Sonnet, the new material in Microwave Office which is created when you input the circuit from Sonnet at the end of the editing session will be named GoldX.
- If you change the definition of a metal type such that it is a lossless metal, any polygons which use the metal type are changed to the default lossless metal type when the EM Structure is brought back into Microwave Office. The definition of the metal type, or material, as it is referred to in Microwave Office, remains the same in Microwave Office.

In both these cases, these changes only affect the display of the EM Structure in Microwave Office. The polygons retain the metal type and its definition when opened again in Sonnet.

Adding, Changing or Deleting Metal Types

When editing in the Native Editor, if you wish to delete a metal type, you must do so both in the Sonnet environment and the Microwave Office environment. Deleting a metal type from the Sonnet project does not automatically delete it from the EM Structures Materials list in Microwave Office.

Port Mapping

Microwave Office only allows consecutive numbering for ports starting at the value of one. The port number must be a positive integer. Sonnet allows you to have non-consecutive and duplicate port numbers and allows you to use negative port numbers. Sonnet's analysis engine, *em*, sums the total current going into all the positive ports with the same port number and sets that equal to the total current going out of all the ports with that same negative port number.

When the EM Structure is sent to Sonnet from Microwave Office, Sonnet uses the port numbers sent by Microwave Office, so there is no issue in going from Microwave Office to Sonnet.

If you are using Sonnet in Native Editor mode, in which you may edit the EM Structure using Sonnet's project editor, you may enter non-consecutive as well as negative port numbers. Any port number entered in Sonnet will be retained as part of the Sonnet project file saved in the Microwave Office project but will only be displayed correctly in the Sonnet environment. The ports are mapped from Sonnet to Microwave Office by mapping the ports in Sonnet in ascending order and deleting negative and duplicate ports to a consecutive order which may be

displayed in Microwave Office. For example, if the Sonnet project uses port numbers 1, 4, -4, and 7, when the project is translated to the Microwave Office environment, the conversion would be as shown in the table below:

Sonnet Port Number	MWOffice Port Number
1	1
4	2
-4	None - the port is deleted
7	3

Ports on Ground Plane

Sonnet allows you to place a port on the ground plane; this is an error condition in Microwave Office which prevents an analysis from being executed. A port on the ground plane in Microwave Office is translated into a via port which extends from the ground plane (substrate) up to the level above.

Port Termination and Excitation

A port may be defined in two ways in Microwave Office: termination and excitation. You define the termination of a port as the real and imaginary parts. This translates directly into Sonnet as the Resistance and Reactance parameter of the port. Microwave Office does not allow you to define the Inductance or Capacitance of a port, unlike Sonnet. When using Microwave Office as the editor, these two values are set to 0. When using the Native Editor, you may enter values for the Inductance and Capacitance. They are saved as part of the Sonnet project, but are not displayed in Microwave Office.

Microwave Office allows you to define the excitation of one port in your circuit by entering the Available Power and Phase. Sonnet does not have the ability to define port excitation in the project editor. Therefore, if an EM Structure is translated into Sonnet, these values are deleted for the port and the termination definition is used.

Sonnet does provide port excitation in both the current density viewer and the far field viewer. For details, see online help for either program.

Off Grid Placement

Microwave Office's resolution between points on the grid is 1/20 of a cell. Sonnet uses a much finer resolution; therefore, if you change the position of an off grid polygon, the placement may appear slightly different when you bring the EM Structure back into Microwave Office.

Coordinate System

Microwave Office and Sonnet use different coordinate systems. The origin (0,0) in Sonnet is the lower left hand corner of the substrate while the origin in Microwave Office is the upper left hand corner of the substrate which Microwave Office refers to as the ground plane.

3D Viewer Scaling

Microwave Office's 3D viewer has a scale setting for the dielectric layers which allows you to display a dielectric layer larger than it is relative to the other dielectric layers. If you open the EM Structure in Sonnet in Native Editor mode there exists no way to specify the 3D viewer scaling.

Meshing, Current Density Plots and Far Field Plots

When you use Sonnet as your analysis engine, you may not view meshing, current density plots or far field radiation plots in Microwave Office. The Animation menu in Microwave Office is disabled when Sonnet is selected as your analysis engine. For details on viewing this data in Sonnet, see "Response Data" on page 248.

Troubleshooting

Unlocking Microwave Office

When you open Sonnet's project editor from Microwave Office, Microwave Office is locked until you close the project editor. You may not make any changes or execute any commands in Microwave Office until it is unlocked when Sonnet is closed. Communication between Sonnet and Microwave Office is done through the Sonnet program, *sonntawr*. When Sonnet closes, *sonntawr* signals Microwave Office that it is unlocked.

If for some reason the Sonnet program is interrupted in such a way as to prevent a graceful shutdown, such as a hung program or a program crash, then Microwave Office is not unlocked. If the project editor (*xgeom*) is no longer running, but Microwave Office is still locked, you may manually unlock it by selecting *Admin* \Rightarrow *MWOffice Interface* \Rightarrow *Unlock MWOffice* from the Sonnet task bar main menu. To open the Sonnet task bar, select *Start* \Rightarrow *Programs* \Rightarrow *Sonnet* \Rightarrow *Sonnet* from the Windows desktop Start menu.

Project Editor does not open due to a licensing problem

When you open the Native Editor from Microwave Office, the project editor attempts to obtain a license before sending Microwave Office the lock message. If the project editor is not successful in obtaining a license, then the lock message is not sent. Microwave Office waits one minute before timing out on waiting for the lock message and is unresponsive during this time.

Coordinate System

Microwave Office and Sonnet use different coordinate systems. The origin (0,0) in Sonnet is the lower left hand corner of the substrate while the origin in Microwave Office is the upper left hand corner of the substrate which Microwave Office refers to as the ground plane. This fact is important to remember if you receive error messages which specify a grid position. Error messages from Sonnet are based on Sonnet coordinates and may need to be converted to Microwave Office coordinates to effectively diagnose the source of the error.

Chapter 15 The DXF and GDSII Translators

The DXF and GDSII translators allow you to convert a DXF or GDSII file, respectively, to a Sonnet project compatible file. Once you have converted your DXF or GDSII file to a Sonnet project, you then need to tweak it using the project editor before you can analyze the circuit with *em*. This chapter discusses some advanced features of the project editor and assumes that the user is very familiar with the project editor.

NOTE:

The DXF and/or GDSII translator is only available if you have purchased a DXF and/or GDSII translator license from Sonnet. Please see your system administrator if you are unsure of the availability of these programs.

The DXF and GDSII translators can also be used to convert a Sonnet project to a DXF or GDSII file, but these files are limited to the information contained in the project. See the *File* \Rightarrow *Export* command in the project editor's Help for details.

Converting your files to Sonnet

The DXF or GDSII translator inputs a DXF or GDSII format input file from a circuit layout program and converts it to a Sonnet project. The translator is accessed through the project editor. When you select *File* \Rightarrow *Import* \Rightarrow *DXF* or *File* \Rightarrow *Import* \Rightarrow *GDSII* from the project editor menu, the following occurs:

- 1 A Browse window opens which allows you to select the DXF or GDSII file you wish to import.
- 2 Once you select a DXF or GDSII file for import, the Import Control dialog box appears which allows you to assign the project to which you wish to import the file.
- 3 If you are importing a GDSII file, the Structure dialog box appears which allows you to select the structure in the GDSII file you wish to convert to Sonnet.
- 4 Then the project editor reads in the layers from the DXF or GDSII file and once the layer mapping is complete, the Import dialog box appears in which you select translation options, if any, and execute the file import.

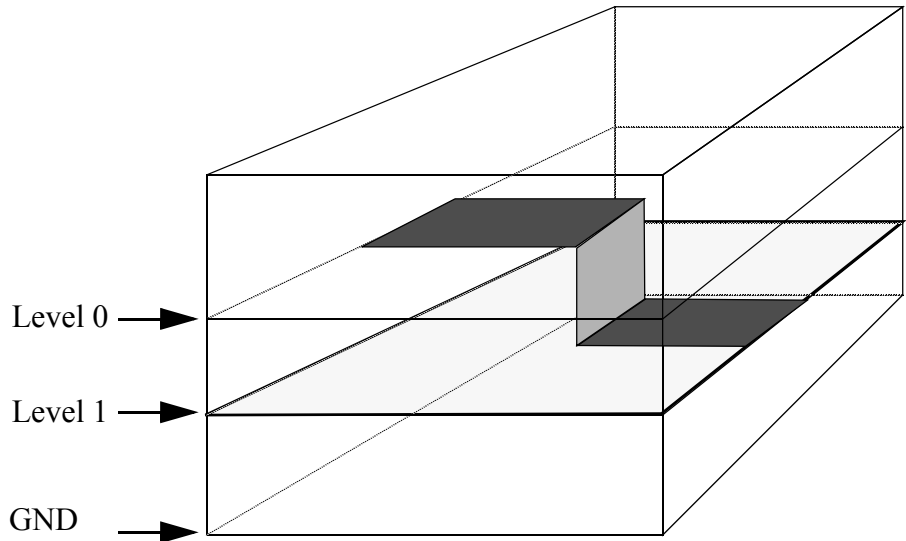
Once the DXF or GDSII file has been imported, you normally need to perform some minor adjustments using the project editor. For a detailed example of a DXF or GDSII import, see Chapter 6, "GDSII and DXF Translator Tutorial," in the **Sonnet Supplemental Tutorials**.

Layer Mapping

The Layer Mapping dialog box, accessible from a button in the Import dialog box allows you to map the DXF layers or GDSII streams to the metal levels in Sonnet, if it is necessary. You may save the settings in this dialog box to an external file for use in future imports. See the appropriate section below for more details.

NOTE:

The top level of your circuit is on Level 0 in the project editor. Next level below is Level 1. (See the figure below)



The Project Editor Level Numbering in a 3 layer circuit.

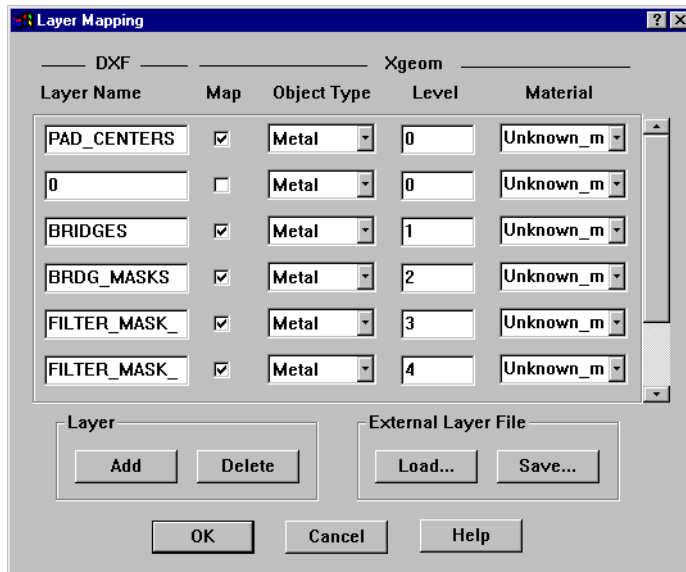
DXF Layer Mapping

Before you convert the file, you must create a DXF file from your circuit layout program. The file should have a suffix of “.dxf”. For example, if you have a filter that you want to analyze, you could call your file “filter.dxf”.

During the input of a DXF file, the DXF translator reads the layers in the DXF file and creates the default layer mapping which is used to execute the import. In most cases, the default layer mapping created by the translator is sufficient for the import. In cases where you need to adjust the mapping, you can use the Layer

Mapping dialog box to edit the layer mapping, save the settings to an external layer mapping file or load a previously saved layer mapping file for use in an import.

The Layer Mapping dialog box, shown below, maps Layer Names to Sonnet Level Numbers and Material Names. After you enter the correct data for your specific needs, you have the option of saving the settings to a “.lay” file for use in future conversions.



The first column refers to the incoming DXF file. The last three columns are the choices to be used in the Sonnet project for the DXF layer identified in the first column.

DXF Layer Name: The layer name defined in your circuit layout program. Note that the layer name has no relationship to the project editor's dielectric layer number or metallization level number. This field can be used to map a single DXF stream to multiple project editor levels. By using the Add button, you can have multiple entries for the same stream layer which are mapped to different levels in the Sonnet project.

Map: Use this checkbox to select which layers in the DXF file you wish to map to the Sonnet project. If this checkbox is off, the DXF layer is not mapped to the Sonnet project. If the checkbox is on, then the DXF layer is mapped to the level specified in the Xgeom Level text entry box.

Xgeom Object Type: This drop list allows you to select what type of object should be input to the Sonnet project. Select Metal for metalization, Brick for dielectric brick, or Via for a via polygon from the drop list. You select the metal type (for Metal or Via) or brick material from the Xgeom Material drop list. This selection is used for the layer number entered for the DXF file in the same row.

Xgeom Level: This specifies the level in the Sonnet project where you wish to place the object being input from the DXF file. The top level is number 0. To put metal on the ground plane, for via placement, set the level number to “gnd” or “GND”.

Xgeom Material: Select the material type you wish to use for the object in the Sonnet project. The choices available depend on what type of object is selected in Xgeom Object type, Metal or Brick and what materials are available in the template project. This selection is used for the layer entered for the DXF file in the same row.

For example, if you are importing a DXF file into a project with four defined types of metal, you may choose from those four metal types for any object being translated in metal in the Sonnet project. You may also enter a new type of material, for which you enter the parameters later in the project editor.

External Layer File: Once the mapping is specified, you may wish to save the settings for subsequent imports of this DXF file. Click on the Save button to save your settings in a file. It is recommended that you use a “.lay” extension which is the default.

If you have an existing layer file with mapping settings for the DXF file you are importing, click on the Load button to select the layer file and load its contents in the dialog box.

GDSII Layer Mapping

Before you convert the file, you must create a GDSII file from your circuit layout program. The file should have a suffix of “.gds”. For example, if you have a filter that you want to analyze, you could call your file “filter.gds”. Next you must enter the layer mapping to tell the GDSII conversion program the mapping of stream layers and data types to project editor levels and data types. To help you with this process, the GDSII translator opens the Layer Mapping dialog box with the layers from your GDSII file that can be used as a starting point.

The Layer Mapping dialog box, shown below, maps Stream Layer Numbers and Data Type Numbers to Sonnet Level Numbers and Material Names. After you enter the correct data for your specific needs, you have the option of saving the settings to a “.lay” file for use in future conversions.

Layer Name	Map	Object Type	Level	Material
PAD_CENTERS	<input checked="" type="checkbox"/>	Metal	0	Unknown_m
0	<input type="checkbox"/>	Metal	0	Unknown_m
BRIDGES	<input checked="" type="checkbox"/>	Metal	1	Unknown_m
BRDG_MASKS	<input checked="" type="checkbox"/>	Metal	2	Unknown_m
FILTER_MASK_	<input checked="" type="checkbox"/>	Metal	3	Unknown_m
FILTER_MASK_	<input checked="" type="checkbox"/>	Metal	4	Unknown_m

The first two columns refer to the incoming GDSII file. The last three columns are the choices to be used in the Sonnet project for the GDSII elements identified in the first two columns.

GDSII Stream: The layer number in your circuit layout program. Note that this layer number has no relationship to the project editor's dielectric layer number or metallization level number. This field can be used to map a single GDSII stream to multiple project editor levels. By using the Add button, you can have multiple entries for the same stream layer which are mapped to different levels in the Sonnet project.

GDSII Type: The data type number as defined in your circuit layout program. Each layer, defined by your circuit layout program, may have multiple data types. This field can be used to map different Data Types in a stream layer to different Sonnet objects and materials. By using the Add button, you can have multiple entries for the same stream layer with different Data Type numbers which are mapped to different objects and/or materials in the Sonnet project.

Map: Use this checkbox to select which layers in the GDSII file you wish to map to the Sonnet project. If this checkbox is off, the GDSII stream layer is not mapped to the Sonnet project. If the checkbox is on, then the GDSII stream layer is mapped to the level specified in the Xgeom Level text entry box.

Xgeom Object Type: This drop list allows you to select what type of object should be input to the Sonnet project. Select Metal for metalization, Brick for dielectric brick, or Via for a via polygon from the drop list. You select the metal type (Metal or Via) or brick material from the Xgeom Material drop list. This selection is used for the stream layer and data type number entered for the GDSII file in the same row.

Xgeom Level: This specifies the level in the Sonnet project where you wish to place the object being input from the GDSII file. The top level is number 0. To put metal on the ground plane, for via placement, set the level number to "gnd" or "GND".

Xgeom Material: Select the material type you wish to use for the object in the Sonnet project. The choices available depend on what type of object is selected in Xgeom Object type, Metal or Brick and what materials are available in the template project. This selection is used for the stream layer and data type number entered for the GDSII file in the same row.

For example, if you are importing a GDSII file into a project with four defined types of metal, you may choose from those four metal types for any object being translated in metal in the Sonnet project. You may also enter a new type of material, for which you enter the parameters later in the project editor.

External Layer File: Once the mapping is specified, you may wish to save the settings for subsequent imports of this GDSII file. Click on the Save button to save your settings in a file. It is recommended that you use a “.lay” extension which is the default.

If you have an existing layer file with mapping settings for the GDSII file you are importing, click on the Load button to select the layer file and load its contents in the dialog box.

Using Your Layout Program to Replace the Project Editor

You may use any layout program that writes DXF or GDSII files as a replacement for the project editor. When you do this, you will probably want to convert the DXF or GDSII file without any translation in the coordinates. Normally, when either translator converts a file, it moves the circuit to a convenient place for use in the project editor. To convert a file without this translation, select “None” from the Translation drop list in the Import Options dialog box. The Import Options dialog box is opened by pressing the Options button in the Import window. For more details, see the DXF or GDSII translator in the project editor Help.

Tweaking the Circuit

Once the conversion is complete, you need to tweak the circuit using the project editor. Invoke the project editor and open the project with the converted DXF or GDSII file.

- 1 Define metallization and dielectric layer parameters.
- 2 Remove parts of the circuit not being analyzed.
- 3 Decide on a proper size substrate and cell size.

- 4 Change polygons to have the proper fill.
- 5 Align the circuit to grid points.
- 6 Move points around as needed.
- 7 Add vias.
- 8 Add ports and reference planes.

Depending on your circuit, all of these steps may not be needed. However, it is recommended that you try to follow the steps in the order given. The “GDSII and DXF Translator Tutorial” in the **Sonnet Supplemental Tutorials** uses an example to describe each of these steps in detail.

Specifying Units for a DXF Translation

The DXF translator assumes that the circuit was input using inches as the units. The Import Options dialog box allows you to specify the units. Select the desired Unit from the Units drop list in the Import Options dialog box. The Import Options dialog box is opened by pressing the Options button in the Import window. For more details, see the DXF translator in the project editor Help.

Error in Processing the DXF File

You may receive an error message from the DXF translator stating that the format of the DXF file is incorrect. The error message comes in many different forms. An example is:

```
dxfgco ERROR - illegal dxf type number at line 205 - $DIMALT
```

Most error messages stop the conversion; however, there are two error messages that do not stop the conversion. They both tell you that there will not be any polygons in the resulting project. One, because all layers in the DXF file or in the Layer Mapping dialog box were marked as “OFF” and the second because there were no polygons in the layers that were marked as “ON.”

In either case, it is most likely you can edit the Layer Mapping using the Layer Mapping dialog box and turn the appropriate layer(s) back "ON". See section "DXF Layer Mapping" on page 267 for details on the layer mapping.

Another possibility is the AutoCad version of your DXF file is not compatible with Sonnet. If you are having trouble importing a DXF file into Sonnet, try saving your DXF file from your drafting tool as an earlier Autocad version. This is usually possible from most tools.

Capabilities and Limitations of DXF Translation

The DXF translator can handle most methods of describing a circuit that is supported by the DXF file format. The DXF translator does not handle any parts of the DXF format that is not meaningful to a description of metallization of a substrate, such as text, dimension lines, and 3D structures. The DXF translator handles four basic methods of circuit description. All of these methods can be used within a single DXF file:

- Closed, zero width polylines whose interiors define the metallization areas. These polylines are closed by the DXF translator, even if they are not marked as closed in the DXF file. Arcs inserted into these polylines are supported.
- Closed or open polylines with a given width. The area covered by the width of the line defines the metallization area.
- Circles or arcs whose interiors define the metallization areas. These circles and arcs will be converted into small line segments that resemble the original object. The arc conversion angle in the DXF Options dialog box is the angular step size used to convert to polygons. The default value is 10 degrees, which would divide a circle into 36 line segments.
- A series of lines, open polylines and arcs which define a closed area of metallization. The DXF translator searches all lines, open polylines and arcs for possible connection of common end points. Any connecting items are strung together to form a closed polygon of the type described in method number 1 above. The Combining Tolerance, entered in the Import Options dialog box, defines a toler-

ance to be used to determine if end points are close enough to be considered connected. The Arc Conversion Angle, also entered in the Import Options, affects how arcs that become part of the polygon are converted to line segments. This method should be used carefully and only when needed. The method will not work properly at points where three or more lines share common end points.

Blocks are handled by the DXF translator, including scaling, rotation and row/column replication and may be useful in any of the above methods.

The following is a list of some of the items not handled by the DXF translator:

- Entities not handled include: POINT, TRACE, SOLID, TEXT, SHAPE, ATTRIBUTES, 3DFACE, VIEWPORT and DIMENSION.
- Tables not handled include: APPID, DIMSTYLE, LTYPE, STYLE, UCS, VIEW and VPORT.
- The only header variables read are: \$EXTMIN, \$EXTMAX, \$LIMMIN and \$LIMMAX. All others are ignored.
- In the layer table, color, line type and frozen and locked flags are ignored. The name, and on/off flags are read.
- The Z coordinate of all objects is ignored. Only the layer is used to determine the Z position of the metal.
- The color and line styles are always ignored.
- Only the starting width of the first point in polylines are used, the ending width and both widths of other points are ignored.
- Polyline curve-fit and spline-fit are not handled.
- No 3D structures are handled, including polygon mesh, polyface mesh and smooth surface.

Error in Processing the GDSII File

You may receive an error message from the GDSII translator stating that the format of the GDSII file is incorrect. The error message comes in many different forms. An example is:

```
ERROR: Encountered a record with the wrong data type  
This record has a data type of NO_DATA  
The data type should be TWO_BYTE_INT  
Record type is HEADER  
Error encountered in record 3
```

The most common reason for this message is that the GDSII file was changed when transferring between computers. This typically occurs when transferring the file from a PC to a UNIX workstation. The GDSII file is a binary file and not an ASCII file. A common mistake made when you transfer programs between computers is to assume that you are transferring ASCII files and, consequently, need to make changes to the files for compatibility.

For example, MS-DOS and UNIX use different characters to indicate the end of a line in ASCII files. This is the reason that most UNIX computers have the commands “unix2dos” and “dos2unix”. These programs are used to convert ASCII files when transferring them between a PC and a UNIX computer. If a “dos2unix” conversion is done on a GDSII file, then it will be corrupted.

To determine if the GDSII file has been corrupted, determine how many bytes the GDSII file has on the original computer and also on the computer where the GDSII translator conversion is done. Both files should have the same number of bytes. When transferring files between computers, make sure that the transfer is using binary transfer mode. For example, FTP has two transfer modes: ASCII and binary. Also, do not perform a “dos2unix” conversion on the GDSII file if transferring the file from a PC to UNIX.

Partial Circuit Conversion in a GDSII Import

You may have done a conversion but the Sonnet project has no circuit or only part of the circuit. First verify that the settings in the Layer Mapping dialog box are correct. You may wish to re-enter the settings.

Another reason that only part of your circuit is shown in the project editor, may be that only part of the GDSII file was converted. Every GDSII file contains structures. A structure may be referenced in another structure. The GDSII file may even have multiple structures that have no relation to each other. If a GDSII file contains multiple structures, then you must select the structure you wish to convert. If you select default, then the conversion program must decide which structure to convert. Typically most layout programs create the GDSII file with the first structure as the parent structure. By default the GDSII translator converts the first structure in the GDSII file. However, your layout program may have created a GDSII file where the structure to convert is not the first structure. In this case you must click on the Specify Structure option in the Structure dialog box, then click on the Specify button. When the list of available structures is opened, select the desired structure, then continue with the import.

Chapter 16 The Agilent Interface

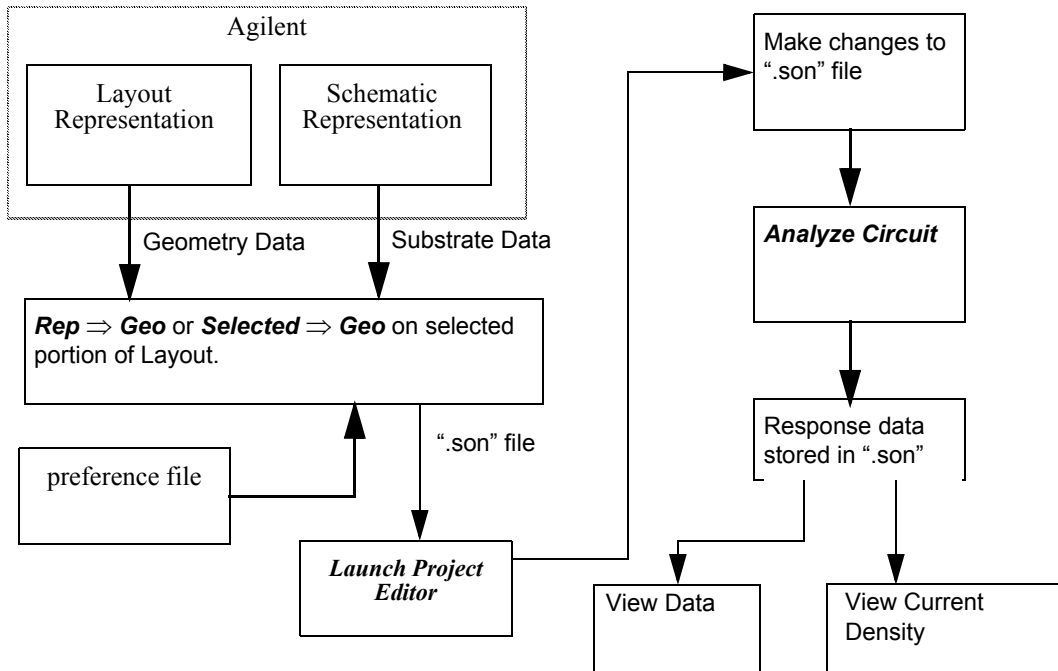
The Agilent interface constitutes an interface between Agilent's ADS circuit analysis program and Sonnet's suite of electromagnetic analysis tools. You must have ADS version 1.3 or above to use the interface. The software provides a menu driven package that can translate the layout of a design to the Sonnet project format, and also launch sessions of Sonnet's project editor, analysis engine and current density viewer. After a translation from ADS, a few minor adjustments are usually required using the project editor before you can analyze the circuit with *em*.

This chapter describes some advanced features of the project editor and assumes that the user is familiar with the Sonnet software suite as well as Agilent's ADS. For instructions on how to install and configure this software, refer to the "Configuration Procedure for *ebriidge*" in the appropriate installation manual. Translated files created by the Agilent Interface are compatible with Sonnet version 10 or higher.

A tutorial on using the Agilent Interface may be found at Chapter 7, “Agilent Interface Tutorial,” in the **Sonnet Supplemental Tutorials**.

Overview of the Agilent Interface

The Agilent interface will load when a Touchstone or Libra project is opened. When loaded, the interface creates a special menu labeled *Ebridge* in the command window. Any previous menu items in the user menu of this window will be removed. If you have trouble getting this menu please see: “Troubleshooting,” page 294. A process flow diagram of the Agilent interface software is shown below.



Basic Program flow for the Agilent interface.

The menu options are listed as follows:

<i>Rep ⇒ Geo</i>	Translate Entire layout into Sonnet 9 format - .son project.
<i>Selected ⇒ Geo</i>	Translate selected portion of layout into Sonnet 9 format - .son project.
<i>Edit Preference File</i>	Opens an editing session of the Agilent Translator preference file. For details about the preference file, please refer to “The Preference File,” page 286.
<i>Launch xgeom</i>	Launch Sonnet project editor session.
<i>Launch Sonnet</i>	Launch Sonnet session. The Sonnet task bar appears on your display.
<i>Find Translated Files</i>	Find the desired Sonnet project and opens the design of the same name.
<i>Launch user window</i>	For UNIX, launch command window in present directory.
<i>Launch Runwin</i>	For Windows, a session of the Sonnet Run Window is launched in the Sonnet “project” directory.
<i>Sonnet Info</i>	Displays information about <i>ebridge</i> and how to contact Sonnet Software.

In order to translate a circuit from layout to Sonnet 9 format, open your design in the layout window. Choose the **Rep ⇒ Geo** item in order to translate the entire design into the Sonnet 9 format. Choose the **Selected ⇒ Geo** item in order to translate the selected portions of the design into the Sonnet 9 format. After translation, the Sonnet project can be examined with the **Launch xgeom** item. After some adjustments are made to the file and saved, analyze the translated file from within **xgeom** and create S-parameter files for incorporation into the design. When the **Selected ⇒ Geo** menu item is used, the S-parameter output file will automatically be referenced in the schematic.

NOTE: **The Agilent Interface does not translate customized design files**

Translation Procedure

There are two ways to initiate a translation, the **Rep** \Rightarrow **Geo** menu item and the **Selected** \Rightarrow **Geo** menu item. The **Rep** \Rightarrow **Geo** command will convert an entire representation (the whole layout) into a Sonnet project. The **Selected** \Rightarrow **Geo** command converts only the selected area of the layout into a project. In addition, **Selected** \Rightarrow **Geo** performs a few other functions in order to make the use of **em** simulation in a design easier. These special features are described later in “Special features of Selected \Rightarrow Geo,” page 284.

Some preparation of the design and preference file may be required in order to ensure that the translation program obtains all the correct data required to perform the translation appropriately. The translator obtains its data from several different sources. These sources are outlined below.

Substrate parameters

Substrate parameters are set by the instances of MSUB, SSUB, SSSUB, SSUBO, PCSUB, TAND, MCOV, and PERM in the schematic of your design. Particular SUB instances will only be recognized if they are referenced by the translated elements. Otherwise default parameters will be used. If no default is found, the substrate parameters will default to microstrip with a zero thickness air dielectric.

Substrate data instances resident in the schematic will only be located if they are referenced by circuit elements in the schematic or if they have “_DEFAULT” as the last part of their name. If more than one substrate instance is located, only the last one found will be used. MCOV, TAND, PERM, TANDM instances are located in either the schematic or the default design. They do not need to be referenced by circuit elements but only the last one found in the schematic will be used by the Agilent interface.

Box Settings

An *em* analysis takes place inside a bounded metal box with lossless sidewalls and user definable loss on the top and bottom of the box (see “The Preference File,” page 286). The box size in the X and Y dimensions (area) is set by the X and Y extent (bounding box) of the translated circuit (including text labels and/or drawing formats) plus a substrate extension added to all sides. The substrate extension is equal to the substrate height multiplied by the preference parameter, EXT. The box size in the Z dimension is specified by the pertinent substrate instance (MCOV, SSUB, etc.). The box gridding parameters are set by the NUM_X_GRIDS and NUM_Y_GRIDS preference parameters. Extension lines and reference planes will always be added in order to provide de-embedded S-parameters to the original circuit dimensions.

Exact control of the substrate size of your circuit is often desirable in order to control the cell size. This can be accomplished by drawing a rectangle around the circuit layout and then setting the EXT preference parameter to zero. This rectangle will set the circuit bounding box (because it is the furthest extent of the circuit) to its dimensions (as long as none of the circuit extends beyond it). It should be drawn on an unused layer and this layer should be identified in the preference file as being ignored (see “The Preference File,” page 286). This procedure is demonstrated in “Using a drawn bounding box,” page 143 in the **Sonnet Supplemental Tutorials**.

For example: A circuit is 100 by 140 mils in total extent on a 10 mil substrate. If EXT is set to 3, the final substrate size would be 160 by 200 mils. If NUM_X_GRIDS is set to 160 and NUM_Y_GRIDS was set to 200, the cell size will be 1 by 1 mil. Reference planes will be extended to the edge of the original 100 by 140 mil circuit. Conversely, a rectangle can be drawn on the layout (see above) of dimensions 200 by 240 mils, EXT can be set to zero, resulting in a final box size of 200 by 240 mils. If NUM_X_GRIDS is set to 400 and NUM_Y_GRIDS is set to 240, the cell size will be 0.5 by 1 mil. Reference planes will be extended to the edge of the original 100 by 140 mil circuit.

Special layers for special circuit elements

Certain special circuit elements such as TFC, MTFC, and MRINDSBR require special layer structures to be implemented. The Agilent interface determines this layer information by scanning the design for these instances. The Agilent interface supports only one layer for each element type (capacitor or inductor). The Agilent interface sets the level parameters to the values calculated for the first instance of a particular type found in the design. Layer structure differences in other elements will be ignored. The Agilent interface also only supports these elements on the main (top) conductor metalization level of a substrate.

The TFC or MTFC instances add a dielectric layer over the top layer of the substrate according to the parameters of the instance. The MRINDSBR instance adds a bridge layer over the capacitor dielectric layer. These elements will be translated onto these layers and connected appropriately.

The Agilent interface also scans for the TFR element. This sets the resistivity (ohms per square) for the TFR layer (number 3). The first TFR element found sets the resistivity value for all the rest.

Special layers can also be implemented for other instances using the CAP_LAYER and BRIDGE_LAYER statements in the preference file (see “The Preference File,” page 286 for more detail).

Special features of Selected \Rightarrow Geo

The **Selected \Rightarrow Geo** menu item is designed to aid designers in inserting electromagnetic models into a design. A user selects the particular part of the design to simulate and pull down the **Selected \Rightarrow Geo** menu item. The design is changed by this command, so the user might want to have two versions: a circuit theory design and an electromagnetic design.

Selected \Rightarrow Geo creates a sub design from the selected portion of the layout and schematic called “<design name>_em_#” where # is equal to 1,2,...N depending on the number of electromagnetic sub-designs created from this design. **Selected \Rightarrow Geo** then translates the layout of the sub design, creating a project called <subdesign>.son. **Selected \Rightarrow Geo** then creates a schematic for the sub design

which references the correct SNP data file <sub-design>.snp (sNN for number of ports greater than 9), wires up the ports and saves the new sub design. The modified top level design is not saved, so that the previous state can be recovered in case of a mistake. The user then checks the project by running **Launch xgeom** and analyzes the file.

If any adjustments are made to the sub design layout after the first translation, use the **Rep** \Rightarrow **Geo** menu item with the sub design in the layout window to recreate the project with the new dimensions.

New and user defined instances

The Agilent interface translates most of the existing planar circuit elements. The software can check each instance against a list of instances that it knows it can handle prior to translation. This feature is turned off by default. If you experience problems with translation, you should enable this feature. See “TEST_VALID,” page 294 for the keyword needed to use this feature.

If an instance is not found on this list, the software issues a warning message and ignores the instance. Instances identified in this manner will be placed in a file called “**bad_instances**” in the project directory. The list of valid instances is located in the file `$SONNET_DIR/ebridge/defaults/valid_instances`. If a file called “**valid_instances**” is located in your project directory, it will be read and appended to the list. If your company has certain user defined instances that you would like to translate, just add them to the list and try them out. If they do not translate correctly, or cause errors in the software, remove them from the list. Sonnet Software cannot guarantee compatibility with these instances.

If you are using custom instances (such as with smart libraries) closely review the LAYER, CAP_LAYER and BRIDGE_LAYER statements in the preference file. Using these statements can add a lot of power to translating your custom instances.

The Preference File

The Agilent interface utilizes a preference file that contains information for customizing translation. To edit the preference file, select **Ebridge** ⇒ **Edit Preference File** from the main menu of ADS.

The Agilent interface first looks for a file called <*design name*>.prf located in the same project directory as the design. If this file does not exist, the file *sonnet.prf* is used. If neither of these two files is found, a default will be copied into the directory and used. The default preference file is shown below.

The preference file format consists of individual command lines (statements) each consisting of a keyword followed by a space and then space delimited keywords and/or data. The preference file can be created or modified with any suitable text editor. An exclamation point (!) indicates a comment; anything appearing after it is ignored. Blank lines are also ignored.

```
! Preference File for EBRIDGE example directory
! Copyright (c) 2001 Sonnet Software, Inc., All rights reserved.
! translation options
! default gridding
NUM_X_GRIDS 256
NUM_Y_GRIDS 256
! grid for circuit_1
! NUM_X_GRIDS 465
! NUM_Y_GRIDS 300
LAYER 13 IGNORE
! comment out the EXT statement not used
! EXT 0
EXT 3
! TOP or BOT set top cover or ground plane loss
! use 377 ohms/square for free space
! TOP 377 0 0
```

Default sonnet.prf file.

Translator preferences

Translator preferences specify how a layout is to be processed to form a Sonnet project. They are not required to perform a translation but they add an extra level of control over the translation process.

NUM_X_GRIDS

Syntax: NUM_X_GRIDS <ncells>

This statements sets the total number of cells across X dimension. The default is ncells=256.

NUM_Y_GRIDS

Syntax: NUM_Y_GRIDS <ncells>

This statements sets the total number of cells across Y dimension. The default is ncells=256.

EXT

Syntax: EXT <n>

This parameter is used in determining box size (X,Y), referred to as Substrate Area by the project editor. A single number following this keyword sets the multiplier for the substrate height used to add extensions to each side of the circuit dimensions to determine the area of the project editor box. De-embedding reference planes and additional lines are added to extend the port S-parameters in to the original circuit dimensions. This parameter insures that the circuit elements are kept sufficiently far away from the box sidewalls in the *em* analysis. The use of this parameter is explained further in “Box Settings,” page 283. If not specified the EXT parameter defaults to 3.

LAYER

Syntax #1: LAYER <n1> <n2> <n3> [VIA] [NO_PORT]

Syntax #2: LAYER <n1> IGNORE

This parameter is used to determine the mapping of Agilent layers to project editor layers and metalization codes. Most layer information is determined at run time by reviewing the instances in the design. This feature is utilized to index metalization loss of layers and for any special instances (possibly user defined). It is also used to remove layers from translation.

n1: a layer number.

n2: the appropriate metalization level in the project editor (usually set to 0). GND will be recognized as the ground plane and will place the Agilent layer on the bottommost layer of the Sonnet project.

n3: a number indexing a RES statement for the metalization loss of this layer.

The RES index number starts at 0 referencing the first RES and/or MET statement in the file and progresses upwards for additional statements. The number -1 is a special case and references the default lossless metal in the project editor.

IGNORE: The keyword IGNORE can be placed after <n1> to indicate that the translator should ignore the corresponding Agilent layer number.

VIA: If the keyword VIA is placed after <n3>, any polygons translated will have vias up to the next project editor level placed on every edge. These layers can be used to connect multilayer custom designs.

NOPORT: If the keyword NO_PORT is placed after <n3> or after a VIA keyword, then no ports will be connected to this layer. This helps the Agilent interface to isolate individual polygons to connect ports to (the Agilent interface can only connect ports to single polygons). This is used when multiple polygons overlap the port connection in a single instance but need to be translated rather than ignored.

RES

Syntax: RES <Rdc> <Rrf> <Xdc> [metal_name]

Set loss values to be used by the LAYER parameter. The effect of these values is detailed more specifically in “Metalization Loss,” page 74 in the **Sonnet User’s Guide**. The RES statements are indexed by the LAYER statements in the order they appear in the preference file.

Rdc: Indicates the DC resistance in Ohms per square.

Rrf: Indicates the skin effect coefficient

Xdc: Indicates the metal surface reactance.

metal_name: This optional name string can be used to label the metal index in the project editor.

TOP or BOT

Syntax1: TOP <Rdc> <Rrf> <Xdc>

Syntax2: BOT <Rdc> <Rrf> <Xdc>

Set the metalization loss of the top cover or ground plane of the **em** analysis. The parameters Rdc, Rrf and Xdc are identical to the RES statement above.

CUSTOM_SUB

Syntax: CUSTOM_SUB

< substrate lines >

END CUSTOM_SUB

This statement defines a default substrate for the translation without requiring an MSUB element.

The Syntax for the substrate lines is as follows:

```
CLAYER <thickness> [Epsilon R] [MU R] [TAND] [TANDM]  
MAIN_LAYER <thickness> [Epsilon R] [MU R] [TAND] [TANDM]
```

MAIN_LAYER indicates that this layer is where all default metal will be placed (i.e. metal not mapped elsewhere). Any additional (CAP and BRIDGE) layers will be added above this layer.

thickness: Thickness (in length units).

Epsilon R: Optional dielectric constant. If not specified, Er=1.

Mu R: Optional relative permeability. If not specified, Mu R=1.

Tand: Optional dielectric loss tangent. If not specified, Tand=0.

Tandm: Optional magnetic loss tangent. If not specified, Tanm=0.

SIZE_X_CELL, SIZE_Y_CELL

Syntax:

SIZE_X_CELL <value>

SIZE_Y_CELL <value>

This statement defines a default cell size for the translation by defining the cell size as a number of length units. The SIZE_X_CELL statement defines the length of the cell in the x direction and the SIZE_Y_CELL statement defines the length of the cell in the Y direction.

If these statements are present, they will override the NUM_X_GRID or NUM_Y_GRID statements.

PORT_FILTER

Syntax: PORT_FILTER <list>]

This statement controls which port instances get recognized as ports by the translation procedure. If this statement is present, only the port instance names listed after the keyword will be translated. If not present, all instances will be translated.

list: Any combination of the following keywords: PORT, EMPORT, ORPHAN, NPPORT. ORPHAN indicates orphaned nodes (nodes that are connected to only one pin).

The following statements may be added to your preference file to add custom layers. When one or more of the statements exists, the appropriate layer(s) is added to the project. Any combination of these statements may be included in the preference file, but only one CAP_LAYER and one BRIDGE_LAYER statement may exist. Additional layers may be added using UPPER_LAYER and LOWER_LAYER. These statements, in combination with the LAYER statements and the “valid_instances” file, can be used to implement custom layered instances specific to a company, such as a smart library. See the descriptions below for details.

CAP_LAYER

Syntax: CAP_LAYER <T> [Er] [Ur] [Tand] [Tanm]

Add a capacitor dielectric layer to the project. The layer is added above the highest main substrate layer. If a TFC or MTFC instance appears in the design, the top TFC or MTFC layer is replaced with this layer.

T: Thickness (in length units).

Er: Optional dielectric constant. If not specified, Er=1.

Ur: Optional relative permeability. If not specified, Ur=1.

Tand: Optional dielectric loss tangent. If not specified, Tand=0.

Tanm: Optional magnetic loss tangent. If not specified, Tanm=0.

Only one CAP_LAYER statement is allowed in a preference file.

BRIDGE_LAYER

Syntax: BRIDGE_LAYER <T> [Er] [Ur] [Tand] [Tanm]

Add an air bridge dielectric layer to the project. The layer is added above the highest main substrate layer and CAP_LAYER statements. If an MRINDSBR instance appears in the design, the top MRINDSBR layer is replaced with this layer. See CAP_LAYER for the definition of each parameter. Only one BRIDGE_LAYER statement is supported in a preference file.

UPPER_LAYER

Syntax: UPPER_LAYER <T> [Er] [Ur] [Tand] [Tanm]

Add a dielectric layer above the main, CAP_LAYER and BRIDGE_LAYER layers. See CAP_LAYER for the definition of each parameter. Any number of UPPER_LAYER statements may be included in a preference file. Each statement adds an additional layer, in the order stated in the file, to the top of the structure. For example, if two UPPER_LAYER statements exist, the layer from the first statement is placed above the layer from the second statement.

LOWER_LAYER

Syntax: LOWER_LAYER <T> [Er] [Ur] [Tand] [Tanm]

Add a dielectric layer below all other layers. See CAP_LAYER for the definition of each parameter. Any number of LOWER_LAYER statements may be included in a preference file. Each statement adds an additional layer, in the order stated in the file, to the bottom of the structure. For example, if two LOWER_LAYER statements exist, the layer from the first statement is placed above the layer from the second statement.

FREQS

Syntax:

FREQS

< Frequency lines >

END FREQS

This statement block defines a default set of analysis frequencies for the translation.

The Syntax for the frequency lines is as follows:

SWEEP <start> [stop] [step]

This is a Linear Sweep which specifies one or more analysis frequencies evenly spaced in ascending order. <start> is the beginning frequency, [stop] is the optional ending frequency and [step] is the optional interval between analysis frequencies.

STEP [frequency]

Single frequency which analyzes your circuit at only one frequency.

ABS_ENTRY <start> <stop>

An adaptive sweep which uses the Adaptive Band Synthesis (ABS) technique to perform a fine resolution analysis of a specified frequency band. <start> specifies the beginning frequency of the frequency band and <stop> specifies the end of the frequency band.

ESWEEP <start> <stop> <# points>

An exponential sweep which specifies an exponential frequency sweep from the starting frequency to the end frequency with a common ratio between the desired number of frequency points. <start> is the beginning frequency, <stop> is the ending frequency and <# points> is the number of frequency points used for the analysis. For example the values Start = 8.0, Stop = 64.0 with the # points = 4 will analyze at 8, 16, 32, and 64.

LSWEEP <start> <stop> <# points>

A linear sweep which uses multiple frequencies evenly spaced in an ascending order. <start> is the starting frequency, <stop> is the ending frequency and <# points> is how many analysis frequencies at which you wish to analyze. The frequency range is divided by the number of points to provide a constant interval between analysis frequencies. For example the values Start = 2.0, Stop = 10.0 with the # points = 5 will analyze at 2, 4, 6, 8 and 10 GHz.

ABS_FMIN NET= <parameter> <start> <stop>

Find Minimum which performs an adaptive sweep on the frequency band specified by the <start> and <stop> fields, then searches for the minimum value of the response specified by <parameter>. For example, the command `ABS_FMIN NET=S21 20.0 40.0` would find the minimum value of S21 in the frequency band 20.0 - 40.0 GHz.

`ABS_FMAX NET= <parameter> <start> <stop>`

Find Maximum which performs an adaptive sweep on the frequency band specified by the <start> and <stop> fields, then searches for the maximum value of the response specified by <parameter>. For example, the command `ABS_FMAX NET=S21 20.0 40.0` would find the maximum value of S21 in the frequency band 20.0 - 40.0 GHz.

TEST_VALID

Syntax: `TEST_VALID`

Enable the valid instance tests. This should be used if you are having problems translating your files.

IGNORE_VALID

Syntax: `IGNORE_VALID`

Disable the valid instance tests. This is the default setting.

Troubleshooting

This section gives some procedures to troubleshoot through some simple problems.

***Ebridge* menu does not appear or is corrupted**

When Agilent is started in a project directory, the *Ebridge* menu does not automatically appear. Also if other programs use the command window user menu feature, they may corrupt the *Ebridge* menu.

If at any time you need to reset this menu, open the command line window by selecting *Option* \Rightarrow *Command line* from the main menu of ADS. The command line dialog box appears on your display. In the command line text entry box, enter the following:

```
reload_ebridge();
```

This will reload the software and reset the menu again. Note that this removes any entries in the command window user menu. If you wish to load the *Ebridge* menu along with another user menu, enter:

```
reload_ebridge_nc();
```

in the command window. This will add the ***ebridge*** menu selections to any existing selections. These procedures can be used in a start up macro as well. If many users start up Libra or Touchstone within a project, we recommend that a system wide macro be implemented to include the `reload_ebridge();` command.

Change in Agilent Configuration File

Another problem that could be preventing the ***ebridge*** menu from appearing is the setting in the “de.cfg” file added during the configuration for the Agilent Interface. If you are running Sonnet on a PC, follow the instructions in “Setting up the Agilent Environment,” page 63 in the **Windows Installation Manual** substituting “USER_AEL” for “SITE_AEL”. If you are running Sonnet on a UNIX workstation, follow the instructions in “Setting up the Agilent Environment,” page 62 in the **UNIX & Linux Installation Manual** substituting “USER_AEL” for “SITE_AEL.”

Error in Processing the Layout

The Agilent interface issues several different warning messages to alert the user to certain assumptions being made due to unavailable data. These are descriptive and can be handled by simply correcting the problem. However if you ever receive an Agilent system error when performing a translation, you probably have a configuration problem.

Error While Creating Hierarchy

When the **Selected** \Rightarrow **Geo** option is used, the Agilent interface creates two hierarchical references by instantiating both the layout and the schematic. The names chosen for these instances are automatically generated by the Agilent interface. If there is a design of the same name resident in memory, it will cause an error. This should not occur in normal use, but if it does, remove the offending designs from memory and try again. Remember to reopen your top level design to recover your original starting position.

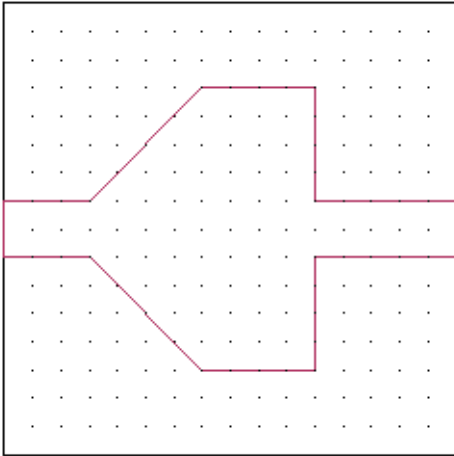
Port Instances Do Not Translate

In order for a port to be located and translated by the Agilent interface it must be directly connected to a translated polygon. If a port is connected to the circuit by a wire in the layout window it will not be translated by the Agilent interface. Make sure all ports are collocated with their connected pins prior to translation.

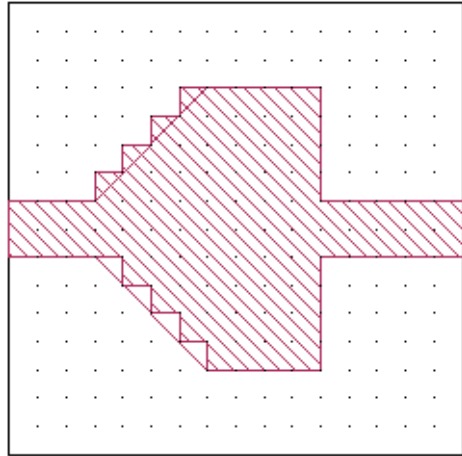
Chapter 17 Using Diagonal Fill

This chapter discusses the use of the diagonal fill option for a metal polygon.

When a polygon is first added to a circuit, the default fill type is staircase. The metal will be “filled” in using whole cells, thus a diagonal edge would resemble a staircase. An example of this is shown in the figure on page 298.



Metal “off”



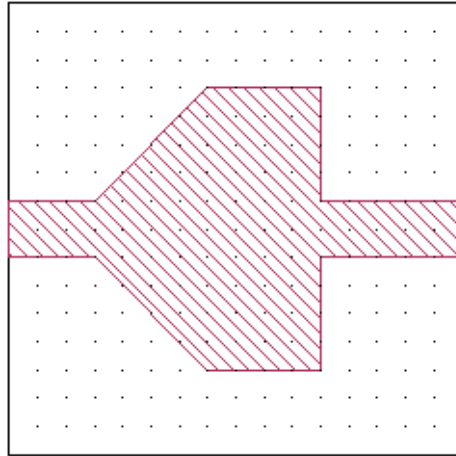
Metal “on”

An example of staircase fill for a metal polygon.

On the left is the outline of the polygon with the metal fill turned “off”. On the right is the polygon with the metal turned “on” with a staircase fill. As can be seen, the cell fill makes a staircase approximating the diagonal edge of the polygon.

Note that the error caused by such an approximation decreases as the X and Y cell sizes are decreased. Thus, it is possible to make this error arbitrarily small by choosing sufficiently small X and Y cell sizes. However, for a coarser cell size, the diagonal fill option, which uses triangular subsections on the edge, may provide a better fit.

This fill option involves additional computation time. However, depending on the circuit, using this option may allow a given level of precision to be achieved more quickly. The circuit on page 298 is shown below with diagonal fill.

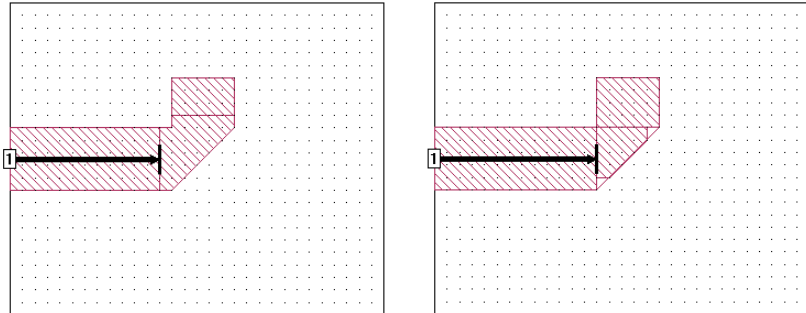


A Coupled Open-Miter with Diagonal Fill

As an example, a right angle mitered bend which is closely interacting with an open end will be analyzed. Attempts to model this discontinuity using a circuit theory based program do not include the fringing field interaction between the two discontinuities.

The circuit is contained in “[Openmite](#)” available in the Sonnet examples and is shown on the left in the figure on page 300. For details on how to use diagonal fill for a metal polygon, see “Modify - Metal Properties” in the project editor’s online help. For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button.

Notice that there are “gussets” where the miter polygon attaches to the transmission line shown on the left in the figure below. Due to the nature of the triangle subsections used in diagonal fill, acute angles (less than 90 degrees) cannot be modeled. *Em* automatically cuts acute angles off, as illustrated on the right. Thus, the tabs on the miter polygon.



On the left, “openmite.son,” with a closely interacting mitered bend and open end discontinuity. If the mitered region is captured as a simple triangle, without tabs, as shown on the right, the vertices with acute angles are cut off.

This analysis required more time due to the use of diagonal fill to model the mitre accurately. Once included, however, the *em* analysis time is relatively insensitive to the amount of diagonal fill.

Analyzing the file yields the following results:

```
10.0000000 1.000000  -150.6
```

Circuits with diagonal edges may benefit from using diagonal fill. The edges of such structures are frequently much better approximated with the diagonal edges allowed by diagonal fill. Include it only on polygons with diagonal edges that carry significant current. See references [46] and [72] in the Sonnet References for a detailed description of diagonal fill (triangle subsections).

Chapter 18 Vias and 3-D Structures

Introduction

Em can handle full 3-D current as well as 2.5 D structures. The third (Z) dimension of current is handled by a special kind of subsection called a via.

The term “via” commonly refers to a connection from metal on the substrate surface to the groundplate beneath the substrate. However, as used in Sonnet, a via can be used to connect metalization between any substrate or dielectric layer, not just bottom layer to ground. Thus, *em*’s vias can be used in modeling airbridges, spiral inductors, wire bonds and probes as well as the standard ground via.

Restrictions on Vias

Em's vias use a uniform distribution of current along their height and thus are not intended to be used to model resonant length vertical structures. The height of the via should be a small fraction of a wavelength. The via height is the same as the thickness of the substrate (or dielectric layer) it penetrates.

If a microstrip substrate is a significant fraction of a wavelength thick, overmoding also becomes a major problem. If vias are used to form, for example, a septum, or an interior wall, you may need to model it with multiple layers to achieve an accurate analysis.

Creating the Vias

Vias may be added to your circuit in a number of ways. Both the direction and type of via is chosen in the *Tools* \Rightarrow *Add Via* menu. The default may be set to go up one level, down one level, or down to ground, through multiple layers if necessary. Vias may be edge vias, rectangular vias, via polygons or circular vias.

Via Direction

A via which goes up one level extends from the level of metalization to which it is added up through the dielectric layer to the level of metalization above it. For example, if you draw a via on level 2 with the direction set to Up One Level (*Tools* \Rightarrow *Add Via* \Rightarrow *Up One Level*), the via extends from level 2 up through the dielectric to level 1 of your circuit.

A via which goes down one level extends from the level of metalization to which it is added down through the dielectric layer to the next level of metalization. For example, you draw a via on level 0 with the direction set to Down One Level (*Tools* \Rightarrow *Add Vias* \Rightarrow *Down One Level*). The via extends from level 0 through the dielectric layer to metalization level 1.

A via which goes down to ground extends from the level of metalization to which it is added through all intervening levels until it reaches the ground (bottom) of the enclosing box. For example, a circuit with 5 dielectric layers has four metalization

levels (3 - 0). If you add a via on metalization level 1 with the direction set to down to ground (*Tools* \Rightarrow *Add Vias* \Rightarrow *Down to Ground*) the via extends from level 1 down through the intervening dielectric layers and metalization levels to the bottom of the enclosing box which is ground. This via is drawn on levels 1, 2, 3 and ground. The ground level is completely metalized, but the via is drawn here to represent the connection from upper levels.

Via Types

There are basically two types of vias: edge and polygon. The via polygons can be rectangles, circles or any arbitrary shape.

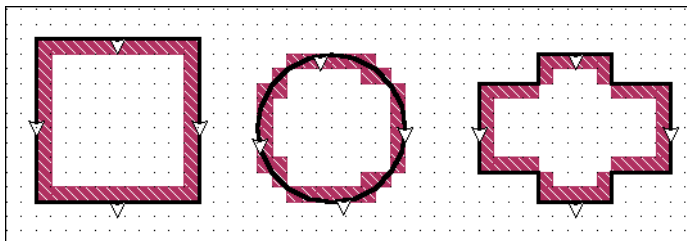
Via polygons are vias which are a separate object from the metalization polygons. They allow you to add vias of any shape whose properties are easy to modify. The via polygon may also be of a different metal type than any metalization polygons to which it is adjacent.

Edge polygons are vias that are attached to the edge of a metal polygon. The via extends for the length of the polygon edge and is one cell wide. The metal type used for an edge via is always the same as the polygon to which the edge via is attached. If the metal type of the polygon is modified, then the metal type of the via is also changed.

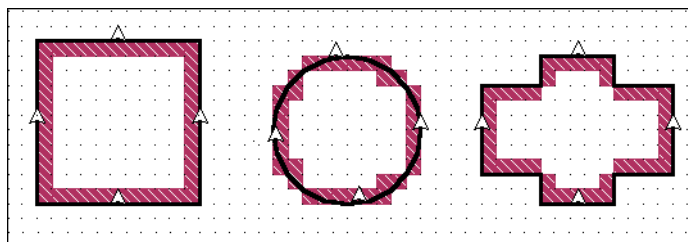
Via polygons are added to your circuit in much the same way that metal polygons are added to your circuit. You may add vias in the preset shapes of rectangles or circles or draw an arbitrary polygon by placing each vertex in its desired location. Once the shape is complete, the via polygon is drawn in your circuit. The via consists of a one cell wide wall of via subsections and is hollow in the middle. Via polygons are hollow in the middle since all current moves on the surface of the via and modeling metal which is not used wastes processing resources. If you wish to have metal in the center of a via polygon on a metalization level, you may add a metal polygon on that level.

Examples of via polygons are shown below. The shape drawn by the user appears in black. The actual via metal is shown by the fill pattern which is the video reverse of the metal pattern. Since current travels on the surface of a via, the middle of the via is hollow, filled with the dielectric material of the dielectric layer that the via traverses.

Tops of Vias



Bottoms of Vias

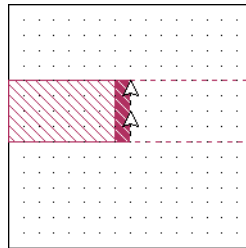


Rectangle Via

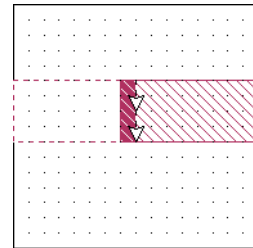
Circle Via

Polygon Via

The example shown below uses an edge via to connect two polygons on adjacent levels. The “up” via symbol indicates that the via connects this level to the next level above. The vias on the upper level are shown with a “down” via symbol which is a “down” triangle. Via symbols were automatically created on the destination level when the via was added to the source level.



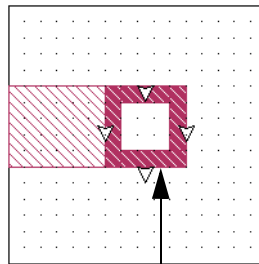
Lower level - up triangles



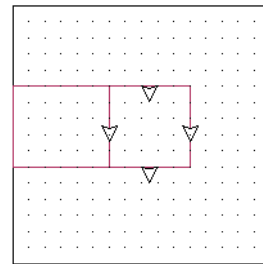
Upper level - down triangles

Via Posts

With the metalization turned on (default setting), by setting *View* \Rightarrow *Cell Fill* to “On”, the via subsections, called “via posts”, are also displayed in reverse video as shown below.



Via posts shown in reverse video with cell fill on.



The via is indicated by only triangles and an outline when cell fill is off.

When *em* subsections the circuit, it subsections each edge via or via polygon into subsectional vias called “via posts”. Each via post is a rectangular cylinder of current, extending between the present level to the next level above or below

(depending on the direction of the via). A via post has a horizontal cross-sectional area equal to one cell and a height equal to the thickness of the dielectric layer. If you change the cell size, then the via is resubsectioned into via posts with the new cell size. The project editor places enough via posts to cover the entire length of the polygon edge for an edge via and the complete perimeter of a via polygon. A via with the via posts detailed is illustrated on page 307.



To view vias as they are being captured, it is convenient to be able to change the viewed level in the project editor quickly. To do so, just type Ctrl-U to go up one level, towards the box top, or Ctrl-D to go down one level. You may also click on the *Up One Level* or *Down One Level* button on the tool bar in the project editor. Most keyboards also support the up and down arrow keys.

If you want a level to be displayed as a “ghost” outline whenever you are not on that level, make the level visible in the Levels dialog box which appears in response to selecting *View ⇒ Metalization Levels*. Then you can see how different levels of metalization line up. You may also use the Levels dialog box to turn off the visibility of any given level. By default, the project editor starts with all levels visible.

Adding a Via to Ground

A via to ground can be added from any metalization level. To add a via to ground, go to the level from which you wish the via to extend downwards and perform the following:

- 1 **Select Tools ⇒ Add Via ⇒ Down to Ground from the project editor menu.**

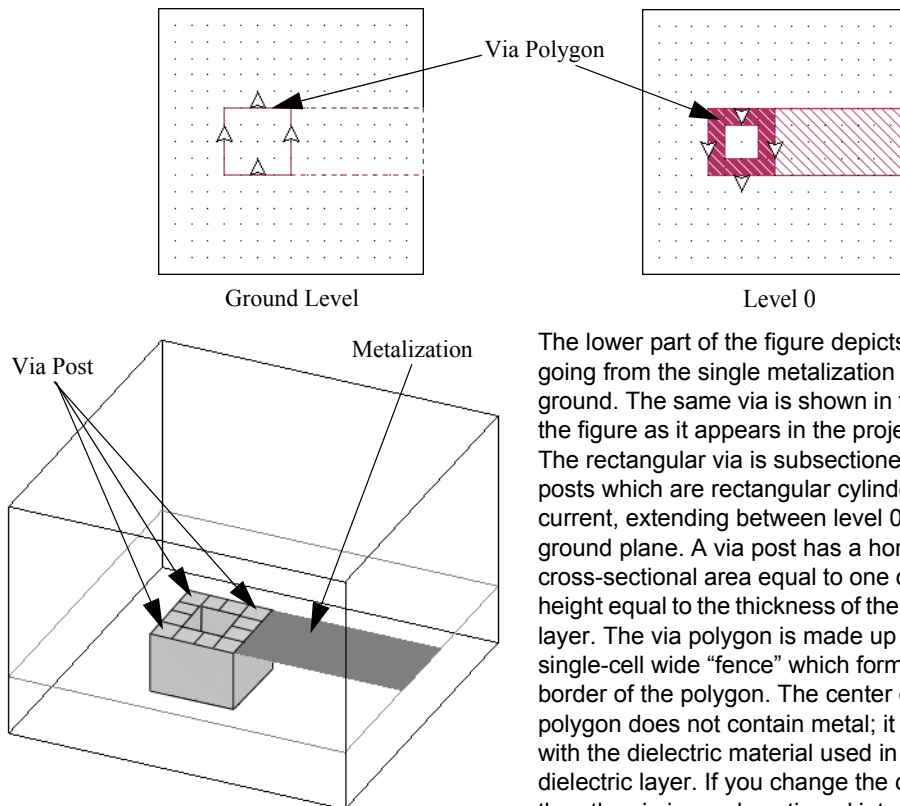
Any vias subsequently added to your circuit will extend from the level to which they are added down through all intervening levels to ground.

- 2 **Select Tools ⇒ Add Via ⇒ <Via Type> to add the desired type of via.**

The command places you in an add via mode; the type of via is dependent on the command you selected. Draw the desired via. The via you input is drawn on the present level and vias are drawn on each level below up to and including the ground plane at the bottom of the box. If you wanted to add an edge via, you would

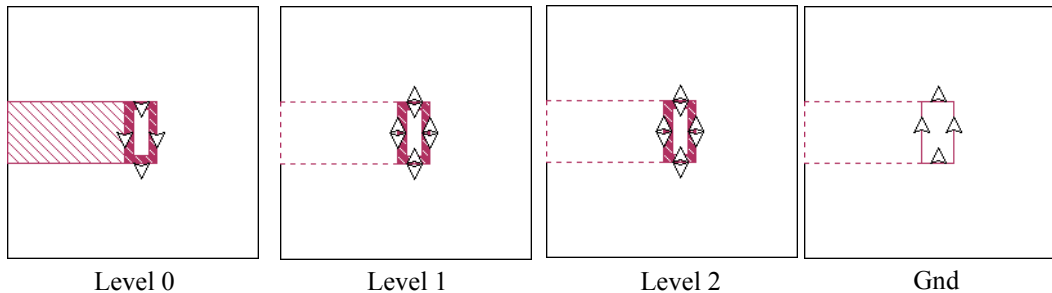
first have had to draw a metal polygon to which to attach the edge via. An example of a via polygon going from level 0 to ground in a two level circuit is pictured below.

The via polygon metalization is shown on Level 0. Note that the arrows are pointing down, indicating the direction of the via. The center of the via does not contain metalization but is filled with the dielectric of the dielectric layer. The ground level is completely metallized; the outline of the via polygon is drawn on ground level simply as a reference with up arrows indicating that there is a via polygon in the level above.



The lower part of the figure depicts a via going from the single metalization level to ground. The same via is shown in the top of the figure as it appears in the project editor. The rectangular via is subsectioned into via posts which are rectangular cylinders of current, extending between level 0 and the ground plane. A via post has a horizontal cross-sectional area equal to one cell and a height equal to the thickness of the dielectric layer. The via polygon is made up of a single-cell wide “fence” which forms the border of the polygon. The center of the via polygon does not contain metal; it is filled with the dielectric material used in the dielectric layer. If you change the cell size, then the via is resubsectioned into via posts with the new cell size.

If the via to ground is added when there are multiple intervening metal levels between the present level and ground, the via polygon can be seen on each level. The intervening levels have via arrows pointing in both directions to indicate that the via extends both upward and downward. Below is shown a rectangular via polygon extending from metal level 0 to ground in a three level circuit.

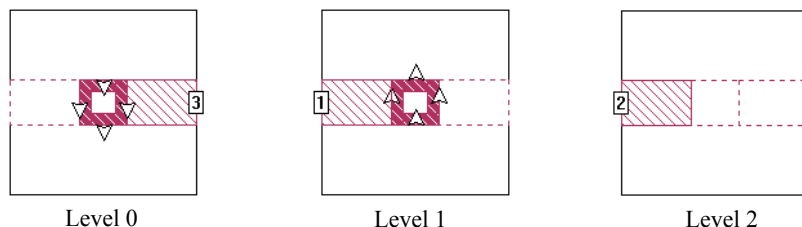


The via shown above extends from level 0, the highest metal level in the circuit down to the ground level. The via arrows on the rectangular via on level 0 point only in the downward direction. The via polygon appears on levels 1 and 2 in the same position but with via arrows pointing in both the upward and downward position indicating that the via extends in both directions from these levels. Only the outline of the via polygon is drawn on the ground plane to indicate its position with the via arrows pointing upward indicating that the via extends upward. Since the complete ground plane is metalization, the via polygon is drawn simply as a reference for the user. Note that the center of the via polygon is not metalization but is a rectangular cylinder of dielectric material.

Multi-layer Vias

It is possible to have a via which traverses more than one dielectric layer. You may insert a via in your circuit originating on any level and ending on any level. The via is automatically drawn on each level it traverses. To create a multi-layer via,

first create a via in your circuit, then modify its properties. For example, you have a four level circuit with an existing via polygon which extends from level 1 to level 0 as shown below.



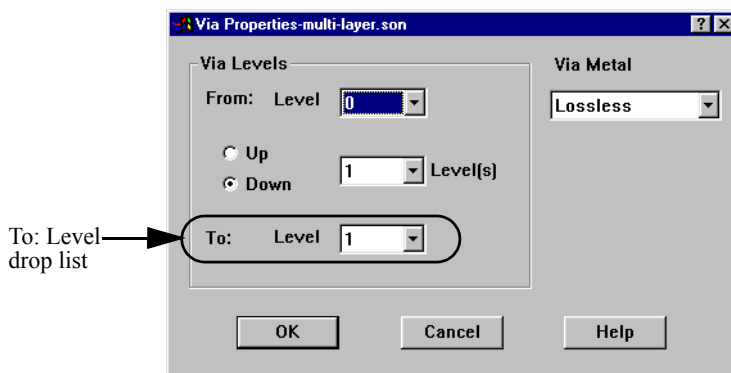
If you want to modify this via such that it extends from level 0 to level 2, you would do the following:

- 1 **Right-click on the via polygon on any level on which it appears.**

A pop-up menu appears on your display.

- 2 **Select Properties from the pop-up menu.**

The Via Properties dialog box appears on your display.

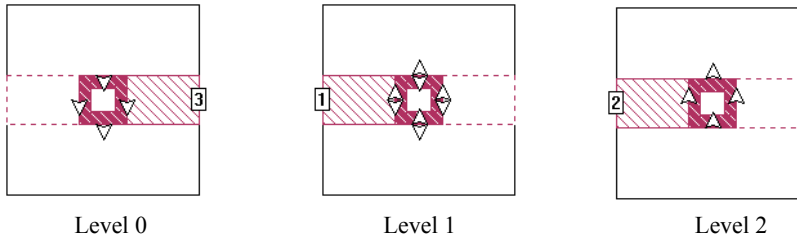


- 3 **Select “2” from the To: Level drop list.**

The via originates on level 0 which you do not wish to change. Selecting 2 from the drop list changes the via so that it goes down to level 2 instead of level 1. Notice that the number of levels is updated from 1 to 2.

4 Click on OK to apply the changes and close the dialog box.

Your circuit is redrawn so that it appears as shown below.



Note that the via polygon which appears on level 1 now has both up and down arrows indicating that the via extends in both direction. The via polygon now appears on level 2, the new endpoint of the modified via.

Deleting Vias

Vias may be deleted on any metalization level on which they appear, even if you are in the middle of the via between the endpoint levels.

Via Polygons

To delete a via polygon, select the via polygon while in pointer mode by clicking anywhere on the via that you wish to delete. Then select *Edit* \Rightarrow *Cut* from the menu or the Delete key to delete it.

Edge Vias

To delete an edge via, select the via while in pointer mode by clicking on a triangle of the via that you want deleted. Then select *Edit* \Rightarrow *Cut* from the menu bar or the Delete key to delete it. Deleting a via deletes the via posts associated with it.

You should also note that if you delete or move a polygon from which an edge via originates, the via is moved or deleted from your circuit as well.

Via Loss

The loss for the via post is determined by the metal type of via polygons and the metalization of the polygon that the via is associated with for edge vias. See “Metalization Loss,” page 74 for an explanation on how to set the metalization loss for a metal polygon.

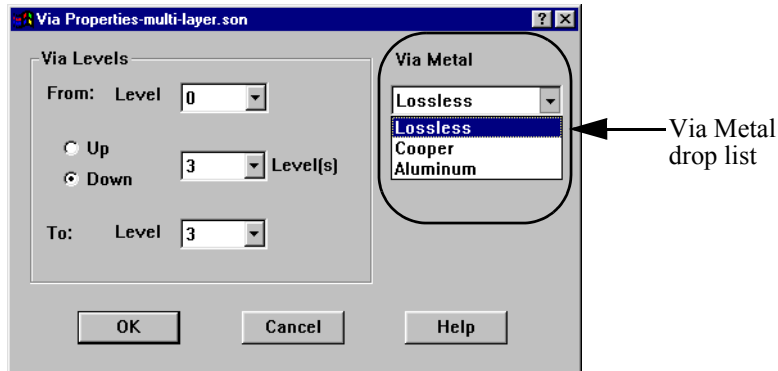
When a via polygon is created, it’s metal type is set to the default metal used for new metalization. This is controlled in the Metal Types dialog box accessed by selecting *Circuit* \Rightarrow *Metal Types* from the project editor’s menu. You may also change the metal type of the via polygon after it has been added to the circuit. You may use any metal type defined in your circuit for a via polygon.

To change the metal type of a via polygon:

- 1 Right-click on the via polygon and select Properties from the pop-up menu which appears on your display.**

The Via Properties dialog box appears on your display.

2 Select the desired metal type from the Via Metal drop list.

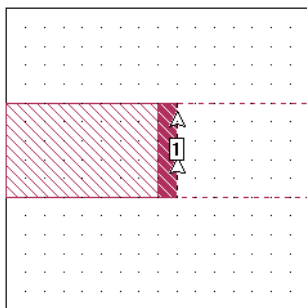


Via Ports

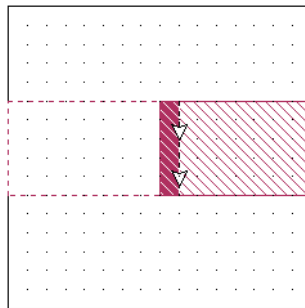
Vias may also be ports. This is a special case of the internal port described in “Ungrounded-Internal Ports,” page 110. The port is inserted between two levels. You are only allowed to add one port to a via polygon or edge via. The port is always added on the bottom level of the via and only appears on the bottom level of the via. If you observe the via on another level of your circuit, the port does not appear. Therefore, if you add a port to a via polygon on a level which is not the bottom of the via polygon, it will seem as if your action had no result. You must go to the bottom of the via to see the added port drawn in the project editor.

A via port is useful since it can use metal on one level as ground and metal on a higher level as the source.

An example of a via port on an edge via is shown below. The example [Dual_patch](#) included with your example files has an example of using a via port. For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button. Also shown at the bottom is a via port on a via polygon.

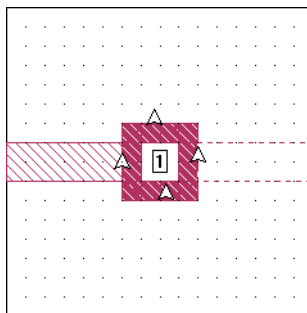


Lower level

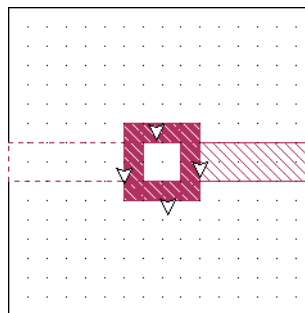


Upper level

A via port on an edge via with both levels shown.



Lower level



Upper level

A via port on a via polygon with both levels



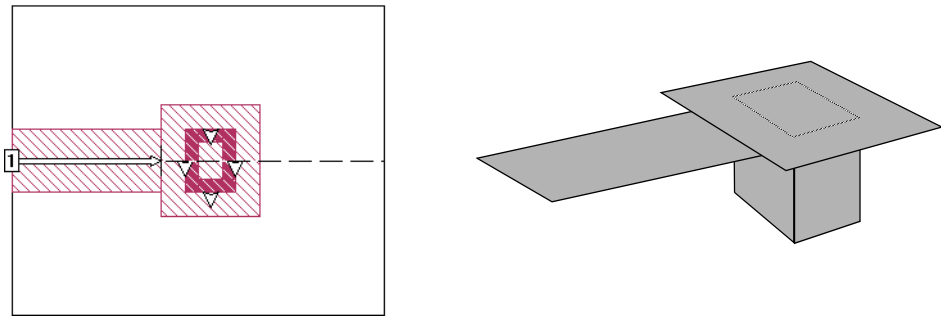
TIP

If you want to have a port from a polygon to ground, consider using an auto-grounded port. For details about autogrounded ports, see “Automatic-Grounded Ports,” page 112.

Simple Via Example

A simple via is stored in the example [Via](#) and is shown in the figure on page 314. For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button.

Note that the top end of the via, shown below, is a “pad” which is larger than the via itself. There are no restrictions on the polygons at the top of a via. *Em*’s subsectioning algorithm handles the subsectioning accurately.



A simple via to ground. On the left, as it would appear in the project editor. On the right, a view in perspective.

A Conical Via

One may simulate a conical ground via with a staircase approximation. Simply divide, say, a 100 μM GaAs substrate into four 25 μM substrates. Then put vias at appropriate places to form a step approximation to the conical via sides. For an example, see [Cvia](#) in the Sonnet examples. This circuit is a conical via to ground placed in the center of a through line, the purpose being to measure the via inductance. For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button.

The “cvia.son” file is a very detailed model of a conical via. If you are modeling a large circuit (say, an inter-stage matching network) with multiple vias, you may want to use a simpler model for faster analysis. Another approach would be to use circuit subdivision where you subdivide the circuit such that the via is simulated separately-thus providing an accurate via simulation. For more information on circuit subdivision, see Chapter 13, “Circuit Subdivision” on page 201.

Chapter 19 Thick Metal

Thick Metal Type

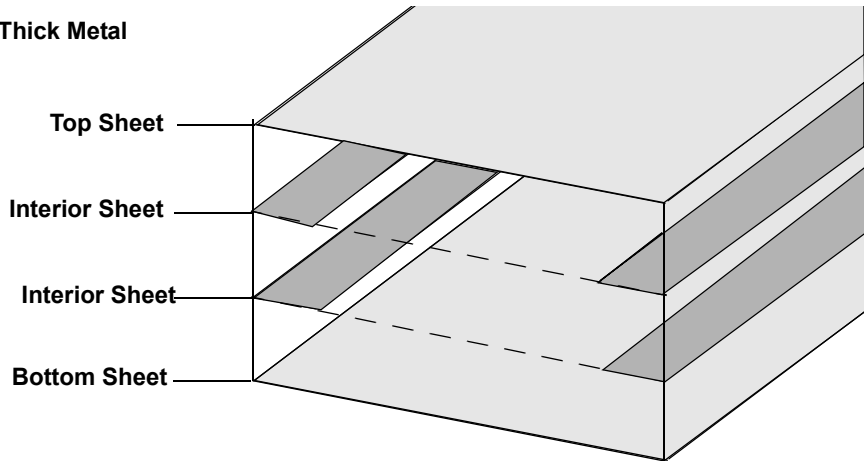
The Thick Metal metal type allows you to model physically thick metal. All other metal types are modeled as having zero thickness; only the loss is affected by the thickness. The thick metal type allows you to more accurately model the true 3D characteristics of thick conductors.

Since thick metal increases both your processing time and memory requirements, it should only be used when necessary. Metal is considered to be thick metal when its thickness is comparable to other dimensions in the circuit such as the width of a conductor or gaps between conductors.

When using thick metal, the structure is approximated by two or more infinitely thin sheets of metal. One sheet represents the top surface of the structure and a second sheet represents the bottom surface of the structure. Vias are placed

automatically around the perimeter to represent the side surfaces of the structure. If you use more than two sheets, the interior sheets only supply a ring of current, on the edge of the structure, as shown in the illustration below.

Four Sheet Thick Metal

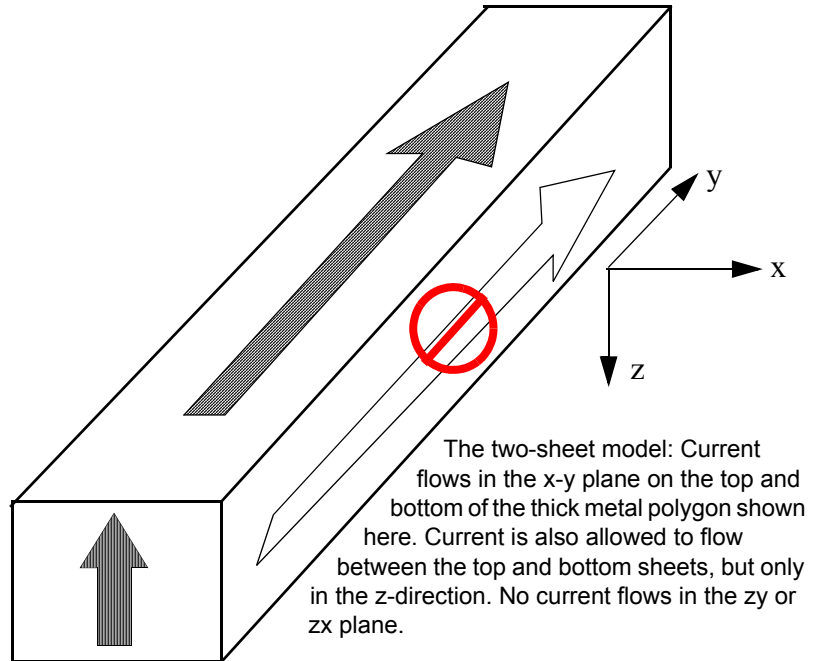


This is a cross section of thick metal modeled using four sheets; the vias are not shown for clarity. Note that the interior sheets are only a ring of metal on the edge of the thick metal structure. This saves on metalization which makes the model more efficient.

As you increase the number of sheets the accuracy of your answer is improved. However, there is a trade-off since an increased number of sheets also demands more processing time and memory use. The recommended default of 2 sheets should prove to be adequate for the majority of circuits.

If you use the Thick Metal on conductors in extremely close proximity, you should increase the number of sheets in order to account for the side to side coupling between the conductors. If the metal thickness is greater than the gap between conductors more than 2 sheets should be used. Enough sheets should be used so that the spacing between sheets is less than the gap between conductors.

This model assumes that current travels on only the top and bottom surface of the thick metal for a 2 sheet model. Current on the sides of the thick conductor can be accounted for by using 3 or more sheets.



This model strikes a reasonable balance between accuracy and inordinate processing time and memory requirements.

Creating a Thick Metal Polygon

To create a thick metal polygon in your circuit, you must first define a metal type using the Thick Metal, then apply that metal type to the polygon in your circuit. To do this, perform the following:

- 1 In the project editor, select **Circuit** \Rightarrow **Metal Types** from the main menu.

The Metal Types dialog box appears on your display.

- 2 Click on the Add button in the Metal Types dialog box.

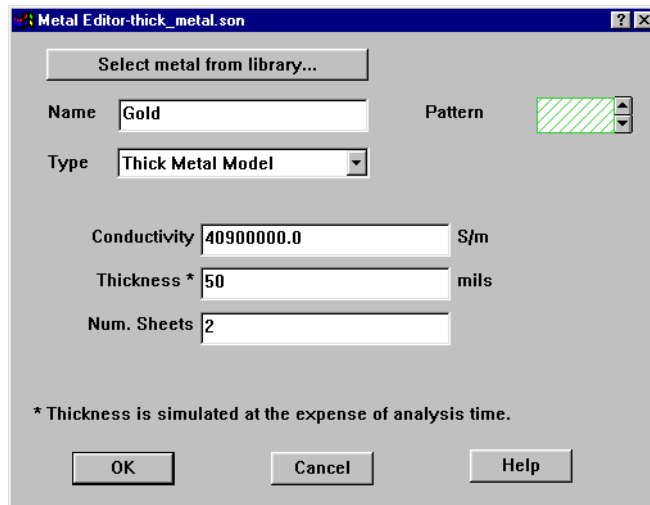
The Metal Editor dialog box appears on your display.

- 3 Select Thick Metal Model from the type drop list in the Metal Editor dialog box.

This updates the dialog box with the text entry boxes for the three parameters needed for the thick metal model: Conductivity, Thickness, and Number of Sheets.

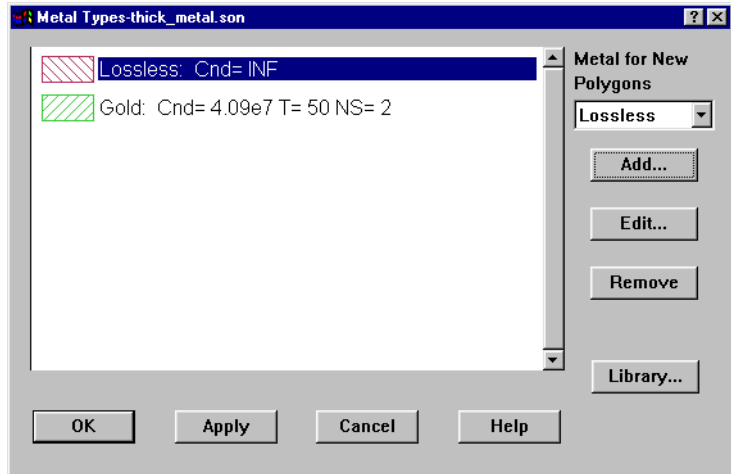
- 4 Enter the three parameters in the appropriate text entry boxes.
- 5 If you do not wish to use the default metal name, enter the desired name for the metal type in the Name text entry box.

The Metal Editor dialog box should appear similar to the picture below.



- 6 Click on the OK button to close the Metal Editor dialog box and apply the changes.

The Metal Types dialog box is updated with the new metal type.



- 7 Click on the OK button to close the Metal Types dialog box.

The thick metal is now available to use in your project.

- 8 Enter the desired polygon, then double-click on the polygon to open the Metalization Properties dialog box.
- 9 Select the thick metal model metal type from the Metal drop list in the Metalization Properties dialog box.

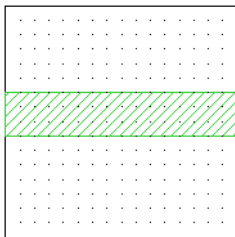
This will apply the metal type which uses thick metal to the selected polygon. The thick metal extends upwards from the level on which the polygon was drawn.

- 10 Click on the OK button to close the Metalization Properties dialog box and apply the changes.

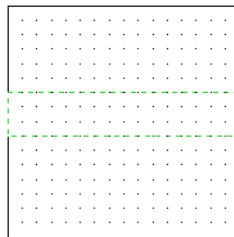
The fill pattern of the polygon changes to the fill pattern used by the thick metal. If the thick metal polygon is thicker than the dielectric layer(s) above it, the polygon also appears on metal levels above.

Viewing Thick Metal in the Project Editor

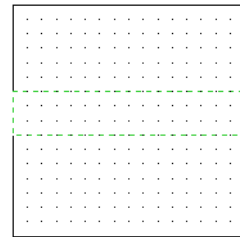
The thick metal extends upward through the dielectric layer from the level on which the polygon is drawn. If the thick metal is not as thick as the dielectric layer above it, then the polygon only appears on the lower level where it was drawn. If the thick metal is the same thickness as the dielectric layer above it appears on both the metal level where it was drawn and on the metal level above. Examples of both instances are shown below.



Level 2

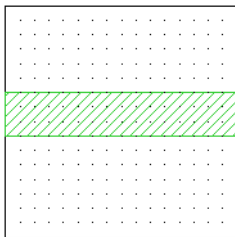


Level 1

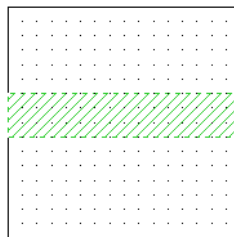


Level 0

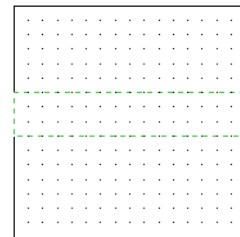
A 3 mil thick metal polygon is drawn on level 2 below a 5 mil thick dielectric layer. The polygon is visible on level 2 where it was drawn, but only the outline is visible on level 1 above since the thick metal does not pierce the dielectric.



Level 2



Level 1

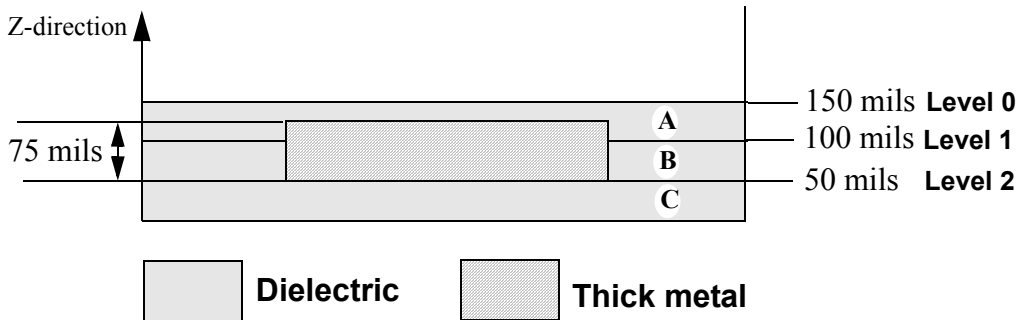


Level 0

A 5 mil thick metal polygon is drawn on level 2 below a 5 mil thick dielectric layer. The polygon is visible on level 2 where it was drawn, and also on level 1 above since it is the same thickness as the dielectric layer. Note that on level 1, the border of the polygon is drawn with a dashed line to indicate that the origin of this polygon is not on this level.

If the thick metal is thicker than the dielectric above, but not thick enough to pierce the next dielectric layer, the polygon appears on the level where it was drawn and on the metal level above. However, note that the top of the thick metal does not appear in the project editor because it is embedded in a dielectric layer. You will be able to view the top sheet of metal in the current density viewer which is discussed later in the chapter.

A side view of a circuit with three 50 mil dielectric layers (A, B, and C) and a 75 mil thick metal polygon on level 2 is shown below. Note that the top of the thick metal only extends halfway through dielectric layer. The top of this thick metal is not visible in the project editor.



Another important thing to note about the modeling of a thick metal which ends in the interior of a dielectric layer is that even though you model the thick metal with 2 sheets, the software actually uses three sheets. The bottom sheet is on level 2 where the polygon originates. The top sheet is in the interior of dielectric layer A. The third sheet is on metal level 1. Whenever a thick metal polygon traverses a metal level, a sheet is added on that level. This adds additional computational time and should be kept in mind when using thick metals which encompass more than one dielectric layer.

If Single Level select is enabled, then you may only select the thick metal polygon on the level where it was drawn.

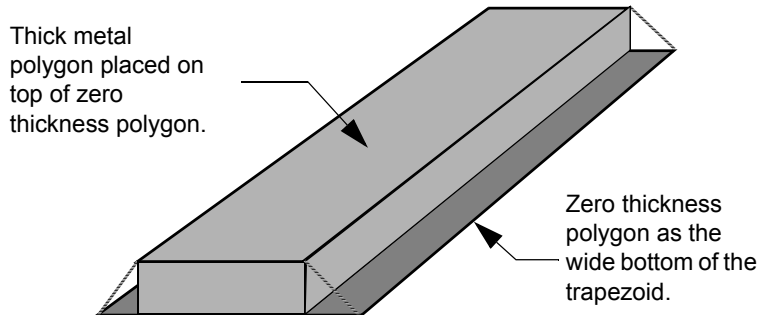
Thick metal polygons are connected to thin metal polygons by drawing the thin metal polygon on the same level on which the thick metal was drawn and placing the thin metal polygon adjacent to or overlapping the thick metal. This connects the two structures electrically.

Restrictions with Thick Metal Polygons

If you are using the thick metal model with more than two sheets of metal, be aware that analyzing a thick metal of 3 or more sheets at low frequencies may introduce error into the DC loss. To avoid this problem, use only 2 sheets for your thick metal when analyzing at very low frequencies.

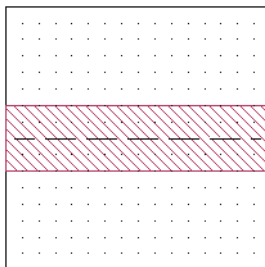
Modeling an Arbitrary Cross-Section

In this section, we use a combination of thick metal and Normal (zero thickness) metal to approximate thick metal lines where the vertical cross-section has an arbitrary geometry. To demonstrate this capability, we use a simple trapezoidal geometry, the cross section shown in the figure below.



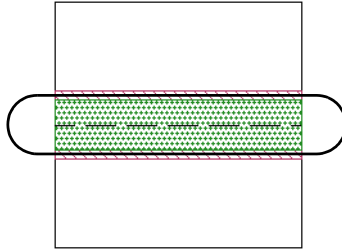
A trapezoidal cross-section transmission line viewed in perspective. If the line has no current going around the edge, it can be modeled, as shown, as two infinitely thin sheets of current, one at the top and the other at the bottom of the actual metal.

To create the thick metal trapezoid, set up the dielectrics so that there is one layer of dielectric with the same thickness as the thick metal. Then, place a polygon representing the wider bottom side of the thick metal on the bottom side of that dielectric layer. This polygon should use the Normal model for the metal type, which is modeled as a zero-thickness metal. To get the proper loss, you should set the thickness of this metal type to one half the total thickness of the metal. This compensates for the fact that current is now flowing through two conductors, instead of the usual single conductor.

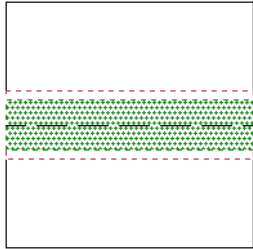


The wide bottom of the trapezoidal line is made up of a polygon using the Normal model for the metal type. This is a zero-thickness metal.

Then place a polygon representing the top side of the thick metal on the bottom side of that dielectric layer using the Thick Metal metal type. Make this polygon as thick as the dielectric layer.



Thick metal polygon on level 1 where it is drawn and placed on top of the wider zero-thickness polygon



The same polygon shown on level 0. Since the thick polygon is the same thickness as the dielectric layer, the metal also appears on this level. Only the outline of the zero-thickness metal is shown on this level.

Next, place any desired ports on the thick metal polygon, not on the thin metal polygon. Since the thick metal polygon is placed on top of the zero-thickness polygon, the two are connected electrically and the port is across both polygons.

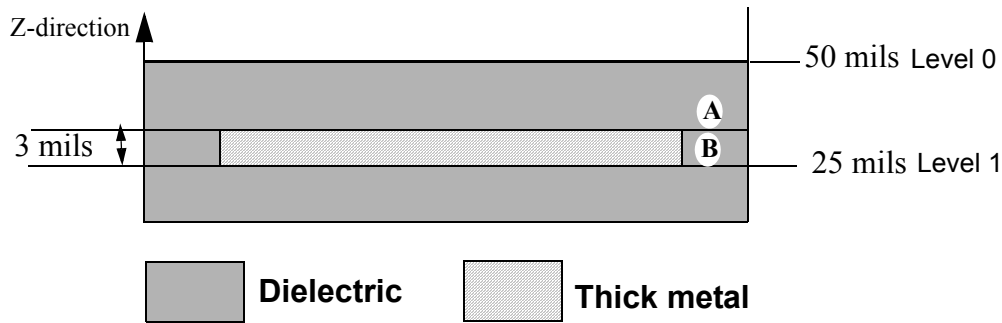
A circuit implementing the above transmission line is stored in [Thkthru](#) and an example of a thick step junction is stored in a project called [Thkstep](#). Copies of these projects can be obtained from the Sonnet examples. For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button.

Thick Metal in the Current Density Viewer

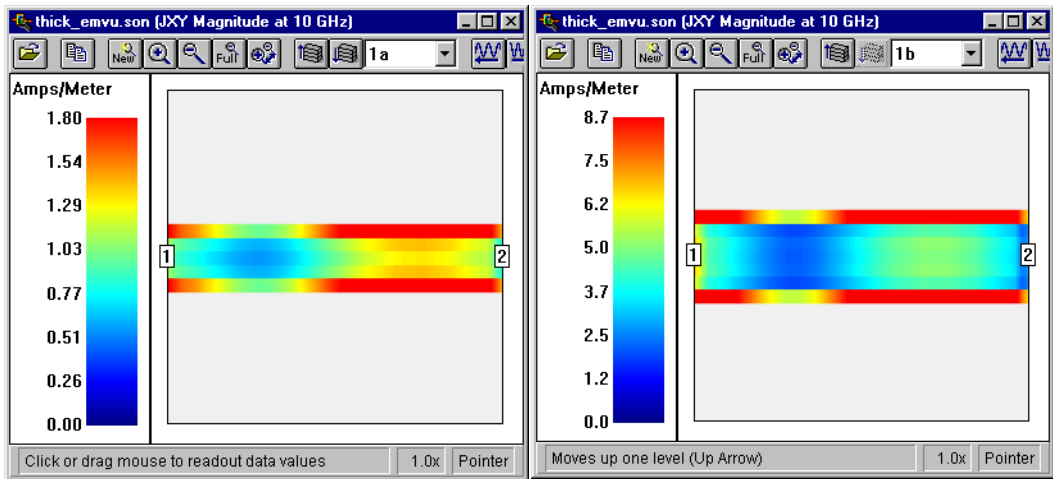
The current density viewer allows you to view the current density distribution in your circuit. When you select the Compute Current Density option in the Analysis Setup dialog box in the project editor, *em* calculates current density data for all the metal levels in your circuit. When you have thick metal in your circuit which ends in the interior of a dielectric layer, then the current density viewer creates "sublevels" of metal in order to display all the current density data.

For instance, you have a circuit with 3 mil thick metal using the default 2 sheets placed on metal level #1 below a 25 mil dielectric layer as pictured below. The top of the thick metal structure is placed in the interior of the dielectric layer. The

current density viewer displays levels 1b and 1a, where 1b is the metal level on which the thick metal was drawn and 1a is the top of the thick metal structure embedded in the dielectric layer.



Below are shown the views of level 1a and 1b in the current density viewer. Note that 1a is the top of the thick metal structure and is not visible in the project editor. 1b is the bottom where the polygon was drawn and is visible in the project editor.



The current density viewer creates as many “sublevels” as are needed. A thick metal which is defined as having 4 sheets placed on level 2 would appear in the current density viewer as 2a, 2b, 2c and 2d with 2a being the top of the thick metal structure and 2d being the bottom drawn on level 2.

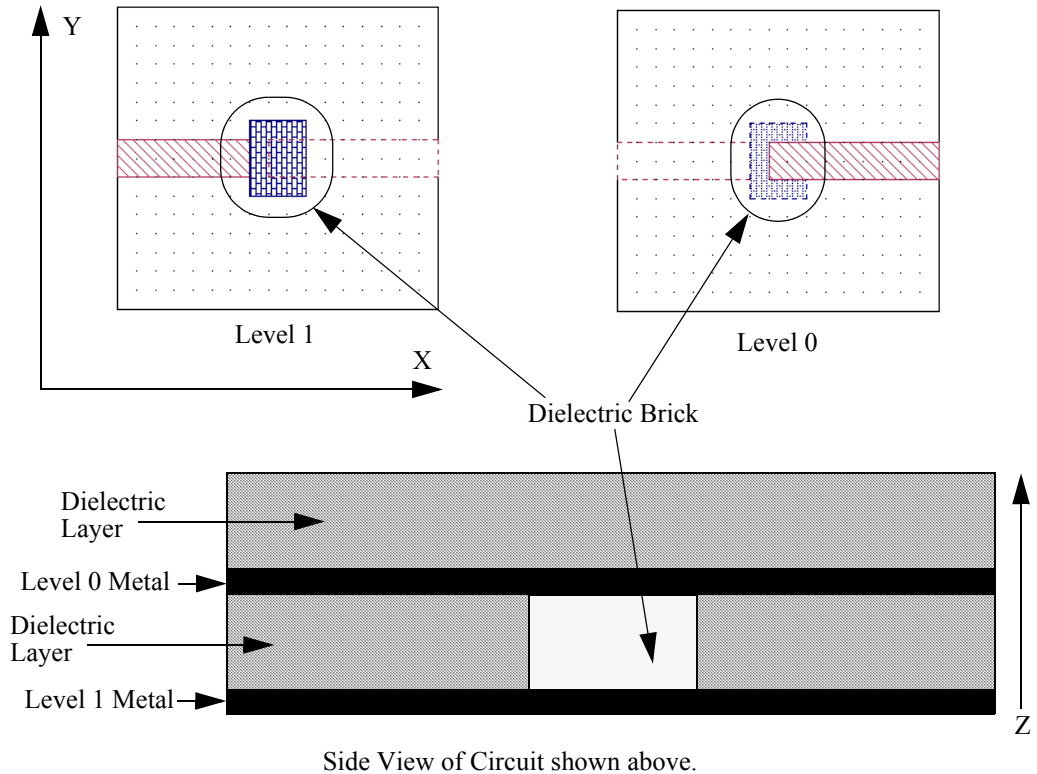
Chapter 20 Dielectric Bricks

Although *em* is primarily a planar electromagnetic simulator, it also has the capability to add “dielectric brick” material anywhere in your circuit. A dielectric brick is a solid volume of dielectric material embedded within a circuit layer. See the illustration below. Dielectric bricks can be made from any dielectric material (including air) and can be placed in circuit layers made from any other dielectric material (including air). For example, dielectric bricks can be used to simulate structures such as an embedded capacitor in an “air” circuit layer, or an “air hole” in a dielectric substrate circuit layer.



WARNING

Misuse of dielectric bricks can lead to significantly inaccurate results. It is highly recommended that you read this entire chapter before attempting to use dielectric bricks.



All realizable values for the dielectric constant, loss tangent and bulk conductivity can be used. Furthermore, it is possible to set these parameters independently in each dimension to create anisotropic dielectric bricks.

Em is appropriate for simple structures using very localized dielectric bricks; however, when your design requires large areas of brick material, you may need a full 3-D electromagnetic analysis tool.

You should also be aware that the use of dielectric bricks can dramatically increase the memory requirements, and thus the simulation time, for your circuit. Bricks should only be used where strictly necessary for the accuracy of your simulation.

Care should be taken when using dielectric bricks, since improper modeling of your dielectric brick can yield highly inaccurate data. We recommend that you run a convergence test by doubling (or halving) the number of your Z-partitions, re-analyzing your circuit and comparing the two results to ensure that you are using a sufficient number of Z-partitions. For more information on Z-partitioning, see "Z-Partitioning" on page 339.

Applications of Dielectric Bricks

The use of dielectric bricks is appropriate for applications where the effects of dielectric discontinuities or anisotropic dielectric materials are important. Examples of such applications include dielectric resonators, dielectric overlays, airbridges, microstrip-to-stripline transitions, dielectric bridges and crossovers, microslab transmission lines, capacitors and module walls.

Guidelines for Using Dielectric Bricks

Subsectioning Dielectric Bricks

A dielectric brick simulates a volume of dielectric material. Because a brick simulates a volume, it must be subsectioned in the X, Y and Z dimensions. The more subsections (finer resolution) used in each dimension, the more accurate the analysis.

X/Y subsectioning of dielectric bricks is identical to X/Y subsectioning of metal polygons. You can control the X/Y subsectioning of both through your choice of grid size, XMIN, YMIN, XMAX, YMAX and subsections-per-wavelength. See Chapter 4, "Subsectioning," for details.

Z subsectioning of dielectric bricks is controlled by the “number of Z-partitions” parameter. This parameter specifies the number of Z subsections for all dielectric bricks on a particular dielectric layer. See the “Z Partitions dialog box” topic in the project editor’s online help for information on setting this parameter.

Using Vias Inside a Dielectric Brick

Vias through dielectric bricks are treated the same as vias through the standard dielectric layers. Note that via ports inside dielectric bricks are not allowed.

Air Dielectric Bricks

Dielectric bricks can be made of any dielectric material and can be placed in any circuit layer. This allows, for instance, “alumina” bricks to be created in an “air” circuit layer. However, it is also possible to reverse this scenario. Dielectric bricks made of “air” can also be created in alumina circuit layers. This is an important consideration to remember. Depending upon the circuit geometry for a given application, this ability to reverse the dielectric characteristics may simplify the circuit and make it faster to analyze.

Limitations of Dielectric Bricks

Diagonal Fill

Diagonal fill is not allowed for dielectric bricks. All dielectric bricks must use “staircase fill”. Thus, dielectric bricks with curved or rounded edges must be stairstep approximated. Note that the error caused by such an approximation decreases as the X and Y cell sizes are decreased. Thus, it is possible to make this error arbitrarily small by choosing sufficiently small X and Y cell sizes.

Antennas and Radiation

The far field viewer does not support dielectric bricks. Circuits containing dielectric bricks can be analyzed with the far field viewer, but the radiation effects of the dielectric bricks are not accounted for in the analysis.

The Agilent Translator, *ebridge*

The Agilent translator does not create dielectric bricks.

Dielectric Brick Concepts

Creating a Dielectric Brick

To create a dielectric brick in the project editor, do the following:

- 1 Move to the circuit level where the base of the dielectric brick is to be located.**

The dielectric brick that is created will rest on this circuit level, and will extend upward to the next level. Dielectric bricks can be placed on any level, including the ground plane. If a brick is placed on the highest circuit level (level 0), it will extend up to the top cover of the metal box.

- 2 Create a base polygon which defines the cross-section of the brick.**

This is done by selecting either *Tools ⇒ Add Dielectric Brick ⇒ Draw Rectangle* or *Tools ⇒ Add Dielectric Brick ⇒ Draw Polygon* from the project editor's main menu. The first option allows the vertices of arbitrarily shaped base polygons to be entered on a point by point basis. This option is used to create dielectric bricks with any cross-sectional shape. However, if the cross-section is rectangular in shape, it is often quicker to create dielectric bricks using the second option.

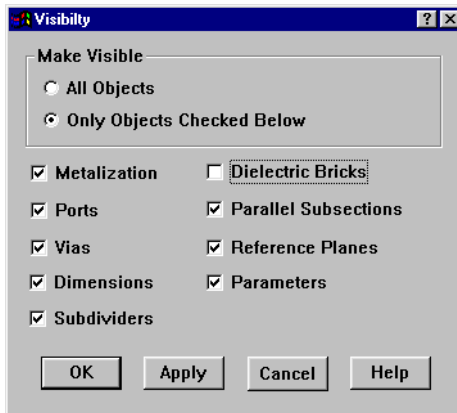
Viewing Dielectric Bricks

Once a dielectric brick has been created in the project editor, it is possible to “see” the brick from both the circuit layer where the base of the brick is located and the circuit layer where the top of the brick is located. On both levels, you will see a polygon which defines the cross-sectional shape of the dielectric brick. The brightness of the polygon, however, will vary. When you are on the top level, you will see a “dim” polygon; on the base level you will see a “bright” polygon.

Only the outline of the dielectric brick is visible from levels other than the origination and termination of the dielectric brick.

Note that while it is possible to “see” a brick from two different circuit levels, “selecting” a brick, for cutting, copying, moving, changing attributes, etc., can only be done from the circuit level where the base of the brick is located if you are in Single layer edit mode. The polygon can be selected on either level if you are in multilayer select.

Finally, it is possible to turn the display of dielectric bricks “on” or “off” in the project editor. You select *View* \Rightarrow *Object Visibility* from the main menu of the project editor which opens the Object Visibility dialog box shown below.



Click on the *Only Objects Checked Below* radio button to enable the object choices, then click on the *Dielectric Bricks* checkbox to turn off the display of the bricks.

This will make any bricks present in the circuit invisible and unselectable, but does not remove them from the circuit. The dielectric bricks can be turned back “on” by once again selecting *View* \Rightarrow *Object Visibility* and clicking the *Dielectric Bricks* checkbox or the *All Objects* radio button.

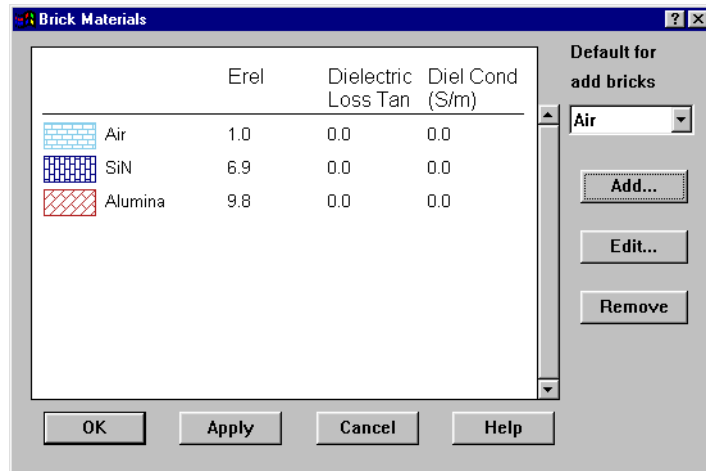
Occasionally, when a circuit contains many layers, with overlapping metal polygons and dielectric bricks, it may be somewhat difficult to distinguish the metal polygons and dielectric bricks from one another. The ability to turn dielectric bricks “off” usually makes it easier to view such circuits.

Defining Dielectric Brick Materials

Just as it is possible to define a variety of metal types, each with different properties, it is also possible to define a variety of dielectric brick materials, each with different values for the dielectric constant, loss tangent, and bulk conductivity.

To define a new dielectric brick material, or to modify the characteristics of an existing material, you use the Brick Materials dialog box, which is opened by selecting *Circuit* \Rightarrow *Brick Materials* from the main menu of the project editor.

The Brick Materials dialog box, shown on page 337, shows all the dielectric brick materials previously defined, the color/fill pattern assigned to each brick material, and whether the material is isotropic or anisotropic. To modify the settings for a particular dielectric brick material, edit that materials text entry boxes. Note that for anisotropic materials all the parameters do not fit in the dialog box simultaneously, so that it is necessary to use the scroll bars to access all settings.



If the brick type is isotropic only one set of parameters, X, will be set. Conversely, if the brick material is set to anisotropic, each parameter is defined separately for the X, Y, and Z dimensions. If you wish to make a brick material anisotropic, click on the *Ani* checkbox.

The “default” material used when new dielectric bricks are created can also be set in the Brick Materials dialog box. Select a brick type from the *Default for add bricks* drop list. Once the default material has been set, all bricks created thereafter will be made of that material.

Changing Brick Materials

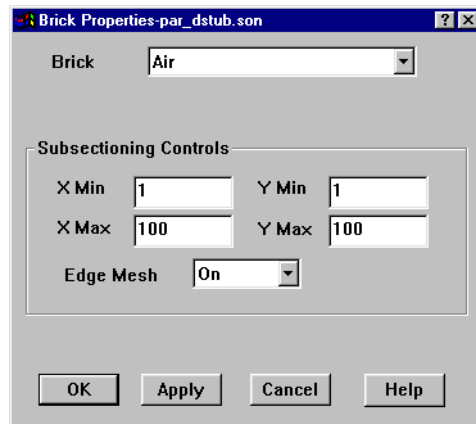
The material type for bricks that already exist in a circuit can be changed by following the procedure given below:

- 1 **Select the brick(s) by clicking on it or lassoing it.**

The brick is highlighted.

- 2 **Select *Modify* ⇒ *Brick Materials* from the main menu of the project editor.**

This will open the Dielectric Brick attributes dialog box, shown below.



- 3 **Select the brick material you desire from the drop list labeled Brick.**

This drop list contains all the types defined for dielectric bricks including the default type, air.

- 4 Click on the OK button to apply your selection and close the dialog box.

Z-Partitioning

A dielectric brick simulates a volume of dielectric material. Because a brick simulates a volume, it must be subsectioned in the X, Y, and Z dimensions. The more subsections (finer resolution) used in each dimension, the more accurate the analysis.

X/Y subsectioning of dielectric bricks is identical to X/Y subsectioning of metal polygons. You can control the X/Y subsectioning of both through your choice of grid size, XMIN, YMIN, XMAX, YMAX, and subsections-per-lambda.

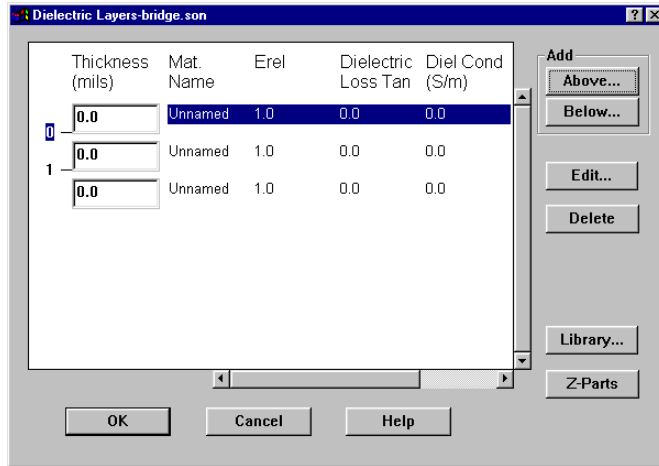
Z subsectioning of dielectric bricks is controlled by the Z Partitions dialog box which is opened when you click on the Z Parts button in the Dielectric Layers dialog box (*Circuit* \Rightarrow *Dielectric Layers*). You may enter a Z Parts value for each dielectric layer in your circuit. This parameter specifies the number of Z partitions for all dielectric bricks on a particular circuit layer.

The default for this parameter is zero so that you are forced to enter a value for this field. If you use a dielectric brick in a layer, and do not set the z-partitions, **em** reports an error and exits the analysis. You must enter a non-zero integer value for this parameter in order to run an analysis. The value of this parameter is highly dependent on your circuit design; therefore, Sonnet cannot determine a “reasonable” value. This is the reason we suggest you run a convergence test, discussed earlier in the chapter, on your circuit to determine the best value for the Z-partitioning.

To set this parameter in the project editor, do the following:

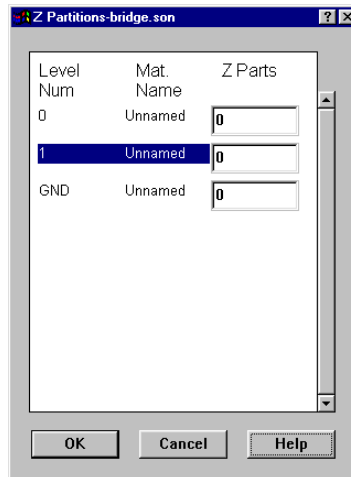
- 1 Select *Circuit* \Rightarrow *Dielectric Layers* from the project editor main menu.

The Dielectric Layers dialog box, shown below, is displayed.



- 2 Click on the Z-Parts button in the Dielectric Layers dialog box.

The Z Partitions dialog box appears on your display.



- 3 Enter the number of z partitions to be used for each dielectric layer in the appropriate Z Parts text entry box.**

Note that changing this value for a particular layer will have absolutely no affect on the analysis if there are no bricks on the layer. If there are multiple bricks on the layer, the Z subsectioning for all of those bricks will be identical. It is not possible to apply different Z partitions to brick polygons which appear on the same layer.

Chapter 21 Antennas and Radiation

To this point, this manual has been focused on using Sonnet for the analysis of high frequency circuits and transmission structures. However, there is a large class of radiating structures for which Sonnet has proven very useful. This chapter describes how to use Sonnet to analyze 3-D planar radiating structures, such as microstrip patch arrays and microstrip discontinuities, using the “Open Waveguide Simulator” technique. The underlying assumptions of this technique are described in detail. Common modeling mistakes are also pointed out. Examples are provided to illustrate the correct use of the modeling technique.

This chapter also discusses the far field viewer, an analysis and viewing tool which calculates far field antenna patterns for arbitrary 3-D planar geometries. The far field viewer uses the current distribution data in the project as input, and creates a pattern. The pattern may be viewed as a cartesian, polar or surface plot.

Background

Since **em** is an analysis of 3-D planar circuits in a completely enclosing, shielding, rectangular box, the analysis of radiating structures is not an application which immediately comes to mind.

However, **em** can be used to simulate infinite arrays using a waveguide simulator. In this technique, as shown in on page 345, a portion of the array is placed within a waveguide. The waveguide tube is vertical, connecting the radiating patches to the termination, which is a matched load. The images formed by the waveguide walls properly model the entire infinite array scanned to a specific angle.

The waveguide simulator inspired what we now call the Open Waveguide Simulator Technique described in the next section.

Modeling Infinite Arrays

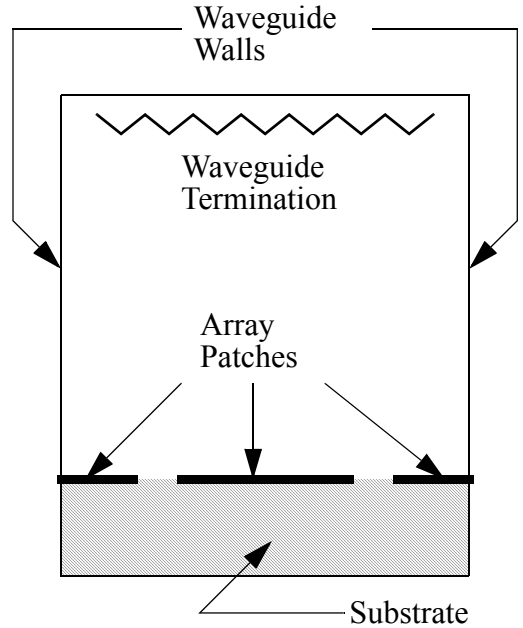
The sidewalls of the shielding box in the **em** analysis easily represent the sidewalls of the waveguide in the infinite array waveguide simulator. A side view is shown in the figure on page 345.

Providing a termination for the end of the waveguide requires a little more thought. Any waveguide mode can be perfectly terminated by making the top cover resistivity in **em** equal to the waveguide mode impedance. This can be done in the project editor automatically at all frequencies and all modes by selecting “WGLOAD” from the metals in the Top Metal drop list in the Box Settings dialog box.

$$Z_{TE} = \frac{\eta}{\sqrt{1 - \left(\frac{f_c}{f}\right)^2}} \quad f > f_c$$

$$Z_{TM} = \eta \sqrt{1 - \left(\frac{f_c}{f}\right)^2} \quad f > f_c$$

$$f_c = \frac{v_c}{2\pi} \sqrt{\left(\frac{m\pi}{A}\right)^2 + \left(\frac{n\pi}{B}\right)^2}$$



The waveguide simulator for infinite arrays inspired the technique described here. In this side view, the waveguide walls form images of the array of microstrip patches, simulating an infinite array. v_c is the velocity of light in the medium filling the waveguide.

In a phased array with the array scanned to a specific direction, a single waveguide mode is generated. The *em* software can model the waveguide simulator of that infinite array just by setting the top cover impedance to the impedance of the excited waveguide mode.

Modeling an Open Environment

If we can use a closed (i.e., terminated) waveguide to model an infinite array, we can also model radiation from a finite array; although, it must be done under certain conditions. It is important to keep in mind that, unless the analysis is

carefully prepared, these conditions are easily violated, yielding incorrect results. When the conditions are met, useful results can be obtained, as shall be demonstrated.

First Condition: *Make both of the lateral substrate dimensions greater than one or two wavelengths.*

When using the Open Waveguide Simulator, we view the sidewalls of the shielding box as forming a waveguide whose tube extends in the vertical direction, propagating energy from the antenna toward the “Termination” as shown on page 345. Radiation is then approximated as a sum of many waveguide modes. If the tube is too small, there are few, if any, propagating modes, violating the First Condition.

There is an easily made mistake when modeling radiation from small discontinuities. Discontinuities are usually small with respect to wavelength. For a discontinuity analysis, the sidewalls are usually placed one or two substrate thicknesses from the discontinuity. In this case, the substrate dimensions are unlikely to meet the First Condition. If the sidewalls form below a cut-off waveguide, there is no radiation.

Second Condition: *Make sure the sidewalls are far enough from the radiating structure that the sidewalls have no affect.*

Another way to look at this condition is to consider the image of the structure (discontinuity or antenna) created by the sidewall. Position the sidewall so that the image it forms has no significant coupling with the desired structure.

Usually two to three wavelengths from the sidewall is sufficient for discontinuities. For single patch antennas, one to three wavelengths is suggested. Requirements for specific structures can easily be greater than these guidelines. If the First Condition requires a larger substrate dimension than the Second Condition, it is very important that the larger dimension is used.

If you are using the far field viewer, the larger the box the better. The far field viewer assumes that S-parameters from **em** are from a perfect open environment. If some of the power is reflected due to a box that is too small, the input power

calculated by the far field viewer will be slightly incorrect. The far field viewer then calculates antenna efficiencies greater than 100%. If this occurs, the box size should be increased.

Third Condition: *Place the top cover outside the fringing fields (i.e., near field) of the radiating structure, preferably a half wavelength.*

If this condition is violated, the resistive top cover becomes involved in the reactive fringing fields which form the near field of the radiator. This changes what would have been reactive input impedance into resistive input impedance, overestimating the radiation loss.

Do not place the top cover thousands of wavelengths away from the radiator. Extreme aspect ratios of the box should be avoided. Empirical data for patch antennas has shown that a distance of about $1/2$ wavelength works best.

Fourth Condition: *Set the top cover to Free Space.*

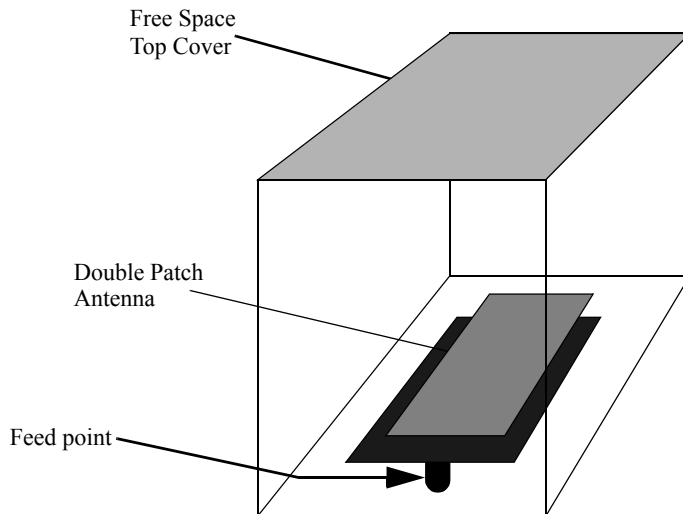
This value is a compromise. As shown by the equations on the previous page, all TE modes have a characteristic impedance larger than 377 ohms (Ω), while all TM modes are lower. Thus, while a 377 Ohms/square top cover does not perfectly terminate any mode, it forms an excellent compromise termination for many modes. This approximates removing the top cover of the box. If the box is large, it, in turn, approximates radiation, as shall be demonstrated.

Fifth Condition: *The radiating structure can not generate a significant surface wave.*

If there is a significant, compared to required accuracy, surface wave, it is reflected by the sidewalls of the box. Unless this is the actual situation, such antennas are inappropriate for this technique. Actually, the Fifth Condition is a special case of the Second Condition, since if there is significant surface wave, the Second Condition cannot be met. This condition is stated explicitly because of its importance.

In general, any surface wave is both reflected and refracted when it encounters the edge of the substrate. This boundary condition is different from either the conducting wall of Sonnet or the infinite substrate provided by a true open space analysis.

A dual patch antenna is illustrated conceptually below.

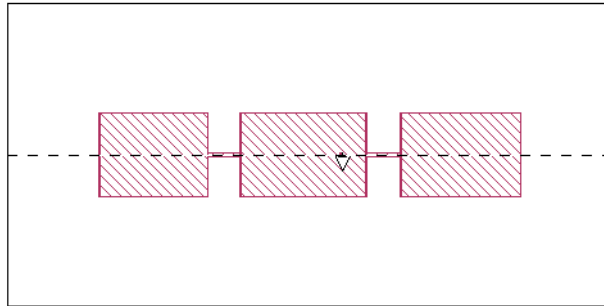


Radiation can be simulated by including a lossy top cover, a lossy dielectric layer (optional) and by placing the sidewalls far from the radiator (drawing not to scale). Place the top cover one half wavelengths from the radiator.

The feed point is created in the project editor by creating a via to ground at the feed point. Then the ground end of that via is specified as a port, just as one would specify a more typical port on the edge of the substrate at a box sidewall. A file showing an antenna similar to this one is named “dual_patch.son” and is available in the Sonnet examples.

Validation Example

For validation, we offer work performed by E. Ongareau of Matra Defense, Antennas & Stealthness Dept., France, as presented at the 1993 EEsof User's Group meeting at HYPER in Paris. (Reprinted with permission.) The antenna is a triple patch structure, with a top view shown below. The antenna is a test realization intended only for validation. It is not designed for optimum VSWR.



Top view of a triple patch antenna (courtesy of Matra Defense). The central patch is fed with a coaxial probe (indicated by a down pointing triangle). Each patch is resonant at a different frequency to increase the overall antenna bandwidth.

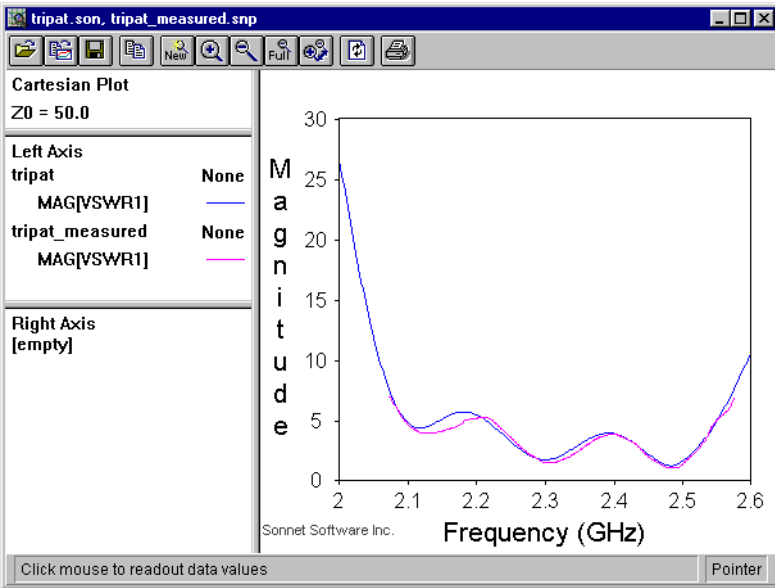
Good results are also regularly obtained on single microstrip patch antennas. We cite this example as one of the more sophisticated antennas analyzed using the Open Waveguide Simulator technique.

In this antenna, each patch has a slightly different resonant frequency, resulting in an increased bandwidth. The antenna is fed from below with a coax probe attached to the central patch. The feed point is indicated with a triangle.

The substrate is 3.04 mm thick with a dielectric constant of 2.94. The drawing is to scale with substrate dimensions of 200 mm x 100 mm. The top cover is 200 mm above the substrate surface. Cell size is 0.78125 mm square. A loss tangent of 0.001 is used in both air and substrate. The small air loss helps terminate the propagating modes.

The antenna project, [Tripat](#), is available in the Sonnet examples. For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button.

The chart below shows the result. We see that the low VSWR points of each patch have differences between measured and calculated of about 1%. This is typical of most analyses of patch antennas using this technique. The differences in resonant frequency (i.e., the reflection zeros) then determine the differences in the rest of the plot. The degree to which these differences are due to analysis error, fabrication error and measurement error cannot be determined from this data.



The measured and calculated data for the triple patch antenna were obtained completely separately, so there was no chance to “tweak” the model for agreement.

If the typical differences between measured and calculated data shown above are acceptable, given the specific requirements for a particular project, then the Open Waveguide Simulator technique can provide useful results.

Far Field Viewer

The purpose of the far field viewer is to calculate the far field pattern of an antenna for a given excitation and set of directions (for example, phi and theta ranges). The far field viewer starts by reading the current density data generated by *em* for the antenna at the desired frequencies. The far field viewer uses the current distribution information in the project and generates the desired far field antenna pattern information. This pattern information is viewed in one of three ways: Cartesian, polar, or surface plot. A default set of values for directions, port excitations and terminations are used to calculate plots for the first frequency upon start-up of the far field viewer. Thereafter, the user specifies the frequencies, directions for the radiation pattern and the desired port excitations and terminations.

Since the far field viewer uses the current density data generated by *em*, it can analyze the same types of circuits as *em*. These include microstrip, coplanar structures, patch antennas, arrays of patches, and any other multi-layer circuit. As with *em*, the far field viewer can analyze any number of ports, metal types, and frequencies. The far field viewer cannot analyze circuits which radiate sideways, structures with radiation due to vertical components, coaxial structures, wire antennas, surface wave antennas, ferrite components, or structures that require multiple dielectric constants on a single layer.

Be aware that although the current data is calculated in *em* with a metal box, the metal box is removed in the far field viewer calculations. The modeling considerations discussed earlier in the chapter are important, however, for the accuracy of the far field viewer data relies on the accuracy of the *em* simulation.

By default, the far field viewer analyzes the first frequency in the current density data stored in the project at a default set of angles and port excitations when the file is opened.

To obtain the antenna pattern for other than the first frequency, you must select *Graph* \Rightarrow *Calculate* from the far field viewer main menu. The Calculate dialog box allows you to set up all the parameters for the data you desire to calculate. The far field viewer calculates the fields radiated by the current that is stored in the

project. The analysis is performed in an open environment with a substrate of infinite extent. For details on the Calculate dialog box, please refer to the online help in the far field viewer program.

Please refer to “A Two-Dimensional Far Field Viewer Tutorial,” page 145 in the **Sonnet Supplemental Tutorials** for a tutorial on using the far field viewer.

Analysis Limitations

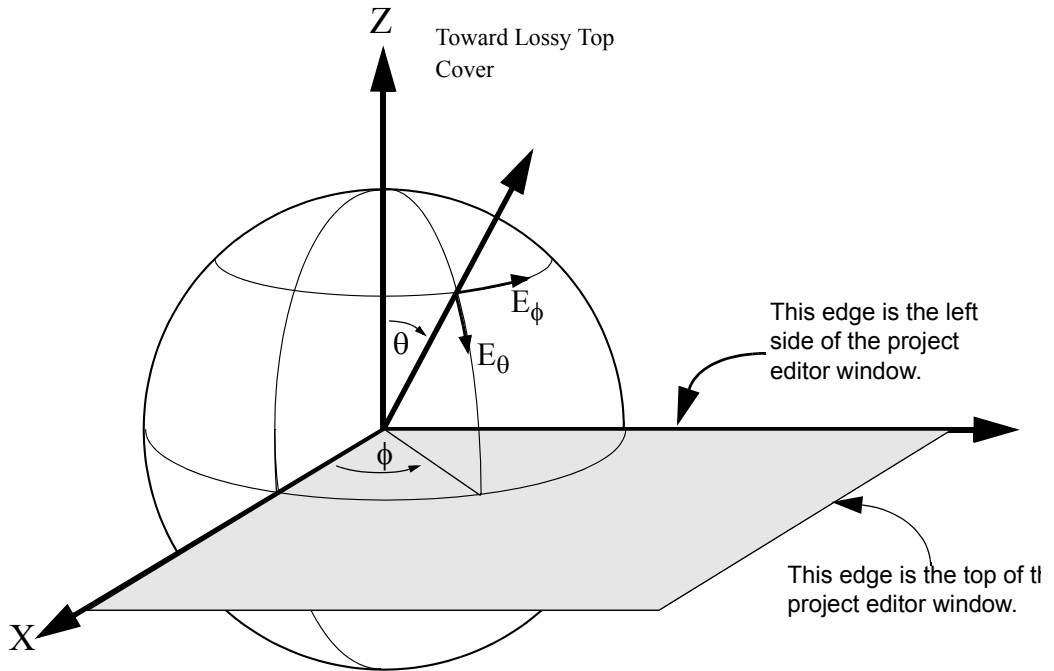
The analysis of the far field viewer has the following limitations:

- The plotted antenna patterns do not represent de-embedded data. Therefore, the effect of the port discontinuity is still included even if you specify de-embedding when running *em*.
- Radiation from triangular subsections (i.e., diagonal fill) is not included.
- The far field viewer patterns are for a substrate which extends to infinity in the lateral dimensions.

Spherical Coordinate System

You view your antenna plot using the spherical coordinate system, which is described below.

To view an antenna plot, the far field viewer uses the spherical coordinate system shown below. The X, Y, and Z coordinates are those used in the analysis engine and the project editor. The XY plane is the plane of your project editor window, with the Z-axis pointing toward the top of the box. The spherical coordinate system uses theta (θ) and phi (ϕ) as shown in the figure below.



The far field viewer allows values for theta from 0 to 180 degrees. However, values of theta greater than 90 degrees are below the horizon and are only useful for antennas without infinite ground planes. To view just the top hemisphere, sweep theta from 0 to 90 degrees and sweep phi from -180 to +180 degrees.

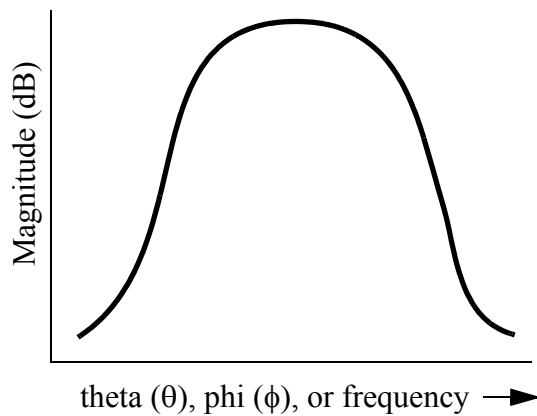
The X and Y axes in the figure above correspond to the X and Y axes in the project editor. The origin is always in the lower left corner of the project editor window.

To look at an E-plane cut or an H-plane cut, set phi (ϕ) to 0 or 90 degrees, and sweep theta (θ) from 0 to 90 degrees. To view an azimuthal plot, set θ and sweep ϕ .

NOTE:

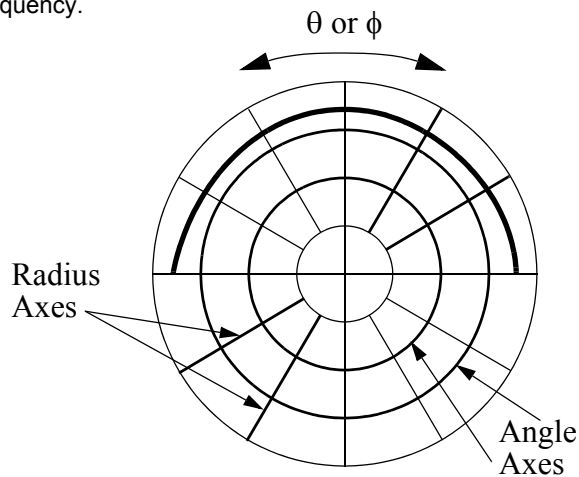
The far field viewer will allow the user to analyze the same space twice with the user determining the appropriate angle ranges for each analysis. For details, see “Graph - Select” in the far field viewer's help.

The far field viewer displays three plot types; cartesian, polar, and surface. All three types of plots are shown below. The cartesian plot allows the magnitude (in dB) to be plotted on a rectangular graph with your choice of theta (θ), phi (ϕ), or frequency for the X-axis as shown in the figure below. The polar plot allows you to select either theta (θ) or phi (ϕ) for the angle axis. The surface plot shows all the calculated values of theta and phi plotted against the gain for a single frequency.

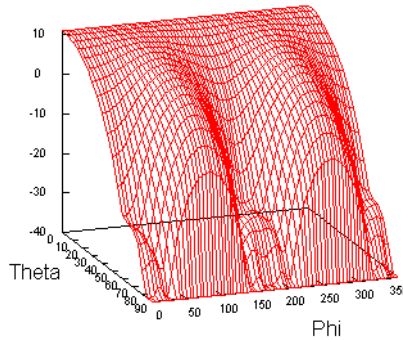


The far field viewer allows you to sweep theta (θ), phi (ϕ), or frequency.

(a) Cartesian



(b) Polar



(c) Surface Plot

Normalization

There are three types of normalization to choose from in the far field viewer. By default, the far field viewer displays the power gain. The far field viewer can also provide directive gain and absolute values. The three types of normalization are discussed below.

The power gain is defined as the radiation intensity divided by the uniform radiation intensity that would exist if the total power supplied to the antenna were radiated isotropically^[1].

Directive gain is defined as the radiation intensity from an antenna in a given direction divided by the uniform radiation intensity for an isotropic radiator with the same total radiation power.^[2]

The Gain and Directive gain may be displayed relative to the isotropic antenna (i.e. 0 dB), the maximum value of E for the antenna, or an arbitrary value.

Selecting Absolute for the normalization displays the radiated power in Watts/steradian at a given angle.

You may change the normalization in the far field viewer using the Select Normalization dialog box which is opened by selecting *Graph* \Rightarrow *Normalization* from the far field viewer main menu.

Polarization

The far field viewer displays the magnitude of the electric field vector for a given direction. The magnitude may be represented as the vector sum of two polarization components, E-theta (E_θ) and E-phi (E_ϕ) as shown in the figure describing spherical coordinates on page 353.

The far field viewer allows you to see the total magnitude or either component of the magnitude. Other polarizations are also available in the far field viewer and are discussed in “Graph - Polarization” in the far field viewer help.

References

- [1] Simon Ramo, John R. Whinnery and Theodore Van Duzer, Fields and Waves in Communication Electronics, John Wiley & Sons, Inc. 1994, pg. 601.
- [2] Ibid, pg. 600.

Chapter 22

SPICE Model Synthesis

Sonnet's analysis engine, *em*, provides a frequency domain solution in the form of S- Y- and Z-parameters. Many time domain simulators, such as traditional SPICE engines, do not have the capability to import frequency domain data, or have problems with efficiency, stability, or accuracy when using frequency domain data.

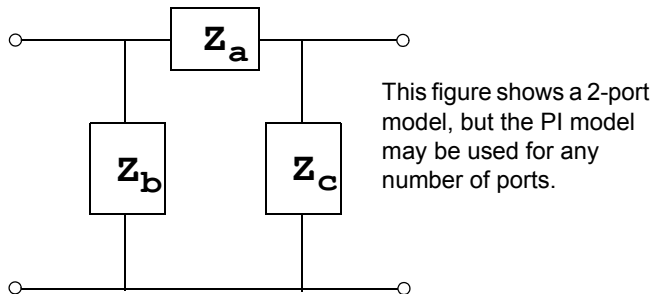
To address these problems, Sonnet provides the capability to output your frequency domain data into a SPICE-compatible file. You may choose from three different models, depending on your needs:

This chapter discusses three solutions provided by Sonnet:

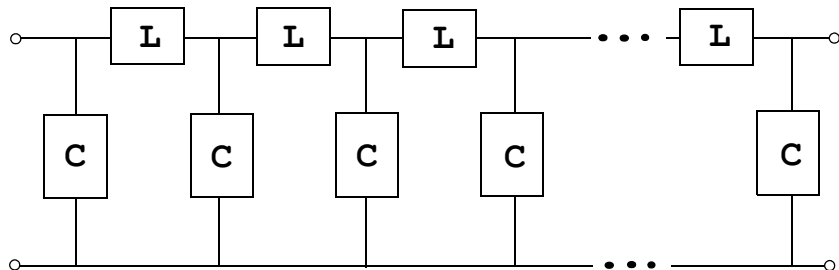
- PI (Lumped Element) Spice Model
- N-Coupled Line (Transmission Line) Spice Model
- Broadband Spice Model using rational polynomial fit (requires a

separate Broadband Spice Extractor license)

The first model is a lumped element fit to the S-parameter data. For this type of model, the circuit often resembles the symbol “ π ,” therefore, this type of model is referred to as a PI Model. Sonnet creates this type of model when you specify a PI Model output file in the PSpice or Spectre format (*Analysis \Rightarrow Output Files* in the project editor or *Output \Rightarrow PI Model File* in the response viewer). The PI model is only applicable for a narrow band. The model that is output is usually one which is intuitive and easily understood by a user as shown below.



The second model is an N-coupled line (transmission line) model. Sonnet creates this type of model when you specify an N-Coupled Line Model output file in the LCT format (*Analysis \Rightarrow Output Files* in the project editor or *Output \Rightarrow N-Coupled Line Model* in the response viewer). This model is valid at only one frequency point. This model produces a RLGC matrix which can be used in many popular Spice programs. The equivalent circuit is shown below for a single lossless transmission line.



There are an infinite number of infinitesimal elements in the model

A third way to bridge the gap between the frequency domain and the time domain is to fit the frequency domain data with a rational polynomial. Sonnet's Broadband Spice Extractor feature uses this method to provide a circuit model which is valid over a broad band. This model, unlike the lumped element model described above, does not yield an intuitive understanding of a design. Instead, the Broadband Spice Extractor feature generates a model that can be used in Spice as a “black box” representing the broad band behavior of your circuit as shown below. This type of model will be referred to as the Broadband Spice model.



This chapter describes how to use Sonnet to automatically synthesize PI Model, N-Coupled Line Model and Broadband Spice Model files. The PI and N-Coupled Line capabilities are useful for circuits which are small with respect to the wavelength at the highest frequency of interest. This includes structures such as discontinuities like step, tee and cross junctions. Other applications include modeling cross-talk and propagation delay in digital interconnect circuits and multiple spectrum circuits that combine digital, analog and RF functions. The Broadband Spice model is fitted over a wide frequency band and can be used in circuit simulators for AC sweeps and transient simulations.

NOTE:

Broadband Spice Extractor is only available if you have purchased a license from Sonnet which includes the Broadband Spice Extractor feature. Please see your system administrator if you are unsure of the availability of this option.

PI Spice Model

Specifying an optional PI Model Spice output file automatically takes the results of the electromagnetic analysis of a circuit and synthesizes a model using inductors, capacitors, resistors and mutual inductors. This information is then formatted and saved in one of two SPICE formats: PSpice or Spectre.

The PI model Spice generation capabilities are intended for any circuit which is small with respect to the wavelength at the highest frequency of excitation. Typically, 1/20th wavelength is an appropriate limit. (If a circuit is too large, you can often split it into two or more circuits and analyze each separately.) This limitation is due to the circuit theory limitations of modeling a circuit with just a few lumped elements. The Sonnet electromagnetic analysis is not intrinsically limited in this fashion.

The model generated by the analysis includes any lumped elements (including mutual inductors) between any ports of the circuit layout. Lumped elements from any port to ground are also included. The synthesis capability does not allow internal nodes (nodes which are not connected to a port in the layout) with the single exception of the internal node required to specify a resistor in series with an inductor.

Any circuit which requires internal nodes for an accurate model should be split into several parts so that the required points become nodes. Internal ports without ground reference give incorrect results. Any internal ports should be carefully specified and checked for reasonable results.

Using The PI Model Spice Option

The PI Model synthesis needs electromagnetic results for at least two frequencies to accomplish its work. It is not possible to create a PI model if the circuit is analyzed at only one frequency. A PI model is created for pairs of frequencies. The second frequency is determined by taking the first frequency and adding a percentage specified by the user. The second frequency then becomes the first frequency for the next pair of frequencies for which a SPICE model is generated. The synthesis continues in this way until all the frequencies have been used.

The default value for the separation percentage is 10%. In this case, a SPICE model is generated using the first frequency and the next highest frequency which provides a 10% gap. This continues until there are no more frequencies. For example, for a frequency sweep from 10 - 40 GHz with a 0.1 GHz resolution using a separation of 10%, the first few frequencies used would be 10, 11, 12.1 and 13.3 GHz.

After completing the analysis, always do a “reality check” for reasonable values. If you have bad data, the frequency may be too high or too low. If the frequency is too low, the solution may have unity S-parameters, causing a strange SPICE model. To be absolutely sure your results are good, select a different frequency band and re-analyze the circuit. You should obtain similar results between the two analyses.

You may obtain PI Model Spice data in two different ways. The first is to specify an optional output file before executing your analysis. The second is to generate a PI model from the response viewer. The second method has the advantage of allowing you to perform the data check mentioned above before creating the SPICE data file.

To specify a PI Model Spice output file from the response viewer, perform the following:

- 1 Analyze the circuit at the desired frequencies.**

The analysis monitor appears on your display to show the progress of the analysis. It is important to note that if your results contain more than two analysis frequencies, then multiple Spice models, one for each pair of frequencies, will be created in one file.



- 2 When the analysis is complete, click on the View Response button on the analysis monitor's tool bar.**

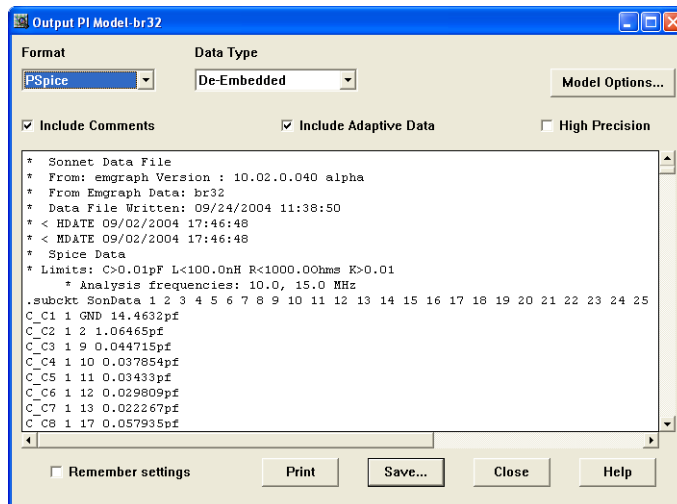
The response viewer is invoked with a plot of your response data.

3 Click on the project name in the response viewer legend to select it.

An outline appears around the project name to indicate that it is presently selected. If you have multiple projects open in the response viewer and have not selected a project before using the extraction command, then a window appears which allows you to select the desired project.

4 Select Output ⇒ PI Model File from the response viewer main menu.

The Output PI Model dialog box appears on your display. The contents of the output window in the Output PI Model dialog box displays the Spice data for the PI Model in the PSpice format which is the default.

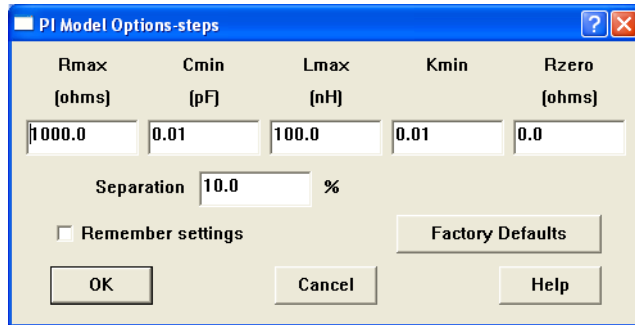


5 Select the desired file format from the Format drop list.

If you select Spectre as the Format (PSpice is the default), the contents of the output window is updated.

6 To change the parameters for the SPICE data, click on the Model Options button.

The PI Model Options dialog box appears on your display. To reduce the number of lumped elements in the model, open circuit limits can be specified here as shown below.



The image shows a dialog box titled "PI Model Options-steps". It contains five input fields for parameters: Rmax [ohms] (1000.0), Cmin [pF] (0.01), Lmax [nH] (100.0), Kmin (0.01), and Rzero [ohms] (0.0). Below these is a "Separation" field set to 10.0 % and a checkbox for "Remember settings" which is unchecked. There are three buttons at the bottom: "OK", "Cancel", and "Factory Defaults". A "Help" button is also present on the right side of the dialog box.

Rmax [ohms]	Cmin [pF]	Lmax [nH]	Kmin	Rzero [ohms]
1000.0	0.01	100.0	0.01	0.0

Separation 10.0 %

☐ Remember settings

Factory Defaults

OK Cancel Help

The values are defined as follows:

RMAX: Maximum allowed resistance (ohms). The default value is 1000.0 ohms.

CMIN: Minimum allowed capacitance (pF). The default value is 0.01 pF.

LMAX: Maximum allowed inductance (nH). The default value is 100.0 nH.

KMIN: Minimum allowed mutual inductance (dimensionless ratio). The default value is 0.01.

RZERO: Resistor to go in series with all lossless inductors (resistance in ohms). Needed for some versions of SPICE. The default value is 0.0

Separation: This is the calculation interval between the two frequencies used to generate the SPICE model specified as a percentage. The second frequency is obtained by adding the specified percentage of the first frequency to the second frequency.

All calculated component values which fall outside the allowed range specified by the user in the model options are excluded from the resulting lumped model. The RZERO entry is provided for those versions of SPICE which need inductors to have some small loss to avoid numerical difficulties. The default value of 0.0 disables this capability.

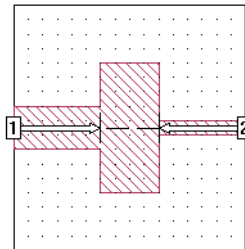
Enter the desired values for the parameters in the PI Model Options dialog box.

- 7 **Click on the OK button in the PI Model Options dialog box to apply the changes and close the dialog box.**
- 8 **Click on the Save button in the Output PI Model dialog box.**

A browse window appears which allows you to save the data displayed in the output window. The file extension depends on which type of SPICE format you have selected.

A Simple Microwave Example

Shown below is the [Ste_sym](#) example, a simple step discontinuity followed by the PI model produced when you set up an optional PI Model Spice output file.

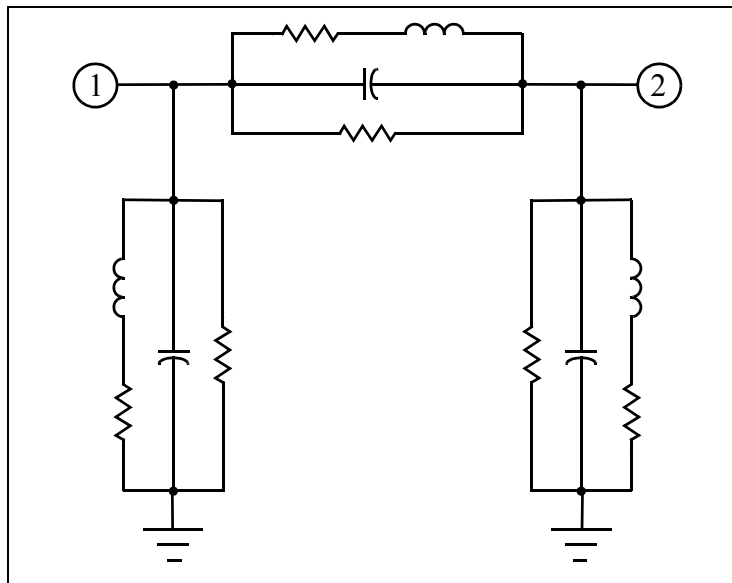


```
* Limits: C>0.01pF L<100.0nH R<1000.0Ohms K>0.01
* Analysis frequencies: 1000.0, 1100.0 MHz
.subckt SonData 1 2 GND
C_C1 1 GND 0.273341pf
C_C2 2 GND 0.232451pf
L_L1 1 2 0.310155nh
.ends SonData
```

There are two capacitors to ground (node GND) and one inductor connected between node 1 and node 2 in the lumped element model.

Topology Used for PI Model Output

The topology of the lumped element model generated by *em* depends on the circuit being analyzed. In general, the model contains an inductor (in series with a resistor if using loss), a capacitor and a resistor (when using loss) connected in parallel from each port to ground. A similar parallel RLC network is also connected between each port. Therefore, a four-port circuit can contain more elements than a two-port circuit. Each inductor may also have a mutual inductance to any other inductor in the network. The figure below shows the most complex equivalent circuit possible for a two-port (mutual inductances not shown). Any values that are outside of the open circuit limits are not included.



Equivalent circuit of a two-port structure using the PI Model. Mutual inductances also exist between all inductors, but are not shown. Any component whose value is outside of the open circuit limits are not printed in the SPICE output file.

N-Coupled Line Option

If your circuit is a single or multiple-coupled transmission line, you should use the transmission line models supplied by your Spice program. Most SPICE programs have an element for single and multi-conductor transmission lines in which you enter the capacitance and inductance per unit length for a single line or a capacitance and inductance matrix for multi-coupled lines. Sonnet can produce a LCT formatted Spice file which may, without too much difficulty, be edited to comply with other formats.

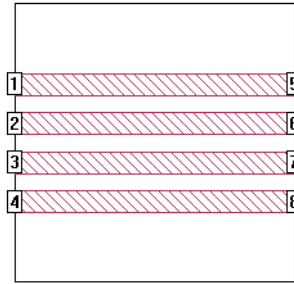
Analyses which use such data are much faster than those which use simple lumped models. In addition, accuracy is maintained at all frequencies for which TEM mode propagation is an adequate approximation.

For a single line, the L and C distributed parameters are each a single number. For N-coupled lines, L and C become N by N matrices. When metal loss is included, we now also have an R matrix. The resistance is in series with the inductance. When there is dielectric loss, a G matrix is also calculated. The conductance is in parallel with the capacitance. The synthesis determines whether a G or R matrix is needed only from the calculated S-parameters, not from the circuit geometry.

To generate RLGC matrices, select *Output* \Rightarrow *N-Coupled Line Model File* in the response viewer or click on the N-Coupled Line Model button in the Generate Default Output Files dialog box which is opened when you select *Analysis* \Rightarrow *Output Files* from the project editor menu. The analysis control needs only one frequency specified. If two frequencies are specified, two RLGC matrix sets are generated and so on.

Your project must be an N-coupled line with ports 1 through N as input and ports N+1 through 2N as output. The input of line M should be port M and its output should be port M+N. The software does not check for this condition, but issues a

warning message if the number of ports is not an even number. This restriction does not apply to generating PI Models, only generating N-Coupled Line Model Spice files. There is no limit on N. An example where $N = 4$ is shown below.



The results are per unit length, where a unit length is the length of the N-coupled line. The length must be short compared to the wavelength at the frequency of analysis.

Broadband SPICE Model

NOTE:

The Broadband Spice Extractor feature is only available if you have purchased a Broadband Spice Extractor license from Sonnet. Please see your system administrator if you are unsure of the availability of this feature.

In order to create a Spice model which is valid across a broad band, the Sonnet broadband SPICE Extractor feature finds a rational polynomial which “fits” the S-Parameter data. This polynomial is used to generate the equivalent lumped element circuits in either PSpice or Spectre format. Since the S-Parameters are fitted over a wide frequency band, the generated models can be used in circuit simulators for AC sweeps and transient simulations.

To create a Broadband Spice file, you open your project in the response viewer and select *Output ⇒ Broadband Model File* from the main menu. (You may also create a Broadband Model by using the *Analysis ⇒ Output Files* command in the project editor. See online Help for Details) This opens the Output Broadband

Model File window which allows you to calculate a broadband SPICE file based on the analysis data for your project. You need a minimum of 50 frequency points in order to generate a Broadband Spice file; therefore, we highly recommend that you use an ABS sweep when analyzing your circuit to ensure the correct number of analysis frequencies. If your circuit contains parameterization or optimization data and you select more than one parameter or iteration combination, then you may choose to create one file which contains all the Broadband Spice Models or multiple files, one for each specified combination.

Class of Problems

Be aware that there are several types of responses for which an accurate Broadband Spice Model may not be produced:

- If your response data contains a data point which sharply deviates from the data curve, such as you would see for a box resonance, or a narrow band spike, the Broadband Spice model may not accurately model that response.
- The Broadband Spice model is generally not accurate for response data below -60 dB.
- A gentle curve may sometimes get fitted with a straight line.
- Broadband Spice Extractor has only been tested for passive circuits.
- Broadband Spice Extractor has only been tested using S-parameters produced by *em*. However, you should be able to use S-parameters produced by other sources such as other simulators or measured data to create a Broadband Spice model.

If you are concerned with the accuracy of the model, you should visually inspect the predicted S-Parameter data, produced by the same rational polynomial which was used to create the Broadband model, to determine the usefulness of the Broadband Spice model.

NOTE:

Be aware that the processing time needed to create your models can be significant. The processing time is proportional to the number of analysis frequencies times the square of the number of ports in your circuit.

Creating a Broadband Spice Model

You use the response data created as the result of an *em* analysis to create a Broadband Spice model. For the best results use an Adaptive sweep (ABS) to analyze your circuit and produce response data spread evenly over the frequency band. The following procedure demonstrates the method to be used in the response viewer. For detailed instructions for setting up a Broadband Model file in the project editor, please refer to online help.

Once you have completed the *em* analysis of your circuit, do the following to create a Broadband Spice model:

- 1 Open your project in the response viewer.**
- 2 Select Output => Broadband Model File from the main menu of the response viewer.**

If you have more than one project open in the response viewer, a window appears which allows you to select the desired project.

- 3 Select the desired project from the project drop list.**

You may only create a Broadband Spice Model for one project at a time.

- 4 Click on the OK button to select the project and close the dialog box.**

The Output Broadband Model File dialog box appears on your display. You may access online help by clicking on the Help button or use context sensitive help for an explanation of a particular control or entry.

- 5 Select the file format of the Broadband Model: PSpice or Spectre from the Format drop list.**

PSpice and Spectre are the two formats of Spice file supported. A PSpice file uses the extension “.lib” and a Spectre file uses the extension “.scs.”

6 Enter the desired name for the Spice model file in the Model File text entry box.

A default filename is provided which places the Broadband Spice model file in the same directory as your source project. If you have selected multiple parameter combinations, a file is produced for each combination. If you wish to use another name or save the file in another location you may edit the text entry box.

7 Enter the desired error threshold.

The lower the error threshold you set, the more processing time is required to calculate the model. The error threshold is the error present between the source data and the fitted curve and is defined as follows:

$$Error = \frac{\left(\sum_{f=1}^N |s(f)_{Source} - s(f)_{Fit}| \right)}{N} \cdot 100$$

where f = the number of the frequency point

N = the total number of frequency points

$s(f)_{Source}$ = the value of the S-parameter at the frequency point f in the project source response

$s(f)_{Fit}$ = the value of the S-parameter at the frequency point f in the fit curve data

The calculation of the Broadband Spice model stops when this error threshold is reached or when it proves impossible to improve the error. We recommend using the default value of 0.5% initially and not setting the threshold below 0.1%.

8 Select the Generate Predicted S-Parameter data file if it is not already selected.

This option is selected by default. When this option is selected, the Predicted S-Parameter data upon which the model is based is also output to a file. The name of the file appears below the checkbox and may not be changed. The file is created in the same directory in which your model file is created. Once the creation of the

model is completed, you may use the response viewer to compare your original response data to the Predicted S-Parameter data to evaluate the accuracy of the Broadband model.

9 Click on the Create button to create the Broadband Spice model.

A progress window appears on your display. Be aware that the processing time needed to create your models can be significant. The processing time is proportional to the number of analysis frequencies times the square of the number of ports in your circuit. If you wish to stop the process before it is complete, click on the Cancel button in the progress window.

10 Once the calculation is complete, the Broadband Spice details dialog box appears on your display.

A log of the creation process appears in this dialog box. The log contains information on the error for each parameter, which parameter had the greatest error, and the filename of the predicted S-Parameter data. It will also indicate whether the model achieved the error threshold. Use this information to determine which parameters to examine in the response viewer. You should look at the S-parameter with the greatest error as well as any critical S-parameters whose error was greater than 0.1%.

For a detailed explanation of the log, see "Broadband Spice Extractor Log" on page 375.

11 Click on the Plot button to open a plot of your response data versus the predicted S-Parameter data.

The Predicted S-Parameter data is opened in the response viewer along with the original response data. You should view the S-parameter with the greatest error and any other critical response data whose error was greater than 0.1% to judge if the "fit" or accuracy of the model is sufficient for your needs.

If the model is not accurate enough, see "Improving the Accuracy of the Broadband Spice Model" on page 377 for suggestions on improving your Broadband Spice model.

Checking the Accuracy of the Broadband Spice Model

It is possible to save the predicted S-parameter data created while calculating your Broadband Spice model so that you may visually check the accuracy of your model when it is complete. To save the predicted S-parameter data, you should select the **Generate Predicted S-Parameter data file** checkbox in the Output Broadband Model File dialog box when you create your model. The predicted S-parameter data file is created in the same directory as your Broadband Spice model file.

Upon the completion of creating your Broadband model, you should open the original project in the response viewer, then add the predicted S-parameter data file to your graph and compare the two responses. If you are creating your Broadband Model in the response viewer, you may do this automatically by clicking on the Plot button in the Broadband Spice Details dialog box which appears upon completion of your model. Use the log information in the Details window, which is detailed in the next section, to determine which parameter had the highest error and any critical parameters whose error was greater than 0.1%. Check these parameters to see how much the curve fit data varies from your circuit response.

If your Broadband Spice Model is to be used for a transient analysis, be aware that the frequency response of the model up to $1/T$ where T is the minimum time step of the transient analysis is important. You should use the Advanced Broadband Model options dialog box to specify additional predicted data up to $1/T$. You access this dialog box by clicking on the Advanced button in the Broadband Model File entry dialog box in the project editor or in the Output Broadband Model dialog box in the response viewer. This allows you to view the frequency response of your model at data points not included in your *em* analysis. You should look for anomalies in the response that indicate a problem with the model, such as S-parameters greater than one or unexpected sharp resonances.

If the model is not accurate enough, see "Improving the Accuracy of the Broadband Spice Model" on page 377 for suggestions on improving your Broadband Spice model.

Broadband Spice Extractor Log

The Broadband Spice Extractor log, displayed in the Broadband Spice details window, contains detailed information about the creation of your Spice model file. You may view a summary of the log or the complete log. To view the summary of the log, click on the Summary button at the bottom of the window. To return to the full log, click on the Complete button.

You use the log to determine which parameters to examine in order to determine if the Spice model is accurate enough for your use. Two log files are shown below, the first log is for a model which achieved the error threshold and the second log

is one in which the error threshold was not achieved on all the parameters. A warning message is issued for all S-parameters whose error is greater than the error threshold.

```
Generating files coup_end.lib and coup_end_predict.snp.
--Model Log for coup_end--
Data set has 201 points and 4 ports

--Model Options--
Error threshold(%) = 0.5      ← Error Threshold
Output predicted file: C:\Program Files\son- ← Curve Fit Data File
net\project\coup_end_predict.snp
Max order target: 200      ← Maximum Order

--Model Results--
S11 Order = 2      Error(%) = 0.03850722
S12 Order = 2      Error(%) = 0.005954352
S13 Order = 2      Error(%) = 0.005379771
S14 Order = 2      Error(%) = 0.02510101
S22 Order = 2      Error(%) = 0.001637623
S23 Order = 2      Error(%) = 0.001033019
S24 Order = 2      Error(%) = 0.005380005
S33 Order = 2      Error(%) = 0.001635971
S34 Order = 2      Error(%) = 0.005951772
S44 Order = 2      Error(%) = 0.03853657

--Model Summary for coup_end--
Maximum error was for S44, Error(%) = 0.0385366 ← S-parameter with greatest error
Total model time: 0.391 seconds
Model coup_end successful ← Indicates that the Error Threshold
                             was achieved for all S-parameters
```

← Error values for S-parameters


```
Generating files matchnet.lib and matchnet_predict.snp.
--Model Log for matchnet--
Data set has 1246 points and 2 ports

--Model Options--
Error threshold(%) = 0.5
Output predicted file: C:\Program Files\son-
net\project\matchnet_predict.snp
Max order target: 200

--Model Results--
S11 Order = 203   Error(%) = 1.387911  WARNING: Error threshold
of 0.5 (%) not achieved
S12 Order = 208   Error(%) = 2.984112  WARNING: Error threshold
of 0.5 (%) not achieved
WARNING: Poor figure of merit on S12 parameter. Visual inspection
of predicted S12 recommended.
S22 Order = 214   Error(%) = 1.833756  WARNING: Error threshold
of 0.5 (%) not achieved
WARNING: Model prediction is not passive at 112 frequency points.
Error threshold may need to be decreased or input data may be non
passive.

--Model Summary for matchnet--
Maximum error was for S12, Error(%) = 2.98411
Total model time: 25 minutes 43 seconds
Model matchnet finished with no errors, 5 warnings
```

Annotations:

- Curve Fit Data File (points to the output file path)
- Error Threshold (points to the 0.5 value)
- Warning Messages (points to the WARNING lines)
- Indicates that a "fit" was found for all S-parameters but for some the error exceeded the error threshold. (points to the model summary line)

Improving the Accuracy of the Broadband Spice Model

If you need to increase the accuracy of your Broadband Spice model, there are several strategies you may use.

- If the Broadband model met your error threshold criteria but is still not acceptable, you may decrease the error threshold to increase the accuracy of the model. Be aware, however, that the processing time may be significantly increased by lowering the error threshold.

Typically, values below 0.1% result in unacceptably long analysis times.

- If there are more than 200 frequency points in your response data, try decreasing the number of frequencies in your response data. To do so, use the *Analysis* \Rightarrow *Clean Data* command in the project editor to remove the response data, then run another Adaptive sweep (ABS) using a coarser resolution to produce less data points but still more than 50 data points. You may change the resolution of an adaptive sweep in the Advanced Options dialog box in the project editor. Select *Analysis* \Rightarrow *Setup*, then click on the Advanced button in the Analysis Setup dialog box which appears.
- Increase the number of data points in the critical frequency band in which you are concerned and decrease the number of data points in frequency ranges which are not as important.

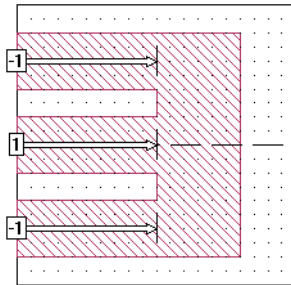
Chapter 23

Coplanar Waveguide and Balanced Ports

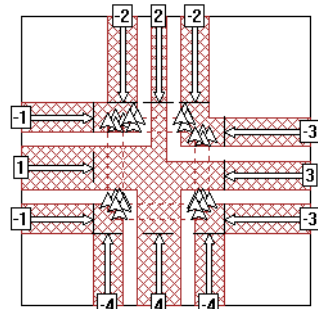
Sonnet can be used to model many different types of coplanar waveguide (CPW) lines, but each type is modeled differently in Sonnet. This chapter describes the different modeling techniques for the most common types of coplanar waveguide circuits.

Ungrounded Coplanar

The most common type of CPW is called “ungrounded coplanar,” where there is no metal under the substrate. Two simple examples are shown below and are available from the Sonnet examples.



A co-planar waveguide short circuit [Cosht_sy](#)



A co-planar waveguide cross junction [Cocross](#)

Notice that the ground ports have been labeled with a negative number (e.g. “-1”), while the signal ports are labeled with positive numbers (e.g. “1”). This represents a balanced port. Any number of ports can have the same positive or negative labels. This is done in the project editor by selecting the ports in question and opening the Port Attributes dialog box by selecting *Modify* \Rightarrow *Port Properties*. **Em** sums the total current going into all the positive ports with the same port number and sets that equal to the total current going out of all the ports with that same negative port number. Thus the name “balanced”, or “push-pull,” port.

Note that we end the width of the ground lines just before they would have touched the sidewall. If the side of the ground line touches the side wall, it shorts out to the sidewall, thus allowing ground current to return via the sidewall instead of through the ground line. This defeats the purpose of the balanced port. Be sure that your ground lines touch the sidewall only at the location of negative port numbers when using balanced ports.

The current density view of coplanar structures may show high current on the outside edge of the ground strip. This is caused from the fact that the infinite ground of an ideal coplanar line is truncated to a finite ground strip. Using a wider polygon to represent the ground strip will reduce this effect, however analysis times will increase. Generally, if the width of the ground strip is 3 or 4 times the gap size, current on the outside edges has only a small effect on the results.

Note that the use of symmetry in the CPW short circuit provided the same results in less time than analyzing without symmetry. When a circuit is symmetric, *em* does not subsection any part of the circuit below the center line. But notice that the “-1” port below the center line is the same distance from that center line as the other “-1” port above the center line. As long as there are ports identical in every way, i.e. impedance, number, etc., equidistant from the center line, you may use symmetry for a circuit.

Different Types of Coplanar Waveguide

Any transmission line with three coplanar conductors, where the center conductor is fed 180 degrees out of phase with the two outer conductors, can be considered a coplanar waveguide line. The following is a list of some of the more popular configurations:

- 1 Ungrounded Coplanar: No metal on the bottom of the substrate.
- 2 Grounded Coplanar: The bottom of the substrate has a metal ground plane which is electrically connected to the two outer conductors.
- 3 Floating Coplanar: Metal exists on the bottom of the substrate, but it is not connected to the two outer conductors. Thus the bottom metal is electrically "floating" at a different potential from the two outer conductors.
- 4 Infinite Coplanar: The two outer conductors are considered infinitely wide. Usually, there is no metal on the bottom of the substrate (case #1), but any of the above three configurations can have finite or infinite outer conductors.
- 5 Shielded Coplanar: Outer conductors are shorted to side-wall of a rectangular waveguide. Thus, all four sides of the waveguide are at the same

ground potential as the coplanar outer conductors. Waveguide dimensions are part of the design.

Push-pull ports can be used to simulate only some types of coplanar waveguide. Since the two outer ports have the same port number, they are electrically connected together. Creating the two -1 ports now redefines port 1 such that the voltage on port 1 is now measured with respect to the two -1 ports.

Another way to think of this is that the positive terminal of the generator is connected to the center conductor (port 1) and the negative terminal is connected to the two -1 ports. You may also think of port 1 as your "signal" and the two -1 ports as your "ground".

If you do not put ports on the two outer conductors, the voltage of port 1 is measured with respect to the six sides of the metal box. If the sidewalls of the box are far away, then this means port 1 is measured with respect to the bottom of the box.

Therefore, the method you use to simulate your coplanar line depends on how you are exciting it. Each of the above cases should be simulated differently:

- 1 Ungrounded Coplanar: Add an extra air layer under the dielectric to move the bottom of the Sonnet box far enough away so it has no effect. The sidewalls of the box should also be moved far away (at least several line widths). If the sidewalls and box bottom are far away, using push-pull ports should give the same answer as shorting the outer conductors to the box wall.
- 2 Grounded Coplanar: Do not use push-pull ports. Just connect the two outer conductors to the sidewall of the box (the same sidewall to which the center conductor is connected). This electrically connects them to the metal ground plane on the bottom of the substrate.
- 3 Floating Coplanar: Use push-pull ports because the metal on the bottom of the substrate is not connected to the two outer conductors.
- 4 Infinite Coplanar: Since the two outer conductors are considered infinitely wide, this cannot be simulated with Sonnet. However, you can approximate this case by truncating the two outer conductors to a finite width. Do not use push-pull ports unless you have a floating coplanar

line as described in #3. Instead, connect the ground conductors to the box sidewall without ports. This is because the error introduced by truncating the outer conductors can be partially compensated for by allowing the current that should have traveled on the outside edge of the outer conductor to travel through the box sidewalls.

- 5 Shielded Coplanar waveguide: In this case, create the Sonnet box to be the same dimensions of your coplanar waveguide box. Do not use push-pull ports. Instead, connect the ground conductors to the box sidewall without ports.

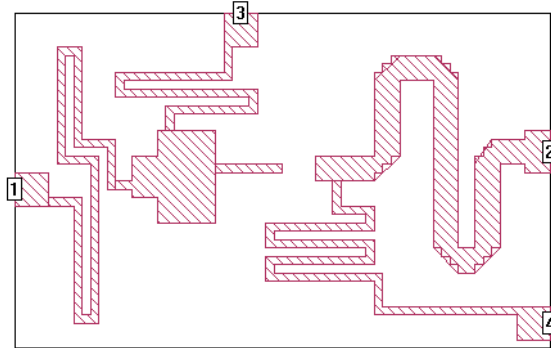
In cases 1-4 the outer conductors do not get connected to the two walls running parallel to the line. In case 5, the conductors typically get connected to the walls.

Chapter 24 Package Resonances

Simply stated, the Sonnet analysis is a solution of Maxwell's equations. These general equations are not limited to a purely TEM or Quasi TEM analysis. For a given structure, if a higher order mode (TE or TM) can propagate or an evanescent mode can exist, it will be included in the results. The strongest evidence of the presence of a "waveguide" mode occurs when the 6 conducting sides of the Sonnet box create a resonant cavity. As most microwave designers can attest to, these box resonances occur in practice as well. The designer can use Sonnet to predict unwanted box resonances in the package or module housing the circuit.

In this chapter, we will outline several ways to detect package resonances within the Sonnet simulation based on an example project. As you will see, this is a great way to prove a package design early in the design cycle. We will also outline the use of the Box Resonance Estimator and give some advice as to how to remove box resonances from a structure.

To obtain the example file (package.son) used in this chapter, get the example folder [Package_resonances](#) from the Sonnet examples. For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button.



The file “package.son” is a model of an amplifier used to check for package resonances. The entire width of the box is not shown.

Box Resonances

The purpose of this section is to give the user a basic understanding of how to detect box resonances in a Sonnet project or simulated data. There are three ways to do so:

- 1 Runtime warning messages
- 2 Observations of simulated results
- 3 The Box Resonance Estimator

Runtime Warning Messages

If the proper selection is made during the analysis setup, Sonnet will detect box resonances and output warning messages in the analysis monitor, while the simulation is being performed. The steps to enable this feature are:

- 1 Select *Analysis* \Rightarrow *Setup* from the project editor menu.

- 2 In the Analysis Setup dialog box which appears, click on the Advanced button.
- 3 In the Advanced Options dialog box which appears, click on the Box Resonance Info check box.
- 4 Click on the OK button to close the Advanced Options dialog box.
- 5 Click on the OK button to close the Analysis Setup dialog box.



Warnings

The warning symbol, shown to the left, appears in the analysis monitor when a box resonance is detected. If you click on the Errors/Warning button on the analysis monitor, you can view all of the warning messages associated with that particular analysis.

Below is an example of the first type of box resonance warning message. When this message appears there is a box resonance detected in the primary structure.

```
Sonnet Warning- EG2680:  
Circuit has potential box resonances.  
Filename: C:\Program Files\sonnet\project\package9.son  
Primary structure.  
First few ideal resonant frequencies are:  
30.0866 GHz TE Mode 0, 1, 1  
31.7587 GHz TE Mode 0, 1, 2
```

Note that the warning message is describing box resonances which appear in the primary structure. The term “primary” refers to the actual structure being analyzed. By “ideal resonant frequency” we mean a theoretical value based on an empty Sonnet box. The specified dielectric stackup is considered, but the effect of any circuit metallization and loss parameters are not.

Below is an example of the second type of box resonance warning message. When this message appears there is a box resonance detected in a calibration standard.

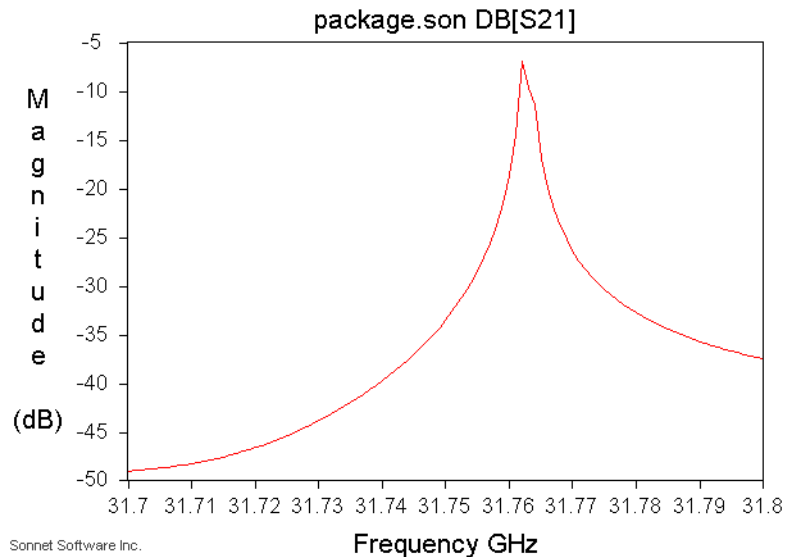
```
Sonnet Warning- EG2680:  
Circuit has potential box resonances.  
Filename: C:\Program Files\sonnet\package.son  
Second de-embedding standard, left box wall.  
First few ideal resonant frequencies are:  
30.0871 GHz TE Mode 0, 1, 1  
31.7625 GHz TE Mode 0, 1, 2
```

Note that the warning message defines which calibration standard is causing the problem. The observant reader will notice that these resonant frequency values are different than with the primary structure. This occurs because Sonnet actually creates and analyzes two calibration standard as a part of the de-embedding procedure. These standards may have a different box size than the primary structure, which causes the change in the resonant frequencies. For more information on the calibration standards and de-embedding, please refer to **Chapter 7, “De-embedding”** on page 119 and **Chapter 8, “De-embedding Guidelines”** on page 131

Observations of Simulated Results

A second way to detect box resonances is with a manual review of the simulated data. Typically box resonances appear as sharp changes (glitches or spikes) in S-parameter magnitude and phase data. They can also be evident in E_{eff} and Z_0 data. This is because there is a resonance in at least one of the standards that **em** creates for de-embedding. Box resonances can also corrupt de-embedding results. Because **em**'s de-embedding feature is based on circuit theory, it possesses the same limitation that all de-embedding algorithms share. It is unable to de-embed a structure contained inside a resonant cavity (box). This means that if a box resonance exists for a de-embedding calibration standard, the final S-parameters will be suspect.

Below is an S-parameter (S21) curve for the example project “package.son”. Notice that at 31.76 GHz there is a sharp change in the data and it approaches unity. This indicates a strong, package resonance induced coupling between the input and output at this frequency.



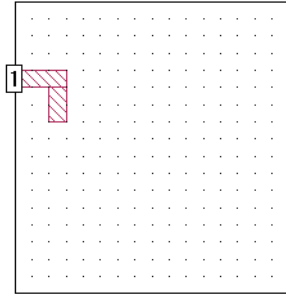
Results of a search for package resonances shows strong coupling between input and output at 31.7625 GHz.

A Box Resonance Example

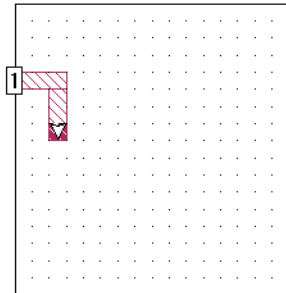
This example describes how to create a simple geometry file you can use to determine box resonance frequencies before you fabricate the wrong enclosure. Errors of less than 0.1% can be achieved with no limit on the number of dielectric layers used.

The basic idea is to re-create the box parameters of your real circuit, using the same substrate size, dielectric layers, etc. but without the metalization.

Once the box parameter setup is complete, you should create a small probe which is used to excite the modes. This is just a small (less than $1/8$ wavelength) open stub with a port on it. If you bend the stub, you have a better chance of exciting both X-directed and Y-directed modes.

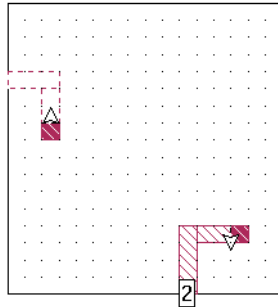


To excite Z-directed modes, connect a via to the end of the probe.

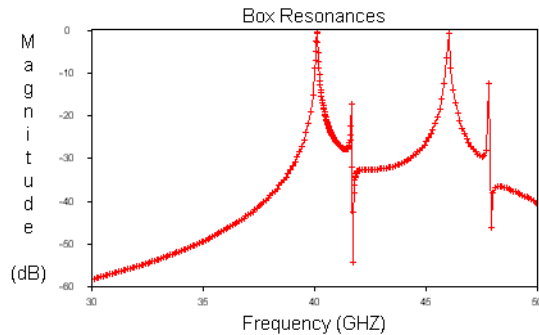


Make sure you don't place the probe precisely in the middle of the box wall. You want to make sure you excite both even and odd modes.

Then create another probe on an adjoining box wall. We do this in order to measure the coupling between the two probes. During a resonant situation, the coupling will increase.



Sweep the frequency in fine steps and look for resonances. When the box resonates, there should be stronger coupling between your two probes. This can easily be seen by plotting the magnitude of S21.



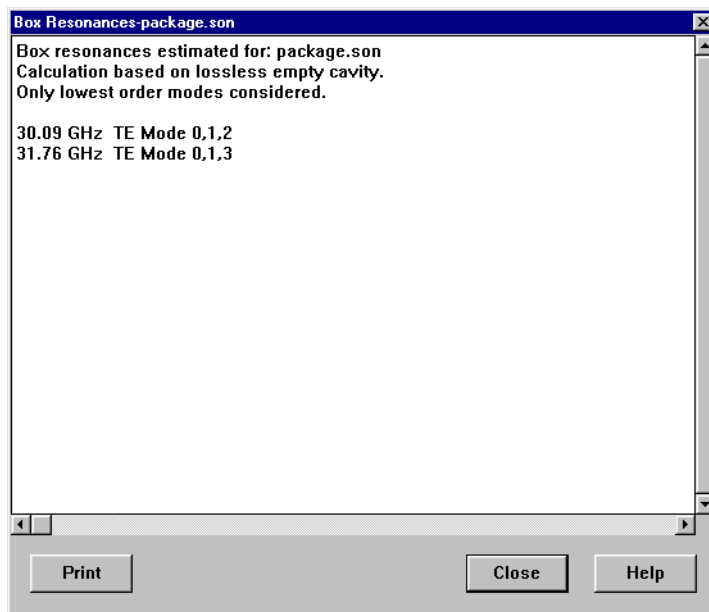
The peaks represent box resonances.

The Box Resonance Estimator

The best way to understand the box resonance situation within your package is to use the Box Resonance Estimator BEFORE running an analysis. It is recommended that this tool be routinely used to prevent wasted simulation time. It is an extremely useful tool because it allows the user to make modifications to

the structure and gauge its impact on box resonances. It can also be used after a simulation is complete to help determine which characteristics of a complex data curve are related to box resonances. As with the runtime warning message, these are theoretical values based on an empty Sonnet box. The specified dielectric layers are considered, but the effect of any circuit metallization and loss parameters are not.

To access the Box Resonance Estimator, select *Analysis* \Rightarrow *Estimate Box Resonances* from the project editor main menu. The Box Resonances dialog box appears on your display. An example is shown below.



The Box Resonance Estimator displays not only the resonant frequencies contained in the simulation frequency range, but also tells the user the particular mode type. It is sophisticated enough to realize that when Symmetry is enabled for a circuit that Even Y modes will not exist.

In this particular instance, two Transverse Electric (TE) modes exist in or around the desired frequency range. The Box Resonance Estimator will detect Transverse Magnetic (TM) modes as well. For a complete description of propagation modes, please refer to any classic field theory textbook.

NOTE:

The Box Resonance Estimator only checks the primary structure not the de-embedding calibration standards.

Box Resonances – Simple Removal

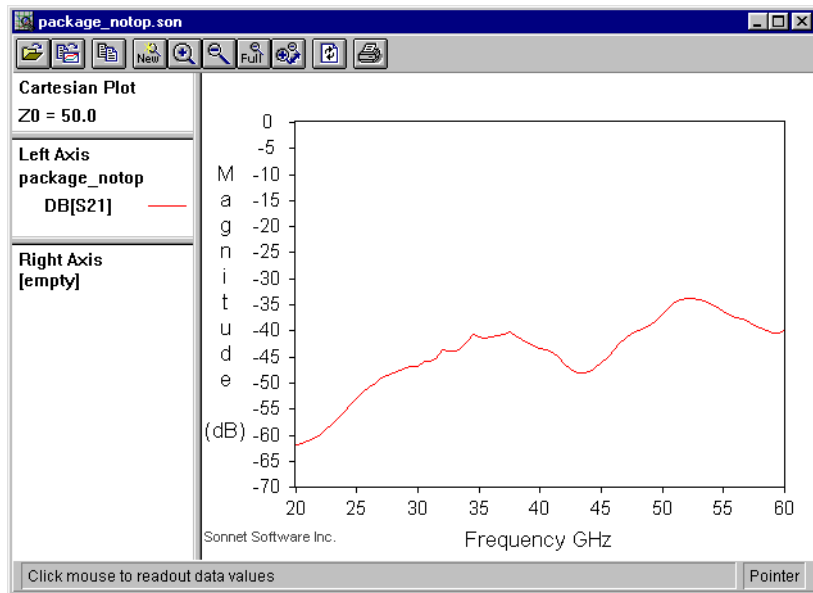
In the preceding section, we have described several ways to detect box resonances within a structure. We would now like to offer some advice as to how to remove them when they are undesirable. By this we mean the case when you simply do not wish to consider their effect and would like them removed from the data. This can occur when you analyze a portion of your overall circuit in Sonnet and the boxwalls artificially introduce resonances. Removal is actually probably not the best term to describe this approach. It is more of an attempt to push the resonances out of the desired frequency band or attenuate their levels.

The best way to remove a box resonance is to change the size of the box, either larger or smaller, to move the resonant frequency out of band. If the problem occurs in de-embedding, you may be able to change the length of the calibration standard in the project editor to move the box resonance out of the band of interest.

Another simple way to remove or at least attenuate the effect of a box resonance is to take off the top cover. We can create an approximation of this condition by setting the top cover resistivity to 377 ohms/square, the impedance of free space. To do this, open the Box Settings dialog window (Circuit -> Box Settings) and change top cover to “Free Space”. This is an accurate approximation provided the cover is not so close that it interacts with the evanescent fringing fields surrounding the circuit. Please note that it is inaccurate to place the top cover directly on top of the circuit without an intervening dielectric layer.

Using either technique will entail changing basic project parameters making it necessary to analyze the project again.

Below is the resulting S-parameter (S21) curve with the top cover set to free space. Please note that while the resonance is still evident its level has been greatly attenuated. Again, the data is from a simulation of the example project “package.son”.



The package resonances disappear when the top cover is removed.

Taking the top cover off works, provided the sidewalls of the box are large enough to form a propagating waveguide up to the top cover, or you can place the top cover close enough to the substrate surface to catch the fields in the box mode. High order “box” modes tend to be confined primarily to the substrate and can be difficult to remove in this manner. As you make the box bigger by increasing the substrate surface area, the modes “loosen up” so that they can propagate to the top cover and become absorbed.

The Capability to Ask: What if?

In the preceding section we mentioned some simple techniques for removing box resonances from simulated data. But what if your package is well defined physically, and you can't simply take the cover off? In this situation, Sonnet can be an invaluable tool. It allows the user to make changes to the structure and evaluate the impact on box resonances. As noted earlier, the Box Resonance Estimator is a great tool to judge, almost in real-time, the effect of the package size. From this the user can gain an understanding of which dimension controls any one resonance type (TE or TM) and may provide some insight as to the solution.

Some of the techniques used to mitigate box resonances include:

- Adding grounded metal geometries to your structure – In many cases, ground vias or groundplanes will help prevent box resonance modes from forming. In simple terms, they have the effect of dividing the structure into smaller compartments, thus pushing the box resonances higher in frequency. The use of a via fence or CPW type transmission line can help “channelize” the circuit, preventing unwanted coupling between traces and reducing unwanted box resonance effects. All of these features can be included in the Sonnet model.
- Adding an absorptive material to the model – Another approach which is normally used once all other package design features have been optimized, is to add an absorbing material in the housing cavity. The material is normally iron or carbon loaded so it can provide a fairly high magnetic or electric loss tangent. It comes in various forms such as liquid or sheet and is usually placed at a position such that it has a minimal effect on the circuit performance and a great effect on the box resonance. The user can easily model this material in Sonnet by adding a dielectric layer to the stackup.

Chapter 25 Viewing Tangential Electric Fields

One reason *em* is so fast is that all of the electric and magnetic fields are solved for analytically, with “pencil,” “paper” and many equations. The computer need only do an FFT and solve for the current distribution.

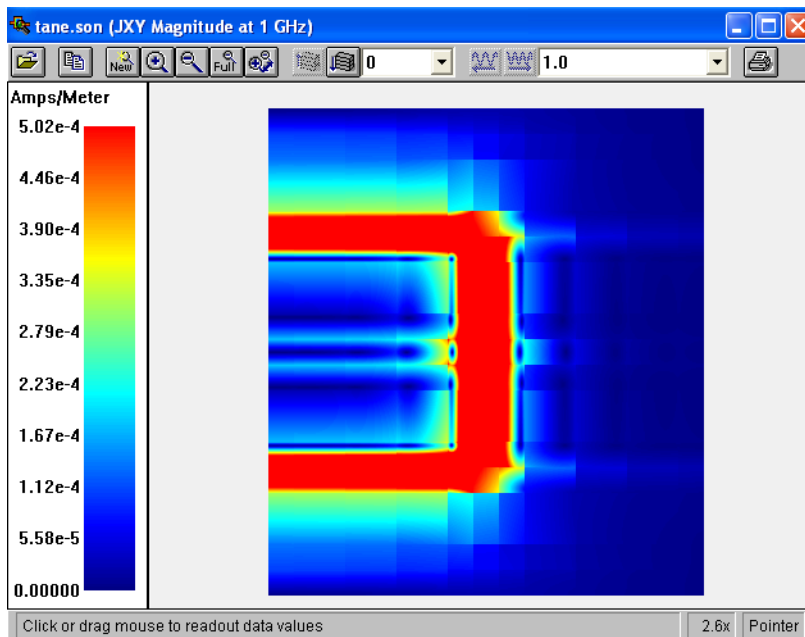
However, on occasion, you want to view the fields, not the current. You do this with what is called a “sense layer”. The sense layer is a rectangular patch of conductor placed where you want to see the tangential electric field.

Actually, describing the sense layer as a conductor is misleading. This is because you set the surface reactance of the conductor to some large value, say 1,000,000 ohms per square and the surface resistance to zero. You may do this by selecting the Sense Metal definition when defining your metal type in the Metal Editor dialog box. To access this dialog box, select *Circuit* \Rightarrow *Metal Types* from the main menu in the project editor, then click on the Add button in the Metal Types dialog box which appears on your display.

You set the reactance to such a large value so that the sense layer has little influence on the original fields. An intuitive analogy is to view a sense layer like inserting a sheet of paper (very high reactance) into the fields. Because the reactance of the sense layer is high, the currents are very small. The sheet of paper does not change the fields.

But even though they are small, what are the currents? The current density is proportional to the tangential electric fields over the area of the sense layer. This is just a two dimensional version of Ohms Law: Current is proportional to Voltage.

An example is shown below in the current density viewer. You may use the Manual Examples list in the Examples PDF documents to get a copy of this project, [Tane](#). For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button.



The tangential electric fields just above a gap discontinuity. The input voltage comes from the left. Strong fields are present across the gap, especially at the corners. This analysis was performed at 1 GHz.

Tane is a good example showing tangential E-field. This circuit uses sense metal set to $X_{dc} = 1.0e6$ ohms/square. So, any current on this metal will be proportional to the E-field.

If you view the current density on level 0, you should see the values in the “red” area are about 0.001 Amps/meter. You can click on the circuit and see the value of the current in the status bar at the bottom of the window. Then, simply use the equation (based on Ohms law):

$$E = JX$$

where,

E= Electric field in Volts/meter

J = Current density in Amps/meter (0.001 A/m)

X = X_{dc} value of the metal in ohms/sq (1.0E6)

This gives E = 1000 V/m. This is assuming a voltage at port 1 of 1.0 Volts (default setting).

Chapter 26 Accuracy Benchmarking

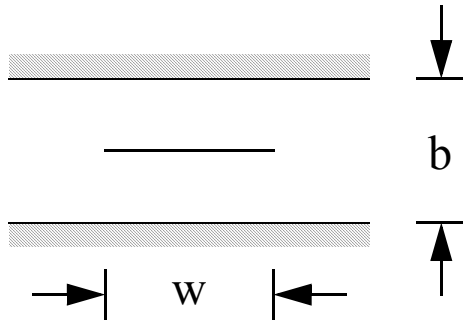
Electromagnetic analyses are often described as providing what is called “Good Agreement Between Measured And Calculated” (GABMAC). However, in the past, there has been little effort to decide just what “good” means. The more useful result is the “Difference Between Measured And Calculated” (DMAC).

In this chapter, we describe a precise benchmark, based on [21], [22] and [24], which allows the evaluation of DMAC for any 2.5-D or 3-D electromagnetic analysis down to the 1×10^{-8} level of accuracy.

There is an example of a coupled stripline benchmark available in online help under Applications.

An Exact Benchmark

What we need to calculate DMAC is an exact benchmark. One source of an exact benchmark is stripline. The characteristic impedance of a stripline has an exact theoretical expression $K(k)$ and is the complete elliptical integral of the first kind. For evaluation on a computer, a polynomial for $K(k)$ is available in **Abramowitz and Stegun, Handbook of Mathematical Functions**, pp. 590 - 592. (Be sure to note the errata, $m1 = 1-m2$, not $1-m$.):



$$Z_0 \sqrt{\epsilon_r} = \frac{\eta_0 K(k')}{4 K(k)}$$

$$k = \tanh\left(\frac{\pi w}{2b}\right) \quad k' = \sqrt{1 - k^2}$$

$$\eta_0 = 376.7303136$$

The expression for $K(k)$ cited above provides an accuracy of about 1×10^{-8} . When programmed on a computer, the following values are obtained for three different transmission line impedances (unity dielectric constant):

Table 3 Stripline Benchmark Dimensions

Z_0 (ohms)	w/b
25.0	3.3260319
50.0	1.4423896
100.0	0.50396767

For a length of stripline, there are two parameters of interest: characteristic impedance and propagation velocity. With the w/b given above, we know the exact answer (to within 1×10^{-8}) for Z_0 . With a dielectric constant of 1.0, we also know the exact answer for the propagation velocity. It is the speed of light, known to about 1×10^{-9} . Any difference from these values is error, or, DMAC.

Each of the above three benchmarks is available in the Sonnet examples. To get the 50 ohm line, get the example [S50](#). The other benchmark circuits are in “[S25](#) and [S100](#)”. For directions on obtaining a Sonnet example, select *Help* \Rightarrow *Examples* from the menu of any Sonnet program, then click on the Instructions button.

The “b” dimension is exactly 1.0 mm, the “w” dimension is given by the above table and the length of each line is 4.99654097 mm with a dielectric constant of 1.0. Each of these lines is precisely 0.25 wavelengths long at 15.0 GHz. The geometry projects have the subsectioning set so the lines are 16 cells wide and 128 cells long.

To evaluate DMAC, do an analysis of the line at 15 GHz, with de-embedding enabled. For the error in characteristic impedance, take the percent difference between the calculated value and the exact value, above. For the error in propagation velocity, take the percent difference between the calculated S_{21} phase and -90 degrees. Total error, in percent, is the sum of the two errors.

Some types of analyses do not calculate characteristic impedance. A detailed error analysis shows that, to first order for a 1/4 wavelength long 50 ohm line, the value of $|S_{11}|$ is equal to the error in characteristic impedance. For example, an $|S_{11}| = 0.02$ means that there is about 2% error in characteristic impedance. To use this approximation for, say, a 25 ohm line, the S-parameters must be converted to 25 ohm S-parameters. This may be done by adding transformers in a circuit theory program.

Residual Error Evaluation

We have performed a detailed analysis of the relationship between subsectioning and residual error (DMAC). The simplest way to subsection a line is to use subsections the same width as the line. In Sonnet, and in many other analyses, this results in a uniform current distribution across the width of the line. In reality, the current distribution is singular at the edges of the line.

Since the current distribution is symmetrical about the center line, using either one or two subsections across the width of the line gives the same amount of error.

We find that a one or two subsection wide line gives 5% to 6% error. If there is not much stray coupling, circuit theory can often give a better result. When the line is 16 cells wide, we see about 1% error, much more reasonable. We have found (and you can verify) that convergence is very strong: Double the number of cells per line width and the error is cut in half.

When we vary the number of cells per wavelength, along the length of the line, we see an inverse square relationship. Double the number of cells per wavelength along the length of the line and the percent error decreases by a factor of four.

An equation which expresses the error as a function of subsectioning is:

$$E_T \cong \frac{16}{N_W} + 2\left(\frac{16}{N_L}\right)^2 \quad N_W \geq 3 \quad N_L \geq 16$$

where

N_W = Number of cells per line width,

N_L = Number of cells per wavelength along line length,

E_T = Total Error (DMAC) (%).

This equation estimates subsectioning error only. For example, any de-embedding errors are added to the above error. This error estimate should be valid for any electromagnetic analysis which uses roof-top subsectioning.

Notice that the quantities used for the error estimate are in terms of cells, not subsections. Cells are the smallest possible subsections size. In Sonnet, subsections in the corners of polygons are one cell on a side. Subsections along the edge of polygons are one cell wide and can be many cells long. Interior subsections can be many cells in both dimensions.

We have found that, for most cases, the cell size is the important parameter in determining error. Or in other words, the smallest subsection size is important. For example, the stripline benchmark geometry projects, mentioned before, are set to make the lines 16 cells wide, even though those 16 cells may be merged into only 4 or 5 subsections. It is the 16 cells which determine the level of error, not the 4 or 5 subsections.

In performing this error evaluation, we also found that the error in characteristic impedance due to N_W is always high, never low. Also, there is very little variation in the error for different impedance lines. The above equation can be very accurate in evaluating error. And, finally, for N_L above about 40 cells per wavelength, all the error is in the characteristic impedance. The error in velocity of propagation is essentially zero.

The above equation can be very accurate in evaluating error. With this precise knowledge of the error, we can now do something about it.

Using the Error Estimates

The above error estimate can be used to estimate the error for an overall circuit. Let's say that a cell size is used that makes some high impedance transmission lines only 1 cell wide. Other, low impedance transmission lines, are, say, 30 cells wide. The 1 cell wide lines give us about 5% error. The 30 cell wide lines give about 0.5% error. In non-resonant situations, you can expect the total error to be somewhere between 5% and 0.5%. If most of the circuit is the low impedance line, the error is closer to 0.5%, etc.

However, let's say that our circuit has resonant structures. Let's say it is a low pass filter. It is easy to verify by means of circuit theory that the low pass filter is very sensitive to the high impedance lines. This means we can expect about 5% error, even though the high impedance lines only make up half the filter.

Given this information, there are several courses of action. First, if 5% error is acceptable, no further effort is needed.

More likely, we wish to analyze the filter with less error. Since we now know the error in the characteristic impedance is 5%, we can physically widen the line so that the characteristic impedance is 5% lower to compensate for the known increase in characteristic impedance due to subsectioning the line only one cell wide. Very precise analyses are possible using this compensation technique.

Chapter 27

Time Required for Analysis

Em is a memory and computation intensive program. Small circuits are analyzed quickly while large circuits can require a considerable amount of time. In some cases, it may be desirable to run the program in the background.

There is no simple rule for calculating the time required for a particular analysis although there are guidelines, presented below, which will afford you some measure of control over that time.

The amount of time required is closely related to the number of subsections, which is shown in the analysis monitor during an analysis. After a few trials, you will have a good idea of whether an analysis is a few seconds, a few hours, or just totally hopeless by looking at the number of subsections.

The most important factor in the execution time required is the amount of memory required by *em* compared with the amount of memory you have. This is also shown in the status in the analysis monitor. If *em* says you need 36 Mbytes and you have 16 Mbytes installed, then it is a good idea to kill the run quickly. You can reduce the memory required by using the Memory Save option, which makes the analysis single precision. You can also try to reduce the area of metalization in the circuit. Try to eliminate any metal that is not carrying current and make connecting lines as short as possible (but not too short, see Chapters 7 and 8 on de-embedding).

Another approach, if your memory requirement is right on the edge, is to free up some of your computer's memory. Make sure no one else is also running a memory intensive program at the same time.

The estimate of required memory printed out by *em* is just an estimate. It is usually within 1 Mbyte or so, but could be off by much more. To get both the memory estimate and the number of subsections without going on to actually analyze the circuit, use the Calculate Memory Usage option, available in the Additional Options dialog box.

For most circuits, the following equation can be used to estimate the amount of memory that will be used by *em*:

$$B = K * N^2$$

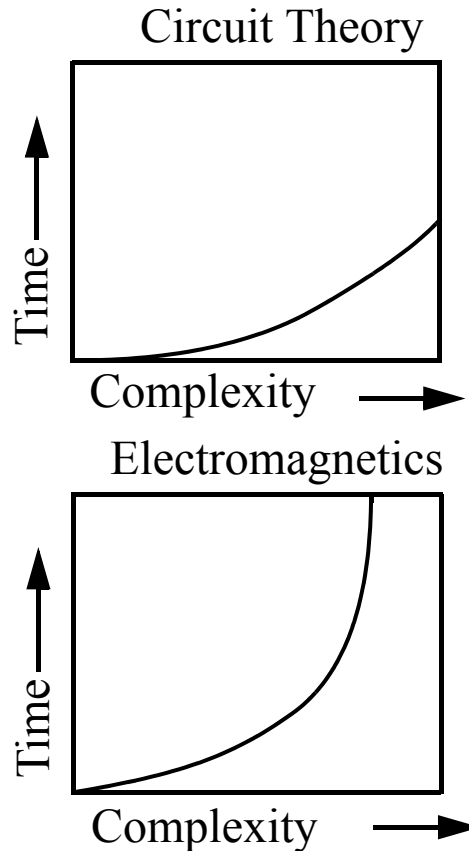
where B is the number of bytes, and N is the number of subsections. K is equal to 8 if running with double precision and loss. This can be circuit metal loss, top or bottom cover loss or dielectric loss. K is equal to 4 if you are running with loss but using Memory Save or running lossless and using double precision. K is equal to 2 if you are running a lossless circuit and using Memory Save. This equation should be used only as an estimate as it only includes the memory used by the final matrix in *em*. Circuits with large boxes (in terms of number of cells) or many layers require more memory. You should use this equation to calculate an upper limit on the number of subsections for your computer.

For UNIX systems, you can check the memory actually used by typing the "ps" (process status) command. Consult your system administrator or UNIX manuals for details on the "ps" command.

To check how much of your system's memory is actually available for your use, select *Help* \Rightarrow *System Info* from the project editor main menu. The System Information dialog box appears on your display and contains the information on system memory use.

The “Wall”

When using circuit theory analysis, an increase in circuit complexity gradually produces an increase in analysis time. With an electromagnetic analysis, the increase happens suddenly. A mere doubling of circuit complexity (say, by using a smaller subsection size) can result in one, or even two, orders of magnitude longer analysis. We call this the “Wall” (see the charts below).



The Wall is frequently encountered when **em** runs out of real memory, as described above, and is forced to start swapping out to disk. Execution time can quickly go from a few minutes to a few hours. Either get more memory or modify the circuit so that there are fewer subsections.

To avoid the wall, *start with no loss (metal or dielectric) and use a large subsection size*. Perform the first analysis at a single frequency to evaluate how long an analysis takes. Then, provided you get fast results, try adding loss or making the subsection size smaller. Keep going until the analysis is as long as you can tolerate and then let it run over a full range of frequencies, perhaps overnight.

The main factor in analysis time is the number of subsections. **Em** prints out the number of subsections if the Verbose option is used. As you gain experience with **em**, you will get a good feel for what can be tolerated. For example, on a Sun SPARCstation 1 with 16 Mbytes of memory, up to 1700 subsections (lossless) or 1200 subsections (with loss) can be calculated in an hour or so. At this point the computer runs out of memory and starts swapping to disk, resulting in huge increases in time.

To avoid the frustrations of getting on the slow side of the wall, *start lossless with big subsections*. You may find that big subsections provide all the accuracy you need.

Detailed Parameter Dependencies

How do changes in the various input parameters affect the analysis time?

First, keep in mind that **em** has two stages in the analysis. In the first stage, **em** fills a large matrix. The matrix has one row/column for each subsection. This is where **em** is calculating the coupling between every possible pair of subsections. In the second stage, **em** is solving the matrix. Here, **em** is performing matrix inversion-like functions and is calculating the currents which allow all boundary conditions to be met. The different parameters affect each stage of the analysis differently.

During analysis of the first frequency in a run, *em* prints out the amount of time spent on various portions of the analysis. If the time spent in any particular section of the analysis is less than one second, it is not printed out. Sections which are timed include the waveguide mode calculation (prior to matrix fill), the matrix fill and the matrix inversion.

Parameters which have no effect on analysis time include the substrate thickness, cover height and number of ports. Each of these parameters is unlimited and have no impact on speed while still maintaining complete accuracy.

Including metalization loss increases the matrix solve time only. And this is only if the rest of the structure is lossless. If there is any dielectric loss or ground plane (or top cover) loss, there is virtually no additional impact from also including metalization loss (the whole calculation is already fully complex). The matrix solution time is increased by about a factor of four (if it becomes complex). Metalization loss has no impact on any other segment of the analysis.

Including dielectric loss, or ground plane or top cover loss, makes the entire calculation complex. The matrix solve time is increased by about a factor of four, while the matrix fill time is increased by about a factor of two.

In calculating the values for the matrix elements during the matrix fill, several two dimensional Fourier Transforms must be calculated. The size of the Fourier Transforms is the same size as the substrate in terms of cells. If a substrate is 128 x 64 cell, each Fourier Transform is 128 x 64 elements. Memory storage is required for only one Fourier Transform at a time and this is usually much smaller than the matrix being filled.

In this release, the FFT has been improved for composite box sizes. A composite number is not a prime number, nor does it contain any large prime factors. For example, 1000 is a composite number because its largest prime factor is 5. But 998 is not a composite number because its largest prime factor is 499. So a 998 by 998 box might take 2 to 3 times longer in the FFT calculation portion of the analysis than a 1000 by 1000 box. The FFT portion of the analysis is usually a small percentage of the total analysis time, unless you have a lot of layers or an especially large box. If either of these conditions are true, then it might be worthwhile to use a composite number in your box size.

For all the above detail, keep in mind that if the substrate is small (less than 256x256 cells), the Fourier Transform time is of little consequence.

Matrix fill time is proportional to the number of subsections squared for large circuits.

The number of subsections, for a given cell size, can be reduced by minimizing the number of vertices and the number of diagonal lines in the polygonal description of the circuit. If the circuit is symmetric with no more than two ports, with both ports on the axis of symmetry, invoke the symmetry option for a significant memory and time savings.

The matrix solve time is proportional to the number of subsections cubed and is the main limitation on the analysis at this time.

Appendix I *Em* Command Line for Batch

If you wish to set up batch or script files to run your analyses overnight or at times of the day when the processing load is lighter, it is possible to use command lines to run *em* from a batch file. You should also be aware that it is possible to setup batch files with start and stop times using the analysis monitor. For directions on how to do so, please see **How do I create a batch file to run multiple analysis jobs?** in online help.

You should be aware that running from the command line does not provide all of the status information that is provided in the analysis monitor while running an analysis.

The syntax of the command line is as follows:

em -[options] <project name> [external frequency file]

where:

<**options**> is one or more of the run options shown in the table below. If you use multiple options they should be typed with no spaces in between after the minus sign. Note that other run options may be set in the Analysis Setup dialog box for your project and will be used during the analysis.

Option	Meaning
-Dlicense	Used for debugging <i>em</i> licensing problems. Displays all environment information relevant to licensing.
-N	Display number of subsections and estimated required memory. <i>Em</i> then exits without running a full analysis.
-fullsolve	Use this option in conjunction with the -N option above to provide accurate memory estimates when using Sonnet Lite, LitePlus, Level2 and Level3 suites.
-test	Run <i>em</i> on a test circuit. Used to verify that <i>em</i> can get a license and run successfully.
-v	Display analysis information as the analysis is performed. The analysis information is output to the command prompt window or terminal from which the batch was executed.
-AbsCacheNone	Disable ABS caching (overrides setting in project file).
-AbsCacheStopRestart	Enable ABS stop-restart caching (overrides setting in project file)
-AbsCacheMultiSweep	Enable ABS multi-sweep plus stop-restart caching (overrides project file)
-AbsNoDiscrete	Used when running ABS with pre-existing cache data. Tells the analysis engine not to do any more discrete frequencies. If pre-existing cache data is sufficient to get converged ABS solution, then that solution is written to output. Otherwise, no processing is performed.

Option	Meaning
-SubFreqHz[value]	where [value] is the subsectioning frequency in Hz. Note there is no space before the value field. This option allows subsectioning frequency to be specified on the command line, thereby overriding the settings in the project file.
-ParamFile <filename>	where <filename> is the name of a file which contains the value(s) which you wish to use for parameter(s) in the circuit being analyzed. These values override the value contained in the geometry project for the analysis, but do not change the contents of the geometry project. The syntax for the parameter file is <parname>=fnum where <parname> is the name of the parameter and fnum is a floating point number which defines the value of the parameter for the analysis.

<project name> is the name of the project which you wish to analyze. If there is no extension, then the extension ".son" is assumed. This field is required.

[external frequency file] is the name of an optional external frequency control file whose extension is ".eff". This extension must be included when specifying the control file. You may create an external frequency control file in the project editor. For details see Frequency Sweep Combinations in online help in the project editor. The frequencies in this file override the frequencies in the project.

For example, if you wish to analyze the project steps.son in a batch file using the -v option, the command line would be:

em -v steps.son

An example of a batch file which runs multiple analyses is shown below.

```
em -v steps.son
em -v filter.son filter.eff
em -v airbridge.son filter.eff
em -v airbridge.son
```

To execute a batch file on the PC, you should create a text file containing the command lines with a “.bat” extension. Then open a DOS prompt window and type the filename at the prompt and press return.

To execute a batch file on UNIX, create a text file containing the command lines. The filename does not need any extension. Then change the permissions mode of the file to allow you to execute it. For example:

chmod a+x <filename>

where <filename> is the name of the batch file you wish to execute. Then, type the name of the file at the UNIX prompt and press return.

On UNIX systems there are several additions to the command line which are useful to know. Placing “nice” before the command runs it at lower priority. Placing “&” at the end of the command runs it in background, so you get your cursor back. Entering “nohup” before the command line allows you to log off while the em job(s) keep running. If you are using the “&” or the “nohup”, you might want to consider redirecting the output using “> outfile”. See your system administrator for details on any of these options.

Appendix II Sonnet References

This appendix contains articles written by Sonnet authors or articles which directly impacted the analysis theory used by Sonnet. An extensive list of articles in which Sonnet was used as the analysis tool is available on Sonnet's website at www.sonnetsoftware.com. Search for "References".

- [1] James C. Rautio, "Accurate and Efficient Analysis of Large Spiral Inductors with Thick Metal and Narrow Gaps Using Space Mapping," IEEE MTT-S International Microwave Symposium, Workshop Notes & Short Courses - WFD-7, 6-11 June 2004.
- [2] James C. Rautio, "In Defense of Uselessness," IEEE Microwave Magazine, Vol. 5, No. 1, March 2004, pp. 100 - 102.
- [3] James C. Rautio, "A Conformal Mesh for Efficient Planar Electromagnetic Analysis," IEEE Transactions on Microwave Theory and Techniques, Vol. 52, No. 1, January 2004, pp. 257 - 264.

- [4] David I. Sanderson, James C. Rautio, Robert A. Groves, and Sanjay Raman, "Accurate Modeling of Monolithic Inductors Using Conformal Meshing for Reduced Computation," *Microwave Magazine*, Vol. 4, No. 4 December 2003, pp. 87 - 96.
- [5] James C. Rautio, "Testing Limits of Algorithms Associated with High Frequency Planar Electromagnetic Analysis," *European Microwave Conference Digest*, Munich, October 2003, pp. 463 - 466.
- [6] Rautio, James C., "Generating Spectrally Rich Data Sets Using Adaptive Band Synthesis Interpolation," *Workshop WFA, IEEE MTT Symposium Digest*, Philadelphia, June 8 - 13, 2003.
- [7] James C. Rautio, "Conformal Subsections for Accurate EM Analysis," *Microwave Journal*, Vol. 46, No. 6, June 2003, pp. 116 - 120. James C. Rautio and Veysel Demir, "Microstrip Conductor Loss Models for Electromagnetic Analysis," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 51, No. 3, March 2003, pp. 915 - 921.
- [8] James C. Rautio, "Electromagnetic Analysis Speeds RFID Design," *Microwaves & RF*, Vol. 42, No. 2, February 2003, pp. 55 - 62.
- [9] James C. Rautio, "EM Approach Sets New Speed Records," *Microwaves & RF*, Vol. 41, No. 5, May 2002, pp. 81 - 96.
- [10] James C. Rautio, Shawn Carpenter, et. al., "CAD/EDA virtual panel discussion," *Microwave Engineering*, May 2002, pp. 15 - 25.
- [11] Shawn Carpenter, "Analysis and Optimization of a Compact CPW Filter Using Planar EM Software," *Microwave Product Digest*, October 2001, pp. 10-14, 28, 50.
- [12] James C. Rautio, "Making Practical High Frequency Electromagnetic Simulators-- Past, Present, and Future," *IEICE Transactions on Electronics*, Vol. E84-c, No. 7, July 2001, pp. 855-860.
- [13] J. C. Rautio, "An Investigation of Microstrip Conductor Loss," *IEEE Microwave Magazine*, Volume 1, Number 4, December 2000, pp. 60-67.
- [14] Shawn Carpenter, "Break and Interpolate Technique: A Strategy for Fast EM Simulation of Planar Filters," *Microwave Project Digest*, October 2000, pp. 18 - 27, 56 - 58.

- [15] James C. Rautio, "The Impact on Education of Widely Available Commercial 3-D Planar Electromagnetic Software," *Computer Applications in Engineering Education*, Vol. 8, No. 2, September 2000, pp. 51 - 60.
- [16] G. L. Matthaei, J. C. Rautio, and B. A. Willemsen, "Concerning the influence of housing dimensions on the response and design of microstrip filters with parallel-line couplings," *IEEE MTT Transactions*, Vol. 48, August 2000, pp. 1361 –1368.
- [17] James C. Rautio, "Tips and Tricks for Using Sonnet Lite - Free EM software will radically change the way you do high frequency design," *Microwave Product Digest*, November 1999, pp. 30 - 34, 67 - 70.
- [18] James C. Rautio, "An Investigation of Microstrip Conductor Loss," *IEEE MTT Magazine*, December 2000, pp. 60-67. This article is available in the Support Section of the Sonnet website, www.sonnetsoftware.com.
- [19] Shawn Carpenter, "Break and Interpolate Technique: A Strategy for Fast EM Simulation of Planar Filters," *Microwave Project Digest*, October 2000, pp. 18 - 27, 56 - 58.
- [20] James C. Rautio, "The Impact on Education of Widely Available Commercial 3-D Planar Electromagnetic Software," *Computer Applications in Engineering Education*, Vol. 8, No. 2, September 2000, pp. 51 - 60.
- [21] James C. Rautio, "Free EM Software Analyzes Spiral Inductor on Silicon," *Microwaves & RF*, September 1999, pp. 165 - 172.
- [22] Takashi MIURA, Hideki NAKANO, Kohji KOSHIJI and Eimei SHU, "Reduction of time required for electromagnetic analysis by dividing circuit," Faculty of Science and Technology, Science University of Tokyo, Japan Institute of Electronics Packaging, pp.79-80, Mar. 1999. (Article in Japanese)
- [23] James C. Rautio, "Tips and Tricks for Using Sonnet Lite - Free EM software will radically change the way you do high frequency design," *Microwave Product Digest*, November 1999, pp. 30 - 34, 67 - 70.
- [24] James C. Rautio, "EM Simulation," 1999 IEEE MTT-S International Microwave Symposium, Microwave and Millimeter-Wave Design Tool Applications Workshop, Anaheim, CA, June 13, 1999.
- [25] James C. Rautio, "Application of Electromagnetic Analysis Software to 3-D Planar High Frequency Design", *International Multilayer Circuits Symposium (IMCS)*, March 1999, pp. B2-1 - B2-17.

- [26] Shigeki Nakamura, "Top Interview: Electromagnetic Analysis is not Difficult - Big Rush to Install PC Version," *Electronic Products Digest*, Vol. 16, No. 1, January 1999, page 48. (Japanese Article)
- [27] James C. Rautio, "Comments on 'Revisiting Characteristic Impedance and Its Definition of Microstrip Line with a Self-Calibrated 3-D MoM Scheme,'" *IEEE Transactions on Microwave Theory and Techniques*, Vol. 47, No. 1, January 1999, pp. 115 - 117.
- [28] James C. Rautio and George Matthaei, "Tracking Error Sources in HTS Filter Simulations," *Microwaves and RF*, Vol. 37, No. 13, December 1998, pp. 119 - 130.
- [29] J.C. Rautio, "-Electromagnetic Analysis for Microwave Applications," *Computational Electromagnetics and Its Applications*, Vol. 5, Boston: Kluwer Academic Publishers, 1997, pp. 80-96.
- [30] Yasumasa Noguchi, Shin-ichi Nakao, Hideaki Fujimoto and Nobuo Okamoto, "Characteristics of Shielded Coplanar Waveguides on Multilayer Substrates," *Electronic Information and Communications University Meeting, Electronics Society Conference*, June 29, 1998. (Japanese Article)
- [31] Erik H. Lenzing and James C. Rautio, "A Model for Discretization Error in Electromagnetic Analysis of Capacitors," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 46, No. 2, February 1998, pp. 162-166.
- [32] J. C. Rautio, "Retracing Key Moments In the Life of Maxwell," *Microwaves & RF*, Vol. 36, No. 11, November 1997, pp. 35-51.
- [33] J. C. Rautio, "Electromagnetic Analysis for Microwave Applications," *NASA CEM (Computational Electromagnetics) Workshop*, Newport News, VA, May 1996.
- [34] J. C. Rautio, "Seven Years Later," *Applied Microwave and Wireless*, November/December 1996, pp. 99-100.
- [35] J. C. Rautio, "Questionable Reviews," *The Institute (IEEE newspaper)*, Jan. 1996, pg. 11.
- [36] J. C. Rautio, "An Investigation of an Error Cancellation Mechanism with Respect to Subsectional Electromagnetic Analysis Validation," *International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering*, Vol. 6, No. 6, November 1996, pp. 430-435.

- [37] J. C. Rautio, "The Microwave Point of View on Software Validation," IEEE Antennas and Propagation Magazine, Vol. 38, No. 2, April 1996, pp. 68-71.
- [38] J. C. Rautio and Hiroaki Kogure, "EMI Applications Of The Electromagnetic Analysis By The Method Of Moments-Electromagnetic Analysis Applied To Analog And Digital PCB Design," JPCA Show 96 Text: Today and Tomorrow of EMI Design, pp. 11-19.
- [39] J. C. Rautio, "EM-Analysis Error Impacts Microwave Designs," Microwaves and RF, September 1996, pp. 134-144.
- [40] James R. Willhite, "Turning Clean Theory into Reality," Wireless Design and Development, March 1996, Vol. 4, No. 3, pp. 19-20.
- [41] J. C. Rautio, "Response #2. Comments on Zeland's Standard Stripline Benchmark Results - MIC Simulation Column," International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering, Vol. 5, No. 6, November 1995, pp. 415-417.
- [42] J. C. Rautio, "EMI Analysis from a Wireless Telecommunication and RF Perspective," Proceedings of the 1995 Nepcon West Conference, Anaheim, CA, USA, pp. 749-755.
- [43] J. C. Rautio and Hiroaki Kogure, "An Overview of the Sonnet Electromagnetic Analysis," Proceedings of the 1994 IEICE Fall Conference, Tokyo, pp. 325-326.
- [44] J. C. Rautio, "An Ultra-High Precision Benchmark For Validation Of Planar Electromagnetic Analyses," IEEE Tran. Microwave Theory Tech., Vol. 42, No. 11, Nov. 1994, pp. 2046-2050.
- [45] J. C. Rautio, "A Precise Benchmark for Numerical Validation," IEEE International Microwave Symposium, Workshop WSMK Digest, Atlanta, June 1993.
- [46] "Comparison of Strategies for Analysis of Diagonal Structures," Sonnet Application Note 51-02.
- [47] J. C. Rautio, "MIC Simulation Column - A Standard Stripline Benchmark," International Journal of Microwave & Millimeter-Wave Computer-Aided Engineering, Vol. 4, No. 2, April 1994, pp. 209-212.
- [48] J. C. Rautio, "Response #3. Standard Stripline Benchmark - MIC Simulation Column," International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering, Vol. 5, No. 5, September 1995, pp. 365-367.

- [49] J. C. Rautio, "Some Comments on Approximating Radiation," *International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering*, Vol. 4, No. 2, 1994, pp. 454-457.
- [50] J. C. Rautio, "Synthesis of Lumped Models from N-Port Scattering Parameter Data," *IEEE Tran. Microwave Theory Tech.*, Vol. 42, No. 3, March 1994, pp. 535-537.
- [51] J. C. Rautio, "Educational Use of a Microwave Electromagnetic Analysis of 3-D Planar Structures," *Computer Applications in Engineering Education*, Vol. 1, No. 3, 1993, pp. 243-254.
- [52] J. C. Rautio, "Characterization of Electromagnetic Software," 42nd ARFTG Conference Digest, San Jose, CA, Dec. 1993, pp. 81-86.
- [53] J. C. Rautio, "Some Comments on Electromagnetic De-Embedding and Microstrip Characteristic Impedance" *International Journal of Microwave & Millimeter-Wave Computer-Aided Engineering*, Vol. 3, No. 2, April 1993, pp. 151-153.
- [54] J. C. Rautio, "Some Comments on Electromagnetic Dimensionality," *IEEE MTT-S Newsletter*, Winter 1992, pg. 23.
- [55] J. C. Rautio, "Sonnet Software Reveals Tangential Fields," *EESof Wavelengths*, Vol. 9, No. 1, March 1993, pg. 12.
- [56] J. C. Rautio, "Sonnet Introduces Antenna Pattern Visualization in New Release," *EESof Wavelengths*, Vol. 9, No. 2, June 1993, pg. 21.
- [57] J. C. Rautio, "EESof Joins Forces With Sonnet Software," *EESof Wavelengths*, Vol. 8, No. 3, Sept. 1992, pg. 14.
- [58] J. C. Rautio, "Electromagnetic Design of Passive Structures - Emerging Technology in Microwave CAD," *IEEE MTT-S Newsletter*, Fall 1990, pp. 21-22.
- [59] J. C. Rautio, "Electromagnetic Microwave Design," *RF/Microwave Applications Conference*, Santa Clara, CA, March 1992, pp. 105-109.
- [60] J. C. Rautio, "Experimental Validation of Microwave Software," *IEEE International Microwave Symposium, Panel Session PSB Digest*, Albuquerque, June 1992.
- [61] J. C. Rautio, "Current Developments in 3-D Planar Microwave Electromagnetics," *Microwave Hybrid Circuits Conference*, Oct. 1991, Arizona.

- [62] J. C. Rautio, "Current Developments in 3-D Planar Microwave Electromagnetics," Microwave Hybrid Circuits Conference, Oct. 1992, Arizona.
- [63] J. C. Rautio, "Current Developments in 3-D Planar Microwave Electromagnetics," Microwave Hybrid Circuits Conference, Oct. 1993, Arizona.
- [64] J. C. Rautio, "Current Developments in 3-D Planar Microwave Electromagnetics," Microwave Hybrid Circuits Conference, Oct. 1994, Arizona.
- [65] J. C. Rautio, "Experimental Validation of Electromagnetic Software," International Journal of Microwave & Millimeter-Wave Computer-Aided Engineering, Vol. 1, No. 4, Oct. 1991, pp. 379-385.
- [66] J. C. Rautio, "Electromagnetic Microwave Analysis," IEEE International Microwave Symposium, Workshop WSA Digest, Albuquerque, June 1992.
- [67] J. C. Rautio, "EM Visualization Assists Designers," Microwaves and RF, Nov. 1991, pp. 102-106.
- [68] J. C. Rautio, "Reviewing Available EM Simulation Tools," Microwaves & RF, June 1991, pp. 16A-20A.
- [69] "Generating Spice Files Using the *em* Electromagnetic Analysis," Sonnet Application Note 104a, Dec. 1998.
- [70] J. C. Rautio, "A New Definition of Characteristic Impedance," MTT International Symposium Digest, June 1991, Boston, pp. 761-764.
- [71] J. C. Rautio, "A De-Embedding Algorithm for Electromagnetics," International Journal of Microwave & Millimeter-Wave Computer-Aided Engineering, Vol. 1, No. 3, July 1991, pp. 282-287.
- [72] J. C. Rautio, "Triangle Cells in an Electromagnetic Analysis of Arbitrary Microstrip Circuits," MTT International Microwave Symposium Digest, Dallas, June 1990, pp. 701-704.
- [73] J. C. Rautio, "Experimental Validation of Microwave Software," 35th ARFTG Conference Digest, Dallas, May 1990, pp. 58-68. (Voted best paper at the conference.)
- [74] J. C. Rautio, "Preliminary Results of a Time-Harmonic Electromagnetic Analysis of Shielded Microstrip Circuits," 27th ARFTG Conference Digest, Dallas, Dec. 1986. (Voted best paper at the conference.)

- [75] J. C. Rautio, "An Experimental Investigation of the Microstrip Step Discontinuity," IEEE Tran. Microwave Theory Tech., Vol. MTT-37, Nov. 1989, pp. 1816-1818.
- [76] J. C. Rautio, "A Possible Source of Error in On-Wafer Calibration," 34th ARFTG Conference, Ft. Lauderdale, FL, Dec. 1989, pp. 118-126.
- [77] J. C. Rautio, "Microstrip Program Improves Accuracy of Circuit Models," Microwaves & RF, Vol. 27, No. 12, pp. 89-96, Nov. 1988.
- [78] J. C. Rautio, "Reflection Coefficient Analysis of the Effect of Ground on Antenna Patterns," IEEE Antennas and Propagation Society Newsletter, Feb. 87, pp. 5-11.
- [79] J. C. Rautio and R. F. Harrington, "An Electromagnetic Time-Harmonic Analysis of Shielded Microstrip Circuits," IEEE Trans. Microwave Theory Tech., Vol. MTT-35, pp. 726-730, Aug. 1987.
- [80] J. C. Rautio and R. F. Harrington, "An Efficient Electromagnetic Analysis of Arbitrary Microstrip Circuits," MTT International Microwave Symposium Digest, Las Vegas, June 1987, pp. 295-298.
- [81] J. C. Rautio and R. F. Harrington, "Results and Experimental Verification of an Electromagnetic Analysis of Microstrip Circuits," Trans. of The Society for Computer Simulation, Vol. 4, No. 2, pp. 125-156, Apr. 1987.
- [82] J. C. Rautio, "A Time-Harmonic Electromagnetic Analysis of Shielded Microstrip Circuits," Ph. D. Dissertation, Syracuse University, Syracuse, NY, 1986.
- [83] J. C. Rautio, "Preliminary Results of a Time-Harmonic Electromagnetic Analysis of Shielded Microstrip Circuits," ARFTG Conference Digest, Baltimore, pp. 121-134, June 1986. (Voted best paper at the conference.)
- [84] J. C. Rautio, "Techniques for Correcting Scattering Parameter Data of an Imperfectly Terminated Multiport When Measured with a Two-Port Network Analyzer," IEEE Trans. Microwave Theory Tech., Vol. MTT-31, May 1983, pp. 407-412.
- [85] R. Horton, B. Easter, A. Gopinath, "Variation of Microstrip Losses with Thickness of Strip", Electronics Letters, 26th August 1971, Vol. 7, No.17, pp.490-481.

- [86] R. F. Harrington, **Time-Harmonic Electromagnetic Fields**, New York: McGraw-Hill, 1961, section 8-11, pg. 8.

Index

A

- ABS caching level 145
 - subsectioning 146
- ABS see *adaptive band synthesis*
- absolute power 356
- accuracy
 - cell size 21
- adaptive band synthesis 141–154
 - ABS caching level 145
 - analysis issues 151–154
 - current density data 153
 - de-embedding 152
 - multiple box resonances 151
 - optional files 153
 - ripple in S-Parameters 153
 - transmission line parameters 152
 - find maximum 149
 - find minimum 149
 - parameter sweep 149
 - resolution 142
 - running an adaptive sweep 143
 - viewing the response 154
- adaptive sweep 143
- add
 - ports
 - auto-grounded 114
 - standard 101
 - via-ports 112
- adding parameters 188
- additional response files 26
- adjacent polygons 175, 176
- adjustable point set 159, 160
- Agilent Interface 24
- air bridges 301
- analysis
 - limitations 352
- analysis controls 25
- analysis frequencies
 - Microwave Office Interface 237
- Analysis Setup dialog box 145
- anchored parameters 157
 - anchor point 157
 - reference point 157
- Angle Axis 354
- anisotropic dielectric bricks 332
- antennas 343–350
 - surface wave 351
 - wire 351
- APPID 275
- arc conversion angle 274
- arcs 274
- att 189, 199
- att_cascade.son 189
- attenuator 196
- ATTRIBUTES 275
- auto-grounded ports 112, 113, 194
- AWR Interface see *Microwave Office Interface*
- axis
 - Z 352

B

- Bad_instances 285
- balanced ports 103, 379, 380
- batch file 415
- benchmark 401–406
 - accuracy 401
 - residual error 404
- blocks 275
- BOT 289
- boundaries 179
- Box parameters 283
- box resonance 151, 391
- box-wall ports 101
- bricks see *dielectric bricks*
- BRIDGE_LAYER 284, 285, 291

C

- caching 39, 43–45
- CAE software 28
- calculate 351
- calculate memory usage option 408
- calibration lengths 107, 108, 109
- CAP_LAYER 284, 285, 291
- capacitors 184, 333
- cartesian plot 343, 351, 354
- cascading data files 183, 189
- cell size 48, 52, 298, 334, 413
 - accuracy 21
 - determining error 405
 - processing time 179
 - selecting 52–55
 - subsectioning 51, 62
 - wavelength 52
- changes in circuit 25
- circles 274
- circuit analyses 183, 185
- circuit description 274

Circuit elements

- MRINDSBR 284
- MTFC 284
- TFC 284
- circuit properties 216
- circuit subdivision 201–219
 - analysis 218
 - coarse frequency sweep 217
 - executing 218
 - illegal conditions for subdivision lines 215
 - main netlist 205
 - placement of subdivision lines 207–213
 - procedure 206
 - subdivision line orientation 214
 - using parameters 206
- circuit theory simulators 27, 410
- circular dependencies 163
- CKT 192
- closed polygon 274
- coarse frequency sweep 217
- coaxial structures 351
- combined circuit analyses 185
- combining data files 183
- command line 415
- Commands
 - reload_ebridge() 295
- commands 265
- conformal mesh 169–181
 - adjacent polygons 175
 - adjacent polygons with interior vertex 176
 - applying 171
 - cell size 179
 - current density viewer 180
 - current striping 180
 - figure-eight polygons 175
 - Manhattan polygons 177
 - memory save 176
 - polygon boundaries 179

- polygon overlap 174
 - processing time 179
 - rules of use 173
 - transmission lines 171
 - using effectively 177
- conjugate gradient optimizer 166
- consistency 25
- convergence test
 - dielectric bricks 333
- coordinate system
 - Microwave Office Interface 262, 264
 - spherical 352
- coplanar 103, 351
- coupling 202, 208
- cross-talk 361
- current density data 153
 - Microwave Office Interface 241
 - saving in Microwave Office Interface 244
- current density file 351
- current density viewer 18, 68, 399
 - Using from Microwave Office 241
- current striping 180
- cut 310
- cvia 314

D

- data blocks
 - CKT 192
- data consistency 25
- data file elements
 - adding 187
- data files 183
 - cascading 189
- data type number 270
- de-embedding 39, 40, 119–139, 352
 - box-wall ports 123, 125
 - coupled transmission lines 126
 - enabling 121

- example 121
 - guidelines 131
 - higher order modes 139
 - port discontinuities 122
 - reference planes 124, 132, 134
- de-embedding error codes 129
- DEFAULT 282
- deleting
 - vias 310
- dependent parameters 161
- design suite 17
- diagonal fill 299, 334
- dialog boxes
 - Analysis Setup 145
 - Dielectric Layers 91
 - Layer Mapping 268
 - Levels 306
 - Metal Editor 320
 - Metal Properties 172
 - Port Impedance 107
- dielectric 351
 - multiple constants 351
- dielectric bricks 331–341
 - air 334
 - applications 333
 - convergence test 333
 - far field viewer 334
 - limitations 334
 - materials 337, 338
 - subsectioning 333
 - vias 334
 - visibility 336
 - Z-partitions 339
- dielectric layers
 - creating new 92
 - loss parameters 91
- Dielectric Layers dialog box 91
- dielectric library 93

- editing 96
 - setting up 97
 - using a dielectric from 94
- digital interconnect 361
- DIMENSION 275
- DIMSTYLE 275
- directions 351
- directive gain 356
- disable instance check 294
- discontinuity 39
 - lumped models 361
 - port 139
- disk swap 47
- DMAC 401
- dual_patch 111, 348
- DUT 119
- DXF file 265
 - error message 273
- DXF layer name 268
- DXF Translator 19
- DXF translator 24
- dxfgao* 19

E

- \$EXTMAX 275
- \$EXTMIN 275
- ebridge* 19
- edge mesh option 65
- edit
 - cut 310
- editing a circuit 17
- editing in Microwave Office 230
- Eeff 134, 152
- electric fields 397
- electrically thick conductors 79
- Electromagnetic models
 - inserting 284
- em*

- batch file 415
- command line 415
- description 20
- reduce circuit size 48
- speed 47
- theory 22–24, 28
- timing 407
- EM structure 226
 - exporting from Microwave Office 252
- EM structure editor
 - choosing Microwave Office 231
- emgraph* 18
- emstatus* 18
- emvu* 18
- E-plane 353
- error codes
 - de-embedding 129
 - nl 134
- error function 166
- Error in processing the layout 296
- error message 273
- Error while creating hierarchy 296
- error, residual 404
- Errors while launching applications 296
- example files
 - att 199
 - att_cascade.son 189
 - benchmark
 - s100 403
 - s25 403
 - s50 403
 - cvia 314
 - dual_patch 111, 348
 - openmite 299
 - package_resonances 386
 - ste_sym 46
 - steps 41, 45, 185
 - tane 399

- Thkthru 326
- Tripat 350
 - via example 314
- examples
 - att 189
- exceptions 275
 - entities not handled 275
- excitations 351
- exporting an EM Structure 252
- EXT 283, 287
- external layer file 269
- EXTMAX 275
- EXTMIN 275
- F**
- far field 343, 351
- far field viewer 25, 351–357
 - dielectric bricks 334
 - normalization 356
 - polarization 357
 - spherical coordinate system 352
- ferrite components 351
- fields, viewing 397
- figure eight polygons 175
- file
 - DXF 265
- Files
 - bad_instances 285
 - SNP 285
 - valid_instances 285, 291
- fill
 - diagonal 299
- fill type
 - Microwave Office Interface 232
 - setting for a single polygon in MWOOffice 235
 - setting globally in MWOOffice 232
 - setting locally in MWOOffice 234
- filter.gds 270
- Find Maximum 149
- find maximum 149
- Find Minimum 149
- find minimum 149
- FINDMAX 149
- FINDMIN 149
- first frequency 351
- first reference point 160
- flags 275
- frequencies 355
 - default value 351
- frequency
 - maximum 53
- frequency interpolation 202
- frequency sweep
 - coarse 217
- fringing fields 139
- G**
- GABMAC 401
- gain
 - directive 356
 - power 356
 - relative to 356
- gds** 19
- GDSII file
 - corruption of 276
 - error message 276
 - structures 277
 - type 276
- GDSII translator 24
- geometry project 24, 191
- geometry subprojects 204
- goal 167
- graph
 - calculate 351
 - normalization 357

ground via 301, 314

H

header variables 275

highest frequency 53

horizontal orientation 214

H-plane 353

I

IGNORE 288

IGNORE_VALID 294

impedance 106, 107

 surface 83

inductors 184

infinite array 344

install

 Microwave Office Interface 222

instance check disable 294

instance checking 294

interface basics 33

internal ports 312

interpolation 202

isotropic 356

iterations 167

K

Keywords

 EMPORT 290

 IGNORE 288

 NOPORT 288

 NPPORT.ORPHAN 290

 ORPHAN 290

 PORT 290

kinetic inductance 81

L

\$LIMMAX 275

\$LIMMIN 275

lateral dimensions 352

Launch EMVU 281

Launch XGEOM 281, 285

LAYER 285, 288, 291

layer file

 external 269

layer mapping 267, 270

Layer Mapping dialog box 268

layer name

 DXF 268

level 269

level number 267

Levels dialog box 306

Libra 280

library

 dielectric 93

 metal 84

licensing

 Microwave Office Interface 223

limitations 352

LIMMAX 275

LIMMIN 275

Lite 34

loss 24, 74

 box top and bottom 89

 low frequency 79

 related to frequency 79

 vias 89, 311

LOWER_LAYER 291, 292

LTYPE 275

lumped elements 187, 194

 capacitors 184

 inductors 184

 inserting 184

 inserting in a geometry 194

 resistors 184

 transmission lines 184

M

- magnetic wall 46
- main menu
 - accesses 29
- main netlist 205
- map checkbox 269
- mapping 267
- material 269
- maximum frequency 53
- maximum response 149
- maximum subsection size 68
- MCOV 282
- memory available 409
- memory save
 - using with conformal mesh 176
- memory save option 47
- menu bar
 - accesses 29
- Menu options
 - Launch EM 281
 - Launch EMVU 281
 - Launch user window 281
 - Launch XGEOM 281
 - Pick EM filename 281
 - Rep => Geo 281
 - Selected => Geo 281
 - Set EM filename 281
- meshing *See subsectioning*
- Metal Editor dialog box 320
- metal library 84
 - editing 86
 - setting up 88
 - using a metal type from 85
- Metal Properties dialog box 172
- metal type
 - thick metal 317
- metal types
 - Microwave Office Interface 258
- metalization
 - commonly used metals 80
 - loss 24
- metalization levels
 - visibility 306
- metallization loss 74
- metallization thickness 79
- MFC 39
- microstrip 351
- Microwave Office
 - exporting an EM Structure to Sonnet 252
 - project browser 228
 - unlocking 252, 263
 - working outside of 252
- Microwave Office Interface 221–264
 - analysis frequencies 237
 - coordinate system 264
 - coordinate system translation 262
 - current density data 241
 - current density viewer 241
 - editing in MWOOffice 230
 - executing the analysis 247
 - fill type 232
 - grid placement 262
 - install 222
 - licensing 223
 - metal type 258
 - new EM structure 226
 - overview 223
 - port mapping 260
 - response data 248
 - saving current density data 244
 - selecting Sonnet's EM Simulator 226
 - Sonnet features not available 256
 - Sonnet options property pages 239
 - subsectioning 236
 - system requirements 222

- translation issues 255
- uninstall 223
- viewing the subsectioning 245
- minima 168
- minimum response 149
- modeled elements 184
 - adding 187
 - inserting 194
 - inserting in a geometry 194
- modeling assumptions 351
- modes
 - higher order 139
 - TEM 139
- modify
 - attributes
 - ports 106, 109, 115
- MSUB 282
- multi-frequency caching 39, 43–45
- multi-sweep plus Stop/Restart caching 145

N

- native editor in Microwave Office 249
- netlist
 - adding data file elements 187
 - adding network elements 187
 - adding networks 186
 - creating 186
 - including response data 187
 - invoking an analysis of a subproject 191
 - modeled elements 187
 - parameters 188
 - project elements 188
- netlist editor 183
- netlist project 24, 183
 - analysis 185
- network elements
 - adding 187
- networks 184

- adding 186
- new features 33
- nl error code 134
- nominal value 156, 157
 - fixed 163
- non-Manhattan polygons 177
- NOPORT 288
- normalization 356
 - changing 357
 - default 356
 - directive gain 356
 - power gain 356
- normalizing impedances 104
- NUM_X_GRIDS 283, 287
- NUM_Y_GRIDS 283, 287
- number of iterations 167
- numerical precision 407

O

- object type 269
- open environment 352
- openmite 299
- optimization 155–168
 - conjugate gradient method 166
 - methodology 166
 - number of iterations 167
 - process 166
 - results 168
 - specifying goals 167
- optional files 153
- optional output files 26
- options
 - arc conversion angle 274
- orientation of subdivision lines 214
- origin 353
- output files
 - lumped.rsp 197
- overview

Microwave Office Interface 223

P

package resonances 385–395

package_resonances 386

parallel subsections 39, 49

parameter sweep 149, 164

parameterization 155–168

Parameters

box 283

EXT 283

NUM_X_GRIDS 283, 287

NUM_Y_GRIDS 283, 287

substrate 282

parameters 156–166

adding to a netlist 188

adjustable point set 158, 160

adjustable point sets 159

anchor point 157

anchored 157

circular dependencies 163

dependent 161

error conditions 163

in a subproject 206

nominal value 156, 157

ports 107

reference planes 163

reference point 157

select dependents 162

symmetrical 159

patch antenna 343

patch antennas 351

pattern 351

patvu 18

PCSUB 282

PERM 282

phased arrays 343

phi 352, 354

specifying range for calculation 354

plot

cartesian 343, 351, 354

polar 343, 351, 354

surface 343, 351, 354

types 354

POINT 275

polar 354

polar plot 343, 351

polarization 357

polyface mesh 275

polygon edges 208

polygon mesh 275

polygon overlap 174

polygon vias 269, 271

polygons

closed 274

polyline 274, 275

port

excitations 351

terminations 351

port discontinuity 39

Port Impedance dialog box 107

Port instances 296

port mapping

Microwave Office Interface 260

PORT_FILTER 290

ports 99–117

add

auto-grounded 114

standard 101

via-ports 112

auto-grounded 112, 113, 194

calibration lengths 109

reference planes 109

balanced 103, 379, 380

box-wall 101, 123, 125

calibration length 107

- discontinuities 122, 139
- impedance 106, 107
- internal 312
- modify attributes 109, 115
- non-sequential 103
- normalizing impedances 104
- numbering 101, 103
- push-pull 103, 380
- reference planes 107
- renumbering 102
- standard
 - calibration lengths 108
 - reference planes 108
- ungrounded-internal 110, 194, 197
- via 111
- via ports 312
- power gain 356
- precision
 - single 47
- Preference file
 - sonnet.prf 286
- probes 301
- processing time 179
- Product Editor origin 353
- project browser 228
- project editor 17, 24
 - menu bar 29
 - native editor 249
 - netlist editor 183
- project elements
 - adding 188
- project files
 - current density data 25
 - geometry 24
 - netlist 24
- projects 24
- properties
 - circuit 216

- property pages
 - Sonnet option 239
- PSpice file 26
- push-pull ports 103, 380

R

- radiation 343, 343–350
- reactive surface impedance 75
- reference planes 107, 108, 109, 124, 134
 - de-embedding without 132
 - parameters 163
 - short length 133
- reference point 157
 - first in symmetrical 160
 - second in symmetrical 160
- references 357
- release 7.0
 - new features 33
- remove top cover 393, 394
- Rep => Geo 281
- RES index number 288
- residual error 404
- resistance 91
- resistors 184
 - thin film 74, 189
- resonance
 - box 391
- response
 - maximum 149
 - minimum 149
- response data 25, 187
- response files 26
- right-click 162
- ripple in S-parameters 153
- rotation 275
- row/column replication 275
- run options
 - edge mesh 65

memory save 47

S

s100 403

s25 403

s2p 26

s50 403

scaling 275

second reference point 160

sections 204

select dependents 162

Selected => Geo 282

selecting *em*

 Microwave Office Interface 226

sense layer 397

Set EM Filename 281

SHAPE 275

simulation 247

single precision 47

smooth surface 275

SNP 285

snp 26

SOLID 275

Sonnet

 suite 17

Sonnet Lite 34

Sonnet options property page 239

Sonnet project 24

S-Parameters 27

 generalized 104

 ripple in 153

special cases

 another layout program 272

 specifying units 272

Special layers

 BRIDGE_LAYER 284

 CAP_LAYER 284

speed 47

spherical coordinate system 352

spiral inductors 301

SSSUB 282

SSUB 282

SSUBO 282

ste_sym 46

step size 274

steps 41, 45, 185

steradian 356

Stop/Restart caching 145

stream layer number 270

stripline benchmark 402

structure 277

STYLE 275

SUB 282

subdividing 218

subdivision 201–219

 procedure 206

subdivision lines 204

 coupling across 202

 illegal conditions 215

 orientation 214

 placement 202, 207–216

 placement examples 208, 209

subprojects 188, 204

 parameters in 188

subs/lambda 68

subsectional vias 305

subsectioning 21, 48

 ABS caching level 146

 cell size 51, 62

 frequency 146

 Microwave Office Interface 236

 viewing in Microwave Office Interface 245

subsections 51, 68

 maximum 49

 minimum 49

 of polygon 57

- reduce number 48
- vias 305
- XMIN 62
- YMIN 60
- Substrate parameters 282
 - MCOV 282
 - MSUB 282
 - PCSUB 282
 - PERM 282
 - SSSUB 282
 - SSUB 282
 - SSUBO 282
 - SUB 282
 - TAND 282
- superconductor 81
 - surface impedance 83
- surface impedance 83
- surface plot 343, 351, 354
- surface reactance 81
- surface resistance 91
- surface wave antennas 351
- swapping 47
- symmetrical parameter
 - first reference point 160
 - second reference point 160
- symmetrical parameters 159
- symmetry 39, 45, 413
- system requirements
 - Microwave Office Interface 222

T

- 3D structures 275
- 3DFACE 275
- T attenuator 191, 194
 - with ungrounded-internal ports 198
- tables not handled 275
- TAN 282
- tane 399

- tangential electric field 397
- TEM modes 139
- terminations 351
- TEST_VALID 294
- TEXT 275
- theory 22–24, 28
- theta 352, 354
 - specifying range for calculation 354
- thick metal 102, 317–329
 - arbitrary cross-section 324
 - creating a polygon 319
 - number of sheets 317
 - restrictions 324
 - viewing in the current density viewer 327
 - viewing in the project editor 322
- thick metal type 317
- thickness
 - metal 79
- thin film resistor 74, 189
 - two connected 189
- Thkthru 326
- 3-D 301
- timing 407
- tool bar 109, 306
- tool box 101
- TOP 289
- top cover 393, 394
 - resistive 401
- top hemisphere 353
- Touchstone 280
- TRACE 275
- transferring files 276
- translation issues
 - Microwave Office 255
- Translation of
 - entire design 281
 - selected portions of design 281
- Translation procedures 282

Translator preferences 287

EXT 287

LAYER 288

NUM_X_GRIDS 287

NUM_Y_GRIDS 287

translators 24

Agilent 24

DXF 24

GDSII 24

transmission line parameters 152

transmission lines 184

structures 194

triangle subsection 299–300

triangular subsections 352

Tripat 350

troubleshooting

DXF file error message 273

GDSII file error message 276

two-port T attenuator 191

U

UCS 275

ungrounded-internal ports 194, 197

uninstall

Microwave Office Interface 223

units

specifying 272

unlocking Microwave Office 263

UPPER_LAYER 291, 292

user interface 33

V

Valid_instances 285, 291

variables

\$EXTMAX 275

\$EXTMIN 275

\$LIMMAX 275

\$LIMMIN 275

header 275

vertical components 351

vertical orientation 214

via 314

via loss 89

via polygons 269, 271

via ports 111, 312

via-posts 305

vias 269, 271, 301

conical 314

deleting 310

example 314

ground 314

inside dielectric bricks 334

loss 89, 311

restrictions 302

subsections 305

symbol 305

to ground 306

via-posts 305

VIEW 275

view

metalization levels 306

VIEWPORT 275

VPORT 275

W

The "Wall" 410

warnings 199

waveguide simulator 344

wavelength 52

what's new 33

wire antennas 351

wire bonds 301

X

xgeom 17

xgeom level number 267

XMAX 64

XMIN 60, 62

Y

YMAX 64

YMIN 60, 62

Z

Z current 301

Z0 134, 135, 152

Z-axis 352

Z-partitioning 333, 334, 339–341