Empipe3D™

User's Manual

Optimization Systems Associates Inc.

Empipe3D™

User's Manual

Version 4.0

July 1997

Optimization Systems Associates Inc.

LIABILITY AND WARRANTY

NEITHER OPTIMIZATION SYSTEMS ASSOCIATES INC. NOR ITS EMPLOYEES, OFFICERS, DIRECTORS OR ANY OTHER PERSON, COMPANY, AGENCY OR INSTITUTION: (1) MAKES ANY WARRANTY, EXPRESS OR IMPLIED AS TO ANY MATTER WHATSOEVER REGARDING THIS MATERIAL, INCLUDING BUT NOT LIMITED TO THE GENERALITY THEREOF, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR THOSE ARISING BY STATUTE OR OTHERWISE IN LAW OR FROM THE COURSE OF DEALING OR USAGE OF TRADE HAVE BEEN AND ARE HEREBY EXPRESSLY EXCLUDED; OR (2) ASSUMES ANY LEGAL RESPONSIBILITY WHATSOEVER FOR THE ACCURACY, COMPLETENESS OR USEFULNESS OF THIS MATERIAL; OR (3) REPRESENTS THAT ITS USE WOULD NOT INFRINGE UPON PRIVATELY OWNED RIGHTS OF THIRD PARTIES. IT IS EXPRESSLY UNDERSTOOD AND AGREED THAT ANY RISKS, LIABILITIES OR LOSSES ARISING OUT OF ANY USE, TRANSFER OR LEASE OF THIS MATERIAL WILL NOT BE ATTRIBUTED TO OPTIMIZATION SYSTEMS ASSOCIATES INC. OR ANY INDIVIDUAL ASSOCIATED WITH THE COMPANY. ACCURACY, COMPLETENESS OR USEFULNESS FOR ANY APPLICATION SHALL BE DETERMINED INDEPENDENTLY BY THE PARTY UNDERTAKING SUCH AN APPLICATION.

IN NO EVENT WHATSOEVER WILL OPTIMIZATION SYSTEMS ASSOCIATES INC., ITS EMPLOYEES, OFFICERS, DIRECTORS, OR AGENTS BE LIABLE FOR ANY DAMAGES, INCLUDING, BUT WITHOUT LIMITATION, DIRECT, INDIRECT, INCIDENTAL AND CONSEQUENTIAL DAMAGES AND DAMAGES FOR LOST DATA OR PROFITS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS MATERIAL.

CONTENTS ARE SUBJECT TO CHANGE WITHOUT NOTICE.

Copyright

Empipe3D User's Manual Version 4.0 first published in 1997. Printed in Canada.

Optimization Systems Associates Inc.
P.O. Box 8083, Dundas, Ontario
Canada L9H 5E7

Tel 905 628 8228
Fax 905 628 8225

Trademarks of Optimization Systems Associates Inc.

Empipe
Empipe3D
OSA90
OSA90/hope
Datapipe
Geometry Capture
Space Mapping
HarPE

Other Trademarks

UNIX is a registered trademark of AT&T.
X-Windows is a trademark of the Massachusetts Institute of Technology.
SPARCstation, SunOS and Solaris are trademarks of Sun Microsystems, Inc.
HP and HP-UX are registered trademarks of Hewlett-Packard Company.
PostScript is a trademark of Adobe Systems Inc.

Empipe3D™ User's Manual

Table of Contents

6 Geometry Capture

7 OSA90 Environment

8 Empipe3D Database

9 Response Interpolation

Index

1

Installation

1

Installation

1.1 Introduction

The installation of Empipe3D can be done by either a system administrator or a user.

In a multi-user environment, the system administrator should install Empipe3D so that it is accessible to all users. If, on the other hand, Empipe3D is to be used by a single user, then that user can carry out the installation himself/herself.

Sections 1.2 and 1.3 describe the procedure of loading the software. Only the person who performs the installation needs to follow this procedure.

However, *every* user of Empipe3D must follow the instructions provided in Sections 1.4 and 1.5 regarding setting UNIX environment variables and copying the tutorial examples.

Empipe3D drives the 3D electromagnetic simulators Ansoft HFSS from Ansoft Corporation and HP HFSS from Hewlett Packard. You must have a valid installation of Ansoft HFSS or HP HFSS on the same computer as Empipe3D.

If you experience difficulties in the installation of Empipe3D, you can contact our technical support staff at

Tel 905 628 8228
Fax 905 628 8225

Email osa@osacad.com

Home page http://www.osacad.com

1.2 Installing the Software

First, you must allocate a directory in which the Empipe3D program and other associated files will be stored. We will refer to this directory as the OSA Installation Directory.

If you are already using other CAD products from OSA, then the OSA Installation Directory should already exist on your computer. You should find that directory and install Empipe3D within it.

Otherwise, you can create the OSA Installation Directory using the UNIX command:

```
mkdir <OSA Installation Directory>
```

If you are a system administrator installing Empipe3D for multiple users, you should create the OSA Installation Directory from the root or another publicly accessible file system. For example, *<OSA Installation Directory>* can be "`/osa_home`" or "`/usr/osa_home`".

System Administrator: please make sure that the OSA Installation Directory has the "read" and "execute" permission for all Empipe3D users.

If you install Empipe3D from your user account, then the OSA Installation Directory will be created within your home directory. In this case, access to Empipe3D is by default restricted to yourself.

1.3 Loading the Software

Change directory to the OSA Installation Directory (described in Section 1.2):

```
cd <OSA Installation Directory>
```

Then follow the instructions for the specific media in your case.

Sun SPARCstations SunOS 4.x, Floppy Disks

```
bar xvpfZ /dev/rfd0c
```

Sun SPARCstations SunOS 4.x, Tape

```
tar xvpf /dev/rst0
```

Sun SPARCstations Solaris 2.x, Floppy Disks

For Solaris 2.2 and higher, you need to disable the volume manager first. Log in as root and type

```
/etc/init.d/volmgt stop
```

Then load the software in the OSA Installation Directory:

```
cpio -id -H bar -I /dev/rdiskette
```

After the loading is finished, eject the floppy disk:

```
eject floppy
```

For Solaris 2.2 and higher, you may restart the volume manager (as root):

```
/etc/init.d/volmgt start
```

Sun SPARCstations Solaris 2.x, Tape

```
tar xvpf /dev/rmt/0m
```

HP 700 Series, DAT Tape

```
tar xvpf /dev/rmt/0m
```

Listing the Files Loaded

You can visually inspect the files loaded by typing

```
ls -la
```

which lists the files in the OSA Installation Directory.

You should be able to see the following subdirectory names listed:

```
bin
empipe3d_examples
osa90msg
```

There may also be other files if you have other OSA products.

You can also list the files in the "bin" subdirectory:

```
ls -la bin
```

You should see the following executable files among the listing:

```
cmpjt
dat2dbs
dbs2dat
empipe3c
empipe3d
osa90
```

License File

The authorization of using Empipe3D is controlled by an encoded license file. Usually the appropriate license file is included in the package supplied to you.

Under some circumstances, such as when the software expiry date is being extended, you may be required to enter a new key in the license file.

In this case, change directory to the OSA Installation Directory, then type

```
cd osa90msg
mv license.osa license.osa.old
vi license.osa
```

Type "i" to enter the "insert" mode; type the key code *exactly* as provided; press <Esc> to exit the "insert" mode; type ":wq" to write to the file and quit the editor.

1.4 Setting UNIX Environment Variables

Every user of Empipe3D must follow the instructions in this Section before attempting to run Empipe3D for the first time.

You must define a UNIX environment variable OSA_DIR to point to the OSA Installation Directory. If you do not know the name of that directory, please ask the person who carried out the software installation.

You should also modify the PATH environment variable so that you can execute Empipe3D from your own directory.

The UNIX environment variable DISPLAY must also be properly defined. For this you need to know the host name of your computer. To obtain the host name, type

```
uname -n
```

Instructions for csh Users

Include the following lines in the .login file in your home directory:

```
setenv OSA_DIR <OSA Installation Directory>
setenv DISPLAY host_name:0
set path=($path $OSA_DIR/bin)
```

Instructions for sh and ksh Users

Include the following lines in the .profile file in your home directory (HP VUE users should include these lines in the .vueprofile file instead):

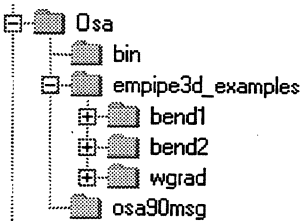
```
OSA_DIR=<OSA Installation Directory>;  export OSA_DIR
DISPLAY=host_name:0;  export DISPLAY
PATH=$PATH:$OSA_DIR/bin;  export PATH
```

Effecting the Environment Changes

After completing the changes to the environment variables, you need to log out and then log in again in order for the changes to take effect.

1.5 Empipe3D Example Directory

The directory hierarchy of the Empipe3D installation is as follows.



OSA Installation Directory
executables
examples
see Chapter 3
see Chapter 4
see Chapter 5
configuration files

We recommend that before using the software, you make a working copy of the examples to a suitable directory and work with that copy.

To copy the Empipe3D examples, first go to your own home directory and then type the UNIX command:

```
cp -r <OSA Installation Directory>/empipe3d_examples .
```

This will create within your home directory an "`empipe3d_examples`" subdirectory.

2

Technical Overview

2

Technical Overview

Empipe3D is a powerful and friendly software system for automated electromagnetic (EM) design optimization, driving the 3D field solvers Ansoft HFSS and HP HFSS.

Ansoft HFSS and HP HFSS are finite element based solvers for full-wave EM field analysis of three-dimensional passive structures. They are capable of computing *S*-parameter responses, electromagnetic field distributions and radiative effects at microwave frequencies.

Empipe3D allows you to designate geometrical and material parameters as candidate variables for optimization in an intuitive and friendly manner. Using the 3D graphical drawing tool within the HFSS package, you create a set of incremental changes to the solid model. The information is then processed by Empipe3D to parameterize the structure using OSA's exclusive Geometry Capture technology.

Empipe3D employs the sophisticated optimizers in OSA90, a program supplied as part of the Empipe3D package. OSA90 offers a comprehensive set of optimizers, including ℓ_1, ℓ_2 (least squares), minimax, Huber, quasi-Newton, conjugate gradient, simplex, simulated annealing and random algorithms, with proven track records in engineering applications. OSA90 also offers a wealth of options for mathematical expressions, response postprocessing, statistical analysis and yield optimization, as well as many graphical display and visualization formats.

Fig. 2.1 illustrates the data flow of the Empipe3D software system.

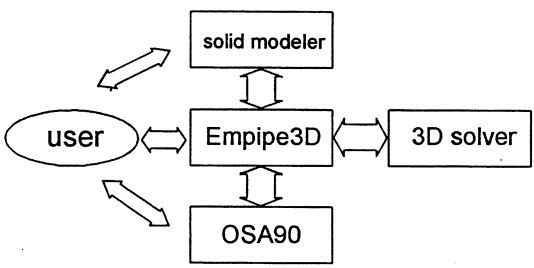


Fig. 2.1 Data flow schematic of the Empipe3D system.

Functional Relationship Between Empipe3D and OSA90

The Empipe3D package includes Empipe3D and OSA90 as two separate executable modules. Their functional relationship is as follows.

As a user, you start the operation with Empipe3D. You define the structure to be optimized as an Empipe3D element. Once the definition of parameters, variables and specifications is completed, you click on a button and OSA90 is invoked. You interact with the OSA90 environment to carry out simulation and optimization. You return to Empipe3D after the optimization is completed.

A series of tutorials is provided in Chapters 3 to 5 to illustrate the various features of Empipe3D. The tutorials are organized in a progressive order, each one introducing something new. We highly recommend that you follow all the tutorials.

Efficient Interpolation and Database

Empipe3D employs parameter discretization, not because it is required by the field solver (HFSS uses variable meshing and therefore does not impose a fixed grid on geometrical dimensions), but in order to improve efficiency during optimization. By applying interpolation to small changes in parameter values within a user-defined interval, the total number of EM field simulations required for optimization can be substantially reduced. This is especially important for gradient calculations.

The user can select from a number of response interpolation schemes, including linear and quadratic interpolations, based on *S, Y* or *Z* parameters.

The *S* parameters from all EM field solutions are saved in an Empipe3D database. This eliminates duplicated EM analyses and, when combined with interpolation, are particularly valuable for making EM-based statistical analysis and yield optimization practical.

Flexibility in Analyzing Large Structures

Empipe3D gives you flexibility in analyzing large, complicated structures. You can decompose a large structure into several substructures, individually simulated by the 3D field solver and then connected via circuit theory. This may produce less accurate results than analyzing the complete structure as a whole, but it can significantly reduce the CPU time needed for the EM analysis.

Optional Equivalent Circuit Model Library

The Empipe3D software system can be upgraded with an optional OSA90/hope module. It offers a comprehensive library of equivalent circuit models, including lumped elements, controlled sources, transmission lines and empirical models for microstrip components.

With this option, you can freely mix within the same circuit EM-simulated structures with equivalent circuit elements. You can use optimization to automate calibration, refinement and new development of models for novel structures.

Nonlinear Harmonic Balance Simulation and Optimization

The optional OSA90/hope upgrade also provides a wealth of nonlinear modeling, simulation and optimization capabilities. It includes a library of nonlinear device models for diodes, FETs, bipolar transistors, HEMTs and user-definable nonlinear controlled sources.

You can perform small-signal, DC and large-signal AC simulation. You can combine EM analysis with harmonic balance analysis to accurately simulate and optimize nonlinear circuits.

3

Tutorial: Bend1

3

Tutorial: Bend1

3.1 Introduction

This tutorial chapter is intended to help you learn to use Empipe3D by going through a simple example step by step. Additional tutorials are provided in Chapters 4 and 5, systematically introducing you to the various features of the program.

Before we proceed, please make sure that the following steps as described in Chapter 1 have been carried out:

1 Empipe3D has been properly installed.

2 You have made a copy of the Empipe3D examples to a working directory.

What You Will Learn From This Tutorial

1 The basics of the user interface of Empipe3D.

2 How to parameterize a geometrical dimension as an optimization variable.

3 How to define a performance specification on the *S* parameters.

4 How to start an EM simulation through Empipe3D.

5 How to start an optimization.

6 How to view and save the optimized solution.

All the EM analysis results necessary for this tutorial have been saved in a database, therefore we can carry out the tutorial without actually invoking the 3D solver.

Generic Reference to the 3D Solvers

We shall use the generic term "3D Solver" in reference to either Ansoft HFSS or HP HFSS.

Description of the Example

For this introductory tutorial, we consider a very simple example: a waveguide bend, as illustrated in Fig. 3.1.

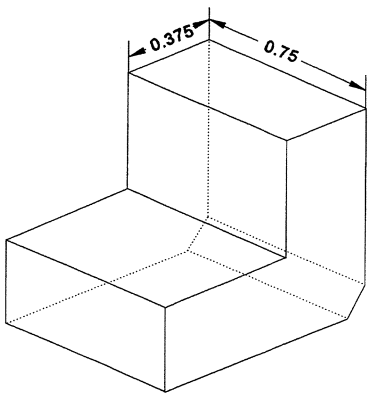


Fig. 3.1 The waveguide bend example.

The objectives of our exercise include:

1 Define the optimizable parameter *d* as shown in Fig. 3.2.

2 Optimize the parameter *d* with respect to the specification

$$20\ log_{10}(|S_{11}|) < -30$$

in the frequency range of 9 to 15 GHz with a frequency step of 1 GHz.

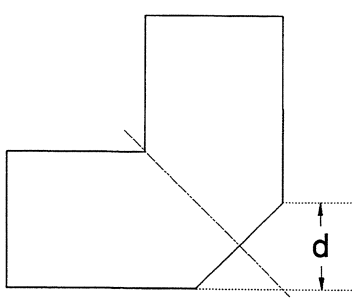


Fig. 3.2 The optimizable parameter d.

3.2 Setting Up The Projects

You should make a copy of the Empipe3D examples to your home directory. The Empipe3D installation directory structure is outlined in Chapter 1.

Go to the directory which contains your copy of the Empipe3D examples:

```
cd empipe3d_examples
```

Then, go to the subdirectory for this tutorial by typing

```
cd bend1
```

This symbol indicates hands-on actions. Please make sure that you follow all the steps indicated by this symbol.

Starting Empipe3D

Type

```
empipe3d bend1
```

The argument "bend1" represents the Empipe3D element name. The Empipe3D window is depicted in Fig. 3.3.

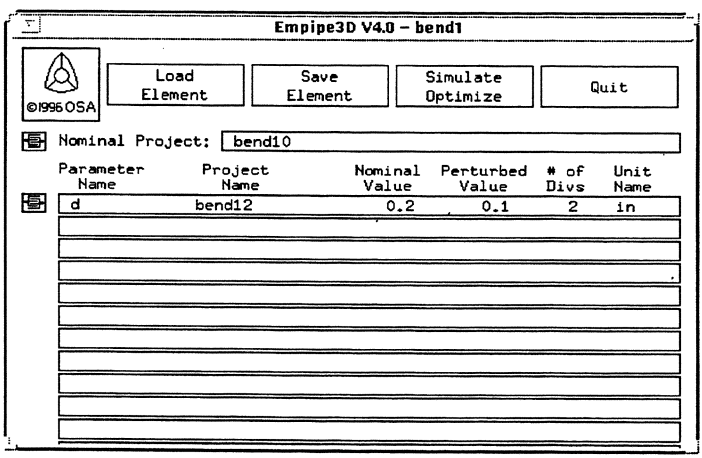
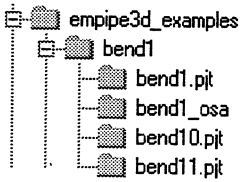


Fig. 3.3 The Empipe3D window.

Project Subdirectories

The project subdirectories relevant to this tutorial are illustrated as follows.



working project automatically created by Empipe3D
parameterization files and database
nominal project
incremental project

The project"bend1.pjt" is used by Empipe3D as a working project, i.e., it is recreated whenever a new EM simulation is required during optimization. The data contained in the subdirectory "bend1_osa" is created and maintained by Empipe3D. Further explanations are given in Chapter 6.

The Nominal Project

First of all, we have to define the waveguide bend structure as a 3D Solver project. We assume that you already know how to create a project using the 3D Solver.

For this tutorial, a project has already been set up for the waveguide bend. It is identified as the nominal project "bend10" in the Empipe3D window, as shown in Fig. 3.3. Following the convention used by the 3D Solver, the extension ".pjt" is implied, i.e., the actual project directory name is "bend10.pjt". The project directory contains the data files required by the 3D Solver.

You can view the solid model for the nominal project by clicking on the  button adjacent to the "Nominal Project" box in the Empipe3D window. Wait for a few seconds and you will see the 3D Solver Solid Modeler window appear on the screen, displaying the waveguide bend structure.

Solid Modeler Invoked by Empipe3D

This button in the Empipe3D window allows you to invoke the 3D Solid Modeler to view a 3D structure.

When the 3D Solid Modeler is invoked from Empipe3D, it will be in the "read-only" mode. You can view the structure but you will not be able to edit it. If you wish to modify the structure, you have to start the 3D Solver directly.

Exit from the Solid Modeler window by clicking on the Solid Modeler menu option "File" and then clicking on the pull-down menu option "Exit".

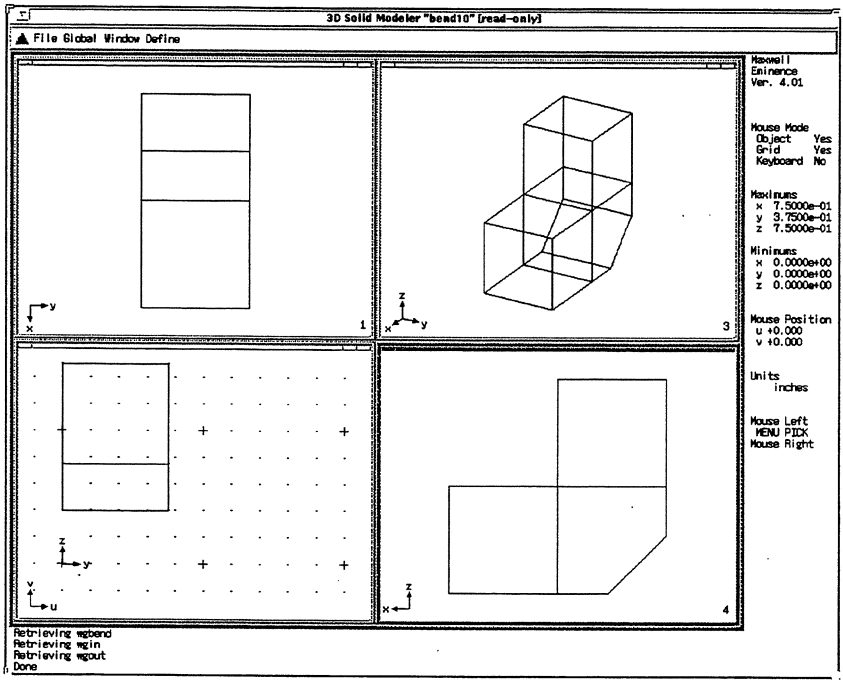


Fig. 3.4 Solid model for the nominal waveguide bend project.

Exploiting Geometrical Symmetry

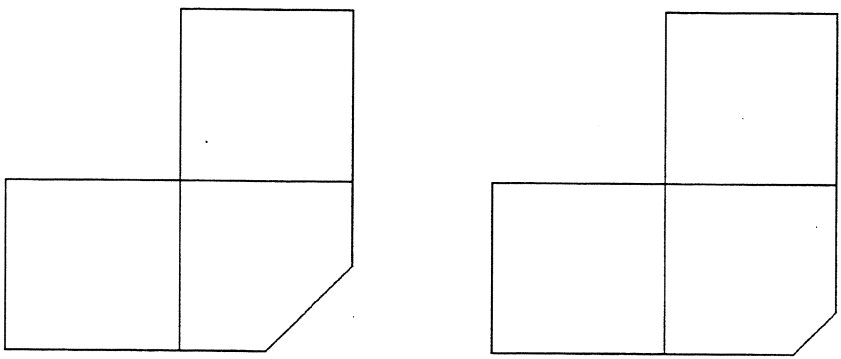
The 3D Solver allows exploitation of geometrical symmetry to reduce computation time. For the waveguide bend structure, we can set up a "Perfect H Boundary" so that only half of the structure needs to be analyzed by the 3D Solver.

The waveguide appears to have a square cross section, since the solid model is one half of the actual structure. The actual waveguide dimension is 0.75 × 0.375 inch.

Creating a New Project to Represent an Incremental Change

In order to parameterize a structure for optimization, we need to create a new project for each parameter to represent an incremental change. By comparing the new project with the nominal project, Empipe3D captures the information necessary for translating parameter values to a corresponding solid model.

For the bend1 tutorial example, we have only one parameter to define, namely d. We need to create a new project to represent an incremental change in the parameter d. The nominal project and the incremental change project represent two distinct values of the parameter d.



nominal project: $d = 0.2$ inch

incremental change: $d = 0.1$ inch

Fig. 3.5 Two distinct values of the parameter d.

Note that these two values of d do not represent the minimum or maximum limit on the value of d (such limits can be defined separately, see Section 3.4). They merely establish a direction of change. The step can be either positive or negative, i.e., the parameter value can be either increased or decreased.

For this tutorial, you do not have to create any projects since all the necessary projects are provided as part of the example. Guidelines for preparing such projects are given in Chapters 4 and 6. Basically, the projects should include the complete setup of material, ports, boundaries and solution parameters. In other words, set up the projects such that they are ready to be solved, without actually solving them.

The number of 3D solid objects must remain the same between the nominal and the incremental change projects. The number of vertices of each object must remain the same. In other words, the basic topology of the structure cannot change. For instance, you can change the size of a rectangle, but you must not change a rectangle into a triangle.

3.3 Geometry Capture Form Editor

Once the nominal and incremental change projects have been set up, we need to enter the information into the Empipe3D form editor. The Empipe3D window is shown in Fig. 3.6.

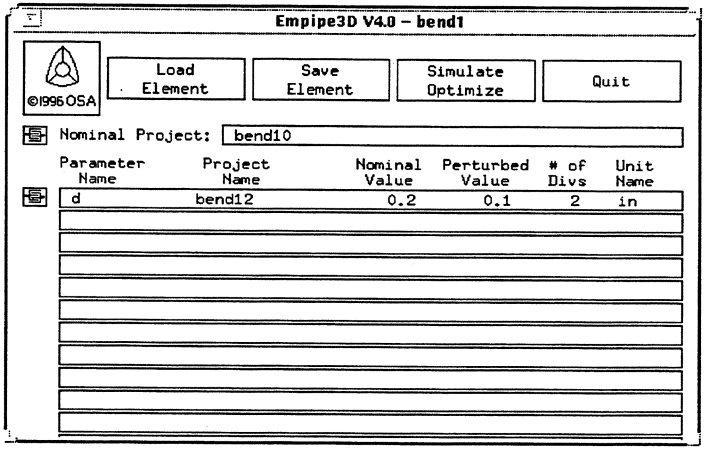


Fig. 3.6 Empipe3D window.

Nominal Project

This entry box identifies the name of the nominal project. For the bend1 tutorial, it is already filled with "bend10". Following the convention used by the 3D Solver, the extension ".pjt" is implied, i.e., the actual project directory name is "bend10.pjt".

Parameter Definition Data

Each entry line under the heading

```
Parameter    Project     Nominal    Perturbed    # of     Unit
  Name         Name       Value      Value      Divs     Name
```

contains the definition of one parameter. Up to 16 parameters can be defined.

For the bend1 tutorial example, we have just one parameter. The entry that defines this parameter is

```
d        bend11        0.2          0.1          2       in
```

These entries are described individually as follows.

Parameter Name is an arbitrary ASCII string of no more than 32 characters. We choose the name "d" to identify the waveguide bend miter parameter.

Project Name identifies the project representing an incremental change in the parameter value with respect to the nominal project. The project that represents an incremental change in the parameter "d" is "bend11" (the project directory is "bend11.pjt").

Nominal Value refers to the value of the parameter represented by the nominal project. For the waveguide bend, the value of "d" in the nominal project is 0.2 inch. The value should be entered as a plain number. The physical unit is entered separately.

Perturbed Value refers to the parameter value after the incremental change. For the waveguide bend, the value of "d" in the incremental change project is 0.1 inch.

Number of Divs means "number of divisions", which specifies the number of interpolation intervals the incremental change encompasses:

$$\textit{Interpolation Interval} = |\ \textit{Perturbed Value} - \textit{Nominal Value}\ |\ /\ \textit{Number of Divs}$$

Empipe3D employs interpolation to improve efficiency. During optimization, the 3D Solver is invoked for an EM simulation only if the parameter has been moved by the optimizer across an interpolation interval. For small changes in parameter values, interpolation is applied instead of EM analysis to obtain the *S* parameters. This can substantially reduce the total number of calls to the 3D Solver required for optimization. This is also an important feature for gradient calculations.

For the parameter "d", the number of divisions is set to 2. Since the incremental change is 0.1 inch, the interpolation interval is 0.1 inch / 2 = 0.05 inch.

A good choice of the interpolation interval is obviously problem dependent. If the interval is too large (i.e., the number of divisions is too small), then the interpolated *S* parameters may not be accurate. On the other hand, a very small interpolation interval will increase the number of 3D EM simulations

and ultimately defeat the purpose of using interpolation.

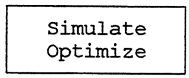
A possible approach to a problem for which you are not sure about the best interpolation interval size is to start with a coarse (large) interval and gradually refine the interval size as necessary. See Chapter 9 for further discussions on the subject of interpolation.

Unit Name identifies the physical unit of the parameter. Recognized unit names include IN (inch), MIL (milli-inch), M (meter), CM (centimeter), MM (millimeter), UM (micron) and NONE (without unit).

This button appears on each line that has a parameter defined. If you click on it, the 3D Solid Modeler will be invoked to display the structure defined by the incremental change project specified on that line.

3.4 Defining Variables and Specifications

In the Empipe3D window, click on the button



Two new windows appear on the screen, entitled "Empipe3D Select Variables" and "Empipe3D Specifications", respectively.

Selecting Optimization Variables

Listed in the "Empipe3D Select Variables" window is the parameter "d" defined for the waveguide bend.

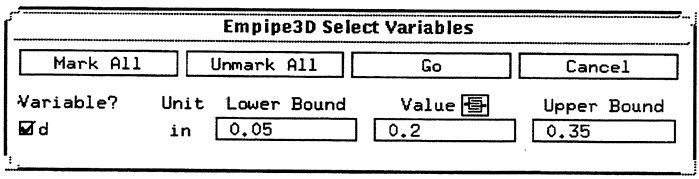


Fig. 3.7 The "Empipe3D Select Variables" window.

In general, all the parameters defined will be listed in the window as candidates for optimization variables. You can select an individual parameter as a variable by clicking on the check mark box associated with that parameter. Alternatively, you can click on the button labelled "Mark All" to select all the parameters as variables.

You can undo the selection of a parameter as a variable by clicking on the check mark box again. You can also use the "Unmark All" button.

Value

The current value for each parameter is shown under the heading "Value". In optimization, this value is used as the starting point. By default, it is set to the parameter's nominal value. If you wish to change it, click on the entry box under "Value" and enter the desired value. You can also click on the button to view the solid model of the current point.

Bounds on the Variables

If a parameter is selected as a variable, you can specify upper and/or lower bound to limit the parameter value range during optimization. Setting suitable bounds on all the variables is advisable since this prevents the optimizer from changing the structure beyond what can be realized physically.

The parameter "d" is assigned the bounds of 0.05 and 0.35 inch.

If the bounds are not given explicitly, they will be determined by the program according to the starting point. Suppose that the starting point of a selected variable is x.

If the lower bound is not given explicitly, it is set to 0 if $x \geq 0$, or $-\infty$ if $x < 0$.

If the upper bound is not given explicitly, it is set to $+\infty$ if $x \geq 0$, or 0 if $x < 0$.

Specifications for Optimization

The initial appearance of the "Empipe3D Specifications" window is depicted in Fig. 3.8.

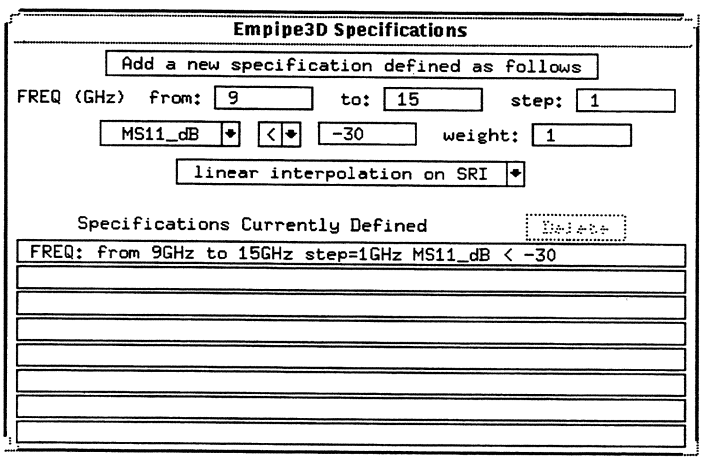


Fig. 3.8 The "Empipe3D Specifications" window.

In general, the definition of a specification involves the following steps:

1 Select a frequency range.

2 Select an *S*-parameter response.

3 Select a specification type (upper, lower or equality specification).

4 Enter a numerical value as the goal.

5 Optionally, enter a weighting factor.

6 Optionally, select an interpolation type.

7 Click on the button "Add a new specification defined as follows".

The specifications currently defined are shown in the lower half of the "Empipe3D Specifications" window. Up to 16 different specifications can be defined.

One specification is defined for the bend1 tutorial example:

$$20\ log_{10}(|S_{11}|) < -30$$

in the frequency range of 9 to 15 GHz with a frequency step of 1 GHz.

Selecting a Frequency Range

The line that appears as

FREQ (GHz) From: [9] to: [15] step: [1]

allows you to select a frequency range for optimization.

The frequency range defined for the nominal project (as part of the 3D Solver "Solution Parameters") is presented as the default.

You can change the start frequency, the end frequency or the frequency step by clicking the entry box and changing the appropriate value.

Multiple specifications over different frequency ranges can also be defined.

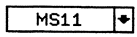
Selecting an S-Parameter Response

The *S*-parameter responses available are represented by these labels:

```
MS11     MS12     ...   MSnn
PS11     PS12     ...   PSnn
MS11_dB  MS12_dB  ...   MSnn_dB
```

where n is the number of ports. MSij represents the magnitude of S_{ij}, PSij represents the phase of S_{ij} in degrees, and MSij_dB represents the magnitude of S_{ij} in decibels.

This is the entry box for selecting an *S*-parameter response:



Click on the arrow and you will see the list of available responses. Click on the label MS11_dB.

Selecting the Type of Specification

The entry box for selecting the type of specification appears as

Click on the arrow and you will see the list of available specification types (upper, lower or equality specification, indicated by the symbol "<", ">" or "=", respectively). Click on the label "<".

Entering a Numerical Value as the Goal

The third entry box on the specification line represents the goal.

Click on the box and then type in the number -30.

Optional Weighting Factor

You can enter an optional weighting factor. The default value is 1. If given, the weighting factor must be a positive number.

Completing the Definition of a Specification

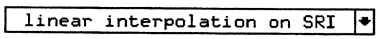
For this tutorial example, the specification has already been predefined. For future projects, once you have completed the selection of the response, the specification type and the goal, you should click on the button labelled "Add a new specification defined as follows". This will add a new line under the heading "Specifications Currently Defined".

Deleting an Existing Specification

To delete an entry under the heading "Specifications Currently Defined", click on the line, this will enable the "Delete" button. Clicking this button will delete the highlighted specification line.

Interpolation Types

The "Empipe3D Specifications" window also includes this entry box



which allows you to select an interpolation type.

For this tutorial example, leave it at the default "linear interpolation on SRI", where "SRI" stands for *S* parameters in the rectangular form ("real and imaginary parts").

The interpolation techniques used by Empipe3D are covered in Chapter 9.

What's Next

We have completed the selection of variables and specifications. Proceed by clicking on the "Go" button in the "Empipe3D Select Variables" window.

3.5 Simulation

After you click on the "Go" button, Empipe3D invokes the OSA90 simulation/optimization environment which has its own window.

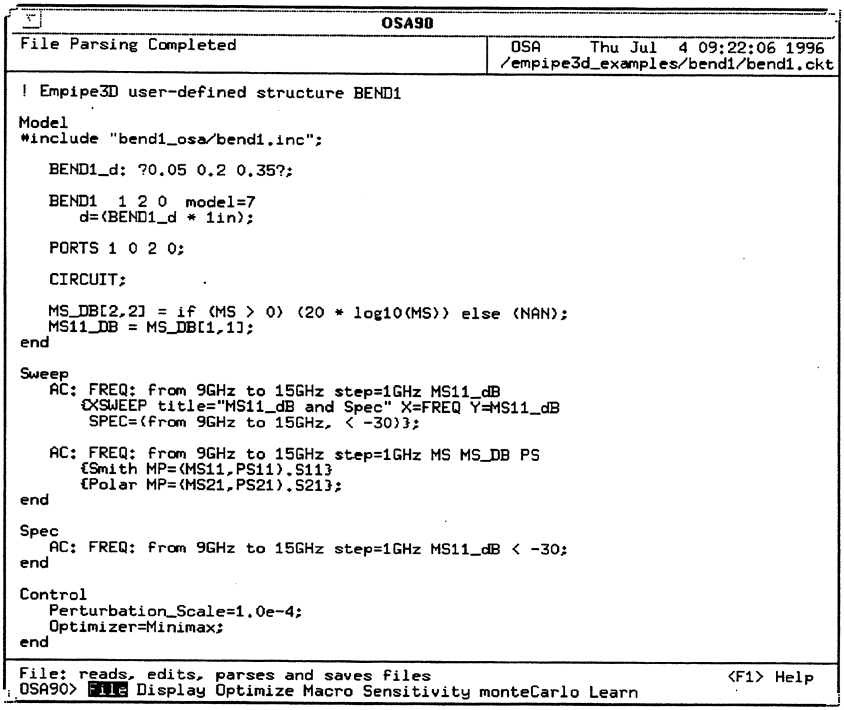


Fig. 3.9 OSA90 window.

At the top of the window is the message area, where the date, time and file name are displayed. At the bottom of the window is the menu area, where the menu options are presented. In the middle is the text area, where the input file (netlist) is shown.

OSA90 Input File (Netlist)

The OSA90 input file (netlist) consists of a number of "blocks". Each block begins with a block identifier, such as "Model", "Sweep", "Spec" and "Control", and ends with the keyword "end". The Model block describes the circuit, the Sweep block selects the simulation outputs, the Spec block defines the specifications for optimization, and the Control block contains one or more operation control options.

In the Model block, notice the statement

```
BEND1_d: ?0.05 0.2 0.35?;
```

The label BEND1_d identifies the parameter d of the element BEND1. The pair of question marks denotes an optimization variable. The values between the question marks represent the lower bound, the starting point and the upper bound.

OSA90 Menu Options

At the bottom of the OSA90 window is a list of the menu options: File, Display, Optimize, etc. You can use the cursor keys or the mouse to move the cursor (pointer) through the menu options. As you do that, the line immediately above the menu shows a brief summary of the function of the highlighted menu option.

To select (activate) a menu option, you simply click the left-hand mouse button on that menu option. You can also move the cursor to highlight the desired option and then press the <Enter> key.

Starting EM Simulation

Before we optimize the bend, we wish to first perform an EM simulation and check the response against the specification.

Click on the menu option Display.

For this tutorial example, the *S* parameters calculated by the 3D Solver have already been saved in a database. If the database does not exist or if this particular set of parameter values is not among those saved in the database, Empipe3D will invoke the 3D Solver automatically.

After the results are retrieved from the database (or, after the 3D Solver analysis is finished) the menu options at the bottom of the window will change to

```
OSA90.Display> Xsweep Parametric ...
```

where the prefix OSA90.Display> indicates that this is the sub-menu under the main menu option Display, and Xsweep, Parametric, ..., represent the various options for displaying the simulation results.

Displaying the Response

Click on the display menu option **Xsweep**. A pop-up window appears, showing the default setting for display. Simply press <Enter> to accept the default setting. The results of the simulation will now be displayed on the screen.

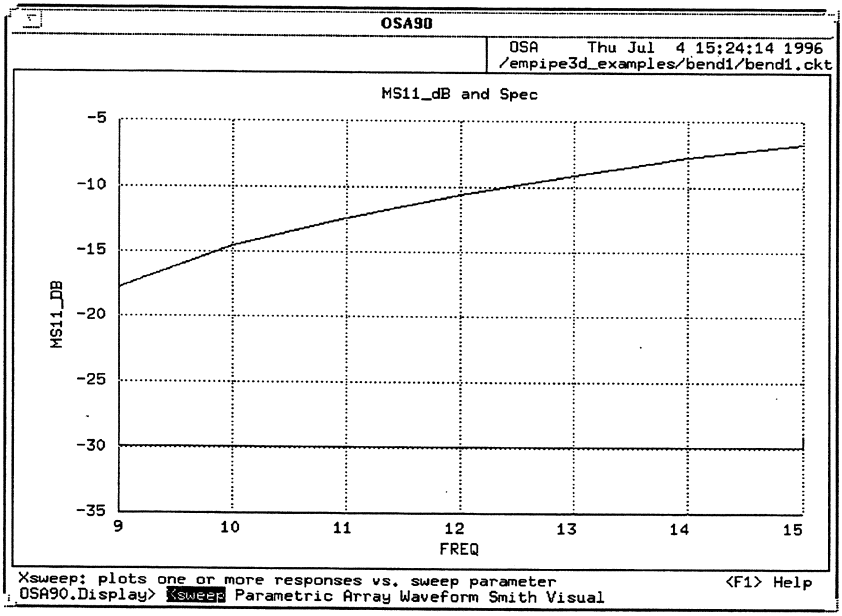


Fig. 3.10 Waveguide bend response before optimization.

Obviously, the specification on **MS11_DB** is not satisfied at the starting point.

Before we proceed to the next section on optimization, we should exit from the display menu. You can do this either by clicking the right-hand mouse button anywhere inside the menu area, or by pressing the <Esc> key. After doing so, you should be back at the main menu

```
OSA90> File Display Optimize Macro ...
```

3.6 Optimization

To start optimization, click on the menu option Optimize. A pop-up window appears, showing a number of options, such as the choice of optimizer, the maximum number of iterations, etc. Press the <Enter> key to accept the default setting.

On the screen, the progress of optimization is reported:

```
Iteration   1/30 Max Error=23.2428
Iteration   2/30 Max Error=22.0957
Iteration   3/30 Max Error=19.2298
Iteration   4/30 Max Error=9.69842
Iteration   5/30 Max Error=19.0108
Iteration   6/30 Max Error=1.45415
Iteration   7/30 Max Error=1.14845
Iteration   8/30 Max Error=1.14157
Solution  Max Error=1.14157
```

The 3D Solver simulation results required for optimizing this tutorial example have been saved in a database, therefore the optimization proceeds rapidly. For new projects, the 3D Solver will be invoked at the starting point and whenever a parameter value is changed by the optimizer out of its interpolation interval.

The optimization of the bend example required 4 simulations by the 3D Solver.

OSA90 offers a variety of optimizers. Empipe3D recommends a suitable optimizer as the default according to the type of specifications you have defined. For the bend example, the minimax optimizer is used. In the display, "Max Error" represents the maximum error, i.e., the worst violation of the specification by the calculated response.

If you wish to select another optimizer, you may do so by using the "Optimize" menu option. Selecting an optimizer from the "Optimize" menu overrides the default setting in the input file.

The numerical values you actually see may be slightly different from those shown here, due to differences in the computer hardware and/or software versions.

Simulation of the Optimized Waveguide Bend

Click on the menu option Display. The *S* parameters of the optimized waveguide bend are retrieved from the database. At the OSA90.Display menu, click on the menu option Xsweep. After the pop-up window appears, press <Enter> to accept the default setting.

The display depicted in Fig. 3.11 shows that the MS11_dB response is greatly improved from that of the starting point.

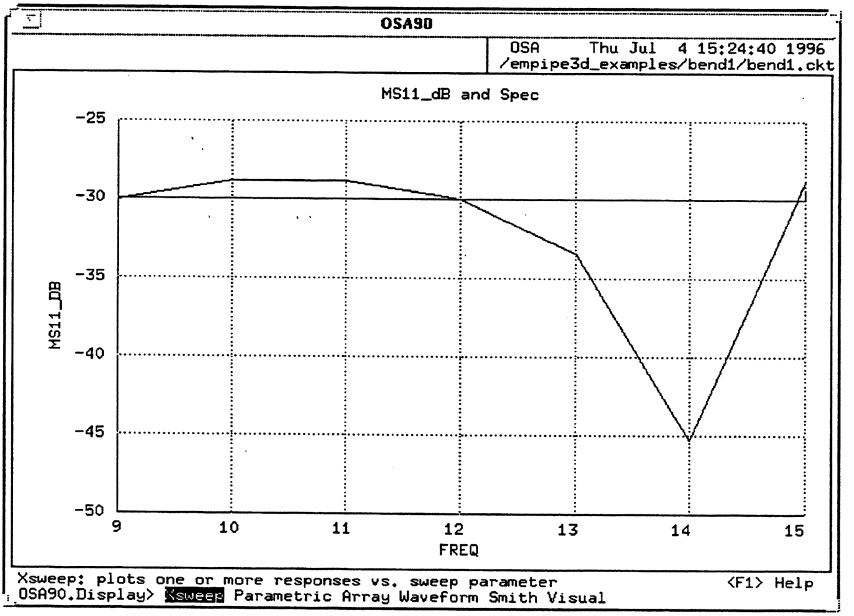


Fig. 3.11 Waveguide bend response after optimization.

The Optimized Parameter Value

Click the right-hand mouse button to exit from the display menu (or press the <ESC> key). At the OSA90 main menu, click the right-hand mouse button (or press the <ESC> key).

You will see this prompt near the bottom of the OSA90 window:

```
Exit from OSA90 (Y/<N>):
```

Click the left-hand mouse button to confirm (or press the <Y> key).

Upon exit from OSA90, the Empipe3D windows reappear on the screen.

The optimized value of the parameter "d" is shown in the "Empipe3D Select Variables" window under the heading "Value".

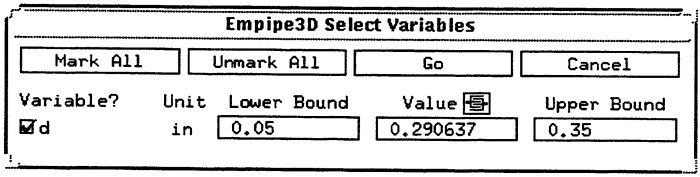
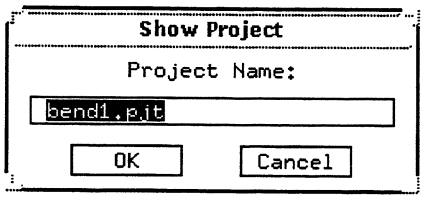


Fig. 3.12 The optimized value of the parameter "d".

Viewing the Optimized Bend

We can use the button above to the "Value" box to invoke the 3D Solid Modeler to view the optimized bend structure.

Then click on the button. A dialog box will appear:



You can accept the default by clicking on "OK". However, the default name "bend1.pjt" is also used by Empipe3D for its working project. Therefore, a project saved under this name may be overwritten by Empipe3D in order to execute another EM analysis. If you wish to save a permanent copy of the optimized project, then you should specify a different project name.

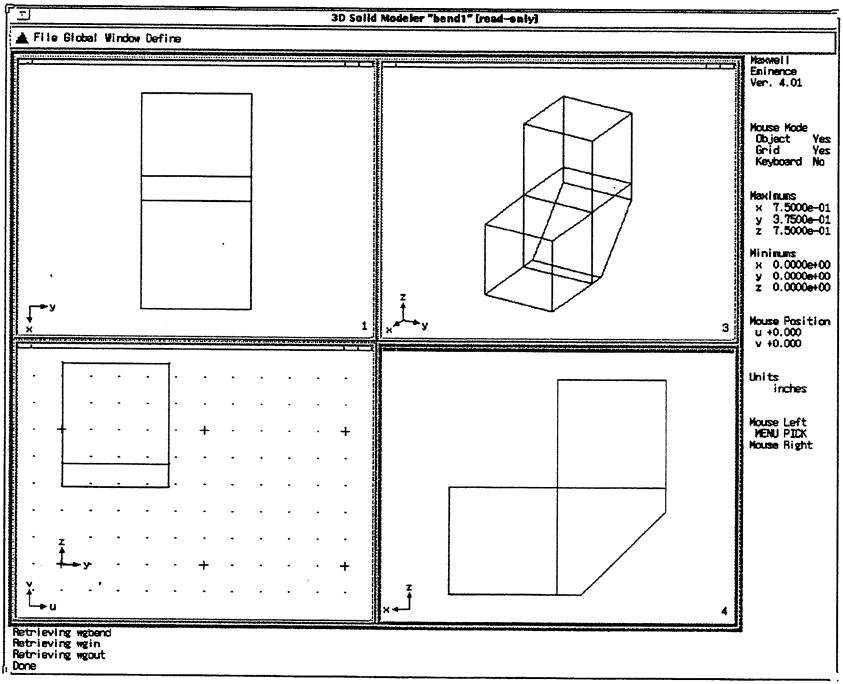


Fig. 3.13 The solid model of the optimized bend.

Finishing the Tutorial

Exit the 3D Solid Modeler by clicking on the menu option "File" and then click on the pull-down menu option "Exit".

Exit the windows for selecting variables and specifications by clicking on the "Cancel" button in the "Empipe3D Select Variables" window.

Exit the Empipe3D main window by clicking on the "Quit" button. You will be prompted whether you wish to save the project file. Click "No" and Empipe3D will exit.

This concludes the first tutorial.

4

Tutorial: Bend2

4

Tutorial: Bend2

4.1 Introduction

This chapter is the second segment of the series of tutorials which systematically introduces you to the various features of Empipe3D.

You should have already gained a basic overview of Empipe3D through the first tutorial in Chapter 3. If you have not, please follow Chapter 3 first.

What You Will Learn From This Tutorial

1 How to parameterize a 3D structure.

2 How to select variables and define specifications.

3 How to start simulation and optimization.

4 How to view Smith Chart and Polar Plot.

5 How to verify the optimized solution.

6 How to generate projects with arbitrary parameter values.

All the EM analysis results necessary for this tutorial have been saved in a database, therefore we can carry out the tutorial without actually invoking the 3D solver.

Description of the Example

This tutorial example is the waveguide bend depicted in Fig. 4.1. It has a different configuration than the bend considered in Chapter 3.

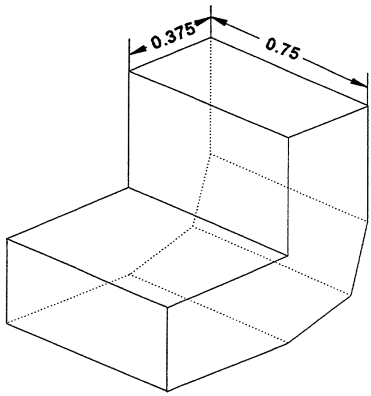


Fig. 4.1 The waveguide bend.

The objectives of our exercise include:

1 Define the optimizable parameter *d* as shown in Fig. 4.2.

2 Optimize the parameter *d* with respect to the specification

$$20\ log_{10}(|S_{11}|) < -40$$

in the frequency range of 9 to 15 GHz with a frequency step of 1 GHz.

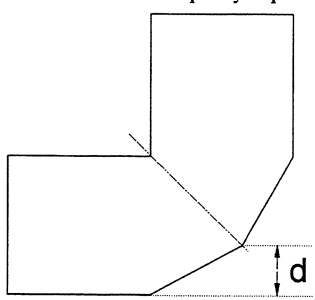


Fig. 4.2 The optimizable parameter d.

4.2 Setting Up The Projects

You should make a copy of the Empipe3D examples to your home directory. The Empipe3D installation directory structure is outlined in Chapter 1.

Go to the directory which contains your copy of the Empipe3D examples:

```
cd empipe3d_examples
```

Then, go to the subdirectory for this tutorial by typing

```
cd bend2
```

This symbol indicates hands-on actions. Please make sure that you follow all the steps indicated by this symbol.

Starting Empipe3D

Type

```
empipe3d bend2
```

The argument "bend2" represents the Empipe3D element name. The Empipe3D window is depicted in Fig. 4.3.

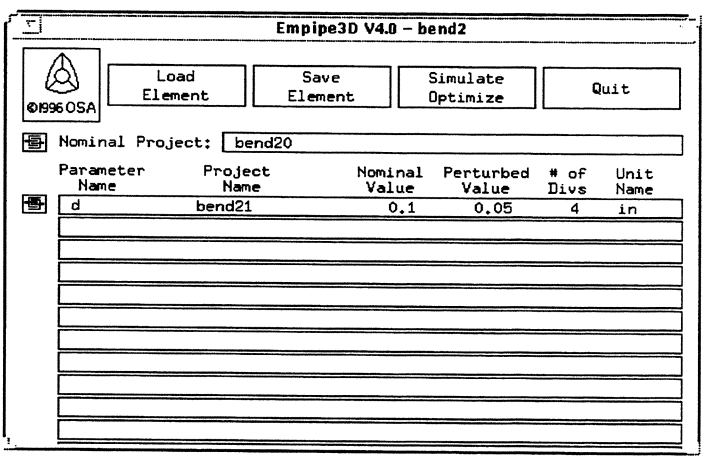
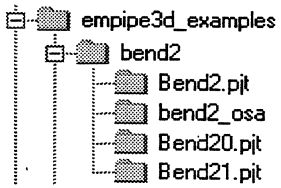


Fig. 4.3 Empipe3D window.

Project Subdirectories

The project subdirectories relevant to this tutorial are illustrated as follows.



The project"bend2.pjt" is used by Empipe3D as a working project, i.e., it is recreated whenever a new EM simulation is required during optimization. The data contained in the subdirectory "bend2_osa" is created and maintained by Empipe3D.

Do not modify the projects for this tutorial. Chapter 6 provides guidelines, rules and hints on how to create projects for a new problem.

The Nominal Project

The nominal project for this tutorial is "bend20", as indicated in the Empipe3D window. It represents the structure at a nominal set of parameter values. There is nothing magic about the nominal project. It can be a design obtained by synthesis, from experience or through optimization of an empirical model. It is by default used as the starting point for optimization by Empipe3D.

You can view the solid model for the nominal project by clicking on the button adjacent to the "Nominal Project" box in the Empipe3D window. Wait for a few seconds and you will see the 3D Solver Solid Modeler window appear on the screen, displaying the waveguide bend structure.

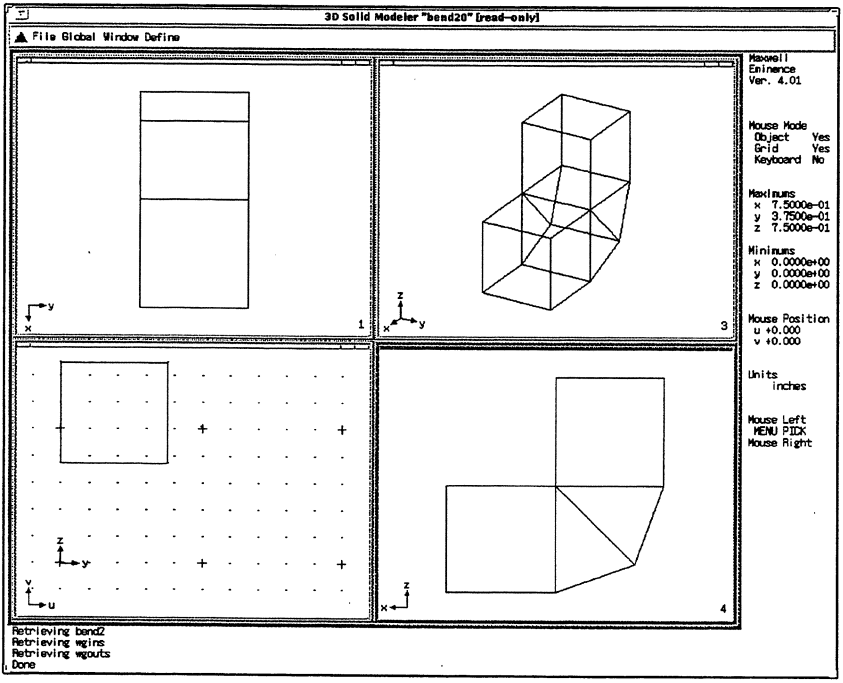


Fig. 4.4 Solid model for the nominal project "bend20".

When you are done viewing the solid model, exit from the Solid Modeler by clicking on the menu option "File" and then clicking on the pull-down menu option "Exit".

Exploiting Geometrical Symmetry

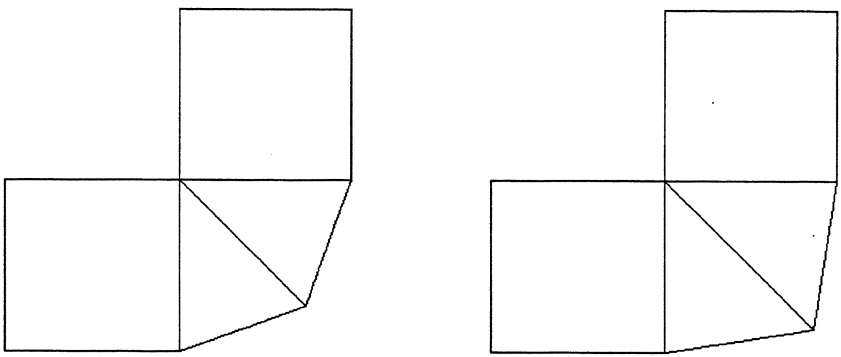
The 3D Solver allows exploitation of geometrical symmetry to reduce computation time. For the waveguide bend structure, we can set up a "Perfect H Boundary" so that only half of the structure needs to be analyzed by the 3D Solver.

The waveguide appears to have a square cross section, since the solid model is one half of the actual structure. The actual waveguide dimension is 0.75 × 0.375 inch, as illustrated in Fig. 4.1.

Creating a New Project to Represent an Incremental Change

In order to parameterize a structure for optimization, we need to create a new project for each parameter to represent an incremental change. By comparing the new project with the nominal project, Empipe3D captures the information necessary for translating parameter values to a corresponding solid model.

For the bend2 tutorial example, we have only one parameter to define, namely d as depicted in Fig. 4.2. We need to create a new project to represent an incremental change in the parameter d. Such a project has already been set up and is provided as part of this example in the directory "bend21.pjt" The nominal project and the incremental change project represent two distinct values of the parameter d.



nominal project: $d = 0.1$ inch

incremental change: $d = 0.05$ inch

Fig. 4.5 Two distinct values of the parameter d.

Note that these two values of d do not represent the minimum or maximum limit on the value of d (such limits can be defined separately, see Section 4.4). They merely establish a direction of change. The step can be either positive or negative, i.e., the parameter value can be either increased or decreased.

A simple and intuitive way to verify that the Geometry Capture projects are correctly implemented and processed, is to ask Empipe3D to generate parameterized projects and visually inspect the results. This is described in Section 4.8.

4.3 Geometry Capture Form Editor

Once the nominal and incremental change projects have been set up, we need to enter the information into the Empipe3D form editor.

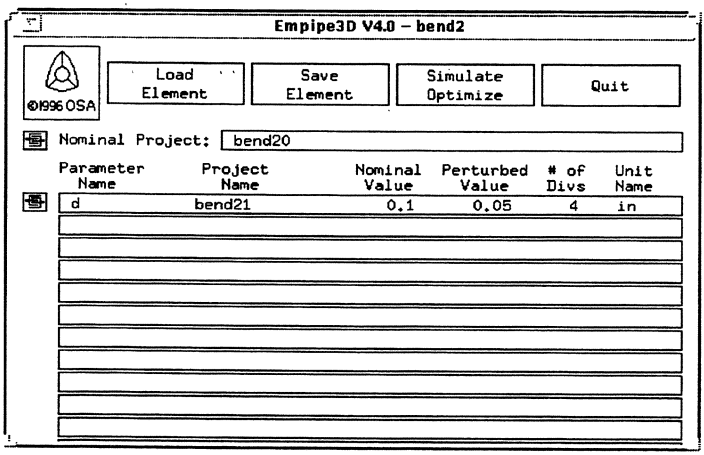


Fig. 4.6 Empipe3D window.

Nominal Project

This entry box identifies the name of the nominal project. For the bend2 tutorial, it is already filled with "bend20". Following the convention used by the 3D Solver, the extension ".pjt" is implied, i.e., the actual project directory name is "bend20.pjt".

Parameter Definition Data

Each entry line under the heading

Parameter Name	Project Name	Nominal Value	Perturbed Value	# of Divs	Unit Name

contains the definition of one parameter. Up to 16 parameters can be defined.

For the bend2 tutorial example, we have just one parameter. The entry that defines this parameter is

d	bend21	0.1	0.05	4	in

These entries are described individually as follows.

Parameter Name is an arbitrary ASCII string of no more than 32 characters. We choose the name "d" to identify the parameter depicted in Fig. 4.2.

Project Name identifies the project representing an incremental change in the parameter value with respect to the nominal project. The project that represents an incremental change in the parameter "d" is "bend21" (the project directory is "bend21.pjt").

Nominal Value refers to the value of the parameter represented by the nominal project. For the waveguide bend, the value of "d" in the nominal project is 0.1 inch (see Fig. 4.5). The value should be entered as a plain number. The physical unit is entered separately.

Perturbed Value refers to the parameter value after the incremental change. The value of "d" in the incremental change project is 0.05 inch (see Fig. 4.5).

Number of Divs means "number of divisions", which specifies the number of interpolation intervals the incremental change encompasses:

$$\textit{Interpolation Interval} = |\textit{Perturbed Value} - \textit{Nominal Value}| / \textit{Number of Divs}$$

Empipe3D employs interpolation to improve efficiency. During optimization, the 3D Solver is invoked for an EM simulation only if the parameter has been moved by the optimizer across an interpolation interval. For small changes in parameter values, interpolation is applied instead of EM analysis to obtain the *S* parameters. This can substantially reduce the total number of calls to the 3D Solver required for optimization. This is also an important feature for gradient calculations.

For the parameter "d", the number of divisions is set to 4. Since the incremental change is 0.05 inch, the interpolation interval is 0.05 inch / 4 = 0.0125 inch.

A good choice of the interpolation interval is obviously problem dependent. If the interval is too large (i.e., the number of divisions is too small), then the interpolated *S* parameters may not be accurate. On the other hand, a very small interpolation interval will increase the number of 3D EM simulations and ultimately defeat the purpose of using interpolation.

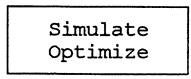
A possible approach to a problem for which you are not sure about the best interpolation interval size is to start with a coarse (large) interval and gradually refine the interval size as necessary. See Chapter 9 for further discussions on the subject of interpolation.

Unit Name identifies the physical unit of the parameter. Recognized unit names include IN (inch), MIL (milli-inch), M (meter), CM (centimeter), MM (millimeter), UM (micron) and NONE (without unit).

This button appears on each line that has a parameter defined. If you click on it, the 3D Solid Modeler will be invoked to display the structure defined by the incremental change project specified on that line.

4.4 Defining Variables and Specifications

In the Empipe3D window, click on the button



Two new windows appear on the screen, entitled "Empipe3D Select Variables" and "Empipe3D Specifications", respectively.

In the first tutorial of Chapter 3, the variables and specifications have all been predefined. This information is purposely left blank for this tutorial, so that you can fill in the data as an exercise.

Selecting Optimization Variables

Listed in the "Empipe3D Select Variables" window is the parameter "d" defined for the bend2 example.

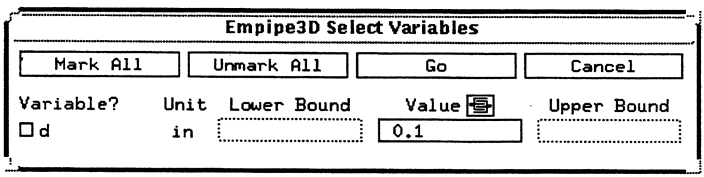


Fig. 4.7 The initial "Empipe3D Select Variables" window.

In general, all the parameters defined will be listed in the window as candidates for optimization variables. You can select an individual parameter as a variable by clicking on the check mark box associated with that parameter. Alternatively, you can click on the button labelled "Mark All" to select all the parameters as variables.

You can undo the selection of a parameter as a variable by clicking on the check mark box again. You can also use the "Unmark All" button.

Click on the blank square check box under the heading "Variable?" for the parameter "d". It should now have a check mark.

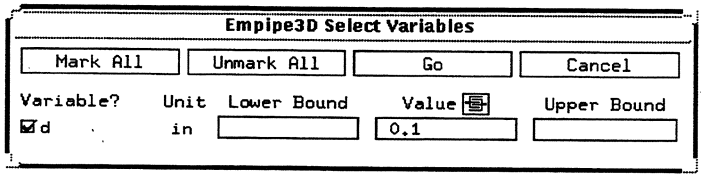


Fig. 4.8 The parameter "d" selected as a variable.

Value

The current value for each parameter is shown under the heading "Value". By default, it is set to the parameter's nominal value. If you wish to change it, click on the entry box under "Value" and enter the desired value. You can also click on the button to view the solid model of the current point.

Bounds on the Variables

You can specify upper and/or lower bound to limit the parameter value range during optimization. Setting suitable bounds on all the variables is advisable since this prevents the optimizer from changing the structure beyond what can be realized physically.

Click on the blank box under the heading "LowerBound" and type "0.025". Then, click on the blank box under the heading "UpperBound" and type "0.35". The entered values appear in the "Empipe3D Select Variables" window.

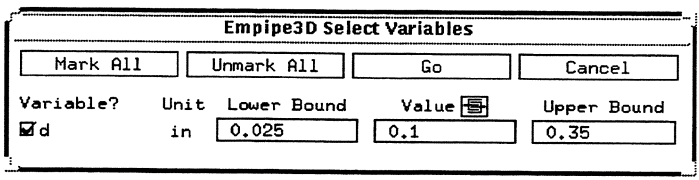


Fig. 4.9 The bounds on the variable "d".

If the bounds are not given explicitly, they will be determined by the program according to the starting point. Suppose that the starting point of a selected variable is x.

If the lower bound is not given explicitly, it is set to 0 if $x \geq 0$, or $-\infty$ if $x < 0$.

If the upper bound is not given explicitly, it is set to $+\infty$ if $x \geq 0$, or 0 if $x < 0$.

Specifications for Optimization

The initial appearance of the "Empipe3D Specifications" window is depicted in Fig. 4.10.

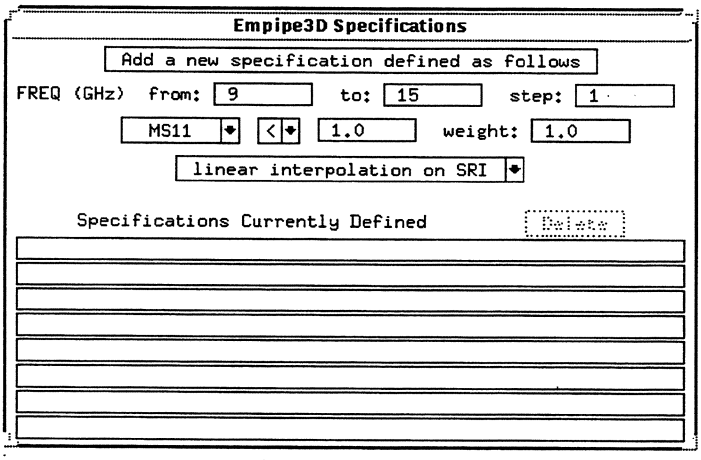


Fig. 4.10 The initial "Empipe3D Specifications" window.

In general, the definition of a specification involves the following steps:

1 Select a frequency range.

2 Select an *S*-parameter response.

3 Select a specification type (upper, lower or equality specification).

4 Enter a numerical value as the goal.

5 Optionally, enter a weighting factor.

6 Optionally, select an interpolation type.

7 Click on the button "Add a new specification defined as follows".

Selecting a Frequency Range

The line that appears as

FREQ (GHz) from: [9] to: [15] step: [1]

allows you to select a frequency range for optimization.

The frequency range defined for the nominal project (as part of the 3D Solver "Solution Parameters") is presented as the default.

The bend2 tutorial example has the following specification:

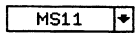
$$20\ log_{10}(|S_{11}|) < -40$$

in the frequency range of 9 to 15 GHz with a frequency step of 1 GHz.

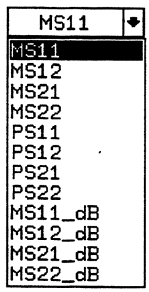
For this example, the default frequency range is fine. In general, you can change the start frequency, the end frequency or the frequency step by clicking on the appropriate entry box and then typing the new value.

Selecting an S-Parameter Response

This is the entry box for selecting an *S*-parameter response:



Click on the arrow and you will see the list of available responses.



The label MS*ij* represents the magnitude of S_{ij}, PS*ij* represents the phase of S_{ij} in degrees, and MS*ij*_dB represents the magnitude of S_{ij} in decibels.

Click on the label MS11_dB.

Selecting the Type of Specification

The entry box for selecting the type of specification appears as

Click on the arrow and you will see the list of available specification types.

The symbols "<", ">" and "=" represent upper, lower and equality specifications, respectively.

Click on the symbol "<".

Entering a Numerical Value as the Goal

The third entry box on the specification line represents the goal.

Click on the box and then type in the number -40.

Optional Weighting Factor

You can enter an optional weighting factor. The default value is 1. If given, the weighting factor must be a positive number. For this tutorial, leave the weighting factor at its default setting.

Completing the Definition of a Specification

Click on the button labelled "Add a new specification defined as follows". A new specification line is added to the lower half of the window under the heading "Specifications Currently Defined".

For this tutorial, we have only one specification to define. In general, you can define multiple specifications with different frequency ranges, responses and goals. Up to 16 specification lines can be defined.

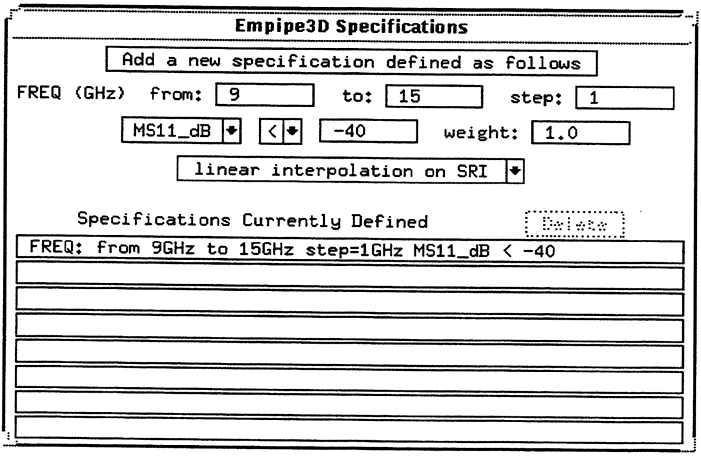


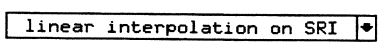
Fig. 4.11 The newly defined specification for the bend2 tutorial.

Deleting an Existing Specification

To delete an entry under the heading "Specifications Currently Defined", click on the line. The button "Delete" will turn from a shade of grey to a solid color ("clickable"). A click on this button will delete the highlighted specification line.

Interpolation Types

The "Empipe3D Specifications" window also includes this entry box



which allows you to select an interpolation type.

The interpolation techniques used by Empipe3D are covered in Chapter 9.

What's Next

We have completed the selection of variables and specifications. Proceed by clicking on the "Go" button in the "Empipe3D Select Variables" window.

4.5 Simulation

After you click on the "Go" button, Empipe3D invokes the OSA90 simulation/optimization environment which has its own window.

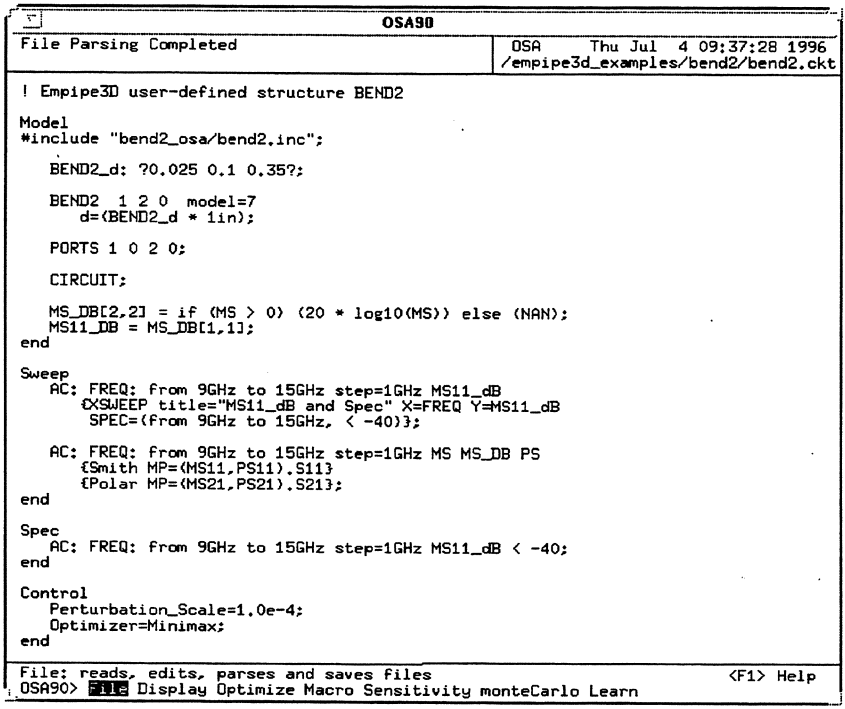


Fig. 4.12 OSA90 window.

At the top of the window is the message area, where the date, time and file name are displayed. At the bottom of the window is the menu area, where the menu options are presented. In the middle is the text area, where the input file (netlist) is shown.

OSA90 Input File (Netlist)

The OSA90 input file (netlist) consists of a number of "blocks". Each block begins with a block identifier, such as "Model", "Sweep", "Spec" and "Control", and ends with the keyword "end". The Model block describes the circuit, the Sweep block selects the simulation outputs, the Spec block defines the specifications for optimization, and the Control block contains one or more operation control options.

In the Model block, notice the statement

```
BEND2_d: ?0.025 0.1 0.35?;
```

The label BEND2_d identifies the parameter d of the element BEND2. The pair of question marks denotes an optimization variable. The values between the question marks represent the lower bound, the starting point and the upper bound.

OSA90 Menu Options

At the bottom of the OSA90 window is a list of the menu options: File, Display, Optimize, etc. You can use the cursor keys or the mouse to move the cursor (pointer) through the menu options. As you do that, the line immediately above the menu shows a brief summary of the function of the highlighted menu option.

To select (activate) a menu option, you simply click the left-hand mouse button on that menu option. You can also move the cursor to highlight the desired option and then press the <Enter> key.

Starting EM Simulation

Before we optimize the bend, we wish to first perform an EM simulation and check the response against the specification. The options listed on the "Display" menu represent the various options for displaying the simulation results.

Click on the menu option Display.

For this tutorial example, the *S* parameters calculated by the 3D Solver have already been saved in a database. If the database does not exist or if this particular set of parameter values is not among those saved in the database, Empipe3D will invoke the 3D Solver automatically.

After the results are retrieved from the database (or, after the 3D Solver analysis is finished) the menu options at the bottom of the window will change to

```
OSA90.Display> Xsweep Parametric ...
```

where the prefix OSA90.Display> indicates that this is the sub-menu under the main menu option Display, and Xsweep, Parametric, ..., represent the various options for displaying the simulation results.

Displaying the Response

Click on the display menu option **Xsweep**. A pop-up window appears, showing the default setting for display. Simply press <Enter> to accept the default setting. The screen display is illustrated in Fig. 4.13.

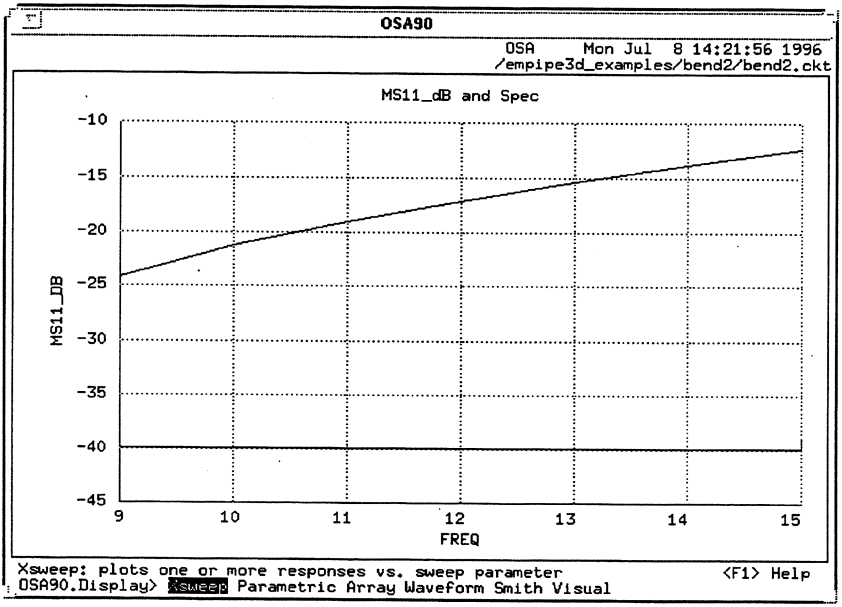


Fig. 4.13 Waveguide bend response before optimization.

The specification on MS11_DB is clearly not satisfied at the starting point.

Viewing the Smith Chart

Click on the display menu option Smith. A pop-up window appears, showing the default setting for display. Simply press <Enter> to accept the default setting. The screen display is illustrated in Fig. 4.14.

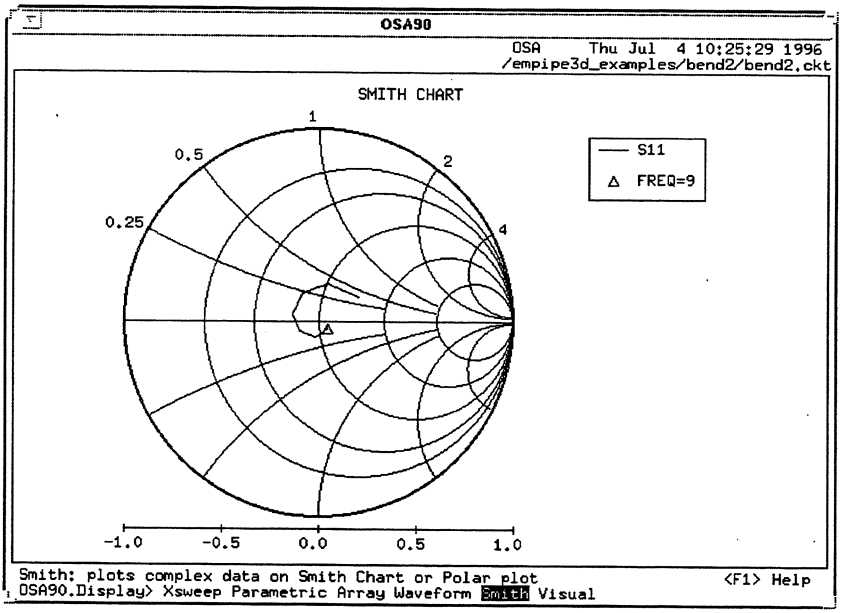


Fig. 4.14 Waveguide bend S_{11} response on the Smith Chart.

Viewing Polar Plot

Click on the display menu option Smith. A pop-up window appears:

```
Output form:    Graphical
Sweep set:      sweep set 1
View:           view_1
Ready to go:

<ENTER> = go  or  <ESC> = cancel
Select item with <UP>/<DOWN>
<F1> help
```

Click on the line "View: view_1". A list of two choices appears: "view_1" and "view_2". Click on "view_2". Then, in the pop-up window click on the line "Ready to go". A Polar Plot will then be displayed.

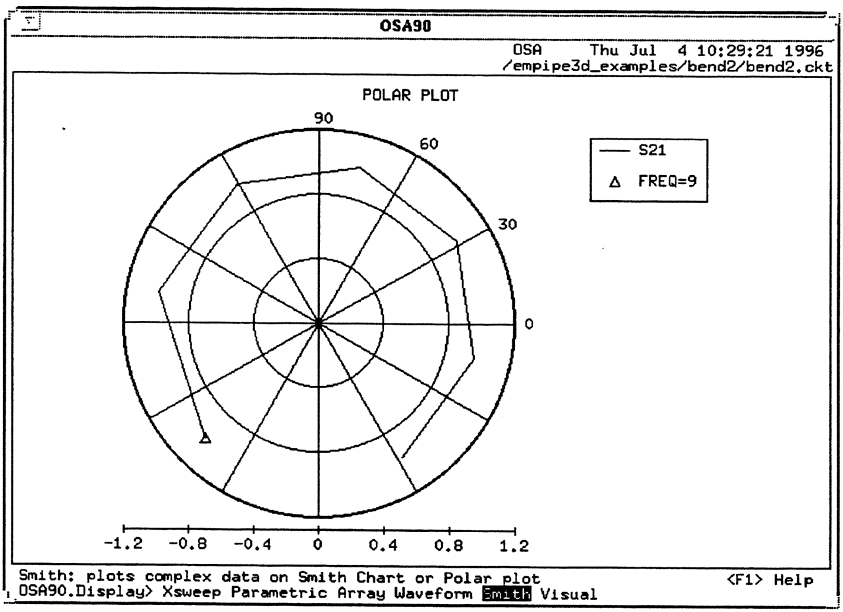


Fig. 4.15 Waveguide bend S_{21} response on the Polar Plot.

4.6 Optimization

Exit from the display menu by pressing the <Esc> key. You are back at the main menu:

```
OSA90> File Display Optimize Macro ...
```

Click on the menu option Optimize. A pop-up window appears, showing the options related to optimization:

```
Optimizer:                Minimax
objective Function:       Minimax
Number of iterations:     30
Accuracy of solution:     0.0001
Display option:           every iteration
Ready to go:

<ENTER> = go  or  <ESC> = cancel
Select item with <UP>/<DOWN>
<F1> help
```

Click on the line "Ready to go:" (or press <Enter>) to accept the default settings. By default, Empipe3D selects an optimizer suitable for the type of specifications you have defined.

On the screen, the progress of optimization is reported:

```
Iteration   1/30 Max Error=27.6123
Iteration   2/30 Max Error=24.2567
Iteration   3/30 Max Error=5.62896
Iteration   4/30 Max Error=27.7546
Iteration   5/30 Max Error=14.416
Iteration   6/30 Max Error=-4.11578
Iteration   7/30 Max Error=-1.44774
Iteration   8/30 Max Error=-4.82617
Iteration   9/30 Max Error=-5.59967
Iteration  10/30 Max Error=-5.86199
Iteration  11/30 Max Error=-5.87395
Solution  Max Error=-5.87395
```

The 3D Solver simulation results required for optimizing this tutorial example have been saved in a database, therefore the optimization proceeds rapidly. For new projects, the 3D Solver will be invoked at the starting point and whenever a parameter value is changed by the optimizer out of its interpolation interval.

The optimization of the bend2 example required 7 simulations by the 3D Solver.

In the display, "Max Error" represents the maximum error, i.e., the worst violation of the specification

by the calculated response.

Simulation of the Optimized Waveguide Bend

Click on the menu option **Display**. The *S* parameters of the optimized waveguide bend are retrieved from the database. At the **OSA90.Display** menu, click on the menu option **Xsweep**. After the pop-up window appears, press **<Enter>** to accept the default setting.

The display shows that the optimized **MS11_dB** response satisfies the specification.

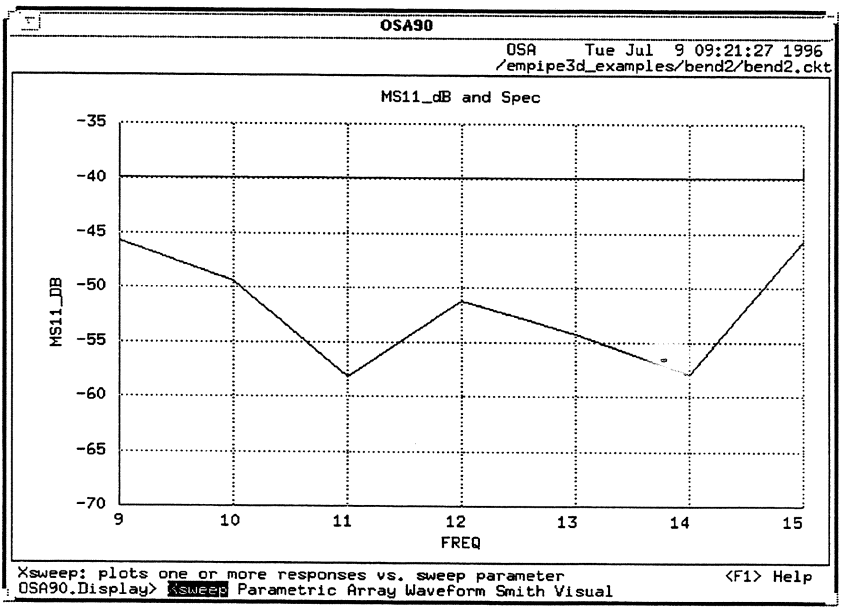


Fig. 4.16 Waveguide bend response after optimization.

Viewing the Optimized Response on the Smith Chart

Click on the display menu option Smith. A pop-up window appears, showing the default setting for display. Simply press <Enter> to accept the default setting. The optimized S_{11} is displayed on the Smith Chart, as depicted in Fig. 4.17.

Since the optimized S_{11} is nearly zero, the curve on the Smith Chart appears to be reduced to a single point.

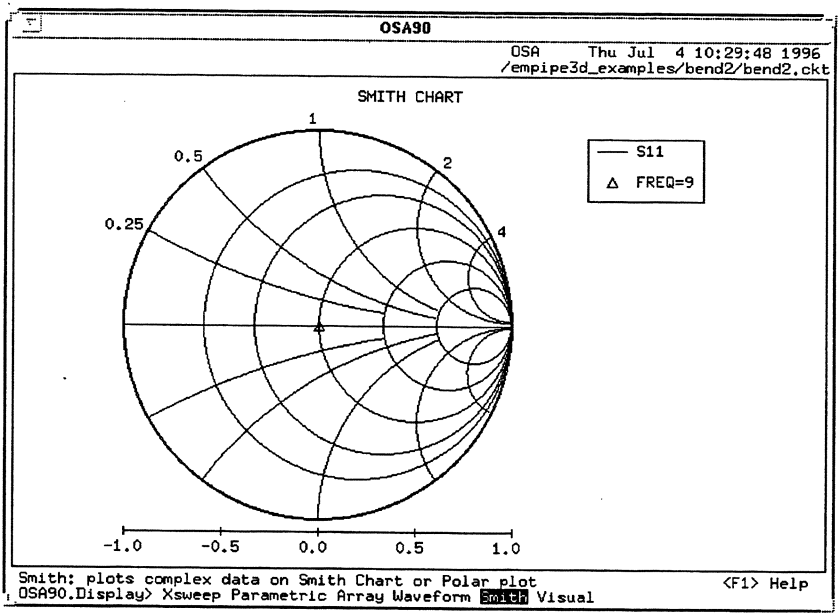


Fig. 4.17 The optimized S_{11} response on the Smith Chart.

4.7 Verifying the Optimized Response

Empipe3D employs interpolation techniques to reduce the number of 3D Solver simulations required for optimization. This leads to the question: Can we trust the optimized response calculated by Empipe3D, since it is likely the result of interpolation?

You can verify the optimized response by turning off the interpolation feature and force Empipe3D to invoke the 3D Solver to simulate the optimized structure.

This section illustrates how this can be done.

Exit from the display menu by clicking the right-hand mouse button or press the <ESC> key. You return to the OSA90 main menu:

```
OSA90> File Display Optimize Macro ...
```

Click on the option "File"..

OSA90 File Editor

You are now within the OSA90 file editor. You can move around the text of the input file (netlist) using the cursor keys and the mouse.

Near the top of the file, you can find the following text:

```
BEND2_d: ?0.025 0.133166 0.35?;

BEND2   1 2 0   model=7
   d=(BEND2_d * 1in);
```

The label "BEND2_d" represents the parameter *"d "* of the "BEND2" structure. Within the pair of question marks, the initial value has been replaced by the optimized value: 0.133166.

The phrase "model=7" contains the flag for interpolation. The number "7" represents the current choice of linear interpolation on *S* parameters in rectangular form (see Chapter 9 for details on the interpolation techniques).

Recall that the interpolation interval of the parameter "*d*" has been defined as 0.0125 (Section 4.3). The optimized value 0.133166 falls within the interval of [0.125 0.1375]. Accordingly, Empipe3D invokes the 3D Solver to compute the *S* parameters for $d = 0.125$ and $d = 0.1375$, and then uses interpolation to obtain the *S* parameters for $d = 0.133166$.

To turn off the interpolation feature, change "model=7" to "model=0" so that the line reads

```
BEND2   1 2 0   model=0
```

The Response Calculated Without Interpolation

Click the left-hand mouse button anywhere within the file editor window. A menu box appears on the screen. Click on the option "Exit from editor". Now you are back at the OSA90 main menu. Click on the Display menu option.

The *S* parameters of the optimized bend computed directly by the 3D Solver have also been saved in the database (if you really wish to see the actual simulation process by the 3D Solver and are willing to wait for such, see Chapter 8 for instructions on how to bypass the database mechanism).

The response of the optimized bend computed without the use of interpolation illustrates that the interpolation actually worked very well. The differences between the responses calculated with and without interpolation mostly occur below -50 dB, which is within the error tolerance of *S*-parameter calculation by the 3D Solver (the noise floor of the 3D Solver is -70 dB according to the User's Manual).

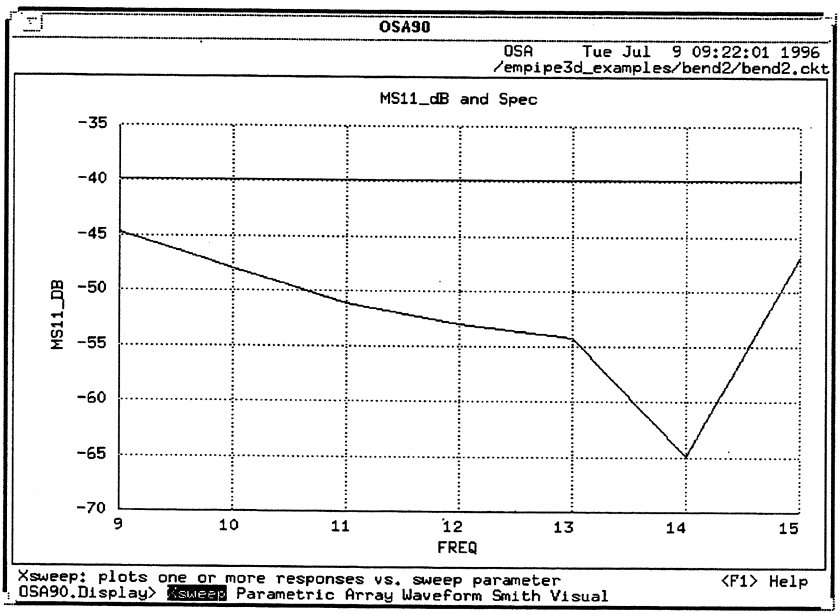


Fig. 4.18 The optimized response calculated without interpolation.

4.8 Generating Projects Using Empipe3D

One of the time-honored practices of engineering is tweaking a design in its final stage by experienced hands. This often involves changing one or two parameters for a number of trials until a satisfactory result emerges.

Imagine that you wish to tweak a design using the 3D Solver. Besides having to wait for the EM field analysis, another tedious task is to redraw the solid model for each trial point. With Empipe3D, this task can be simplified. Once the structure has been parameterized, you can ask Empipe3D to generate new 3D Solver projects for arbitrary parameter values at one click of a button.

Generating a Project for the Optimized Bend

Click the right-hand mouse button to exit from the display menu (or press the <ESC> key). At the OSA90 main menu, click the right-hand mouse button (or press the <ESC> key). You will be prompted:

```
Exit from OSA90 (Y/<N>):
```

Click the left-hand mouse button to confirm (or press the <Y> key).

The "Empipe3D Select Variables" window reappears on the screen.

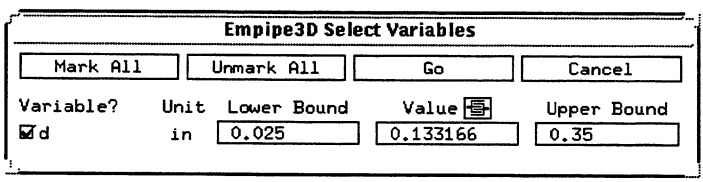
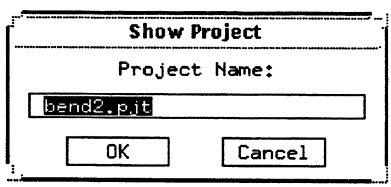


Fig. 4.19 The "Empipe3D Select Variables" window.

Click on the button. A dialog box appears:



You can accept the default by clicking on "OK". However, the default name "bend2.pjt" is also used by Empipe3D for its working project. Therefore, a project saved under this name may be overwritten by Empipe3D in order to execute another EM analysis. If you wish to save a permanent copy of the optimized project, then you should specify a different project name.

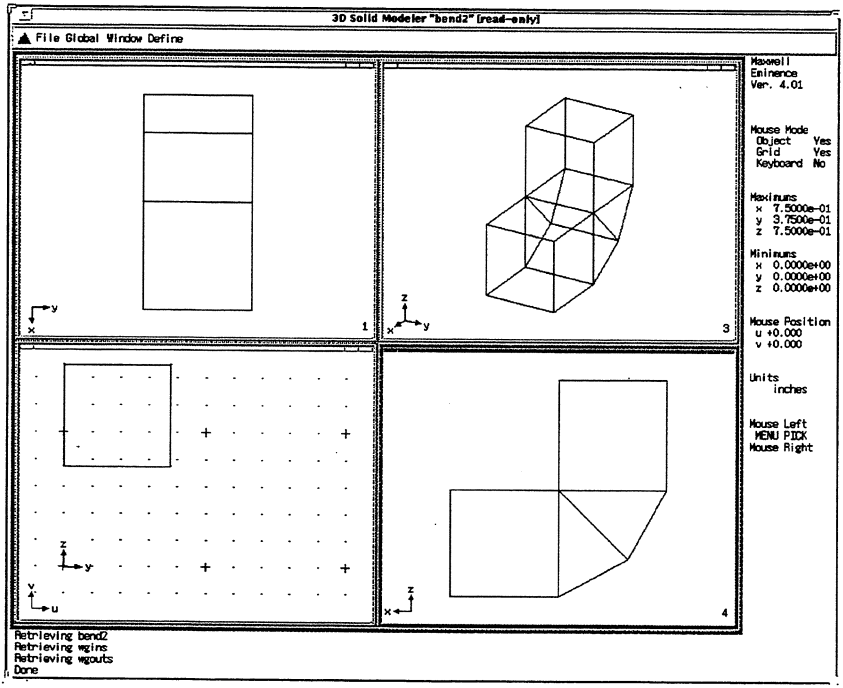
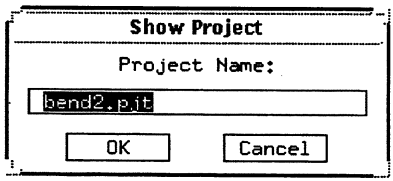


Fig. 4.20 The solid model of the optimized bend.

Exit the 3D Solid Modeler by clicking on the menu option "File" and then clicking on the pull-down menu option "Exit".

Generating Projects with Arbitrary Parameter Values

Click on the entry box under the heading "Value", and type in an arbitrary value, for example "0.2". Then click on the button. A dialog box appears:



You can accept the default name for the project directory or enter a different name of your choice. The solid model of the new project is displayed.

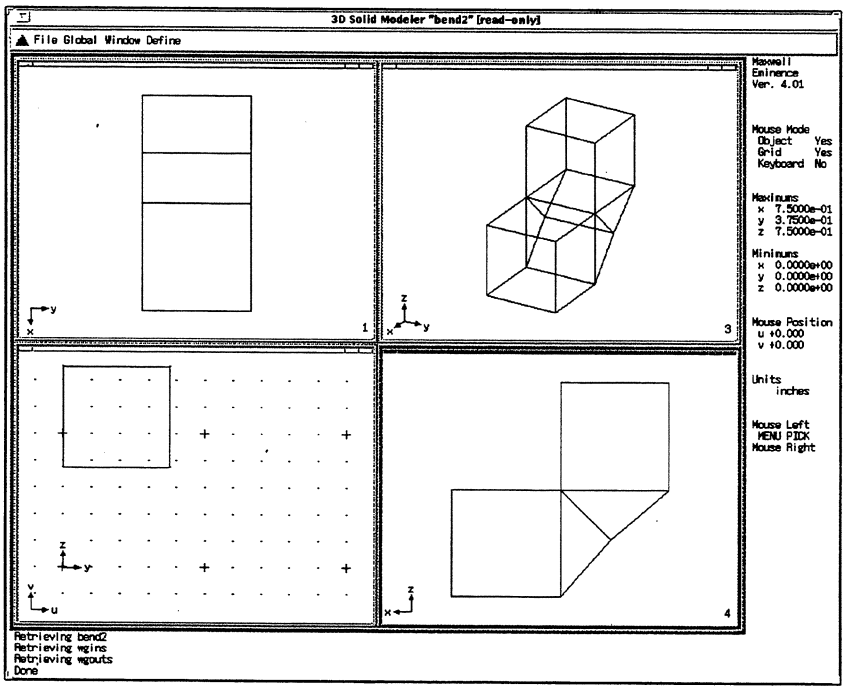


Fig. 4.21 The solid model for d = 0.2 inch.

Validating the Result of Geometry Capture

Generating new projects for arbitrary parameter values can also be used as a means of validating the result of Geometry Capture.

As stated in Section 4.3, a set of rules has to be followed in preparing the nominal and incremental projects for Geometry Capture. If the rules are violated, the parameterization will produce scrambled results.

One way to check this is to ask Empipe3D to derive a new project from the parameterization data. You can enter a new set of parameter values (new in the sense that it is different from the parameter values of the nominal and incremental change projects) in the box or boxes under the heading "Value". Then click on the button. Once the solid model is displayed on the screen, perform a visual inspection to make sure that it is consistent with the parameter values you have entered. You can try this one parameter at a time, which makes it easier to spot any inconsistency.

Finishing the Tutorial

Exit the 3D Solid Modeler by clicking on the menu option "File" and then click on the pull-down menu option "Exit".

Restore the starting point to its original value by clicking on the entry box under the heading "Value" and then type "0.1".

Exit the windows for selecting variables and specifications by clicking on the "Cancel" button in the "Empipe3D Select Variables" window.

Exit the Empipe3D main window by clicking on the "Quit" button. You will be prompted whether you wish to save the file. Click "No" and Empipe3D will exit.

This concludes the second tutorial.

5

Tutorial: Impedance Transformer

5

Tutorial: Impedance Transformer

5.1 Introduction

This chapter is the third segment of the series of tutorials which systematically introduces you to the various features of Empipe3D.

You are strongly encouraged to follow the first two tutorials in Chapters 3 and 4, if you have not already done so, before continuing with this tutorial.

What You Will Learn From This Tutorial

1 Geometry Capture of a 3D structure with multiple parameters.

2 Verifying the parameterization result by visual inspection.

3 How to manipulate the various display options.

All the EM analysis results necessary for this tutorial have been saved in a database, therefore we can carry out the tutorial without actually invoking the 3D solver.

Description of the Example

This tutorial example is a waveguide impedance transformer as depicted in Fig. 5.1.

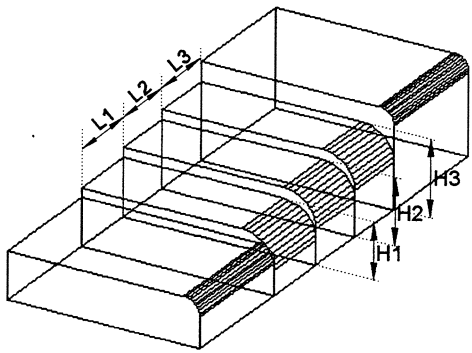


Fig. 5.1 The waveguide impedance transformer.

This is a three-section transformer designed for impedance matching between a WR-75 half height and a WR-75 full height waveguides.

All the sections have identical width of 0.75 inch. The heights and lengths of the three sections will be captured as optimizable parameters, as indicated in Fig. 5.1.

In Fig. 5.1, only the upper-right quadrant of the structure is shown. We exploit the geometrical symmetry of the transformer by setting up both a "Perfect H Boundary" and a "Perfect E Boundary" so that only a quarter of the structure needs to be analyzed by the 3D Solver.

This model explicitly takes into account the round corners of the waveguides. This is different from the typical waveguide models with perfect rectangular cross sections. Taking the corner radius into account should lead to a more accurate simulation. We hope that a relatively large corner radius can be accommodated in order to reduce machining cost without violating the design specification.

Including the corner radius in the model makes it unsuitable for mode matching simulators and necessitates the use of a finite element solver.

The design specification is

$$20\ log_{10}(|S_{11}|) < -30$$

in the frequency range of 9.5 to 15 GHz with a frequency step of 0.5 GHz.

5.2 Projects for Geometry Capture

You should make a copy of the Empipe3D examples to your home directory. The Empipe3D installation directory structure is outlined in Chapter 1.

Go to the directory which contains your copy of the Empipe3D examples:

```
cd empipe3d_examples
```

Then, go to the subdirectory for this tutorial by typing

```
cd wgrad
```

This symbol indicates hands-on actions. Please make sure that you follow all the steps indicated by this symbol.

Starting Empipe3D

Type

```
empipe3d wgrad
```

The argument "wgrad" represents the Empipe3D element name. The Empipe3D window is depicted in Fig. 5.2.

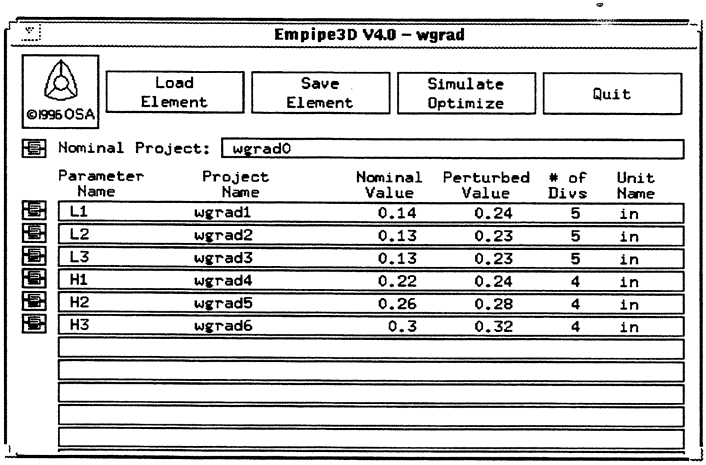
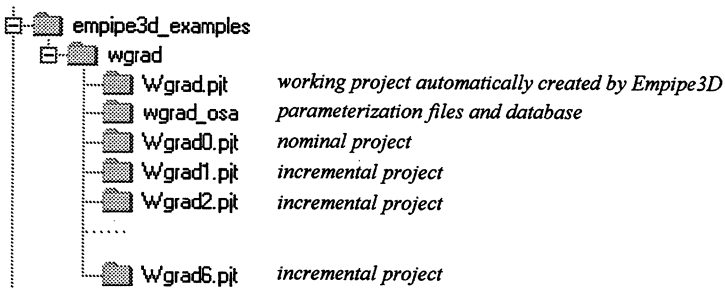


Fig. 5.2 Empipe3D window.

Project Subdirectories

The project subdirectories relevant to this tutorial are illustrated as follows.



The project"wgrad.pjt" is used by Empipe3D as a working project, i.e., it is recreated whenever a new EM simulation is required during optimization. The data contained in the subdirectory "wgrad_osa" is created and maintained by Empipe3D.

Do not modify the projects for this tutorial. Chapter 6 provides guidelines, rules and hints on how to create projects for a new problem.

The Nominal Project

The nominal project for the wgrad tutorial is "wgrad0", as indicated in the Empipe3D window. The project directory name is "wgrad0.pjt".

You can view the solid model for the nominal project by clicking on the button adjacent to the "Nominal Project" box in the Empipe3D window. Wait for a few seconds and you will see the 3D Solver Solid Modeler window appear on the screen, displaying the waveguide bend structure.

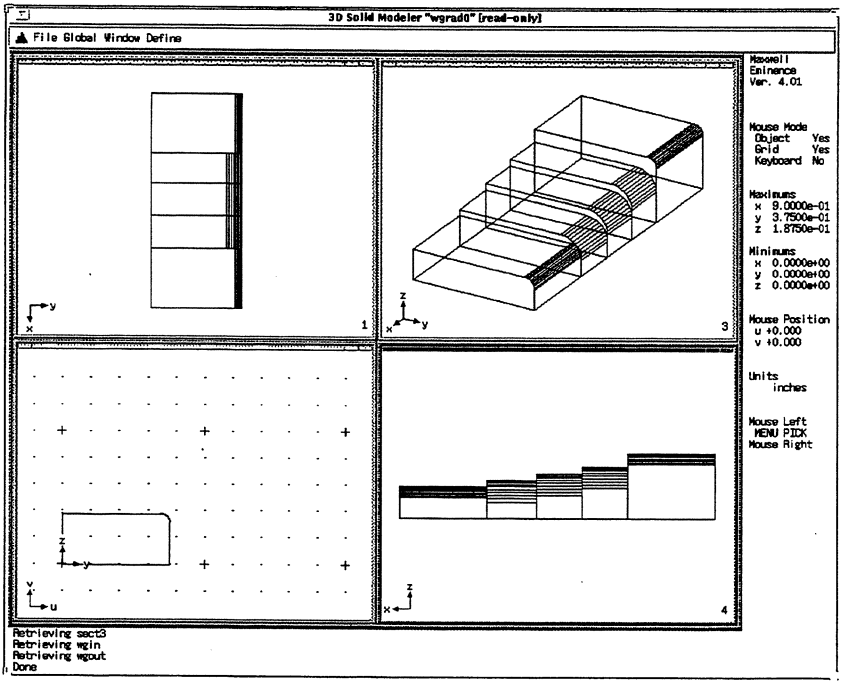


Fig. 5.3 Solid model for the nominal project "wgrad0".

When you are done viewing the solid model, exit from the Solid Modeler by clicking on the menu option "File" and then clicking on the pull-down menu option "Exit".

Incremental Change Projects

In order to parameterize a structure for optimization, we need to create a new project for each parameter to represent an incremental change. By comparing the new project with the nominal project, Empipe3D captures the information necessary for translating parameter values to a corresponding solid model.

The wgrad tutorial example has six parameters, namely L1, L2, L3, H1, H2 and H3, as depicted in Fig. 5.1. We need to create six new projects, each representing an incremental change in one of the parameters. Fig. 5.4 illustrates the nominal project and the incremental changes for two of the parameters.

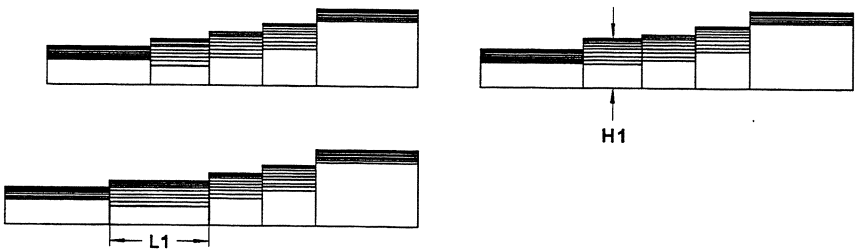


Fig. 5.4 Illustration of the nominal project and the incremental changes for two of the six parameters. The drawings depict the side view of the solid models.

For the tutorials, all the incremental change projects are already properly set up and are stored in the directories "wgrad1.pjt" through to "wgrad6.pjt".

Parameter Definition Data

In the Empipe3D window, each entry line under the heading

Parameter Name	Project Name	Nominal Value	Perturbed Value	# of Divs	Unit Name

contains the definition of one parameter.

For the wgrad tutorial example, there are six lines containing the definitions of the six parameters.

Parameter Name is an arbitrary ASCII string of no more than 32 characters. The names chosen for the waveguide transformer parameters are L1, L2, L3, H1, H2 and H3.

Project Name identifies the project representing an incremental change in the parameter value with respect to the nominal project. The projects that represent the incremental changes for the parameters L1, L2, L3, H1, H2 and H3 are "wgrad1", "wgrad2", "wgrad3", "wgrad4", "wgrad5" and "wgrad6", respectively.

Nominal Value refers to the value of the parameter represented by the nominal project. The nominal values for the parameters L1, L2, L3, H1, H2 and H3 are 0.14, 0.13, 0.13, 0.22, 0.26 and 0.3, respectively.

Perturbed Value refers to the parameter value after the incremental change. For instance, the incremental change in L1 changes its value from 0.14 to 0.24.

Number of Divs means "number of divisions", which specifies the number of interpolation intervals the incremental change encompasses:

$$\textit{Interpolation Interval} = | \textit{Perturbed Value} - \textit{Nominal Value} | / \textit{Number of Divs}$$

Empipe3D applies interpolation to small changes in parameter values within the appropriate interpolation interval. This can substantially reduce the total number of calls to the 3D Solver required for optimization.

The parameter L1, for instance, has an incremental change of 0.1 inch and the number of divisions is defined as 5. Therefore, its interpolation interval is 0.1 / 5 = 0.02 inch.

A suitable size of the interpolation interval may vary for different parameters, depending on the sensitivity of the responses. If the responses of interest are very sensitive with respect to a parameter, then the interpolation interval may have to be made relatively small. On the other hand, too small an interpolation interval will increase the number of 3D EM simulations and ultimately defeat the purpose of using interpolation.

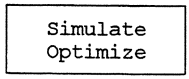
A possible approach to a problem for which you are not sure about the best interpolation interval size is to start with a coarse (large) interval and gradually refine the interval size as necessary. See Chapter 9 for further discussions on the subject of interpolation.

Unit Name identifies the physical unit of the parameter. Recognized unit names include IN (inch), MIL (milli-inch), M (meter), CM (centimeter), MM (millimeter), UM (micron) and NONE (without unit).

This button appears on each line that has a parameter defined. If you click on it, the 3D Solid Modeler will be invoked to display the structure defined by the incremental change project specified on that line.

5.3 Defining Variables and Specifications

In the Empipe3D window, click on the button



Two new windows appear on the screen, entitled "Empipe3D Select Variables" and "Empipe3D Specifications", respectively.

Selecting Optimization Variables

Since this is a tutorial, the selection of variables is already defined. Notice that the starting values for the parameters are different from their nominal values, i.e., their values in the nominal project. The reason is as follows. The parameter values for the nominal project were chosen rather casually, mainly for the convenience of drawing using the 3D Solid Modeler.

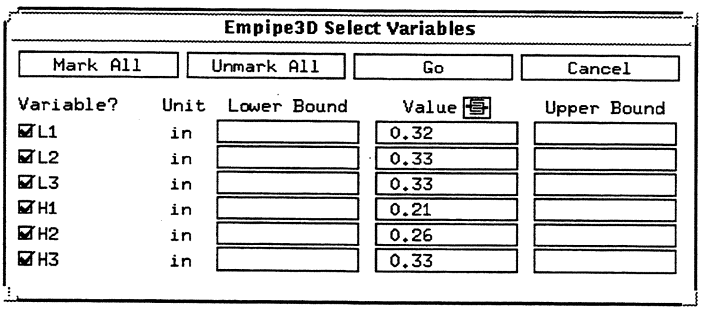


Fig. 5.5 The "Empipe3D Select Variables" window.

Although the nominal dimensions were in reasonable proportions, they were not optimized. The starting values defined in the "Empipe3D Select Variables" window, on the other hand; come from an optimized solution of the same waveguide impedance transformer using an empirical model which ignores the corner radius.

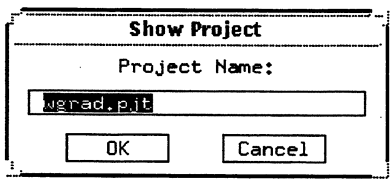
This reflects a sound design approach in engineering practice. One should not jump into EM optimization using a CPU intensive solver from an arbitrary starting point. Instead, a reasonably good starting point can be obtained from experience, intuition, synthesis, optimization of an empirical model, etc. This can greatly accelerate the convergence of EM optimization as well as enhance the uniqueness of the solution.

Verifying the Result of Geometry Capture

We have defined six parameters using Geometry Capture. Before proceeding, we would like to make sure that the incremental change projects have been implemented and processed properly. In designs such as this one that involve many parameters one can easily make a mistake in preparing the different projects. Consequently, the parameterization results may be totally scrambled. If optimization is carried out before a careful check is done, valuable time and computer resources can be wasted.

A verification which is easy to perform is to ask Empipe3D to derive a new project from the parameterization data. You would enter a new set of parameter values (new in the sense that it is different from the parameter values of the nominal and incremental change projects) in the box or boxes under the heading "Value". Then click on the [⇒] button. Once the solid model is displayed on the screen, perform a visual inspection to make sure that it is consistent with the parameter values you have entered. You can try this one parameter at a time, which makes it easier to spot any inconsistency.

For example, click on the entry box under the heading "Value" for the parameter L2, and type a new value "1". Then click on the [⇒] button. A dialog box appears:



Click on "OK" to accept the default.

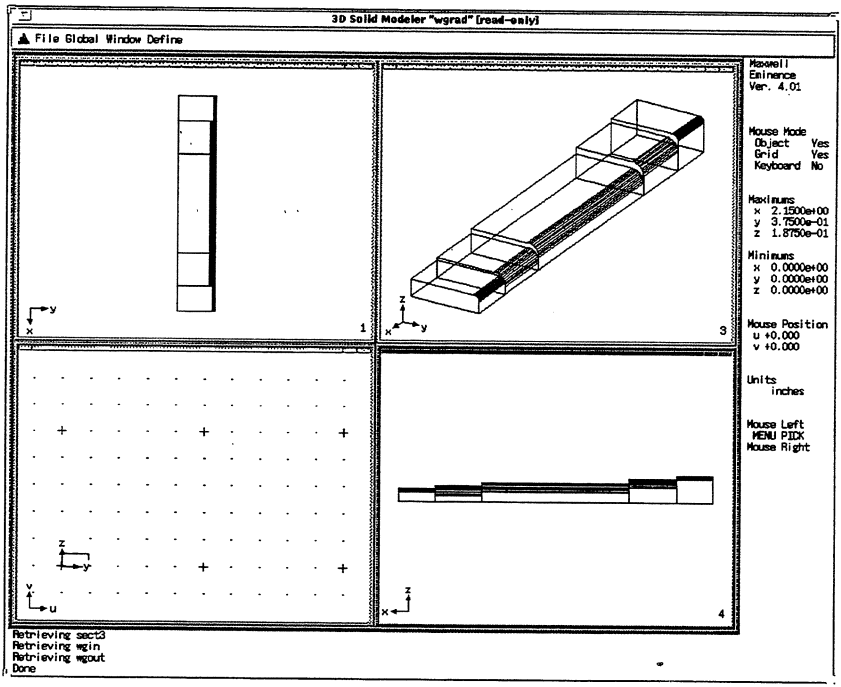


Fig. 5.6 The solid model for L1 = 1.

We have changed the length of the middle section from 0.33 inch to 1 inch. From Fig. 5.6, we can visually confirm that the parameterization of L2 seems to be correct. If you wish to be absolutely sure, you can use the 3D Solid Modeler's menu option "Window/Measure" to verify that the length L2 is indeed 1 inch.

Let us test for another parameter. First, restore the starting value for L2 by clicking on the entry box for L2 and type "0.33". Now click on the entry box for the parameter H3 and type "0.375". This is the height of output waveguide (WR-75 full height), so we expect to see the third section flush with the output waveguide.

Click on the button. When the dialog box appears for the project name, click on "OK" to accept the default.

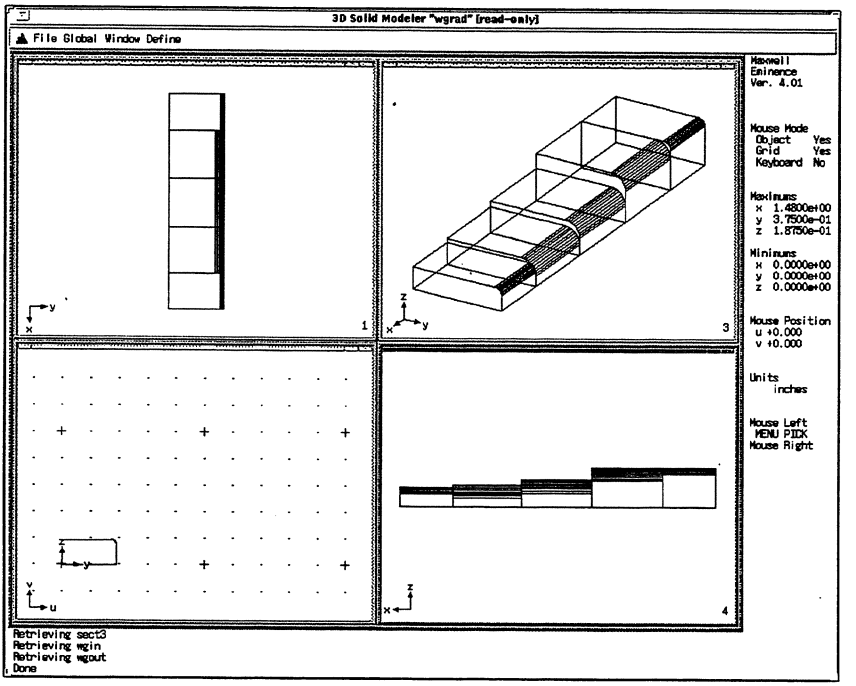


Fig. 5.7 The solid model for H3 = 0.375.

This confirms that the parameter H3 has been correctly captured by Empipe3D. The other parameters can be verified in a similar way.

Restore the starting value for H3 by clicking on the entry box for H3 and type "0.33".

Do not forget to restore the starting parameter values which have been modified. Otherwise, the subsequent tutorial on the optimization of the impedance transformer will be adversely affected.

Specifications for Optimization

We define the specification for this example in the "Empipe3D Specifications" window following the steps outlined in the tutorials of Chapters 3 and 4. The specification is:

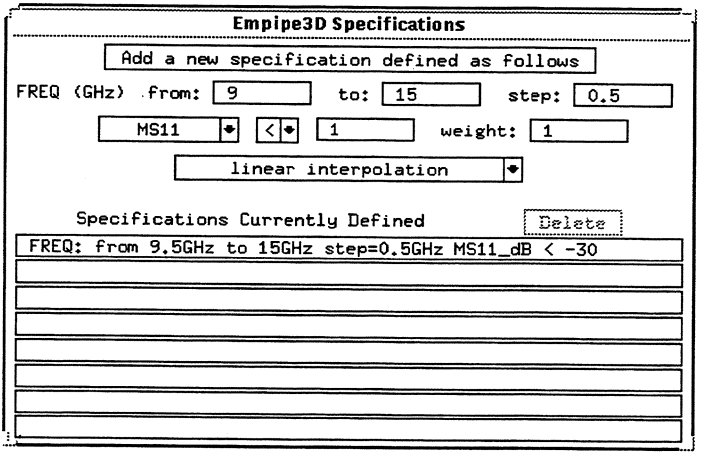


Fig. 5.8 The "Empipe3D Specifications" window.

What's Next

We have completed the selection of variables and specifications. Proceed by clicking on the "Go" button in the "Empipe3D Select Variables" window.

5.4 Simulation and Display

After you click on the "Go" button, Empipe3D invokes the OSA90 simulation/optimization environment.

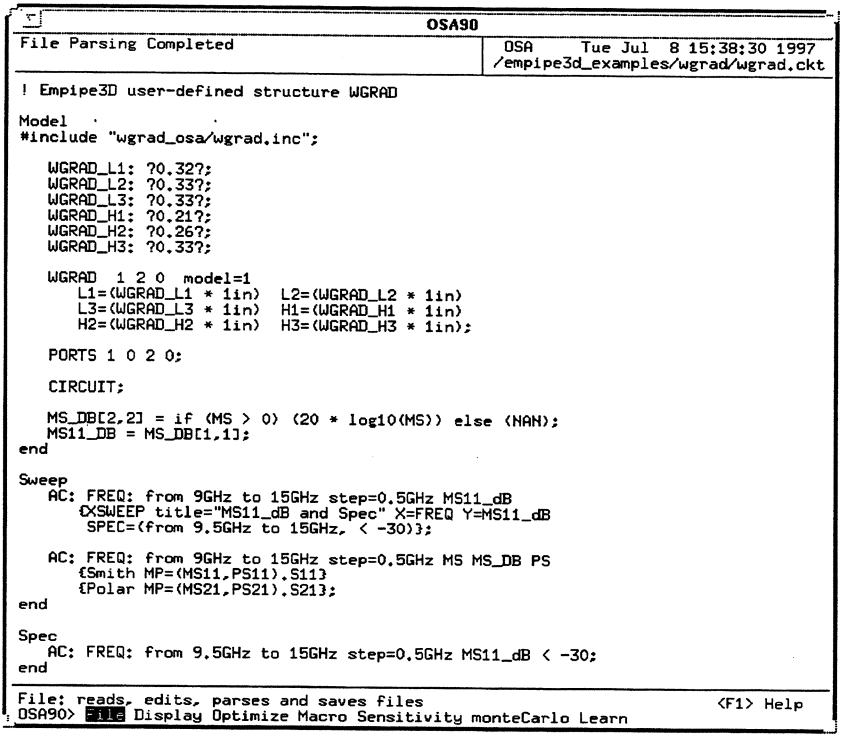


Fig. 5.9 OSA90 window.

At the top of the window is the message area. In the middle is the text area, where the input file (netlist) is shown. The OSA90 input file consists of a number of "blocks". Each block begins with a block identifier, such as "Model", "Sweep", "Spec" and "Control", and ends with the keyword "end". The Model block describes the circuit, the Sweep block selects the simulation outputs, the Spec block defines the specifications for optimization, and the Control block contains one or more operation control options.

At the bottom of the window is the menu area, where the menu options are presented.

Simulation before Optimization

Click on the menu option Display. The *S* parameters of the transformer simulated by the 3D Solver have already been saved in a database. After the data is retrieved and the display menu options appear, click on Xsweep. A pop-up window appears, showing the default setting for display. Press <Enter> to accept the default setting. The screen display is illustrated in Fig. 5.10.

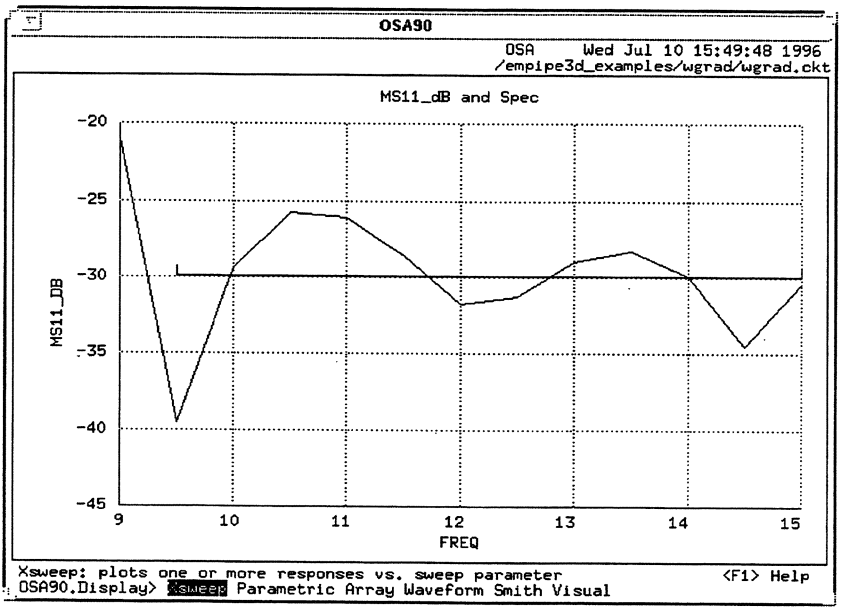


Fig. 5.10 The impedance transformer response before optimization.

Even before optimization, the response is not too far away from satisfying the specification. This is due to the fact that the starting point comes from an optimized empirical design.

Display Options

Click on the display menu option Xsweep. A pop-up window appears::

```
Output form:    Graphical
Draw type:      Curves
Zoom scale:     <TAB> to set scale
Sweep set:      sweep set 1
Y-axis:         view_1
X-axis:         FREQ
Ready to go:

<ENTER> = go  or  <ESC> = cancel
Select item with <UP>/<DOWN>
<F1> help
```

A number of display options are presented.

The options in the section "Numerical Output" allow you to toggle between graphical and numerical outputs.

The button "Style" causes another dialog box to appear, allowing you decide how the data should be plotted: as a continuous curve, a series of symbols (points, dots, circles, etc.).

The button "Zoom" also causes a dialog box to appear which provides the facility of changing the range of data to be displayed as well as the division of ticks along the X- and Y-axes.

The options "Sweep set", "Y-axis" and "X-axis" allow you to select one or more responses from those available for display.

In the following, we will demonstrate how to utilize some of these options to tailor the appearance of the display.

The display options are described in full detail with many illustrations in the *OSA90/hope User's Manual*.

Changing the Types of Drawing

The option "Draw type" allows you decide how the data should be plotted: as a continuous curve, a series of dots, or some other shapes.

Click on the option "Draw type". Another pop-up window appears, showing the available choices: Curves, Bars, Points, Dots, Bins and Circles. Click on the choice Dots. Then click on the line that says "Ready to go". The resulting display is then plotted on the screen.

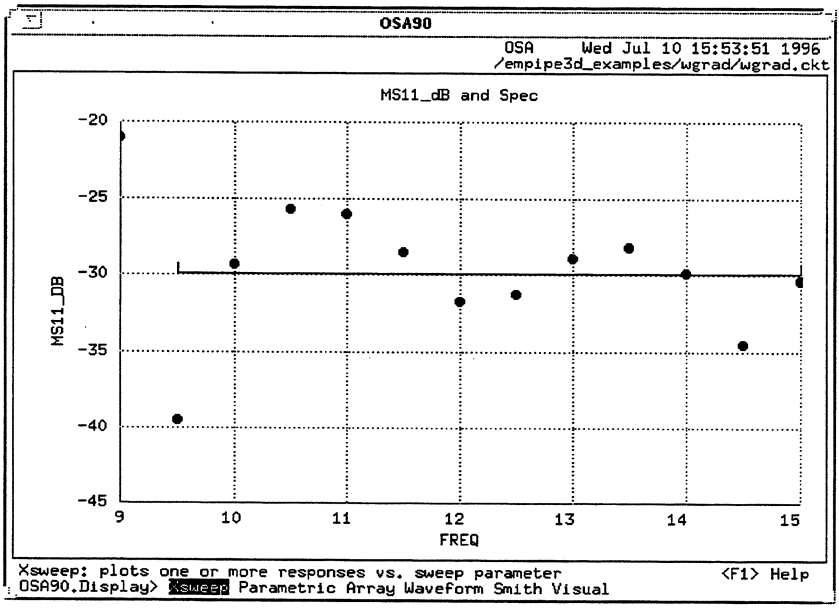


Fig. 5.11 The data is plotted as dots.

Changing the Zoom Scale

By default, the scale of the Y-axis is automatically determined according to the minimum and maximum values of the response or responses to be displayed.

You can change the default setting using the "Zoom" button.

Click on the display menu option Xsweep. In the pop-up window, click on the option "Zoom scale". Another pop-up window appears.

```
Zoom feature:       Disabled
Ymin of the plot:   -45
Ymax of the plot:   -20
Xmin of the plot:   9
Xmax of the plot:   15
N x-axis ticks:
N y-axis ticks:
Ready to go:

<ENTER> = go  or  <ESC> = cancel
Select item with <UP>/<DOWN>
<F1> help
```

The parameters "Ymin", "Ymax", "Xmin" and "Xmax" define the corners of the display. The parameters "N x-axis ticks" and "N y-axis ticks" specify the number of intervals (divisions) on the X-axis and Y-axis, respectively.

Move the cursor to the line labelled "Ymax", and type "0". This changes the maximum of the Y-axis (i.e., the top) from -20 to 0. Then press <Enter>. The window for zoom scale should now look like this:

```
Zoom feature:       Enabled
Ymin of the plot:   -45
Ymax of the plot:   0
Xmin of the plot:   9
Xmax of the plot:   15
N x-axis ticks:
N y-axis ticks:
Ready to go:

<ENTER> = go  or  <ESC> = cancel
Select item with <UP>/<DOWN>
<F1> help
```

Click on the line that says "Ready to go". This closes the zoom scale window and you are back at the display pop-up window. Click on the line that says "Ready to go" in this window. The resulting display is shown in Fig. 5.12.

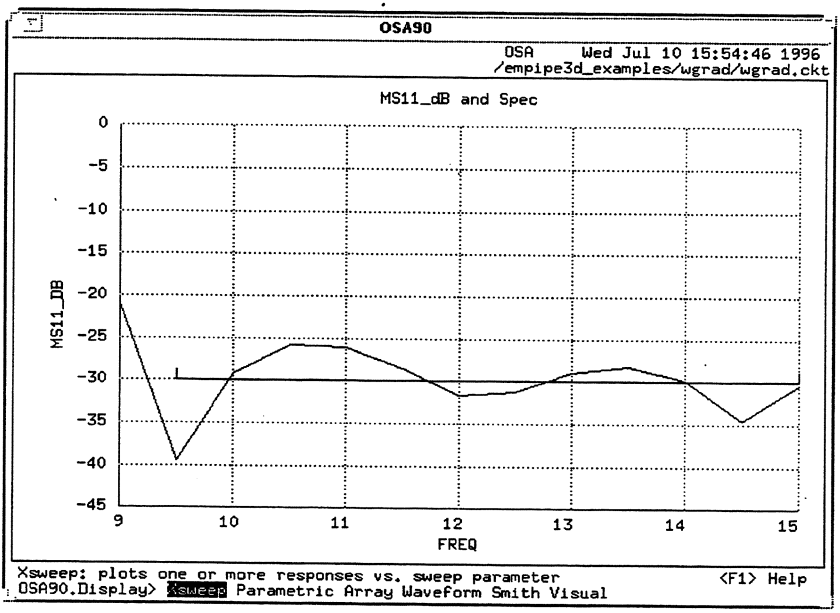


Fig. 5.12 The display with the modified scale.

Numerical Display

Click on the display menu option **Xsweep**. In the pop-up window, click on the option "**Output form**". Another window appears, showing the options: "**Graphical**", "**Numerical**" and "**Numerical append**". Click on "**Numerical**" and then click on the line that says "**Ready to go**". The display now shows the response as numerical data, as depicted in Fig. 5.13.

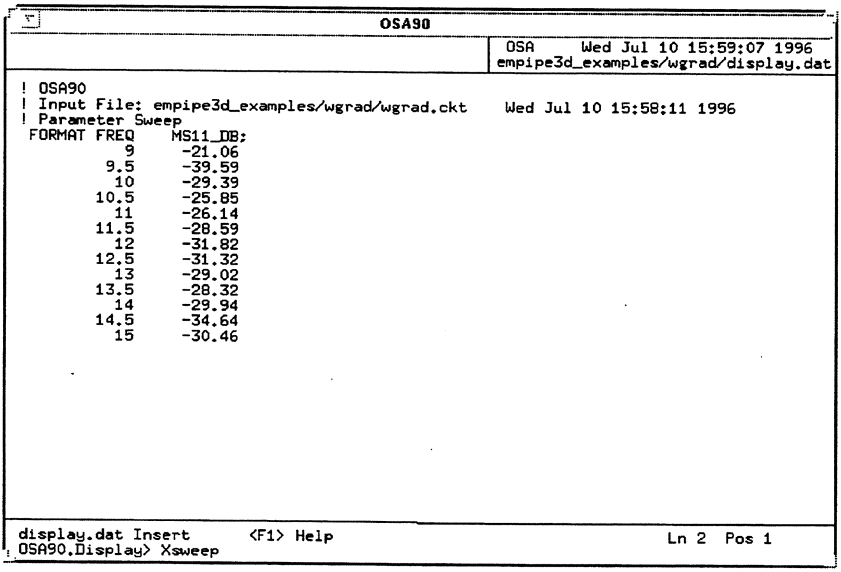


Fig. 5.13 The display of numerical data.

The numerical data is actually displayed within the **OSA90** file editor. You can edit the data and save it to a disk file, if so desired. Operations within the file editor are described in Chapter 7.

Exit from file editor by pressing the **<Esc>** key. Also, restore the output form to graphical output: click on the display menu option **Xsweep**; in the pop-up window, click on the option "**Output form**" and then click on the option "**Graphical**".

Displaying Different Responses

In the pop-up window, click on the option "**Sweep set**". When another window appears and shows the choices, click on "**sweep set 2**". Then, click on the option "**Y-axis**" and a list of available responses is presented: **MS11, MS12, MS21, MS22, MS_DB[1,1]**, etc. Click on the label "**PS11**". Click on the "**Ready to go**" line. The display is shown in Fig. 5.14.

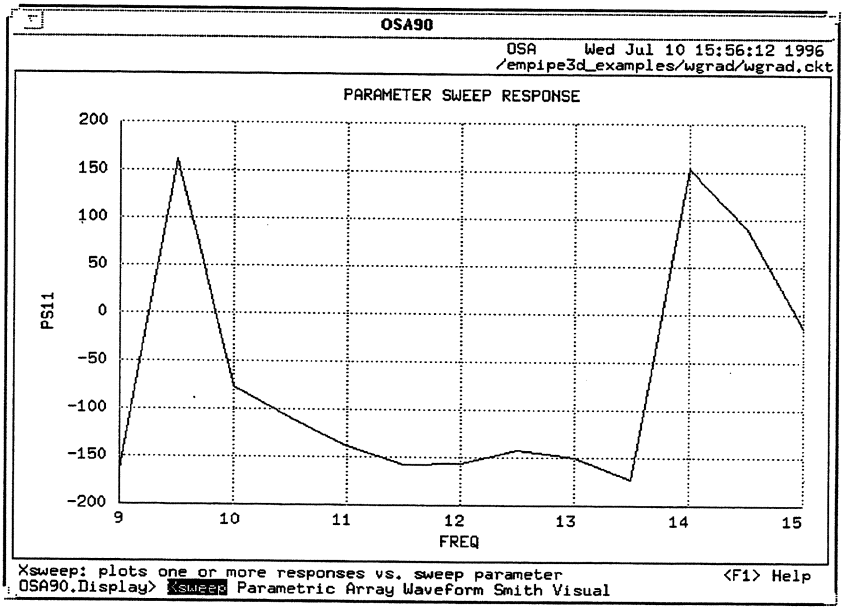


Fig. 5.14 The display of the response PS11.

What's Next

Before we proceed to the next section on optimization, we should exit from the display menu. You can do this either by clicking the right-hand mouse button anywhere inside the menu area, or by pressing the <Esc> key. After doing so, you should be back at the main menu

```
OSA90> File Display Optimize Macro ...
```

5.5 Optimization

To start optimization, click on the menu option Optimize. A pop-up window appears, showing the options for optimization. Press <Enter> to accept the default setting.

On the screen, the progress of optimization is reported:

```
Iteration   1/30 Max Error=4.15486
Iteration   2/30 Max Error=2.75391
Iteration   3/30 Max Error=1.29814
Iteration   4/30 Max Error=2.1607
Iteration   5/30 Max Error=0.491133
Iteration   6/30 Max Error=0.482656
Iteration   7/30 Max Error=0.328469
Iteration   8/30 Max Error=0.292162
Iteration   9/30 Max Error=0.248581
Iteration  10/30 Max Error=1.93186
Iteration  11/30 Max Error=0.228823
Iteration  12/30 Max Error=0.225896
Iteration  13/30 Max Error=0.209504
Solution  Max Error=0.209504
```

The 3D Solver simulation results required for optimizing this tutorial example have been saved in a database, therefore the optimization proceeds rapidly. For new projects, the 3D Solver will be invoked at the starting point and whenever a parameter value is changed by the optimizer out of its interpolation interval.

Simulation of the Optimized Impedance Transformer

Click on the menu option Display. At the OSA90.Display menu, click on the menu option Xsweep. After the pop-up window appears, press <Enter> to accept the default setting..

The optimized response of the impedance transformer is displayed on screen.

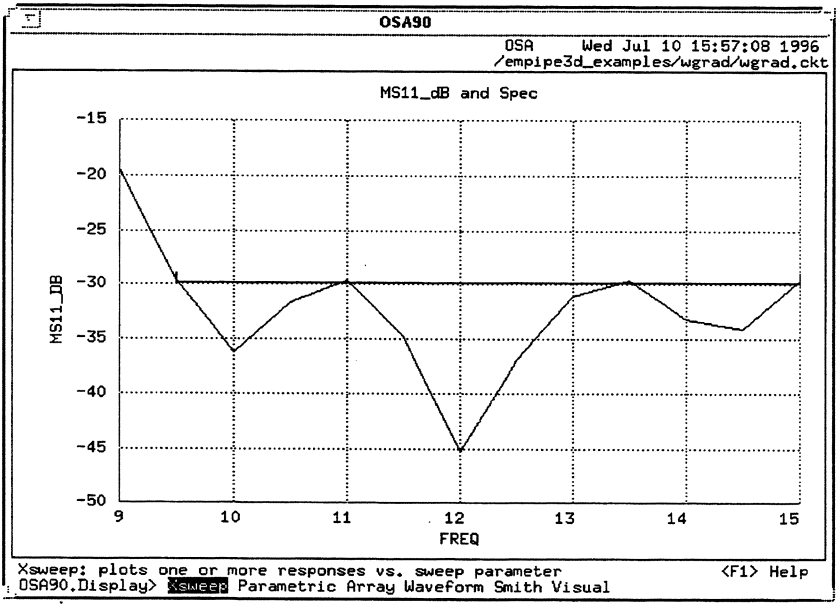


Fig. 5.15 The impedance transformer response after optimization.

The Optimized Parameter Value

Click the right-hand mouse button to exit from the display menu (or press the <ESC> key). At the OSA90 main menu, click the right-hand mouse button (or press the <ESC> key). You will be prompted:

```
Exit from OSA90 (Y/<N>):
```

Click the left-hand mouse button to confirm (or press the <Y> key).

Upon exit from OSA90, the Empipe3D windows reappear on the screen.

The optimized value of the parameter "d" is shown in the "Empipe3D Select Variables" window under the heading "Value".

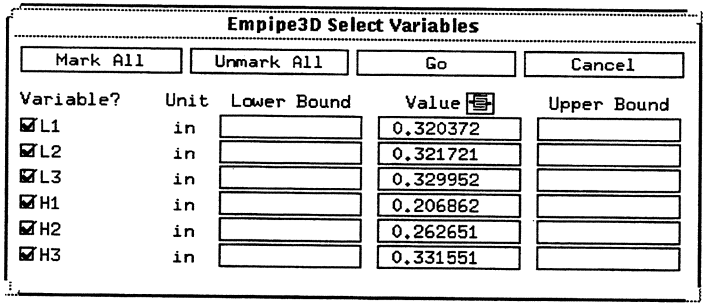


Fig. 5.16 The "Empipe3D Select Variables" window.

Finishing the Tutorial

Exit the windows for selecting variables and specifications by clicking on the "Cancel" button in the "Empipe3D Select Variables" window.

Exit the Empipe3D main window by clicking on the "Quit" button.

This concludes the tutorials.

6

Geometry Capture

6

Geometry Capture

6.1 Introduction

We hope that the series of tutorials in Chapters 3 to 5 has acquainted you with the features and operation of Empipe3D.

The steps of using Empipe3D for 3D EM optimization can be summarized as follows.

1 Create a 3D Solver project to represent the nominal structure.

2 For each parameter you wish to define, create a new project which describes how the structure will be affected by an incremental change in the parameter value. This is done by using the 3D Solid Modeler to copy and modify the nominal project.

3 Invoke Empipe3D to process the set of projects by Geometry Capture.

4 Define optimization variables and specifications using Empipe3D.

5 Perform simulation and optimization within the OSA90 environment.

6 Save the optimized structure as a 3D Solver project.

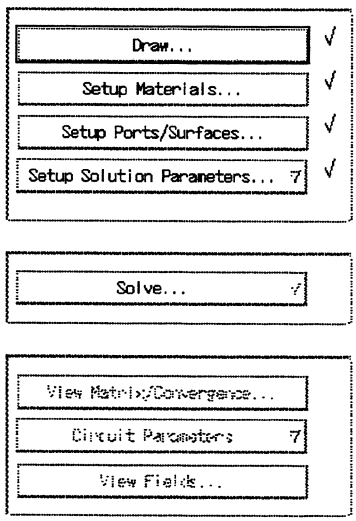
This chapter provides a formal description of the Geometry Capture procedure, focusing on steps 1 to 4.

6.2 Guidelines for Preparing Projects

This section describes the general guidelines for setting up the 3D Solver projects for Geometry Capture. These guidelines apply to both the nominal and incremental change projects. There are also additional guidelines that are specific to the nominal or the incremental change projects, and these are discussed in Sections 6.3 and 6.4.

1 The names of the nominal and incremental projects must be different from the name of the Empipe3D element. For instance, if the Empipe3D element is named "wgrad", then the nominal and incremental change projects for this element cannot be named "wgrad.pjt". Empipe3D reserves the name "wgrad.pjt" for its working project. We recommend that you name the projects using a number sequence, such as "wgrad0.pjt", "wgrad1.pjt" and so on. You may also use descriptive names, such as "wgrad_nom.pjt", "wgrad_H1.pjt" and "wgrad_L1.pjt".

2 The nominal and incremental change projects must include the complete setup of material, ports, boundaries and solution parameters. When this is the case, the 3D Solver main operation menu should look like this:



You can also check the "*.sat" file in the "*.pjt" directory, which should have this statement near the end of the file:

```
status BOUNDARIES_DEFINED
```

3 Do not solve or mesh any of the projects. The reason is that the meshing process alters the solid model file, rendering it unsuitable for Geometry Capture. You can specify seeded mesh, but do not actually generate the mesh. Do not invoke the "Boundary Display" option in the 3D Solver's "Setup Ports/Surfaces" menu since it will start mesh generation. In case you have a project which is already solved or meshed, here is a trick to "undo" the meshing. First, make of a copy of the project if you wish to save the meshed project. Open the project in the 3D Solver, and choose the "Draw" command. When you are prompted for "View Only" or "Modify", choose "Modify". In the Solid Modeler window, without making any actual changes, click on the "File" menu and choose "Exit". Then, choose "Save the current model". This sequence of operations forces the 3D Solver to remove the current mesh from the solid model file.

4 Although you can invoke the Solid Modeler from Empipe3D, it is invoked with the "View Only" mode. For the purpose of creating or modifying a project, you will need to start the 3D Solver directly.

5 Exploit geometrical symmetry to reduce the computation time whenever possible by defining "Perfect E" and/or "Perfect H" boundaries. In doing so, both nominal and incremental change projects need to be defined consistently.

6 To control the 3D Solver solution refinement, you can specify the maximum number of adaptive meshing passes and the required *S*-parameter accuracy ("delta S"). Also, Empipe3D can recognize and process both discrete and "fast" frequency sweeps.

6.3 The Nominal Project

The nominal project typically represents a starting point in the design process. It may be obtained by synthesis, from experience or through optimization of an empirical model.

Please follow the general guidelines for preparing projects for Empipe3D as outlined in Section 6.2. The following provides a few additional notes for setting up the nominal project.

1 If you are creating the nominal project from scratch, please carefully take note of the sequence of the drawing operations. This information is valuable when you create the incremental change projects later.

2 Complete the project setup by including material, ports, boundaries and solution parameters. In other words, the project should be ready to be solved.

3 Do not solve or mesh the nominal project. If you intend use a project which has already been solved or meshed, please follow the suggestions given in Step 3 in Section 6.2.

4 Before preparing the incremental change projects, you may wish to verify the nominal project by solving it using the 3D Solver. To do this, make a copy of the project and run the 3D Solver on the copy. This way, you can keep the original project before mesh generation as required by Empipe3D.

6.4 The Incremental Change Projects

In addition to the nominal project, you need to prepare n projects to represent n design parameters. The design parameters may include geometrical dimensions and material parameters. Each design parameter has an associated incremental change project. Empipe3D captures the design parameters by comparing each incremental change project with the nominal project.

Consider the example shown in Fig. 6.1. We wish to define two parameters: W and L.

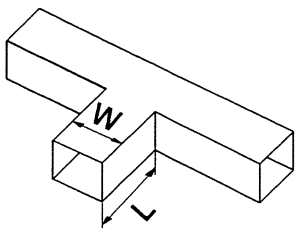


Fig. 6.1 A simple structure with two parameters.

For each parameter, a new project needs to be created to represent an incremental change in the parameter value. It describes how the structure is affected by the change in the parameter value.

For instance, Fig. 6.2 depicts the nominal structure and the structure representing an incremental change in the parameter L.

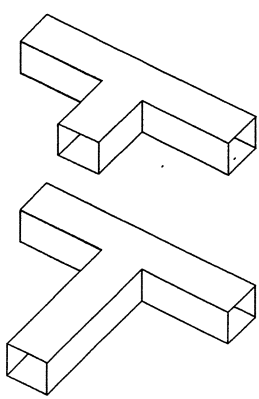


Fig. 6.2 The nominal structure and an incremental change in the parameter L.

By comparing the nominal and incremental change projects, Empipe3D extracts the relevant differences to establish the dependency of the project files on the parameters. This information enables Empipe3D to translate any given set of parameter values into the appropriate solid model and other project files.

Fig. 6.3 depicts the nominal structure and the structure representing an incremental change in the parameter *W*.

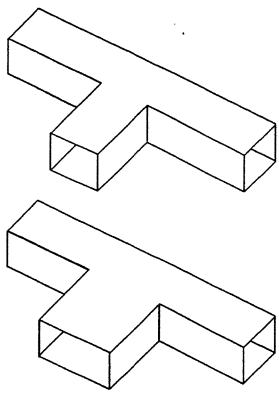


Fig. 6.3 The nominal structure and an incremental change in the parameter W.

The illustrations in Figs. 6.2 and 6.3 purposely exaggerate the effects of the incremental changes. In practice, the incremental changes do not have to be so dramatic.

Incremental Changes Are Not Bounds

Note that the incremental change does not institute a bracket on the range of the parameter value. A separate arrangement is used to impose bounds on the parameter values for optimization (see Section 6.5). The incremental change of a parameter merely establishes a direction for that variable. The change in the parameter value can be either positive or negative, i.e., the parameter value can be either increased or decreased with respect to the nominal value.

Guidelines for Preparing Incremental Change Projects

Please follow the general guidelines for preparing projects for Empipe3D as outlined in Section 6.2. The following provides a few additional notes for setting up the incremental change projects.

1 Invoke the 3D solver directly.

2 From the 3D Solver project control panel, copy the nominal project to a new project. This new project, after being appropriately modified, will become an incremental change project. We recommend that you follow the naming scheme suggested in Step 1 of Section 6.2.

3 Modify the new project to reflect an incremental change in the parameter value. If the parameter represents a geometrical dimension, invoke the 3D Solver "Draw" operation. To modify a 3D object, you have to delete the existing object and redraw a new one. If the change affects the position of ports or boundaries, then you also need to make an appropriate adjustment to the "Setup Ports/Surfaces" option of the Solid Modeler. Dielectric and other material parameters can be defined by modifying the appropriate project setup options.

4 The number of 3D solid objects must remain the same between the nominal and the incremental change projects.

5 The number of vertices of each object must remain the same. In other words, the basic topology of the structure must not change. For instance, you can change the size of a rectangle, but you must not change a rectangle into a triangle.

6 You must be careful to assign the same name to the new object as the one it replaces. This ensures that the objects are recorded in the solid modeler file in an order that is consistent with the nominal project. If the objects appear in a different order, the results will be scrambled.

7 It is recommended that when you redraw an object, the order of the vertices remains the same. That is, you start drawing the new object from the same vertex as that of the old object and follow the same direction (clockwise or counter-clockwise).

8 The incremental change projects must include the complete setup of material, ports, boundaries and solution parameters, so that they are ready to be solved. Do not solve or mesh any of the projects.

9 To verify that the incremental changes are correctly implemented and processed, you can ask Empipe3D to generate parameterized projects and visually inspect the results. This approach is illustrated in the tutorial of Chapter 4.

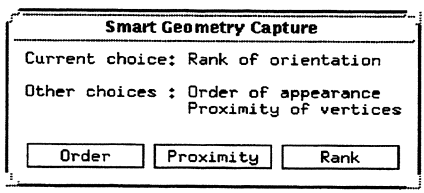
Smart Geometry Capture

Smart Geometry Capture allows Empipe3D to properly interpret the data stored in the nominal and incremental project files even if the order of drawing the objects is not consistent between those projects (see Item 6, Guidelines for Preparing the Incremental Change Projects). This can be particularly useful in reusing an existing project that was created some time ago or by another person.

There are three methods of carrying out the parameterization process and the user is given access to control its selection. Normally, there is no need to exercise this control unless you notice improper interpretation of the Geometry Capture projects when you try to verify that the incremental change projects are correctly processed (see Item 2, Hints for Preparing the Geometry Capture Projects).

To access the Smart Geometry Capture options click on the OSA logo button in the main Empipe3D window. A pop-up window titled "Empipe3D License Information" will appear. It will include a button titled "Options". Click on the "Options" button and the "Smart Geometry Capture" window will appear. This window lists the options for "Smart Geometry Capture", including the currently selected method.

The options are "Order" (Order of appearance), "Proximity" (Proximity of vertices), and "Rank" (Rank of orientation).



1 **Order of appearance.** To invoke this option you must be sure that all the solid objects in the incremental change projects have been drawn in the exact same order as that of the nominal project. This means that when you redraw a perturbed object, you must start from the same vertex and follow the same direction (clockwise or counter-clockwise) as you did for the nominal project. This can become very tedious for a complex project, and therefore this method should be used only as the last resort when the other approaches fail.

When the "Order" method is to be used you can verify the consistency between the nominal and incremental change projects using the utility program "cmpjt" supplied with Empipe3D. It compares the key files in the project directories. For instance, use

```
cmpjt bend20 bend21
```

to compare the projects "bend20.pjt" and "bend21.pjt". The files compared include the solid model file (*.sld), the port/boundary and solution definition file (*.sat), and the material definition file (*.db). A summary of these files can be found in the 3D Solver User's Manual in the "Implementation and Theory" section.

Typically, the incremental change in one parameter affects relatively few lines in these files. If the differences listed by "cmpjt" appear to be numerous, then very likely one or more of the rules outlined here have been violated.

2 **Proximity of vertices.** This method works well when the displacement of the vertices in the incremental change projects is small as compared with the minimum distance between different vertices in the nominal object. However, very small changes may lead to unnecessarily small interpolation intervals.

3 **Rank of orientation.** This is the most robust method for typical structures. It is based on the assumption that the perturbed objects preserve the orientation of the nominal project with respect to the *x*, *y* and *z* axes. Some caution is needed if the incremental change involves rotation of some vertices, such as in the case of defining a parameter representing an angle. Rank of orientation is the default option for all new projects.

Implicit Correlations and Constraints

Geometry Capture allows you to express graphically any correlations between parameters and subtle constraints on the geometry which are otherwise difficult to define.

For instance, Fig. 6.4 can be used to define a scaling parameter. When this parameter value changes, it affects both the width and length of the structure.

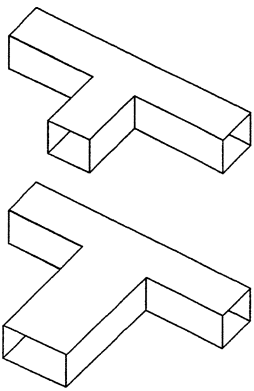


Fig. 6.4 The nominal and incremental change structures for a scaling parameter.

Let us consider another example. In Fig. 6.5, the incremental changes represent two separate and independent parameters, namely *L1* and *L2*. Fig. 6.6, however, implies that the parameters *L1* and *L2* are strongly correlated: an increase in *L1* implies a decrease in *L2* and vice versa. This is necessary in order to maintain the overall length (*L1* + *L2*) constant, which is a commonly encountered constraint in layout within a fixed enclosure. In fact, for the structure depicted in Fig. 6.6, there is really only one degree of freedom, and therefore it will suffice to define just one parameter (*L1* or *L2*).

An alternative way of enforcing geometrical constraints is to bind the variables through equations. This can be done in OSA90 using expressions (see *OSA90 User's Manuals* for detail).

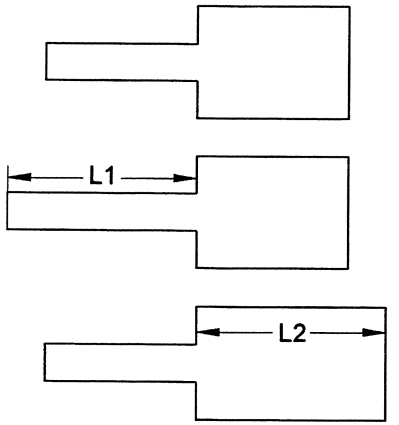


Fig. 6.5 Definition of two independent parameters L1 and L2.

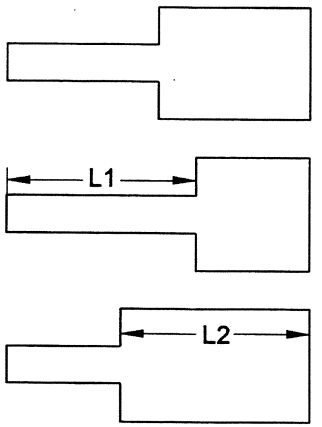


Fig. 6.6 The parameters L1 and L2 are interdependent.

6.5 Defining an Empipe3D Element

Once the projects have been prepared, start by typing

```
empipe3d element
```

where *element* represents the Empipe3D element name. It also serves as the base name for all the working files produced by Empipe3D.

We recommend that you choose the project names in a systematic way. For example, in the tutorial of Chapter 5, the Empipe3D element name is "wgrad" ("waveguide with corner radius"). The nominal project is named "wgrad0" and the incremental change projects are named wgrad1, wgrad2, wgrad3, wgrad4, wgrad5 and wgrad6.

Empipe3D stores the results of Geometry Capture in a subdirectory named "*element*_osa" (for example, "wgrad_osa"). Within this subdirectory, the parameterization data is saved in a file named "*element*.inc", and the database file is named "*element*.dbs".

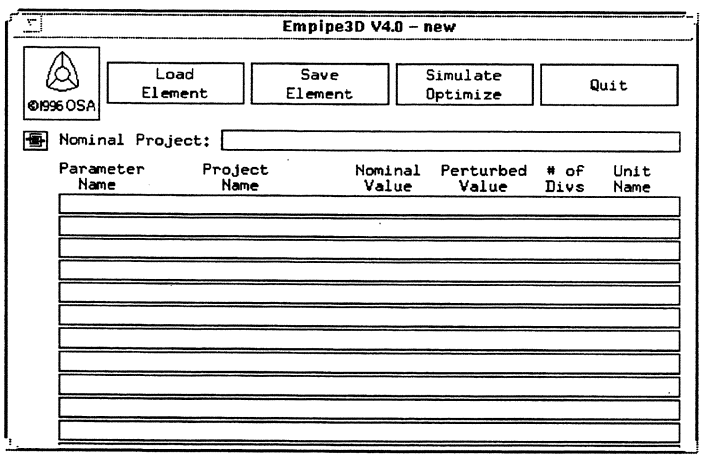


Fig. 6.7 A blank Empipe3D window for a new problem.

Nominal Project

Fill this entry box with the name of the nominal project.

This button appears adjacent to the entry box. If you click on it, Empipe3D will invoke the Solid Modeler to display the nominal structure.

Parameter Definition Data

Each entry line under the heading

Parameter Name	Project Name	Nominal Value	Perturbed Value	# of Divs	Unit Name

contains the definition of one parameter. Up to 16 parameters can be defined.

Parameter Name

This can be an arbitrary ASCII string of no more than 32 characters, such as "W", "Width", or "Length_of_Section_1".

Project Name

This entry identifies the project representing an incremental change in the parameter value with respect to the nominal project.

You can click on this button to view the solid model.

Nominal Value

This refers to the value of the parameter represented by the nominal project. It should be entered as a plain number. The physical unit, if any, is entered separately.

Perturbed Value

This refers to the parameter value after the incremental change.

Number of Divs

This means "number of divisions", which specifies the number of interpolation intervals the incremental change encompasses:

$$\textit{Interpolation Interval} = |\textit{Perturbed Value} - \textit{Nominal Value}| \,/\, \textit{Number of Divs}$$

Empipe3D employs interpolation to improve efficiency. During optimization, the 3D Solver is invoked for an EM simulation only if the parameter has been moved by the optimizer across an interpolation interval. For small changes in parameter values, interpolation is applied instead of EM analysis to obtain the *S* parameters. This can substantially reduce the total number of calls to the 3D Solver required for optimization. This is also an important feature for gradient calculations.

A good choice of the interpolation interval is obviously problem dependent. If the interval is too large (i.e., the number of divisions is too small), then the interpolated *S* parameters may not be accurate. On the other hand, a very small interpolation interval will increase the number of 3D EM simulations and ultimately defeat the purpose of using interpolation.

A possible approach to a problem for which you are not sure about the best interpolation interval size is to start with a coarse (large) interval and gradually refine the interval size as necessary. See Chapter 9 for further discussions on the subject of interpolation.

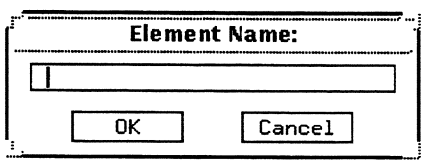
Unit Name

This entry identifies the physical unit of the parameter. Permissible unit names include IN (inch), MIL (milli-inch), M (meter), CM (centimeter), MM (millimeter), UM (micron) and NONE (without unit).

Menu Buttons in the Empipe3D Window

Load Element

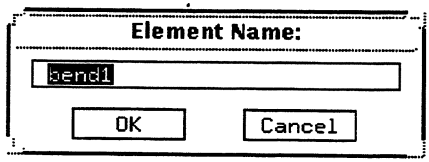
Click on this button to load a different Empipe3D element definition into the Empipe3D window. You will be prompted for the new element name:



Empipe3D then looks for the subdirectory "*element*_osa". If it is found, the parameterization data will be retrieved and shown in the Empipe3D window. Otherwise, Empipe3D will assume that a new element is to be defined.

Save Element

Click on this button to save the current data in the Empipe3D window to a disk file. You will see a prompt similar to this one:



The current element name is offered as the default, but you have the opportunity to give it a new name. This feature can be utilized to rename an element.

Simulate Optimize

Click on this button to start simulation and optimization. It leads to separate Empipe3D windows for selecting optimization variables and defining specifications, which is the subject of the following section (Section 6.5).

Quit

Click on this button to exit from Empipe3D.

This button provides license information about Empipe3D and access to "Smart Geometry Capture" setting through the "Options" button.

6.6 Optimization Variables and Specifications

When you click on the "Simulate Optimize" button in the Empipe3D window, two new windows will appear for you to select optimization variables and define specifications.

The "Empipe3D Select Variables" window is depicted in Fig. 6.8.

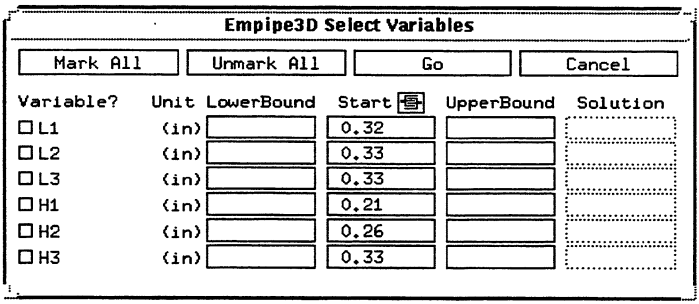


Fig. 6.8 The "Empipe3D Select Variables" window.

All the parameters are listed as candidates for optimization variables. For each parameter, there is a check box under the heading "Variable?" and adjacent to the parameter name. If you click on this box, a check mark will appear in the box to indicate that the associated parameter is now selected as an optimization variable. If you click on the box a second time, the check mark will disappear. In other words, the check box acts as a toggle switch, turning on and off the status of selection.

Alternatively, you can click on the "Mark All" button to select all the parameters or the "Unmark All" button to undo the selection of all the parameters.

The "Go" and "Cancel" buttons apply to both the "Empipe3D Select Variables" and the "Empipe3D Specifications" windows. Click on the "Go" button after you have completed both windows. The "Cancel" button cancels both windows.

Value

The starting point (initial value) for each optimizable parameter is shown under the heading "Value". By default, it is set to the parameter's nominal value. If you wish to choose a different starting point, click on the entry box under "Value" and enter the desired value.

After Simulation and Optimization, the Optimized Solution is displayed under "Value".

You can click on the  button to view the solid model of the starting point. This feature can also be used to generate new projects for arbitrary parameter values, as demonstrated in the tutorials of Chapters 4 and 5.

Upper and Lower Bounds

You can impose bounds on the optimization variables using the entry boxes under the headings "LowerBound" and "UpperBound".

During optimization, the parameter values are constrained within the lower and upper bounds if they are specified.

If the bounds are not given explicitly, they will be assigned automatically. Suppose that the starting point of a variable is x.

If the lower bound is not given explicitly, it is set to 0 if $x \geq 0$, or $-\infty$ if $x < 0$.

If the upper bound is not given explicitly, it is set to $+\infty$ if $x \geq 0$, or 0 if $x < 0$.

Specifications for Optimization

The "Empipe3D Specifications" window is illustrated in Fig. 6.9.

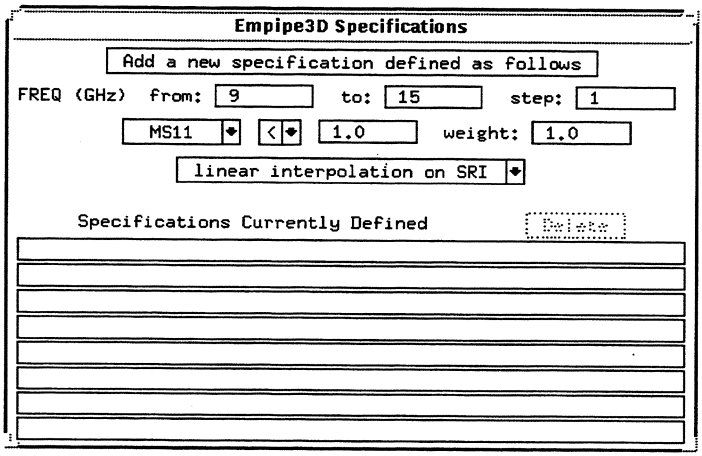


Fig. 6.9 The "Empipe3D Specifications" window.

In general, the definition of a specification involves the following steps:

1 Select a frequency range.

2 Select a response.

3 Select a specification type (upper, lower or equality specification).

4 Enter a numerical value as the goal.

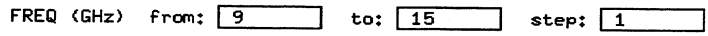
5 Optionally, enter a weighting factor.

6 Optionally, select an interpolation type.

7 Click on the button "Add a new specification defined as follows".

Selecting a Frequency Range

The entry line for selecting a frequency range appears similar to this one

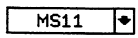


The frequency range defined for the nominal project (as part of the 3D Solver "Solution Parameters") is presented as the default.

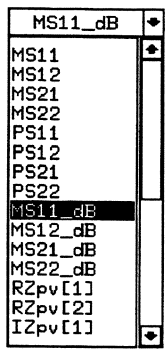
To change any of the three values (start, stop and step values), click on the appropriate box and type in the new value.

Selecting a Response

This is the entry box for selecting a response:



Click on the arrow adjacent to the box, and a list of available responses is displayed.



The label MS*ij* represents the magnitude of S_{ij}, PS*ij* represents the phase of S_{ij} in degrees, and MS*ij*_dB represents the magnitude of S_{ij} in decibels.

The labels RZpv[*i*], IZpv[*i*], RZpi[*i*], IZpi[*i*], Rzvi[*i*] and Izvi[*i*] represent the real and imaginary parts of the complex port impedances Zpv, Zpi and Zvi for the *i*th port, respectively.

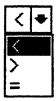
The labels RGamma[*i*] and IGamma[*i*] represent the real and imaginary parts of the propagation constant associated with the *i*th port, respectively.

Not all the available responses are shown at the same time due to the limited display area. You can use the scroll bar to browse through the list. To select a response from the list, simply click on it. If you wish to cancel the list while it is shown, click the mouse button outside the list window.

Selecting the Type of Specification

The entry box for selecting the type of specification appears as

Click on the arrow and you will see the list of available choices:



The symbols "<", ">" and "=" represent upper, lower and equality specifications, respectively.

Entering a Numerical Value as the Goal

The third entry box on the specification line represents the goal.

Optional Weighting Factor

You can enter an optional weighting factor. The default value is 1. To enter the weighting factor, click on the box and then type in the appropriate number (the weighting factor must be a positive number).

Completing the Definition of a Specification

Once you have selected the frequency range, the *S*-parameter response, the specification type, the numerical goal and, optionally, the weighting factor, click on the button labelled "Add a new specification defined as follows". The specification you have just defined is added to the next available line under the heading "Specifications Currently Defined".

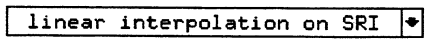
You can define multiple specifications with different frequency ranges, responses and goals. Up to 16 specification lines can be defined.

Deleting an Existing Specification

To delete an entry under the heading "Specifications Currently Defined", click on the line. The button "Delete" will turn from a shade of grey to a solid color ("clickable"). A click on this button will delete the highlighted specification line.

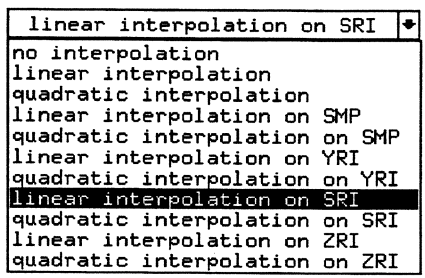
Interpolation Types

The "Empipe3D Specifications" window also includes this entry box



It allows you to select an interpolation type. The default is "linear interpolation on SRI", where "SRI" stands for *S* parameters in the rectangular form ("real and imaginary parts").

If you click on the arrow the list of available choices will be displayed:



The interpolation techniques used by Empipe3D are covered in Chapter 9.

Commencing the OSA90 Optimization Environment

When you have completed the definition of all the specifications *and* the selection of optimization variables, click on the "Go" button in the "Empipe3D Select Variables" window.

7

OSA90 Environment

7

OSA90 Environment

7.1 Introduction

OSA90 is a general-purpose simulation and optimization environment. It is included in the Empipe3D package to serve as the optimization engine. OSA90 is also marketed separately as a linear and nonlinear circuit CAD system.

OSA90 offers an impressive collection of state-of-the-art optimizers, including minimax, ℓ_1, ℓ_2 (least squares), quasi-Newton, conjugate gradient, Huber, simplex, random, simulated annealing and yield optimization algorithms.

OSA90 also offers a comprehensive set of algebraic operators and mathematical functions to facilitate user-defined pre- and post-processing of variables and responses. You can define labels, equations, conditional expressions (if and else), vectors and matrices. You can utilize built-in functions for matrix algebra, LU factorization, eigenvalues, eigenvectors, discrete Fourier transform, piece-wise linear and cubic spline interpolations, and so on.

You can plot responses and functions in a variety of formats: parameter sweeps, parametric plots, Smith chart and polar plots, even 3D visualization and contours.

Tools for statistical analysis are also at your finger tip: uniform, normal, exponential, lognormal and sample distributions, absolute and relative tolerances, correlation matrices, Monte Carlo analysis, histograms, run charts and scattering diagrams.

You can also license the OSA90/hope option, which offers additional circuit simulation and optimization capabilities, including nonlinear DC, small-signal AC and nonlinear large-signal harmonic balance analyses, comprehensive libraries of nonlinear active device and linear passive element models, user-definable linear subcircuits, user-definable nonlinear device models. You can simulate, display and optimize DC and AC voltages, currents, *S*, *Y* and *Z* parameters, insertion loss, stability factor, group delay, large-signal harmonic distortion, compression, intermodulation products, intercept points, and more. You can optimize designs of small-signal amplifiers, power amplifiers, filters, mixers, frequency multipliers and oscillators. You can take advantage of the unique Datapipe technology to connect external software for functionally integrated simulation and optimization.

The wealth of features of OSA90 is covered in detail in a separate manual: *OSA90/hope User's Manual*. In this chapter, we provide a summary of the input file structure and menu hierarchy, confined to the scope of Empipe3D applications.

7.2 OSA90 Window

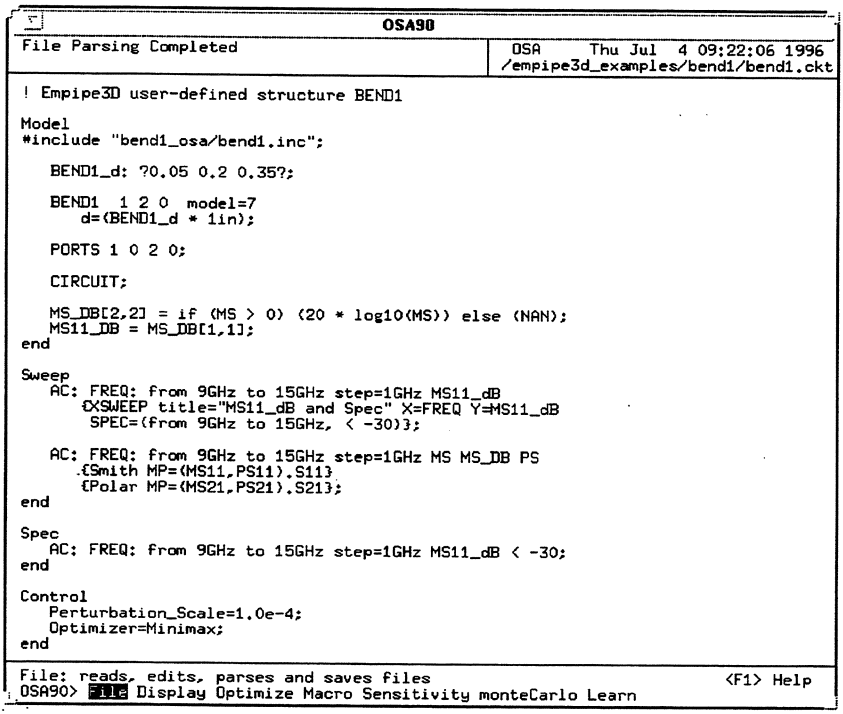


Fig. 7.1 OSA90 window.

At the top of the window is the message area, where the date, time and file name are shown. At the bottom of the window is the menu area, where the menu options are presented. In the middle is the display area, which contains either the text of the input file (netlist) or the graphics of the calculated responses.

7.3 OSA90 Input File

An OSA90 input file is like a netlist. It is an ASCII text file with the extension ".ckt". The input file contains definitions of parameters, variables, labels, models, responses, equations, frequency ranges, parameter sweeps, simulation outputs, optimization specifications, operation control options, statistical tolerances and distributions, etc.

The contents of an input file are divided into sections called file blocks. Each file block is designated to supply a particular type of information. Table 7.1 lists the input blocks that are relevant in the context of Empipe3D. More details are available in the *OSA90/hope User's Manual*.

TABLE 7.1 INPUT FILE BLOCKS

Block Name	Contents
MODEL	variables, labels, equations, models
SWEEP	simulation types, ranges, outputs
SPECIFICATION	optimization specifications
CONTROL	operation control options
MONTECARLO	statistical analysis ranges, outputs
STATISTICS	statistical correlation matrices
TRACE	record of optimization variables

Each input file block begins with a block name and ends with the keyword "end". Block names and keywords are case insensitive. For example, MODEL, Model and model are all treated as identical.

Contained within an input file block are statements. A statement consists of one or more lines of text and must always be terminated by a semicolon.

Example:

```
BEND1_d: 0.2;
```

Here is an example of a multi-line statement:

```
WGUIDE  1 2 0
   W=0.75in
   L=VAR1;
```

Example of OSA90 Input File: bend1.ckt

The following is the OSA90 input file for the waveguide bend example used in the tutorial of Chapter 3. The file name is "bend1.ckt". We will dissect this file to illustrate the structure and syntax of the OSA90 input files.

```
! Empipe3D user-defined structure BEND1

Model
#include "bend1_osa/bend1.inc";

   BEND1_d: ?0.05 0.2 0.35?;

   BEND1  1 2 0  model=7
      d=(BEND1_d * 1in);

   PORTS 1 0 2 0;

   CIRCUIT;

   MS_DB[2,2] = if (MS > 0) (20 * log10(MS)) else (NAN);
   MS11_DB = MS_DB[1,1];
end

Sweep
   AC: FREQ: from 9GHz to 15GHz step=1GHz MS11_dB
      {XSWEEP title="MS11_dB and Spec" X=FREQ Y=MS11_dB
       SPEC=(from 9GHz to 15GHz, < -30)};

   AC: FREQ: from 9GHz to 15GHz step=1GHz MS MS_DB PS
      {Smith MP=(MS11,PS11).S11}
      {Polar MP=(MS21,PS21).S21};
end

Spec
   AC: FREQ: from 9GHz to 15GHz step=1GHz MS11_dB < -30;
end

Control
   Perturbation_Scale=1.0e-4;
   Optimizer=Minimax;
end
```

Comments

The first line of "bend1.ckt" is a comment:

```
! Empipe3D user-defined structure BEND1
```

OSA90 treats any text following an exclamation mark until the end of line as a comment.

Include Files

The first statement following the Model block header is

```
#include "bend1_osa/bend1.inc";
```

It instructs OSA90 to merge the contents of the file "bend1_osa/bend1.inc" into the body of the input file at the location where the #include directive appears.

The file "bend1.inc" is generated by Empipe3D. It contains the parameterization data for the waveguide bend. The use of include files is an efficient and convenient way of merging information stored in separate files.

Labels and Optimization Variables

The next statement in the Model block is

```
BEND1_d: ?0.05 0.2 0.35?;
```

This statement defines a label representing an optimization variable.

In the OSA90 input file, optimization variables are delimited by a pair of question marks.

If there is only one number between the question marks, as

label: ?*x*?;

then *x* represents the starting point of the variable. The lower bound is implicitly set to 0 if $x \geq 0$, or $-\infty$ if $x < 0$. The upper bound is set to $+\infty$ if $x \geq 0$, or 0 if $x < 0$.

If there are three numbers between the question marks, then they represent the lower bound, the starting point and the upper bound of the variable, respectively.

You can also define labels to represent numerical constants, vectors, matrices and formulas. For details see *OSA90/hope User's Manual*.

Element Model

The next statement in the Model block is

```
BEND1  1 2 0  model=7
   d=(BEND1_d * 1in);
```

It refers to the Empipe3D element named BEND1. Following the element name are three integers representing the connection nodes. Following the nodes are the parameters of the element. The parameter "model" specifies the type of interpolation (see Chapter 9). The parameter "d" is assigned an expression which converts the dimensionless value of the label BEND1_d to inches.

In general, you can assign parameter values by numerical constants, optimization variables, labels and expressions.

Port Definition

Ports of the circuit are defined by this statement in the Model block:

```
PORTS 1 0 2 0;
```

The waveguide bend is defined as a two-port. In general, for an *n*-port, the keyword PORTS is followed *n* pair of nodes, each pair (two consecutive nodes) defines one of the ports.

A related keyword PORT can be used to define a single port, followed by a pair of nodes.

The CIRCUIT Statement

The statement

```
CIRCUIT;
```

indicates the completion of the circuit definition. When the file parser encounters this statement, it checks the connections of the whole circuit and, if no error is found, generates the circuit response labels.

Defining Responses

OSA90 has a set of predefined circuit response labels, such as those listed in Table 7.2.

TABLE 7.2 RESPONSE LABELS

Label	Response
MS*ij*	magnitude of S_{ij}
PS*ij*	phase of S_{ij}
RS*ij*	real part of S_{ij}
IS*ij*	imaginary part of S_{ij}
RY*ij*	real part of Y_{ij}
IY*ij*	imaginary part of Y_{ij}
RZ*ij*	real part of Z_{ij}
IZ*ij*	imaginary part of Z_{ij}
GD*ij*	group delay

In addition to the scalar labels listed in Table 7.2, you can also use matrix labels by omitting the indices. For instance, MS represents the matrix of all MS*ij*, i.e., the magnitude of all S parameters. The dimension of such matrices is $n \times n$, where n is the number of ports.

From these predefined labels, other responses can be derived using expressions. For instance, the last two statements in the Model block of the file "bend1.ckt" define the S-parameter responses in dB:

```
MS_DB[2,2] = if (MS > 0) (20 * log10(MS)) else (NAN);
MS11_DB = MS_DB[1,1];
```

The first statement defines a 2 by 2 matrix MS_DB to be calculated from the built-in response label MS. The second statement establishes an alias for the matrix element MS_DB[1,1].

These statements are generated by Empipe3D according to the responses involved in the specifications you have defined for optimization.

Sweep Block

The Sweep block defines frequency range and simulation outputs. The Sweep block in the file "bend1.ckt" contains two statements. Each statement in the Sweep block defines a sweep set, with its own frequency range and response labels.

The first sweep set is defined as

```
AC: FREQ: from 9GHz to 15GHz step=1GHz MS11_dB
    {XSWEEP title="MS11_dB and Spec" X=FREQ Y=MS11_dB
     SPEC=(from 9GHz to 15GHz, < -30)};
```

The leading keyword AC indicates the simulation type as small-signal AC (the OSA90/hope option, if licensed, supports two additional types of simulation: DC and harmonic balance).

Following the keyword AC is the frequency range definition. After that is the response label MS11_dB. The next two lines define a graphical view, which is used to format the graphical display.

In generating the OSA90 input file, Empipe3D automatically selects the response(s) of interest (primarily the ones for which specifications are given) to be included in the Sweep block. The waveguide bend example has a specification on the response MS11_dB, therefore MS11_dB is selected as the response label for the first sweep set. The specification itself is also included in the graphical view.

The second sweep set is defined as

```
AC: FREQ: from 9GHz to 15GHz step=1GHz MS MS_DB PS
    {Smith MP=(MS11,PS11).S11}
    {Polar MP=(MS21,PS21).S21};
```

This set includes the response labels for the magnitude and phase of all the S parameters. This will enable you to select any of these responses for viewing in addition to MS11_dB.

Also included are two graphical view definitions for displaying S_{11} on the Smith Chart and S_{21} on the Polar Plot, respectively. If you wish to view other S parameters on the Smith Chart or Polar Plot, you can use these view definitions as a template.

The definition of graphical views is described in full detail in the *OSA90/hope User's Manual*.

Specification Block

The statements in the Spec block represent the specifications you have defined using Empipe3D. The syntax of these statements are quite self-explanatory.

The Spec block in the file "bend1.ckt" contains the only specification for that example:

```
AC: FREQ: from 9GHz to 15GHz step=1GHz MS11_dB < -30;
```

Control Block

The Control block allows you to modify the default setting of a number of operation control parameters of OSA90.

In the file "bend1.ckt", the Control block contains two statements:

```
Perturbation_Scale=1.0e-4;
Optimizer=Minimax;
```

The keyword "Perturbation_Scale" controls the step size for estimating gradients. The value used here is chosen for Empipe3D applications. Do not change it.

The keyword "Optimizer" can be used to select the desired optimization algorithm. The choice here is made by Empipe3D based on the types of specifications you have defined.

Other control options are also available. See *OSA90/hope User's Manual*.

7.4 OSA90 Menus

Near the bottom of the OSA90 window is the menu area, where the menu options are presented. The main menu looks like this:

```
OSA90> File Display Optimize Macro Sensitivity monteCarlo Learn
```

You can use the mouse or the cursor keys to move the cursor to highlight the different menu options. As you do so, the line immediately above the menu line displays a brief comment on the function of the highlighted option.

To select a menu option, you can click the left-hand mouse button on the desired option, or you can move the cursor to highlight the option, and then press the <Enter> key.

Some menu options lead to another level of menu options. The OSA90 menu hierarchy is summarized in Table 7.3.

TABLE 7.3 OSA90 MENU OPTIONS

Menu Option	Brief Description
OSA90.File	reads, edits, parses and saves files
OSA90.Display	calculates and displays responses and functions
OSA90.Optimize	initiates optimization
OSA90.Macro	operates OSA90 from a macro command file
OSA90.Sensitivity	calculates and displays parameter sensitivities
OSA90.MonteCarlo	performs statistical (Monte Carlo) analysis
OSA90.Learn	learns user inputs to create a macro command file
OSA90.Display.Xsweep	displays responses versus parameter sweeps
OSA90.Display.Parametric	displays parametric plots of two responses
OSA90.Display.Array	displays elements of arrays
OSA90.Display.Waveform	displays time-domain waveforms
OSA90.Display.Smith	displays Smith charts and polar plots
OSA90.Display.Visual	displays 3D visualization and contours
OSA90.MonteCarlo.Xsweep	displays statistical sweep responses
OSA90.MonteCarlo.Parametric	displays statistical parametric plots
OSA90.MonteCarlo.Histogram	displays histogram of individual responses
OSA90.MonteCarlo.RunChart	displays run chart of individual responses
OSA90.MonteCarlo.Yield	displays yield estimated by Monte Carlo analysis
OSA90.MonteCarlo.Sensitivity	displays yield versus parameter sweeps
OSA90.MonteCarlo.Max	displays histogram of the maximum errors
OSA90.MonteCarlo.Scatter	displays scatter diagram between two responses

Exit and Cancelation

In OSA90, the right-hand mouse button and the <Esc> key are used to exit from a menu or to cancel an operation.

If you click the right-hand mouse button in the menu area or pressing the <Esc> key, you will exit from the current menu and return to the next higher-up menu level, if one exists. If you do so at the main menu, you will exit from OSA90 (you will be prompted first to confirm your intention).

Pop-Up Windows

Some OSA90 menu options lead to a pop-up window like the one illustrated here:

```
Output form:    Graphical
Draw type:      Curves
Zoom scale:     <TAB> to set scale
Sweep set:      sweep set 1
Y-axis:         MS11
X-axis:         FREQ
Ready to go:

<ENTER> = go  or  <ESC> = cancel
Select item with <UP>/<DOWN>
<F1> help
```

You can accept the default setting by clicking on the "Ready to go" line or pressing the <Enter> key. If you wish to change an option, move the cursor to highlight that option and a message is shown near the bottom of the pop-up window indicating the type of action that you may take. Some are multiple-choice options, for which you can click the left-hand mouse button to see a list of the available choices or you can use the <Left> and <Right> cursor keys to toggle through the choices. Some options expect a numerical entry, in which case you simply type in an appropriate value.

On-Line Help

In many cases OSA90 provides on-line help for menu and window options. Try the <F1> key if you need on-line help.

7.5 OSA90 File Editor

OSA90 has a built-in full screen ASCII text file editor. From the OSA90 main menu, you can invoke the editor by selecting the menu option File.

The editor is integrated with the OSA90 input file parser. So, after you finish modifying the file and exit from the editor, the file parser is automatically activated to check the syntax. If any error is detected, a message will be displayed and you will be returned to the editor to make the correction immediately.

The editor features search and replace, cut and paste, undo, macros and more.

Editor Function Menu

In the editor, you can click the left-hand mouse button to activate a menu of the editor functions, as listed in Table 7.4. Click the left-hand mouse button on a function to select it. Click the right-hand mouse button to cancel the menu.

TABLE 7.4 EDITOR FUNCTION MENU

Function	Description
Exit from editor	exit the editor and invoke the input file parser
Terminate program	terminate the program and exit from OSA90
Help	request help messages
Generate report	generate report of simulation results
Append report	append simulation results to the report
Toggle file	toggle between the primary and secondary windows
Read a new file	read (load) a new file to the editor
Save file to disk	save the file or the marked block to a disk file
Search	search for a specified string
Replace	replace a string and with another string
Change directory	change the default path (working directory)
Print text	print the file or the marked block of text
Clear file buffer	discard the current file to start a new file
On-line Manual	request for on-line User's Manual
Trim file	delete trailing blanks throughout the file
New color map	change the display color map
New key map	change the key map
Exit and keep	exit the editor and keep the simulation results

Editor Function Keys

The default set of function keys defined for the OSA90 file editor is listed in Table 7.5. It is possible to re-map the function keys (see the *OSA90/hope User's Manual*).

TABLE 7.5 EDITOR FUNCTION KEYS

Key	Function
<Back Space>	delete the character to the left of the cursor
<F5>	change the default path (working directory)
<Ctrl-N>	discard the current file to start a new (empty) file
<Ctrl-Y>	change the display color map
<Del>	delete the character under cursor or the marked block
<Ctrl-W>	delete the word under cursor
<F6>	toggle between edit and read-only modes
<Ctrl-Home>	move cursor to the end of the line
<Esc>	terminate the program or cancel a command
<F7>	exit the editor and invoke the input file parser
<Ctrl-F7>	exit the editor and keep the simulation results
<Ctrl-Down>	move cursor to the end of the file
<Ctrl-Up>	move cursor to the beginning of the file
<F1>	request help messages
<Home>	move cursor to the beginning of the line
<Ins>	toggle between insert and typeover modes
<Ctrl-K>	change the key map
<Ctrl-D>	delete from cursor to the end of line
<F9>	begin/end the definition of an editing macro
<F10>	execute the macro defined
<F4>	request for on-line User's Manual
<F2>	mark/unmark a block of text for copying or moving
<F3>	copy the marked block, or paste the copied text
<Pg Dn>	move cursor down by one page
<Pg Up>	move cursor up by one page
<Ctrl-P>	print the file or the marked block of text
<Ctrl-I>	input (read in) a new file to the editor
<Ctrl-R>	replace a string and with another string
<Ctrl-G>	generate report of simulation results
<Ctrl-A>	append simulation results to the report
<Ctrl-S>	save the file or the marked block to a disk file
<Ctrl-B>	search backward for a specified string
<Ctrl-F>	search forward for a specified string
<F8>	toggle between the primary and secondary windows
<Ctrl-X>	delete trailing blanks throughout the file (trimming)
<Ctrl-U>	undo the last sequence of consecutive deletions
<Ctrl-Left>	move cursor to the left by one word
<Ctrl-Right>	move cursor to the right by one word

7.6 OSA90 Display

The OSA90 menu option "Display" allows you to view the responses calculated by the 3D Solver in a variety of formats. When you click on the "Display" menu option, it leads to another level of menu which appears as

```
OSA90.Display> Xsweep Parametric Array Waveform Smith Visual
```

The meaning of these display formats is summarized in Table 7.6.

TABLE 7.6 DISPLAY FORMATS

Option	Brief Description
Xsweep	displays one or more responses versus frequency or a parameter
Parametric	displays one or more responses versus another response
Array	displays elements of an array
Waveform	displays time-domain waveforms (for the OSA90/hope option)
Smith	displays Smith charts and polar plots
Visual	displays 3D visualization and contours

Xsweep Display

The menu option OSA90.Display.Xsweep provides the most typical format of presentation: a rectangular plot of one or more response labels versus the frequency or another parameter.

When you select Xsweep, a pop-up window will appear as

```
Output form:    Graphical
Draw type:      Curves
Zoom scale:     <TAB> to set scale
Sweep set:      sweep set 1
Y-axis:         MS21
X-axis:         FREQ
Ready to go:

<ENTER> = go  or  <ESC> = cancel
Select item with <UP>/<DOWN>
<F1> help
```

You can move the cursor to select and change the various options. For instance, you can choose Numerical output instead of Graphical, or you can change the Draw type from Curve to Point, etc.

Defining a Parameter Sweep

A very useful feature is to display responses versus model parameters. For instance, you may ask OSA90 to sweep the miter parameter "*d*" of the waveguide bend and then display the *S*-parameter responses versus the parameter "*d*".

First, you have to define the parameter sweep in the **Sweep** block of the OSA90 input file. For example:

```
Sweep
   AC: BEND1_d: 0.1 0.2 0.25 0.3
       FREQ: from 9GHz to 15GHz step=1GHz MS11_dB
       ...
end
```

The parameter label ("**BEND1_d**" in the example) must be followed by a colon ":".

You can assign a set of discrete values for the sweep parameter, as shown in the above example. You can also assign a set of uniformly spaced points with an interval:

```
BEND1_d: from 0.1 to 0.35 step=0.05
```

This translates into a set of values for **BEND1_d** as 0.1, 0.15, 0.2, 0.25, ..., 0.35.

You can also specify the number of divisions within the sweep interval:

```
BEND1_d: from 0.1 to 0.35 N=5
```

This translates into a step size of (0.35 - 0.1) / 5 = 0.05. Consequently, the parameter **BEND1_d** will be swept for these values: 0.1, 0.15, 0.2, ..., 0.35.

Parameter sweeps with exponential step size are also possible. See the *OSA90/hope User's Manual* for detail.

Selecting a Sweep Parameter for the X-axis

If you have defined a parameter sweep in addition to a frequency sweep, then you can select to display the response(s) versus either the frequency or the sweep parameter.

In the pop-up window for the **Xsweep** display, click on the "X-axis:" option. The available choices are presented from which you can make a selection.

Selecting Response Labels for the Y-axis

For sweep sets which contain multiple response labels, you can choose to display one of the responses or all of the responses. In the pop-up window for the **Xsweep** display, click on the "Y-axis:" option. The available choices are presented from which you can make a selection.

7.7 OSA90 Optimization

When you invoke the "OSA90.Optimize" menu option to start optimization, a pop-up window will appear:

```
Optimizer:                L1
objective Function:       L1
Number of iterations:     30
Accuracy of solution:     0.0001
Display option:           every iteration
Ready to go:

<ENTER> = go  or  <ESC> = cancel
Select item with <UP>/<DOWN>
<F1> help
```

The options in the pop-up window are summarized in Table 7.7.

TABLE 7.7 OPTIONS FOR OSA90.Optimize

Option	Available Choices
Optimizer	L1, L2, Minimax, Quasi-Newton, Random, Simplex, Conjugate Gradient, Huber, One-Sided L1, Simulated Annealing, Yield, Yield Huber
Objective Function	*depending on the choice of Optimizer*
Number of iterations	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 30, 50, 100, 999, 9999
Accuracy of solution	0.01, 0.001, 1.0E-4, 1.0E-5, 1.0E-6
Display option	every iteration, best iterations

Optimizer and Objective Function

The choices available for the Optimizer option and the corresponding choices of objective functions are listed in Table 7.8. By default, Empipe3D selects a suitable optimizer based on the type of specifications you have defined.

TABLE 7.8 OPTIMIZERS AND OBJECTIVE FUNCTIONS

Optimizer	Gradient-Based	Objective Function
L1	Yes	L1
L2	Yes	L2
Minimax	Yes	Minimax
Quasi-Newton	Yes	Generalized L2, Sum of Errors
Random	No	Generalized L2, Sum of Errors
Simplex	No	Generalized L2, Sum of Errors
Conjugate Gradient	Yes	Generalized L2, Sum of Errors
Huber	Yes	Huber, Huber+
One-Sided L1	Yes	One-Sided L1
Simulated Annealing	No	Generalized L2, Sum of Errors

Number of Iterations

This option in the pop-up window allows you to limit the maximum number of iterations. The optimizer will stop once this limit is reached even if a solution within the desired accuracy has not been found. If this happens, the best set of values of the variables (in terms of minimizing the objective function) is presented as the solution.

Accuracy of Solution

This option allows you to specify the desired accuracy of the solution. The optimization will stop when the step size separating the current step and the next step is smaller than the specified accuracy. The step size is relative to the norm of the variable vector.

Display Option

During optimization, OSA90 displays the iteration count and current value of the objective function on the screen. By default, this information is displayed at every iteration.

For example:

```
Iteration   1/30 Max Error=23.2428
Iteration   2/30 Max Error=22.0957
Iteration   3/30 Max Error=19.2298
Iteration   4/30 Max Error=9.69842
Iteration   5/30 Max Error=19.0108
Iteration   6/30 Max Error=1.45415
Iteration   7/30 Max Error=1.14845
Iteration   8/30 Max Error=1.14157
Solution  Max Error=1.14157
```

Of the iteration count, the first number is the current iteration and the second one is the maximum number of iterations specified. For example, 3/30 means that the current iteration is the 3rd out of a maximum of 30.

Sometimes it is possible to observe the value of the objective function getting larger than that of a previous iteration. In the example, the 5th iteration produces a higher objective function value than that of the 4th iteration. This is not unusual. It is a part of the optimization algorithm.

You can instruct the program to display only those iterations that lead to a better (smaller) objective function value than the previous ones. You can do so by changing the "Display option" to "best iterations". You may find this especially desirable for the random optimizer, since it usually requires a large number of random explorations which result in fluctuating objective function values.

7.8 Producing Plotter Files

OSA90 can produce HPGL (Hewlett-Packard Graphics Language) and PostScript plotter files for the graphical displays.

HPGL and PostScript files are ASCII files which can be directly sent to a compatible printer or plotter or imported into word processing programs.

You can request an HPGL or PostScript file only when the OSA90 window is displaying graphics. Press the key <Ctrl-P>. A pop-up window will appear, as shown in Fig. 7.2.

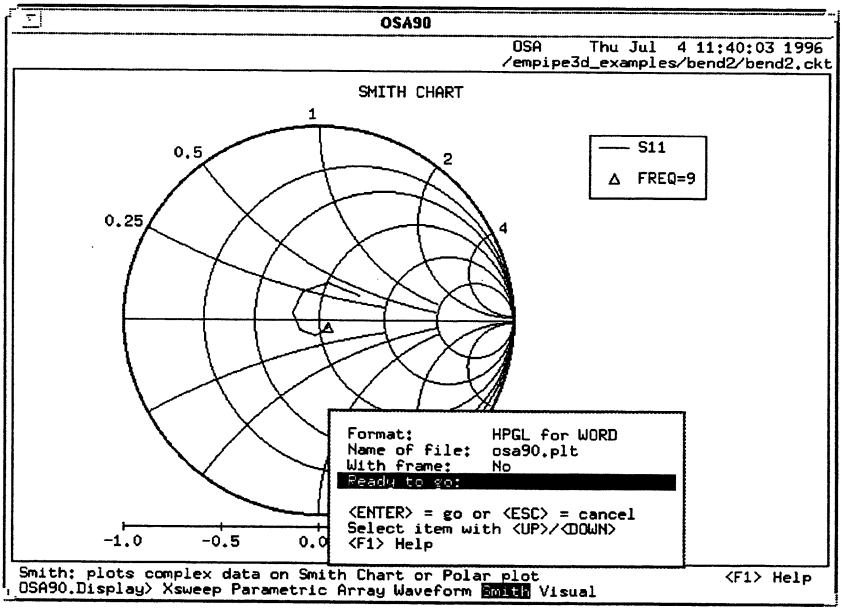


Fig. 7.2 OSA90 window for producing a plotter file.

Selecting the File Format

The "Format" option in the pop-up window allows you to select from the available formats listed in Table 7.9. The differences between the HPGL formats are their interpretation of colors and font sizes.

TABLE 7.9 PLOTTER FILE FORMATS

Format	Description
HPGL for WORD	HPGL file to be imported into Microsoft Word
HPGL for WP	HPGL file to be imported into WordPerfect
HPGL 6-Pen Plotter	HPGL file for HP 6-pen plotter
PS Portrait	PostScript file, portrait orientation
PS Landscape	PostScript file, landscape orientation

Plotter File Name

The option "Name of file" in the pop-up window allows you to specify a name for the plotter file. The default file name is "osa90.plt". To change it, point the cursor to the option "Name of file" and type in the desired file name.

Including or Excluding the Frame of the Graphical Display

The graphical display on the screen has a frame (white border lines). You can include or exclude this frame from the plotter file by choosing "Yes" or "No" for the "With frame" option. The default is "No", i.e., the frame is excluded from the plotter file.

HPGL for 6-Pen Plotter

For HP 6-pen plotters, if you wish the colors to closely match the colors displayed on the screen, you should load the plotter pen carousel in the following color sequence (pen 1 to pen 6): black, red, green, blue, yellow and pink.

8

Empipe3D Database

8

Empipe3D Database

8.1 Database Index

Empipe3D employs a database system to avoid duplicate EM analyses. Every time Empipe3D invokes the 3D Solver, the simulation results are stored in a database, together with the corresponding parameter values. Subsequently, when EM analysis results are needed, Empipe3D checks the database first.

Empipe3D database file names have the form of

element_i.dbs

where *element* identifies the structure and *i* is an integer index. Database files are stored in a subdirectory of the name "*element*_osa" within the current working directory.

For example, for the tutorial in Chapter 5, the database file name is wgrad_1.dbs and it is stored in the subdirectory "empipe3d_examples/wgrad/wgrad_osa".

The database index is used to distinguish different versions of the same structure. The default index is 1.

Normally, there are few reasons to create a new database index. One reason may be to compare different versions of the solver by storing the results in separate database files.

The only way to change the database index is to modify the OSA90 input file. Locate within the input file the Empipe3D element reference and add the INDEX parameter.

Example:

```
WGRAD  1 2 0  model=1
   INDEX=2
   ...
```

The database created after this modification will be named wgrad_2.dbs.

You can disable the database mechanism entirely by assigning INDEX = 0. This instructs Empipe3D to ignore the database and always invoke the 3D Solver for EM simulation. Furthermore, the EM simulation results will not be saved in a database.

8.2 Converting Database to ASCII Data File

As part of the Empipe3D package, a utility program called dbs2dat is provided to facilitate converting Empipe3D database files to ASCII data files.

Usage:

```
dbs2dat database datafile
```

where *database* denotes the name of an Empipe3D database file and *datafile* is the name of the output file which will contain the converted data.

Example:

```
dbs2dat bend1_1.dbs bend1.dat
```

where bend1_1.dbs is the database file for the tutorial in Chapter 3. The converted data written to bend1.dat is partially shown as follows.

```
Data From File bend1_1.dbs (Last Modified on  Jun 28 19:03:53 1996)
Empipe3D

N_PARS 1
N_FREQS 7
N_PORTS 2

 0.1
   9 0.186917 -49.2323 0.982376 -139.709 0.982376 -139.709 0.186917 -50.1858
   10 0.281397 -109.284 0.959592 160.347 0.959592 160.347 0.281397 -110.022
   11 0.380162 -162.148 0.92492 107.675 0.92492 107.675 0.380162 -162.502
   12 0.491302 147.252 0.87099 57.2164 0.870989 57.2164 0.491302 147.181
   13 0.614729 96.6448 0.788739 6.66031 0.788739 6.66031 0.614729 96.6758
   14 0.741078 44.6906 0.671419 -45.3146 0.671419 -45.3146 0.741078 44.6802
   15 0.851659 -9.31611 0.524096 -99.3675 0.524096 -99.3675 0.851659 -9.41884

 0.2
   9 0.127494 -46.819 0.991841 -136.811 0.991838 -136.811 0.127494 -46.8031
   10 0.184463 -104.839 0.982839 165.166 0.982839 165.166 0.184463 -104.829
   11 0.237751 -155.014 0.971326 114.939 0.971326 114.939 0.237751 -155.108
   12 0.291917 157.823 0.956428 67.7232 0.956459 67.7232 0.291917 157.624
   13 0.347959 111.482 0.93751 21.3804 0.93751 21.3804 0.347959 111.279
   14 0.404822 64.5394 0.914396 -25.5461 0.914396 -25.5461 0.404822 64.3684
   15 0.459344 15.5981 0.88826 -74.4548 0.888257 -74.4548 0.459344 15.4924

 0.3
   9 0.0206621 -40.5676 0.999786 -133.36 0.999786 -133.36 0.0206621 -46.1585
   10 0.0197056 -95.8704 0.999806 170.495 0.999806 170.495 0.0197056 -103.14
...
...
```

In the data file, a header line identifies the source database, followed by three lines listing the number of parameters, the number of frequencies and the number of ports, respectively.

The data is divided into data sets, separated by a blank line. Each data set begins with a list of parameter values, followed by the *S* parameters. For example, the first data set in bend1.dat is

```
0.1
 9 0.186917 -49.2323 0.982376 -139.709 0.982376 -139.709 0.186917 -50.1858
 10 0.281397 -109.284 0.959592 160.347 0.959592 160.347 0.281397 -110.022
 11 0.380162 -162.148 0.92492 107.675 0.92492 107.675 0.380162 -162.502
 12 0.491302 147.252 0.87099 57.2164 0.870989 57.2164 0.491302 147.181
 13 0.614729 96.6448 0.788739 6.66031 0.788739 6.66031 0.614729 96.6758
 14 0.741078 44.6906 0.671419 -45.3146 0.671419 -45.3146 0.741078 44.6802
 15 0.851659 -9.3161 0.524096 -99.3675 0.524096 -99.3675 0.851659 -9.41884
```

The first line shows the value(s) of the parameter(s). In this example there is only one parameter (see Chapter 3).

The next seven lines correspond to the seven frequencies. The data on each line represents

```
FREQ MS11 PS11 MS21 PS21 MS12 PS12 MS22 PS22
```

where FREQ denotes the frequency (in GHz), MS*ij* denotes the magnitude of S_{ij} and PS*ij* denotes the phase of S_{ij} (in degrees).

8.3 Creating Database from Data File

A utility program called **dat2dbs** is included in the **Empipe3D** package. It can be used to create an **Empipe3D** database file from an ASCII data file.

Usage:

```
dat2dbs datafile database
```

where *datafile* denotes the name of an existing ASCII data and *database* is the name of the database file to be created.

Example:

```
dat2dbs bend1.dat bend1_2.dbs
```

where **bend1.dat** is produced by **dbs2dat** from **bend1_1.dbs**, as described in Section 8.2. The resulting database **bend1_2.dbs** should contain the same data as in **bend1_1.dbs**.

In general, the data file must contain a header like this

```
N_PARS  n1
N_FREQS n2
N_PORTS n3
```

where *n1*, *n2* and *n3* denote the number of parameters, the number of frequencies and the number of ports, respectively.

Editing Empipe3D Database

By combining the two utility programs **dbs2dat** and **dat2dbs**, you will be able to edit the contents of an **Empipe3D** database in three steps.

1. Use **dbs2dat** to convert an existing database to a data file.

2. Edit the data file using a text editor. You can delete unwanted data sets to trim down an oversized database. You can manually expand the database by adding new sets of *S* parameters produced by the 3D Solver. You can even modify the numbers, but that is cheating.

3. Use **dat2dbs** to convert the edited data file to a new database, perhaps using an index different from the original database.

9

Response Interpolation

9

Response Interpolation

9.1 Introduction

The 3D Solvers use variable meshing, theoretically, once a structure is parameterized, Empipe3D can drive the 3D Solver with arbitrary parameter values. Increments in parameter values do not have to conform to a fixed grid.

The use of response interpolation, therefore, is not out of necessity but in order to improve efficiency.

Each parameter of an Empipe3D element is associated with an interpolation interval. During optimization, the 3D Solver is invoked for an EM simulation only if at least one parameter has been moved by the optimizer out of its previous interpolation interval. For small changes in parameter values, interpolation is applied to obtain the *S* parameters. This can substantially reduce the total number of calls to the 3D Solver needed for optimization.

Another important benefit of instituting an interpolation interval is numerical stability. For instance, Empipe3D estimates the gradients by extracting the difference between two sets of responses at slightly different parameter values. If the field solver is invoked to simulate two structures with slightly different dimensions, the difference in the responses cannot be reliably extracted due to the inherent numerical "noise". Such a problem can be especially pronounced when variable meshing is involved. In this sense, the use of interpolation has a smoothing effect.

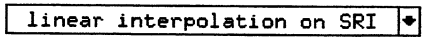
The syntax of defining an interpolation interval is illustrated in the tutorials of Chapters 3 to 5, as well as in the descriptions of Geometry Capture in Chapter 6.

A good choice of the interpolation interval is obviously problem dependent. If the interval is too large (i.e., the number of divisions is too small), then the interpolated *S* parameters may not be accurate. On the other hand, a very small interpolation interval will increase the number of 3D EM simulations and ultimately defeat the purpose of using interpolation.

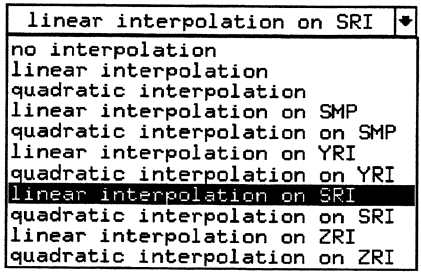
A possible approach to a problem for which you are not sure about the best interpolation interval size is to start with a coarse (large) interval and gradually refine the interval size as necessary.

9.2 Types of Interpolation

The "Empipe3D Specifications" window has a box for selecting the interpolation type:



If you click on the arrow a list of available choices is displayed:



The interpolation type is also indicated in the OSA90 input file (netlist) by the Empipe3D element parameter MODEL, as summarized in Table 9.1.

TABLE 9.1 EMPIPE3D ELEMENT PARAMETER: MODEL

Value	Description
0	interpolation disabled
1	linear interpolation based on *S* parameters, magnitude and phase
2	quadratic interpolation based on *S* parameters, magnitude and phase
3	linear interpolation based on *S* parameters, magnitude and phase
4	quadratic interpolation based on *S* parameters, magnitude and phase
5	linear interpolation based on *Y* parameters, real and imaginary parts
6	quadratic interpolation based on *Y* parameters, real and imaginary parts
7	linear interpolation based on *S* parameters, real and imaginary parts
8	quadratic interpolation based on *S* parameters, real and imaginary parts
9	linear interpolation based on *Z* parameters, real and imaginary parts
10	quadratic interpolation based on *Z* parameters, real and imaginary parts

Example:

```
BEND1   1 2 0   model=7
   ...
```

This means that linear interpolation on *S* parameters in the rectangular form is selected for the Empipe3D element "BEND1".

Currently, the choices 1 and 3 are equivalent, and so are the choices 2 and 4. This is done with the intention of replacing the choices 1 and 3 with a sophisticated automatic selection scheme planned for a future release.

Disabling the Interpolation Feature

The response interpolation feature can be entirely disabled by setting MODEL=0. In this case, if the *S*-parameter responses are needed (and are not already saved in a database), Empipe3D will invoke the 3D Solver regardless of the parameter values.

This can be useful for verifying the solution of an optimization (see the tutorial in Chapter 4 for an example).

9.3 Linear and Quadratic Interpolations

The interpolation intervals of all the parameters form a multidimensional interpolation grid. A point is on-grid if all the parameter values are exact multiples of their respective interpolation intervals. The EM analysis of on-grid points are done directly by the 3D Solver.

A point if off-grid if at least one of the parameter values falls in the interior of its interpolation interval.

Linear interpolation of an off-grid point requires EM analyses at $n + 1$ base points, where n is the number of parameters whose values are off-grid. The placement of the base points is illustrated in Fig. 9.1 for the case of $n = 2$.

Linear interpolation is adequate when the grid size is sufficiently small that the variation of the response of interest within the interpolation interval can be approximated with reasonable accuracy by a linear function.

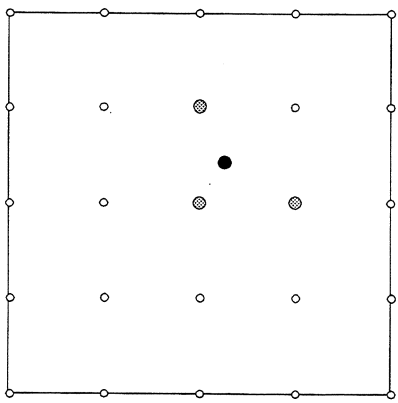


Fig. 9.1 The base points (shaded) needed for linear interpolation of an off-grid point (solid).

Quadratic Interpolation

Quadratic interpolation generally provides more accurate results than linear interpolation, at the expense of increased computational effort. Quadratic interpolation of an off-grid point requires EM analyses at $2 \times n + 1$ base points, where n is the number of parameters whose values are off-grid. The placement of the base points for quadratic interpolation is illustrated in Fig. 9.2 ($n = 2$).

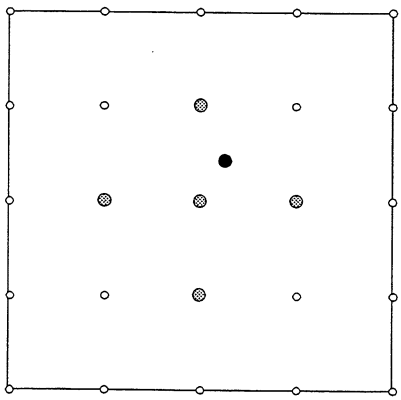


Fig. 9.2 The base points (shaded) needed for quadratic interpolation of an off-grid point (solid).

9.4 Choosing S, Y or Z Parameters

In using interpolation, a problem may arise when the response function to be interpolated has a local minimum or maximum with respect to the frequency and its location on the frequency axis shifts significantly between two base points.

Consider the illustration in Fig. 9.3. The two solid lines represent the response MS11 at two base points. The response has a local minimum with respect to the frequency and its location changes from 8.53 GHz for one base point to 8.72 GHz for the other base point. A linear interpolation applied to these two base points leads to the erroneous result depicted by the dashed line in Fig. 9.3.

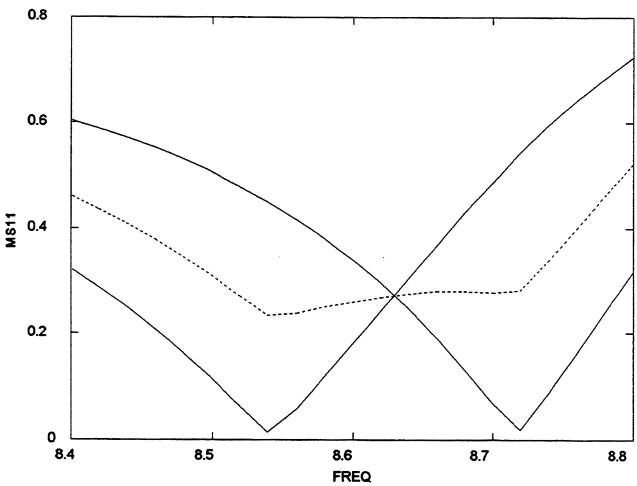


Fig. 9.3 A problem with interpolation.

When encountering this problem, you can reduce its severity by using a smaller interpolation interval.

An alternative remedy is to choose the *Y* or *Z* parameters instead of the *S* parameters as the response functions used in the interpolation. This can solve the problem if the *Y* or *Z* parameters do not contain any shifting minima or maxima. The most significant advantage of this solution is that it does not require any new field solutions. Empipe3D first converts the *S* parameters of the base points to the *Y* or *Z* parameters, performs the interpolation and then converts the results back to *S* parameters.

Currently, choosing among the *S*, *Y* and *Z* parameters for interpolation can only be done manually on a trial and error basis. An automated scheme is under development.

Index

OSA

Dundas, Ontario, Canada