RSOFT
Design Group

USER GUIDE

FullWAVE™

THE SOURCE FOR
PHOTONIC & NETWORK
DESIGN SOFTWARE

THIS PAGE IS INTENTIONALLY LEFT BLANK

FullWAVE 6.3

RSoft Design Group, Inc.
400 Executive Blvd. Ste 100
Ossining, NY 10562 USA

ph: 1.914.923.2164
info@rsoftdesign.com

www.rsoftdesign.com

FullWAVE 6.3

Revision 110919

RSoft Design Group, Inc. warrants that under normal use, the physical goods (installation disc, hardlock key, and documentation) will be free of material defects for a period of thirty days from the date of purchase. Upon written notice, RSoft Design Group Inc. will replace any defective media. While every precaution has been taken in the preparation of this documentation, RSoft Design Group Inc. assumes no responsibility for errors or omissions. No other warranty, either expressed or implied, is provided with this software. No liability for damage to equipment or data, or any other liability, is assumed by RSoft Design Group, Inc.

Contents

Preface 1

Foreword 1

How to Read This Manual 2

1 Introduction 5

1.A. Program Installation & System Requirements 5

1.B. Physical Conventions 5

1.C. Program Executables 6

1.D. Program Operation 7

1.D.1 GUI Program Operation 7

1.D.2. CLI Operation 9

1.E. Example Files 9

1.F. README File 9

1.G. Technical Support & Software Upgrades 9

2 Background 11

2.A. The FDTD Algorithm 11

2.B. The FullWAVE FDTD implementation 13

2.B.1. Physical Parameters 13

2.B.2. Numerical Parameters 13

2.C. Additional Comments 15

2.D. References 15

3 Using FullWAVE 17

3.A. Spatial Domain, Grid, and FDTD Options 17

3.A.1. Domain & Grids....18
3.A.2. FDTD Options....18
3.B. Default Boundary Conditions....19
3.C. Time Domain & Grid....19
3.D. Choosing an Excitation....20
3.E. Saving Data From a Simulation....21
3.E.1. Monitor Options....21
3.E.2. Additional Output Options....23
3.F. Displaying Data During a Simulation....24

4 Excitation Types 27

4.A. Basic Excitation Definition....27
4.B. The Launch Parameters Dialog....28
4.C. Spatial Excitation....29
4.C.1. Choosing a Source Type....29
4.C.2. Defining the Field Profile Type....30
4.C.3. Using Optional Field Profile Data....33
4.C.4. Launching at an Angle....35
4.C.5. Viewing Launch Field....37
4.D. Temporal Excitation Types....37
4.D.1. CW....37
4.D.2. Pulse....38
4.D.3. Impulse....39
4.D.4. Custom Time Envelopes....40
4.D.5. Setting Optional Temporal Options....40
4.E. Polarization Manipulation....41
4.F. Launching Multiple Fields....43
4.G. Special Source Properties....43
4.H. Launching a Plane Wave....44
4.H.1. Using Periodic Boundary Conditions....44
4.H.2. Using PML Boundary Conditions....44
4.H.3 Other Options....45
4.I. Using an Enclosed Launch....45

5 Monitoring and Analysis 47

5.A. Monitor Overview....47
5.A.1. Main Types of Monitor Output....47
5.A.2. Enabling Monitor Output....48
5.B. Monitor Position & Geometry....48

5.B.1. Monitor Geometry Overview 49
5.B.2. Specifying Monitor Position 50
5.B.3. Specifying Monitor Size & Orientation 51
5.B.4. Specifying Monitor Grid for Calculation & Storage 52
5.B.5. Additional Settings 52
5.C. Single-Value Output Types 53
5.C.1. Field Values at a Single Point 53
5.C.2. Power 54
5.C.3. E Density 54
5.C.4. H Density 55
5.C.5. T Density 55
5.C.6. Absorption 55
5.C.7. Overlap 55
5.C.8. Time-Averaging Single-Value Output 57
5.C.9. Combining Multiple Single-Value Output Types 58
5.C.10. Absorption & E/H/T Density Results for Specific Materials 58
5.D. Spatial Output Types 59
5.D.1 Field Output over Monitor Domain 59
5.D.2. Power 60
5.D.3. E Density 60
5.D.4. H Density 60
5.D.5. T Density 61
5.D.6. Poynting Vector 61
5.D.7. Absorption 61
5.D.8 Spatial Output Format 62
5.D.9 Time-Averaging Spatial Output 62
5.E. FFT Frequency Analysis 62
5.E.1. Enabling FFT Output 62
5.E.2. FFT Output Files & Units 63
5.E.3. Memory Intensive FFT Output 64
5.E.4. FFT Usage Guidelines 64
5.F. DFT Frequency Analysis 65
5.F.1. Enabling DFT Output 65
5.F.2. DFT Output Files & Units 66
5.F.3. Using DFT Output for Steady-State Analysis 66
5.F.4. DFT Usage Guidelines 67
5.G. Typical Monitor Configurations 67

6 Advanced Features 71

6.A. Anisotropy, Non-Linearity, Dispersion, and Metals........71
6.B. Computing Far Fields........72
6.B.1. Single-Plane Calculation........72
6.B.2. Multi-Plane Calculation........74
6.C. Boundary Conditions........75
6.C.1. PML Boundaries........76
6.C.2. Periodic Boundaries........77
6.C.3. Symmetric & Anti-Symmetric Boundaries........78
6.C.4. PEC Boundaries........79
6.D. Saving a State File & Continuing a Completed Simulation........80
6.D.1. Saving a State File & Continuing a Simulation........80
6.D.2. Using a State File for Further Post-Processing........80
6.D.3. Manually Working With State Files........81
6.D.4. Viewing a State File........81
6.E. Additional Simulation Algorithms........81
6.E.1. Single vs. Double Precision........81
6.E.2. Radial FDTD........82
6.E.3. Complex FDTD........83

7 Clustering 85

7.A. Licensing, Installation, & Testing........85
7.B. Cluster Settings........85
7.C. Cluster Display Settings........88
7.D. Additional Cluster Settings........88
7.E. Running Cluster Simulations from the Command Line........89

8 Cavity Modes & Q Factors 91

8.A. Theoretical Background........91
8.A.1. Finding Cavity Resonances & Mode Profiles........92
8.A.2. Calculating *Q*........93
8.B. Manually Calculating Cavity Modes and *Q* Factors........95
8.B.1. Manual Calculation of Cavity Resonances........96
8.B.2. Manual Calculation of Mode Profile........96
8.B.3. Manual Calculation of *Q*........97
8.C. Using Q-Finder to Calculate Cavity Modes & *Q* Factors........97
8.C.1 Preparing an .ind File For Use With Q-Finder........98
8.C.2. The Q-Finder dialog........99
8.C.3. Performing a Simulation........104
8.C.4. Output Files........105

8.C.5. Advanced Q-Finder Techniques ... 106

9 Basic Tutorials 109

Basic Tutorial 1: Basic 2D Simulation ... 109
Basic Tutorial 2: Basic 3D Simulation ... 125

10 Advanced Tutorials 131

Tutorial 1: Ring Resonator ... 131
Device Layout: ... 131
Simulation: Pulsed Excitation ... 136
Simulation: CW Excitation ... 139
Tutorial 2: Transmission Spectra of Semi-Infinite Square PBG Lattice ... 141
Structure Overview ... 141
Running the Simulation ... 142
Areas for Further Exploration: ... 142
Tutorial 3: PBG Y-Branch Power Splitter ... 145
Structure Overview ... 145
Running the Simulation ... 146
Areas for Further Exploration: ... 146
Tutorial 4: Manually Computing Cavity Modes and *Q* Factors of a PBG Cavity ... 147
Structure Overview ... 147
Creating the Structure in the CAD Environment ... 147
Computing the Cavity Resonances ... 150
Computing a Cavity Mode Profile ... 151
Computing the *Q* Factor ... 153
Areas for Further Exploration ... 156
Tutorial 5: Incorporating Material Dispersion ... 157
Method 1: Dispersion via the Single Value Method ... 157
Method 2: Dispersion via the Expression Method ... 158
Method 3: Dispersion via the Data File Method ... 160
Method 4: Dispersion via the Material Editor ... 162
Summary ... 163
Areas for Further Exploration ... 163
Tutorial 6: Incorporating Non-Linearities ... 165
Device Layout ... 165
Exploring the Effect of the Non-Linearity ... 167
Tutorial 7: Using Q-Finder to Automate Cavity Mode and *Q* Calculations ... 169
Structure and Simulation Overview ... 169
Using Q-Finder ... 170

Basic Scanning and Optimization.. 171
Areas for Further Exploration 172
Tutorial 8: Using a Non-uniform Grid 175
Structure Overview 175
Advantages of Using a Non-Uniform Grid for this Structure 175
Using a Non-Uniform Grid 176
Simulating with a Non-Uniform Grid 177
Simulating with a Uniform Grid 178
Comparing the Results 178
Areas for Further Exploration: 179
Tutorial 9: Modelling Materials with Optical Gain 181
Structure Overview 181
Performing a Simulation 181
Tutorial 10: Simulating Surface Plasmon Effects 183
Example 1: Simple Plasmonic Waveguide 183
Example 2: Propagation through a Periodic Array of Metal Slits 184
Tutorial 11: Anisotropic Wave-Plate 187
Structure Overview: 187
Performing a Simulation 188
Areas for Further Exploration 190
Tutorial 12: Negative Refractive Index 191
Example 1: Negative Prism 191
Example 2: Negative Lens 193

Appendix A Tips and Traps 195

A.A. Common FullWAVE mistakes 195
A.B. Some Good FullWAVE habits to learn 196
A.C. Tips for Improving Simulation Speed 197
A.D. Using the Command Line 198

Appendix B FullWAVE File Formats 201

B.A. Standard RSoft File Format 201
B.B. Monitor File Formats 201
B.B.1. Single-Value Output File 201
B.B.2. Spatial Output Files 202
B.B.3. FFT Output Files 202
B.B.4. DFT Output Files 203

Appendix C Symbol Table Variables 205

C.A. FDTD Simulation Parameters 205
C.B. Advanced Grid Parameters 206
C.C. Launch Parameters 206
C.D. Display/Output Options 207
C.E. Advanced Parameters 208
C.F. Advanced Features 209

Appendix D Units & Nonlinearity 211

Appendix E Release Notes 223

Index 235

Preface

Foreword

The FullWAVE simulation engine is a part of the RSoft Photonic Suite, and is based on the well-known finite-difference time-domain (FDTD) technique. It calculates the electromagnetic field as a function of time and space in a given index structure in response to a given electromagnetic excitation. FullWAVE is ideal for studying the propagation of light in a wide variety of photonic structures.

FullWAVE is fully integrated into the RSoft CAD environment which allows the user to define the material properties and structural geometry of a photonic device. FullWAVE can accurately model both dielectric and magnetic materials, as well as anisotropic, dispersive, and non-linear materials. It also can utilize a computing cluster in order to achieve both larger and faster simulations.

To use FullWAVE effectively, it is critical to have a working knowledge of the RSoft CAD interface. The CAD tool is described in full in the RSoft CAD manual, which is included in the FullWAVE package. The reader is strongly encouraged to study the CAD manual before reading beyond Chapter 3 of this manual.

As a member of the RSoft Photonics Suite, FullWAVE is designed to work with RSoft's other passive device simulation modules. This modular approach to the design and simulation of photonic devices is one of RSoft's Photonic Suite's greatest strengths. Each program in the suite is designed to "play nice" with the other programs, creating an environment in which data can be shared between the modules. Virtually all the input and output files are in a simple ASCII text format, which allows even greater user control over program operation as well as third-party programs to be integrated into the suite.

While the RSoft Photonics Suite is designed to be used via the GUI (Graphical User Interface), command line operation is also possible. This, coupled with the modularity of the Suite, allows for complex scripting capability. The Suite is not limited to a single scripting language, but rather uses the native scripting language of your operating system. For example, Windows users can use DOS batch files, while Linux users can use bash scripts. Additionally, users familiar with languages such as Perl, Python, C, or C++ can create custom scripts in these languages. The RSoft Photonics Suite provides the best of both worlds: it allows for simulations to be performed via the GUI, and for complicated custom simulations to be performed via a script. New and advanced users alike are able to realize the full power of the Suite.

How to Read This Manual

While this manual can (and should!) be used as a reference manual at times, it is recommended that you read the first five chapters in their entirety to get a firm foundation from which to interpret the rest of the documentation.

If you are a new user, please reject the notion that you can simply turn to a section of interest in this manual. It can be hard to resist this urge, but doing so will reward you with better usage habits, a fuller understanding of how the software works, and will help you from suffering from common usage problems. Furthermore, we recommend that you look through the RSoft CAD manual before proceeding too far into this manual as it provides a foundation for creating design files for use with FullWAVE.

How is This Manual Organized?

The manual can be logically split into several main parts:

Introductory Information

Chapters 1 and 2 provide an overview of the installation of the program, an introduction to the program and its components, information about technical support and product upgrades, and background of the FDTD algorithm. All new users should read these chapters in their entirety.

Basic FullWAVE Usage

Chapters 3-5 explain the basic usage of FullWAVE including setting FullWAVE simulation parameters, choosing a launch condition, output and display options and measurement and analysis. All new users should read these chapters in their entirety.

Advanced FullWAVE Usage

Chapters 6-8 cover advanced topics such as including advanced material systems, far-field output, non-default boundary conditions, clustering, and cavity modes and *Q*-Factors. These chapters can be treated as a reference manual and do not need to be read by all new users.

Tutorials

Chapters 9-10 contain tutorial examples that span both basic and advanced FullWAVE topics. The basic tutorials (Chapter 9) are recommended for new users.

Appendices

The appendices provide detailed discussion of file formats and other relevant information. The appendices can be treated as a reference manual and do not need to be read by all new users.

Note that this manual should be used alongside the RSoft CAD manual as described in the next section.

Where is the Documentation For…

The documentation for the RSoft Photonics Suite is divided into several manuals. The manuals are structured using a simple rule:

Anything defining geometry and/or material parameters is in the CAD manual. Anything else is in an appropriate simulation manual.

Using this rule, almost any topic can be found. As with any rule, there are a few exceptions. The major exceptions are:

- *Installation*

  The installation procedure for RSoft software, including the CAD and all simulation modules, is covered in the separate document entitled "RSoft Installation Guide". Specific installation instructions for FullWAVE can be found in Chapter 1 of the FullWAVE manual (this document).

- *Using a Non-Uniform Grid*

  The creation of a non-uniform grid is covered in Chapter 9 of the CAD manual.

- *Material Editor*

  The definition of advanced materials via the Material Editor is covered in Chapter 8 of the CAD manual.

- *Parameter Scanning/Scripting/Batch Operation*

  These topics are very similar, and are shared by all the simulation modules. They are discussed in Chapter 8 of the CAD manual.

- *Computing the Index Profile*

  Computing the index profile is discussed in the CAD manual.

- *Computing Transverse Mode Profiles*

  Computing transverse mode profiles via the BPM-based mode solvers (which can be used without a license for BeamPROP) as well as TmmSIM is discussed in the CAD manual.

- *Pathways*

Pathways define the location and geometry of launch, or initial fields in BeamPROP and FullWAVE. They are documented in the CAD manual.

- *Command Line Utilities*

  The RSoft Photonics CAD Suite ships with several command line utilities which perform a variety of tasks. These utilities are documented in an Appendix in the CAD manual..

- *RSoft Expressions*

  Virtually any numeric field in any RSoft product can accept an analytical expression involving pre-defined and user-defined variables. The form of these expressions, including valid arithmetic operators and functions can be found in an Appendix in the CAD manual.

Anytime this rule is violated, a note will direct the reader to the proper section in the proper manual.

Where are these manuals located?

All documentation is placed on your computer during the program installation. Online versions can be accessed through the RSoft CAD via the Help menu item, or the two help buttons on the right of the top toolbar. The actual files can be found in the subdirectory `help` in the installation directory. Additionally, PDF versions can be found in the subdirectory `docs`. These files require the Adobe Acrobat Reader, which can be obtained from Adobe (www.acrobat.com) at no charge.

Typographical Conventions

A number of typeface and layout conventions are followed in this manual.

- The *names* of fields and controls in the GUI dialogs are written in **boldface**
- The *values* of pull-down menus and radio button controls are written in *italics*.
- File names and paths, symbol table variables and values, expressions typed in GUI edit fields, and code snippets are written in `monospace`.
- In referring to example CAD files, the installation directory for the CAD tool is specified as `<rsoft_dir>`, and should be replaced with the correct value for your installation.

1

Introduction

This chapter explains the installation procedure for FullWAVE, a discussion of physics conventions used in this manual, information on running the program, and notes about getting product updates and technical support.

1.A. Program Installation & System Requirements

The installation process is outlined in the RSoft Installation Guide and is not repeated here. The Installation Guide can be found in electronic form on the distribution CD or downloaded from RSoft's support website.

System Requirements

Most RSoft products have 32- and 64-bit versions for both Windows and Linux systems. Please check the 'System Requirements' section of the relevant product on RSoft's website (www.rsoftdesign.com) for a list of the specific OS versions we currently support.

Note that once a product has been successfully installed, it may not function correctly after changes such as OS updates or other software/hardware modifications are made to the computer system. In such cases, RSoft will attempt to resolve any issues for customers that have a current annual maintenance contract with RSoft, but does not guarantee success.

1.B. Physical Conventions

As with any branch of science, there are a number of concepts in the study of photonic devices for which there exist several different definitions exist in the literature. There are the conventions adopted in FullWAVE.

Units

The units used in are as follows:

- The standard unit of length is measured in microns [μm].
- The angular unit used is degrees.
- The units of imaginary refractive index are defined as:

$$n_{imag} = \frac{\gamma\lambda}{4\pi}$$

where λ is the wavelength and γ is the usual exponential loss coefficient defined such that the power decays as $e^{-\gamma z}$ and is given in units of μm^{-1}.

Polarization

Polarization is defined, in terms of the **E** field, as follows in FullWAVE:

Simulation Type	TE	TM
2D (in XZ plane)	Ey	Ex
3D	Ex	Ey

Please consult Appendix G in the CAD manual for a more detailed discussion.

1.C. Program Executables

The directory `<rsoft_dir>\bin\` contains many executable files. The following is a list of the executable for the RSoft CAD, FullWAVE, the WinPLOT graphing tool, and MOST, RSoft's scanning and optimization tool.

Product	Windows Name	Linux Name
The RSoft CAD tool	`bcadw32.exe`	`xbcad`
BeamPROP simulation tool	`bsimw32.exe`	`xbeam`
FullWAVE simulation tool	`fullwave.exe`	`xfullwave`
WinPLOT graphing tool	`winplot.exe`	`xplot`
MOST tool	`rsmost.exe`	`xmost`

These executable have different names under Windows and Linux though in most documentation the Windows names are used. Also, these programs can be run from the command line; more information on this can be found in the next section.

1.D. Program Operation

A FullWAVE Simulation can be started via either the Graphical User Interface (GUI) or a Command Line Interface (CLI). Both of these methods are equally powerful, and are useful in different circumstances. Beginning users are encouraged to use FullWAVE via the GUI interface in order to understand the usage of the program. Advanced users who wish to incorporate FullWAVE into scripts/batch operations are encouraged to use the scripting capabilities of FullWAVE.

1.D.1 GUI Program Operation

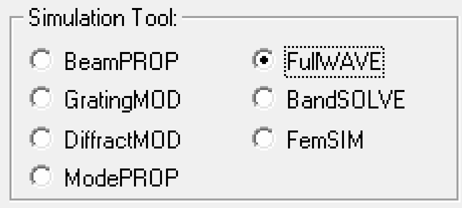
The GUI operation of FullWAVE is a very straightforward process, and can be broken down into 5 basic steps:

Step 1: Laying out the structure

We begin by assuming that the main CAD program has been started, and that a layout window has been opened which contains the circuit to be simulated; this could be a newly created circuit, or a previously saved circuit that was loaded via the File/Open menu item. The general operation of the CAD program is described in the CAD manual.

Step 2: Choosing the FullWAVE simulation engine

The first step in initiating a FullWAVE simulation is to ensure that the **Simulation Tool** has been set to FullWAVE/FDTD in the Global Settings dialog box. This indicates to the CAD program that a FullWAVE FDTD simulation is to be performed.



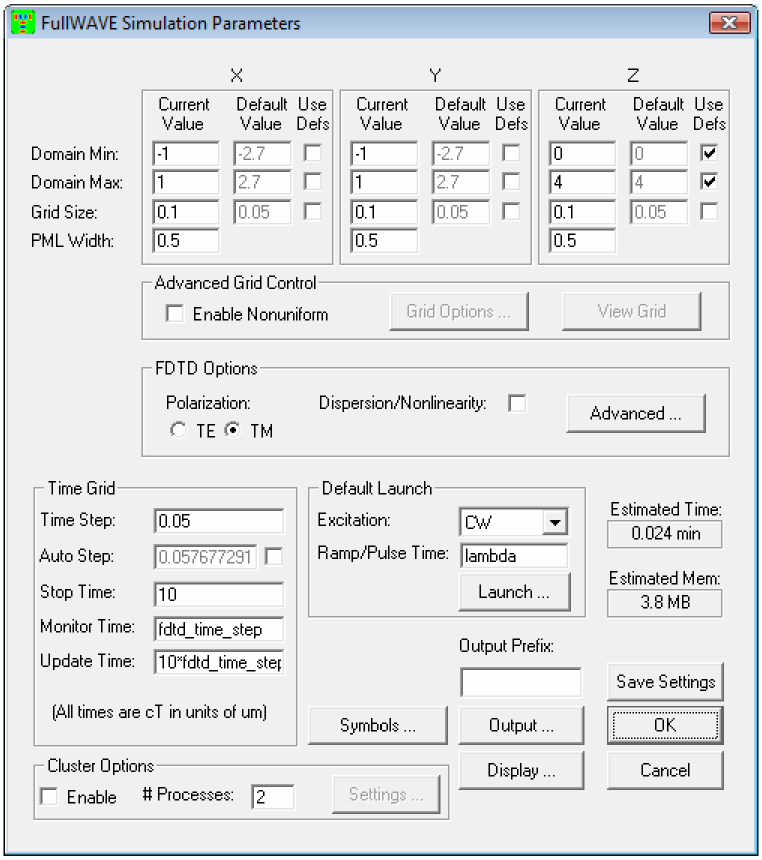
Simulation Tool:

○ BeamPROP ● FullWAVE
○ GratingMOD ○ BandSOLVE
○ DiffractMOD ○ FemSIM
○ ModePROP

Figure 1-1: The Simulation Tool option as shown in the Global Settings dialog box.

Step 3: Setting the Simulation Parameters

Next, open the Simulation Parameters dialog shown in Fig. 1-2 by clicking the **Perform Simulation** icon (green light) in the left CAD toolbar. This dialog allows the basic parameters required for a numerical simulation to be entered, and controls other important aspects of the simulation as well. See Chapter 3 for a complete description of this dialog.



FullWAVE Simulation Parameters

	X			Y			Z		
	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs
Domain Min:	-1	-2.7	□	-1	-2.7	□	0	0	☑
Domain Max:	1	2.7	□	1	2.7	□	4	4	☑
Grid Size:	0.1	0.05	□	0.1	0.05	□	0.1	0.05	□
PML Width:	0.5			0.5			0.5		

Advanced Grid Control

□ Enable Nonuniform

FDTD Options

Polarization: ○ TE ● TM

Dispersion/Nonlinearity: □

Time Grid

Time Step: 0.05

Auto Step: 0.057677291 □

Stop Time: 10

Monitor Time: fdtd_time_step

Update Time: 10*fdtd_time_step

(All times are cT in units of um)

Default Launch

Excitation: CW

Ramp/Pulse Time: lambda

Estimated Time: 0.024 min

Estimated Mem: 3.8 MB

Output Prefix:

Cluster Options

□ Enable # Processes: 2

Figure 1-2: The FullWAVE Simulation Parameters window where basic numerical simulation parameters are entered.

The default values can be accepted as is, or a different value for any parameter can be entered by editing the current value. Note that to save the simulation results it is necessary to enter an **Output Prefix** and select the desired data files via the **Output…** button. FullWAVE's output options are discussed in more detail in Section 3.D and Chapter 5.

Step 4: Running the Simulation

Once the numerical parameters and other options are acceptable, click on **OK** to begin the simulation. Note that **Cancel** rejects any changes and returns to the CAD interface, and **Save Settings** saves any changes made. Once a simulation is initiated, the computation begins in a new window separate from the main program. Within this window, the field, as a function of time, and any monitor output (see Chapter 5) are displayed during the calculation. The simulation can be paused and restarted if needed. Different field components to be viewed, and, in 3D allows different cross-sections of the selected field component to be viewed via the buttons in the simulation toolbar and the View menu item. When the simulation is finished, the title bar of the simulation program indicates "Computation Completed", and displays the coordinates of the cursor for reference.

Step 5: Accessing Results

To view or print saved simulation results, click on the WinPLOT icon in the top toolbar of the CAD interface, (the graph types are listed in the **Files of type** field). Each output is contained in two ASCII files: one file contains the raw data, and the other contains WinPLOT plotting commands. The file chosen via the above method is the WinPLOT command file so as to correctly display the raw data. To access the raw data, the data file can be opened in any text editor.

1.D.2. CLI Operation

As an alternative to using the FullWAVE GUI to initiate a simulation, it can sometimes be useful to run the simulation program from the command line. This is particularly true when a series of simulations needs to be run in which some parameter is to be varied, or when incorporating FullWAVE into a larger scripting environment.

When operating FullWAVE via the command line, it is important to note that the user can access the exact same feature set as if they were running FullWAVE though the GUI. All simulation options within the GUI are controlled by variables. A list of these variables can be found in Appendix C.

For a discussion of the command line syntax for non-clustered FullWAVE, please see Chapter 10 in the CAD manual. For the command line syntax for clustered FullWAVE, please see Chapter 7.

1.E. Example Files

The subdirectory `<rsoft_dir>\examples\FullWAVE` contains example files for FullWAVE. These files have an extension *.ind (named for the refractive index distribution which describes the circuit). In addition to these examples, there is a `Tutorials` subdirectory which contains the index files described in the Tutorial chapters of this manual.

1.F. README File

The latest product information cab be found in the `README` files located in the `/readme` subdirectory of the installation directory. These files provide important last minute information about FullWAVE that is not contained in this manual, including new or improved features and options.

1.G. Technical Support & Software Upgrades

RSoft software normally comes with one year of maintenance that includes both technical support and product updates. Updates are released on our support page www.rsoftdesign.com/support every one to three months, and include program corrections as well as new features. The necessary login information can be found in the email sent by RSoft that contains the license file for product purchases and upgrades. If you cannot find your login information, please contact your support contact at RSoft.

Information regarding each update is located in the `README` file, which can be accessed on the website to determine if you need or want to upgrade, and should be read thoroughly after downloading and installing any update. If you have any questions regarding your maintenance contract, or to renew your maintenance, please contact RSoft.

2

Background

This chapter provides technical information and references on the simulation methods used in FullWAVE. This material may be skipped initially; however it introduces terminology and notation which is important for understanding other discussions in the manual.

The objective of FullWAVE is to provide a general simulation package for computing the propagation of light waves in arbitrary geometries. The simulation is based on the well-known finite-difference time-domain (FDTD) technique. The FDTD method is a rigorous solution to Maxwell's equations and does not have any approximations or theoretical restrictions. This method is widely used as a propagation solution technique in integrated optics, especially in situations where solutions obtained via other methods such as the Beam Propagation Method (BPM) cannot cope with the structure geometry or are not adequate solutions. Since FDTD is a direct solution of Maxwell’s curl equations, it therefore includes many more effects than other approximate methods.

2.A. The FDTD Algorithm

Imagine a region of space which contains no flowing currents or isolated charges. Maxwell's curl equations in can be written in Cartesian coordinates as six simple scalar equations. Two examples are:

$$\frac{\partial H_x}{\partial t} = -\frac{1}{\mu}\left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y}\right)$$

$$\frac{\partial E_y}{\partial t} = -\frac{1}{\varepsilon}\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x}\right) \tag{1}$$

The other four are symmetric equivalents of the above and are obtained by cyclically exchanging the x, y, and z subscripts and derivatives. Maxwell's equations describe a situation in which the temporal change in the **E** field is dependent upon the spatial variation of the **H** field, and vice versa. The FDTD method solves Maxwell's equations by first discretizing the equations via central differences in time and space and then numerically solving these equations in software.

The most common method to solve these equations is based on Yee's mesh [3] and computes the **E** and **H** field components at points on a grid with grid points spaced Δx, Δy, and Δz apart. The **E** and the **H** field components are then interlaced in all three spatial dimensions as shown in Fig. 2-1. Furthermore, time is broken up into discrete steps of Δt. The **E** field components are then computed at times $t = n\Delta t$ and the **H** fields at times $t = (n+1/2)\Delta t$, where n is an integer representing the compute step. For example, the **E** field at a time $t = nt$ is equal to the **E** field at t = $(n$-1) Δt plus an additional term computed from the spatial variation, or curl, of the **H** field at time t.

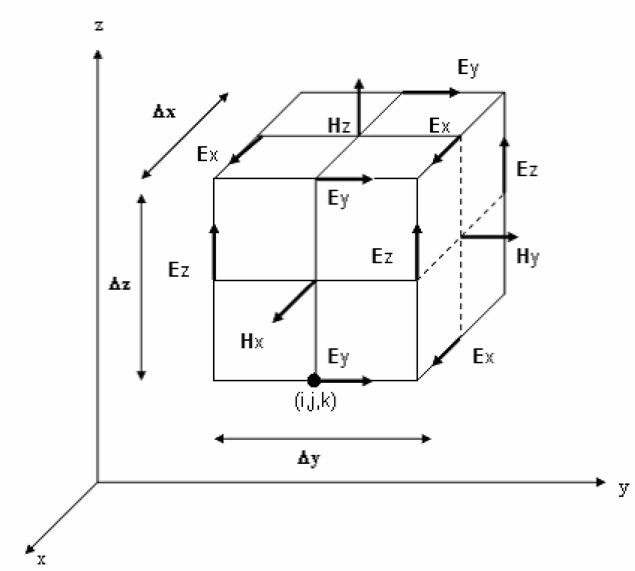


Figure 2-1: In a Yee cell of dimension Ax, Ay, Az, note how the H field is computed at points shifted one-half grid spacing from the E field grid points [3].

This method results in six equations that can be used to compute the field at a given mesh point, denoted by integers *i*, *j*, *k*. For example, two of the six are:

$$H_{x(i,j,k)}^{n+1/2} = H_{x(i,j,k)}^{n-1/2} + \frac{\Delta t}{\mu \Delta z}\left(E_{y(i,j,k)}^{n} - E_{y(i,j,k-1)}^{n}\right) - \frac{\Delta t}{\mu \Delta y}\left(E_{z(i,j,k)}^{n} - E_{z(i,j-1,k)}^{n}\right)$$

$$E_{x(i,j,k)}^{n+1} = E_{x(i,j,k)}^{n} + \frac{\Delta t}{\varepsilon \Delta y}\left(H_{z(i,j+1,k)}^{n+1/2} - H_{z(i,j,k)}^{n+1/2}\right) - \frac{\Delta t}{\varepsilon \Delta z}\left(H_{y(i,j,k+1)}^{n+1/2} - H_{y(i,j,k)}^{n+1/2}\right)$$

These equations are iteratively solved in a leapfrog manner, alternating between computing the **E** and **H** fields at subsequent $\Delta t/2$ intervals.

2.B. The FullWAVE FDTD implementation

In order to utilize the FDTD implementation used in FullWAVE, both physical and numerical parameters are required. This section outlines the definition of these parameters.

2.B.1. Physical Parameters

Two physical parameters are required in order to perform a FullWAVE simulation: the material parameters, including the relative permittivity $\varepsilon(\mathbf{r},\omega)$ and relative permeability $\mu(\mathbf{r},\omega)$ as a function of space and/or frequency, and the electromagnetic field excitation. Based on these parameters, the physics dictates the electromagnetic field as a function of $(\mathbf{r},\mathrm{t})$, or space and time.

Material Parameters

In order to use the FDTD algorithm to study the propagation of light fields in a structure, the material parameters must be specified. FullWAVE utilizes the following formulas to specify the material properties of a waveguide:

$$\mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P}$$
$$\mathbf{B} = \mu_0 \mathbf{H} + \mathbf{M}$$

where **P** and **M** are the polarization and magnetization can be, dispersive, non-linear, and anisotropic. This information is specified through the use of the CAD interface. Information on the specification of both a linear and non-linear index can be found in the CAD manual, and information related to the use of anisotropic, dispersive, and non-linear effects can be found in Section 6.A of this manual.

Electromagnetic field excitation

In order to perform a simulation, an initial launch condition ϕ_L at time t=0 is needed, as well as a driving function in time. This consists of both a spatial and temporal excitation, such as

$$\phi_L(\mathbf{r},t) = f(\mathbf{r}_0)g(t)$$

where $f(\mathbf{r}_o)$ is the spatial excitation at the launch plane and $g(t)$ is the temporal excitation. The specification of these parameters is discussed in Chapter 4.

2.B.2. Numerical Parameters

The solution algorithm requires additional input in the form of the numerical simulation parameters such as a finite computational domain, the boundary condition(s), the spatial grid sizes, and a temporal grid, or time step and the length of time for the simulation. The software attempts to estimate appropriate values for these parameters, but allows the user to override them.

As with any simulation, confidence in the accuracy of the numerical solution requires experimentation to determine the sensitivity of results to the numerical parameters.

Computational Domain

The choice of the computational domain must include the portion of the structure to be simulated. The specification of the computational domain is discussed in Section 3.B.

Boundary Conditions

The boundary conditions at the spatial edges of the computational domain must be carefully considered. Many simulations employ an absorbing boundary condition that eliminates any outward propagating energy that impinges on the domain boundaries. One of the most effective is the perfectly matched layer (PML) [1], in which both electric and magnetic conductivities are introduced in such a way that the wave impedance remains constant, absorbing the energy without inducing reflections.

Periodic boundary conditions (PBC) are also important because of their applicability to periodic structures such as PBGs. There are a number of variations on the PBC, but they all share the same common idea: the boundary condition is chosen such that the simulation is equivalent to an infinite structure composed of the basic computational domain repeated endlessly in all dimensions.

The PML boundary condition is briefly discussed in Section 3.B and a more in-depth discussion of PML and PBC, as well as several others, can be found in Section 6.C.

The Spatial and Temporal Grid

FullWAVE supports both a uniform and non-uniform spatial grid. In order to produce an accurate simulation, the spatial grid must be small enough to resolve the smallest feature of the field to be simulated. Usually this is dictated by the wavelength in the material(s) to be simulated, but, in some cases, can be dictated by the geometry of the photonic device. Typically, the grid spacing must be able to resolve the wavelength in time, and therefore usually be less than $\lambda/10$ where λ is not the free space wavelength, but rather the wavelength in the material(s). FullWAVE will provide intelligent choices for these parameters, but the user is strongly encouraged to refine the grid sizes in order to produce an efficient and accurate simulation.

Since the FDTD algorithm is based in the time domain, FullWAVE has several parameters which relate to the temporal grid. To obtain a stable simulation, one must adhere to the Courant condition which relates the spatial and temporal step size:

$$c\Delta t < \frac{1}{\sqrt{\left(1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2\right)}}$$

where c is the velocity of light and, for the case of a non-uniform grid, the grid sizes represent the smallest grid size in the simulation. FullWAVE will automatically enforce this condition and can optionally automatically use the largest temporal time size allowed. This condition changes slightly for dispersive simulations, and so it might be necessary to set a smaller time step in this case.

More information on the specification of the spatial grid can be found in Section 3.A, and information on the temporal grid can be found in Section 3.C.

2.C. Additional Comments

The computational steps in FDTD are relatively simple, using only addition, subtraction, and multiplication which improves the simulation speed. The technique is extremely versatile because it is inherently full-vectorial without limitations on optical effects such as direction of propagation, index contrast, or backward reflections.

The FDTD method is time-tested and stable, and can efficiently handle material dispersion and nonlinearities. Also, because it is based in the time-domain, it can cover a wide frequency range with a single simulation run. Furthermore, FDTD lends itself to cluster computing. This allows the computational demand for a single problem to be shared among several computers on a network, and permits researchers to simulate problems otherwise impossible on a single computer.

The FDTD method is computationally demanding, requiring a fairly dense grid of points at which all three vector components of both the **E** and **H** fields must be maintained. A full three-dimensional simulation of some fairly mundane problems like a simple, planar, evanescent splitter can be near impossible. Such devices are usually long (20 to 30 mm) and would require a computer with terabytes of memory, notwithstanding an impossibly long simulation time. Fortunately, such devices are handled imminently well by a BPM simulation package such as BeamPROP.

2.D. References

1 J.P. Berenger, J. *Comput. Phys.*, **114**, 185 (1994)

2 A. Tavlove, Computational Electrodynamics: The Finite-Difference Time-Domain Method, (Artech House, Norwood, MA, 1995)

3 K.S. Yee, “Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media” IEEE Trans. Antennas Propagat., **AP-14**, 302 (1966)

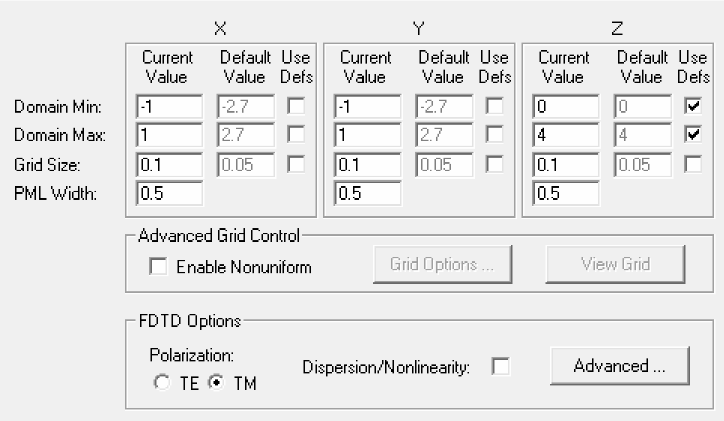
3

Using FullWAVE

This chapter describes the key features of the FullWAVE's FDTD based simulation engine. Some familiarity with the CAD is assumed; see the CAD manual for information on creating design files.

3.A. Spatial Domain, Grid, and FDTD Options

FullWAVE solves for the electromagnetic fields within a given structure as a function of time in a *finite domain* on *a spatial grid*. It is therefore important to define this domain and grid correctly, so as to ensure correct simulation results. Furthermore, the FDTD algorithm has several options which allow for further control over the simulation. The upper portion of the FullWAVE Simulation Parameters dialog allows the user to specify this information.



	X			Y			Z		
	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs
Domain Min:	-1	-2.7	□	-1	-2.7	□	0	0	☑
Domain Max:	1	2.7	□	1	2.7	□	4	4	☑
Grid Size:	0.1	0.05	□	0.1	0.05	□	0.1	0.05	□
PML Width:	0.5			0.5			0.5		

Advanced Grid Control

□ Enable Nonuniform

FDTD Options

Polarization: ○ TE ● TM

Dispersion/Nonlinearity: □

Figure 3-1: The section of the FullWAVE Simulation Parameters window where spatial grid and PML information is set. The PML specifications are discussed in the next section. The non-uniform portion of this dialog is described in Chapter 9 of the CAD manual.

3.A.1. Domain & Grids

Each spatial direction can be defined separately, and the applicable options are (all units are in µm):

Domain Min and **Domain Max**

This field sets the domain limits to be used during the simulation. For accurate results, the simulation domain must include all parts of the structure that you want to simulate, including any evanescent fields outside of the structure. It shouldn't be excessively large as nothing can be gained by simulating parts of the domain where no field is present.

Grid Size

This field sets the grid size for each dimension. The grid size used for a simulation must be small enough to resolve the smallest feature in the fields and structure during a simulation. For most structures, this means that the grid size should be small enough to resolve in time the wavelength of the field *in the material* being simulated. It is recommended that you perform a convergence study with the grid size to ensure accurate results.

FullWAVE supports the use of a non-uniform grid. For further details about this, please read through Chapter 9 in the CAD manual.

When starting a design project, you can save time during initial exploratory studies by using a course grid (which of course satisfies the above condition) and sacrificing accuracy. General results for your structure can often be obtained with remarkably course grids. At the same time, when a structure includes fine features, the results can be quite inaccurate with an overly coarse grid. You have not completed your design project until you have confirmed the convergence of your results at a higher resolution.

3.A.2. FDTD Options

The FDTD algorithm has many additional options which further control a FDTD simulation. These options are set either in the FullWAVE Simulation Parameters dialog or in the Advanced Parameters dialog which can be opened by clicking the **Advanced...** button.

The Advanced Parameters dialog is described in Chapter 6.

Polarization

The polarization definitions are slightly different for 2D and 3D simulations:

- *2D Simulations*

  For 2D FDTD simulations, Maxwell's equations are separable, and therefore each polarization can be modeled separately. A *TE* simulation models the electric field along y, and includes the E_y, H_x, and H_z field components. A *TM* simulation models the magnetic field along y, and includes the H_y, E_x, and E_z components.

- *3D Simulations*

  For 3D FDTD simulations, Maxwell's equations are coupled and can not be separated. It is best to set the polarization to the component in which the majority of the field resides: *TE* for E_x, and *TM* for E_y.

For a more detailed look at polarization, see the appendix in the CAD manual.

Dispersion/Nonlinearity

This option enables the use of dispersive and/or nonlinear materials in a simulation. These types of materials can be defined in the Material Editor described in the CAD manual. It is important to enable this option when these material types are used; if it is not enabled, the simulation can be unstable. See Section 6.A for more details.

3.B. Default Boundary Conditions

In addition to a spatial grid, FullWAVE also needs to define how the fields behave at the boundaries of the selected domain. By default, FullWAVE uses Perfectly Matched Layer, or PML, boundary conditions. A PML boundary, which consists of several points which are added to the edge of the domain, is designed to act as a highly lossy material which absorbs all incident energy without producing reflections. This allows field energy which is incident on the boundary to effectively leave the domain. These parameters control how the PML behaves during a FullWAVE simulation.

FullWAVE supports other boundary conditions such as Periodic, Symmetric, Anti-Symmetric, and PEC boundaries. Please see Section 6.C for more details about both PML and these additional boundary condition types.

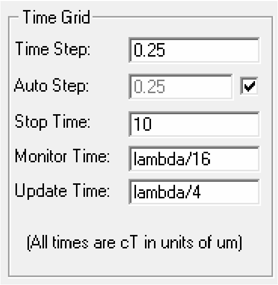
The PML option shown in Fig. 3-1 is:

PML Width

This field sets the width of the PML (Perfectly Matched Layer) used as a boundary condition. This width is then added to the edge of the computational domain.

3.C. Time Domain & Grid

The FDTD algorithm is based in the time-domain, and so in addition to a spatial grid, a temporal grid is needed. The lower left portion of the FullWAVE Simulation Parameters dialog box shown in Fig. 3-2 specifies the key time-domain parameters of the program.



Time Grid

Time Step: 0.25

Auto Step: 0.25 ☑

Stop Time: 10

Monitor Time: lambda/16

Update Time: lambda/4

(All times are cT in units of um)

Fig 3-2: The part of the FullWAVE Simulation Parameters dialog where temporal parameters are set.

All times must be specified as cT in units of µm. While these are generally preferable units, note that the speed of light in µm/s is a built in constant `c`, so that an absolute time, such as 1 femtosecond, can be entered as "`c*1e-15`".

These include:

Time Step and **Auto Step**

The **Time Step** field sets the grid size in the time dimension. The time step chosen must satisfy the Courant condition described in Section 2.B.2. It is not possible to perform a simulation unless this condition is met. The **Auto Step** field displays an optimal time step based on the Courant condition and other criteria such as sampling for DFT monitors, whether **Dispersion/Nonlinearity** enabled, and the **Free Space Wavelength**. To use this optimal time step, select the check box.

Stop Time

This determines the length, in time, of the calculation. The simulation will run from t=0 to the time specified.

Monitor Time

This determines the interval at which FullWAVE monitors save single-value output. For more information on single-value output, please see Section 5.C.

Update Time

This field determines the interval at which fields are displayed on screen during a simulation.

3.D. Choosing an Excitation

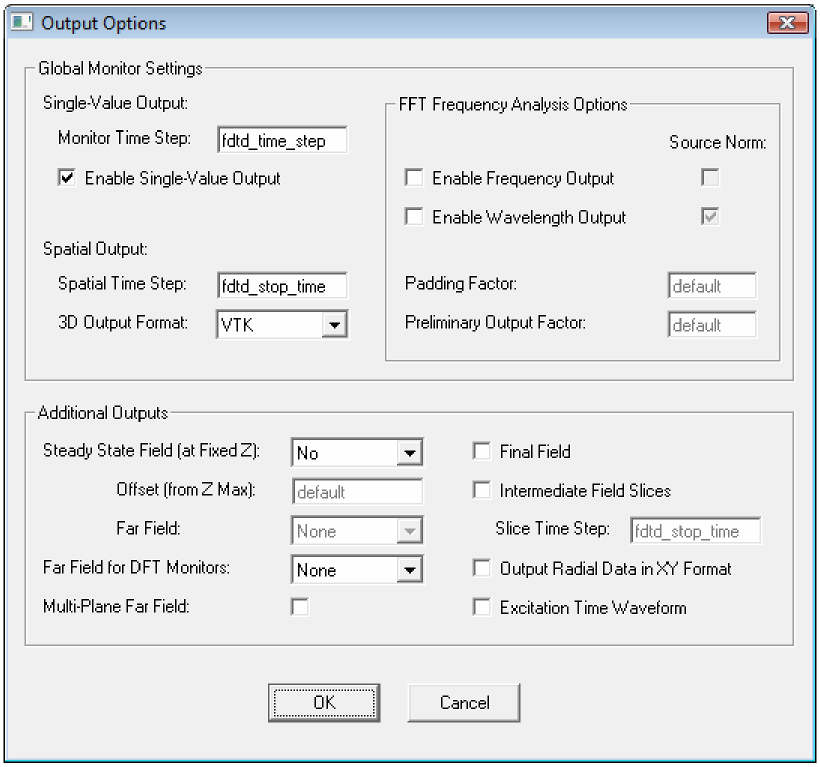
In order to perform a simulation, the initial field, or excitation of the system, needs to be defined. The excitation, also referred to as the launch field, is composed of both a spatial and temporal aspect and is defined via the Launch Parameters dialog. This dialog can be opened via the **Launch...** button in the FullWAVE Simulation Parameters dialog.

The specification of the excitation type can be found in Chapter 4.

3.E. Saving Data From a Simulation

FullWAVE contains a robust set of options to save data from a simulation. These options are controlled via monitors and the settings in the FDTD Output Options dialog shown in Fig. 3-3 which can be opened via the **Options...** button in the FullWAVE Simulation Parameters dialog. A description of monitors can be found in Chapter 5.

An **Output Prefix** field must be specified in the FullWAVE Simulation Parameters dialog in order to enable data output.



Global Monitor Settings

Single-Value Output:

Monitor Time Step: fdtd_time_step

☑ Enable Single-Value Output

Spatial Output:

Spatial Time Step: fdtd_stop_time

3D Output Format: VTK

FFT Frequency Analysis Options

Source Norm:

☐ Enable Frequency Output ☐

☐ Enable Wavelength Output ☑

Padding Factor: default

Preliminary Output Factor: default

Additional Outputs

Steady State Field (at Fixed Z): No

Offset (from Z Max): default

Far Field: None

Far Field for DFT Monitors: None

Multi-Plane Far Field: ☐

☐ Final Field

☐ Intermediate Field Slices

Slice Time Step: fdtd_stop_time

☐ Output Radial Data in XY Format

☐ Excitation Time Waveform

Fig 3-3: The FDTD Output Options dialog box.

These options can broadly be divided into two groups: Monitor Options and Additional Outputs:

3.E.1. Monitor Options

Monitors record information from a simulation. A detailed description can be found in Chapter 5. These options globally control several monitor options:

Single-Value Output Options:

Single-Value output includes the monitor outputs described in Section 5.C.

Monitor Time

This option sets the intervals at which temporal output measurements are made by monitors, and is the same as the control found in the FullWAVE Simulation Parameters dialog.

Enable Single-Value Output

This option selects whether or not to save the monitor results. By default, when a Pulsed or Impulse excitation is used, the monitor information is not saved. If this information is needed, this field should be manually enabled. For more details on the type of monitor output, please see Section 5.C.

Spatial Output Options:

Spatial Output includes the monitor outputs described in Section 5.D.

Spatial Time Step

This option sets the intervals at which spatial output measurements are made by monitors. By default, it is set equal to stop time.

3D Output Format

This sets the output format for the output of data from a 3D monitor. By default, data is output in a VTK format which can be opened with the MayaVI program. Alternatively, the data can be output in the RSoft format described in Appendix B which can be easier to use for custom post-processing.

Frequency Analysis Options

As described in Section 5.E and Section 5.F, FullWAVE can output frequency analysis data.

Frequency Output

This global option enables the calculation of an FFT of the time response of a system and to save the results as a function of frequency. By default this output is not normalized to the source spectrum, but this behavior can be changed by selecting **Source Norm**. For more details on the use of FFT output, please see Section 5.E.

Wavelength Output

This global option enables the calculation of an FFT of the time response of a system and to save the results as a function of wavelength. This option is automatically selected for the pulsed and impulse excitation types. By default this output is normalized to the source spectrum, but this behavior can be changed by selecting **Source Norm**. For more details on the use of FFT output, see Section 5.E.

Source Norm

These options control whether or not the Frequency (FFT & DFT) or Wavelength (FFT) output is normalized to the source spectrum.

Padding Factor

This factor attempts to increase the resolution of the FFT analysis by padding the time response with zeros. The factor must be a power of two, and it is only appropriate when the time response of the system has diminished to zero at the end of a simulation. By default, this option is not used. When this field is set to a value greater than 1, FullWAVE will extend the time window by factors of 2 up to the specified factor before performing the FFT. It will then generate a series of additional files. The `<prefix>.wmn` file will contain the normal graph obtained form the original time window. The `<prefix>.wm2`, `<prefix>.wm4`, ... files will be those obtained at 2, 4, ... times the original time window up to the specified factor. The `<prefix>.pwn` file will show the highest resolution file. For more details on the use of FFT output, please see Section 5.E.

Preliminary Output Factor

This factor allows for the output of preliminary FFT output during a simulation. FullWAVE will output FFT results at intermediate time windows, starting from the original window divided by the specified factor which much be a power of 2 greater than 1. The file prefixes for the intermediate results will have `_1`, `_2`, `_3`,... appended to their names. For more details on the use of FFT output, please see Section 5.E.

3.E.2. Additional Output Options

Steady State Field (at Fixed Z)

This feature enables the output of steady-state field data which can be used, for example, as the input into a BeamPROP BPM simulation. When this option is enabled, FullWAVE will store the field at the output cross-section in standard RSoft field format. The *Default* option saves only the major component of the field, while the *All Components* saves all the field components in the simulation. The results will be stored in the files `<prefix>_bpm.fld` (actual field data) and `<prefix>_bpm.pfd` (corresponding graph). By default, the cross-section will be taken at the last Z position in the computational domain (not including the PML); however it is sometimes better to take the cross section somewhat in front of the PML. This position can be set via the **Offset (from Z Max)** option.

For a CW simulation, the output will be at the excitation wavelength. For a pulsed simulation, the steady state field will be output on a similar frequency grid as DFT monitors: 33 points at and around the center frequency. To set this frequency range directly, use the variables `bpm_field_fmin`, `bpm_field_fmax`, and `bpm_field_df`.

This option only outputs steady-state field data at a fixed Z value; the output of such data on an arbitrary domain can be obtained through the use of a monitor's DFT output options. Please see Section 5.E.3 for more details.

Offset (from Z Max)

This field allows the user to specify an offset from the last Z position for the **Steady State Field (at Fixed Z)** output option. A negative value indicates a location less than the last Z position, putting it in front of the PML.

Far Field

This option enables the output of a far-field projection of the **Steady State Field (at Fixed Z)** output. For more information about computing far fields, see Section 6.B.1.

Far Field for DFT Monitors

This option enables the output of a far-field projection of the spatial output from a DFT monitor. To enable this type of output, the DFT option must be set so that a steady-state field is measured. See Section 5.F for information about using DFT monitors and Section 6.B.1 for details on far field calculation.

Multi-Plane Far Field

This option enables the output of a far-field computed from the near-field at all edges of the simulation domain. See Section 6.B.2 for details.

Final Field

This option indicates that the field should be output at the completion of a simulation.

Intermediate Field Slices

This option outputs field slices at the cross-section defined by the Cross Section and Slice Position defined in the FDTD Display Options dialog. Slices will be output at times specified by the **Slice Time** during a simulation. To save output slices on an arbitrary domain, use the spatial output options described in Section 5.D instead.

Slice Time

This option sets the interval at which **Intermediate Field Slices** are saved.

Output Radial Data in XY Format

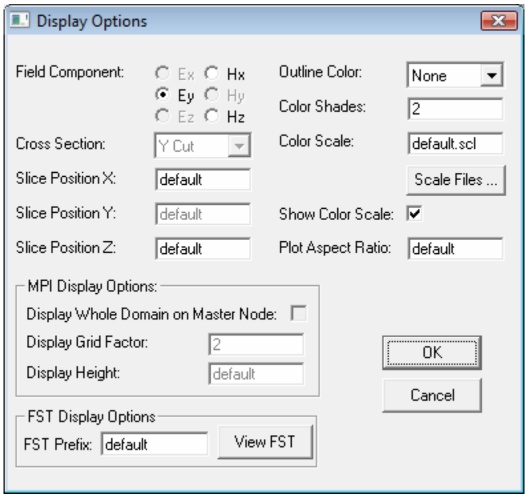
This option instructs the program to save output from a radial computation in XY format in addition to radial format.

Excitation Time Waveform

This option indicates that the instantaneous excitation amplitude should be output. To save the envelope instead, set the variable `fdtd_stm_output_envelope` to 1.

3.F. Displaying Data During a Simulation

During a FullWAVE simulation a field cross-section as well as monitor output are displayed and updated during the simulation at intervals defined by the **Update Time** option described in Section 3.C. The data displayed can be controlled further by the options in the FDTD Display Options dialog shown in Fig. 3-4 which can be opened via the **Display...** button in the FullWAVE Simulation Parameters dialog.



Display Options

Field Component: ○ Ex ○ Hx ● Ey ○ Hy ○ Ez ○ Hz

Cross Section: Y Cut

Slice Position X: default

Slice Position Y: default

Slice Position Z: default

Outline Color: None

Color Shades: 2

Color Scale: default.scl

Show Color Scale: ☑

Plot Aspect Ratio: default

MPI Display Options:

Display Whole Domain on Master Node: □

Display Grid Factor: 2

Display Height: default

FST Display Options

FST Prefix: default

Fig 3-4: The FDTD Display Options dialog box.

These options are as follows:

Field Component

This option selects the field component to be displayed during a simulation. Note that the field component can be changed during a simulation via the View/Field Component option in the FullWAVE simulation window.

Cross Section, Slice Position X, Y, and Z

These options set the cross section and slice position for the displayed field profile in a 3D simulation for a given **Cross Section** type. This setting can be changed during a simulation via the View/Cross Section option in the FullWAVE simulation window.

Outline Color

This option enables the display of a structure outline on top of the field display, and if so, what color scheme to use.

Color Shades

This option indicates the number of color shades to be used render the displayed field can also be set.

Show Color Scale

This option indicates that the color scale should be displayed in the simulation window.

Color Scale

This option sets the color scale to be used when rendering the displayed field in the simulation window. Scale files can either be entered by name, or chosen via the **Scale Files...** button.

Plot Aspect Ratio

Field plots do not normally display in a 1:1 aspect ratio, and this option sets the aspect ratio of the plot to 1 or any other desired aspect ratio. For stored plots generated with previous versions, or generated without having set this option, the aspect ratio can be changed in WinPLOT under Options/Axes/Aspect-Ratio... or via the `/aspect` option. Please consult the WinPLOT manual for more information.

MPI Display Options

These options control display properties for use during a clustered simulation. Please see Chapter 7 for a description of these options.

FST Display Options

These options allow the user to display a previously saved `.fst` state file. See Section 6.D.4 for a description of these options.

4

Excitation Types

A key part of a FullWAVE simulation is the launch field, or excitation, used to excite the structure. It should be chosen to correspond to the type of simulation results required. For example, a CW excitation should be used when the steady-state performance of a structure is desired and a Pulsed excitation can be used when the spectral characteristics of a structure are needed.

4.A. Basic Excitation Definition

The excitation is composed of both spatial and temporal components, and is represented as a sum of individual excitations and can be expressed by the following equation:

$$\phi_L(\mathbf{r},t) = \sum_i \phi_i(\mathbf{r},t)$$

The individual excitations $\phi_L(\mathbf{r},t)$ are defined as:

$$\phi_i(\mathbf{r},t) = \sqrt{P_i}\, f_i(\mathbf{r}) g_i(t) e^{i\theta_i t}$$

where P_i and θ_i are the power and phase of the excitation (see Section 4.F), and *fi*(**r**) and *gi*(*t*) are the spatial and temporal components of the excitation. The exact definition of these parameters and functions will be discussed in Section 4.C and Section 4.D respectively. Each individual excitation in the above equation is defined in the same way, and is then combined to realize the final excitation function.

Before the actual definition of the excitation functions is discussed, it will be beneficial to outline the available types of excitation.

Field Sources

Field sources can be thought of as an incident beam. They are defined along a specific plane, and radiate from that plane. These are the most common types of excitations used within FullWAVE.

Current Sources

Current sources are excitations due to currents present in the simulation domain. Both the spatial aspect of the field as well as the current direction can be specified. FullWAVE supports both single current sources as well as a plane of incoherent current sources.

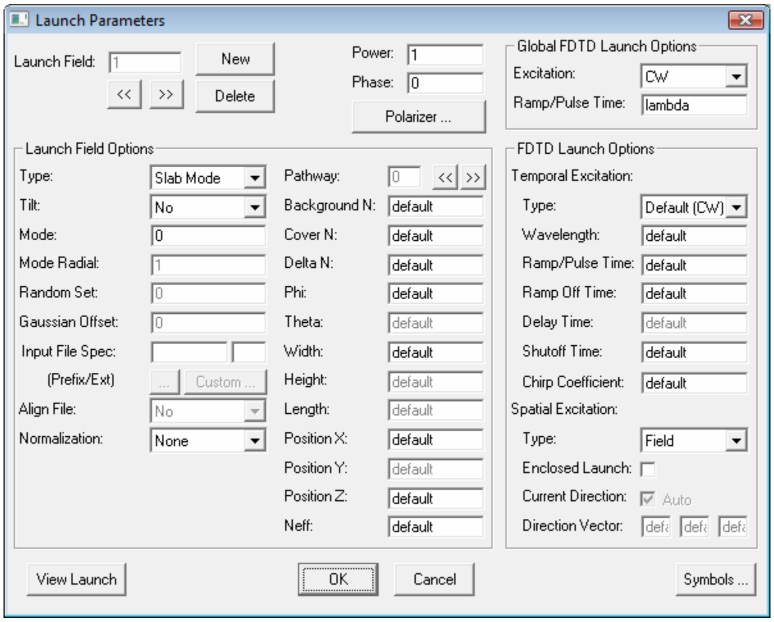
Enclosed Sources

An enclosed source allows a plane wave to be launched inside an enclosed area/volume. This feature is useful for scattering calculations and is described in Section 4.I.

4.B. The Launch Parameters Dialog

A launch field is set via the Launch Parameters dialog box shown in Fig. 4-1 which can be accessed either via the **Edit Launch Field** button in the left menu in the CAD interface, or via the **Launch...** button located in the FullWAVE Simulation Parameters dialog.

Several global launch options can be set via options in the FullWAVE Simulation Parameters dialog as well as in the upper right corner of the Launch Parameters dialog and are discussed in Section 4.D.


Launch Parameters

Launch Field: 1

Power: 1
Phase: 0

Global FDTD Launch Options
Excitation: CW
Ramp/Pulse Time: lambda

Launch Field Options
Type: Slab Mode
Tilt: No
Mode: 0
Mode Radial: 1
Random Set: 0
Gaussian Offset: 0
Input File Spec:
(Prefix/Ext)
Align File: No
Normalization: None

Pathway: 0
Background N: default
Cover N: default
Delta N: default
Phi: default
Theta: default
Width: default
Height: default
Length: default
Position X: default
Position Y: default
Position Z: default
Neff: default

FDTD Launch Options
Temporal Excitation:
Type: Default (CW)
Wavelength: default
Ramp/Pulse Time: default
Ramp Off Time: default
Delay Time: default
Shutoff Time: default
Chirp Coefficient: default
Spatial Excitation:
Type: Field
Enclosed Launch: ☐
Current Direction: ☑ Auto
Direction Vector: defa defa defa

Figure 4-1: The Launch Parameters dialog which selects both the type and characteristics of the optical input field that will be launched into the circuit for the simulation.

This dialog allows the user to define multiple launch fields which can be linearly combined with different powers and phases to realize the function $\phi_L(\mathbf{r},t)$ This multiple launch feature is controlled by several fields at the top of the dialog, and is discussed in Section 4.F.

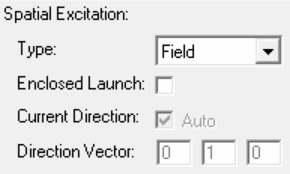
We begin by describing the launch options for an individual launch field which are controlled by the remaining fields in the dialog.

4.C. Spatial Excitation

The function $f_i(\mathbf{r})$ above represents the spatial component of the launch field. The spatial component can be divided into two main sections: the desired field profile type, and then optional launch data which describe the desired profile. This section describes the definition of this function for both a field source and a current source, and then the field profile and its associated optional parameters.

4.C.1. Choosing a Source Type

FullWAVE supports several different types of spatial excitations, or source types.



Spatial Excitation:

Type: Field

Enclosed Launch: □

Current Direction: ☑ Auto

Direction Vector: 0 1 0

Figure 4-2: The part of the Launch Parameters dialog where Spatial Excitation options are set.

Field Sources

This is the most commonly used spatial excitation type and can be thought of as an incident beam. To select this type of spatial excitation, set the **Spatial Excitation Type** to *Field*. The beam profile can be selected from a built-in profile or a data file can be used. Additionally, the optional fields discussed below allow for further customization of the launch profile.

A special type of field source is an **Enclosed Launch**. This type of launch is useful for scattering calculations and is described in Section 4.I.

Current Sources

A current source is designed to produce fields through the introduction of a current with in the simulation domain by introducing the current term into Maxwell's curl equations. To select this type of spatial excitation, set the **Spatial Excitation Type** to *Current*.

Since current is a vector, the direction of the current needs to be set. By default, the direction is set along the major polarization direction. To use the default direction, set the **Current Direction** to **Auto**. In 2D, the source can be considered a wire along the Y direction. For 2D TE simulations, the wire contains an electrical current (E_y); for 2D TM simulations, the wire contains a magnetic current (H_y).

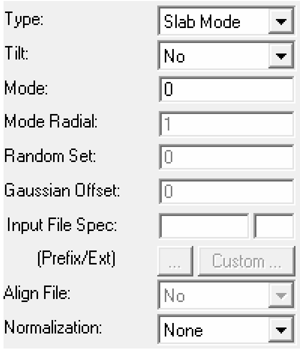
For 3D, a user defined direction can be used. To use a custom direction, uncheck the **Auto** box and manually set the **Direction Vector**. The field profile is assumed to be Gaussian in X, Y, and Z. The Gaussian width in each of these directions is defined by the values of the **Width**, **Height**, and **Length** fields discussed below. The other launch options are not applicable at this time.

Current Profile

This source type produces a continuous distribution of current sources that lie within a plane. This can be useful for modeling incoherent sources such as LEDs, etc. The amplitude and phase of these current sources can be set via the launch **Type**. For example, choosing *Rectangle* gives a uniform amplitude and phase across the distribution, and choosing *File* takes this information from the user-defined files as usual. By default, the phase distribution is uniform and the current direction is exactly the same as a single current source. However, setting `launch_random_spatial_phase` and/or `launch_random_spatial_dir` to `1` allows for a random phase and/or current direction. The spatial distribution of this randomness is characterized by a correlation length which is set by `launch_random_spatial_corlen` which defaults to `1`. Different random patterns can be selected by changing `launch_random_spatial_seed` (defaults to `1`) to an integer greater than 1.

4.C.2. Defining the Field Profile Type

A field source is defined along a plane in the simulation domain. To accomplish this, we need to define the function $fi(\mathbf{r})$ described above along that plane. For example, if we are going to launch from a constant Z position, we need to define the profile of the launch field given by the function $fi(x, y, z_0)$ where z_0 defines the incident plane. For more information on launching at a particular angle, see Section 4.C.4.



Type:	Slab Mode
Tilt:	No
Mode:	0
Mode Radial:	1
Random Set:	0
Gaussian Offset:	0
Input File Spec:	
(Prefix/Ext)	... Custom ...
Align File:	No
Normalization:	None

Figure 4-3: The portion of the Launch Parameters dialog where the Field Profile Type is chosen.

The options shown in Fig. 4-3 which control the field profile for a field source will be discussed in this section. The individual parameters for each profile type will be discussed in the next section.

Launch Type

This drop down box selects the type of input field. The available choices are:

- *File*

The input field is obtained from one or more ASCII data files which can be the result of a previous simulation, mode calculation, or a user-defined data set. The data files are selected via the **Input File Spec** option described below.

- *Computed Mode*

  This option is not used by FullWAVE.

- *Slab Mode* or *Fiber Mode*

  This option indicates that an analytically calculated slab mode (in 2D) or fiber mode (in 3D) should be used as the spatial input field and is the default. The characteristics of the mode are, by default, set via the input component in the CAD, but can be set directly via the Optimal field profile data described in the next section. For the case of a slab mode, the mode number is set via the **Mode** option described below, and for the case of a fiber mode, the mode number is set by the **Mode** and **Mode Radial** options described below to yield an LP_{mn} fiber mode.

- *Gaussian*

  This option indicates that the spatial input field will be a Gaussian function defined as

  $$f(x,y,z_0)=\exp[-x^2/a^2]\exp[-y^2/b^2]$$

  where $a=w/2$, $b=h/2$, and w and h are either the 1/e width/height of the Gaussian, which are set to the dimensions of the input component in the CAD buy can be set directly by the Optional Profile Data described in the next section. By default, the Gaussian is launched from the waist, but an offset can be set via the **Gaussian Offset** option described below.

- *Rectangle*

  This option indicates that the spatial input field will be a rectangular step function of unit value with dimensions set the same as the input component in the CAD but can be set directly via the Optional Profile Data described in the next section.

- *MultiMode*

  This is a specialized option useful for studying highly multimode guides. The spatial input field will be a superposition of all slab (2D) or fiber (3D) modes supported by the input component with equal power in each mode and a random phase for each mode. The set of random phases is determined by a pseudo-random number generator that does not vary from run to run; however, the user can select different sets of random phases through the **Random Set** field described below. Also, the set of modes that is launched may be controlled through the **Launch Mode** field described below.

Tilt

This option enables the launch field to be tilted so as to enter the simulation domain at an angle. See Section 4.C.4.

Mode

This option sets the integer mode number of the launch field. For a *Slab Mode* input type, a value of `0` represents the fundamental mode and is the default. For a *Fiber Mode* input type, this value corresponds to the azimuthal mode number m of an LP_{mn} fiber mode. A positive value of m corresponds to a $\cos(m\phi)$ dependence and a negative value of m corresponds to a $\sin(m\phi)$ dependence. For a Multimode input type, this value corresponds to the range(s) of modes to be launched. The syntax is `low-high:step` and the wildcard `*` symbol can be used. For example, a value of '*' selects all modes and is the default, a value of '`0-5`' selects the first 6 modes, a value of '`*:2`' selects all even modes, and a value of '`1-*:2`' selects all odd modes.

Mode Radial

This option sets the radial mode number n of the corresponding LP_{mn} fiber mode for an input type of *Fiber Mode*.

Random Set

This integer option sets the particular random set to be used for the random phases for an input type of *Multimode*. The first set is numbered `0` which is the default.

Gaussian Offset

This option sets the offset from the Gaussian waist of an input type of *Gaussian*. A positive value will effectively put the waist 'in front of' the launch plane; a negative value will effectively put the waist 'behind' the launch plane. When using this option it is critical that the width/height of the Gaussian are still defined at the waist.

Input File Spec

These options set the data file(s) to be used for a **Launch Type** of *File*. These files can be automatically set by a file **Prefix** and **Extension** or manually using the controls provided by clicking **Custom...**.

- *Automatic Method:*

  The automatic method provides a simple way to define the launch files used for individual components by setting a file prefix and extension. The files must have the default output naming style used by RSoft software for mode and field output: `<prefix>_<comp>.<ext>` where `<comp>` is the field component. For example, if the data files are named `myprefix_ex.m00`, `myprefix_ey.m00`, etc., enter `myprefix` and `m00` for the **Prefix** and **Extension** respectively. These settings can be easily set by choosing one of the field component files via the '**...**' button. As a convenience, the data files the software will look for based on the prefix and extension specified can be seen by clicking the **Custom** button. Note that this feature will also work with older naming conventions such as `<prefix>.m00`, `<prefix>.n00`, etc, though the new style is preferred.

  The software will, based on the **Model Dimension** and **Polarization**, attempt to locate the files corresponding to the necessary field components. If any of the necessary component files are missing, the software will attempt to either ignore or calculate the required components. In 2D, a TE polarization requires E_y data and a TM polarization requires E_x. If present, it will also include corresponding H field data. In 3D, a TE polarization requires E_x data and TM polarization requires

E_x. If present, it will also include the other E field and/or corresponding H field data. Generally, all necessary components should be made available; in some cases such as low index-contrast simulations, all components may not be necessary.

- *Custom Method:*

  To manually define the launch files used for individual components, click the **Custom** button, uncheck **Auto**, and define the necessary components. The major and minor components refer to different components of the electric **E** and magnetic field **H** and the exact definition depends on the **Model Dimension** and **Polarization** used.

Field files produced by mode solving (BeamPROP, FemSIM, TmmSIM, etc) or propagation (BeamPROP, FullWAVE, etc) will be appropriately named and formatted and can easily be used with the automatic method. Data files manually created must be in the standard file format discussed in Appendix B and can be used with the automatic method if named appropriately.

It is possible to use a variable to define a launch file: set the launch file option to `$<variable>` where `<variable>` is a variable created in the Symbol Table set equal to the name of the desired launch file. If a 2D data file is launched into a 3D simulation it will be interpreted as a radial profile and a 3D field distribution is automatically generated. The **Mode** option sets the azimuthal mode number.

Normalization

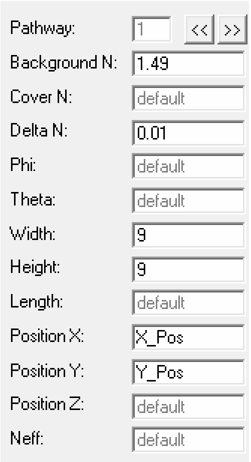
This option selects how the input field is normalized. The default choice is *None* which does not modify the values in a data file for an input type of *File* and sets the peak value of all other launch fields to 1. A choice of *Unit Peak* sets the peak value to 1 in all cases, and a choice of *Unit Power* normalizes the field such that the power in the input field is 1. The units the power is measured in, except when simulating non-linear materials, is not important and should not be changed. For a non-linear case, the power units are (in 3D) V^2 but can be changed to W by setting `launch_power_unit` to `1`. In 2D, these units are $V^2/\mu m$ and $W/\mu m$ respectively.

Align File

This option controls alignment of the launch field with the **Pathway** when the input type is *File*. The default value of *No* indicates that the associated data file is not automatically aligned with the **Pathway** since it is assumed that the field data is coming from the result of a previous simulation and that the coordinates should be left unchanged. Setting this option to *Yes* indicates that it should be aligned. This is useful, for example, if the data in the file is the result of a mode calculation and the launch position is not the same as the position of the waveguide used in calculating the mode. Note that for this option to work properly, the component used in calculating the mode must have its X and Y position set equal to 0.

4.C.3. Using Optional Field Profile Data

The launch data specified in Fig. 5-4 sets the parameters used to calculate the launch profile.



Pathway: 1
Background N: 1.49
Cover N: default
Delta N: 0.01
Phi: default
Theta: default
Width: 9
Height: 9
Length: default
Position X: X_Pos
Position Y: Y_Pos
Position Z: default
Neff: default

Figure 4-4: The portion of the Launch Parameters dialog where the Launch Data is set.

The right column shown in Fig. 4-2 sets the optional field profile data used to calculate the launch profile. The default values for these options are set by an input component and can be manually defined if needed.

If optical pathways have been defined as described in the CAD manual, they can be used to define the input component. To choose a pathway to be used for a particular launch field, use the **<<** and **>>** buttons. The number of the currently selected pathway will be shown. Once a pathway has been selected the first (lowest Z value) in that pathway will be used as the input component. If no pathway is defined, the component with the smallest starting Z value within the computation domain is used. If more than one component satisfies this condition then the component with the lowest segment number is used.

In either case, several characteristics of the launch field, such as position, mode width, and input angle, are determined from the corresponding properties of the input component. These characteristics may be overridden individually by entering explicit values.

The following options are set to the keyword `default` by default which indicates that the value will be taken from the input component.

Background N

This option sets the background index for the slab/fiber mode to be launched. The default value corresponds to the background index of the design file.

Cover N

This option sets the cover index for the asymmetric slab mode to be launched. The default value corresponds to the background index difference of the design file.

Delta N

This option sets the index difference for the slab/fiber mode to be launched. The default value corresponds to the index difference of the input component.

Phi and **Theta**

These options control the angle at which a tilted launch enters the simulation domain. See Section 4.C.4.

Width, **Height**, and **Length**

These options set the width (measured along X), height (measured along Y) and length (measured along Z) of the input field to be launched. The default width and height values correspond to the width and height of the input component, and the default length is 0. The height option can only be used in 3D, and the length option can only be used with either an impulse or current source.

Position X, **Position Y**, and **Position Z**

These options set the position at which the input field is launched. The default values correspond to the input component. For an input type of *File* the X and Y options assume that the field in the data file is centered at 0 and will effectively offset the launch file data. The Z position defaults to the Z **Domain Min** value.

Neff

This numeric input field selects the effective index corresponding to the propagation constant of the input field. For an input type of *Slab Mode* or *Fiber Mode*, the default value is chosen to correspond to the actual propagation constant. For an input type of *File* when the data file used is the result of a mode calculation, the default value is set to the effective index stored in the file. This option is only used for field sources and not for current sources.

4.C.4. Launching at an Angle

By default, FullWAVE launches from a constant Z plane along the positive Z axis. The direction of the launch field, however, can be changed via the **Phi**, **Theta**, and **Tilt** options in the Launch Options dialog.

Definition of Launch Angles

The launch angles are illustrated in Fig. 4-5.

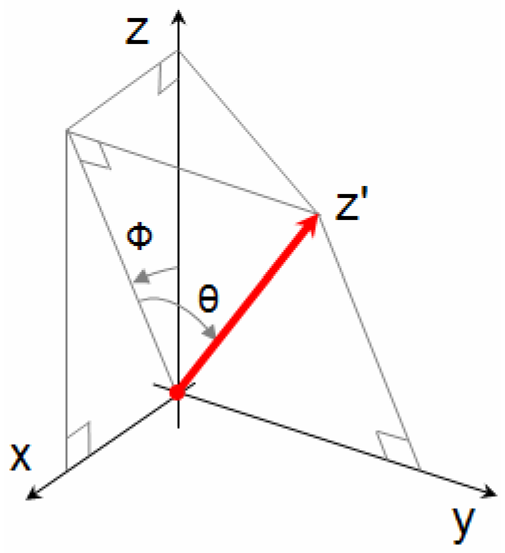


Figure 4-5: Illustration of the launch angles phi and theta. Note that for clarity, the X' and Y' axes are not displayed.

The angle ϕ is defined as a rotation around the Y axis, and is measured from the Z axis. The angle θ is defined as a rotation around the X' axis and is measured from the XZ plane. Both angles are defined in degrees and have default values set by the orientation of the input component.

The definition of a non-zero angle or the use of a tilted input component *only* sets the launch plane and does not tilt the launch. The **Tilt** option must be used to actually tilt the launch.

Definition of Launch Plane

All FullWAVE launch fields (not current sources) emanate from a specific launch plane and have a specific orientation. By default, this launch plane is the Z domain minimum along the +Z axis, but can be changed by setting the X, Y, and/or Z position along with the launch angles Phi and Theta. Consider the launch angle ϕ to illustrate the dependency of the launch direction on the launch angles:

The value of ϕ here can be either set by the user or taken from the default value corresponding to the orientation of the input component. If the **Tilt** option is not enabled, the launch angle used is noted. If the **Tilt** option is used, a phase front will be applied to the launch field to achieve the actual ϕ value.

- If ϕ is in the range (-45, 45], the launch field will emanate from a constant Z plane along the positive Z axis (ϕ=0 degrees when **Tilt** option not enabled).
- If ϕ is in the range (45, 135], the launch field will emanate from a constant X plane along the positive X axis (ϕ=90 degrees when **Tilt** option not enabled).
- If ϕ is in the range (135, 225], the launch field will emanate from a constant Z plane along the negative Z axis (ϕ=180 degrees when **Tilt** option not enabled).

- If ϕ is in the range (225, 315], the launch field will emanate from a constant X plane along the negative X axis. (ϕ=270 degrees when **Tilt** option not enabled).

The launch plane definitions for the other launch angle θ are similar.

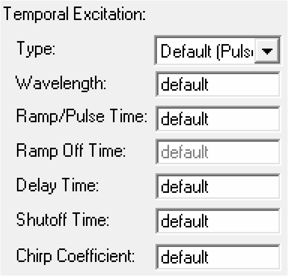
It is possible to set the launch plane manually which can be useful launching an oblique plane wave from a specific plane. See Section 4.H.3.

4.C.5. Viewing Launch Field

The **View Launch** button in the Launch Parameters dialog will compute and display the current spatial launch field for diagnostic purposes before running a simulation.

4.D. Temporal Excitation Types

In addition to the spatial excitation, a FullWAVE launch field requires a temporal excitation function *$gi(t)$*. The type of temporal excitation that should be used will be determined by the desired results.



Temporal Excitation:

Type:	Default (Puls▾
Wavelength:	default
Ramp/Pulse Time:	default
Ramp Off Time:	default
Delay Time:	default
Shutoff Time:	default
Chirp Coefficient:	default

Fig 4-6: The portion of the Launch Parameters dialog box where temporal excitation settings are set.

FullWAVE currently supports three temporal excitation types: *CW*, or continuous wave, *Pulse*, and *Impulse*, and is controlled locally for each launch field by the **Temporal Excitation Type** field shown in Fig. 4-6. The keyword `default` corresponds to the global default set in the FullWAVE Simulation Parameters dialog.

For clarity, the default values of the additional temporal excitation options shown in Fig. 4-6 are discussed in the next section.

These types, as well as guidelines to their appropriate use, are as follows:

4.D.1. CW

The default values for the options shown in Fig. 4-6 are discussed in Section 4.D.5.

A *CW* excitation type is normally used when the steady state performance of the design. It consists of two envelope functions multiplying a sinusoidal carrier. This can be expressed as

$$g(t) = \xi_{up}(t)\xi_{down}(t)\sin\left[\frac{2\pi}{\lambda}(t) + At^2\right]$$

where the wavelength λ is set by the **Wavelength** field and the factor A, which has units of µ&mu;m^{-2}, produces a chirped excitation, and can be set by the field **Chirp Coefficient**.

Smoothly turning on a CW excitation

A CW excitation can be smoothly turned on to avoid numerical noise in a simulation. The envelope function $\xi_{up}(t)$ ramps up from 0 to 1 over the ramp up time, *tu* set in units of cT in the **Ramp/Pulse Time** field. Specifically, it is defined as:

$$\xi_{up}(t) = \sin\left[\pi/2\left(t/t_u\right)\right]$$

when $t < t_u$ and is set to 1 for times after t_u.

Smoothly shutting off a CW excitation

A CW excitation can be smoothly shut off using the envelope function $\xi_{down}(t)$. This function ramps down from 1 to 0 over the ramp down time, t_d, and is set in units of cT in the **Ramp Off Time** field so that the source is completely shut off at the **Shutoff Time** t_s specified. Specifically, it is defined as:

$$\xi_{down}(t) = \cos\left[\pi/2\left(\left(t_s - t\right)/t_d\right)\right]$$

and is set to 1 for times prior to $t_s - t_d$.

4.D.2. Pulse

The default values for the options shown in Fig. 4-6 are discussed in Section 4.D.5.

A *Pulse* excitation is normally used when the frequency response of the design is desired. It consists of a Gaussian envelope function multiplying a sinusoidal carrier. This can be expressed as

$$g(t) = \exp\left[-\left(\frac{t}{\tau} - t_d\right)^2\right]\sin\left[\frac{2\pi}{\lambda}(t) + At^2 + \phi_0\right]$$

where the wavelength λ is set by the **Wavelength** field. The Gaussian envelope function is defined such that τ is the pulse time in units of cT set in the **Ramp/Pulse Time** field, and t_d is the **Delay Time** in units of τ, and therefore unitless. The factor A, which has units of µm^{-2}, produces a chirped excitation, and can be set by the field **Chirp Coefficient**. The phase constant Φ_0 is chosen such that the phase is 0 at the maxumum of the Gaussian function. The excitation can be shut off at a desired time using the **Shutoff Time**, which represents the total time from t=0 that the excitation is active.

To change the hyperbolic secant (sech) time envelope for the launch pulse which is useful for setting the launch field for nonlinear soliton-forming simulations. This can be done by setting the variable `fdtd_excitation` to `TIME_ENVELOPE_SECH`.

4.D.3. Impulse

The default values for the options shown in Fig. 4-6 are discussed in Section 4.D.5.

An *Impulse* excitation is normally only used for a cavity mode calculation, and results in predominantly forward traveling waves. It has a different meaning than the other excitation types: It sets an initial field pattern at t=0, then lets it propagate for t>0 without any further driving of the field. This can be expressed as:

$$\phi_L(\mathbf{r},t) = f\left(x,y\right)h\left(z\right)\delta\left(t_0\right)$$

The initial field pattern for *Impulse* excitation is determined as follows:

- *Spatial Dependence in X and Y:*

  The spatial distribution in X and Y, $f(x,y)$ is given by the normal spatial launch field as described in Section 4.C.

- *Spatial Dependence in Z:*

  The Z-dependence is given by the function $h(z)$, which is by default defined as the following "delta-function':

  $$h\left(z\right) = \delta\left(z_0\right)$$

  where z_0 is the Z launch position. This default can be changed and given a Lorentzian profile by entering a non-zero value of the launch **Length**. In this case, $h(z)$ is defined as:

  $$h\left(z\right) = \exp\left[-\left(z - z_0/l\right)\right]$$

  where l is the **Length**.

- *Time Dependence:*

  The time dependence is given by a "delta-function" $\delta(z_0)$ centered at the Z launch position. An *Impulse* excitation does not have a specific driving function $g(t)$ in time, and is simply allowed to radiate/decay in time.

The launch can be rotated through the use of the **Phi** and **Theta** fields described above. Also, for this to work properly, the **Launch Neff** in the Launch Parameters window must be set equal to the index at the center of the launch field. In most cases FullWAVE will be able to determine this, but in some it will not and so this parameters should be set explicitly.

When using the *Impulse* option, the wavelength has no direct bearing on the excitation or the simulation. However, if any other settings, such as a **Grid Size** or **Time Step** for example, have been set to be a formula dependent on the global wavelength `free_space_wavelength` or `lambda`, then there will of course be an indirect effect on the simulation when changing the **Free Space Wavelength** setting in the Global Settings dialog.

4.D.4. Custom Time Envelopes

FullWAVE allows for a custom time envelope to be defined. The feature currently applies globally to all launch fields. To use this option, set `fdtd_excitation_envelope` to `TIME_ENVELOPE_USER`. The desired time envelope can then be specified by setting `fdtd_user_envelope` = `<function of t>`. Currently, the symbol `t` must be defined in the symbol table and not be otherwise used.

Note that this option sets the specified time envelope on top of the normal carrier. For example, an unending ramp time envelope for the launch can be achieved by setting `fdtd_user_envelope` = `t` in the symbol table. Note that this feature can be applied to both CW and Pulsed excitation types, however even by setting the same envelope these two excitations differ in the choice of the phase of the sinusoidal carrier (CW phase is zero at t = 0, pulse is zero at t = `fdtd_delay_time`). In addition, CW and Pulse launch differ in the defaults for certain analysis options. It is recommended that this option be used with the option `fdtd_stm_output` = `1` which allows the time sequence of the launch source. The source will be output in a file `<prefix>.stm`.

4.D.5. Setting Optional Temporal Options

The additional temporal excitation settings shown in Fig. 4-5 allow control over the temporal excitation types. Their exact usage is described in the previous section, and they all are not enabled for each **Temporal Excitation Type**.

Additionally, the global default value of several of these fields is set via the fields in the FullWAVE Simulation Parameters dialog box.

Wavelength

This field sets the wavelength at which the excitation is driven. Its default value corresponds to the **Free Space Wavelength** setting in the Global Settings dialog box, and is given in units of µm.

Ramp/Pulse Time

This field sets the ramp up time for a CW simulation or the pulse time for a pulsed simulation. It defaults to a value equal to the pre-defined variable `lambda`, which is set equal to the **Free Space Wavelength**, and is given in units of cT [µm].

Ramp Off Time

This field sets the ramp off time for a CW simulation. This parameter has a default value of `0` and is defined in units of cT [µm].

Delay Time

This field sets the delay time for a pulsed excitation type. Its default value is `4`, and is in units of the value given in the **Ramp/Pulse Time** field. This option has no effect for a CW or impulse launch.

Shutoff Time

This field sets the time at which a CW or pulsed excitation should be turned off. Its default value is infinite, therefore disabling the shutdown of the excitation. It is given in units cT [µm], and, for the CW case, includes the time specified in the **Ramp Off Time** field.

Chirp Coefficient

This field sets the chirp coefficient for a CW or Pulsed simulation. Its units are µm^{-2}.

4.E. Polarization Manipulation

Once the **Model Dimension** and **Polarization** has been defined the default polarization is clearly defined. For example, a setting of 3D TE corresponds to a major field of E_x and a minor field of E_y. Generally speaking, the launch field primarily consists of the major component of the field with little or no energy in the minor component. The Launch Polarizer, which can be opened by clicking the **Polarizer…** button in the Launch Parameters dialog, allows this default behavior to be changed to achieve a custom launch polarization.

The Launch Polarizer has two modes, each of which affects the polarization differently. The first mode is the Transformer Mode, which is the default, transforms the polarization of the launch field via a Jones Matrix **T**. The polarization setting of the untransformed launch field therefore affects the final polarization. It uses both built-in and custom Jones Matrices. The second mode is the Controller Mode which forces to the launch polarization to the set value while conserving the input power. The polarization setting of the untransformed launch field does not effect the final polarization.

Transformer Mode

This default polarization mode allows the user to define a Jones Matrix to modify the polarization of the launch field. The transformation can be expressed as:

$$\begin{pmatrix} u_x' \\ u_y' \end{pmatrix} = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}$$

To place the Launch Polarizer in Transformer Mode, set the **Polarizer Mode** to *Transformer*. The type of polarization manipulation is set via the **Launch Polarizer** option:

- *None*

  This option disables the Launch Polarizer.

- *Custom*

  This option allows the user to define a custom 2x2 complex-valued Jones Matrix.

- *Linear*

  This option produces a linear polarization at a specified angle via a Jones Matrix of

$$\mathbf{T} = \operatorname{sign}\left[\cos\phi\right] \begin{pmatrix} \cos^2\phi & \cos\phi\sin\phi \\ \cos\phi\sin\phi & \sin^2\phi \end{pmatrix}$$

where ϕ is the angle specified in the **Angle** field.

- *Circular Left* and *Circular Right*

  These options produce a left or right circularly polarized field. The Jones Matrix is therefore

$$\mathbf{T} = \frac{1}{2}\begin{pmatrix} 1 & -i \\ i & 1 \end{pmatrix}, \ \mathbf{T} = \frac{1}{2}\begin{pmatrix} 1 & i \\ -i & 1 \end{pmatrix}$$

  respectively.

- *Rotation*

  This option produces a simple rotation via a Jones Matrix of

$$\mathbf{T} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}$$

  where ϕ is the angle specified in the **Angle** field.

Controller Mode

This polarization mode allows polarization of the launch field to be set directly while preserving the input power. This is achieved mathematically as:

$$\begin{pmatrix} u_x' \\ u_y' \end{pmatrix} = \left(\sqrt{u_x^2 + u_y^2}\right)\begin{pmatrix} \text{sign}\left[\cos\phi\right]\sqrt{1-k}e^{i\phi_x} \\ \text{sign}\left[\sin\phi\right]\sqrt{k}e^{i(\phi_x+\Delta\phi)} \end{pmatrix}$$

where ϕ_x is the actual phase of the field u_x, k is the power splitting parameter, and Δϕ is the desired phase difference between the fields u_x and u_y. The power splitting parameter k is defined as:

$$k = \frac{1}{2}\left[1 - \cos(2\eta)\cos(2\varepsilon)\right]$$

where η is the azimuth and ε is the ellipticity and are measured in degrees. The desired phase difference Δϕ is defined as:

$$\Delta\phi = \arcsin\left(\frac{1}{2}\frac{\sin(2\varepsilon)}{\sqrt{k(1-k)}}\right)$$

To place the Launch Polarizer in Controller Mode set the **Polarizer Mode** to *Controller*. The type of polarization manipulation can be set with the **Launch Polarizer** option:

- *None*

  This option disables the Launch Polarizer.

- *Custom*

  This option allows arbitrary values of the **Azimuth** and **Ellipticity** to be defined.

- *Linear*

This option sets the polarization to linear at an angle set by the **Azimuth** field. The ellipticity is forced to equal zero.

- *Circular Left* and *Circular Right*

  These options set the polarization to be left and right circular. The azimuth is forced to 45 degrees for right circular and -45 degrees for left circular resulting in k=0.5 and $\Delta\phi$ is set to 90 for left circular and -90 for right circular. The azimuth setting has no effect and is disabled.

4.F. Launching Multiple Fields

It is possible to define multiple launch fields in the same design file. This is controlled by several options at the top of the Launch Parameters dialog shown in Fig. 4-1. The **Launch Field** setting indicates the number of the individual launch field currently being edited. To add an additional launch field, click the **New** button and select the characteristic of that individual field in the remaining fields of the launch dialog as documented above. The **<<** and **>>** buttons can be used to edit each launch field defined. To delete an individual launch field, first select the launch field to be deleted and then click the **Delete** button.

When performing a simulation, the individual launch fields are calculated and linearly combined to create the final launch field. The relative power and phase of each launch field can be controlled with the **Power** and **Phase** fields in the upper right of the dialog respectively. It is essential to note that when interpreting the power, the setting of the **Normalization** option is critical. If two individual launch fields are combined without first being normalized to *Unit Power*, the values in the **Power** field will not in general reflect the actual relative power unless the individual fields have the same amplitude and shape.

4.G. Special Source Properties

BandSOLVE, RSoft's design and simulation tool for photonic band structures, can use the FullWAVE simulation engine. To accommodate this, FullWAVE has a launch option that allows BandSOLVE to work correctly. This option is set in the FDTD Advanced Parameters dialog which can be accessed via the **Advanced...** button in the FullWAVE Simulation Parameters dialog.

Repeat in Each Unit Cell (non-cubics)

This option periodically repeats the launch field within a structure. When BandSOLVE is used with a FullWAVE engine, a non-cubic domain must be made cubic by simulating more than one unit cell. Thus, the source must be repeated in order to preserve the periodicity. This option is automatically set when using BandSOLVE.

4.H. Launching a Plane Wave

Though it is possible to launch a plane wave in FullWAVE, there are several things to consider. Since FullWAVE uses a finite spatial domain, but plane waves are inherently infinite, it is important to understand how a launch plane behaves at the domain boundaries.

The basic way to launch a plane wave is to use a *Rectangle* launch type that is wider than the simulation domain. If the structure is periodic, then it is recommended that you use periodic boundary conditions. If not, then you should use PML, symmetric, or asymmetric boundaries.

4.H.1. Using Periodic Boundary Conditions

The structure must be periodic to use periodic boundary conditions as this type of excitation can excite transverse lattice modes. For a discussion of periodic boundary conditions, see Section 6.C.

Non-tilted

A non-tilted plane wave with periodic boundary conditions is the easiest to create. Simply enable periodic boundary conditions and create a *Rectangle* launch type that is wider than the simulation domain.

Tilted

To launch a plane wave at an angle, the appropriate launch angles will need to be set and a phase difference must be given across the periodic boundaries. Using the **Auto** option for **ka/pi Value** in the Advanced Parameters dialog will attempt to set the ka/pi values appropriately based on the following vector relation:

$$\mathbf{k} = k_0 n_0 \left[\hat{x}\left(D_x \sin\phi \cos\theta\right) + \hat{y}\left(D_y \sin\theta\right) + \hat{z}\left(D_z \cos\phi \cos\theta\right) \right]$$

where *k0* is the free space wavelength in units of k, *n0* is the index at the launch plane, and *Dx*, *Dy*, and *Dz* are the domain sizes along each coordinate. The value of *n0* can be set manually via the variable `fdtd_bc_ka_n0`.

The **Auto** option for **ka/pi Value** requires the **Use Complex Implementation** in the Advanced Parameters dialog to use the complex valued field in the simulation. See Section 6.E.3 for more details.

4.H.2. Using PML Boundary Conditions

For non-periodic structures, PML boundaries usually need to be used. The basic consideration for PML boundaries is that while the field is normally attenuated within the PML region, the plane wave should not be.

In order to meet this requirement, the field is launched in the PML by default. This is controlled by the variable `fdtd_launch_pml`. It’s values are:

Value	Description
0	Truncates the field and does not launch in the PML.

1 Launches the field unchanged in the PML, which results in a good non-tilted plane wave. This is the default value for this parameter.

2 Automatically calculates the required gain and attenuation in the PML for proper launching. Since this applies gain, this option is off by default for safety. The variable `fdtd_launch_pml_factor` adjusts the gain/loss by the specified factor (default=`1`), in case the predicted amount is not ideal.

4.H.3 Other Options

These two additional options allow for multiple planes at right angles to each other to be coupled together in phase to create a plane wave emanating from the interior of the domain:

This behavior of these features are subject to change in future releases.

Setting the primary direction of a launch:

The primary direction of the launch plane can be set via the variables `fdtd_launch_direction` or `fdtd_launch_direction_#` for a specific launch field #. Normally, the launch angles determine the primary direction. For shallow angles, the primary direction is +Z. As the angles exceed +/- 45, 90, etc., the primary direction changes. The values `+/- 1`, `2`, `3` correspond to +/- X,Y,Z. A value of 0 disables this feature and uses the default direction.

Setting the phase reference:

The variable `launch_phase_ref` is used to control the phase reference for the *Rectangle* launch type (e.g. a plane wave). The default value of `0` references the phase to the launch position. A value of `1` allows the phase to be referenced to the coordinate origin.

4.I. Using an Enclosed Launch

An enclosed launch allows the user to launch a plane wave inside an enclosed area/volume, and is useful for scattering calculations. Any field that exists within the enclosed area/volume consists of both launch and scattered field; any field that exists outside the enclosed area/volume consists only of scattered field.

To use an enclosed launch, set the launch **Type** to *Rectangle* and select the **Enclosed Launch** option in the Spatial Excitation portion of the Launch Properties dialog. The size of the enclosed area/volume is set by the **Width** (X), **Height** (Y), and **Length** (Z) options in the Launch Parameters dialog, and the incident angle is set via the **Phi** and **Theta** options as useful. This feature works well with the far field outputs discussed in Section 6.B.2 to find the far field projection of the scattered field.

5

Monitoring and Analysis

FullWAVE calculates the electromagnetic field as a function of time and space for a user-defined refractive index structure in response to an excitation, and can display this field at time intervals as described in the previous chapter. To record this field information as a function of time or frequency for post processing and analysis purposes, we introduce the concept of monitors. A monitor records both field values and common electromagnetic quantities such as power or energy density as a function of time or frequency. This output can be used as final output(s), or as intermediate steps for further post-processing.

This chapter will first discuss the types of output that a FullWAVE monitor can produce, then the steps for creating a monitor, and finally a detailed description of the usage of monitors. The last few sections will discuss several example monitor configurations.

5.A. Monitor Overview

This section provides an overview of the possible types of monitor output and a description of how to enable monitor output.

5.A.1. Main Types of Monitor Output

A FullWAVE monitor can produce two main types of output as well as perform a FFT/DFT analysis of these output types.

- *Single-Value Output*

  The single-value output types are the most commonly used output types in FullWAVE. This type of output consists of a single value for a given quantity recorded as a function of time. This includes measurements of one or more field components at a single fixed point, power flow through the monitor domain, energy density in the monitor domain, or the result of an overlap integral. It can produce a basic summary of the simulation results as a function of time. In addition to being saved to disk, single-value output is displayed in the lower portion of the FullWAVE simulation window during a simulation. This type of output is discussed in Section 5.C.

- *Spatial Output*

  The spatial output types consist of measurements which are recorded as a function of space at fixed values of time. This includes measurements of one or more field components, the amplitude of the pointing vector, energy density, or power flow along a specified axis, all as a function of space. Spatial output options require much more disk storage space than single-value output, but are sometimes necessary in order to obtain desired data. This type of output is discussed in Section 5.D.

- *Frequency Output (FFT/DFT)*

  FullWAVE monitors can perform a frequency analysis in order to obtain the spectral characteristics of a structure over a range of wavelength/frequency values. Both DFT and FFT based algorithms can be used. The names 'FFT' and 'DFT' are slight misnomers as both methods are in fact discrete. However, we have used these names in order to be consistent with existing nomenclature in FDTD literature. An FFT analysis can be performed on only single-value output and is calculated at the end of a simulation. A DFT analysis can be performed on both single-value output and spatial output at either one frequency or over a range of frequencies and is calculated as the simulation progresses. FFT output is discussed in Section 5.E and DFT output is discussed in Section 5.F.

5.A.2. Enabling Monitor Output

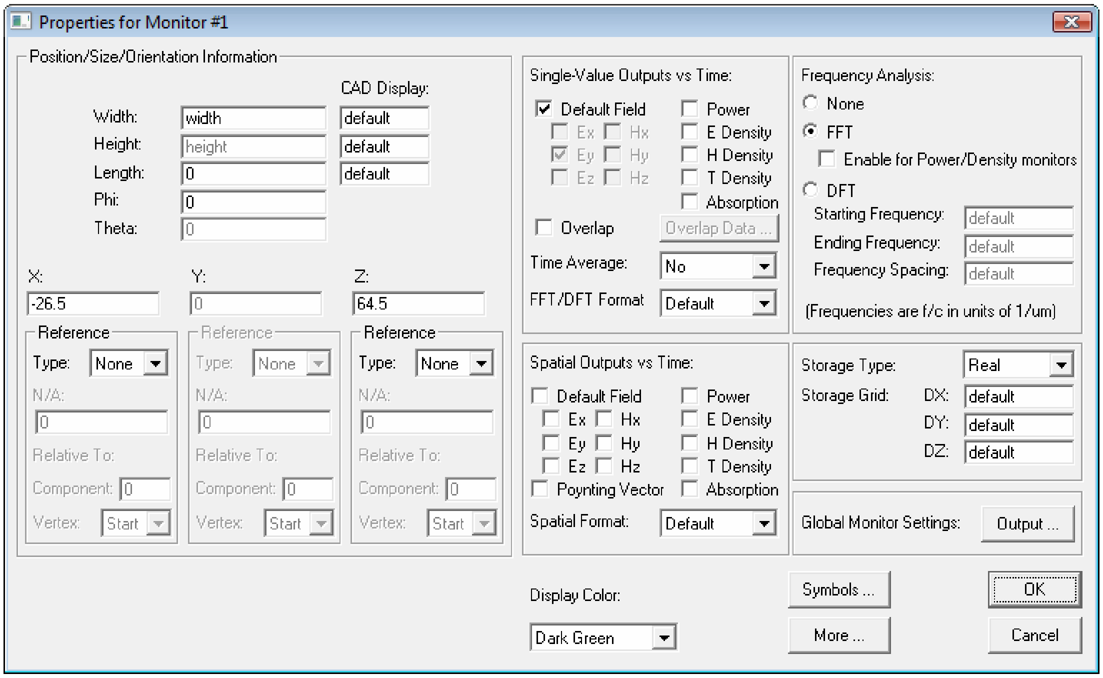
When a design file that contains monitors is run, any single-value output (except when DFT frequency analysis is selected) will be displayed in the bottom portion of the simulation window below the field display. The data will be automatically sampled as the simulation progresses at intervals defined by the **Monitor Time**. The interval at which the simulation window display updates can be set directly via the variable `fdtd_monitor_update_time`. Setting this to a larger value can improve speed in rare cases.

If an **Output Prefix** has been entered in the FullWAVE simulation parameters dialog before running the simulation and the appropriate options are selected in the Output Options dialog box, the results from the monitors will be saved. Depending on the type of single-value or spatial output selected, as well as any frequency analysis options, different files will be created. See specific output sections below for output file formats.

5.B. Monitor Position & Geometry

Monitors are added to a design file by selecting the **Monitor** drawing mode form the left toolbar in the CAD. The monitor properties dialog will appear as shown in Fig. 5-1. The monitor properties window contains several regions, each of which will be discussed in more detail in the following sections.

A monitor measures a quantity of interest at either a single point or over a specified domain as a function of time. The domain of the monitor is defined as the portion of space which the monitor occupies, and is important for some of the single-value output options and all of the spatial output options. This section will cover the definition of a monitor domain, or geometry.



Properties for Monitor #1

Position/Size/Orientation Information

		CAD Display:
Width:	width	default
Height:	height	default
Length:	0	default
Phi:	0	
Theta:	0	

X:	Y:	Z:
-26.5	0	64.5
Reference	Reference	Reference
Type: None	Type: None	Type: None
N/A: 0	N/A: 0	N/A: 0
Relative To:	Relative To:	Relative To:
Component: 0	Component: 0	Component: 0
Vertex: Start	Vertex: Start	Vertex: Start

Single-Value Outputs vs Time:

☑ Default Field ☐ Power
☐ Ex ☐ Hx ☐ E Density
☑ Ey ☐ Hy ☐ H Density
☐ Ez ☐ Hz ☐ T Density
☐ Absorption
☐ Overlap
Time Average: No
FFT/DFT Format: Default

Frequency Analysis:

○ None
● FFT
☐ Enable for Power/Density monitors
○ DFT
Starting Frequency: default
Ending Frequency: default
Frequency Spacing: default
(Frequencies are f/c in units of 1/um)

Spatial Outputs vs Time:

☐ Default Field ☐ Power
☐ Ex ☐ Hx ☐ E Density
☐ Ey ☐ Hy ☐ H Density
☐ Ez ☐ Hz ☐ T Density
☐ Poynting Vector ☐ Absorption
Spatial Format: Default

Storage Type: Real
Storage Grid: DX: default
DY: default
DZ: default

Global Monitor Settings:

Display Color: Dark Green

Figure 5-1: The monitor properties dialog which is used to define monitors for analysis.

5.B.1. Monitor Geometry Overview

The geometric parameters of a FullWAVE monitor are defined by a center point (X_0, Y_0, Z_0), values for the width w, height h, and length l, and two tilt angles ϕ and θ. These parameters control the position, size, orientation, and therefore the domain of the monitor.

Figure 5-2 shows the definition of the width, height, and length of a monitor, and Fig. 5-3 shows the definition of the tilt angles.

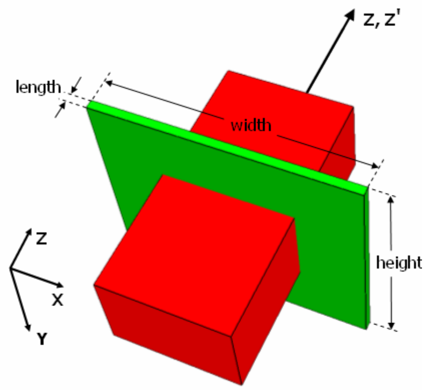


Figure 5-2: Illustration of how the monitor width, height, and length are defined. Since this monitor is not rotated, the Z and Z' axes are the same.

Note that the width, height, and length of a monitor are always made relative to the normal Z' axis, even if the monitor is rotated as shown in Fig. 5-3.

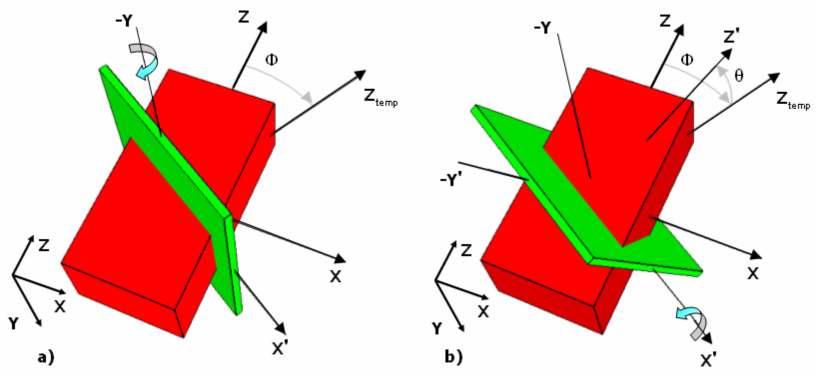


Figure 5-3: Illustration of how the monitor tilt angles phi and theta are defined. Though not necessary, it is easier to understand these angles if the ϕ rotation is performed first (a), and then the θ rotation (b). The temporary axis Z_{temp} is shown for illustrative purposes.

As indicated by Fig. 5-2 and Fig. 5-3, each monitor has its own local coordinate system (X',Y',Z'). This local coordinate system is defined for the values:

$$X' \in \left[\frac{-w}{2},\frac{w}{2}\right] \;,\; Y' \in \left[\frac{-h}{2},\frac{h}{2}\right] \;, \text{and } Z' \in \left[\frac{-l}{2},\frac{l}{2}\right]$$

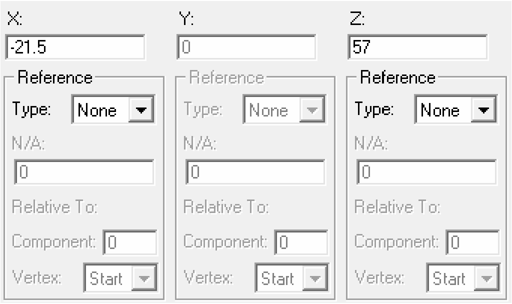
where *w*, *h*, and *l* are the width, height, and length of the monitor domain. These local coordinates are related to the global coordinate system for the design file by the following transformation:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix} \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}$$

In the 2D case, the above relation is only defined for X and Z and the value of θ is assumed to be 0.

5.B.2. Specifying Monitor Position

Figure 5-4 shows the portion of the monitor properties dialog that defines the monitor's center point (X_0, Y_0, Z_0). This is done by defining the center point in either actual coordinates, or in reference to other components in the design file. This process is identical to the definition of a component's position in the RSoft CAD. Please see Chapter 4 in the CAD manual.

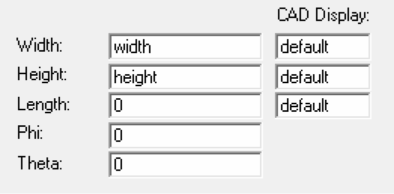


X:	Y:	Z:
-21.5	0	57
Reference	Reference	Reference
Type: None	Type: None	Type: None
N/A: 0	N/A: 0	N/A: 0
Relative To:	Relative To:	Relative To:
Component: 0	Component: 0	Component: 0
Vertex: Start	Vertex: Start	Vertex: Start

Figure 5-4: The portion of the monitor properties dialog box which controls the monitor position, or center point.

5.B.3. Specifying Monitor Size & Orientation

The monitor's size and orientation are controlled by the quantities labeled in Fig. 5-2 and Fig 5-3 and are set in the portion of the monitor properties dialog shown in Fig. 5-5.



		CAD Display:
Width:	width	default
Height:	height	default
Length:	0	default
Phi:	0	
Theta:	0	

Figure 5-5: The portion of the monitor properties dialog box which controls the monitor size and orientation.

Width, **Height**, and **Length**

These options set the width (along X'), height (along Y'), and length (along Z') of the monitor in µm. The default values are equal to global **Component Width**, **Component Height**, and `0` respectively. These parameters can be set to an infinite value via the keyword `inf`.

CAD Display

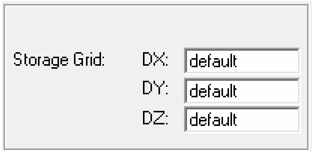
These fields set the X, Y, and Z display sizes of the monitor respectively. These sizes are for display purposes only and will not effect other monitor properties or the simulation results. The default values for these fields correspond to the values given in **Width**, **Height**, and/or **Length** and are in units of µm.

Phi and **Theta**

These fields set the angles ϕ and θ illustrated in Fig. 5-3. Phi (ϕ) is defined as a rotation around the Y axis, and is measured from the Z axis. Theta (θ), which is only used in 3D simulations, is defined as a rotation around the X' axis, and is measured from the X'Z plane. Both angles are in units of degrees.

5.B.4. Specifying Monitor Grid for Calculation & Storage

The lower right region of the monitor properties dialog sets storage-related parameters. By default, the monitor uses the simulation grid for single-value outputs that require integration over a domain (power/densities) and for spatial output. The **Storage Grid** settings allow for user control over this grid in the local monitor coordinates. The three options specify an alternate grid along the monitor axes (X', Y', Z') to be used for spatial output in order to conserve the usage of system memory.



Storage Grid: DX: default
DY: default
DZ: default

Figure 5-6: The portion of the monitor properties dialog where storage parameters are set.

When using rotated monitors, the default monitor grid for a particular axis may not be equal to the global grid size along that axis. As an example, consider a monitor that has been rotated by ϕ=90 so that the local X' and Z' axes are oriented along the global Z and X axes respectively. In this case, the **DX** setting here refers to the grid along the local coordinate X' which is along the global Z axis.

Setting Output Coordinates

By default, the coordinates of any spatial data are the normalized monitor coordinates discussed in Section 5.B.1. In some cases, it can be useful to disable this behavior and use the global coordinate system. To do this, set the variable `monitor_coords` to `1`. Note that this feature is not meant to be used with arbitrarily tilted monitors.

5.B.5. Additional Settings

A monitor has the following additional settings:

Display Color

This sets the display color used to display the monitor in the drawing window.

Global Monitor Settings

The **Output...** button opens the Output Options dialog described in Section 3.E where global monitor output options can be set.

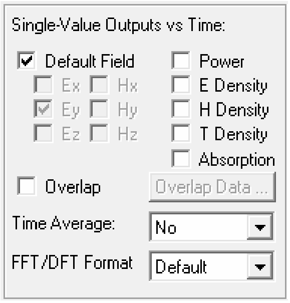
Setting Monitor Names in Plots and Data Files

By default, the name of the monitor in all plots and output files produced by FullWAVE will be related to the monitor number. It is possible, however, to set the monitor name directly which can be useful in cases where multiple monitors exist at the same position, etc. To set the names of individual monitors, click the **More...** button in the monitor properties dialog and set the **Component Name** to the desired unique name. This behavior can be disabled if needed by removing any **Component Name** settings or by setting the variable `monitor_labeling` to `0`.

5.C. Single-Value Output Types

The upper center region of the monitor properties dialog defines the type of single-value output desired. The data output from these data types will be saved in a file `<prefix>.tmn` with a corresponding WinPLOT command file `<prefix>.ptm`. The results from these output types will be displayed in the lower portion of the simulation window during a simulation. For information on the file format used, see Appendix B.B.

The single-value output is measured at time intervals given by the **Monitor Time** specified in the FullWAVE simulation parameters dialog box. It is also possible to perform a frequency analysis of these outputs: see Section 5.E and Section 5.F for more details.



Single-Value Outputs vs Time:

☑ Default Field □ Power
□ Ex □ Hx □ E Density
☑ Ey □ Hy □ H Density
□ Ez □ Hz □ T Density
□ Absorption
□ Overlap
Time Average: No
FFT/DFT Format Default

Figure 5-7: The portion of the monitor properties dialog box which sets the desired time output types.

The formulae listed in this section are illustrative; the actual formulae used are proprietary.

5.C.1. Field Values at a Single Point

These options can record any of the field components at the center of the monitor as a function of time. For example, the option *Ex* will return $E_x(X_0, Y_0, Z_0, t)$ where (X_0,Y_0,Z_0) is the center of the monitor.

The *Default Field* setting automatically chooses an appropriate field based on the model dimensions and the polarization setting. In 2D, the default field is *Ey* for TE and *Hy* for TM. In 3D, the default field is *Ex* for TE and *Ey* for TM. In 2D, only 3 components of the field are simulated and so only these field components will be output. The 3 fields depend on the polarization choice for the simulation.

The monitor output shown in the simulation window as well as plots viewed via the `<prefix>.ptm` file shows the square of the fields, namely E_x^2, E_y^2, etc. The output saved in the file `<prefix>.tmn` is the exact field values (un-squared) from the simulation data.

5.C.2. Power

This option records the power flow through the monitor domain as a function of time. The integral used to evaluate this quantity is

$$S(t) = \frac{1}{S_0} \mathrm{Re}\left[\int_A \left[\mathbf{E}(t) \times \mathbf{H}^*(t) \right] \cdot dA \right]$$

where **E** and **H** are the electric and magnetic fields and S_0 is the launch power. The integral is evaluated over an area A. This definition of this area depends on the monitor **Length** (measured along the Z' direction):

- *Zero Length*

  When the **Length** is `0`, the monitor measures the power flow along the Z' direction at the plane (X', Y', Z'=0) through the line or rectangle defined by the monitor **Width** and **Height**.

- *Non-Zero Length*

  When the **Length** is non-zero, the monitor measures the power flow through all surfaces of the monitor volume defined by the **Width**, **Height**, and **Length**. This can be useful to measure the power loss in optical cavities when solving for the quality factor Q.

By default, the launch power S_0 is the power of the first launch field. To change the launch field used to compute S_0, set the variable `fdtd_p0_index` to the number of the desired launch field. To effectively remove all normalization (i.e. set S_0, to 1), set `fdtd_p0_index` = `0`, but note that this is not recommended unless absolutely needed (i.e. during nonlinear calculations) since the monitor results cannot be interpreted in a standard way.

The monitor output shown in the simulation window as well as plots viewed via the `<prefix>.ptm` file shows the absolute value of the power $|P(t)|$. The output saved in the file `<prefix>.tmn` is the signed square root of the power $p(t)$= sign[$P(t)$]*sqrt[$|P(t)|$].

5.C.3. E Density

This option records the electric field energy density. The integral used to evaluate this quantity is

$$U_E(t) = \frac{1}{2} \int_V \mathrm{Re}\left[\varepsilon(\mathbf{r}') \right] \left| \mathbf{E}(t) \right|^2 dV$$

where **E** is the electric field, ε is the spatially dependent permittivity, and V is the volume of the monitor defined by the **Width**, **Height**, and/or **Length**. The result is normalized by the input power; set `fdtd_density_normalization` to `0` to disable this normalization. This output type can be used with the volume weight feature described in Section 5.C.10.

The monitor output shown in the simulation window as well as plots viewed via the `<prefix>.ptm` file show the energy density. The output saved in the file `<prefix>.tmn` is the square root of the density.

5.C.4. H Density

This option records the magnetic field energy density. The integral used to evaluate this quantity is

$$U_H(t) = \frac{1}{2}\int_V \mu \left|\mathbf{H}(t)\right|^2 dV$$

where **H** is the magnetic field, μ is assumed to be 1, and V is the volume of the monitor defined by the **Width**, **Height**, and/or **Length**. The result is normalized by the input power; to disable this normalization, set `fdtd_density_normalization` to `0`. This output type can be used with the volume weight feature described in Section 5.C.10.

The monitor output shown in the simulation window as well as plots viewed via the `<prefix>.ptm` file shows the energy density. The output saved in the file `<prefix>.tmn` is the square root of the density.

5.C.5. T Density

This option records the total energy density, and is equivalent to the sum of an *E Density* and *H Density* monitor defined over the same volume. This monitor type can be used with the volume weight feature described in Section 5.C.10.

The monitor output shown in the simulation window as well as plots viewed via the `<prefix>.ptm` file shows the energy density. The output saved in the file `<prefix>.tmn` is the square root of the density.

5.C.6. Absorption

This option records the total power absorbed within the domain of the monitor defined by the **Width**, **Height**, and/or **Length**. The integral used to evaluate this quantity is

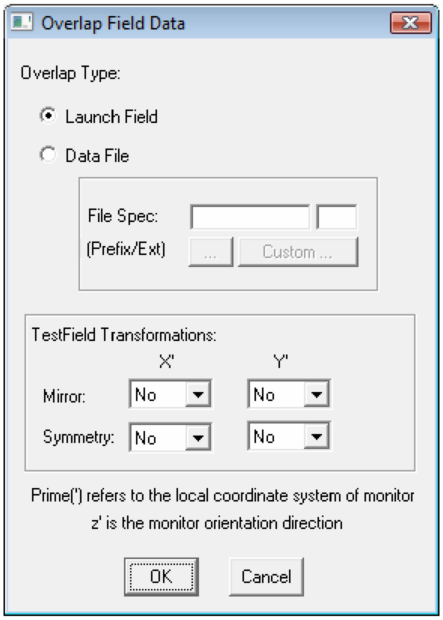
$$U_A(t) = \omega\int_V \mathrm{Im}\left[\varepsilon(\omega)\right]\left|\mathbf{E}(t)\right|^2 dV$$

where **E** is the electric field, ε is the spatially dependent permittivity, and V is the volume of the monitor defined by the **Width**, **Height**, and/or **Length**. This result is normalized by the input power, to disable this normalization, set `fdtd_density_normalization` to `0`. This monitor type can be used with the volume weight feature described in Section 5.C.10.

The monitor output shown in the simulation window as well as plots viewed via the `<prefix>.ptm` file shows the absorption. The output saved in the file `<prefix>.tmn` is the square root of the absorption.

5.C.7. Overlap

This option computes the overlap integral between the propagating field at the location of the monitor, $\phi_p(t)$ and a test field ϕ_0. By default, the test field is the first launch field, but data files can be used as well (see below). Figure 5-8 shows the Overlap Field Data dialog which can be opened by clicking the **Overlap Data…** button in the monitor properties dialog.



Overlap Field Data

Overlap Type:

● Launch Field

○ Data File

File Spec:
(Prefix/Ext)

TestField Transformations:

	X'	Y'
Mirror:	No	No
Symmetry:	No	No

Prime(') refers to the local coordinate system of monitor
z' is the monitor orientation direction

Figure 5-8: The Overlap Field Data dialog.

- The overlap integral is calculated by translating the test field to the center location of the monitor and rotating it so that it lies in the X'Y' plane of the local monitor coordinates described by the monitor tilt angles **Phi** and **Theta** and illustrated in Fig. 5-3. Overlap monitors distinguish between forward and backward traveling components, so it is important to ensure that the monitor is oriented the correct direction.
- The monitor result is normalized to both the test field as well as the input field. Set the variable `fdtd_total_overlap_monitor` to 0 to disable this normalization.
- The **Width**, **Height**, and **Length** settings usually have no effect for an *Overlap* monitor as the overlap is computed over the entire size of the computational domain along the plane defined by the X' and Y' axes of the monitor. When combining multiple monitor outputs types, the width, height, and or length might have an effect depending on which single-value output types are enabled. See Section 5.C.9 for details.

The monitor output shown in the simulation window as well as plots viewed via the `<prefix>.ptm` file shows the square of the overlap. The output saved in the file `<prefix>.tmn` is the actual signed overlap function.

Using Custom Data Files

To use custom data files for the overlap test field, set **Overlap Type** to *Data File*. The **File Spec** option is then used to specify the data file(s) to be used for the overlap. These files can be automatically set by a file **Prefix** and **Extension** or manually using the controls provided by clicking **Custom…**.

- *Automatic Method:*

  The automatic method provides a simple way to define the files used for individual components by setting a file prefix and extension. The files must have the default output naming style used by RSoft software for mode and field output: `<prefix>_<comp>.<ext>` where `<comp>` is the field component. For example, if the data files are named `myprefix_ex.m00`, `myprefix_ey.m00`, etc., enter `myprefix` and `m00` for the **Prefix** and **Extension** respectively. These settings can be easily set by choosing one of the field component files via the '**...**' button. As a convenience, the data files the software will look for based on the prefix and extension specified can be seen by clicking the **Custom** button. Note that this feature will also work with older naming conventions such as `<prefix>.m00`, `<prefix>.n00`, etc, though the new style is preferred.

  The software will, based on the **Model Dimension** and **Polarization**, attempt to locate the files corresponding to the necessary field components. If any of the necessary component files are missing, the software will attempt to either ignore or calculate the required components. In 2D, a TE polarization requires E_y data and a TM polarization requires E_x. If present, it will also include corresponding H field data. In 3D, a TE polarization requires E_x data and TM polarization requires E_x. If present, it will also include the other E field and/or corresponding H field data. Generally, all necessary components should be made available; in some cases such as low index-contrast simulations, all components may not be necessary.

- *Custom Method:*

  To manually define the launch files used for individual components, click the **Custom** button, uncheck **Auto**, and define the necessary components. The major and minor components refer to different components of the electric **E** and magnetic field **H** and the exact definition depends on the **Model Dimension** and **Polarization** used.

Field files produced by mode solving (BeamPROP, FemSIM, TmmSIM, etc) or propagation (BeamPROP, FullWAVE, etc) will be appropriately named and formatted and can easily be used with the automatic method. Data files manually created must be in the standard file format discussed in Appendix B and can be used with the automatic method if named appropriately.

It is possible to use a variable to define a launch file: set the launch file option to `$<variable>` where `<variable>` is a variable created in the Symbol Table set equal to the name of the desired launch file. If a 2D data file is launched into a 3D simulation it will be interpreted as a radial profile and a 3D field distribution is automatically generated. The **Mode** option sets the azimuthal mode number.

5.C.8. Time-Averaging Single-Value Output

The **Time Average** option for single-value output indicates that all single-value outputs should be averaged over a moving window equal to one cycle of the excitation and multiplied by 2 to more easily compare with peak values. This is mathematically expressed as

$$Q_{avg}(t) = \left| \frac{2}{T_0} \int_{t-T_0}^{t} |M(t)|\, dt \right|$$

where $M(t)$ is the non-averaged monitor output as described above and T_0 is the period over which the average is taken and is equal to the **Free Space Wavelength** specified in the Global Settings dialog box. Note that for power monitors, in order to account for power flow in opposite directions the modulus of $M(t)$ is not taken.

A time averaged response is useful when studying steady state output. It is particularly useful during a CW parameter scan (see CAD manual for more on scanning), since the final monitor values are plotted versus the scanned variable(s) and without time averaging the plot would show the monitor value at an arbitrary phase which is generally useless. The **Time Average** field in the dialog has three settings: *No*, *Yes*, and *Auto*. The *Yes*/*No* settings enable/disable time averaging as expected, while the *Auto* option enables it for CW excitation and disables it for pulsed excitation. This feature is useful since pulsed excitation is usually used to obtain spectral information as discussed below, which requires time averaging to be disabled. The *Auto* option avoids the user from having to toggle the time averaging setting when switching between CW and pulsed excitation.

5.C.9. Combining Multiple Single-Value Output Types

When enabling overlap and power/density single-value outputs from a single monitor it is necessary to consider the domain that the monitor uses. Overlap monitors extend the integration range to cover the entire cross-section of the simulation domain whereas power and energy monitors use the actual finite extent of the monitor defined by the **Width**, **Height**, and **Length**. When combining an overlap monitor with a power/energy monitor, the software will not extend the integration range as usual which can affect the results of the overlap monitor. Careful attention should be made to ensure that the monitor will measure the intended results. If it is not possible to achieve the desired results, simply create two monitors at the same position, one for overlaps and one for power/densities.

5.C.10. Absorption & E/H/T Density Results for Specific Materials

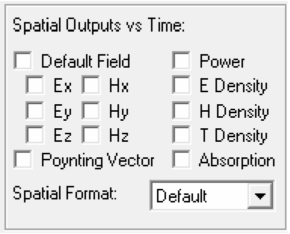
It is possible to use Absorption & E/H/T Density monitors to measure results within certain materials that intersect the monitor domain. This is useful, for example, to measure the energy absorbed within a specific material only or the energy density within an arbitrarily-shaped region. To use this feature, first define an energy monitor that encloses the region/material(s) of interest. To accomplish this, a weight profile must be defined and a monitor setting must be made:

- By default, the weight profile is 0 in the entire simulation domain. To add a material and therefore set the weight profile to 1 in the regions where that material is used, open the Material Editor, select the material of interest, choose the Semiconductor tab, click the **Edit Table…** button, and define the symbol `vm_weight` to a value of `1`. Repeat this process for any other materials that should be included. Any material that does not have this setting will not be included.

- By default, monitors do not use the weight profile function and output the absorption within the entire monitor domain. To enable the use of this feature, open the monitor properties dialog, click **More...**, and select the **Use Volume Monitor Weight Profile** option.
- When using the `vm_weight` option, tt is possible to include only grid points that completely fall within the monitor domain in the calculation and effectively ignore points near material interfaces where FullWAVE has averaged the refractive index. This averaging is necessary to achieve accurate results with a discrete grid but can sometimes skew absorption or density results. To ignore these points, set the variable `vm_weight_bias` to `1`.

5.D. Spatial Output Types

The lower center region of the monitor properties dialog defines the type of spatial output desired. The output data will be saved in an individual file for each output option selected as well as each slice in time. The spatial output is measured at time intervals given by the **Spatial Output Time** specified in the Output Options dialog. It is also possible to perform a frequency analysis of these outputs: see Section 5.E for more details. For 3D monitors (where **Width**, **Height**, and **Length** are all non-zero) the format of the output can be controlled in the Output Options dialog.



Spatial Outputs vs Time:

□ Default Field □ Power
□ Ex □ Hx □ E Density
□ Ey □ Hy □ H Density
□ Ez □ Hz □ T Density
□ Poynting Vector □ Absorption

Spatial Format: Default

Figure 5-9: The portion of the monitor properties dialog where spatial output types are set.

The formulae listed in this section are illustrative; the actual formulae used are proprietary.

5.D.1 Field Output over Monitor Domain

These options record any field component over the monitor domain at specific points in time. For example, choosing the option **Ex** will return $E_x(\mathbf{r}',t_S)$ where $\mathbf{r}'=(X',Y',Z')$ varies over the entire monitor domain as shown in Fig. 5-2 and the values of t_S are fixed multiples of the **Slice Time** given in the FullWAVE simulation parameters dialog. The grid sizes used to store data can be increased in order to conserve system memory. More details on this can be found in Section 5.B.4.

The **Default Field** setting automatically chooses an appropriate field based on the number of dimensions and the polarization setting of the design file. In 2D, the default field is **Ey** for TE and **Hy** for TM. In 3D, the default field is **Ex** for TE and **Ey** for TM.

In 2D simulations, only 3 components of the field are simulated and so only these field components can be output. The fields used depend on the polarization choice for the simulation. In 3D, all six components can be output.

This output is saved in files such as `<prefix>_m#_t#_ex.pcs` (for the E_x component) with accompanying `.dat` data file where `<prefix>` is the **Output Prefix**, `m#` is the monitor number, and `t#` is the simulation time at which the data was saved in cT. The displayed data as well as the raw data file contain the exact field data.

5.D.2. Power

This type of output records the power flow along the Z' direction as defined in Fig. 5-2. This is equivalent to the Z' component of the Poynting Vector. The expression used to evaluate this quantity is

$$S(\mathbf{r}',t_s) = \mathrm{Re}\left[\mathbf{E}(\mathbf{r}',t_s)\times\mathbf{H}^*(\mathbf{r}',t_s)\right]_{z'}$$

where **E** and **H** are the electric and magnetic fields and **r**' is the spatial coordinate. The grid sizes used to store data can be increased in order to conserve system memory. More details on this can be found in Section 5.B.4.

This output is saved in a file `<prefix>_m#_t#_pow.pct` with accompanying `.dat` data file where `<prefix>` is the **Output Prefix**, `m#` is the monitor number, and `t#` is the simulation time at which the data was saved in cT. The displayed data as well as the raw data file contain the exact power data.

5.D.3. E Density

This type of output records the electric field density as a function of space at specific times. The expression used to evaluate this quantity is

$$U_E(\mathbf{r}',t_s) = \frac{1}{2}\mathrm{Re}\left[\varepsilon(\mathbf{r}')\right]\left|\mathbf{E}(\mathbf{r}',t_s)\right|^2$$

where **E** is the electric field, ε is the spatially dependent index, and **r**' is the spatial coordinate. The grid sizes used to store data can be increased in order to conserve system memory. More details on this can be found in Section 5.B.4.

This output is saved in files such as `<prefix>_m#_t#_eden.pcs` with accompanying `.dat` data file where `<prefix>` is the **Output Prefix**, `m#` is the monitor number, and `t#` is the simulation time at which the data was saved in cT. The displayed data as well as the raw data file contain the exact energy data.

5.D.4. H Density

This type of output records the magnetic field density as a function of space at specific times. The expression used to evaluate this quantity is

$$U_H(\mathbf{r}',t_s) = \frac{1}{2}\mu\left|\mathbf{H}(\mathbf{r}',t_s)\right|^2$$

where **H** is the magnetic field, μ is assumed to be 1, and **r**' is the spatial coordinate. The grid sizes used to store data can be increased in order to conserve system memory. More details on this can be found in Section 5.B.4.

This output is saved in files such as `<prefix>_m#_t#_hden.pcs` with accompanying `.dat` data file where `<prefix>` is the **Output Prefix**, `m#` is the monitor number, and `t#` is the simulation time at which the data was saved in cT. The displayed data as well as the raw data file contain the exact energy data.

5.D.5. T Density

This type of output records the total field density as a function of space at specific times. This is equivalent to the sum of an **E Density** and **H Density** spatial monitor. The grid sizes used to store data can be increased in order to conserve system memory. More details on this can be found in Section 5.B.4.

This output is saved in files such as `<prefix>_m#_t#_tden.pcs` with accompanying `.dat` data file where `<prefix>` is the **Output Prefix**, `m#` is the monitor number, and `t#` is the simulation time at which the data was saved in cT. The displayed data as well as the raw data file contain the exact energy data.

5.D.6. Poynting Vector

This type of output records both the amplitude and individual components of the Poynting Vector as a function of space at specific times. The Poynting Vector is computed as:

$$\mathbf{S}(\mathbf{r}',t_s) = \mathbf{E}(\mathbf{r}',t_s) \times \mathbf{H}^*(\mathbf{r}',t_s)$$

where **E** and **H** are the electric and magnetic fields and **r**' is the spatial coordinate. The amplitude is computed as:

$$S(\mathbf{r}',t_s) = \sqrt{\mathrm{Re}[S_x]^2 + \mathrm{Re}[S_y]^2 + \mathrm{Re}[S_z]^2}$$

The grid sizes used to store data can be increased in order to conserve system memory. More details on this can be found in Section 5.B.4.

This output is saved in several files: the individual components are saved in files such as `<prefix>_m#_t#_sx.dat` (for x component) and a vector plot (`<prefix>_m#_t#_poy.pcs`) is saved. The amplitude of the Poynting Vector is saved in a file `<prefix>_m#_t#_poymag.pcs` with accompanying `.dat` data file. In both these cases, `<prefix>` is the **Output Prefix**, `m#` is the monitor number, and `t#` is the simulation time at which the data was saved in cT. The displayed data as well as the raw data file contain the exact power data.

5.D.7. Absorption

This type of output records the power absorbed as a function of space at specific times. The expression used to evaluate this quantity is

$$U_E(\mathbf{r}',t_s) = \omega \operatorname{Im}\left[\varepsilon(\mathbf{r}')\right]\left|\mathbf{E}(\mathbf{r}',t_s)\right|^2$$

where **E** is the electric field, ε is the spatially dependent index, and **r**' is the spatial coordinate. The grid sizes used to store data can be increased in order to conserve system memory. More details on this can be found in Section 5.B.4.

This output is saved in files such as `<prefix>_m#_t#_absorption.pcs` with accompanying `.dat` data file where `<prefix>` is the **Output Prefix**, `m#` is the monitor number, and `t#` is the simulation time at which the data was saved in cT. The displayed data as well as the raw data file contain the exact power data.

5.D.8 Spatial Output Format

The **Spatial Format** option sets the spatial output format.

5.D.9 Time-Averaging Spatial Output

To produce time-averaged spatial output, use the DFT options described in Section 5.F.

5.E. FFT Frequency Analysis

FullWAVE monitors measure certain electromagnetic parameters in the time domain; however, many useful measurements are based in the frequency domain. FullWAVE includes two types of frequency analyses in order to make these measurements: FFT and DFT. This section will describe FFT output; see Section 5.F for more information on DFT output.

FullWAVE's FFT frequency analysis can be expressed as a Fourier sum which is based on a discretized Fourier Transform evaluated over a finite domain:

$$\tilde{\phi}(\mathbf{r}',\omega) = \alpha \sum_{t=0}^{t_s} \phi(\mathbf{r}',t) e^{i\omega t} dt$$

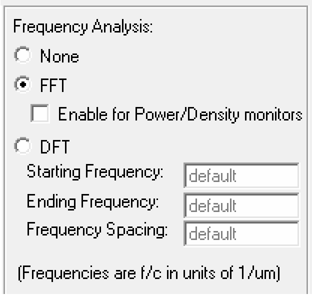
where $\phi(\mathbf{r}',T)$ is the data produced by the monitor, t_s is the simulation stop time, and w is the frequency. The normalization factor α is chosen such that the length of the time window will not change the value of the integral, and depends on the **Excitation** type chosen. The complete time response of any selected single-value output is recorded and the Fourier sum is computed when the simulation has completed. This type of frequency output is optimized for simulation speed.

DFT output, which is described in Section 5.F, solves the same Fourier sum. The difference is how the sum is computed.

5.E.1. Enabling FFT Output

The default behavior for FFT output depends on the excitation type: it is disabled for a CW excitation type and enabled for any other type as long as one or more monitors have *FFT* Frequency Analysis type selected (Fig 5-10) and one or both of the **Wavelength Monitor** or **Frequency Monitor** options in the Output Options dialog box are enabled (Section 3.E.1). Turning both of these options off will disable FFT output even if it is enabled for a monitor.

The time response of the FFT monitor will be displayed in the simulation window during a simulation and the wavelength/frequency results will be saved if an **Output Prefix** is set.



Frequency Analysis:

○ None

● FFT

□ Enable for Power/Density monitors

○ DFT

Starting Frequency: default

Ending Frequency: default

Frequency Spacing: default

(Frequencies are f/c in units of 1/um)

Figure 5-10: The portion of the monitor properties dialog box which controls frequency output.

An FFT analysis can only be used with single-value output. See Section 5.E.3 for details on using FFT output with memory intensive power/density monitor outputs.

5.E.2. FFT Output Files & Units

The files produced by FFT output and the units used depend on the type of output selected in the Options dialog and the **FFT/DFT Format** option:

- *'Frequency' FFT Output*

  When the **Frequency Output** option in the Output Options dialog is set, a plot of the FFT results vs. frequency [μm^{-1}] is produced. This output is saved in a data file named `<prefix>.fmn` with a corresponding WinPLOT command file `<prefix>.pfm`. An additional file `<prefix>.smn` is produced which contains the source spectrum vs. frequency. The format of this file can be set via `fdtd_source_spectrum_format` and contains amplitude data by default.

- *'Wavelength' FFT Output*

  When the **Wavelength Output** option in the Output Options dialog is set, a plot of the FFT results vs. wavelength [μm] is produced. This output is saved in a data file named `<prefix>.wmn` with a corresponding WinPLOT command file `<prefix>.pwm`.

In both cases it is possible to normalize the results to the source spectrum via the **Source Norm** option in the Output Options dialog. Also, the format of the data stored in the file is set by the **FFT/DFT Format** option. The data has the same units as the time data did before the Fourier transform and is similarly squared for display purposes. See Appendix B.B for information about the format of these files and for additional details.

5.E.3. Memory Intensive FFT Output

The FFT analysis of a power or energy monitor requires additional memory in order to obtain a physical result. To avoid unneeded memory usage, these types of output are disabled by default. To enable this type of analysis, check the **Enable for Power/Density monitors** box described above. While a FFT analysis of these monitor types is possible, it may be more economical to use a DFT analysis for the frequency range of interest instead of using the memory intensive FFT analysis.

These types of output require additional memory because the entire time response of the monitor is needed *before* the appropriate monitor integral shown in Section 5.C. The memory requirement also depends linearly on the size of the monitor domain as well as the **Monitor Time**.

The data format of memory intensive FFT output is not controlled by the **FFT/DFT Format** option, but rather by the variable `fdtd_powdens_spectrum_format` which defaults to `OUTPUT_REAL`.

5.E.4. FFT Usage Guidelines

Understanding the effect simulation parameters have on frequency results is critical when performing a Fourier transform analysis of time based systems. Both the sampling rate and the total number of points sampled have an important effect on discrete Fourier transforms. This section outlines these consequences and will provide important guidelines when choosing these parameters.

For purposes of this discussion, we will assume the following definitions: δt is the sampling rate in time which is set by the **Monitor Time** in FullWAVE simulation parameters dialog and not the time step, N is the total number of points sampled and is always an integer, and T_0 is the total length of time domain data and is equal to $N^*\delta t$.

- The frequency resolution is $\delta f = 1/T_0 = 1/(N\delta t)$ and the maximum frequency bandwidth is $\Delta f = 1/(2\delta t)$. Therefore, the greater T_0 the smaller the frequency resolution and the smaller the time sampling rate δt the greater the frequency bandwidth.
- Only frequencies that the excitation field excites can be recorded by a monitor.
- Data will be output with a frequency resolution given by δf defined above. To increase this resolution, the length of time data, T_0, should be increased.
- In cases when the field has diminished to zero after a certain period of time, there is no more information to be had by propagating further. The **Padding Factor** option, which is documented in Section 3.E, combats this difficulty by padding the monitor response with zeros before performing the FFT analysis to obtain higher resolution.
- FFTs are useful for frequency analysis of the entire frequency spectrum, and are usually less computationally intensive than a DFT for this task. However, some FFTs can be prohibitively memory intensive in which case a DFT is preferred.

5.F. DFT Frequency Analysis

FullWAVE monitors measure certain electromagnetic parameters in the time domain; however, many useful measurements are based in the frequency domain. FullWAVE includes two types of frequency analyses in order to make these measurements: FFT and DFT. This section will describe FFT output; see Section 5.E for more information on DFT output.

FullWAVE's DFT frequency analysis can be expressed as a Fourier sum which is based on a discretized Fourier Transform evaluated over a finite domain:

$$\tilde{\phi}(\mathbf{r}',\omega)=\alpha\sum_{t=0}^{t_s}\phi(\mathbf{r}',t)e^{i\omega t}dt$$

where $\phi(\mathbf{r}',T)$ is the data produced by the monitor, t_s is the simulation stop time, and w is the frequency. The normalization factor α is chosen such that the length of the time window will not change the value of the integral, and depends on the **Excitation** type chosen. The Fourier sum is calculated during the simulation at selected frequency values. Each additional term in the sum is calculated at each time indicated by **Monitor Time** for temporal output and **Slice Time** for spatial output. This type of frequency output is optimized to minimize memory usage.

FFT output, which is described in Section 5.E, solves the same Fourier sum. The difference is how the sum is computed.

5.F.1. Enabling DFT Output

To enable DFT output for a particular monitor, set **Frequency Analysis** type to *DFT* in the monitor properties dialog shown in Fig. 5-10.

The time response of the DFT monitor will not be displayed in the simulation window during a simulation and it is not possible to save the time-domain data. The output format for single-value output and spatial output can be selected via the **FFT/DFT Format** option and the **Spatial Format** option respectively. See note in Section 5.F.4 for additional control over the output units of DFT overlap output.

A DFT analysis can be performed with both single-value and spatial output.

The DFT analysis is performed over a selected range of frequencies. The default value for this range depends on the **Excitation** type chosen:

- *CW Launch*

  The default range corresponds to a single frequency equal to the **Free Space Wavelength** set in the Global Settings dialog with a frequency spacing of 0. This configuration is useful when performing steady-state analysis; see the next section for more details.

- *Other Launch Types*

The default range is defined to be centered at the value of the **Free Space Wavelength**, has a width equal to the width of the pulse width of the launch, and has a frequency spacing such that 32 points are used. For more information on the width of a launch pulse, please see Section 4.D.2.

The default frequency range can be changed via the **Starting Frequency**, **Ending Frequency**, and **Frequency Spacing** options shown in Fig. 5-10. All frequencies specified in these fields are given in units of μm^{-1}, or *f*/*c* where *f* is the frequency in Hertz and *c* is the speed of light in vacuum.

5.F.2. DFT Output Files & Units

The files produced by DFT output depend on the output format(s) chosen and the units used depend on the **FFT/DFT Format** option.

- *Single-Value Output*

  DFT results for selected single-value output types is saved in data files `<prefix>_m#.dmn` for each monitor, where `m#` is the monitor number, with a single plot file `<prefix>.pdm`. An additional file `<prefix>.smn` which contains the source spectrum vs. frequency is also produced. The format of this file can be set via `fdtd_source_spectrum_format` and contains amplitude data by default. By default the results are not normalized to the input spectrum; enable the **Source Norm** option for **Frequency Output** in the Output Options to force this normalization.

  The format stored in the file can be set via the **FFT/DFT Format** option. For single-value field output, the data stored in the file is in field units and the data displayed in the plot file is squared. For single-value power, E/H/T Density and Absorption output the data in the file and displayed is in power units. For overlap output, the units depend on the **FFT/DFT Format** setting. When the format is either *Default* or *Amplitude*, the data stored in the file is in field units and is squared; for other formats the data is stored in the file and displayed in power units. Set the variable `fdtd_overlap_unit_dft` to `0` always use 'field' units for overlap output, set to `1` to always use 'power' units.

- *Spatial Output*

  DFT results for selected spatial output types is saved in data files `<prefix>_m#_f#_type.dat` for each monitor and frequency value, where `_m#` is the monitor number, `_f#` is the frequency at which the DFT was taken, and `type` is the output type (`ex`, `hz`, `pow`, etc). Corresponding WinPLOT plot files are also produced for each output. The format of the output can be set via the **Spatial Format** option.

See Appendix B.B for information about the format of these files and for additional details.

5.F.3. Using DFT Output for Steady-State Analysis

The DFT of spatial output calculated at the driving frequency provides steady-state or time-averaged output. This type of output can be used as input for a steady-state simulator such as a BPM simulation or for the calculation of a far-field profile.

For a CW simulation, the DFT calculation is done during the last period of the simulation at the default frequency specified by the **Free Space Wavelength** field to avoid any transient effects. For pulse or impulse simulations, the DFT is calculated over the entire simulation time window.

This type of output is especially useful for FullWAVE users who own BeamPROP (or similar steady-state simulator) and wish to perform a hybrid simulation using both tools. DFT field output can be used as a FullWAVE output field along an arbitrary plane which can then be used as a launch field. This is very similar to the **Steady State Field (at Fixed Z)** option discussed in Section 3.E, but allows for more output options. See Section 5.G for an example of this type of setup. It is also possible to automatically take the far field projection of DFT steady-state results; see Section 6.B.

5.F.4. DFT Usage Guidelines

Understanding the effect simulation parameters have on frequency results is critical when performing a Fourier transform analysis of time based systems. Both the sampling rate and the total number of points sampled have an important effect on discrete Fourier transforms. This section outlines these consequences and will provide important guidelines when choosing these parameters.

- Only frequencies that the excitation field excites can be recorded by a monitor.
- Since the DFT outputs data at a user-specified frequency resolution, data can be computed at a resolution finer than the frequency spacing defined by typical Fourier sampling theory (see Section 5.F.4 for details). Frequencies that are between two points on the 'standard' frequency grid do not contain any additional spectrum information and can be thought of as representing an 'averaged' value of the two surrounding points on the 'standard' grid.
- DFTs are useful for frequency analysis at a small number of frequencies, or as a replacement for memory intensive FFT-based frequency analysis of power monitors. DFTs can, however, be more computationally expensive than FFTs in other situations.
- DFT overlap monitors can be output in either 'power' or 'field' units. 'Field' units allow for access to phase information. When the **FFT/DFT Format** is set to either *Default* or *Amplitude* the data is output in 'power' units; otherwise 'field' units are used. Set the variable `fdtd_overlap_unit_dft` to `0` to force 'field' units, set to `1` to forces 'power ' units.

5.G. Typical Monitor Configurations

This section contains sample monitor configurations for common applications. Most users find these typical configurations and arrangements to be the most useful for most FullWAVE simulations. Of course, the user is not limited to these configurations.

Specific Field Component

This type of monitor is the simplest to create. In order to monitor the major field component during a simulation, the user need only to select a temporal output of *Default Field*. If a specific field component is required, simply select the desired component for temporal output. The field is measured at the center of the monitor.

Measuring Backward Reflections

Many design problems call for an analysis of backward reflections. This may seem like impossibility at first, as the FDTD method can simulate reflections but cannot distinguish between a forward and backward traveling field.

The trick to singling out only the backward traveling field is to move the launch position away from its default value of the Z **Domain Min**, and then place a monitor behind the launch to measure the desired quantity. Since the launch is only forward traveling, this monitor will then record only the backward traveling reflected field as desired.

Power Flux through Specified Surface

In order to measure the power flux through a specific area, we need to calculate the component of the Poynting Vector along a normal to the surface. To do this, utilize a monitor type of *Power*. This type of monitor records the desired output. The surface is defined by the monitor position, width along X', and height along Y'. If the length along Z' is non-zero, the power flux will be measured through the monitor domain.

Power Spectrum

A power spectrum is nothing more than a frequency analysis of the time response of a system. There are two ways to calculate a power spectrum:

- *With a FFT*

  This method records the necessary time response of the system, and calculated the Fourier Sum described in Section 5.E at the end of the simulation. This results in a plot of the measured quantity versus frequency. Fourier sampling principles apply, and so the frequency resolution is inversely related to the length of the time window.

  This method is recommended for use when a quick overview of the frequency performance of a system is desired.

- *With a DFT*

  This method records the frequency output at selected frequencies, and calculates the Fourier Sum described in Section 5.F during the simulation. This results in values for power output at the selected frequencies. The frequency resolution of this method is not related to the length of the time window.

  This method is recommended for use when power output at specific frequencies is desired.

To use both of these methods, simply create a monitor with a Time Output Type of Power, and then selected the desired frequency analysis options.

Steady-State Field/Energy/Power Distribution

In order to analyze the steady-state properties of a structure, the structure needs to be driven at one wavelength/frequency. The DFT analysis options of a monitor can then be used to measure the steady-state properties, and also to create data files to be used in a BeamPROP/BPM simulation.

- In order to drive the structure at one wavelength/frequency, set the desired wavelength in the **Free Space Wavelength** field in the Global Settings dialog box, and choosing an **Excitation** type of *CW*.
- The domain over which to solve for the field/energy/power distribution must be specified.
- The type of spatial output must be specified. To measure a steady-state field/power distribution, choose the desired spatial output types. Selecting the desired **Spatial Format** indicates whether or not to include phase information in addition to amplitude information.
- Lastly, the desired frequency range must be chosen. Set the **Frequency Analysis** for each monitor to *DFT* and then leave the DFT parameters.
- Run the FDTD simulation with sufficient simulation time in order for the system to come to steady-state.

The monitor will automatically calculate the DFT of the time response at the driving frequency over one period at the end of the simulation.

Calculating Cavity Modes

In order to calculate the cavity mode of a resonant structure, two measurements are needed: first, the resonant frequencies of the cavity are needed; second, the field profile of the cavity mode must be found. Both of these tasks can be accomplished through the use of a FullWAVE monitor.

Although this procedure is based on the usage of FullWAVE monitors, a discussion of its use has been deferred to Chapter 8.

Custom Post-Processing

FullWAVE monitors provide quick ways to measure typical electromagnetic properties such as field values, power, and energy density. However, it is possible that a particular design project calls for a measurement which is not one of the built-in output types.

In this situation, it is still usually possible to measure the required parameter(s) since most electromagnetic properties are a function of index and field values. Monitors which measure all necessary field components can be created, and index values can be measured and saved via the **Display Material Profile** button. Once these values have been calculated and saved, the user can then calculate any desired parameter.

6

Advanced Features

FullWAVE has several advanced capabilities which extend the basic 2D and 3D FDTD algorithm, as well as several advanced simulation options to include material effects and decrease simulation time. These advanced features are described in this chapter.

6.A. Anisotropy, Non-Linearity, Dispersion, and Metals

FullWAVE can support materials with complex material properties such anisotropy, non-linearity, dispersion. Materials with these properties can be defined via the Material Editor. See the CAD manual for more details. Below are some FullWAVE-specific notes on these material properties.

Anisotropy

Anisotropy is defined in FullWAVE via a full 3x3 complex valued index tensor. Each segment can have a different index tensor, allowing for the possibility of customized design. To enable the use of anisotropic materials in a simulation, check **Use Anisotropic Materials** in the Advanced Parameters dialog box. Anisotropic materials can be defined via the Material Editor as discussed in the CAD manual.

Non-Linearity and Dispersion

FullWAVE supports a wide variety of material systems, including the linear and non-linear relative permittivity (epsilon) and permeability (mu) as a function of frequency. To utilize dispersive or non-linear materials with FullWAVE, it is necessary to enable the **Dispersion/Nonlinearity** option in the FullWAVE Simulation Parameters dialog. Non-linear and dispersive materials can be defined via the Material Editor as discussed in the CAD manual.

When using dispersive materials in simulations where multiple wavelengths are present (pulse excitations, multiple CW excitations at different wavelengths, etc), the multiple-Lorentzian material model must be used; see Tutorial 5 for more details.

The use of this option is required when performing simulations with lossy materials, even when a single CW excitation is used: though one wavelength is excited, other wavelengths are present in small amounts due to the finite ramp time of the CW launch. These wavelengths encounter random noise and gain and suffer from causality concerns which are not an issue when only dielectric materials are present.

In some cases, instabilities can occur when using metals. Two ways to combat this is to reduce the time step as the Courant condition can vary slightly for dispersive materials, or to use the **PML Kappa** feature described in Section 6.C.1.

6.B. Computing Far Fields

FullWAVE supports a wide range of features to compute far field profiles. There are several types of far fields possible: far fields of a single cross-section of field in one direction, and the far field from an area or volume of data in all directions.

6.B.1. Single-Plane Calculation

This type of far-field is computed from the near-field at single cross-section and is appropriate for a confined field (i.e. goes to 0 near the edges) such as a waveguide or fiber mode, or a Gaussian.

Before a far-field transformation can be taken, it is necessary to first calculate the desired steady-state near-field. FullWAVE provides two ways to compute the near-field.

Single-Plane Far-Fields from a Fixed Z Plane

To compute a steady-state near-field at a fixed Z plane, use the **Steady State Field (at Fixed Z)** output option in the Output Options dialog. To compute the far-field of this output, enable the desired output type in the **Far Field** option in the same dialog. All field components required will be used for the far-field calculation. For pulse/impulse simulations the far-field will be calculated for each frequency a near-field is produced. See Section 3.E.2 for information on the **Steady State Field (at Fixed Z)** option.

When the **Steady State Field (Fixed Z)** output is used, the data file produced is `<prefix>_bpm.far` which contains the requested far-field format as a function of angle(s). These angles have the same definition as the launch angles described in Section 4.C.4. Two WinPLOT command files are also produced `<prefix>_bpm.pfr` and `<prefix>_bpm_polar.pfr` where `<prefix>` is the **Output Prefix** used. For the *Amplitude/Phase* far-field output type, WinPLOT command files are also output to display the phase of the far-field. When multiple far-fields at different frequencies are calculated, the frequency 'index' will be noted in the filename (`_f#`).

Single-Plane Far-Fields from X, Y, or Z Planes

Far-field projections can be made from near-fields at X, Y, or Z planes using the DFT of the spatial field output of a monitor and by enabling the **Far Field for DFT Monitors** option in the FDTD Options dialog.

To compute a steady-state near-field at any X, Y, or Z plane, position a monitor along the plane of interest. The monitor must have a **Length** of `0` and rotated using **Phi** and **Theta** so that it lies along a single X, Y, or Z plane (ϕ and θ must be `0`, `90`, `180`, or `270`). The monitor **Width** and **Height** must be large enough to cover the near-field region; covering the entire simulation domain is usually safest.

The required spatial output field components of the monitor must be enabled and the DFT frequency analysis option must be chosen to get steady-state field output (see Section 5.F.3).

All required field components must be enabled in the monitor spatial output for far-field output. At a minimum, in 2D, this means the transverse E (TE) or H (TM) component must be output, in 3D the 2 E field components transverse to the monitor must be output. If these requirements are not met, no far-field will be output. If it is unclear which components are needed, enable all of them.

When a DFT monitor is used, the data file produced will have a name `<prefix>_m#_f#.far` which contains the requested far-field format as a function of angle(s). These angles have the same definition as the launch angles described in Section 4.C.4 with the exception that they are within the local monitor coordinate system (i.e. ϕ = θ =0 is the direction normal to the monitor). Two WinPLOT command files are also produced `<prefix>_bpm.pfr` and `<prefix>_bpm_polar.pfr`. In these filenames `<prefix>` is the **Output Prefix** used, `m#` is the monitor number, and `f#` denotes the frequency 'index' of the far-field. For the *Amplitude/Phase* far field output type, WinPLOT command files are also output to display the phase of the far-field.

Single-Plane Far-Field Options

There are several additional options which control the far-field calculation:

- The angular resolution of the output can be controlled by setting the `farfield_da` to the required resolution in degrees. The resolution is set by the domain of the calculation and this feature extends the domain during the far-field calculation to achieve the requested resolution.
- The far-field is calculated in a homogenous air region by default. To calculate the far-field in a different medium, set the `farfield_index` variable to the desired refractive index.
- The variables `farfield_axmin`, `farfield_axmax`, `farfield_aymin`, and `farfield_aymax` can be used to set the angular range used in far-field plots.
- When the near field is periodic (for example when periodic boundaries are used, set the variable `farfield_periodic` to `1` to produce the far-field as a list of diffraction efficiencies vs. diffraction orders. Further control in 2D is possible of via `farfield_per_de` which, when set to `1`, outputs the diffraction efficiency vs. angle, and the variable `farfield_per_zeroangle` (defaults = `0`) and sets the angle of incidence. These options are useful when comparing FullWAVE results to DiffractMOD results.

- The variable farfield_normalization controls the normalization of the output. A value of 0 (the default) leaves the field unnormalized and a value of 1 normalizes the peak field value to 1.
- Far fields of this type can be computed from any previously generated near-field data files using the command line utility bdutil described in the appendices of the RSoft CAD manual though all options may not be available.

6.B.2. Multi-Plane Calculation

This type of far-field is computed from the near-field on all faces of the simulation domain and is appropriate for the field from a radiating dipole or the field from a plane wave scattered by an object. The results are only valid when the simulation domain lies in a homogeneous region with a single index value.

This type of far-field can be enabled using the **Multi-Plane Far Field** option in the Output Options dialog. This creates hidden 'spatial DFT' monitors at the edge of the simulation domain and computes the far-field from the near field data on this surface.

The data file produced will have a name <prefix>_farfield_f#.dat which contains the far-field as a function of angle(s). These angles have the same definition as the launch angles described in Section 4.C.4. Several WinPLOT command files are also produced including <prefix>_farfield_f#.pcs and <prefix>_polar_f#.pcs. For 3D simulations, two polar plots are produced for the upper and lower hemispheres. In these filenames <prefix> is the **Output Prefix** used and f# is the frequency index.

Multi-Plane Far-Field Options

Several options can be used to customize this output:

- The hidden monitor can be offset in from the domain edges using fdtd_farfield_offset which defaults to 0. The grid on which the hidden monitor stores the near field can be set using fdtd_farfield_dx, fdfd_farfield_dy, and fdtd_farfield_dz. These grids equal the simulation grids by default.
- The required DFT frequencies used by the hidden monitor to compute the near field can be set by fdtd_farfield_fmin, fdtd_farfield_fmax, and fdtd_farfield_df. The default vales of these parameters is based on the excitation type (CW or Pulse), and are exactly the same as those described in Section 5.F.1. In most cases, the defaults are sufficient and do not need to be changed.
- The far-field is calculated in a homogenous air region by default. To calculate the far-field in a different medium, set the fdtd_farfield_index variable to the desired refractive index.
- The angular resolution can be set by fdtd_farfield_da which controls both ϕ and θ (3D only). Individual control is possible by using the fdtd_farfield_dphi and fdtd_farfield_dtheta variables. The angles ϕ and θ are defined on the ranges [-180,180) and [-90, 90) respectively.

- The normalization of the output can be set by the variable `fdtd_farfield_normalization`. A value of `1` means unnormalized and a value of `2` means normalized so that the power contained in the field is unity.

- It is possible to also output the far-field in Cartesian coordinates at a fixed Z value. This is similar to the single-plane calculation described in Section 6.B.1 except the near-field is defined on an enclosed box instead of a fixed plane. To enable this option, set `fdtd_farfield_distance` to the desired distance [µm] of the Z plane measured from the center of the enclosed near-field box. The X and Y domain size of the output plot can be controlled by `fdtd_farfield_domain`. Individual control is possible with the variables `fdtd_farfield_xmin`, `fdtd_farfield_xmax`, `fdtd_farfield_ymin`, and `fdtd_farfield_ymax`. The grid size of the output plot can be controlled by `fdtd_farfield_ffdx` and `fdfd_farfield_ffdy`.

- By default, the far-field is output as intensity. It is possible to also output the **E** and **H** field values using the variable `fdtd_farfield_outeh`. Setting this variable to a value of `1` will output field data in the usual spherical (radial in 2D) coordinates and produces plots of the **E** and **H** fields with the radial (*r*) dependence removed. The radial component of the **E** and **H** fields is always 0 and is not output. The *r* dependence (a factor of e^{ikr}/r in the output) can be added by setting the variable `fdtd_farfield_distance` to the desired distance [µm]. Setting `fdtd_farfield_outeh` to `2` will output data in Cartesian coordinates at a fixed Z value as described above and, for this case, the variable `fdtd_farfield_distance` again sets the desired distance [µm] of the Z plane. The same domain and grid controls described above also still apply.

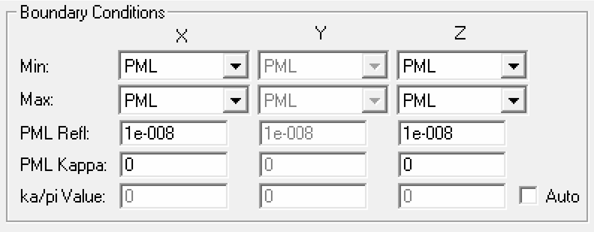
- By default, all six (four in 2D) planes are used to compute the far-field. It is possible to include a subset of these planes using the variable `fdtd_farfield_use_planes`. Using the keywords `left`, `right`, `bottom`, `top`, `front`, and `back` (Xmin, Xmax, Ymin, Ymax, Zmin, and Zmax respectively) particular planes can be included in the calculation. Multiple planes can be included as well. For example, setting `fdtd_farfield_use_planes` to a value of '`left,top,right`' (without quotes) will include only the Xmin, Xmax, and Zmin planes.

- Information about the total power contained in the computed far-field is produced in a file `<prefix>.tcsc` with associated plot file `<prefix>.ptscs`. Data is produced for each frequency used by the hidden DFT monitors described above.

- Far fields of this type can be computed from saved simulation data (in the binary state file form described in Section 6.D) using the command line utility `fwmodekit` described in the appendices of the RSoft CAD manual though all options may not be available.

6.C. Boundary Conditions

FullWAVE needs to define how the fields behave at the boundaries of the selected domain. FullWAVE uses *PML* (Perfectly Matched Layer) boundary conditions by default, but can also use Periodic, Symmetric, Anti-Symmetric, and PEC boundary conditions. These boundary conditions can be applied to each of the three coordinates separately, and in some cases, to each of the six boundaries independently. The boundary type can be set in the FDTD Advanced Parameters dialog which is accessed via the **Advanced...** button in the FullWAVE simulation parameters dialog.



Boundary Conditions

	X	Y	Z	
Min:	PML	PML	PML	
Max:	PML	PML	PML	
PML Refl:	1e-008	1e-008	1e-008	
PML Kappa:	0	0	0	
ka/pi Value:	0	0	0	□ Auto

Figure 6-1: The portion of the FDTD Advanced Parameters dialog where the boundary conditions are set.

The following two options allow the user to select the type of boundary conditions used. The rest of the options are described in the next boundary condition-specific sections.

Min.

These fields control the boundary types of the negative side on the respective coordinates. Its choices are *PML*, *Periodic*, *Symmetric*, *Anti-Symmetric*, and *PEC*. Note that when set to *Periodic*, the **Max** boundary is automatically set to this value as well. These fields default to *PML*.

Max.

These fields control the boundary types of the positive side on the respective coordinates. Its choices are *PML*, *Periodic*, *Symmetric*, *Anti-Symmetric*, and *PEC*. Note that when set to *Periodic*, the **Min** boundary is automatically set to this value as well. These fields default to *PML*.

6.C.1. PML Boundaries

By default, FullWAVE uses PML boundary conditions. A PML boundary, which consists of several points which are added to the edge of the domain, is designed to act as a highly lossy material which absorbs all incident energy without producing reflections. This allows field energy which is incident on the boundary to effectively leave the domain without reflection. PML boundary conditions can be applied to any combination of the three coordinates as well as each of the six boundaries individually.

The options that control the PML boundaries are:

PML Width

This is the basic control of the PML, and is located in the *FullWAVE* Simulation Parameters dialog discussed in Section 3.B. It sets the width of the PML (Perfectly Matched Layer) used as a boundary condition. This width is then added to the edge of the computational domain.

It is possible to create a perfect conductor boundary condition by setting the **PML Width** to `0`, though careful attention should be paid to the polarization.

PML Refl.

This field sets the maximum desired reflectivity of the PML region. This option is set in the FDTD Advanced Parameters dialog.

PML Kappa

This field sets the kappa value to be used for the PML. It controls how the PML absorbs evanescent waves and can be useful in controlling unwanted reflections that occur from the PML. This option is only recommended to be used for cases where the problem being solved is going unstable in the PML region, and increasing the width of the PML doesn't help. This option is set in the FDTD Advanced Parameters dialog.

Other PML Options:

Further control of the PML is provided via the variable `fdtd_pml_index`. This variable controls how the refractive index is calculated within the PML region. Its possible values are:

Value	Description
`-1`	The structure at the edge of the simulation is artificially extended into the PML. This is the default value for this parameter.
`0`	The PML region consists of the structure defined in the CAD for the PML region.
`value`	The index in the PML region is set to the value given.

6.C.2. Periodic Boundaries

Periodic boundary conditions are quite useful when working with periodic structure types. A periodic boundary stipulates that any field which leaves the boundary on one side of the domain should reenter the domain on the opposite side. This can be expressed mathematically as

$$\mathbf{E}(x_i) = \mathbf{E}(x_i + \Delta)e^{ik_i x_i}$$

where the structure is assumed to be periodic along the coordinate x_i with period Δ and a phase difference k_i, defined by the **ka/pi Value**. The period is defined by the length of the domain along the specified coordinate x_i. Periodic boundaries can only be applied to any combination of the three coordinates, but cannot be applied to each boundary individually.

The option that controls the PML boundaries are:

ka/pi Value

These fields set the phase difference between the boundaries for periodic boundary conditions. Its units are such that a value of `1` corresponds to a pi phase shift. The use of this field requires that the boundary type for the desired coordinate be set to *Periodic*. The default value of this field is `0`. To automatic set these options when launching a tilted plane wave and using periodic boundary conditions, enable the **Auto** option. For more information about this option, see Section 4.H.1.

6.C.3. Symmetric & Anti-Symmetric Boundaries

Symmetric and anti-symmetric boundary conditions assume that one or more components of the field is symmetric or anti-symmetric about one or more planes of reflection, and are also known as "even" and "odd" boundary conditions respectively. These boundary types are useful when both the structure and simulated fields have symmetry about one or more planes of reflection.

Symmetric and anti-symmetric boundaries can be expressed mathematically as $u(x_i) = u(-x_i)$ and $u(x_i) = -u(x_i)$ where x_i is the specified coordinate about which the field is to be assumed symmetric. The position of the reflection plane is given by the domain definition. Note that for anti-symmetric boundaries, the field is set to 0 at the boundary. These boundary conditions types can be applied to any combination of the three coordinates as well as each of the six boundaries individually.

The choice between symmetric and anti-symmetric boundaries is not intuitive: consult the notes in the next section ot determine what condition to choose.

Symmetry Relations

The conditions implied by the *Symmetric* and *Anti-Symmetric* boundary conditions are quite complex. As discussed in Section 2.A, the FDTD algorithm is implemented using a Yee mesh and the **E** field is computed at points shifted one-half the grid spacing from the **H** field. Therefore all field components are not calculated at the domain boundaries and cannot be set directly. However the curl terms in Maxwell's equations cause the parities of the fields to be interrelated.

Consequently, it is not true for example, that with a symmetric boundary condition all the fields are even functions about the symmetry axis. The actual symmetry properties implied by the boundary conditions are as follows.

3D simulations:

For 3D TE simulations, all field components are nonzero and are either even or odd functions at each boundary given any selected symmetric/anti-symmetric boundary types:

	X Sym.	X A-Sym.	Y Sym.	Y A-Sym.	Z Sym.	Z A-Sym.
Ex	even	odd	even	odd	odd	even
Ey	odd	even	odd	even	odd	even
Ez	odd	even	even	odd	even	odd
Hx	odd	even	odd	even	even	odd
Hy	even	odd	even	odd	even	odd
Hz	even	odd	odd	even	odd	even

So, for example, if an X boundary is set to *Symmetric*, then E_x, H_y, and H_z will be even functions (symmetric) at the boundary and E_y, E_z, and H_x will be odd functions (anti-symmetric) at the boundary. Thus $E_x(x,y,z) = E_x(-x,y,z)$ but $E_y(x,y,z) = -E_y(-x,y,z)$.

This table is for 3D TE simulations only. 3D TM simulations have the opposite values.

In practice, we choose the combination of symmetries that provide the desired parities for a particular field. Suppose for instance, that we seek to find a certain x-polarized dipole mode in a 3D photonic crystal slab defect cavity. In preliminary work, either using BandSOLVE to find the defect mode or by using FullWAVE with low resolution but no symmetry conditions, we establish that for this mode, the major component E_x is an even function of all 3 coordinates. Then the above table indicates that we could find this mode 8 times faster by using the 1st octant (X≥0, Y≥0, Z≥0) with symmetric boundary conditions along *X* and *Y*, but anti-symmetric conditions along *Z*.

These rules can be quite counter-intuitive and it is best to use this table carefully rather than guess. For instance, in the x-dipole case just described, it would be very tempting to use symmetric boundary conditions along all axes.

2D Simulations:

In 2D, there is no Y-axis and certain components of the field are null, so the relations are somewhat different. For 2D TE simulations, the following relations apply:

	X Sym.	X A-Sym.	Z Sym.	Z A-Sym.
Ey	even	odd	even	odd
Hx	even	odd	odd	even
Hz	odd	even	even	odd

For 2D TM simulations, the following relations apply:

	X Sym.	X A-Sym.	Z Sym.	Z A-Sym.
Ex	even	odd	odd	even
Ez	odd	even	even	odd
Hy	even	odd	even	odd

6.C.4. PEC Boundaries

A perfect electric conductor (PEC) boundary sets the tangential component of **E** at the boundary to zero. It is the equivalent of choosing either Symmetric or Anti-symmetric boundary conditions taking the polarization and boundary coordinate (X, Y, or Z) into account as described in the tables in the above section.

6.D. Saving a State File & Continuing a Completed Simulation

FullWAVE can save a state file (`*.fst`) at the end of a simulation which contains all the final field data. It can be used to restart the simulation and/or for further post-processing, and has a binary format which may change in future releases.

6.D.1. Saving a State File & Continuing a Simulation

To save a state file, the **Continue Simulation** option in the FDTD Advanced Parameters dialog must be set before the simulation is started. This option, depending on its setting, will automatically save the state of the simulation to a state file, load that state file when restarting a completed simulation, and append monitor data to existing monitor data. This option is controlled by the variable `fdtd_continue`.

When performing a cavity mode computation, this option does not save the final field state, but rather saves the computed mode profile. This is useful for refining the mode profile or custom post-processing. Also, state files can be saved after a simulation has completed via the File menu item in the FullWAVE simulation window.

This option has three settings. They are:

- *No*

  This choice disables this feature and is equivalent to `fdtd_continue` = `0`.

- *Yes*

  This choice, along with the definition of an **Output Prefix**, enables this feature. On the first execution, when a state file does not exist, the initial field is determined based on the normal excitation/launch settings as usual, and a new monitor file is created. At the end of the simulation, a state file is produced. On subsequent executions, the simulation will load the existing state file, restart the simulation, append additional data to the same monitor file, and produce another state file. This is equivalent to `fdtd_continue` = `1`.

  To reset the calculation the user must either choose a new **Output Prefix**, delete the state file associated with the current prefix (`<prefix>.fst`), or select *Restart* to force initialization.

- *Restart*

  This choice forces initialization. After a simulation is run with this option, the user must set it to either *No* or *Yes* to stop or continue using this feature respectively. This is equivalent to setting `fdtd_continue` to `-1`.

6.D.2. Using a State File for Further Post-Processing

Once a state file has been saved as described above, it can be useful to extract field data from the state file. This is especially important when computing 3D cavity modes or when computing far-field patterns as described in Section 6.B. The standard FullWAVE distribution includes `fwmodekit`, a command line utility that performs these types of calculations. A description of the `fwmodekit` utility can be found in the Utilities appendix of the CAD manual.

6.D.3. Manually Working With State Files

The **Continue Simulation** option described above automatically saves a state file, restarts a simulation, appends any monitor data, and saves another state file. In some cases it is not necessary to perform all of these steps such as when a user wants to save a state file for post-processing without restarting the simulation, or when a user wants to launch the same state file without it being overwritten every time the simulation completes, such as in a parameter scan. The following steps will reproduce the use of `fdtd_continue`, but can be performed individually as needed:

- Set `fdtd_state_output` to `1` to save the simulation state at the end of the simulation to the file `<prefix>.fst`.
- Set `fdtd_excitation_file` to the name of a state file to load the state file when the next simulation is started. The field pattern and other variables will be set to allow a simulation to be continued in time exactly where it was left off. This assumes that certain other parameters, such as the grid, have not changed.
- Set `fdtd_append_tmonitor` to `1` to append monitor data to the same `.tmn` file (when using same **Output Prefix**).

6.D.4. Viewing a State File

The contents of a previously computed `.fst` state file can be opened for viewing purposes via the **View FST** button in the Display Options dialog. The state file will open in a FullWAVE simulation window where the typical controls for selecting a cross-section as well as field component can be used to examine the field.

6.E. Additional Simulation Algorithms

FullWAVE offers additional simulation algorithms.

6.E.1. Single vs. Double Precision

FullWAVE uses single precision for all numerical bookkeeping during a simulation. This means that all field, index, and simulation coefficient values are stored as a single precision number. The user can change this default to double precision either locally in each design file or globally for all design files on a particular computer. To change this setting locally, use the **Single Precision** setting in the FDTD Advanced Parameters dialog. To change this setting globally, use the Use Single Precision setting in the CAD Preferences dialog which can be opened via the Options\Preferences menu item in the CAD. Single precision is the best choice as it requires roughly half the system memory when compared to a double precision simulation of the same problem size.

6.E.2. Radial FDTD

FullWAVE includes an option for Radial FDTD, which allows certain 3D problems with azimuthal symmetry to be simulated as 2D problems, such as fiber structures with circular symmetry. This creates a simulation with an efficiency close to a 2D problem since the azimuthal dependence is treated analytically and only the radial (R) and axial (Z) coordinates are treated numerically. To enable the Radial FDTD feature, either check **Radial Calculation** in the Global Settings dialog, or define the variable `radial` and set it to `1` in the symbol table.

Using Radial FDTD

To use this option, choose a 2D waveguide model and consider X=0 as the center of symmetry when defining the structure. When this scheme is selected, the transverse coordinate x is then interpreted as the radial coordinate R. Note that while the structure can be defined for X<0 and X >0, and this is in general the optimal things to do , the problem will only be solved in the half space X>0. The user need not select this – the program will automatically enforce it.

Setting the Azimuthal Behavior

While it is implied that the structure is azimuthally symmetric, the field can have other implied azimuthal behavior. The option **Azimuthal Mode #**, which appears in the FullWAVE Simulation Parameters dialog when performing a radial computation, controls the azimuthal dependence. This mode number corresponds to the *m* in the HE_{mn} type designation, which differs by 1 from the LP_{mn} designation. For example, the true fundamental mode of a fiber is the HE_{11} mode, which is approximated by the LP_{01} mode.

This setting corresponds to the azimuthal behavior of the longitudinal field components, which is typical for full-vector treatments of structures with circular symmetry. The azimuthal behavior of the transverse components differs by 1. Thus, *m*=1 corresponds to the fundamental, and is the default value for **Azimuthal Mode #**, and produces a Gaussian-like intensity pattern for the power flow. The *m*=0 case corresponds to the pure TE and TM waves (which do not always exist) and produces a doughnut-shaped intensity pattern.

The use of dispersive and/or non-linear materials as well as several monitor features are not currently enabled while using Radial FDTD. Also, note that when using a **Launch Type** of *Fiber Mode* for a radial calculation indicates that a LP mode should be used. It is not currently possible to launch a genuine TE or TM mode.

Displaying 3D Results

The azimuthal pattern is normally not displayed unless a result from a radial FDTD calculation is launched into a 3D FDTD calculation. However, it possible to output the actual 2D field profiles using the **Output Radial Data in XY Format** in the Output Options dialog. This data will be output in files with "`_3d`" appended to their names. All six field components (E_x, E_y, E_z, H_x, H_y, H_z) can be generated.

This option is for visualization purposes only; radial field data can be launched into a 3D calculation automatically.

6.E.3. Complex FDTD

Certain simulation configurations require the use of complex-valued fields, such as when applying a phase different across periodic boundary conditions. FullWAVE contains a complex formulation of FDTD which can be enabled by enabling the **Use Complex Implementation** in the FDTD Advanced Parameters dialog.

While complex-valued FDTD has been tested for use in configurations that require it, it has not been fully tested with all possible FullWAVE features. Also, since it uses twice as much memory it is not recommended for use unless required.

7

Clustering

The FDTD algorithm can easily be adapted for cluster computing. A clustered FullWAVE simulation employs multiple CPUs, which can be on a single computer, distributed over a network, or both utilizing the message passing interface (MPI). FullWAVE simulation can therefore access additional system resources compared to running on a single machine, resulting in not only a faster simulation, but also the ability to work with larger problems.

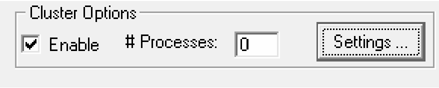
This chapter gives guidelines for use of the clustering feature. It assumes that the user is familiar with the use of FullWAVE on a single machine. Please contact RSoft if you have any additional questions.

7.A. Licensing, Installation, & Testing

FullWAVE's clustering feature is included with every copy of FullWAVE and allows clustering using multiple cores or CPUs on the same machine. Multiple FullWAVE licenses can be clustered across multiple machines on a network. The installation procedure for a FullWAVE cluster is covered in the RSoft Installation Guide. If you have not already followed these instructions, please do so. The rest of this chapter will assume that the user has successfully installed and tested the standalone portion of the FullWAVE program and also performed any specific steps required to install the included MPI implantation. FullWAVE clustering supports both LAN-based networks and Myrinet networks. For more information about Myrinet support, please contact RSoft.

7.B. Cluster Settings

The lower left portion of the FullWAVE Simulation Parameters dialog contains the settings for clustered simulation.


Cluster Options

☑ Enable

Processes: 0

Figure 7-1: The lower left portion of the FullWAVE Simulation Parameters dialog where cluster settings are made.

These settings are:

Enable

This option enables the use of clustering in a simulation. If this option is unchecked, a standard standalone simulation will be performed regardless of any of the other cluster settings. This is a local setting that can be set differently in each design file.

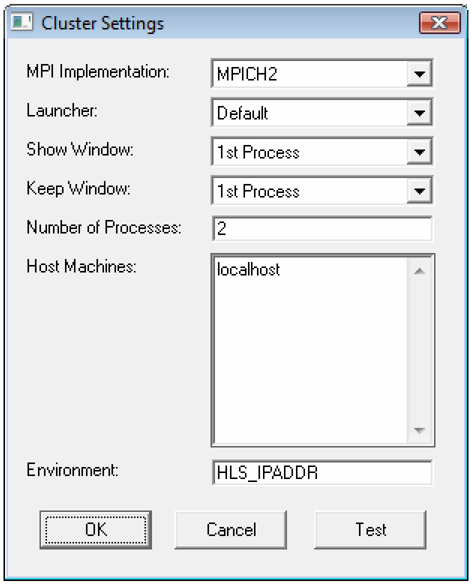
If you are not licensed for a clustered simulation, this option will be unavailable to you.

# Processes

This field determines how many processes will be involved in the clustered simulation. This is a global setting that is the same for all design files on a particular computer. Changing this setting in one file will change all other files. This field controls the same parameter as the field in the Cluster Settings dialog. The number `0` indicates that clustering will not be used.

The Cluster Settings Dialog

The Cluster Settings dialog allows the user to control several aspects of a clustered simulation. It can be opened via the **Settings…** button in the FullWAVE simulation properties dialog or via the menu item Run/Cluster-Settings….


Cluster Settings

MPI Implementation: MPICH2

Launcher: Default

Show Window: 1st Process

Keep Window: 1st Process

Number of Processes: 2

Host Machines: localhost

Environment: HLS_IPADDR

Figure 7-2: The Cluster Settings dialog box.

The fields shown, including the **Test...** button, in the dialog in Fig. 7-2 are:

MPI Implementation

This field must be set to the version of MPI that you have installed. Users are encouraged to use the MPICH implementation that is included in the FullWAVE distribution.

Launcher

This field determines the program that launches the simulations on each machine. By default, the launcher that is native to the MPI Implementation is used.

Show Window

This field determines which simulation is shown on the master node.

Keep Window

This field determines whether or not the window remains active or is closed when the simulation is completed.

Number of Processes

This field determines how many processes will be involved in the clustered simulation. This is a global setting that is the same for all design files on a particular computer. Changing this setting in one file changes all other files. This field controls the same parameter as the field in the FullWAVE Simulation Parameters dialog. The number `0` indicates that clustering will not be used.

Host Machines

This field lists the names of the machines that will be involved in the clustered simulation. Each of these machines must have the MPI and FullWAVE software installed on them as noted earlier. The default list contains the name `localhost`, which refers to the local machine, whatever its name.

If the number of processes is greater than the number of machines, the list is automatically repeated. It is also possible to assign a particular number of processes to a specific machine with the following syntax: `computer_name:#` where `computer_name` is the host name and # is the desired number of processes.

For large clusters it can be beneficial to group several nodes together when defining a host list. To do this, create a text file that contains one node name on each line. To load this file, use the syntax `@node_group_file` in the **Host Machines** list.

Environment

This field allows the user to enter system variables and their values that will be passed to cluster nodes during a clustered simulation. The syntax is `NAME=VALUE, NAME=VALUE,`....

Test...

This button will launch a series of tests to determine if the cluster has been set up correctly. It is useful for debugging purposes to ensure that the computers in the cluster are connected and can access each other.

7.C. Cluster Display Settings

Cluster display settings can be found in the Display Options dialog. See Section 3.F. These options are:

Display Whole Domain on Master Node

This option will enable the display of the entire domain during a FullWAVE simulation using clustering. Note that this option uses additional network resources, and its use will result in a decrease in simulation time. To reduce this effect, the following two options can be used.

Display Grid Factor

This option enables a courser grid when using the **Display Whole Domain on Master Node** option is enabled in order to speed up the simulation. For example, the default value of this parameter, 2, indicates that every other point should be displayed.

Display Height

This option sets the height along the Y coordinate for data to be transmitted to the master node when the **Display Whole Domain on Master Node** option is used. The default value of this field is 0, which corresponds to a single plane of data in the XZ plane.

7.D. Additional Cluster Settings

This section documents additional cluster settings that are controlled in the FDTD Advanced Parameters dialog which can be opened by clicking **Advanced…** in the FullWAVE Simulation Parameters dialog.

Use Master Node

It can be advantageous to use the master node solely as a control node and not for simulation purposes. To do this, uncheck this option.

Domain Cut Direction

By default, FullWAVE will divide the simulation among the cluster nodes along the X axis. However, it can be advantageous to divide the problem along the Z axis. This is especially for structures where load balancing between cluster nodes is more uniform using a Z-cut rather than the default X-cut, such as structures that are primarily oriented along Z or for structures with complicated Z dependencies.

7.E. Running Cluster Simulations from the Command Line

The procedure for running a non-clustered FullWAVE simulation from the command line interface is covered in Chapter 10 of the CAD manual.

Clustered simulations require additional information such as a number of processes and a list of host machines to be specified when running from the command line interface. This information is provided to FullWAVE via the program `fwmpirun` with the syntax:

```
fwmpirun [mpiopts] index-file-name [name=value  name=value  ..]
```

where the available MPI options are:

Option	Description
`-np`	number of processes
`-machine file file`	machine file
`-hosts name,name,…`	list of hosts (alternative to above)
`-show #`	nodes to show (0=none,1=1st,2=all)
`-hide #`	nodes to hide (0=none,1=1st,2=all)

For example, to run a cluster simulation of a design file `wg.ind` with two processes on two nodes with names computerA and computerB the command would be:

```
fwmpirun -np2 -hosts computerA,computerB wg.ind
```

For further examples, run a clustered simulation using the graphical interface and check the log files created for the commands used.

8

Cavity Modes & Q Factors

FullWAVE has several features that enable the calculation of cavity modes and *Q* factors for certain types of structures.

In this chapter, we describe the manual procedure for cavity mode computation, and explain several methods for finding *Q* factors. We then explore the basic and advanced uses of Q-Finder, an automated tool for calculating the center frequency, quality factor *Q*, and spatial profiles of resonant modes. If you must, you can skip over the background and manual procedure sections and jump straight to the automated procedures in Section 8.C, but reading through these preliminary sections will provide a much deeper understanding of Q-Finder.

Note that due to the nature of the technique, it may be difficult or impossible to obtain accurate modes and *Q*-factors for some very low or very high *Q* problems. Also, since cavity mode and *Q* calculations can be computationally expensive, it would be good to read through the hints on improving simulation speed in the Tips and Tricks section of Appendix A.

8.A. Theoretical Background

The study of leaky modes and cavity resonances is one of the most important applications of time-domain analysis in present-day electromagnetics. In many structures such as photonic crystal cavities or micro-ring resonators, light cannot be trapped indefinitely. Instead, it leaks away, either because the confinement is imperfect or through unavoidable scattering losses. In such systems, the states in which energy is stored in the cavity are not true modes of the system, but finite-lifetime resonances. In the event that a given spectral resonance is reasonably isolated from any others, the stored energy decays exponentially in time and exhibits a Lorentzian line profile.

The mode profile itself can be found via a Fourier analysis during a FullWAVE simulation. The rate at which energy is lost is frequently characterized by the *quality factor* or simply *Q*, which can be defined and calculated in a number of ways.

8.A.1. Finding Cavity Resonances & Mode Profiles

The basic procedure for finding a cavity resonance is as follows:

- Launch an arbitrary field into the cavity at *t*=0 to excite a superposition of resonant states and radiation states.
- Let the field decay for some time without further driving, and measure the time response at a random low-symmetry point.
- Perform a frequency analysis operation, such as a Fourier transform, on this time-domain data to obtain the frequency response of the system.
- Look for resonant peaks in the spectrum.
- If needed, the resonant field profile can be isolated through the Fourier analysis of an additional simulation and saved for later use.

This procedure is described in more detail in Section 8.B and illustrated in Tutorial 4 in Chapter 10. As usual, it is important to perform a convergence study with simulation parameters such as the grid sizes to determine if accurate results have been achieved.

Additional Comments

The following concepts are relevant for cavity mode calculations:

- Since most cavity modes exhibit some form of symmetry, it is usually possible to use symmetric boundary conditions to accelerate the simulations. See Section 6.C for more details on using symmetric boundary conditions efficiently and correctly.
- As with any spectral computation, an appropriate time window is required to get good results. However, if the mode is too leaky, a long time window will be useless: the mode will decay too quickly to get meaningful results. This difficultly can be overcome using the **Continue Simulation** option described in Section 6.D to successively refine the mode profile using a series of shorter simulations.
- A simple method to improve the precision of the center frequencies and corresponding accuracy of the mode profile is to perform a longer simulation. However, this approach can result in prohibitively long simulations. Section 8.A.2 describes some preferred techniques based on ideas from harmonic analysis.

8.A.2. Calculating *Q*

Once the resonance has been found, there are different methods to solve for Q. While all these methods would give the same results for very long simulations with very high spectral resolution, different approaches yield very different performance in terms of accuracy, speed and convergence.

Method 1: Measure Ratio of Stored Energy and Leaking Power

A common, very general definition for Q is 2π times the 'stored energy' divided by the 'energy lost per oscillation'. To measure these quantities, a computed cavity mode would be launched into the structure using the FDTD Continue options (see Section 6.D), and then allowed to decay away. During the simulation, it is straightforward to use FullWAVE monitors to calculate the "stored energy", which is the value of an energy density monitor, and the "energy lost per oscillation", which is the value of a power monitor divided by the center frequency of the resonance.

One drawback to this method is that both of the monitor results commonly exhibit periodic oscillations about their mean value. As a result, the ratio of the two quantities strongly oscillates making it difficult to extract a reliable Q value. One approach is to average time-averaged versions of these two quantities, which produces better results. An additional problem is that energy and power monitors *measured over large domains* can be quite computationally expensive especially when clustering. On the whole therefore, this is not the recommended approach, as the methods described next are faster, more accurate and easier to use.

One application when this method must be used is in separating in-plane and out-of-plane losses in 3D photonic crystal slabs. In that case, we need to separately measure the energy leaving in different directions and the calculation of the power flux is central to this. This type of calculation is discussed in Section 8.C.5.

Method 2: Measure Energy Decay in Cavity

Since the stored energy of an isolated mode decays as $U(t) = U_0 e^{-\alpha t}$ where $\alpha>0$ it is easy to show based on the definition given in the previous method that, for a resonance centered at a frequency ν, $Q = 2\pi\nu/\alpha$. To use this method, the resonant mode is launched into the cavity via the FDTD Continue options (see Section 6.D), and a time-averaged energy density monitor can be used to calculate the stored energy in the cavity as the mode decays away.

The decay rate α can then be found by either measurements from the graph using the mouse coordinates, or by using WinPLOT's fitting features. Since this method relies on curve-fitting, the oscillations seen in Method 1 are less of a problem.

Method 3: Measure width of resonance

The Q factor can also be defined as the ratio of the center frequency and full-width at half-maximum of the resonant peak as $Q = \nu_0/\nu_{FWHM}$. (It is a worthwhile exercise to show that for a Lorentzian line, this definition and the other formulae above are equivalent.) The best way to find the FWHM of the peak is to fit a Lorentzian to the spectral line of the resonance. After fitting the Lorentzian to the spectral power, the expression takes the form:

$$P(\nu) = \frac{P_0}{1 + \frac{(\nu - \nu_0)^2}{(Q/(2\nu_0))^2}}$$

The value of Q can either be extracted from this fit, or if the peak is sufficiently well resolved, by directly measuring the FWHM. The critical drawback to this method is the frequency resolution for high-Q structures. Consider a structure where Q>5500. This implies that the FWHM $\Delta\nu<1.1\text{x}10^{-4}$. Thus to resolve this peak well, a frequency resolution of about 10^{-5} would be needed. Using the basic time-frequency relation of the DFT used by the FullWAVE monitors, the simulation would need approximately 100,000 time steps which might be computationally prohibitive depending on the particular structure.

Therefore, this is not a practical approach for high-Q structures and it is not used directly. However, the next section describes how a modified version of this method can be very successful and efficient.

Method 4: Use Techniques of Advanced Harmonic Analysis

All the Q finding methods described so far rely strongly on the DFT of the field response. This is obviously true for Method 3, which attempts to measure the peak FWHM by fitting the spectrum, but is also true for the other methods since they rely on the accurate computation of a mode at the resonant frequency. The accuracy of the mode profile is itself limited by the DFT frequency resolution.

The Problem

While any procedure that improved the time-frequency limit would have obvious benefits, the usual tricks don't work. Running the simulations longer to improve the resolution is not an attractive option since in FDTD work we are almost always trying to reduce our simulation time. Zero-padding the time response is also impossible since it artificially broadens the resonance peaks. Another solution is needed.

The Solution

Naively, we might expect that the frequency resolution would be limited by the DFT time-frequency uncertainty relation—finer frequency resolution requires a longer time series. However, this is only true we have no knowledge of the form of the underlying signal. In this case, there is substantial knowledge: the field induced at a field measurement point x_0 by the arbitrary initial field will behave roughly as a sum of complex exponentials

$$f\left(x_0, t\right) = \sum_k c_k \exp\left(i2\pi M_k t\right) + \text{c.c.} \tag{1}$$

where c.c. denotes the complex conjugate. The complex frequency, which can be represented as $M_k = \nu_k = \gamma_k$ contains both the oscillation frequency ν_k and the loss rate γ_k. With FullWAVE's convention of positive frequencies, a positive value of γ_k represents a decaying mode. Each mode excited by the initial field is represented by a different term in the sum (1). Even the portion of the field that radiates away quickly can be considered to behave like this but with a very high loss rate.

The expression (1) allows us to tackle the problem with the tools of harmonic analysis or harmonic inversion. A number of such procedures exist with greater or lesser degrees of speed, accuracy, and robustness but all have the following ideas in common (ideas which date back to Baron de Prony in the late eighteenth century). We replace a linear operator, the DFT, by a nonlinear fit of the parameters c_k and M_k. This immediately allows much higher precision in the extracted parameters than would be permitted by the DFT. As qualitative proof, consider for example that if Eq. (1) contains only one term, then with a couple of lines of algebra, we can extract the exact value of M_k from just two time points whereas a DFT applied to two points is essentially useless. However a nonlinear fit implemented poorly would be much slower than the FFT operation and potentially unstable. Modern techniques of harmonic analysis have revealed how to implement such fitting rapidly and reliably and recover the speed of the FFT by concentrating on narrow frequency bands of interest.

As one example of these techniques, consider the approach known as Padé interpolation. This method works as follows:

Take the DFT formula:

$$f(\nu) = \sum_{n=0}^{N-1} f(t) \exp^{i2\pi n\nu t}$$

By using the parameter substitution $z = \exp[i2\pi\nu t]$ it is apparent that the DFT is a polynomial of the parameter z, as is well known to all students of discrete signal processing. However, many functions are not well approximated by polynomials, particularly functions with zeros and poles. In the present case, the fact that the signal decays indicates that its spectrum features at least one pole in the complex plane. Such functions are typically much better approximated by rational functions, i.e. ratios of polynomials. In the technique of Padé approximation, we construct the unique rational function of two polynomials each of order $N/2$, such that the Taylor series of the Padé representation agrees with the original polynomial up to order N. Although there is little formal justification for the result, we frequently find that the Padé solution provides a much better approximation. Thus better frequency resolution can be obtained *without* using longer simulation times.

RSoft's Q-Finder tool incorporates a number of such ideas under the general label of *Fast Harmonic Analysis* or FHA. With these techniques, Q-Finder is able to extract accurate complex frequencies from relatively short propagation lengths. Moreover, the FHA algorithms are available in a standalone command line tool called `fhakit` described in the Utilities Appendix of the CAD manual.

8.B. Manually Calculating Cavity Modes and *Q* Factors

This section outlines the manual calculation of cavity modes and *Q* factors. It is only included so that you can understand the operations performed automatically by Q-Finder. You should work through this exercise and the calculation described in Tutorial 4 in Chapter 10. However, for real work, always use Q-Finder which is fast, convenient and less prone to accidental mistakes in setting up the problem.

8.B.1. Manual Calculation of Cavity Resonances

The procedure for the calculation of cavity resonances is a straightforward process, and can be outlined with the following steps:

- Draw the desired structure in the CAD.
- Set the **Excitation** to *Impulse* as described in Section 4.D.
- Launch an appropriate field shape in the center of the cavity.
- Use a monitor to measure the field within the cavity structure. Create a monitor and enable the Single-Value output type *Default Field*. Try to avoid points of symmetry and/or nodes to ensure a good time response.
- Make sure the **Wavelength Monitor** output option in the FDTD Output Options dialog box is set to *Yes*. This option is set by default when using an impulse excitation unless set otherwise by the user.
- Set an **Output Prefix** and run the FDTD simulation.
- Determine the resonant frequencies from the wavelength spectrum contained in the plot file `<prefix>.pwn`. The raw spectrum data is contained in the file `<prefix>.wmn`. Alternatively, the frequency spectrum, using the files `<prefix>.pfn` and `<prefix>.fmn`, can be used.

In general, it is a good idea to perform a convergence study on the spectrum to ensure that accurate results are found. This is especially important if the spectral resolution is not adequate when a longer simulation might be needed.

8.B.2. Manual Calculation of Mode Profile

Once the resonant frequency is known, the mode profile can be computed by enabling the **Compute Mode** option in the FDTD Advanced Parameters dialog. This option computes the cavity mode profile at the wavelength given by the value of **Free Space Wavelength**.

This procedure will result in the calculation of the major field component of a cavity mode of the whole simulation domain:

- Ensure that the **Excitation** is still set to *Impulse*.
- Enable the **Compute Mode** option in the FDTD Advanced Parameters dialog.
- If multiple field components of the mode are needed, if the mode found is to be refined in later simulations, or if the mode is to be used to calculate a Q factor, enable the **Continue Simulation** option in the FDTD Advanced Parameters dialog. This option outputs a binary state file which can be used as an initial field for later simulations, or provide the raw data out of which to extract field data using the utility `fwmodekit`. See Section 6.D.2 for more details on the use of this option.
- Set the **Free Space Wavelength** value in the Global Settings dialog to the desired resonant wavelength. Note that when using an impulse excitation, this value has no effect on the launch.

Note that this can be overridden by setting the variable `fdtd_modefrequency_real` to the desired frequency. The imaginary part of the mode frequency (for mode attenuation) can be specified by setting `fdtd_modefrequency_imag` to the desired value.

- Rerun the FDTD simulation. The major component of the mode will be saved in the file `<prefix>.m00`, with accompanying plot file `<prefix>.p00`.

In general, it is a good idea to perform a convergence study on the mode profile to ensure that accurate results are found.

8.B.3. Manual Calculation of *Q*

The preferred method for manually calculating *Q* Factors is Method 2 described in Section 8.A.2:

- Perform a cavity mode computation as described above to find the mode resonance and field profile. Repeat this step until a good solution is found. Be sure to save the computed mode profile using the **Continue Simulation** option.
- Add a monitor that encompasses a majority of the cavity region and measures the power.
- Launch the computed mode profile and run for a certain time while the power monitor measures the power flow from the cavity. The exponential decay of this monitor can be used to calculate *Q*. Repeat with a longer simulation time as necessary to ensure that an accurate *Q* was found.

For more details on how these steps can be performed, please turn to Tutorial 4 in Chapter 10. You should work through this tutorial now, before reading on to learn about Q-Finder.

8.C. Using Q-Finder to Calculate Cavity Modes & *Q* Factors

We now turn to use of the automated Q-Finder tool. Make sure you have read sections 9.A, 9.B and Tutorial 4 (in Chapter 11) before reading this section.

Despite the name, Q-Finder is designed to automate both the calculation of *Q* and the extraction of mode profiles, whether or not the *Q* is desired. The *Q* is always returned since its calculation is cheap compared to that of the mode profile.

Q-Finder automates several of the methods described above and possesses some “intelligence” in order to reliably locate peaks given an initial rough guess as to the peak location. The Q-Finder dialog can be opened via the CAD menu option Utility/Q-Finder.

Tutorial 7 in Chapter 10 explores the use of Q-Finder.

8.C.1 Preparing an .ind File For Use With Q-Finder

In order to use Q-Finder, you must first define the geometry in the RSoft CAD. This section describes several suggestions and requirements.

CAD File Prerequisites

The technique for finding modes and *Q*s places certain requirements on a CAD `*.ind` file. For example, there must always be a simple monitor present with a single-value output type of field, and, if measuring *Q* by energy decay, a monitor a single-value output of the energy in at least a large portion of the resonance is required. When a simulation is initiated, Q-Finder tries to determine that these prerequisites are satisfied, and displays an error if it finds a problem. The following discussion on the use of Q-Finder assumes that these prerequisites have been met.

Speed Issues

In some cases, the speed of a *Q* calculation can be greatly increased by slightly changing the display and output options. For example, updating the graphical field display every 1000 time steps might reduce the simulation time by one-third. Similarly, increasing the update time for the monitors can have a small but noticeable effect. A good general rule is that, unless you are specifically interested in watching the field evolve, you should always set these update rates so that the simulation is not drawing too frequently.

Appendix A contains tips and tricks for increasing simulation speed, but several items bear repeating here:

- Set `fdtd_update_time` such that a refresh of the field display only happens every second or less.
- Set `fdtd_monitor_update_monitor` such that a refresh of the monitor display only happens every second or less.
- Deactivate the wavelength monitor. The frequency monitor is more fundamental.
- Take advantage of symmetry conditions whenever possible.
- Disable energy density and power monitors when their results are not specifically needed. This is possible using the `fdtd_monitor_tmask` and `fdtd_monitor_smask` variables. (Q-Finder always applies these settings automatically.)
- If outputting spatial field slices, keep `fdtd_slice_time` as large as possible.
- Output spatial fields on a coarser grid than the simulation grid if at all possible.

The first two of these items are performed automatically if the option `SpeedOptimizations=1` is set in the `[QFINDER]` section of your `.ini` file.

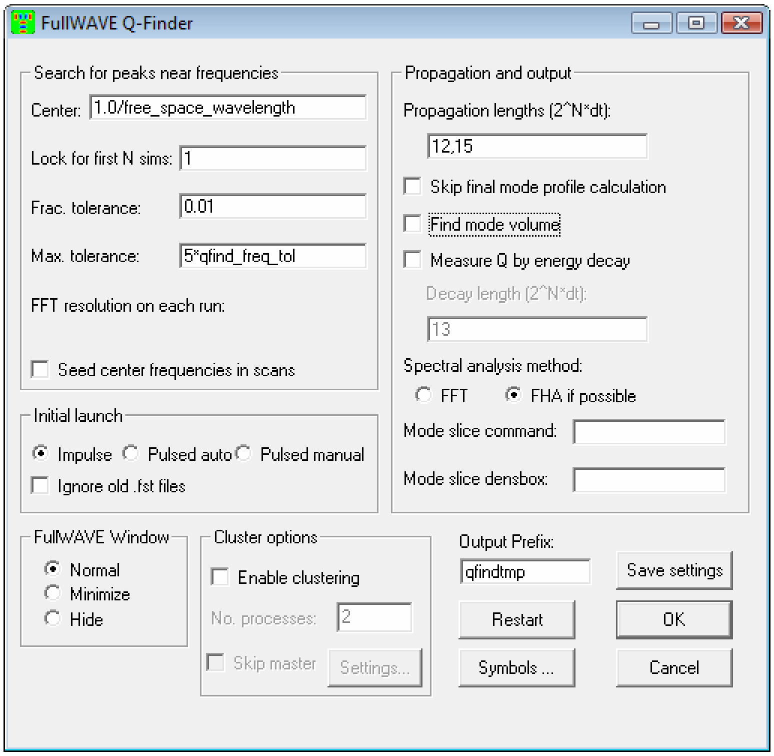
Using Symmetry Conditions

Since cavity-mode solving is a numerically intensive problem, it is important to exploit *FullWAVE*'s symmetry conditions whenever possible. For a 3D problem with full symmetry, this can cut simulation times by a factor of eight. However, it is very important to select the symmetry conditions carefully. Remember that when some field components are symmetric, others are anti-symmetric. When trying to excite a particular cavity mode and when reproducing literature results, check carefully that you are applying the correct conditions and consult the tables in Section 6.C.3 frequently. You may also wish to find your resonance using BandSOLVE, if you own it, on a coarse grid to check that the symmetry conditions are correct.

When comparing to results in the published literature, remember that most authors use *Z* to refer to the vertical axis, whereas for most designs in the RSoft CAD, *Y* is the natural vertical axis. Therefore when copying symmetry conditions from papers, check carefully that you are considering the correct field components and the correct symmetry directions. Our own experience is that it is easy to make mistakes on this point. Check everything at least twice!

8.C.2. The Q-Finder dialog

Once the CAD file is prepared, a Q-Finder calculation is initiated by opening the Q-Finder dialog, shown in Fig 8-1, using the Utility/Q-Finder menu item.


FullWAVE Q-Finder

Search for peaks near frequencies

Center: 1.0/free_space_wavelength

Lock for first N sims: 1

Frac. tolerance: 0.01

Max. tolerance: 5*qfind_freq_tol

FFT resolution on each run:

□ Seed center frequencies in scans

Initial launch

● Impulse ○ Pulsed auto ○ Pulsed manual

□ Ignore old .fst files

Propagation and output

Propagation lengths (2^N*dt): 12,15

□ Skip final mode profile calculation

□ Find mode volume

□ Measure Q by energy decay

Decay length (2^N*dt): 13

Spectral analysis method: ○ FFT ● FHA if possible

Mode slice command:

Mode slice densbox:

FullWAVE Window

● Normal
○ Minimize
○ Hide

Cluster options

□ Enable clustering

No. processes: 2

□ Skip master

Output Prefix: qfindtmp

Figure 8-1: The Q-Finder dialog.

Frequency Controls

Since any given cavity may support many resonances, *Q-Finder* needs hints about where to look for resonances. Q-Finder looks for resonances around a center frequency within defined tolerances. These settings are set by the controls in the top left of the dialog.

All these controls are defined in terms of frequency [μm^{-1}], rather than wavelength, since frequency is the more fundamental quantity.

Center

This is the initial guess for the peak of the desired resonance in μm^{-1}. By default, this value is internally updated as additional simulations are performed to refine the frequency spectrum unless the **Lock for first N syms** option described below is used. Q-Finder can calculate *Q*-factors for more than one resonance at a time. To calculate results for several modes, simply enter a list of several starting frequencies separated by commas. Each member of the list may be an arbitrary symbol table expression.

Lock for first N syms

This value controls the number of simulations for which the user-supplied center frequency is applied as the current center frequency. By default, it is only applied for the first simulation, and then Q-Finder uses the location of the peak in the most recently acquired Fourier spectrum to obtain the new estimate for the center frequency. On some occasions however, you may have accurate knowledge of the peak location from earlier work. This means that you may know the peak location more accurately than Q-Finder can determine from an early simulation of relatively short duration and coarse frequency resolution. In such a case, it is beneficial to apply the accurate value for several simulations until the frequency resolution of the later simulations exceeds that of your knowledge.

Another situation where this option is useful is if the cavity supports two or more closely spaced resonances. In that case, the early short simulations may fail to resolve the resonances and Q-Finder could lock on to the incorrect resonance. Imagine for instance that the sought-after resonance with higher *Q* happened to be more weakly excited by the launch field than the adjacent lower-*Q* resonance. Then Q-Finder may incorrectly lock on to the stronger but lossier second resonance and the potential *Q* of the cavity will be under-reported.

Frac. tolerance and **Max. tolerance**

These two tolerance settings control how Q-Finder searches for a resonance peak in the spectrum of the device. Initially, Q-Finder will look for a peak within the frequency band $\nu_0(1\text{-}\delta,1\text{-}\delta)$ where ν_0 is the current center frequency and δ is the value of the **Frac. tolerance**. If any peak is found within that range, it is selected as the peak of interest.

If however, the spectrum is monotonic in the search window, the window is increased in size and the search is continued. This continues until the tolerance has reached the value **Max. tolerance**. If a peak has not been located at this stage, Q-Finder terminates with an error. (However, if a peak is not found after the first propagation, Q-Finder performs a second simulation with the original frequency guess, since the spectrum of the first propagation is often very noisy, and improves on subsequent runs.) In this way, it is relatively easy to isolate the mode of interest, even if our initial knowledge of its frequency is not very precise. At the same time, by keeping the **Max. tolerance** relatively small, we can avoid Q-Finder locking on to the wrong mode.

The setting **Frac. tolerance** can also be set by the variable `qfind_freq_tol` and has a default value of 0.01. It is convenient to specify the **Max. tolerance** in terms of that parameter. The default value is five times `qfind_freq_tol`.

FFT Resolution

These non-editable fields display the frequency resolution for the current values of the **Propagation lengths** described in the next section.

Seed center frequencies in scans

This option enables the seeding of the center frequency values during a MOST scan. This is useful when the center frequencies shift during a scan. When this option is set, the user-defined center frequencies are used for the first point of the scan, while subsequent points use the center frequencies found for the previous point.

Propagation and output

The settings in the top-right panel control the sequence of propagations and various output options.

Propagation lengths

As seen in the discussion of the manual method of mode computation in Section 8.B and Tutorial 4, finding resonant modes and *Q* values typically involves a sequence of calculations. This is always the case for finding a mode profile or a *Q* by energy decay: there must be at least one propagation to determine the resonant frequency, a second to construct the mode profile at the desired frequency and a third to monitor the energy decay. If only the *Q* is required, the FHA method is often able to extract accurate results from just a single propagation, however, it can also benefit from using a series of runs of increasing length that progressively "clean-up" the spectrum as the resonant state is isolated.

Thus Q-Finder is built around the idea of specifying a sequence of (usually increasing) propagation lengths. The idea is that short initial calculations allow most of the extraneous components of the initial launch field to be radiated away and the peak frequency to be determined roughly. Typically, three or four propagations are sufficient for finding resonant modes. For the FHA method without a resonant profile, you may settle on two propagations or even just one if experience has shown that it is giving reliable answers for your structure. On the other hand, for some very low-*Q* structures, it may be necessary to use quite a number of relatively short propagations.

After many multiples of the characteristic decay time $\tau_d = Q/2\pi\nu_0$, essentially all the mode energy will have decayed away and there is little point in simulating for additional time. However, it may be necessary to use longer simulation times for structures that support a number of closely spaced resonances. In this case, the initial simulation needs to resolve each of these peaks properly and you need to specify the center frequency of the desired resonance accurately enough to isolate it from the neighboring ones.

It is important to appreciate that the appropriate number and length of propagations varies considerably from structure to structure. It can even vary strongly from mode to mode for a single structure, if the modes have highly different *Q*s, a not-uncommon situation. As always, some initial investment in experimenting with parameters will pay off and cut your overall project time.

The simulation lengths are specified as "time exponents" or powers of two times the FullWAVE **Time Step** value given by the built-in symbol `fdtd_time_step`. Thus a time exponent of `10` would cause the FullWAVE simulation to perform 2^10, or 1024, time steps.

The sequence of lengths is set by the field:

Propagation lengths (2^N*dt)

This field provides the "time-exponents" for the length of these simulations, and should contain a comma-separated list of increasingly larger exponents. For most purposes, three simulations are sufficient, with the last one possibly being much longer than the first two.

Choosing the Q Calculation Method

As we have seen, Q-Finder can calculate the *Q*-factor using two methods.

- *Fast Harmonic Analysis (FHA)*

  This is the standard method, which is always performed, uses the FHA techniques discussed in Section 8.A.2. If using this method and a mode profile is not needed, a little time can be saved by selecting the option **Skip final mode profile calculation**. This prevents calculation of the mode profile on the final simulation run only.

- *Energy Decay*

  This method measures the rate of decay of an energy monitor (Method 2 in Section 8.A.2). This is usually slower and less accurate than the FHA techniques but has two possible uses. Firstly, it can be used as a check on the FHA results. Secondly, by introducing multiple energy and power monitors, it can be used to separate the energy loss into contributions in different directions. For instance, the *partial-Qs* for in-plane and out-of-plane loss from a photonic crystal slab can be obtained in this way. The FHA method can only generate an overall decay rate for the whole mode. The energy decay method is activated by selection the **Measure Q by energy decay** option (see below).

  If the energy decay option is selected, the **Decay Length** field sets the length of the decay propagation as a time exponent. Usually this value can be set considerably smaller than the longest propagation length. You should aim to have a decay length so that the energy falls by a factor of 1/e or so though reasonable results can be obtained with somewhat shorter propagation times.

Other options

Find Mode Volume

This option is not compatible with the **Skip final mode profile calculation** option.

This option enables the computation of the mode volume, which is defined as:

$$Volume = \int_V \frac{U_E(\mathbf{r})}{\max\left[U_E(\mathbf{r})\right]} dV,$$

where

$$U_E(\mathbf{r}) = \frac{1}{2}\varepsilon(\mathbf{r})\left|\mathbf{E}(\mathbf{r})\right|^2$$

is the electric energy density, **E** is the electric field, ε is the spatial dependent index, **r** is the spatial coordinate, and *V* is the whole simulation domain. Since this function oscillates as a function of time, it is evaluated at the point of the optical cycle where the electric energy reaches its maximum. This gives consistent answers for the mode volume. Note that the mode volume is defined in terms of the electric energy density, rather than the magnetic or total densities. The electric definition is most useful since the volume is usually used in calculations involving electric dipole radiation.

The mode volume calculation is actually performed by the tool `fwmodekit`, which can also be called from the command line on an existing `.fst` file.

Spectral analysis method

This option determines whether the estimate of the resonant frequency obtained after each propagation is determined by an FFT or by the FHA method. The default is to use the FHA, but in rare circumstances the FFT method may be slightly more robust if less precise. You should only ever change this option if Q-Finder is having trouble locating a resonance.

Mode slice command and **Mode slice densbox**

These options allow the user to append commands to the calls Q-Finder make to `fwmodekit`.

Initial Launch

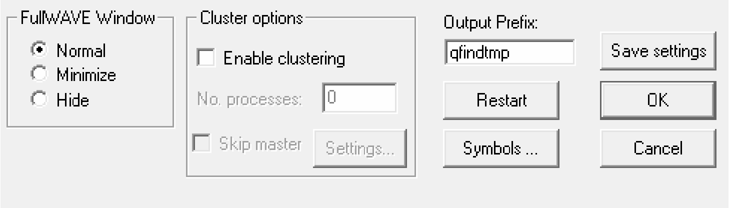
The initial launch used can be set by the user.

Ignore old .fst files

Q-Finder uses FullWAVE `.fst` files to store intermediate results. If any of the simulation domain, grid resolution or time step are changed in the CAD, this may cause the `.fst` files on disk to become mismatched with the new simulation. This is indicated by a FullWAVE error. In such cases, activating this option ensures that any old files are ignored.

8.C.3. Performing a Simulation

The bottom of the Q-Finder dialog controls allows the user to perform simulations, and controls how they are performed.



FullWAVE Window

● Normal
○ Minimize
○ Hide

Cluster options

□ Enable clustering
No. processes: 0
□ Skip master

Output Prefix: qfindtmp

Figure 8-2: The bottom portion of the Q-Finder dialog where simulations options are set.

Running the Simulation

To run a calculation, enter an **Output Prefix** and click **OK** to initiate the calculation. The Q-Finder status window opens itself, and then it opens the window for the first FullWAVE simulation.

As the calculation proceeds, the Q-Finder status window will show the calculation progress in its log window. Information such as the current center frequency as well a Purity parameter are displayed. The Purity parameter is a measure of how much the field within current mode profile estimate is associated with the desired frequency; a value near 100% indicates that the mode is just about perfect, while a much lower value indicates there is substantial energy from another peak. The Q-Finder status bar indicates the total number of simulations performed so far, as well as the progress and duration of the current FullWAVE simulation. This is especially useful when the FullWAVE window has been hidden.

At the end of the *Q* calculation, the Q-Finder status window will display the computed *Q* value, as well as any other requested output, such as mode volume.

FullWAVE Window Control

The **FullWAVE Window** controls shown in Fig. 8-2 set how the FullWAVE simulation window will appear during a simulation. There are three settings:

- *Normal*

  A graphical FullWAVE window will be displayed as the simulation progresses.

- *Minimize*

  The graphical FullWAVE window will be opened in the task bar.

- *Hide*

  The FullWAVE simulations will be performed entirely in the background.

We suggest using the *Normal* setting when first experimenting with calculations, and then the *Hide* setting when performing long or routine runs to reduce your desktop clutter. This is particularly important when using MOST to perform scans or optimizations with Q-Finder since there are then potentially three different windows associated with the calculation – the governing MOST window, the Q-Finder window controlled by MOST and the FullWAVE simulations launched by Q-Finder.

Clustering Controls

The **Cluster Options** shown in Fig. 9-2 allow Q-Finder to use the clustered version of FullWAVE if you own a cluster license.

Since dealing with clustering is a complex operation you should be sure that you fully understand the procedures for running plain FullWAVE cluster calculations on your system, before you attempt to combine clustering with Q-Finder.

The clustering options are as follows:

Enable clustering

This checkbox enables the use of clustering.

No processes

This sets the number of processes to use for the clustered simulation.

Skip Master

This checkbox indicates that the master node should be excluded from the simulation, and corresponds to the variable `fdtd_nx_master`.

Settings...

This button opens the Cluster Settings dialog described in Chapter 7.

Post processing

This feature attempts to rerun the calculation but without repeating simulations. It can be useful if making slight adjustments to frequency tolerance settings to isolate a peak.

8.C.4. Output Files

Q-Finder produces a number of output files.

Main Output

The principal file is `<prefix>_qvs.dat`. It contains the values of the center frequency, *Q*, and the mode volume.

Depending on the settings and algorithm used, not all of these values may be meaningful. For example, the mode volume is always zero unless the **Find mode volume** option is active.

Intermediate Output

In trying to understand the behavior of resonances it can also be useful to examine several intermediate files, typically .tmn (monitors), .fmn (frequency monitors) and .p00/.m00 (mode profiles) files. Time and frequency monitors are stored at the end of each FullWAVE simulation. They are named as follows: <prefix>_m<j>_<stage>.tmn or <prefix>_m<j>_<stage>.fmn, where <j> is an index over the number of center frequencies starting from 0, and <stage> is an index over the sequence of time exponents starting from 1. So the monitors for the third simulation for the second resonance would be stored in the file <prefix>_m1_3.tmn.

Note that for the purposes of increased speed, energy density and power monitors are disabled except during the final decay stage of the energy decay method, which is normally the only time they are useful. Spatial monitors, which are very expensive, are always disabled by the Q-Finder. To override this behavior, set qfind_monitor_tall=1 (calculate all temporal monitors including energy and power) or qfind_monitor_sall=1 (calculate all spatial monitors), but only do so if you absolutely must obtain intermediate energy or spatial data.

Remember also that FullWAVE's .tmn and .fmn files always contain the *signed square roots* of the data displayed by the corresponding .ptm of .pfm file. You must add /powy2 to a WinPLOT file in order to graph the squared data. The standard FullWAVE output files include this option.

A convenient way to examine the intermediate output is using DataBROWSER. The DataBROWSER button on the Q-Finder toolbar opens DataBROWSER in *regex* mode with a prefix that extracts just the final data: <prefix>_m[0-9]+.p. To examine intermediate data, just remove the last 2 characters to obtain <prefix>_m[0-9]+ and hit the return key. You can adjust the regular expression to concentrate on the files of interest. The most important are usually the .tmn and .fmn files discussed above. See the DataBROWSER manual if you need to learn about regular expressions and DataBROWSER's regex mode. A small investment in learning about the regex mode will increase enormously your ability to quickly explore and understand large numbers of data files.

8.C.5. Advanced Q-Finder Techniques

In this section, we look at some of the more complicated operations that can be performed with Q-Finder.

Command-line interface

Q-Finder can be run as a command line tool like other RSoft simulators by passing the option '–c':

- *Windows:*

  ```
  qfinder.exe –c <indfile> [name=value [,…]]
  ```

- *Linux:*

  ```
  xqfinder –c <indfile> [name=value] [name=value] […]
  ```

  or

```
qfinder –c <indfile> [name=value] [name=value] […]
```

When using Q-Finder in this mode it supports some special symbol table variables to control the behavior of FullWAVE simulations spawned by Q-Finder. These symbols and their possible values are

Variable	Description
`qfind_child_simwin`	Controls the style of the FullWAVE simulation windows. Possible values are: `0` (normal), `1` (minimized), or `2` (hidden).
`qfind_cluster_np`	Sets the number of processes to be used in a clustered simulation. This setting is required if clustering is to be used. Without it, the calculation will revert to a non-clustered configuration. A value of 0 or 1 indicates a normal non-clustered calculation.
`qfind_cluster_machinefile`	Sets the name of the machine file that contains the list of cluster hosts to be used. This setting is optional, and if not used, the standard list of machines from the **Cluster Settings** dialog will be used.

As an example, to start a Q-Finder calculation with four processes on the two dual-core machines `scylla` and `charibdis`, type the following:

```
qfinder.exe qfind_cluster_np=4 qfind_cluster_machinefile=mymachines.txt
```

where the contents of the file `mymachines.txt` is:

```
scylla
charibdis
```

Calculating Power Fluxes and Partial Qs

To this point, we have concentrated solely on the total *Q* describing the total loss rate of energy from the cavity. In some applications, it can be useful to separately evaluate the energy lost through different channels. For example, for cavities in photonic crystal slabs, it is common to separately consider the in-plane losses due to leakage through the crystal, and the out-of-plane losses due to wave-vector components of the field that lie above the light line of the slab. The energy lost through each channel can be associated with its own *Q*. We will term such a quantity a partial *Q* with

$$Q_{\text{partial},i} = 2\pi \frac{\text{Total stored energy}}{\text{Energy lost per oscillation through channel } i}$$

Partial *Qs* add up like resistors in parallel:

$$\frac{1}{Q_{\text{total}}} = \frac{1}{Q_1} + \frac{1}{Q_2} + \frac{1}{Q_3} + \dots$$

Q-Finder is able to calculate partial *Q*s. The idea is to identify a single energy density monitor as the dominant or representative energy. If there is more than one energy monitor present, the one that samples the largest fraction of the simulation domain is considered the dominant monitor. All power monitors in the domain are then considered to measure energy leaking away from the dominant energy monitor. Q-Finder automatically calculates a partial *Q* for each power monitor it finds.

For example, for a 3D calculation of a PC slab membrane, it would be appropriate to define a 3D energy density monitor that matches the membrane region. Power monitors can be placed along each face of the membrane.

Note that these calculations can only be performed in the energy-decay configuration, since they require the launch of a resonant field profile. It should also be noted that for high *Q* cavities, the calculation of partial *Q*s is often quite susceptible to cancellation problems, and it can be challenging to obtain good results.

Scanning and Convergence

Like all tools in RSoft's Photonic Component Design Suite, the Q-Finder can be operated from within MOST to perform scanning and optimization. To activate Q-Finder within MOST, select the entry FullWAVE Q-Finder for the simulation tool. There are a number of standard measurements for quantities such as the complex, real and imaginary resonant frequency, the *Q* as calculated by FHA or energy decay, and the mode volume.

Restarting Q-Finder Calculations

It is possible to restart a Q-Finder calculation at different stages by skipping any number of stages in the spectral search. T do this, set the variable `qfind_skip_stages` to the number of stages to skip. For example, setting this variable to `2` skips the first two stages and uses the existing files `<prefix>_m0_[12].tmn/fmn`. This variable has a default value of `0`. When using this option, do not change the value of `qfind_tpow_spec`.

9

Basic Tutorials

This chapter contains several basic tutorial examples which illustrate the basic use of the FullWAVE simulation engine. These tutorials are increasingly complex and many of them build on concepts established in earlier tutorials, it is important to perform them in order, even if a particular topic is of most interest to you. It is also recommended that users work through the basic tutorials provided in this chapter before moving to the advanced tutorials in Chapter 10.

These tutorials do not have associated `*.ind` files.

Basic Tutorial 1: Basic 2D Simulation

This tutorial illustrates basic CAD layout features by creating a new circuit and demonstrates a sample workflow to perform a simulation and obtain simulation results. Our goal will be to layout and simulate the propagation of a mode at 1.55 µm along a single 2D slab waveguide with a cladding index of 1, a core index of 1.5, and a width of 1 µm. We will calculate the transmitted power along 10 µm of this waveguide.

This tutorial will introduce basic concepts about the RSoft CAD interface and the FullWAVE simulation engine that are critical for every user to understand; it is recommended that every new user work through this tutorial. For additional information about the RSoft CAD, please consult the CAD manual.

CAD Window Basics

We begin by opening up the RSoft CAD window. Windows users can do this via the Start menu and Linux users can issue the command `xbcad` at a command prompt.

The CAD window, shown in Fig. 1, contains the following elements:

- *A menubar near the top of the window*

  The menubar allows access to various standard file and editing operations such as opening and saving index files and cutting and pasting waveguide components in a circuit.

- *Two toolbars with icons just below the menubar*

  The top two toolbars contains icons for these standard file and editing operations, viewing options, as well as other common functions.

- *A toolbar along the left edge of the window*

  The left toolbar contains icons for standard CAD operations such as selection and zooming, as well as icons for choosing the different waveguide components that can be added to the circuit. Additional icons control the simulation and related aspects of the program.

- *A status line at the bottom of the window*

  The status line provides information on the current layout mode, along with coordinate display during drawing.

For a complete description of the functionality of the RSoft CAD interface, please consult the CAD manual.

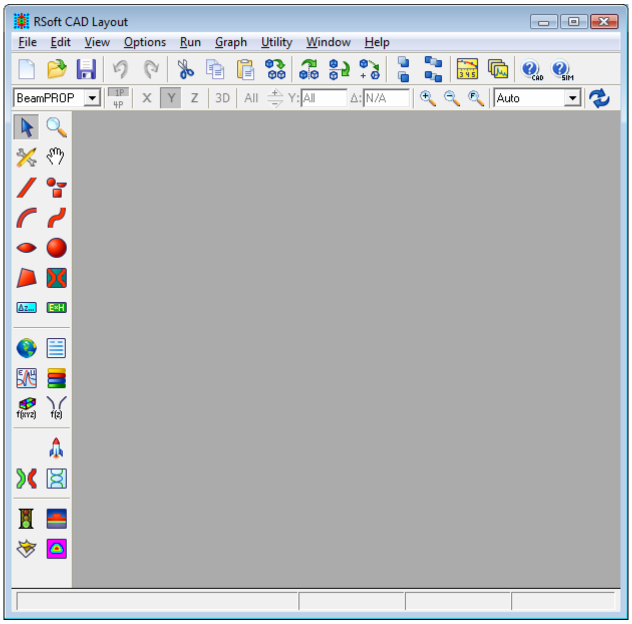


Figure 1: The RSoft CAD program window, showing the menubar at the top, the top and left toolbars, and the status line at the bottom.

Creating a New Circuit

To create a new circuit, click on the **New Circuit** icon in the top toolbar (the leftmost icon on the top toolbar). Alternatively, choose File/New from the menu. The startup dialog appears, as shown in Fig. 2. This dialog requests basic information about the circuit to be modeled, including the wavelength of the light, the background refractive index, the default index change between the background material and the structure, and the approximate overall dimensions of the device.

The startup dialog is shared by other RSoft packages; all options are not applicable to FullWAVE.

Choosing Initial Settings:

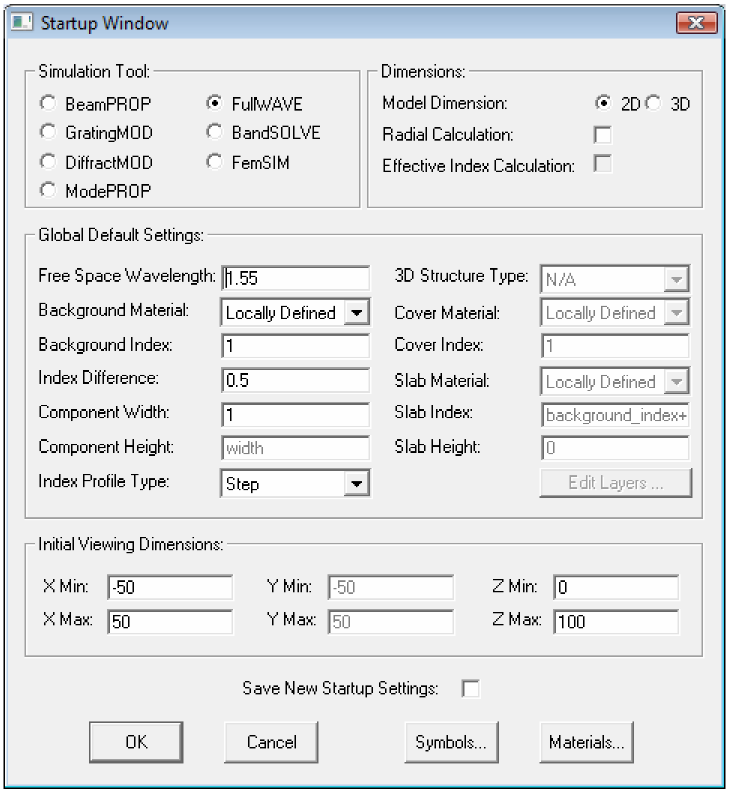
For this simulation, we will use the following settings:

Parameter	Value	Description
Simulation Tool	*FullWAVE*	Sets FullWAVE as the simulation engine to be used.
Model Dimension	*2D*	Creates a 2D simulation.
Free Space Wavelength	`1.55`	Sets the global wavelength [μm] for the simulation. For a CW simulation, this represents the driving wavelength, and for a Pulsed simulation this represents the center wavelength.
Background Index	`1.0`	Represents the real refractive index of the background material where no structure is defined.
Index Difference	`0.5`	Represents the default index difference between a component and the background material. In this case, our structure will have a default index of 1.5; each component can have a unique value if desired.
Component Width	`1.0`	This sets the default width of a component [μm]; each component can have a unique value if desired.

Once these settings are made, click **OK** to continue. Note that changes can be made to these settings via the Global Settings dialog box which can be opened by clicking the 'Globe' icon in the left toolbar.

The Layout Window:

A layout window is created within the main CAD program window, as shown in Fig. 3. A coarse coordinate grid is indicated by dots, and the X and Z axes are indicated by light gray lines. When the mouse is moved into the layout window, the cursor becomes a cross-hair, and the coordinate display in the status line indicates the mouse position in real coordinates [μm].



Startup Window

Simulation Tool:

- ○ BeamPROP
- ○ GratingMOD
- ○ DiffractMOD
- ○ ModePROP
- ● FullWAVE
- ○ BandSOLVE
- ○ FemSIM

Dimensions:

Model Dimension: ● 2D ○ 3D

Radial Calculation: □

Effective Index Calculation: □

Global Default Settings:

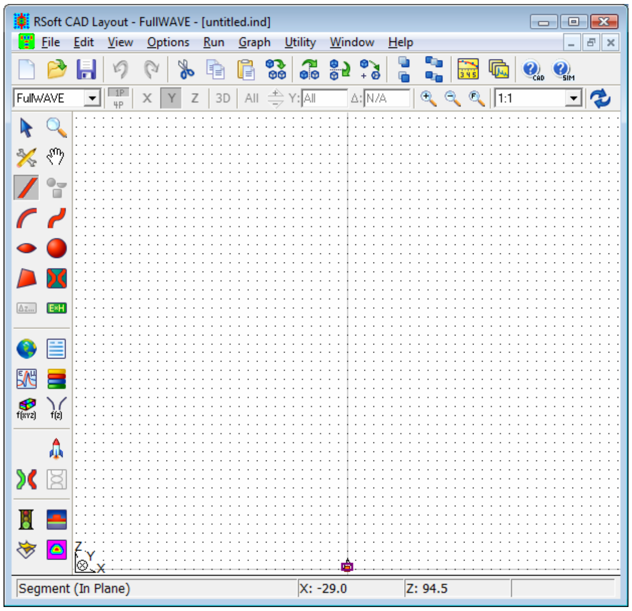
Free Space Wavelength:	1.55	3D Structure Type:	N/A
Background Material:	Locally Defined	Cover Material:	Locally Defined
Background Index:	1	Cover Index:	1
Index Difference:	0.5	Slab Material:	Locally Defined
Component Width:	1	Slab Index:	background_index+
Component Height:	width	Slab Height:	0
Index Profile Type:	Step		Edit Layers ...

Initial Viewing Dimensions:

X Min:	-50	Y Min:	-50	Z Min:	0
X Max:	50	Y Max:	50	Z Max:	100

Save New Startup Settings: □

Figure 2: The startup dialog which appears whenever a new circuit is created, and requests basic information about the circuit to be modeled.



RSoft CAD Layout - FullWAVE - [untitled.ind]

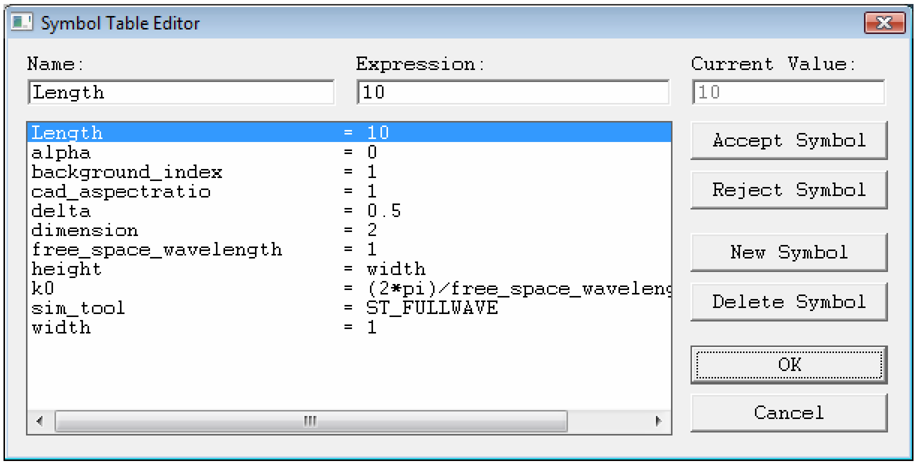
Segment (In Plane) X: -29.0 Z: 94.5

Figure 3: The layout window where waveguide components are added to the circuit.

Creating Variables via the Symbol Table

An important concept in the RSoft CAD is the Symbol Table. The Symbol Table allows the user to create/modify both built-in and user-defined variables. These variables can be used to define virtually any parameter within the CAD. Each symbol can be a function of any other symbol, creating a parametric design environment.

The Symbol Table can be opened via the **Edit Symbols** icon in the CAD window. Figure 4 shows the symbol table for this design file which contains several symbols already defined that correspond to the settings made in the **New Circuit** dialog box above.



Name: Length

Expression: 10

Current Value: 10

```
Length                   = 10
alpha                    = 0
background_index         = 1
cad_aspectratio          = 1
delta                    = 0.5
dimension                = 2
free_space_wavelength    = 1
height                   = width
k0                       = (2*pi)/free_space_waveleng
sim_tool                 = ST_FULLWAVE
width                    = 1
```

Figure 4: The Symbol Table Editor. The variable `Length` has been defined with a value of `10`.

In this tutorial, we are going to create a variable which will be used to set the length of our slab waveguide. Define a symbol `Length`, and set it equal to `10`. To do this, click **New Symbol**, enter the symbol **Name** (`Length`) and **Value** (`10`), and then click **Accept Symbol**. The variable `Length` should now appear in the symbol table.

The names of symbols in the symbol table are case-sensitive. Also, symbols which begin with a capital letter are displayed at the top of the list, and so it is recommended that any user-created variables start with a capital letter so that they will be easier to find in the symbol table.

Adding the Waveguide

We can now add the waveguide structure to the CAD layout window.

Drawing the Waveguide:

The CAD has several convenient ways to add a component. For this tutorial we choose the straight waveguide component, or **Segment Mode**, which has already been selected on the left toolbar by default. To draw the waveguide, move the cross-hair cursor to the beginning of the straight waveguide section, which we will conveniently locate at coordinates (X=0, Z=0). After positioning the cursor, press but do not release the left mouse button. Now move the cursor to the end of the straight waveguide section, located around the coordinates (X=0, Z=10). We will fix the length of this waveguide segment using the variable we defined previously.

As you move the cursor, the center line of the waveguide is repeatedly drawn in blue and erased (this process is termed rubberbanding), and the status line coordinate display is continuously updated. When the cursor is positioned at the desired end point, release the mouse button. The waveguide segment is drawn as shown in Fig. 5.

If a mistake is made when adding any waveguide component, simply click on the **Undo Last Change** icon in the top toolbar (the reversing arrow), and the waveguide circuit will be restored to its previous state.

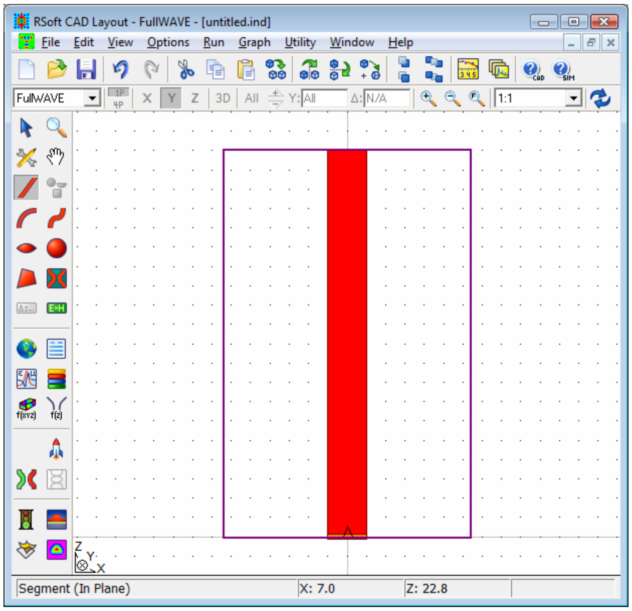
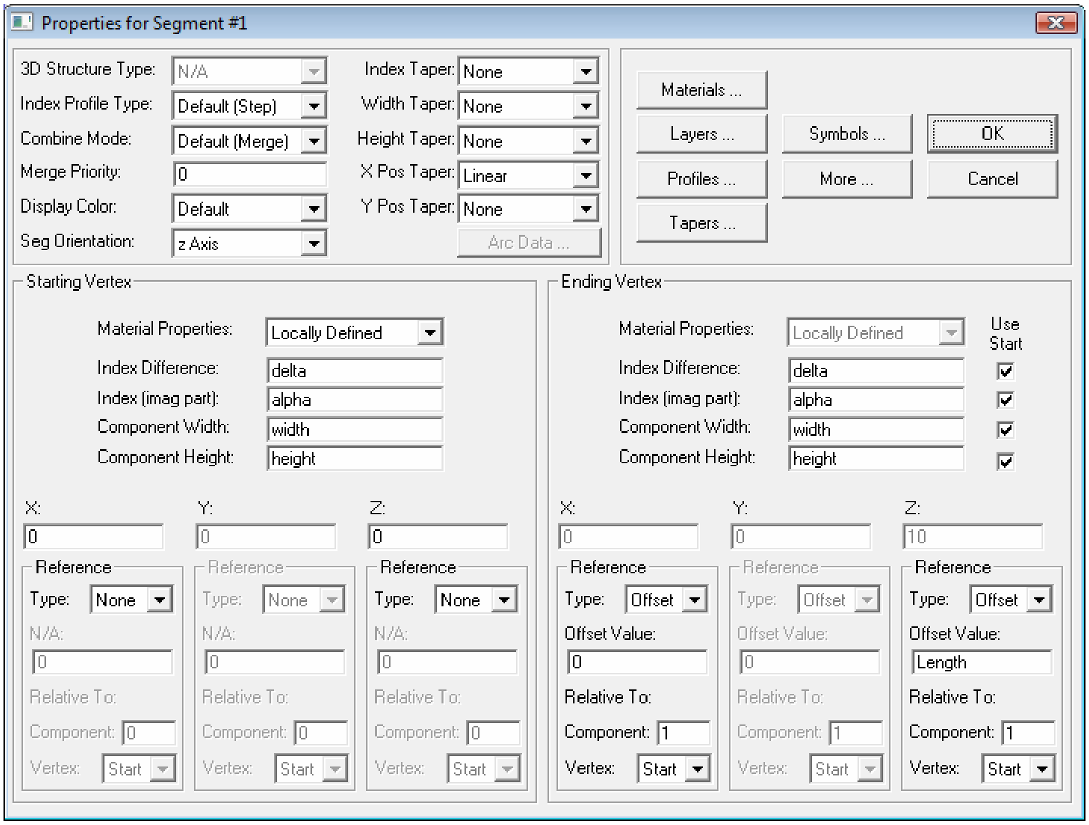

RSoft CAD Layout - FullWAVE - [untitled.ind]
File Edit View Options Run Graph Utility Window Help
FullWAVE
X Y Z 3D All Y: All Δ: N/A 1:1
Segment (In Plane)
X: 7.0
Z: 22.8

Figure 5: A straight waveguide segment as it appears after drawing. The simulation domain and launch field are also shown.

Setting the Length of the Waveguide:

For this example, we are going to set the length of the waveguide to 10 µm. Open the Segment Properties box for this waveguide by right clicking on the segment in the CAD window. The dialog box shown in Fig. 6 will appear. This dialog box enables the user to set the waveguide properties such as refractive index and waveguide geometry for the waveguide we have drawn.

The CAD is an object oriented design environment which so each segment can has a different Segment Properties dialog box, and therefore a different set of parameters. For the purposes of this example, we will not explore each the use of all the fields in this dialog. A detailed description of this dialog can be found in Chapter 5 of the CAD manual.



Properties for Segment #1

3D Structure Type: N/A
Index Profile Type: Default (Step)
Combine Mode: Default (Merge)
Merge Priority: 0
Display Color: Default
Seg Orientation: z Axis

Index Taper: None
Width Taper: None
Height Taper: None
X Pos Taper: Linear
Y Pos Taper: None

Starting Vertex

Material Properties: Locally Defined
Index Difference: delta
Index (imag part): alpha
Component Width: width
Component Height: height

	X:	Y:	Z:
	0	0	0
Reference Type:	None	None	None
N/A:	0	0	0
Relative To: Component:	0	0	0
Vertex:	Start	Start	Start

Ending Vertex

		Use Start
Material Properties:	Locally Defined	
Index Difference:	delta	☑
Index (imag part):	alpha	☑
Component Width:	width	☑
Component Height:	height	☑

	X:	Y:	Z:
	0	0	10
Reference Type:	Offset	Offset	Offset
Offset Value:	0	0	Length
Relative To: Component:	1	1	1
Vertex:	Start	Start	Start

Figure 6: The Segment Properties dialog box for the waveguide segment in this design.

In order to set the length of this segment, make sure the **Reference Type** of the Z coordinate of the Ending Vertex is *Offset*, and **Relative To** the starting vertex of **Component** `1`. This tells the software that the Z coordinate of the ending vertex of this segment, or Segment #1, should be computed as an offset from the starting vertex, or vertex 0, of this segment, or component 1. This offset will therefore equal the length of the segment. To set the offset, which is the length in this case, set the **Offset Value** equal to the variable `Length` we previously defined. The dialog box should now appear as in Fig 6.

While this may seem like a lot of configuration for a simple structure, the functionality described here will allow much more complicated structures to be created in a simple fashion, and allow for parametric designs which can easily be scanned/optimized to achieve optimal performance.

You may notice that the waveguide you have drawn does not have the same shape as the waveguide shown in Fig. 5. This is most likely due to the aspect ratio and viewing window of the CAD display. To set the aspect ratio to 1:1, use the pulldown menu on the view toolbar. To view the entire structure, click the View/Full button in the view toolbar.

In general, the next step would be to continue to add elements to the CAD layout in order to create the desired device. This process is described further in the CAD manual. For the purposes of this example, there are no more elements to add.

Choosing the Excitation

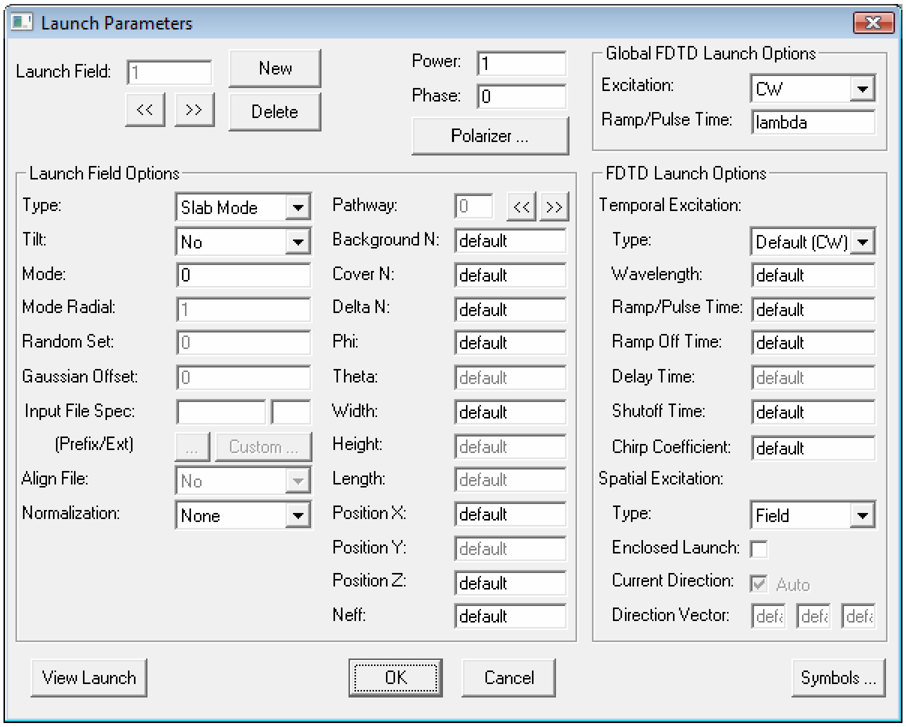
The excitation, or launch field, has two main aspects: a temporal aspect and a spatial aspect. The properties of the launch field are controlled via the Launch Parameters dialog (opened via the **Edit Launch Field** button on the left toolbar in the CAD). Several aspects of the launch field can also be set via the FullWAVE Simulation Parameters window (opened via the **Perform Simulation** button on the left toolbar) as a convenience.

In this tutorial, we will launch the fundamental mode of this waveguide at 1.55 µm. with a CW modulation. We will set the spatial aspects of the field in the Launch Parameters dialog, and the temporal aspects in the FullWAVE Simulation Parameters dialog.

The Spatial Aspect:

This spatial aspect of our launch field should be the fundamental mode of the waveguide component we have created. Since this a simple 2D slab structure, FullWAVE can calculate an analytical mode solution. Had this been a more complicated structure, we would have to either calculate the mode using the built-in mode solvers or another software package, and then launched the mode as a data file.

To set the launch field to the analytically calculated mode profile, click the **Edit Launch Field** button on the left toolbar in the CAD to open the Launch Parameters dialog.



Launch Field: 1
Power: 1
Phase: 0

Global FDTD Launch Options
Excitation: CW
Ramp/Pulse Time: lambda

Launch Field Options
Type: Slab Mode
Tilt: No
Mode: 0
Mode Radial: 1
Random Set: 0
Gaussian Offset: 0
Input File Spec: (Prefix/Ext)
Align File: No
Normalization: None
Pathway: 0
Background N: default
Cover N: default
Delta N: default
Phi: default
Theta: default
Width: default
Height: default
Length: default
Position X: default
Position Y: default
Position Z: default
Neff: default

FDTD Launch Options
Temporal Excitation:
Type: Default (CW)
Wavelength: default
Ramp/Pulse Time: default
Ramp Off Time: default
Delay Time: default
Shutoff Time: default
Chirp Coefficient: default
Spatial Excitation:
Type: Field
Enclosed Launch: ☐
Current Direction: ☑ Auto
Direction Vector: defa defa defa

Figure 7: The Launch Parameters dialog.

In this case, the default launch settings are sufficient. Note the pertinent settings:

Option	Value	Description

Type *Slab Mode* This indicates that an analytical slab mode should be launched.

Mode 0 This indicates that the fundamental mode should be used.

Pathway 0 (unset) The pathway used to compute the properties of the launch field, including effective index, width, and position of the slab mode to be launched.

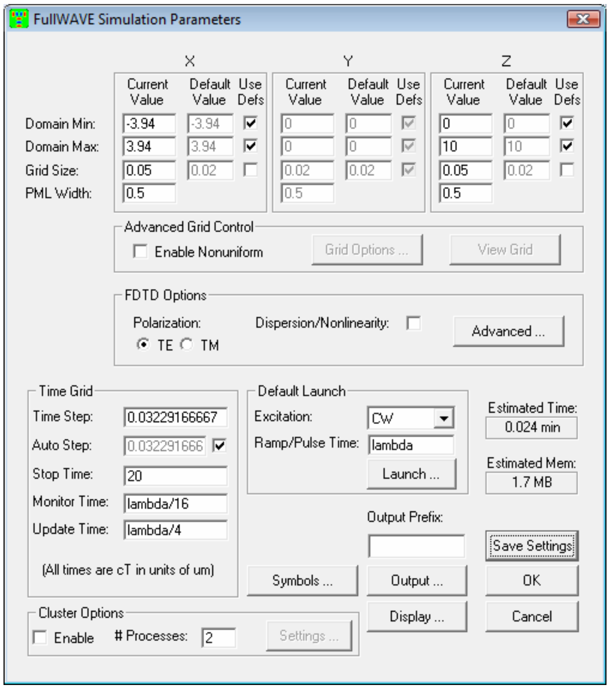
By default FullWAVE will choose the component with the lowest Z position in the design for the launch position. This behavior can be changed by defining a pathway for the launch, or by directly setting the field parameters (such as index, width, and position) listed. More information on pathways can be found in the CAD manual.

A complete description of the options in this dialog can be found in Chapter 4.

The Temporal Aspect:

The temporal aspect of the launch field describes how the field varies in time. For purposes of this tutorial, we will use a Continuous Wave, or CW, excitation. To set this, we will use the global setting in the FullWAVE Simulation Parameters dialog. This dialog can be opened by clicking the **Perform Simulation** button on the left toolbar in the CAD.

The excitation could also be set locally in the **FDTD Options** section of the Launch Parameters dialog, but it is better to use the global definition provided in the FullWAVE Simulation Parameters dialog. The local setting should only be used when multiple launch fields are present with different excitations.



FullWAVE Simulation Parameters

	X			Y			Z		
	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs
Domain Min:	-3.94	-3.94	☑	0	0	☑	0	0	☑
Domain Max:	3.94	3.94	☑	0	0	☑	10	10	☑
Grid Size:	0.05	0.02	□	0.02	0.02	☑	0.05	0.02	□
PML Width:	0.5			0.5			0.5		

Advanced Grid Control
□ Enable Nonuniform

FDTD Options
Polarization: ● TE ○ TM
Dispersion/Nonlinearity: □

Time Grid
Time Step: 0.03229166667
Auto Step: 0.032291666 ☑
Stop Time: 20
Monitor Time: lambda/16
Update Time: lambda/4
(All times are cT in units of um)

Default Launch
Excitation: CW
Ramp/Pulse Time: lambda

Estimated Time: 0.024 min
Estimated Mem: 1.7 MB

Output Prefix:

Cluster Options
□ Enable # Processes: 2

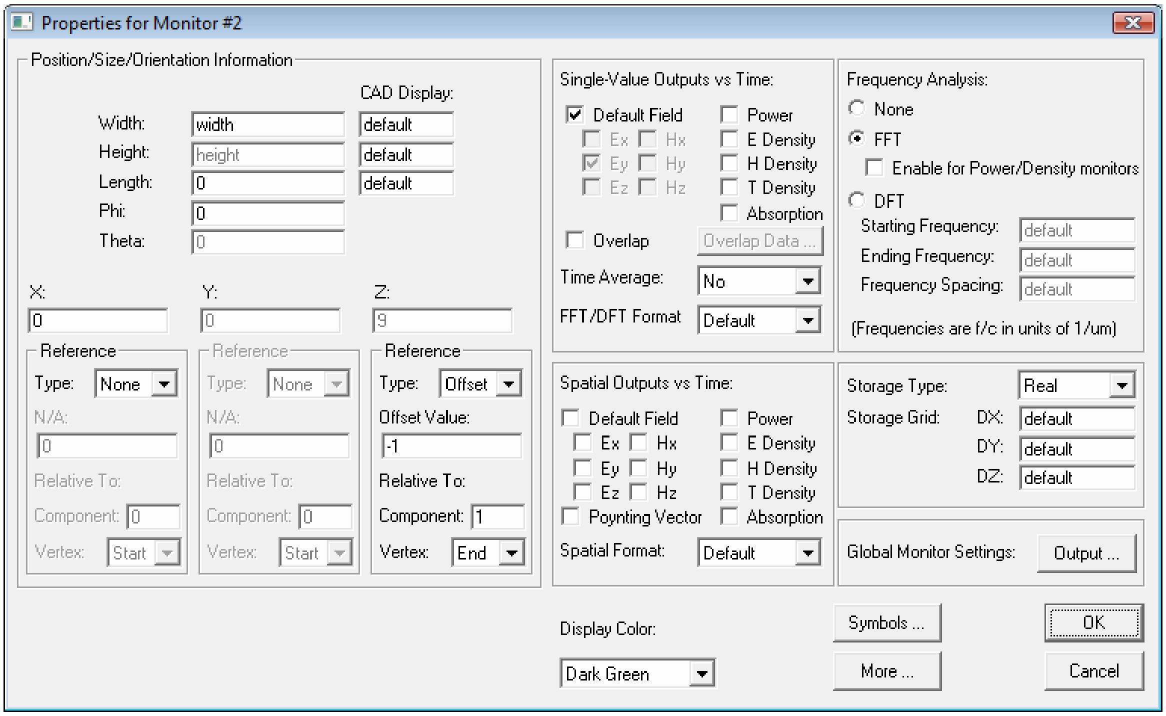
Figure 8: The FullWAVE Simulation Parameters dialog where the Default Launch settings are located. This dialog also controls the numeric simulation options discussed below.

Again the default setting of *CW* for **Excitation** is correct. Even though the setup is not complete, click **OK** to run the simulation. The FullWAVE simulation window will open showing the field propagating in the waveguide created. When the simulation is complete, close the window and return to the CAD window. Once the setup is complete the simulation will be performed again and the results saved.

Alternatively, you could also click the **Save Settings** button and then the Cancel button to return the CAD window without performing the simulation. This is useful when setting up the default launch settings or numeric simulation parameters without performing a simulation.

Adding a Monitor

FullWAVE monitors allow measurements to be made during a simulation for later viewing or processing. To add a monitor to the layout, use the Monitor drawing mode in the left CAD toolbar. After drawing the monitor, right click on the monitor to open its properties dialog as shown in Fig. 9.



Properties for Monitor #2

Position/Size/Orientation Information

		CAD Display:
Width:	width	default
Height:	height	default
Length:	0	default
Phi:	0	
Theta:	0	

X: 0 Y: 0 Z: 9

Reference	Reference	Reference
Type: None	Type: None	Type: Offset
N/A: 0	N/A: 0	Offset Value: -1
Relative To:	Relative To:	Relative To:
Component: 0	Component: 0	Component: 1
Vertex: Start	Vertex: Start	Vertex: End

Single-Value Outputs vs Time:
☑ Default Field □ Power
□ Ex □ Hx □ E Density
☑ Ey □ Hy □ H Density
□ Ez □ Hz □ T Density
□ Absorption
□ Overlap
Time Average: No
FFT/DFT Format: Default

Frequency Analysis:
○ None
● FFT
□ Enable for Power/Density monitors
○ DFT
Starting Frequency: default
Ending Frequency: default
Frequency Spacing: default
(Frequencies are f/c in units of 1/um)

Spatial Outputs vs Time:
□ Default Field □ Power
□ Ex □ Hx □ E Density
□ Ey □ Hy □ H Density
□ Ez □ Hz □ T Density
□ Poynting Vector □ Absorption
Spatial Format: Default

Storage Type: Real
Storage Grid: DX: default DY: default DZ: default

Global Monitor Settings:

Display Color: Dark Green

Figure 9: The properties dialog for a monitor.

This dialog allows for many different types of data to be saved, as well as frequency analysis to be performed.

More information on monitors can be found in Chapter 5 and the next tutorials.

For purposes of this tutorial, we will calculate the power in the waveguide slab mode at a specific position as a function of time. To do this, set the **Single Value Output** to *Overlap*. This will calculate the overlap of the propagating field with the launch field. We will measure the transmitted power at a position of 1 μm before the end of the waveguide. To do this, set the Z **Reference Type** to *Offset*, the Z **Offset Value** to `-1`, and the **Relative To** the ending vertex of **Component** `1`, as shown in Fig. 6.

Click **OK** to return to the CAD window. The monitor will appear as a green line at the end of the waveguide as shown in Fig. 10.

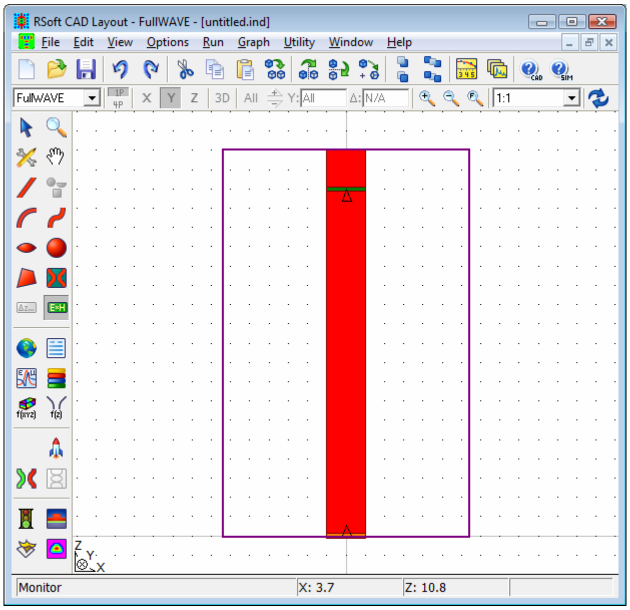

RSoft CAD Layout - FullWAVE - [untitled.ind]
File Edit View Options Run Graph Utility Window Help
FullWAVE
X Y Z 3D All Y: All Δ: N/A 1:1
Monitor
X: 3.7
Z: 10.8

Figure 10: The completed structure in the CAD with the monitor created.

Performing the Simulation

Once the layout and analysis setup is complete, we can initiate a simulation by clicking the **Perform Simulation** button on the left toolbar to open the FullWAVE Simulation Parameters dialog. This is the same dialog that was used to set the default launch parameters. This dialog allows the basic parameters required for a numerical simulation such as the spatial and temporal grid to be entered, and controls other important aspects of the simulation as well. Intelligent values for parameters such as grid sizes are automatically computed by the program based on the device being modeled, and are entered as defaults.

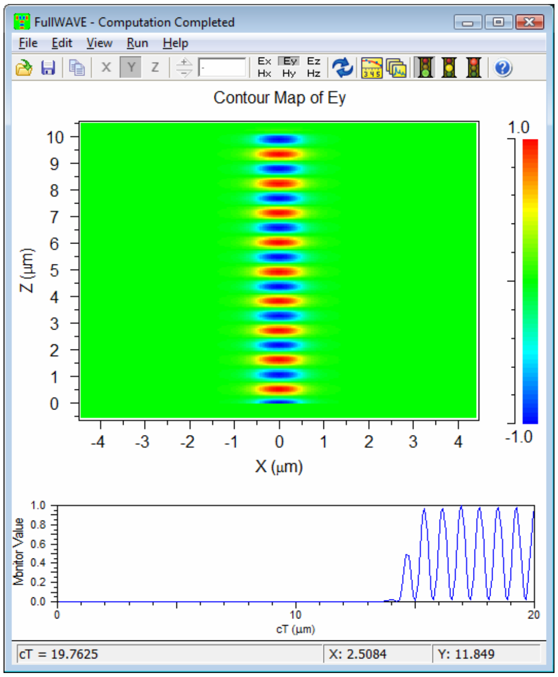
When performing an actual design, it is important to perform a convergence study with the numerical parameters to ensure that accurate results are obtained.

At this point you can choose to use the default values, or enter a different value for any parameter. For this tutorial, we will change the **Grid Size** values from their default of `0.02` μm to a slightly larger value of `0.05` μm so that the simulation will run faster. To do this, first locate the entry field corresponding to the current value of the grid size along X. Next, move the mouse cursor into this field and click on it in order to make it active. Now edit the field by deleting the value `0.02` and entering the value `0.05`. Repeat this for the grid size along Z.

In order to calculate the guided power, we need to run the simulation long enough so that the field reaches steady-state at the monitor. Set the **Stop Time** to `20`. Also, to save simulation results enter a prefix such as `test` in the **Output Prefix** field of the simulation parameters dialog and select the desired data files via the **Output…** button. This will save results to disk in files whose names start with the prefix entered. We will open the saved results later in the tutorial. The default values for the rest of the settings are sufficient for this tutorial. Once the numerical parameters are set, click the **OK** button to begin the simulation.

The Simulation Window

After a simulation is initiated, the computation begins in a new window separate from the main program, as shown in Fig. 8. The graphical display shows a color-coded contour plot of the optical field which is updated at the time interval specified in the **Update Time** field of the FullWAVE simulation parameters dialog as well as the time domain results of the monitors. The simulation program runs as a background task, so that you can switch away to another program, or back to the main program to edit another structure or begin another simulation. When the simulation is finished, the titlebar of the simulation program indicates "Computation Completed", and displays the coordinates of the cursor for reference.



cT = 19.7625 | X: 2.5084 | Y: 11.849

Figure 11: The separate simulation program, after simulating the waveguide structure. The graphical display shows a color-coded contour plot of the optical field.

Accessing Saved Data

All data saved by FullWAVE is in ASCII format, and so it is easy to access. For most output, two files are created: a raw data file, and a plotting command file that instructs the WinPLOT plotting program how to display the data. Generally, plotting command files have an extension that starts with the character '`p`'.

Opening Plots:

To view or print the saved simulation results, click on the **View Graphs** icon in the top toolbar of the CAD window (the yellow graph), and select the desired graph from the dialog that is presented. In this case, several plotting files with the **Output Prefix** entered should be present, including `test.ptm`. This file contains the output from the monitor that we created. Choose this file and click **OK** to open it.

A WinPLOT window will open which will show the monitor data. The mouse can be used to zoom in the graph to areas of interest. To zoom out again, simply right-click. Note that the data saved is normalized to the input power, and so a value of 1 corresponds to perfect transmission. Furthermore, the stored data is in field unit; however, the data is squared before display. See the next sections for more details.

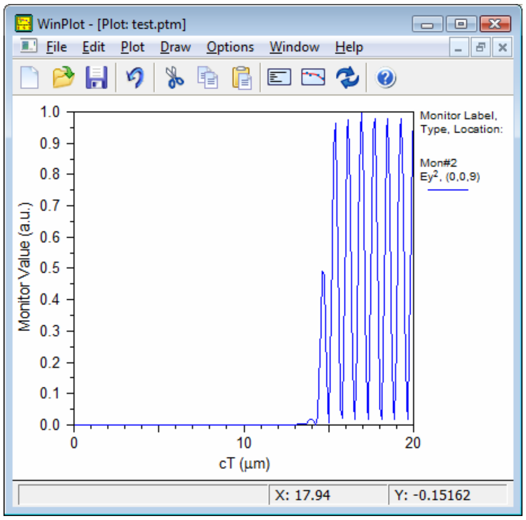
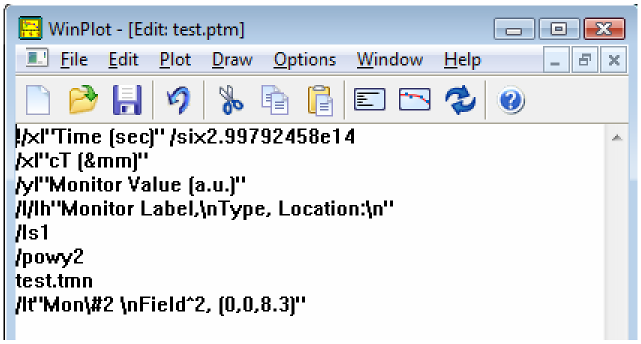


Figure 12: A WinPLOT window which shows the monitor result for this simulation.

Viewing the Plotting Commands:

The plotting command file `test.ptm` that was opened in WinPLOT is actually just a text file that contains plotting commands. To view this file, either open it in a text editor, or click the **View Editor** button in the WinPLOT window. The contents of this file are shown in Fig. 13. These commands indicate the axis labels to use, the legend text, as well as the raw data file `test.tmn`. The command '`/powy2`' squares the date before display. Descriptions for these and other plotting commands can be found in the WinPLOT manual. Clicking the **View Plot** button returns the view to the plot.

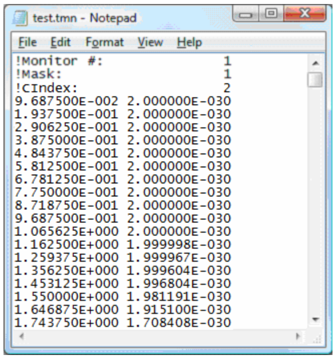


```
/xl"Time (sec)" /six2.99792458e14
/xl"cT (&mm)"
/yl"Monitor Value (a.u.)"
/l/lh"Monitor Label,\nType, Location:\n"
/ls1
/powy2
test.tmn
/lt"Mon\#2 \nField^2, (0,0,8.3)"
```

Figure 13: The plotting commands contained in the file `test.ptm` used to display the monitor data stored in the file `test.tmn`.

Viewing the Raw Data:

The raw data file `test.tmn` can be opened in any text editor, and contains two columns: the first shows the time coordinate in units of µm (cT); the second shows the monitor result.



```
!Monitor #:                 1
!Mask:                      1
!CIndex:                    2
9.687500E-002 2.000000E-030
1.937500E-001 2.000000E-030
2.906250E-001 2.000000E-030
3.875000E-001 2.000000E-030
4.843750E-001 2.000000E-030
5.812500E-001 2.000000E-030
6.781250E-001 2.000000E-030
7.750000E-001 2.000000E-030
8.718750E-001 2.000000E-030
9.687500E-001 2.000000E-030
1.065625E+000 2.000000E-030
1.162500E+000 1.999998E-030
1.259375E+000 1.999967E-030
1.356250E+000 1.999604E-030
1.453125E+000 1.996804E-030
1.550000E+000 1.981191E-030
1.646875E+000 1.915100E-030
1.743750E+000 1.708408E-030
```

Figure 14: The raw data produced by this simulation.

Viewing other Types of Output:

Most other types of output produced by FullWAVE are parallel to the monitor output discussed in this section. A detailed description of a FullWAVE monitor can be found in Chapter 5, and examples of several output types are given in the next tutorials.

Areas for Further Exploration:

Several areas to explore include:

- *Viewing Different Field Components*

  Different field components can be viewed by using the buttons in the simulation toolbar. Note that since this is a 2D simulation, not all components will be available.

- *Producing Time-Averaged Output:*

In the above example we used an instantaneous monitor. To see a time averaged result, go to the monitor properties box and set **Time Average** to *Yes*. A moving-window average over the last period is used will be averaged over the last period. This is useful when studying the steady-state performance of a system as the monitor results. If both averaged and non-averaged data is needed, create two monitors at the same location.

- *Showing the Structure Outline:*

  It can be useful to display the structure outline in the simulation window. To do this, open the FullWAVE Simulation Parameters dialog and click the **Display...** button. Set the **Outline Color** to *Default* and click **OK** twice to perform the simulation. The structure outline will now be displayed in the simulation window.

- *Changing the Display Aspect Ratio During a Simulation:*

  To change the display aspect ration during a simulation, open the FullWAVE Simulation Parameters dialog and click the **Display...** button. Set the **Plot Aspect Ratio** to the desired value. The value of `default` maximizes the size of the simulation display.

- *Changing the Color Scale*

  A color scale is used to map the field values displayed during a simulation to particular colors for display. To change the color scale used, open the FullWAVE Simulation Parameters dialog and set the **Color Scale** field to the desired file. The **Scale Files...** button can be used to locate the color scale needed. For information regarding the use and creation of color scale files can be found in an Appendix in the CAD manual.

Basic Tutorial 2: Basic 3D Simulation

This tutorial illustrates a basic 3D FullWAVE simulation. Our goal will be to layout and simulate the propagation of a mode at 1.55 µm along a fiber with a cladding index of 1.0, a core index of 1.5, and a diameter of 1 µm. We will calculate the transmitted power along 2 µm of this fiber.

This tutorial will introduce basic concepts for a 3D simulation. Familiarity with the concepts presented in Basic Tutorial 1 are assumed. If you have not already read through Basic Tutorial 1, please do so now.

3D Specific CAD Options

We begin by opening up the RSoft CAD window.

A 3D simulation not only requires the settings for a 2D simulation as well as several new settings such as **3D Structure Type** and **Component Height**. Also, note that the definition of polarization is different.

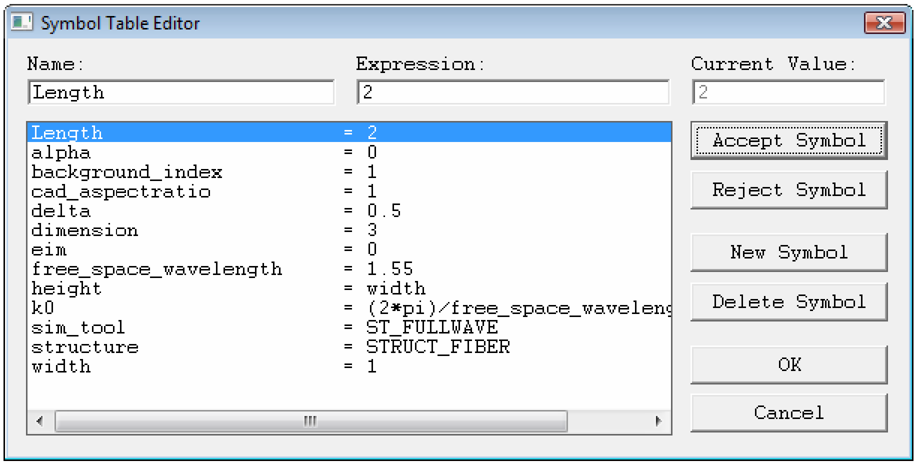
Click the **New Circuit** icon in the top toolbar and make the following settings:

Parameter	Value	Description
Simulation Tool	*FullWAVE*	Sets FullWAVE as the simulation engine to be used.
Model Dimension	*3D*	Creates a 3D simulation.
Free Space Wavelength	`1.55`	Sets the global wavelength [µm] to be used for the simulation. For a CW simulation, this represents the driving wavelength, and for a Pulsed simulation this represents the center wavelength.
Background Index	`1.0`	Represents the real refractive index of the background material where no structure is defined.
Index Difference	`0.5`	Represents the default difference between a component and the background material. In this case, our structure will have a default index of 1.5; each component can have a unique value if desired.
3D Structure Type	*Fiber*	Sets the profile to be used for 3D components. The choice *Fiber* indicates that, by default, a component should have a circular cross section with a horizontal and vertical diameter given by the **Component Width** and **Component Height** fields respectively. By setting these fields equal, a circular cross-section is achieved.
Component Width	`1.0`	This sets the default width of a component [µm]; each component can have a unique value if desired.
Component Height	`width`	This set the default height of a component in [µm]; each component can have a unique value if desired. The variable `width` is the built-in variable which corresponds to the value of the **Component Width** option.

Once these settings are made, click **OK** to continue.

Creating Variables

For this design we will create a variable which sets the length of our fiber segment. Click the Edit Symbols button on the left toolbar and create a symbol `Length` with a value of 2 µm. The symbol table should appear as shown in Fig. 12.



Name: Length
Expression: 2
Current Value: 2

```
Length                    = 2
alpha                     = 0
background_index          = 1
cad_aspectratio           = 1
delta                     = 0.5
dimension                 = 3
eim                       = 0
free_space_wavelength     = 1.55
height                    = width
k0                        = (2*pi)/free_space_waveleng
sim_tool                  = ST_FULLWAVE
structure                 = STRUCT_FIBER
width                     = 1
```

Figure 1: The Symbol Table Editor. The variable Length has been created with a value of 2 µm.

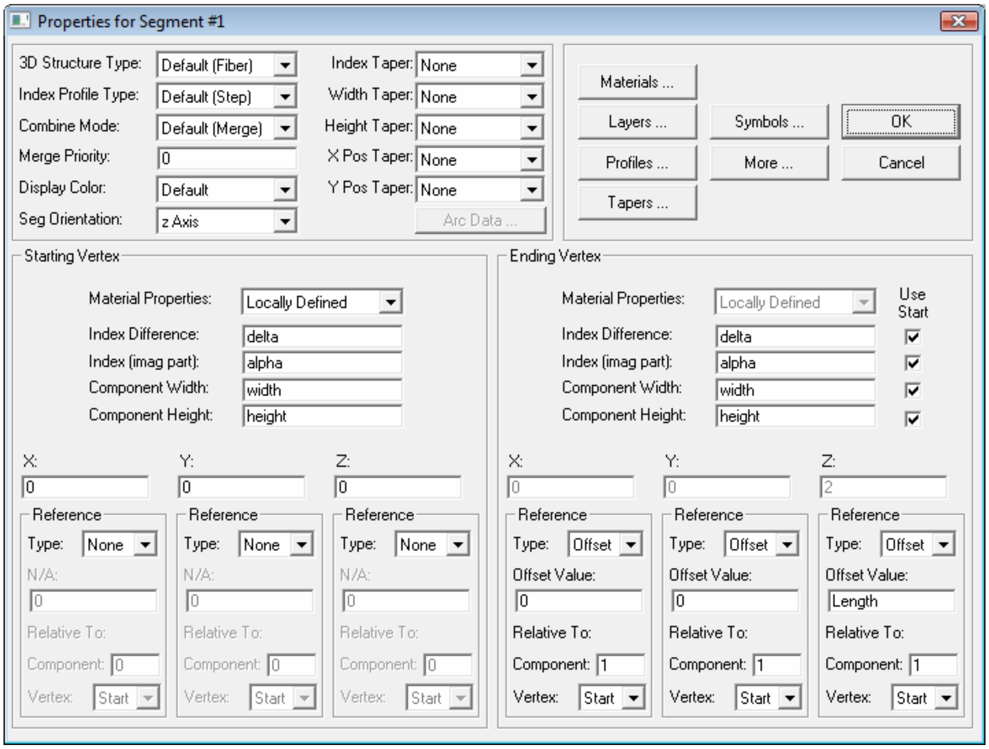
Any variable in the symbol table can be set to be a function of any other variable. For example, the variable `k0` is set to a value of `(2*pi)*free_space_wavelength` corresponding to the magnitude of the wavenumber **k** in free space at the wavelength value chosen. A complete list of built-in functions and constants for the CAD can be found in an Appendix in the CAD manual, and built-in constants for FullWAVE can be found in Appendix C of this manual.

Adding the Fiber Segment

To add the fiber segment, make sure you are in **Segment Mode**. The waveguide is drawn in the same way as a 2D simulation.

This segment will have a circular cross-section because the **3D Structure Type** is set to *Fiber*. More information about selecting a segment's profile, including creating a user-defined profile, can be found in the CAD manual.

Once the fiber segment has been created, we can set its length to the symbol `Length` (2 µm) as desired. Right click on the segment to access its Segment Properties dialog, and set the starting vertex coordinates to (`0,0`) and the ending vertex to be *Offset* from the starting vertex by a value of (`0, Length`). When done, the dialog should appear as shown in Fig. 2.



3D Structure Type: Default (Fiber)
Index Profile Type: Default (Step)
Combine Mode: Default (Merge)
Merge Priority: 0
Display Color: Default
Seg Orientation: z Axis

Index Taper: None
Width Taper: None
Height Taper: None
X Pos Taper: None
Y Pos Taper: None

Starting Vertex

Material Properties: Locally Defined
Index Difference: delta
Index (imag part): alpha
Component Width: width
Component Height: height

X: 0 Y: 0 Z: 0

	X	Y	Z
Reference Type:	None	None	None
N/A:	0	0	0
Relative To:			
Component:	0	0	0
Vertex:	Start	Start	Start

Ending Vertex

Material Properties: Locally Defined — Use Start
Index Difference: delta ☑
Index (imag part): alpha ☑
Component Width: width ☑
Component Height: height ☑

X: 0 Y: 0 Z: 2

	X	Y	Z
Reference Type:	Offset	Offset	Offset
Offset Value:	0	0	Length
Relative To:			
Component:	1	1	1
Vertex:	Start	Start	Start

Figure 2: The Segment Properties dialog box for the fiber segment in this design.

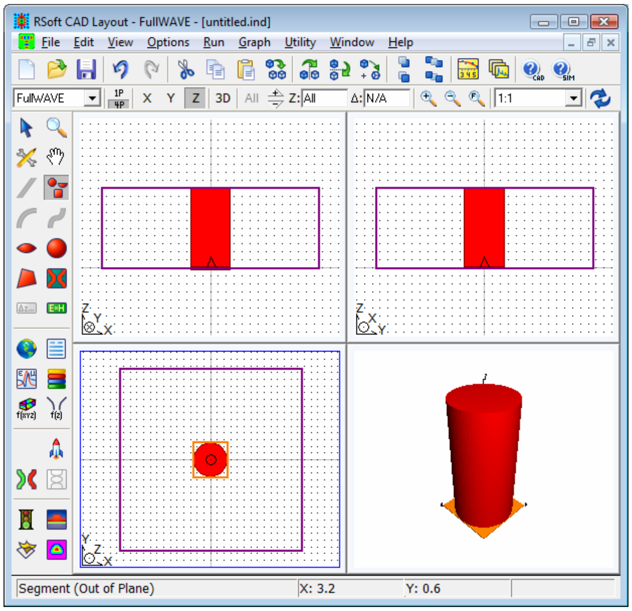


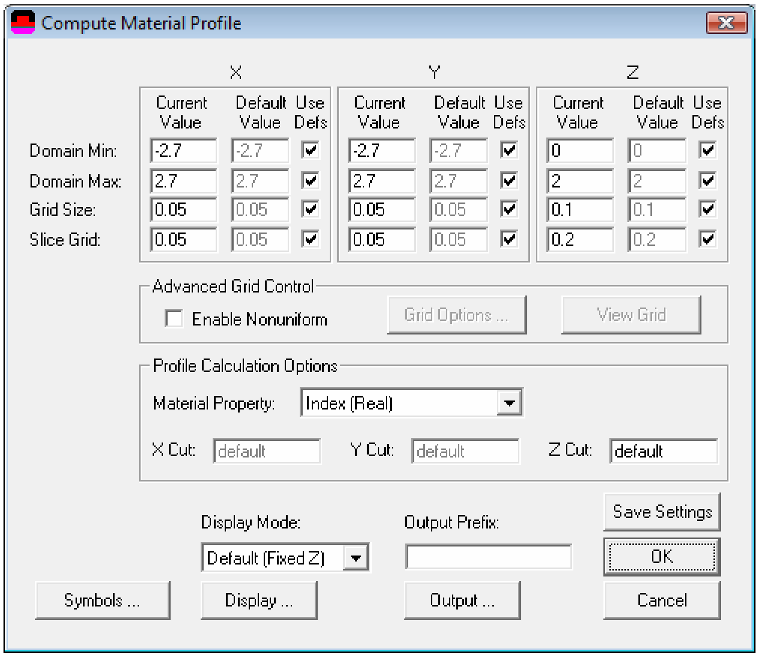
Figure 3: The CAD window in multi-pane mode.

Viewing the Structure in 3D

The default view of the structure in the CAD is along the Y axis (the XZ plane). However, the structure can be viewed along the X and Z axes individually, as well as along all three axes. To change this view, click the X, Y, and/or Z buttons in the view toolbar in the CAD. To enter multi-pane mode, click the 4P button; to return to single-pane mode, click the 1P button. More information about these views can be found in the CAD manual.

Checking the Index Profile

A good habit to learn is to check the index profile of your design in order to ensure that no mistakes have been made defining the geometry and refractive index. If any mistakes have been made, the results of any simulations cannot be trusted. The index profile can be calculated via the Compute Index Profile dialog shown in Fig. 4.



Compute Material Profile

	X			Y			Z		
	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs
Domain Min:	-2.7	-2.7	☑	-2.7	-2.7	☑	0	0	☑
Domain Max:	2.7	2.7	☑	2.7	2.7	☑	2	2	☑
Grid Size:	0.05	0.05	☑	0.05	0.05	☑	0.1	0.1	☑
Slice Grid:	0.05	0.05	☑	0.05	0.05	☑	0.2	0.2	☑

Advanced Grid Control

☐ Enable Nonuniform

Profile Calculation Options

Material Property: Index (Real)

X Cut: default   Y Cut: default   Z Cut: default

Display Mode: Default (Fixed Z)

Output Prefix:

Figure 4: The Compute Index Profile dialog.

To open this dialog, click the **Display Material Profile** button on the left toolbar in the CAD.

A complete description of this dialog can be found in the CAD manual.

For purposes of this tutorial, the default settings are sufficient. However, note the following settings:

Parameter	Description
Grid Size	These fields set the calculation grid used for the simulation.
Slice Step	These fields set the grid on which the index information is saved and displayed. For most calculations, it is best to set the slice grid sizes equal to the calculation grid sizes in order to best see the index stricture.
Display Mode	This field sets the display type to be used for the index calculation. Setting this to *Fixed at Z-Min* computes the cross-sectional index profile at the Z value specified in the Z **Domain Min** field.
Output Prefix	This sets the output prefix to be used when saving data. The index profile is

saved in simple ASCII format and can be opened with any text editor.

Click **OK** to start the computation. The index profile will be displayed as shown in Fig. 5.

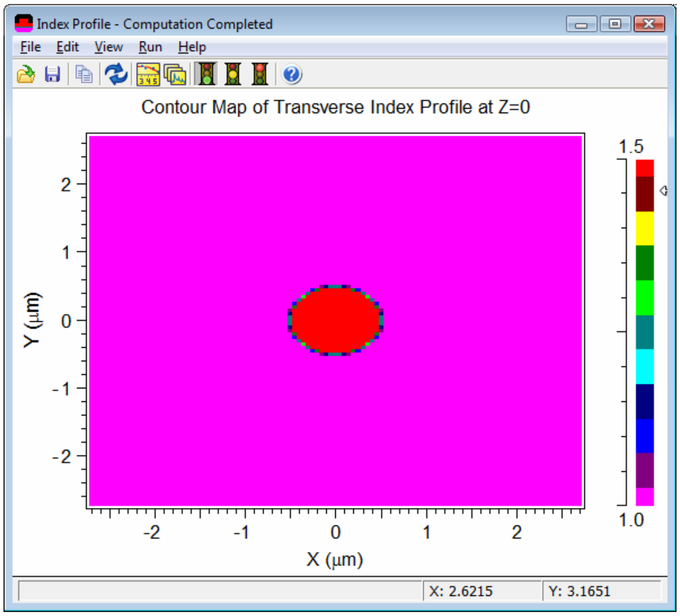

Index Profile - Computation Completed
File Edit View Run Help
Contour Map of Transverse Index Profile at Z=0
Y (µm)
2
1
0
-1
-2
-2
-1
0
1
2
X (µm)
1.5
1.0
X: 2.6215
Y: 3.1651

Figure 5: The computed index profile. The fiber cross-section appears elliptical because the aspect ratio of the plot is not set to 1. This can be changed via the **Plot Aspect Ratio** option in the Display Options dialog.

Choosing the Excitation

The launch settings for this simulation are similar to those described in Tutorial 1 with one change: instead of an analytical slab mode we will launch an analytical fiber mode. This change is automatically made when the **Model Dimension** is changed from *2D* to *3D*, but if you want to open the Launch Parameters dialog and see this change, you may.

Adding a Monitor

In order to calculate the transmitted power along this fiber, we will use an overlap monitor that takes measures the overlap between the propagating field and the analytical fiber mode that we are using as a launch field. The creation and setup of this monitor is very similar to the monitor used in Tutorial 1. Add a monitor to the layout and set the **Single Value Output** to *Overlap* and set its position to be 0.1 µm before the end of the fiber.

Performing the Simulation

To initiate a simulation, click the **Perform Simulation** icon in the left toolbar. Like Tutorial 1, we are going to change the default **Grid Size** of 0.02 µm to 0.05 µm so the simulation will run faster. This should be done for the X, Y, and Z coordinates.

When performing an actual design, it is important to perform a convergence study with the numerical parameters to ensure that accurate results are obtained.

The rest of the parameters are fine left to their default values. Once the numerical parameters are set, enter an **Output Prefix** of `test` and click **OK** to begin the simulation.

Accessing Saved Data

The data saved from this simulation can be opened and viewed as described in Tutorial 1.

Areas for Further Exploration

Several areas to explore further include:

- *Experiment with Different Index Profile Output Options:*

  This tutorial used a **Display Type** of *Fixed at Z-Min* for the index computation. Other display types such as different cross-sections at arbitrary positions and 3D models can be used as well. Try computing the index with different modes and see what type of output is created. The CAD manual details the use of these modes.

- *Change Field Component and Cross-Section Displayed During a Simulation*

  The default field component displayed during a simulation and the type and position of the cross-section shown can be changed either via the controls in the Display Options dialog, or via the toolbar in the simulation window.

- *Use the DataBROWSER to Quickly View All Output Data*

  The RSoft DataBROWSER allows the user to quickly sort through the output files from a simulation. It can be accessed via the **Launch DataBROWSER** button on the top toolbar in the CAD. Please consult the DataBROWSER manual for more information.

10

Advanced Tutorials

This chapter contains several advanced tutorial examples which illustrate the use of the FullWAVE simulation engine. Corresponding to each example there is one or more `.ind` and/or data files, which are located in the `<rsoft_dir>\examples\FullWAVE\tutorial\` subdirectory.

Unlike the basic tutorials in Chapter 9, these tutorials, except where noted, do not need to be performed in any order. However, it is recommended that you perform the basic tutorials in Chapter 9 before attempting to work through these tutorials as they explain basic concepts in more detail.

While the following examples attempt to cover the key aspects of the software, the breadth of the package allows enormous flexibility in both the number of applications that can be covered as well as the approach used to tackle a given application. If you do not find your application adequately addressed by the following examples, please contact RSoft by emailing (info@rsoftdesign.com) or faxing (914-923-2169) a sketch of the problem and we will assist you in finding a solution if possible.

Tutorial 1: Ring Resonator

This tutorial discusses the creation and analysis of a ring resonator, a device that acts as a high Q filter for wavelength filtering applications. The setup and layout of the structure will be illustrated, and both pulsed and CW simulations will be performed to study the frequency response and steady-state operation of the device.

Device Layout:

The ring resonator consists of 0.2 μm wide waveguides with and index of 3 embedded in air. For purposes of this tutorial, we will study the resonances around a wavelength of 2 μm.
Open the RSoft CAD and click the **New Circuit** icon in the top toolbar. For this tutorial, we will use the following settings:

Parameter	Value	Description
Simulation Tool	*FullWAVE*	Sets FullWAVE as the simulation engine to be used.
Model Dimension	*2D*	Creates a 2D simulation.
Free Space Wavelength	`2`	Sets the global wavelength [μm] for the simulation. For CW simulations, this will be the driving wavelength, for pulsed simulations, this will be the central wavelength.
Background Index	`1.0`	Sets the refractive index of the background region.
Index Difference	`2`	Sets the default index difference between the waveguides and the background material resulting in an index of 3.
Component Width	`0.2`	Sets the waveguide width.

Once these settings have been made click OK to create the new file.

Defining Variables

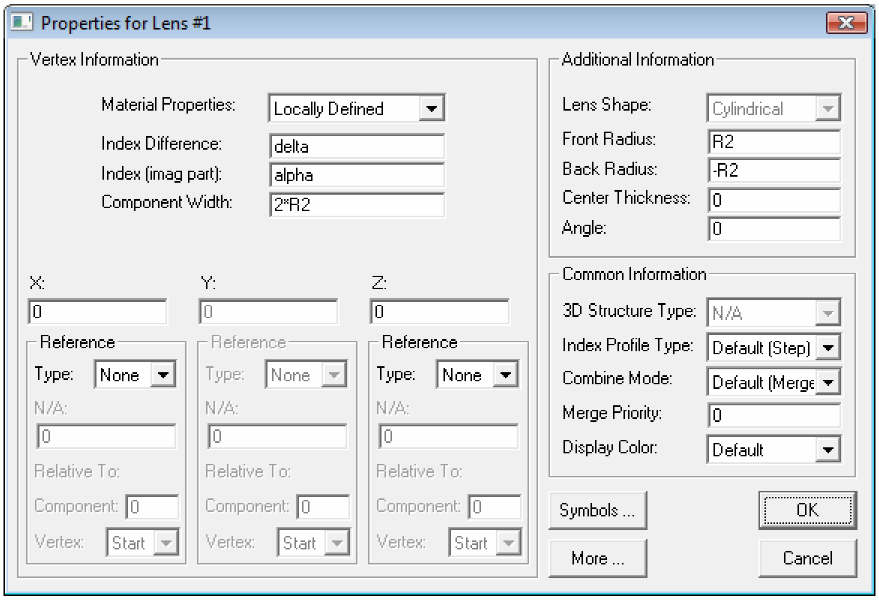
It will be useful to define several variables which will be used to define various structural properties. Click on the **Edit Symbols** icon and define the following variables:

Name	Value	Description
`Gap`	`0.2`	The distance between the ring and bus waveguides.
`L`	`0.5`	The length the bus waveguides extend past the ring.
`R`	`1.7`	The radius of the ring.
`R1`	`R-width/2`	The inside radius of the ring.
`R2`	`R+with/2`	The outside radius of the ring.

Using variables allows the design to easily change after it has been created.

Drawing the Structure

We will start by drawing a circular waveguide for the ring. We will first draw a disc with a radius equal the outer radius of the ring, and then 'carve' a hole in it using another disc with a radius equal to the inner radius of the ring. Select the Lens Mode from the left CAD toolbar and draw a lens near (X,Z) = (0,0). Right click on the lens object to open the Lens Properties dialog. Change the **Component Width** to `2*R2`, **Front Radius** to `R2` and **Back Radius** to `-R2`. When done, press **OK**.



Properties for Lens #1

Vertex Information

Material Properties: Locally Defined

Index Difference: delta

Index (imag part): alpha

Component Width: 2*R2

X: 0

Y: 0

Z: 0

Reference
Type: None
N/A: 0
Relative To:
Component: 0
Vertex: Start

Reference
Type: None
N/A: 0
Relative To:
Component: 0
Vertex: Start

Reference
Type: None
N/A: 0
Relative To:
Component: 0
Vertex: Start

Additional Information

Lens Shape: Cylindrical

Front Radius: R2

Back Radius: -R2

Center Thickness: 0

Angle: 0

Common Information

3D Structure Type: N/A

Index Profile Type: Default (Step)

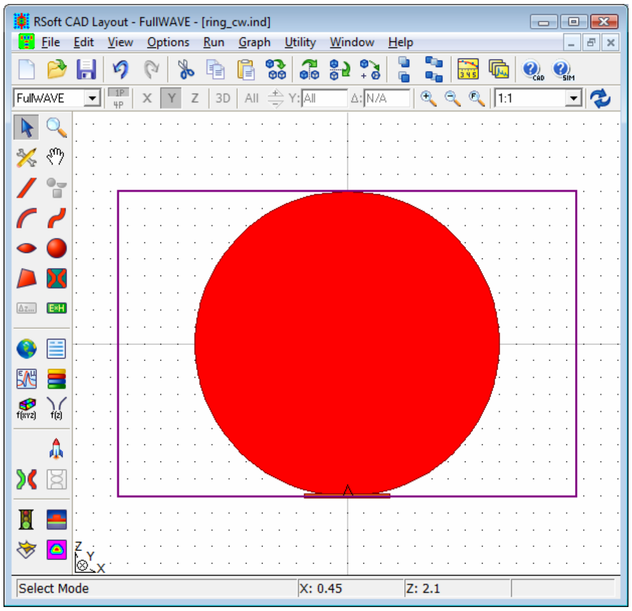
Combine Mode: Default (Merge

Merge Priority: 0

Display Color: Default

Figure 1: Properties for Lens #1. This lens will form the outer portion of the circular waveguide in the ring resonator.

After clicking **OK**, the circular lens will be displayed in the CAD window as shown in Fig 2. You may have to click the View/Full button in the view toolbar so that the display is the same.



RSoft CAD Layout - FullWAVE - [ring_cw.ind]

Select Mode | X: 0.45 | Z: 2.1

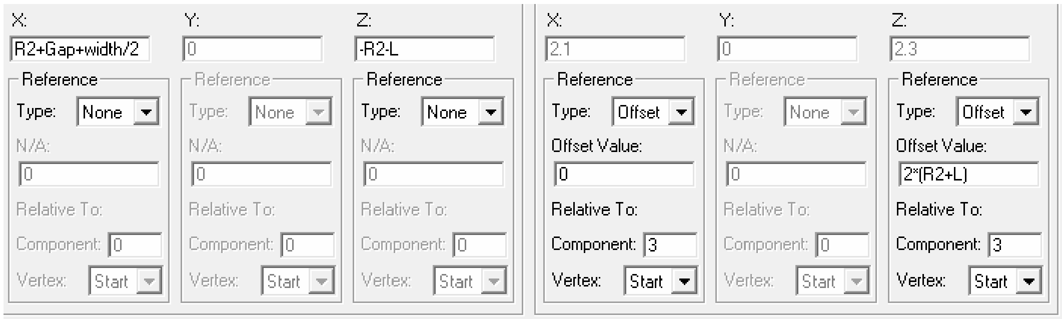
Figure 2: The lens as seen in the CAD window.

To draw the inner disc we can simply duplicate our current disc and change its properties. Click the **Select Mode** icon in the left toolbar, click on the lens to select it, and click the **Duplicate Selection** icon in the top toolbar. Right click on the lens and note that the Lens Properties dialog box shows that this is Lens #2 in the title. Change the **Component Width** to `2*R1`, **Front Radius** to `R1`, **Back Radius** to `-R1`, the **Index Difference** to be `0`, and the **Display Color** to *Yellow*. Also set the **Merge Priority** to `1` which indicates that the index of the smaller circle will have priority over the index of the larger circle.

When overlapping segments are drawn, the CAD needs to decide which refractive index to use. The default behavior is to use the higher absolute index which works well in most cases. In this case, however, we want the lower index (air) to be used. To force this behavior, we give the inner lens a higher **Merge Priority**. See the CAD manual for more information about overlapping segments.

Click **OK** to return to the CAD window.

To draw the bus waveguides, click the **Segment Mode** icon in the left toolbar and draw a segment to the right of the ring. Right-click on the new segment to set its properties directly: set the **Starting Vertex X** to `R2+Gap+width/2`, the **Starting Vertex Z** to `-R2-L`, and the **Ending Vertex Z** to *Offset* from the starting vertex of **Component** `3` with an **Offset Value** equal to `2*(R2+L)` to set the length of the segment. These values set the location of this waveguide offset from the ring by the distance specified by `Gap` in the symbol table. Click **OK** to return to the CAD window.



X:	Y:	Z:	X:	Y:	Z:
R2+Gap+width/2	0	-R2-L	2.1	0	2.3
Reference	Reference	Reference	Reference	Reference	Reference
Type: None	Type: None	Type: None	Type: Offset	Type: None	Type: Offset
N/A:	N/A:	N/A:	Offset Value:	N/A:	Offset Value:
0	0	0	0	0	2*(R2+L)
Relative To:	Relative To:	Relative To:	Relative To:	Relative To:	Relative To:
Component: 0	Component: 0	Component: 0	Component: 3	Component: 0	Component: 3
Vertex: Start	Vertex: Start	Vertex: Start	Vertex: Start	Vertex: Start	Vertex: Start

Figure 3: The geometry settings for the first bus waveguide.

To create the second bus waveguide, we will duplicate the first one. Click on the first bus waveguide and click the **Duplicate Segment** button in the top toolbar. Right-click on the new segment and change the **Starting Vertex X** value to `-R2-Gap-width/2` effectively moving the second waveguide to the other side of the ring. Click **OK** to see the completed structure.

Checking the Index Profile

It is always recommended to check the index profile of any structure to ensure the layout is correct. Click the **Display Material Profile** button in the left toolbar and set the following parameters:

Parameter	**Value**
Display Mode	*ContourMap(XY)*
X **Domain Min**	`-2.7`
X **Domain Max**	`2.7`
Z **Domain Min**	`-2.28`
Z **Domain Max**	`2.28`
Grid Size (X & Y)	`0.02`
Slice Grid (X & Y)	`0.02`

The Slice Grid setting allows a subset of the index information to be displayed, but in this case we want to display all the index information. Click **OK** to compute the index profile.

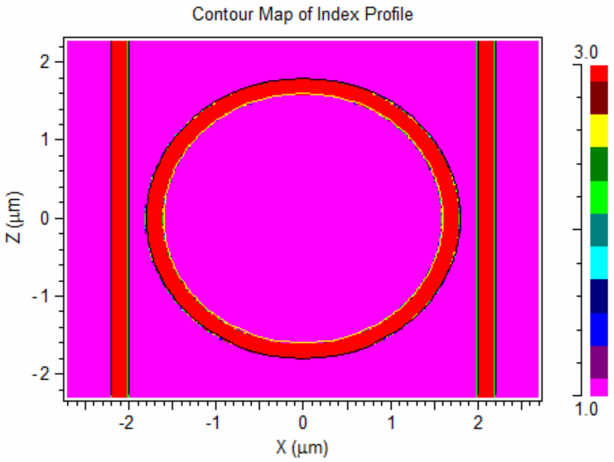


Figure 5: The computed index profile. The effect of the second ring's higher priority is plainly visible.

Launch Field

This section will set the spatial aspect of the launch. The temporal aspect will be set later in the simulation sections.

In this example we will launch the fundamental slab mode of the bus waveguide into the input port (bottom left waveguide). To specify which waveguide to launch into, we need to define a pathway. Click the **Edit Pathways** button in the left toolbar and click **New Pathway**. Click on the left bus waveguide and it will turn bright green. Note that we have created Pathway #1 and click **OK** to return to the CAD window. Click the **Edit Launch Field** button in the left toolbar and note that the **Launch Pathway** is set to `1`. The **Launch Type** is set to *Slab Mode* as we want by default, and so click **OK** to return to the CAD window.

Adding Monitors

The next step is to insert monitors into the design to make measurements during simulations for analysis purposes. We will create two monitors: one to measure the transmitted field at the top-left port and one to measure the dropped field at the bottom right port.

Click the **Monitor** button in the left toolbar and add a monitor near the top of the left waveguide by clicking on the waveguide. Right-click on the monitor to set its properties directly and change the **X** and **Z** vertex **Reference Type** to *Offset*, **Z Offset Value** to `-0.5`, **X** and **Z** vertex **Relative To** the ending vertex of **Component** `4`. Finally, select only *Overlap* under **Single-Value Output**, set **Time Average** to *Auto*, and set **Frequency Analysis** to *FFT*. Click OK to return to the CAD window.

These settings position the monitor 0.5 µm from the end of the waveguide regardless of the length of the waveguide. For extra credit, open the Symbol Table and change the value of the variable `Length` to a larger value. If everything is set up correctly, the length of the bus waveguides will increase and the monitor will ‘move’ with the waveguide. Revert the changes made to `Length`.

Since this waveguide only supports a single mode and we will be launching this mode, the Overlap setting will measure the power transmitted out of the waveguide. By setting the time average to *Auto* and selecting FFT output the monitor will automatically time-average the results and not perform an FFT when using a CW excitation and not time-average the results and perform an FFT when using a pulsed simulation.

To create the second monitor, duplicate the first monitor as we did before using the Duplicate Selection button and then right-click on the new monitor to set its properties directly. Change the vertex paramters such that it is offset from the starting vertex of component 3 by a value of (X,Z) = (`0`, `0.5`). This places the monitor at the bottom of the right waveguide to measure the field which exits the simulation domain in this waveguide. Click **OK** to return to the CAD window.

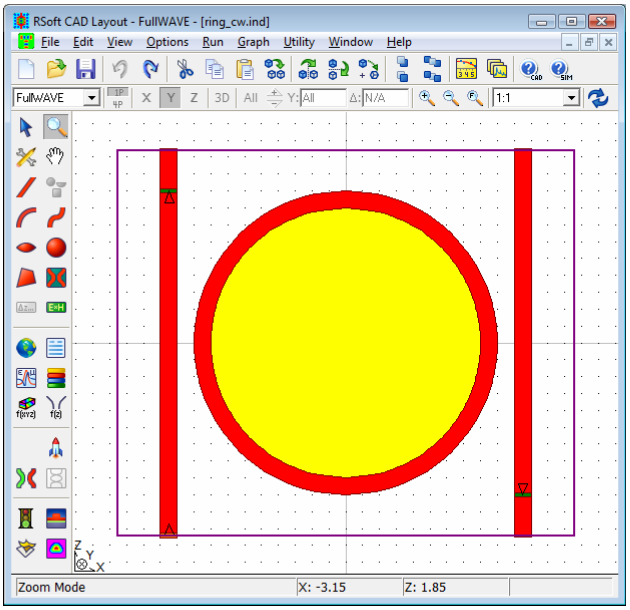


Figure 6: The completed ring resonator structure with monitors.

Simulation: Pulsed Excitation

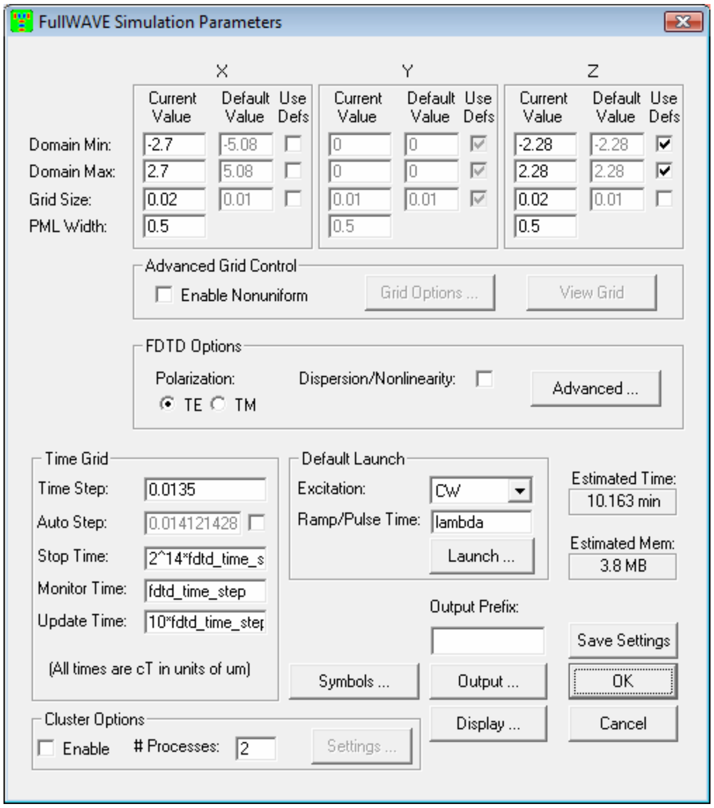
The wavelength/frequency spectrum will first be calculated using a pulsed simulation and then the steady-state operation at a resonance will be studied using a CW simulation.

Click the **Perform Simulation** button in the left toolbar and set the following simulation parameters:

Parameter	**Value**	**Comments**
Time Step	`0.0135`	The time step has to be small enough to satisfy the Courant condition.
Stop Time	`2^13*fdtd_time_step`	The stop time needs to be long enough to record the full time response of the system. This value is a good starting poing, but we will find out later it will not be long

		enough for good frequency resolution.
Update Time	`10*fdtd_time_step`	This sets the rate at which the graphical display is updated.
Monitor time	`fdtd_time_step`	This indicates that the monitors should record data at every time step.
Excitation	*Pulse*	Sets a pulsed temporal excitation.

Next, click the **Output…** button to open the FDTD Output Options dialog. Make sure that **Temporal Output**, **Wavelength Output**, and **Frequency Output** are all selected. Click **OK** to close this window and click the Display… button top open the FDTD Display Options dialog. Set the **Outline Color** to *Black* so that we can easily see the circuit outline in the simulation window. Click **OK**, enter an **Output Prefix** like `ring_pulse1` and click **OK** to start the simulation.



FullWAVE Simulation Parameters

	X Current Value	X Default Value	X Use Defs	Y Current Value	Y Default Value	Y Use Defs	Z Current Value	Z Default Value	Z Use Defs
Domain Min:	-2.7	-5.08	□	0	0	☑	-2.28	-2.28	☑
Domain Max:	2.7	5.08	□	0	0	☑	2.28	2.28	☑
Grid Size:	0.02	0.01	□	0.01	0.01	☑	0.02	0.01	□
PML Width:	0.5			0.5			0.5		

Advanced Grid Control
□ Enable Nonuniform

FDTD Options
Polarization: ● TE ○ TM
Dispersion/Nonlinearity: □

Time Grid
Time Step: 0.0135
Auto Step: 0.014121428 □
Stop Time: 2^14*fdtd_time_s
Monitor Time: fdtd_time_step
Update Time: 10*fdtd_time_step
(All times are cT in units of um)

Default Launch
Excitation: CW
Ramp/Pulse Time: lambda

Estimated Time: 10.163 min
Estimated Mem: 3.8 MB

Output Prefix:

Cluster Options
□ Enable # Processes: 2

Figure 7: The FullWAVE Simulation Parameters dialog.

The simulation will then run until the **Stop Time** is reached. When the simulation is complete, the pulse will have almost completely left the simulation domain.

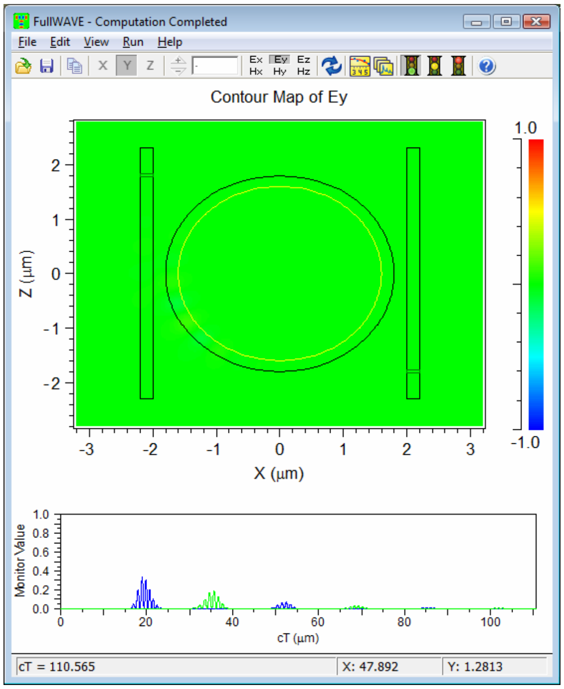


Figure 8: The completed FDTD simulation window. The bottom graphs displays the temporal monitor results.

To view the wavelength spectrum, click on the **View Graphs** icon and select `ring_pulse1.pwm`.

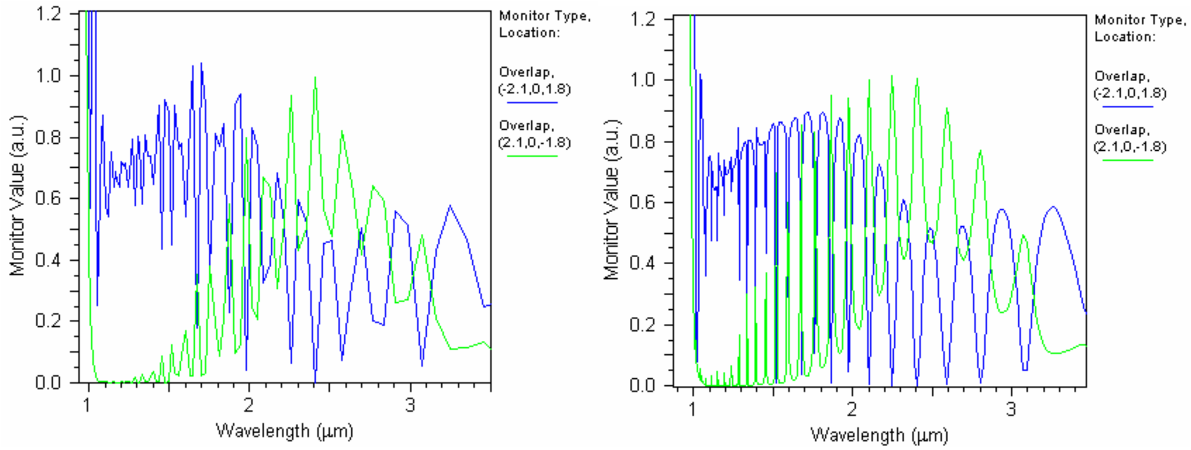


Figure 9: a) The wavelength spectrum from the pulsed simulation of the ring resonator. The green line represents the through port, and the blue line represents the drop port. b) The same wavelength spectrum with a longer stop time.

Figure 9 shows the Fourier transform of the time response of this system, which is the wavelength response of the circuit. Note that the frequency spectrum can be found in the file `ring_pulse1.pfm`. This graph has a very poor frequency resolution due to the short **Stop Time**. Basic Fourier theory says that increasing the number of sample points increases the resolution of the Fourier transform.

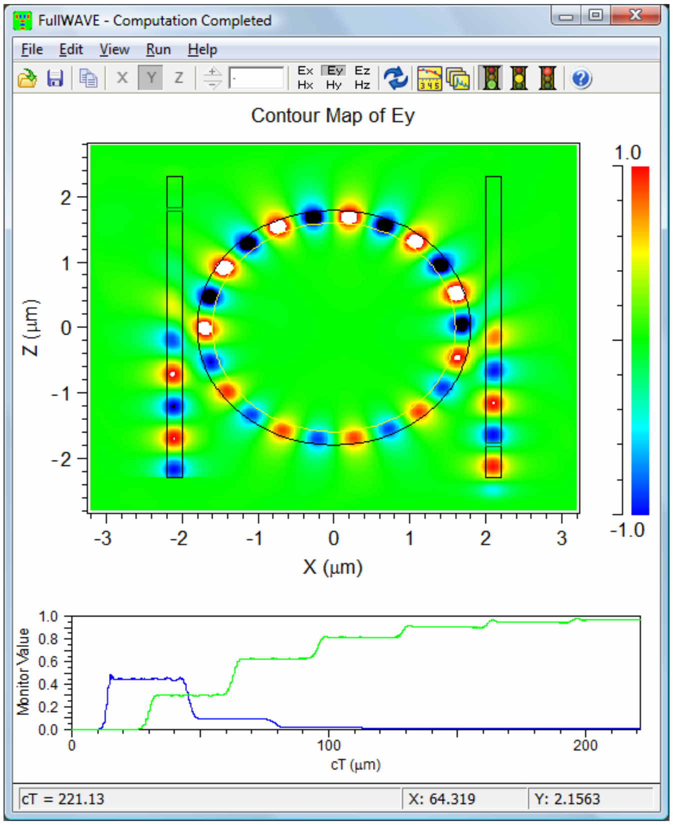
To increase the resolution of the FFT, we need to increase the number of sample points. Click the **Perform Simulation** button and set **Stop Time** to `2^15*fdtd_time_step`. Enter an **Output Prefix** like ring_pulse2 and then click **OK**. The graph ring_pulse2.pwm will have a better resolution as shown in Fig.10.

Simulation: CW Excitation

In the previous section we found the wavelength spectrum of the ring resonator. This section will focus on the resonance around 1.977 µm using a CW excitation. To change the waveguide, click the **Edit Global Settings** button in the left toolbar and change the **Free Space Wavelength** to 1.977. Click the **Perform Simulation** button an d set **Excitation** to *CW*, **Stop Time** to 2^14*fdtd_time_step, and enter a new **Output Prefix** such as ring_cw. Click **OK** to run the CW simulation.

Figure 11 shows the CW simulation results which clearly show that we are at a resonance of the system as predicted by the pulsed calculation. The energy in the waveguide builds up and, by the end of the simulation, almost the entire field is propagating in the output waveguide.

By utilizing both the pulsed and CW calculations, we have calculated the spectral response of this device and illustrated the CW operation of the device at a resonance.



cT = 221.13 | X: 64.319 | Y: 2.1563

Figure 11: The CW time evolution of the ring resonator.

Tutorial 2: Transmission Spectra of Semi-Infinite Square PBG Lattice

This tutorial discusses the computation of a transmission spectrum for a semi-infinite photonic crystal lattice. Several band-gaps will be found for one propagation direction with in the crystal.

Structure Overview

Open the file `<rsoft_dir>\examples\FullWAVE\tut2\square_pbg.ind`.

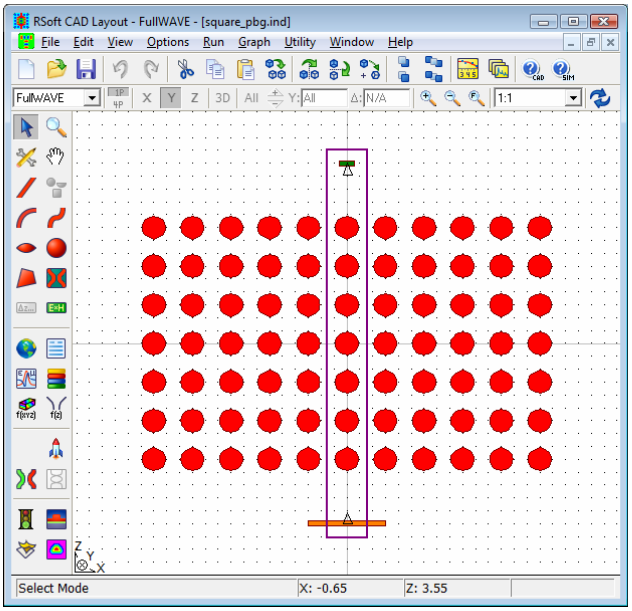

RSoft CAD Layout - FullWAVE - [square_pbg.ind]
File Edit View Options Run Graph Utility Window Help
FullWAVE
X Y Z 3D All Y: All Δ: N/A 1:1
Z
Y
X
Select Mode
X: -0.65
Z: 3.55

Figure 1: The semi-infinite PBG lattice in the CAD.

Note the following about the design:

- *PBG Crystal Details*

  The PBG lattice is composed of a 2D square array of dielectric circles with a refractive index of 3.4 in a background of air. The lattice period Λ is 0.6 μm and the hole radius is 0.3*Λ. The array was created using the Array Layout Utility described in the CAD manual.

- *Simulation Domain & Boundary Conditions*

  The simulation domain has been chosen to encompass one row of holes along Z, and one period along X. The boundary conditions are set to be Periodic along X, and PML along Z. This configuration indicates that a semi-infinite structure is to be simulated: the structure will be infinite along X, and finite (7 rows) along Z.

- *Monitor*

  One monitor has been created to record the transmitted field on one side of the structure. The Frequency Analysis options for this monitor have been set to FFT in order to compute the spectra.

- *Launch Field*

  The launch field has a rectangular spatial profile and is wider than the simulation domain, therefore effectively creating a plane wave launch. It has a TE polarization and has a pulsed excitation type which is useful for computing spectra. The pulse's center wavelength is equal to Λ/0.45.

- *Choosing a Stop Time*

  In this case, the stop time is set to `(2^15)*fdtd_time_step` indicating that a total of (2^15) time steps will be used.

Running the Simulation

Open the FullWAVE Simulation Parameters dialog and enter an **Output Prefix** such as `run_te` and press **OK** to start the simulation. The pulse can be seen entering the simulation from the bottom and slowly decaying over time. When the simulation has finished, most of the field energy will have left the domain.

To view the computed spectra, open the files `run_te.pfm` and `run_te.pwm` which contain the frequency and wavelength spectra respectively. In both cases several band gaps are clearly visible.

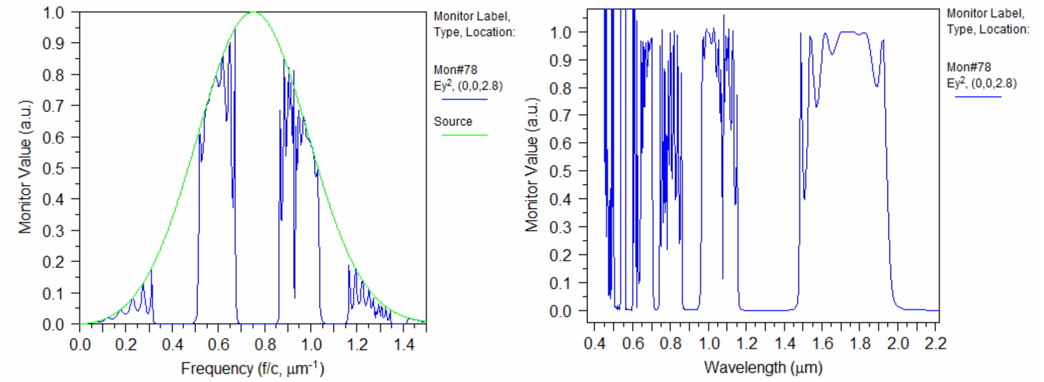


Figure 2: The computed a) unnormalized frequency and b) normalized wavelength spectra.

By default, the frequency spectrum (Fig. 2a) is not normalized, and also displays the source spectrum. The wavelength spectrum (Fig. 2b) is normalized to the source spectrum, and so is only valid around the center wavelength which is about 1.33 μm in this case. These normalizations can be changed if needed in the FDTD Output Options dialog.

Areas for Further Exploration:

Several areas to explore further are:

- *Perform a Convergence Study*

  As always, the effect of the simulation grids on the simulation should be studied. The above calculations should be repeated for smaller grid sizes and the accuracy compared: if comparable accuracy is obtained, the results are accurate. If there is a large difference, then successively smaller grids should be used until accurate results are found. This convergence process should also be done with the stop time to ensure accurate spectral results.

- *Try TM Polarization*

  Band gaps within a 2D PBG are polarization-specific. Rerun the simulation for the TM case.

- *Try more or less rows along Z*

  The results obtained are specific to a lattice that is only 7 rows thick along Z. Rerun the simulations with more or less rows and see how the band gaps change. In the limit of an infinite number of rows, the results will be the same as other methods that assume an infinite crystal such as RSoft's BandSOLVE simulation package.

Tutorial 3: PBG Y-Branch Power Splitter

The most basic optical components for integrated optics applications are linear waveguides, waveguide bends and Y splitters. In this tutorial, a Y PBG power splitter will be studied.

Structure Overview

Open the file `<rsoft_dir>\examples\FullWAVE\tut3\pbg_splitter.ind`.

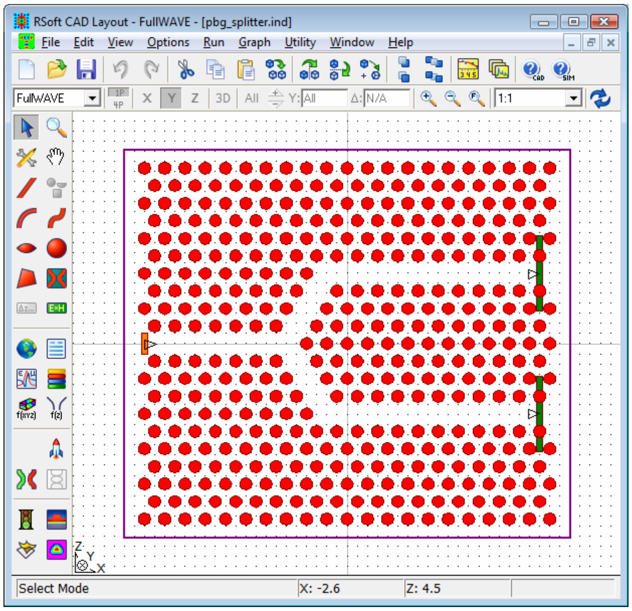

RSoft CAD Layout - FullWAVE - [pbg_splitter.ind]
File Edit View Options Run Graph Utility Window Help
FullWAVE
X Y Z 3D All Y: All Δ: N/A 1:1
Select Mode
X: -2.6
Z: 4.5

Figure 1: The PBG splitter structure in the CAD.

Note the following about the design:

- *PBG Crystal Details*

  The PBG lattice is composed of a 2D hexagonal array of air holes in a dielectric background with a refractive index of 3.4. The lattice period Λ is 0.4 μm and the hole radius is 0.3Λ. The array was created using the Array Layout Utility described in the CAD manual, and the splitter was formed by removing holes.

- *Monitors*

  Two monitors are used in this simulation to record data on both output arms. In both cases the power flux through the monitor is measured. Both monitors have been rotated 90 degrees so that they face the positive X axis.

- *Launch Field*

  The launch field has a Gaussian profile, a width equal to the period Λ, a TM polarization, a CW excitation type, and a wavelength of 1.533 μm. The launch has been rotated 90 degrees so it launches towards the positive X axis.

Running the Simulation

It is well known that PBG structures can have band gaps for certain frequency ranges. This 2D hexagonal array of air holes has a band gap for a wavelength range of 1.45 µm to 1.9 µm. Users who are licensed for BandSOLVE can verify this quite easily. The operating wavelength of 1.533 µm was chosen to maximize the output in each arm (see 'Areas for Further Exploration' below for more details). Open the FullWAVE Simulation Parameters dialog and click **OK** to start the simulation. After the simulation has completed, the results show about 35% of the power is in each output arm.

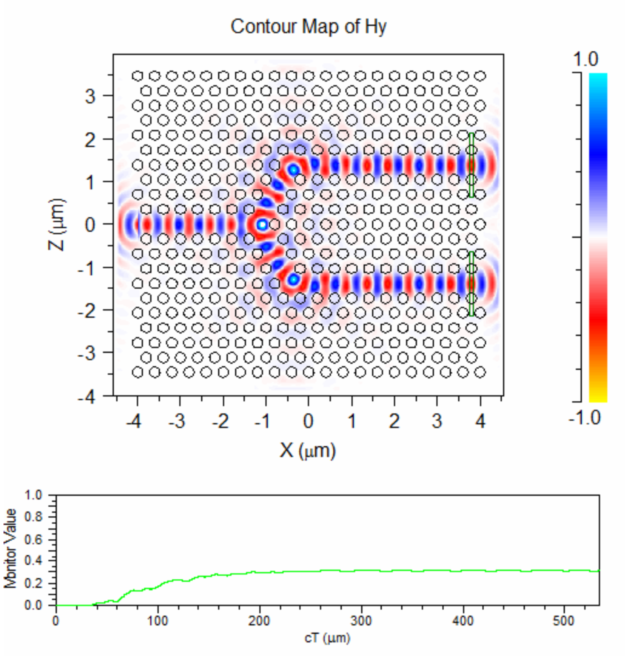


Figure 2: The simulation results for a CW excitation.

Areas for Further Exploration:

Several areas to explore further include:

- *Use a Pulsed Excitation*

  Change the **Excitation Type** to Pulsed to determine what the transmission spectrum is for this structure. This is how the operating wavelength used above was found. Note that when using a pulsed excitation, power output from a monitor is not used by default; it would be better to use field monitors. Also, longer simulations might be necessary to obtain accurate frequency output.

- *Perform a Convergence Study*

  As always, the effect of simulation grids on the simulation results should be studied. The above computations should be repeated for smaller grid sizes and the accuracy compared: if comparable accuracy is obtained, the results are accurate. If there is a large difference, then successively smaller grids should be used until accurate results are found.

Tutorial 4: Manually Computing Cavity Modes and *Q* Factors of a PBG Cavity

This tutorial illustrates the computation of a cavity mode of a PBG lattice with a defect using FullWAVE. It will cover the calculation of the cavity spectrum to determine the resonant frequencies, then determine the field profile at one resonant frequency, and then calculate its *Q* factor.

This Q factor portion of this tutorial uses the manual method for computing Q with FullWAVE. Tutorial 7 illustrates the use of Q-Finder, an automated utility for computing Q factors. While it may be tempting to use the automated approach at first, it is recommended that you work through the manual method first in order to understand how the automated approach works.

Structure Overview

This structure is composed of a hexagonal array of air holes in a silica slab (n=3.37). The lattice constant *a* used is 0.4 μm and a hole diameter of 0.65a. To create the defect, the center hole is removed. Defect modes will be looked for around a wavelength of 1.5 μm.

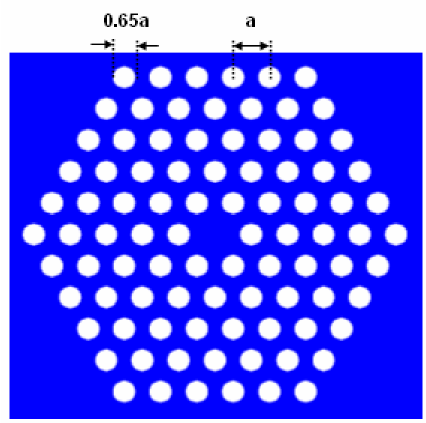


Figure 1: The PBG structure with defect used in this tutorial. The lattice constant a is 0.4 μm, and consists of a hexagonal array of air holes in a silica slab with index of 3.37.

Creating the Structure in the CAD Environment

This structure will be created in the CAD by modifying the output of the Array Layout Utility. The utility can be opened via Utility/Array Layout in the CAD menu. A base 2D hexagonal array in the XZ plane can be created with the following settings:

Parameter	**Value**
Dimensions	*2D XZ*
Lattice Type	*Hexagonal*
Lattice Size	**L** = *11*, **M** = *11*

Unit Cell Contents	*Circle*

Enter a **Output Prefix** of `defect` and click **OK** to create the file.

Defining Symbols

Next, press the **Edit Symbols** icon in the left toolbar and set/modify the following variables:

Variable	Value	Description
`Period`	`0.4`	The period of the lattice. This variable was created by the Array Layout Utility.
`Radius`	`0.325*Period`	The radius of the holes in the lattice. This variable was created by the Array Layout Utility.
`Nslab`	`3.37`	Sets the index of the slab.
`Nhole`	`1`	Sets the index of the holes
`background_index`	`Nslab`	Index of background material, which in this case is the slab.
`delta`	`Nhole-background_index`	Index difference of hole objects.
`polarization`	`1`	The polarization of the simulation to TM.
`sim_tool`	`ST_FULLWAVE`	Sets FullWAVE as the simulation engine to be used. The Array Layout Utility automatically sets BandSOLVE. This is equivalent to setting this parameter in the Global Settings dialog.

To create the structure, delete objects until it appears as shown in Fig. 1.

Adding a Monitor

A monitor will be used to record the time response of the system. Insert a monitor to record the field at a position (X, Z) of (`0.234*Period`, `0.411*Period`) and a **Width** of `Period`. The semi-random position avoids highly symmetric positions where defect modes are generall located. The width setting is not necessary, since the field value is measured at the center of the monitor, but makes the layout window neater.

Once these settings have been made, the structure will appear as shown in Fig. 2.

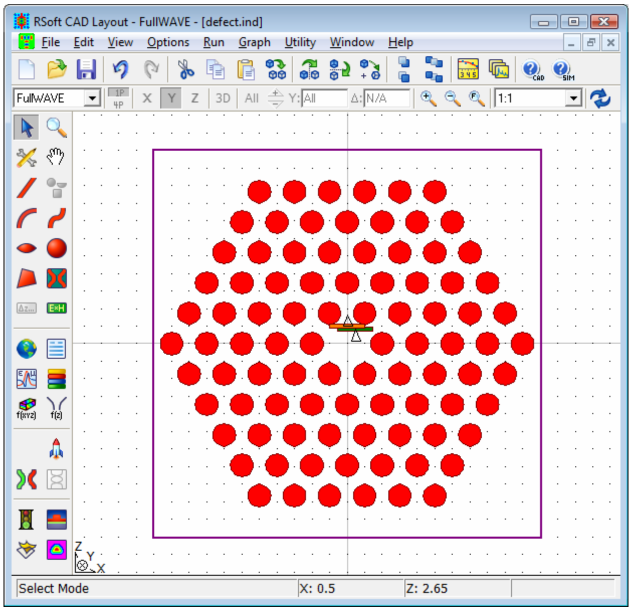

RSoft CAD Layout - FullWAVE - [defect.ind]
File Edit View Options Run Graph Utility Window Help
FullWAVE
X Y Z 3D All
Y: All
Δ: N/A
1:1
Select Mode
X: 0.5
Z: 2.65

Figure 2: The completed structure with monitor seen in the CAD window.

Checking the Index Profile

Click the **Display Material Profile** icon on the left toolbar and make the following changes:

Parameter	Value	Description
Display Mode	*ContourMap(XZ)*	Sets the type of display used to show data.
X **Grid Step**	`PeriodX/16`	Sets the X grid size used for computation so that each period along X has 16 grid points. The variable `PeriodX` was automatically created by the Array Layout Utility.
X **Slice Grid**	`grid_size`	Sets the X display grid used for index calculations equal to the computation grid defined by the X **Grid Size**.
Z **Grid Step**	`PeriodZ/16`	Sets the Z grid size used for computation so that each period along Z has 16 grid points. The variable `PeriodZ` was automatically created by the Array Layout Utility.
Z **Slice Grid**	`step_size`	Sets the Z display grid used for index calculations equal to the computation grid defined by the Z **Grid Size**.
X/Z **Domain Min**	`-2.2`	Sets the X and Z domain minimum
X/Z **Domain Max**	`2.2`	Sets the X and Z domain maximum

Next, click the **Display...** button and set the **Outline Color** to *Default* and the **Aspect Ratio** to `1`. These settings will display an outline of the structure on simulation results and fix the aspect ratio of the simulation plots to unity. Click **OK** twice to compute the index profile.

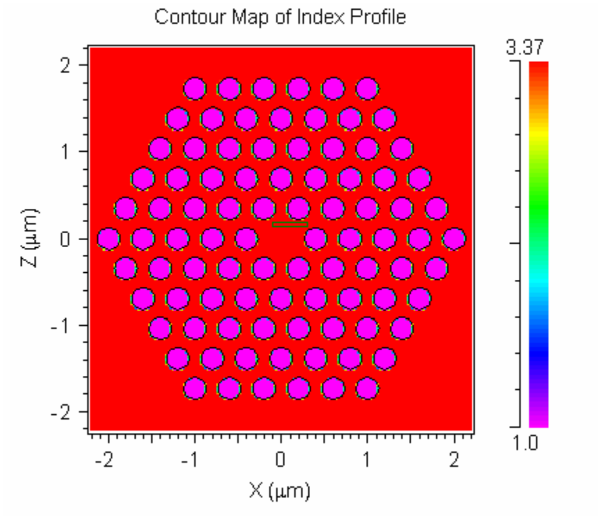


Figure 3: The computed index profile of the defect structure.

Computing the Cavity Resonances

The calculation of a cavity mode can be broken into two main steps: finding the resonant frequency and then solving for the mode profile at that frequency. This section will discuss the computation of the cavity resonances; the next will discuss the computation of the mode profile.

To find the resonances a cavity supports a wide range of frequencies must be excited. The frequency response of the cavity is measured and the peaks in the response represent cavity mode frequencies.

Setting the Excitation

To excite a wide range of frequencies an impulse excitation should be used. This type of excitation is a delta function in time and therefore excites a large frequency range. Click the **Perform Simulation** button in the left toolbar to open the FullWAVE Simulation Parameters dialog. Set **Excitation** to *Impulse*. Like the position of the monitor, the launch field should be placed at a low symmetry position unless a certain symmetry is desired. Click the **Launch...** button and set the following parameters:

Parameter	Value	Description
Position X	`0`	Sets the X position of the launch. Even though this value has high symmetry, it is used because the mode we will find is degenerate, and this value chooses the mode with desired symmetry.
Position Z	`0.2`	Sets the Z position of the launch.
Width	`Period`	This sets the width of the launch field.

Performing the Simulation

To perform a cavity mode calculation, click the **Advanced...** button in the FullWAVE Simulation Parameters dialog and enable the **Compute Mode** option. This indicates that the wavelength spectrum of the monitor results will be calculated, as well as the mode profile found at the **Free Space Wavelength** specified in the Global Settings dialog. Since we don't know the exact resonant frequency, we will not set the wavelength until we have examined the spectrum.

Open the FullWAVE Simulation Parametsrs dialog and set the following parameters:

Parameter	Value
Stop Time	(2^14)*fdtd_time_step
X PML Width	8*grid_size
Z PML Width	8*step_size

The rest of the settings are correct for this simulation; enter an **Output Prefix** of `mode` and click **OK** to perform the simulation. Even though it looks like there is no activity in the simulation window beyond cT = 50 μm, a longer stop time is needed to get better wavelength/frequency resolution in the Fourier transformation.

The **Free Space Wavelength** setting in the Global Settings dialog was not a resonance of the structure, and so the mode calculated will not be important. To find the resonances, we need to look at the wavelength spectrum.

Picking a Resonance

To view the frequency spectrum, click the **View Graphs** icon on the top toolbar and select the file `mode.pfm`. Each of the peaks in this graph represent a resonance of the structure.

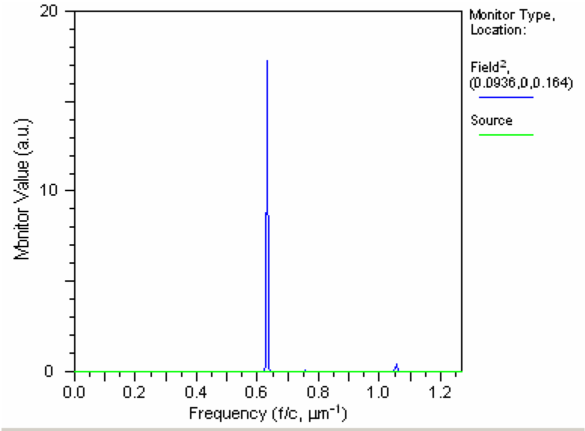


Figure 4: a) The frequency spectrum found from the impulse simulation of the defect structure. The peak at 0.63479 μm^{-1} represents a wavelength of 1.575 μm. Note that the utility `fhakit` is the preferred way to find peaks.

Computing a Cavity Mode Profile

For this example, we will compute the cavity mode profile for the resonance at 1.575 μm shown in Fig. 4. To calculate the cavity mode profile at this wavelength, change the **Free Space Wavelength** in the Global Settings dialog to `1.575`. Also, enable the output of a FullWAVE state file by using the **FDTD Continue** option in the FDTD Advaned Parameters dialog.

Though not necessary for the cavity mode compuation, it can be useful to save the computed mode in a FullWAVE state file for later use using the FDTD Continue option. This can be useful for measuing the Q of the mode as will be done later in this tutorial.

To compute the cavity mode, rerun the calculation with the same **Output Prefix**. At the end of the simulation, the mode plot is saved in the file `mode.p00`, the raw data for this component is contained in `mode.m00`, and the entire mode, including all field components, is contained in the state file `mode.fst`.

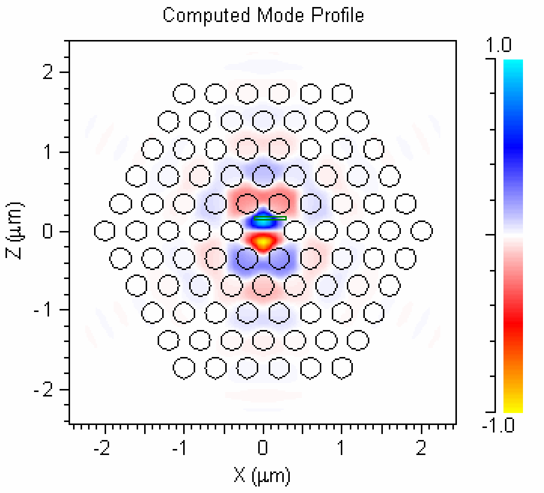


Figure 5: The H_y component of the computed cavity mode for this defect structure at the chosen resonance. The color scale used is fireice_white.scl and is included with the RSoft distribution.

Using Symmetric Boundary Conditions

The H_y mode shown in Fig. 5 has even symmetry along the X axis and odd symmetry along the Z axis. We can take advantage of this knowledge and accelerate subsequent calculations: we will use a domain of a quarter of the size resulting in a four-fold increase of effective simulation speed. Using the symmetry conditions chart for 2D TM polarization in Section 6.C.3, the symmetry properties of this mode require symmetric boundary conditions along X, and anti-symmetric boundary conditions along Z. As usual, PML conditions will be used on the outer edges of the domain.

The choice of boundary symmetry is not intuitive; though this example requires symmetric boundary conditions for a symmetric field, this is not always the case. Always be sure to check the tables in Section 6.C.3.

To implement these boundary conditions, open the FDTD Advanced Parameters dialog and set the **X Min** boundary setting to *Symmetric*, and the **Z Min** boundary setting to *Anti-Symmetric*. Also, set the X and Z **Domain Min** in the FullWAVE Simulation Parameters dialog to `0` since we will only simulate half the domain along both axes.

The position of the monitor and launch field were purposely chosen so that they would lie within the first quadrant (X > 0, Z > 0). If, for some reason, you placed them elsewhere, you will have to move them so that they lie within the reduced simulation domain.

Re-calculating the mode on the reduced domain gives the same resonance in the wavelength spectrum, though some peaks may be missing if the modes they represent do not have the symmetry enforced by the boundary conditions.

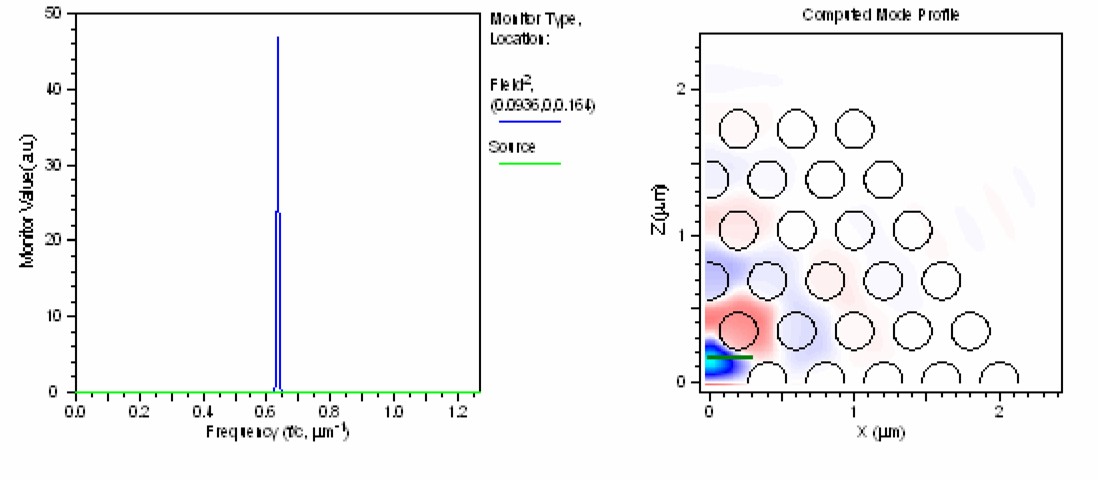


Figure 6: a) The frequency spectrum, and b) Hy component of the cavity mode found on the reduced domain. Note the same resonance at 0.63479 µm-1 (1.575 µm), and the similarity of the computed mode profile to the previous results.

Since the **FDTD Continue** option is enabled, the computed mode will be saved in a binary state file, `mode.fst`, assuming an **Output Prefix** of `mode` was used. This file contains all field information from the simulation, and can be 're-launched' into the simulation in order to measure its decay, and therefore the *Q* of the mode.

Computing the *Q* Factor

The calculation of *Q* is done by performing one final simulation which monitors the energy density inside the cavity as the mode decays in time.

This section describes a manual method for Q computation; Chapter 8 describes several methods, and Tutorial 7 describes an automated method using the Q-Finder utility.

Letting the Resonance Decay

In order to measure the total energy within a volume, add a total energy density monitor to the domain. The monitor should have the following parameters set:

Parameter	Value	Description
Width	`inf`	Sets the width (measured along X) of the monitor to infinity so that it effectively measures the entire simulation domain.
Length	`inf`	Sets the length (measured along Z) of the monitor to infinity so that it measures the entire simulation domain.

CAD Display (Width & Length)	`1`	Sets the display width and length of the monitor to a finite value so that the CAD file is easier to manipulate.
Single Value Output	*T Density*	Enables the output of the total energy density.
Time Average	*Yes*	Enables the output of time average data for

The position of the monitor is not critical since it has infinite size. Place it so that the majority of it lies within the simulation domain. Also, enable the **Time Monitor** output in the FDTD Output Options dialog since, by default, the time domain data is not output when using the *Impulse* excitation type.

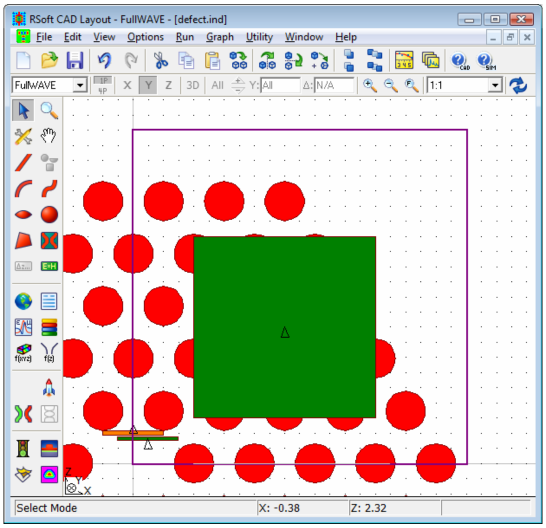


Figure 7: The total energy density monitor as seen in the CAD window.

Next, the mode we computed previously needs to be used as the initial field at t=0, and let to decay in time via the impulse excitation type. To do this, set the variable `fdtd_excitation_file` to `mode.fst`, our state file, and note that an excitation type of *Impulse* is already set. Also, since we do not want to compute the cavity mode, deactivate the **Compute Mode** setting in the FDTD Advanced Parameters dialog.

Rerun the simulation with an **Output Prefix** of `decay`. The initial field profile should be the computed mode which decays as the simulation progresses.

Measuring Energy Decay Constant

The simplest way to calculate Q is to measure the rate at which the energy decays. This can easily be accomplished by fitting the energy density monitor results to a decaying exponential ($\alpha>0$) as described in Section 8.A.2: $U(t) = U_0 e^{-\alpha t}$. The raw output data from the previous simulation is located in the file `decay.tmn`, and can be viewed by opening the file `decay.ptm` in WinPLOT. The decay constant α can be measured by one of the following methods:

- *Directly form the graph using the mouse*

  Open the file `decay.ptm` with WinPLOT, and move the mouse over two points along the curve to measure the quantities U_1, U_2, and Δt as illustrated in Fig. 8a. The decay constant α can then given by $\alpha = \ln(U2/U1)/\Delta t$.

- *From the raw data file* `decay.tmn`

  The file `decay.tmn` contains the raw simulation output data. The values of U_1, U_2, and Δt can be directly measured from this file and used to calculate the decay constant α. For this example, the first column contains the time, and the second data column (third column in the file) contains the energy monitor result. Also, note that a `.tmn` file actually records the square root of the energy and power, and so these values should be squared before the calculation. The `/powy2` command is used in the file `decay.ptm` to automatically to this when using WinPLOT to view the data.

- *Using WinPLOT's curve fitting features*

  WinPLOT can automatically fit an exponential curve to the data using built-in curve fitting features. The easiest way to use these features is via the command prompt. Open a command prompt in the current directory by selecting *Utility/Terminal* from the CAD menu, and issue the following command:

  ```
  winplot /powy2 /u2 /fexp /fout"coeffs.fit" decay.tmn
  ```

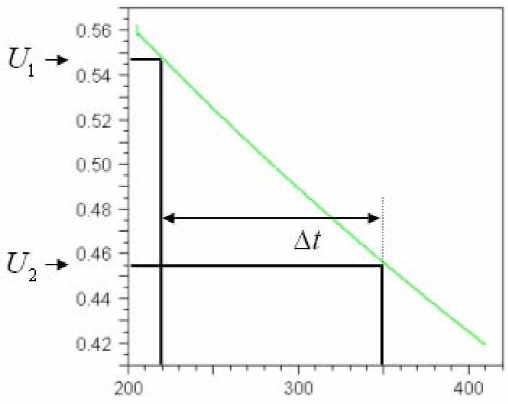
  The command `/powy2` is used for the same reason described in the previous method, and the command `/u2` was used because the energy data is in the second column of `decay.tmn`. The constant α equals the value on the **P1** line in the newly-created file `coeffs.fit` as shown in Fig. 8b.

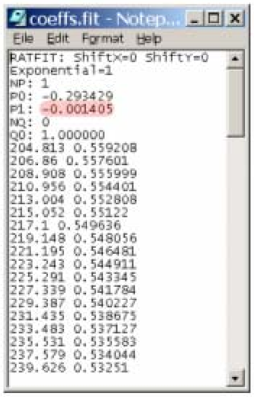
These methods give a value of α = 0.001405.

Calculating the Q Value

As shown in Section 8.A.2, the Q value is given by $Q = 2\pi\nu/\alpha$. Given a value of α = 0.001405, the resonant frequency $\nu = 1/\lambda = 0.635$, a value of Q = 2840 is obtained.

Note that a convergence study has not been performed, and so the value obtained may not be accurate. Also, since this is not a converged value, the Q value found here might differ from values you find due to improvements made to the FullWAVE algorithm which can change the convergence process. This underscores the importance of always performing a convergence study, as described below, to ensure accurate results.





```
RATFIT: ShiftX=0 ShiftY=0
Exponential=1
NP: 1
P0: -0.293429
P1: -0.001405
NQ: 0
Q0: 1.000000
204.813 0.559208
206.86 0.557601
208.908 0.555999
210.956 0.554401
213.004 0.552808
215.052 0.55122
217.1 0.549636
219.148 0.548056
221.195 0.546481
223.243 0.544911
225.291 0.543345
227.339 0.541784
229.387 0.540227
231.435 0.538675
233.483 0.537127
235.531 0.535583
237.579 0.534044
239.626 0.53251
```

Figure 8: Illustration of different methods described above for calculating the decay constant α: a) Using the mouse or the raw data in file `decay.tmn`. Only the total energy density monitor is shown; the field monitor has been removed for clarity. b) The WinPLOT curve-fitting results in the file `coeffs.fit`.

Areas for Further Exploration

There are several areas that can be explored further in this case. They are:

- *Convergence study using a larger structure and/or finer Grid Sizes*

  As always, the effect of grid sizes on the simulation results should be studied. The results should be repeated for smaller grid sizes: if comparable accuracy is obtained, the results are accurate; if there is a large difference, then successively smaller grids should be used until accurate results are found.

  Similarly, the size of the structure should be studied: a larger supercell that contains more periods of the lattice should be used: if the results are similar, then the smaller structure is adequate; if not, larger structures should be used until accurate results are found.

- *Longer simulations for better frequency resolution*

  The frequency resolution is directly related to the length, or number of time steps, in the simulation. Try rerunning the mode computation with a greater stop time and explore this relationship. A good way to do this is to increase the exponent for the power of 2 used to define the stop time: try `(2^14)*fdtd_time_step`, then `(2^15)*fdtd_time_step`, etc.

- *Using Q-Finder*

  This tutorial described the manual approach to solving for mode profiles and *Q* values. RSoft's Q-Finder utility can automate much of this process. Q-Finder is described in Chapter 9, and Tutorial 7 gives examples of its use.

Tutorial 5: Incorporating Material Dispersion

FullWAVE simulations with dispersion materials can be broadly broken into two main categories:

- *Using a Single Wavelength*

  When using a single wavelength, such as most simulations with a CW excitation, it is usually only necessary to define material properties at this wavelength. However, it may still be necessary to enable some of FullWAVE's dispersion options to ensure accurate results. Parameter scans over wavelength also fall into this category since each step of the scan is at a single wavelength.

- *Using Multiple Wavelengths*

  When using multiple wavelengths, such as simulation with a pulsed or impulse simulation, or CW simulations with non-linear effects such as four-wave mixing, it is necessary to define material properties over the wavelength range used in the simulation.

This tutorial assumes that the user is familiar with the methods outlined in Chapter 8 of the CAD manual used to define material dispersion in the CAD, and will give examples of each. The dispersive method used will depend on what type of FullWAVE simulation being performed, as well as the format of the material data available to the user. Examples and guidelines for each method will be given. Example files for this tutorial are located in the directory:

```
        <rsoft_dir>\examples\FullWAVE\Tut5\
```

The example files in this tutorial do not necessarily represent real-world structures and are meant to only illustrate the creation of dispersive materials.

Method 1: Dispersion via the Single Value Method

The single value method simply defines the refractive index of a material by a single value.

This method is valid for simulations with one wavelength, and is not valid for parameter scans over wavelength with a CW excitation where dispersive effects are important.

Open the file `single_value_method.ind`. This file contains a waveguide with a refractive index equal to 1.5+2*i*. We will perform a simulation with a CW excitation.

Setting the Waveguide Parameters

Open the Symbol Table and note the variables Index, alpha, and delta which represent the real, imaginary, and index delta of the waveguide.

Performing the Simulation

Click the **Perform Simulation** button in the left toolbar and click OK to start the simulation. Note how the field is reflected from the waveguide like the example shown in Fig. 1b. It is important to note that for this specific example, it was necessary to enable the **Dispersion/Nonlinearity** option in the

FullWAVE Simulation Parameters dialog since the material is lossy. Had it not been enabled, the simulation would have appeared as shown in Fig. 1a.

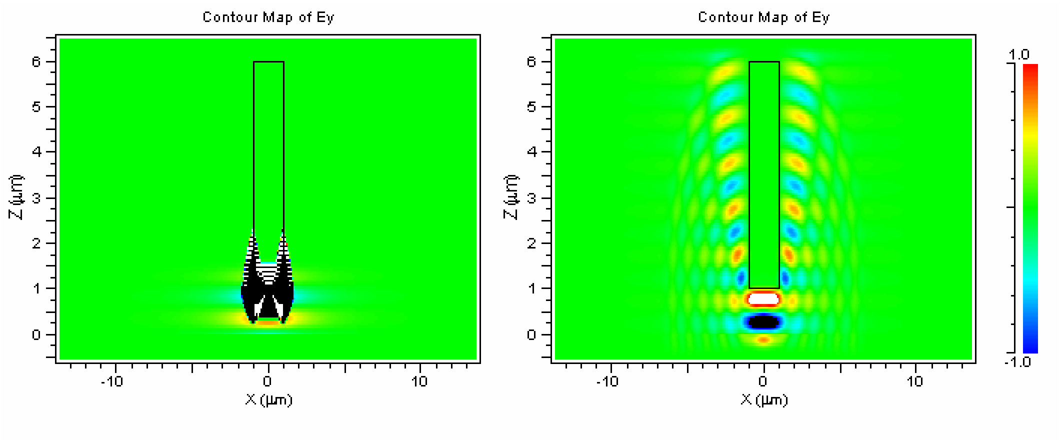


Figure 1: An illustration of the effect of FullWAVE's built-in Drude dispersion model. These two simulations windows show the same structure with the model a) off and b) on.

Method 2: Dispersion via the Expression Method

The expression method defines the refractive index of a material by a mathematical expression in the Symbol Table.

This method is valid for simulations with one wavelength as well as parameter scans over wavelength since the wavelength with a CW excitation.

In this example, the waveguide's refractive index as a function of wavelength via a mathematical expression. A series of CW simulations will be performed at different excitation wavelengths to determine the frequency response of the system. While this could be accomplished by manually entering a different value of the index and performing a simulation via the single value method, the use of an expression and RSoft's MOST scanning tool is a much more efficient approach.

For this example, we will use a waveguide composed of borosilicate crown glass (BK7). The Sellmeier Equation, which can be used to represent the dispersive properties of the material, is given by the following formula:

$$n^2(\lambda) = 1 + \sum_i \frac{B_i \lambda^2}{\lambda^2 - C_i}$$

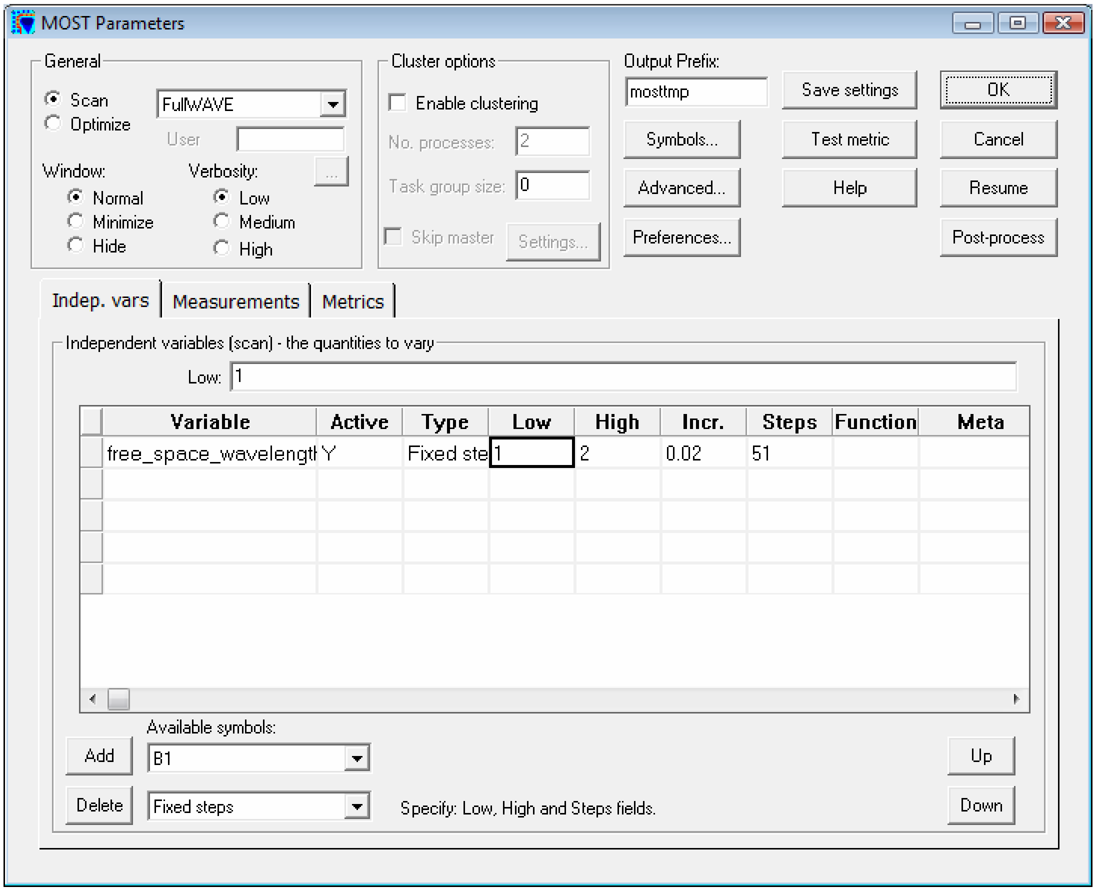
This formula can be expressed in the symbol table as `Index = sqrt(1+S1+S2+S3)` where `S1`, the first resonance, is equal to `(B1*lambda^2)/(lambda^2-C1)` where `B1` and `C1` are the coefficients of the first resonance and `lambda` is the built-in variable that is equal to the simulation wavelength. The second and third resonances, `S2` and `S3`, are similarly defined.

Open the file `expression_method.ind` and note that it consists of a single waveguide surrounded by air. The launch field is below the waveguide, and a monitor is above the waveguide. Because of this configuration, the waveguide will act as a resonant cavity.

Open the symbol table and note that the variables `Index`, `S`, `B`, and `C` variables which have been defined. Also note that the variable delta, which is used to set the index difference of the waveguide, has been set to be `Index-background_index`. This material has only a real refractive index; there is no loss coefficient. For other material systems, an expression can be used to define the loss as well.

Using MOST to Scan Over the Wavelength

Click the **Launch MOST** button in the left toolbar to view the settings for this scan. The variable `free_space_wavelength`, which sets the simulation wavelength, will be scanned from 1.0 to 2.0 µm with 51 steps.



General

● Scan ○ Optimize

FullWAVE

User

Window: ● Normal ○ Minimize ○ Hide

Verbosity: ● Low ○ Medium ○ High

Cluster options

□ Enable clustering

No. processes: 2

Task group size: 0

□ Skip master

Output Prefix: mosttmp

Independent variables (scan) - the quantities to vary

Low: 1

Variable	Active	Type	Low	High	Incr.	Steps	Function	Meta
free_space_wavelengtl	Y	Fixed ste	1	2	0.02	51		

Available symbols: B1

Fixed steps

Specify: Low, High and Steps fields.

Figure 2: MOST scan setup.

Click **OK** to start the scan. The program will proceed to iterate through each value of the wavelength. The refractive index of the waveguide will change for each simulation according to the function that is in the symbol table. By default MOST will produce a plot of the last monitor results vs. the scanned parameter. When the scan has completed, press the **Open DataBROWSER** icon in the MOST simulation window and open the results:.

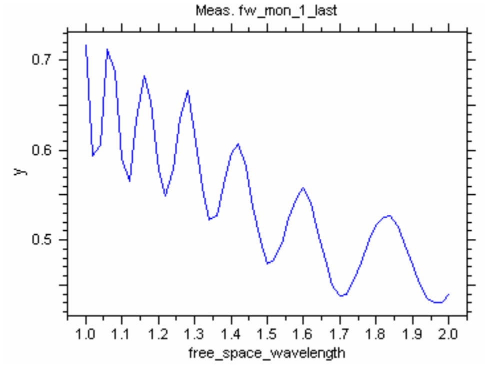


Figure 3: MOST scan results from scan with index changing as a function described by an equation.

The oscillations of the resonant cavity can clearly be seen in Fig. 3. Also, since the refractive index of the cavity changes as a function of wavelength, the resonance response changes as well.

It is important to note that for this specific example, it was not necessary to enable the **Dispersion/Nonlinearity** option since only one wavelength was simulated and the material was not lossy. Had the material been lossy, this option would have been needed.

Method 3: Dispersion via the Data File Method

The data file method defines the refractive index of a material by a data file.

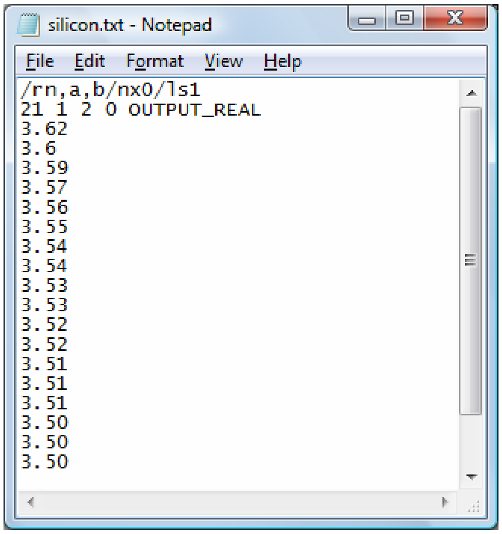
This method is valid for simulations with one wavelength as well as parameter scans over wavelength since the wavelength with a CW excitation.

Open the file `data_file_method.ind`. This file contains a waveguide composed of silicon that is surrounded by air. This structure is similar to the one used to illustrate the expression method, and will act as a resonant cavity.

The Data File

The data file used in this example is `silicon.txt`, and is located in the same directory as the `.ind` file for this design. This file contains the real refractive index of silicon between a wavelength of 1 and 2 μm in a format described in Appendix B of the CAD manual.

This file can be incorporated into the CAD through the `userdata()` function which is described in Appendix C of the CAD manual. This function allows variables to take values from a data file based on another variable. Open the Symbol Table and note the definition of the variable `delta` as `Index-background_index` where `Index` is defined as `userdata("silicon.txt", lambda)`. This indicates that the value of the `Index` variable will be taken from the file `silicon.txt` based on the value of `lambda`.



```
/rn,a,b/nx0/ls1
21 1 2 0 OUTPUT_REAL
3.62
3.6
3.59
3.57
3.56
3.55
3.54
3.54
3.53
3.53
3.52
3.52
3.51
3.51
3.51
3.50
3.50
3.50
```

Figure 4: Data file with the real index of silicon.

Using MOST to Scan Over the Wavelength

Click the **Launch MOST** button in the left toolbar to view the settings for this scan. The variable `free_space_wavelength`, which sets the simulation wavelength, will be scanned from 1.0 to 2.0 µm with 101 steps. Press **OK** to start the scan. The program will proceed to iterate through each value of the wavelength. The refractive index of the waveguide will change for each simulation according to the function that is in the symbol table. By default MOST will produce a plot of the last monitor results vs. the scanned parameter. When the scan has completed, press the **Open DataBROWSER** icon in the MOST simulation window and open the results:

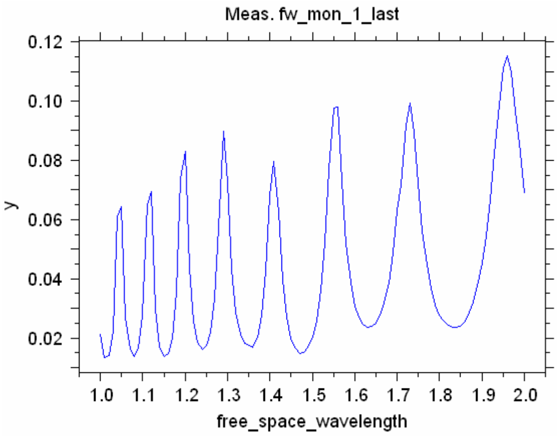


Figure 5: MOST scan results from scan with index changing as a function described by a data file. The plot shows the reflected field from the waveguide.

The oscillations of the resonant cavity can clearly be seen in Fig. 5. Also, since the refractive index of the cavity changes as a function of wavelength, the resonance response changes as well.

It is important to note that for this specific example, it was not necessary to enable the **Dispersion/Nonlinearity** option since only one wavelength was simulated and the material was not lossy. Had the material been lossy, this option would have been needed.

Method 4: Dispersion via the Material Editor

RSoft's Material Editor allows the user to create materials with complicated dispersive and non-linear properties. The specification of such a material is covered at length in the CAD manual.

This method is valid for any type of simulation, including CW, pulsed, and Impulse excitations. When used in simulations of one wavelength, the waveguide will have the refractive index defined for that wavelength; when used in simulations of multiple wavelengths, the each wavelength in the simulation will see the appropriate refractive index.

Open the file `material_editor_method.ind`. This file contains a waveguide that uses the Material Editor to define the dispersive properties of a waveguide. This material properties of the waveguide are based on the following reference:

> Aleksandar D. Rakic, Aleksandra B. Djurisic, Jovan M. Elazar, and Marian L. Majewski, "*Optical properties of metallic films for vertical-cavity optoelectronic devices*," Applied Optics, **37,** 5271 (1998).

The waveguide will be excited with a pulse, and the transmitted and reflected field will be measured as a function of wavelength.

The Waveguide, Monitor, and Launch Field

The waveguide in this design file has a width of 10 μm and a length of 0.2 μm. Note that its **Material Properties** are set to *Ag*, and the **Index Difference** and **Index (imag part)** cannot be edited. This is because the properties of the waveguide are defined via the Material Editor.

The launch field is placed between the lower monitor and the waveguide so that the lower monitor only records the reflected field. There are two monitors defined; both are configured to compute the FFT of the time response to produce the frequency response of the system.

To look at the properties of the material, click the **Edit Materials…** button in the left toolbar and select *Ag* from the Project Material Category. The refractive index can be viewed by clicking the **Display** button in the **Eps Linear** tab.

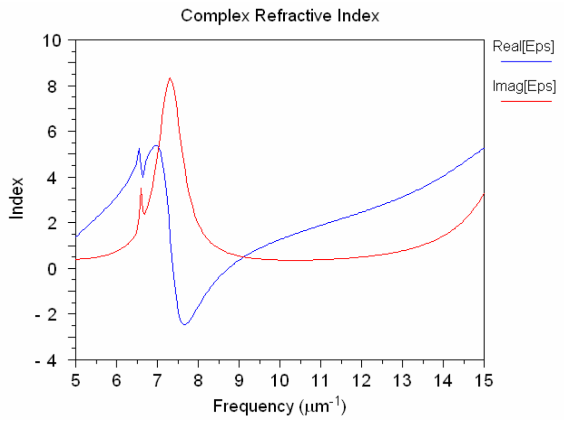


Figure 6: The complex valued refractive index as a function of frequency. Note that the x-axis is in inverse frequency, µm^{-1}.

Performing the Simulation

Click the **Perform Simulation** button in the left toolbar, enter an **Output Prefix** such as `pulse`, and click **OK** to start the simulation. During the simulation the monitors will record the time response of the simulation, and, when completed, the frequency response will be computed. When completed, click the **View Graphs** button and open the file `pulse.pfm`.

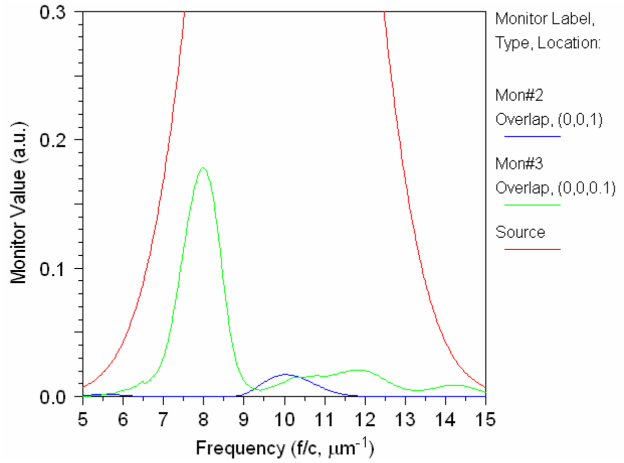


Figure 7: Frequency response of the dispersive system. The blue line corresponds to the transmitted field, the green line to the reflected field, and the red line to the source spectrum.

The frequency results, shown in Fig. 7, show that there is a peak transmission at 10µm^{-1}, or a wavelength of 0.1 µm. The peak reflection occurs at 7.5 µm^{-1}, or a wavelength of 0.133 µm.

Summary

This tutorial has gone through the process of setting up dispersive situations when using FullWAVE. The internal **Dispersion/Nonlinearity** feature must be turned on when doing either a CW case where the device is highly lossy or a pulsed/impulse case where dispersion needs to be included.

For CW simulations, dispersion can be implemented through an equation, datafile, or via the material editor. However, for pulse or impulse simulations, dispersion can only be implemented via the material editor method.

Areas for Further Exploration

There are several areas that can be explored further in this case. They are:

- *Try using a frequency-dependent imaginary index with the expression and data file methods*

  The examples used in this tutorial used only a real-valued refractive index. Rerun the examples with a complex valued index and see how the results change.

- *Look at other frequency ranges when using a pulsed excitation and the material editor method*

Because we used a pulsed excitation in this example we were only able to study the narrow portion of the frequency spectrum that the pulse excited. By changing the central wavelength and size of the pulse in time, we can look at different frequencies of the response of this material. Also, by using an impulse excitation properly, we can look at a very large range of frequencies.

- *Explore Other Material Systems*

  In addition to this, please note that the file `material_editor_method.ind` that comes with this tutorial has all of the material properties that are contained in the reference. Thus, there are several metals that can be found in the material editor that have already been converted from the reference.

Tutorial 6: Incorporating Non-Linearities

Non-linearities, the change of index as a function of the intensity of the field, are not intrinsically included in a FDTD simulation. FullWAVE has a feature that can automatically include non-linear material in a simulation. This non-linearity is a third order, or χ^3, non-linearity based on a Kerr model. This tutorial will go through the setup of a dispersive and non-linear calculation for FullWAVE and one way to analyze the results. A temporal soliton system will be created, simulated, and analyzed. User familiarity with the CAD interface is assumed. This discussion will reproduce the results from:

> P. Goorjian, A. Taflove, R. Joseph, S. Hagness, "Computational Modeling of Femtosecond Optical Solitons from Maxwell's Equations," *IEEE J. of Quan. Ele.*, **28**, No 10., 2416 (1992).

Example files for this tutorial are located in the directory

```
<rsoft_dir>\Examples\FullWAVE\Tutorial\Tut6\.
```

Device Layout

Open the file `nl_waveguide.ind`. This file contains a single waveguide, the properties of which are defined via the Material Editor.

Converting Parameters

In order to illustrate the conversion of the parameters from the units described in the reference to the units that need to be input into the CAD, one set of symbols represent the values given in the reference, and another set of symbols represents the same parameters with the necessary unit conversions.

Open the symbol table and note the following symbols which correspond to the units defined in the reference:

Symbol	Value	Description
`Freq_ref`	`1.37e14`	The driving frequency given in the reference (in s^{-1}).
`Eps_inf_ref`	`2.25`	The epsilon value at infinite frequency given in the reference.
`Eps_s_ref`	`5.25`	The epsilon value at zero frequency given in the reference.
`W0_ref`	`4.0e14`	The resonant frequency given in the reference (in s^{-1}).
`Delta_ref`	`2.0e9`	The damping factor given in the reference (in s^{-1}).
`Chi3_ref`	`0.085`	A modified χ^3 value from the reference since we will not consider a dispersive nonlinearity in this example. This quantity is assumed to be dimensionless since a **Launch Normalization** of *Unit Peak* will be used. More details about this can be found in the notes at the end of Chapter 8 of the CAD manual.

Also note the following symbols which were created to represent the necessary unit conversions:

Symbol	Value	Description
`Freq`	`Freq_ref/c`	The driving frequency to be used in the simulation (in μm^{-1})
`W0`	`W0_ref/c`	The resonant frequency to be used in the simulation (in μm^{-1}).
`DeltaEps1`	`(Eps_s_ref-Eps_inf_ref`	A quantity that will be used to define the material's dispersive

	`)*W0^2`	properties.
`B1`	`2*Delta_ref/c`	A quantity that will be used to define the material's dispersive properties.
`C1`	`W0^2`	A quantity that will be used to define the material's dispersive properties.
`free_space_wavelength`	`1/Freq`	Sets the driving wavelength for the simulation in µm.

In addition to unit conversion, several symbols represent quantities that are used to define the material's dispersive properties.

The Material Properties

Next, open the Segment Properties dialog for this segment and click **Materials...** to open the Material Editor and view the settings for this material:

Option	Value	Description
Epsilon at Infinity	`Eps_inf_ref`	Sets the epsilon at infinite frequency.
delta (#1)	`DeltaEps1`	Sets the strength of the first dispersive resonance.
a (#1)	`1`	Sets the fitting coefficient a_1.
b (#1)	`B1`	Sets the fitting coefficient b_1.
c (#1)	`C1`	Sets the fitting coefficient c_1.
Chi3	`Chi3_ref`	Sets the non-linear term χ^3

Click the **Display** button to see the resulting dispersion curve.

The exact formulation of the dispersive and non-linear parameters set here is described in detail in Chapter 8 of the CAD manual.

The Launch Parameters

The reference describes a 1D simulation of a pulse with a hyperbolic secant envelope function with a characteristic time constant of 14.6 fs propagating along a waveguide segment. To emulate this type of simulation, we will perform a 2D simulation, launch a plane wave along Z, and utilize periodic boundary conditions at the X boundaries. Therefore it is not be necessary to define a specific background index, nor a width for the waveguide.

Open the Launch Parameters dialog and note that the **Type** is set to *Rectangle*, **Width** to `inf`, and **Normalization** to *Unit Peak*. The selection of the launch normalization plays an important role in non-linear calculations, and is outlined in the notes at the end of Chapter 8 of the CAD manual in more detail.

Next, open the Simulation Parameters dialog and note that the **Excitation** is set to *Pulse*, and, in the symbol table, the symbol `fdtd_excitation` has been set to `TIME_ENVELOPE_SECH`. This indicates that a pulse with a hyperbolic secant envelope function should be used. The **Ramp/Pulse Time** is set to `Pulse_width_ref/c` which is the value from the reference in the correct units for FullWAVE.

The Monitors

Two monitors are used to record field values as the pulse propagates down the waveguide. One monitor is located at Z=`Z1`, and the other at Z=`Z2`.

The Simulation Parameters

Since we are emulating a 1D simulation, there is little need to simulate a large region along the X axis. Therefore, the X **Domain Min** and X **Domain Max** have been set to `-grid_size` and `grid_size` respectively so that a minimal number of points are used along X.

Exploring the Effect of the Non-Linearity

In order to explore the effects of the non-linearity, we will perform two simulations: one with dispersion only, and another with dispersion and non-linearity. The results from the first will show the pulse broadening due to the dispersion; the results from the second will show the effect of the non-linearity.

The Linear Waveguide

To disable the non-linearity, temporarily set the variable `Chi3_ref` to `0.085*0`. Run the simulation with an **Output Prefix** of `linear`. Once the simulation has completed, open the file `linear.ptm` to see the time response of the linear system.

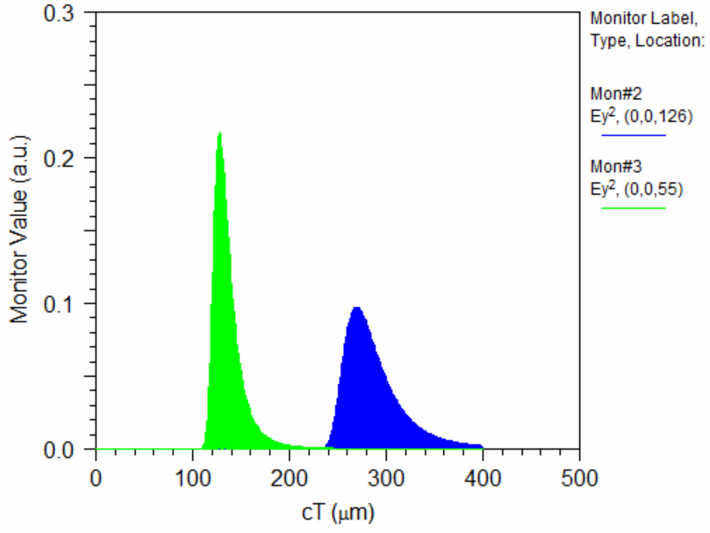


Figure 1: The time monitor response of the linear waveguide. Note that the pulse width changes as the pulse propagates down the waveguide.

The Non-Linear Waveguide

Next, reset the value of the variable `Chi3_ref` to `0.085`. Run the simulation with an **Output Prefix** of `nonlinear`. Once the simulation has completed, open the file `nonlinear.ptm` to see the time response of the non-linear system.

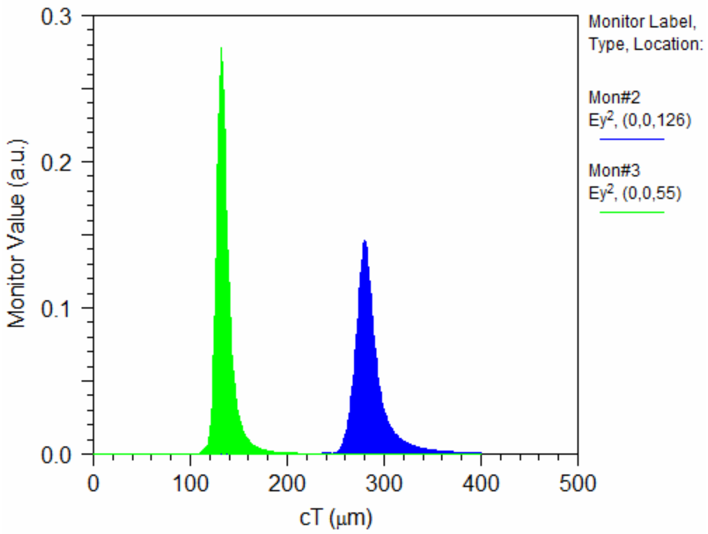


Figure 2: The time response of the nonlinear waveguide. Note that the pulse width stays constant as the pulse propagates down the waveguide.

The non-linear case pulse is more “narrow” and does not disperse as the linear case does, which is consistent with a soliton system and agrees with the reference. Due to the non-linearity, the pulse is not dispersing in time.

Tutorial 7: Using Q-Finder to Automate Cavity Mode and *Q* Calculations

This advanced tutorial illustrates the use of Q-Finder, an automated utility for computing cavity modes and *Q* factors, through several examples. A basic straightforward example will be given to illustrate the use of Q-Finder, and then a simple scanning example will be performed

This example uses the same structure as Tutorial 4, but will use Q-Finder to calculate the mode resonance, profile, and *Q*. The files associated with this tutorial can be found in the subdirectory:

```
<rsoft_dir>\examples\FullWAVE\Tutorial7
```

Though not necessary, it is recommended that you read through Tutorial 4 before proceeding with this tutorial in order to better understand the *Q* computation process. Also, a detailed description of *Q* calculations and the use of Q-Finder can be found in Chapter 8.

Structure and Simulation Overview

This structure is composed of a hexagonal array of air holes in a silica slab (n=3.37). The lattice constant *a* used is 0.4 µm and a hole diameter of 0.65a. To create the defect, the center hole is removed. Defect modes will be looked for around a wavelength of 1.5 µm.

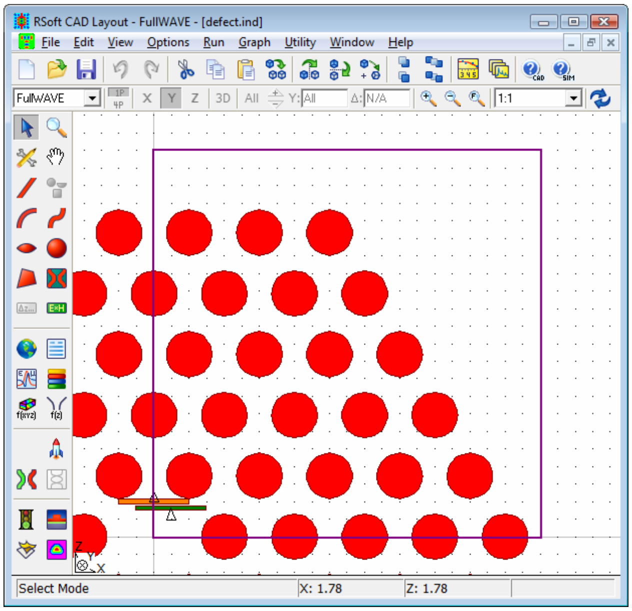


Figure 1: The PBG structure with defect shown. This is the same structure used in Tutorial 4.

The procedure to create this structure is given in Tutorial 4 and will not be repeated here. Open the file `defect.ind` and note the following:

- *Field monitor*

A field monitor has been placed within the defect region of the crystal at a position with low symmetry. This field monitor will be used by Q-Finder to compute the spectral response of the cavity.

- *Energy density monitor*

  Unlike the structure created in Tutorial 4, this structure does *not* have an energy density monitor. This is because the FHA method is sufficient to find the *Q* value for this application, and the energy decay method is not needed. Were the energy decay method to be used, an energy density monitor would be needed. See Section 8.C.2 for more details.

- *Launch settings*

  An impulse excitation type has been set up at a position of low symmetry.

- *Boundary settings*

  In Tutorial 4, it was found that the TM mode of interest at 1.575 μm exhibited even symmetry along the X axis and odd symmetry along the Z axis. The boundary conditions in this file have been set accordingly in order to speed up the computations.

Q-Finder automates the procedure of finding the resonance, mode profile (if applicable), and *Q* value.

Using Q-Finder

Open Q-Finder via the CAD menu (Utilities/Q-Finder). The default settings in the Q-Finder dialog will be used in this simulation, but note the following settings:

Parameter	Value	Description
Center	`1.0/1.5`	Sets the initial frequency to begin looking for resonant modes in μm^{-1}. This is equivalent to a wavelength of 1.5 μm.
Propagation lengths (2^N*dt)	`12,15`	Sets the length of the series of simulations used to find the resonant frequency and mode profile. The three values set here indicate that three simulations will be performed with stop times of `2^12` and `2^15`, times the **Time Step**.
Find Mode Volume	`enabled`	Enables the mode volume computation.
Spectral analysis method	*FHA if possible*	The FHA method will be used to compute the *Q* factor.

To start the simulation, enter an **Output Prefix** of `qfact` and click **OK**. The Q-Finder simulation window will open and display the progress of the simulation. During the computation, a series of FullWAVE processes will open and close as the simulation progresses. The basic process is as follows:

- Several FullWAVE simulations of increasing length are performed in order to determine the resonant frequency and mode profile of the cavity mode of interest. The number and length of the simulations are set by the **Propagation Lengths** setting. The simulations use the **FDTD Continue** option to launch the mode found in one step into the next simulation in order to produce a “clean” mode profile.

- Once the simulations are completed, the final spectrum is analyzed using the FHA method to determine the *Q* value.
- During this last step, the mode profile is calculated (this can be disabled by selecting the **Skip final mode profile calculation** option), and, as requested, the mode volume will be computed. A parameter called Mode Purity is also computed. This parameter is described below.

This method finds a *Q* value of 3363 at a resonant frequency of 0.63497 μm^{-1}, and a mode volume of 0.12.

Note that a convergence study has not been performed, and so the value obtained may not be accurate. Also, since this is not a converged value, the *Q* value found here might differ from values you find due to improvements made to the FullWAVE and Q-Finder algorithms. This underscores the importance of always performing a convergence study, as described below, to ensure accurate results.

Mode Purity

Q-Finder ouputs a mode purity which is a measure of how much of the field within the current mode profile is associated with the desired frequency. A value near 100% indicates that the mode is near perfect; a much lower value indicates that there is substantial energy from another peaks present in the mode profile.

Using the Energy Decay Method

The above calculation did not use the energy decay method to compute the *Q* value. This method is only necessary when performing a partial *Q* calculation, or when checking the FHA results. This method will not be illustrated here.

Basic Scanning and Optimization

Like all tools in RSoft’s Photonic Component Design Suite, Q-Finder can be operated from MOST for scanning and optimization. This example will use the same structure as above, but adds a central hole to the defect region. The size of this hole will be scanned using MOST and the *Q* will be calculated.

Users not familiar with MOST can find additional details on scanning in the MOST manual.

Structure and Simulation Overview

Open the file `defect_scan.ind` and note that an additional blue hole is located in the defect region with a radius defined by the variable `RadDef`. Also, the mode profile and mode volume output have been disabled in the Q-Finder dialog to speed up the simulation. Finally, the option to **Seed center frequencies in scan** is enabled.

Using MOST

Click the MOST button on the left toolbar to open the MOST dialog, and note the following:

- *Simulation Tool*

  The simulation tool is set to *FullWAVE Q-Finder*.

- *Variables*

  One variable, `RadDef`, has been defined and will scan from a value of `0` to `0.2` in 11 steps.

- *Measurements*

  Under the measurements tab, the measurements that Q-Finder supports can be seen. Here two of the most important have been selected: `qfind_nu_c` and `qfind_q_fha` which return the complex resonant frequency and *Q* value of the mode (found via the FHA method) respectively.

Click **OK** to start the scan.

By default MOST opens a Q-Finder window which will in turn open several FullWAVE windows. Since multiple windows can be distracting when running long calculations, it can be beneficial to suppress them. The *Q-Finder* window can be suppressed by selecting **Window** = *Hide* in the MOST dialog, while the FullWAVE windows can be suppressed using the corresponding setting in the Q-Finder dialog. If needed, the information which would have appeared in the Q-Finder window can always be found in the log files which will appear in the `<prefix>_work/raw` directory where `<prefix>` is the **Output Prefix** set for the scan.

When the scan has finished, the results can be examined using the DataBROWSER.

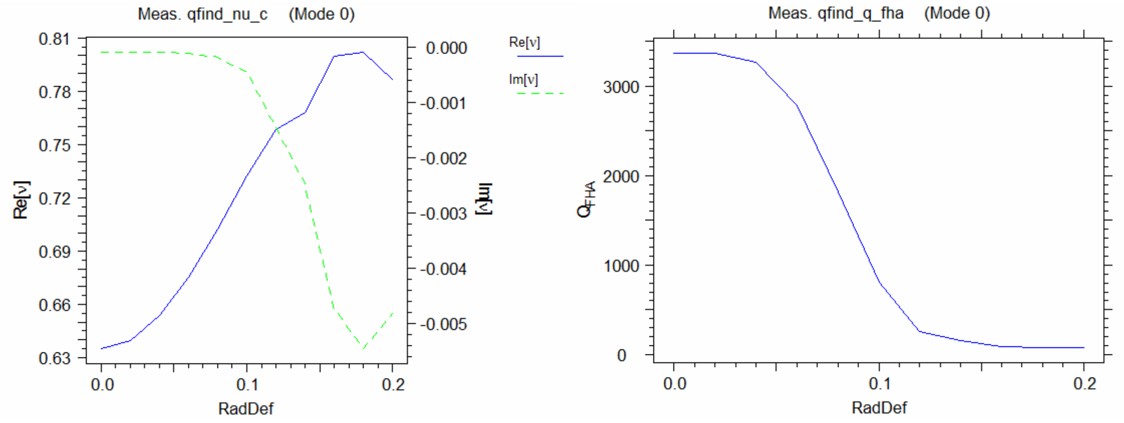


Figure 2: The results of the scan over `RadDef`. a) the resonant frequency of the cavity mode b) the calculated Q values.

Areas for Further Exploration

There are several areas that can be explored further in this case. They are:

- *Perform a Convergence Study*

  As always, the effect of grid sizes on the simulation results should be studied. The above computations, especially when finding the resonances, should be repeated for smaller grid sizes, and the accuracy compared: if comparable accuracy is obtained, the results are accurate; if there is a large difference, then successively smaller grids should be used until accurate results are found.

Similarly, the size of the structure should be studied: a larger supercell that contains more periods of the lattice should be used: if the results are similar, then the smaller structure is adequate; if not, larger structures should be used until accurate results are found.

- *Longer simulations for better frequency resolution*

  The frequency resolution is directly related to the length, or number of time steps, in the simulation. Try rerunning the mode computation with a greater stop time and explore this relationship. A good way to do this is to increase the exponent for the power of 2 used to define the stop time.

- *Try the energy decay method*

  Though not necessary, it is possible to use the energy decay method for this structure. Enable the **Measure Q by energy decay** option and rerun the simulation. This method requires the use of an energy density monitor that samples a large portion of the simulation domain. Though this method is slower, it should give similar results.

Tutorial 8: Using a Non-uniform Grid

This tutorial describes the use of a non-uniform grid with FullWAVE, and illustrates the computational benefits it can offer over standard uniform grid simulations. This tutorial will cover the settings for various advanced grid control options required to correctly setup the non-uniform simulation grid for the problem. It also presents a comparison between the results obtained with uniform and non-uniform grid and shows the advantages that non-uniform grid can offer in most problems. User familiarity with the CAD interface is assumed.

The files associated with this tutorial can be found in the subdirectory:

```
<rsoft_dir>\examples\FullWAVE\Tutorial8
```

Structure Overview

Open the file `nonuniform.ind`. This structure is a slit (gray segment) in a gold (Au) metal slab (yellow segment). The parameters which define the material properties of the gold segment are defined in the material table. The thickness of the metal slab is 0.1 μm and the slit diameter is 0.4 μm. The variables for the metal slab thickness have been defined to be `T_metal` and `W_slit` respectively.

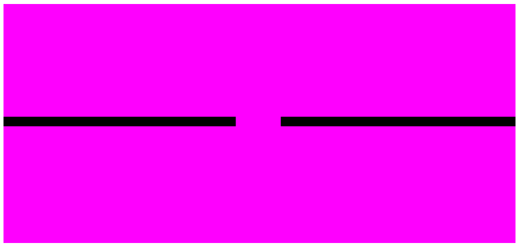


Figure 1: The metal slab with a slit used in this tutorial. The black region represents the metal.

Advantages of Using a Non-Uniform Grid for this Structure

The benefits of using a non-uniform grid for this structure are due to the vastly different length scales present: an ultra thin (sub-wavelength) metal slab in a relatively large computational domain. In order to correctly resolve the field in the smallest structure, the thin slab, a minimum of 10 grid points would be needed in the thickness of the slab, resulting in a required Z grid size of 0.1/10 = 0.01 μm. When using a uniform grid, this fine grid size would have to be used throughout the simulation domain, even in the air regions above and below the slab. This would be an unnecessary waste of computational resources as this empty space region can be resolved with a much coarser grid size of $\lambda/15 = 0.1\mu m$. If a non-uniform simulation grid were to be used, this extra computational burden could be avoided by creating an efficient grid which is fine where essential and much coarser in other areas.

While this tutorial presents one typical example of a situation where a non-uniform grid can offer large computational benefits; however, there is a wide range of applications where the use of a non-uniform grid can be useful.

Using a Non-Uniform Grid

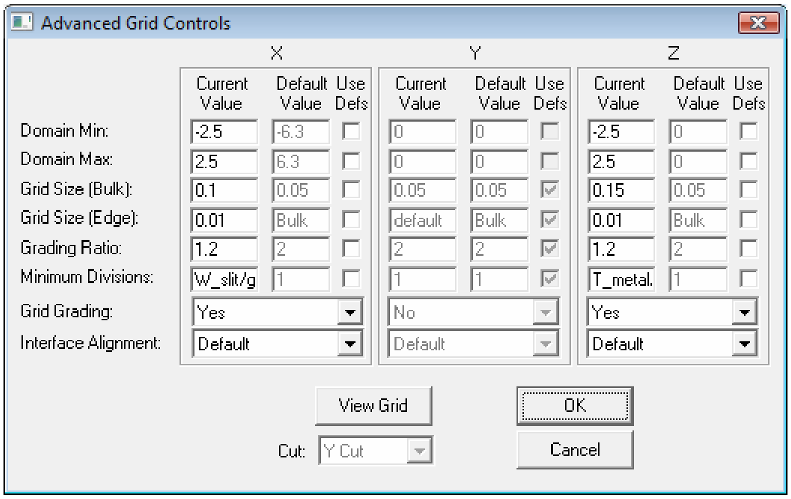
To enable the use of a non-uniform grid set **Enable Nonuniform** in the FullWAVE simulation parameters dialog.

Choosing Non-Uniform Grid Parameters

It is very important to use the correct simulation grid for any problem. When using a uniform grid, this means that the grid size needs to accurately resolve the finest field structure in the problem, which might be the wavelength if there are no sub-wavelength structures. When using a non-uniform grid, there are many more parameters to be set.

The non-uniform grid options are set in the Advanced Grid Options dialog which can be opened via the **Grid Options...** button in the FullWAVE Simulation Parameters dialog.

For a more complete look at these parameters, please see the CAD manual.



Advanced Grid Controls

	X			Y			Z		
	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs	Current Value	Default Value	Use Defs
Domain Min:	-2.5	-6.3	□	0	0	□	-2.5	0	□
Domain Max:	2.5	6.3	□	0	0	□	2.5	0	□
Grid Size (Bulk):	0.1	0.05	□	0.05	0.05	☑	0.15	0.05	□
Grid Size (Edge):	0.01	Bulk	□	default	Bulk	☑	0.01	Bulk	□
Grading Ratio:	1.2	2	□	2	2	☑	1.2	2	□
Minimum Divisions:	W_slit/g	1	□	1	1	☑	T_metal.	1	□
Grid Grading:	Yes			No			Yes		
Interface Alignment:	Default			Default			Default		

Cut: Y Cut

Figure 2: The Advanced Grid Controls dialog where the options that control a non-uniform grid are set.

This dialog allows the grid options to be set for each coordinate direction.

The first three fields, **Domain Min**, **Domain Max**, and **Grid Size (Bulk)** are equivilent to the options located in the FullWAVE Simulation Parameters Dialog. The **Grid Size (Bulk)** option has been renamed since it represents an "average" grid size to be used in the bulk regions that are sufficiently far from material interfaces.

The options are set as follows:

- **Grid Size (Bulk)**

  These fields are set to the relatively coarse values of `0.1` and `0.15` respectively, which roughly correspond to λ/10.

- **Grid Size (Edge)**

  These fields specify the grid size to be used near material interfaces, and should therefore be set to the resolves the finest structural feature along each coordinate. Along Z it is set to a value that accurately resolves the metal slab thickness, and along X it is set to a value that accurately resolves the slit width.

- **Minimum Divisions**

  This field defines the minimum number of grid cells to place within a material region. In this case it is set as a function of the **Grid Size (Edge)** values such that there are 10 cells across the metal slab thickness and 40 cells across the slit.

These settings ensure that a fine grid size of 0.01 µm where required and a coarser grid size of 0.1 µm elsewhere.

Since abrupt grid size transitions between adjacent cells can create inaccuracies in the simulation results, the **Grid Grading** options have been enabled. This option enables a gradual transition from the fine grid size at interfaces to the coarse grid size in bulk regions. This transition is controlled by the **Grid Ratio** option which sets the smoothness of the transition. Here we have chosen a value of 1.2, though any number between 1 and 1.5 should work. Note, however, that small values of **Grid Ratio** will increases computational burden outweighing the benefits of the non-uniform grid, and large values (>2) might create inaccuracies in the simulation.

Viewing the Non-Uniform Grid

It is recommended that the user view the simulation grid before performing a simulation to ensure that the grid accurately represents the structure. If any potential problems in the grid are seen, the user should modify the grid settings as necessary and recheck the grid.

To view the grid, click the **View Grid** button in the FullWAVE Simulation Parameters dialog.

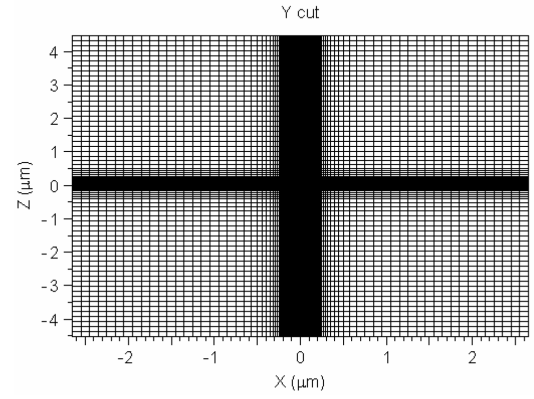


Figure 3: The non-uniform grid for the metal slit structure. Note the increased density of grid points for better resolution along the slab and in the slit region.

Simulating with a Non-Uniform Grid

In this simulation, the power transmission through the metal slit will be measured. A plane wave source has been created, as well as a monitor to record the time-averaged power.

To run the simulation, enter an **Output Prefix** of `nonuniform` and click **OK**. After the simulation has completed, open the file log.txt in the working directory (this can easily be done via the View\Open Log CAD menu option) and note the simulation time and memory used.

The computer used to create this example took ~0.2 minutes to complete this simulation and used about ~1 MB of simulation memory (not including base memory which will be the same for most simulations). Users will obviously obtain different results depending on the computer system used. The comparison between the uniform and non-uniform simulations performed on the same computer should, however, be similar.

The results output by the monitor show a transmitted power of around 0.0164 and is shown in Fig. 4a.

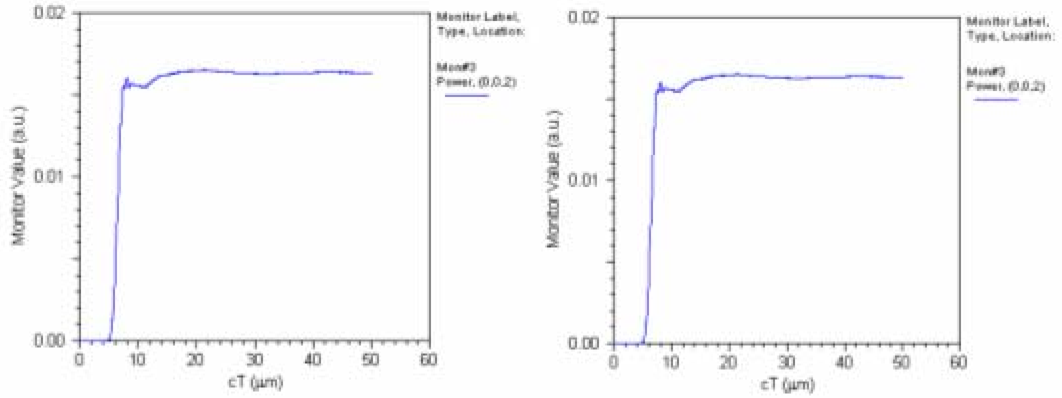


Figure 4: The monitor output when using the a) non-uniform grid, and b) uniform grid.

Simulating with a Uniform Grid

Next we will perform the same simulation on a uniform grid so that we can compare the results. To do this, the **Enable Nonuniform** option in the FullWAVE Simulation Parameters dialog needs to be disabled. Also, since a the uniform grid needs to resolve the complete structure, a fine grid needs to be used. Change the **Grid Size** to `0.01` for both X and Z.

To run the simulation, enter an **Output Prefix** of `uniform` and click **OK**. The simulation will run, though much slower than the non-uniform case. When the simulation completes, open the log file again and note the simulation time and memory used.

The computer used to create this example took ~4 minutes to complete this simulation and used about ~18 MB of simulation memory (not including base memory which will be the same for most simulations). Users will obviously obtain different results depending on the computer system used. The comparison between the uniform and non-uniform simulations performed on the same computer should, however, be similar.

The results output by the monitor show a transmitted power of around 0.017 and is shown in Fig. 4b.

Comparing the Results

The results can easily be compared:

- *Total Simulation Time:*

The use of a non-uniform grid decreased the total simulation time from about 4 minutes to 0.2 minutes, resulting in a simulation that is about 20 times faster. Remember that results may vary depending on the computer system used. but a similar speedup should be seen.

- *Required Simulation Memory:*

  The use of a non-uniform grid decreased the required simulation memory from about 18 MB to 1 MB. This is because the non-uniform grid uses less grid points than a uniform grid.

- *Accuracy of the result:*

  The steady-state results from the monitors shows that the result from the non-uniform grid is only 3% different than the result found with the uniform grid, even though the non-uniform grid was coarse in the bulk regions.

Note that the comparison of a non-uniform and uniform grid are highly problem dependent. Some problems will see a dramatic increase in speed and memory usage, while others might only see marginal improvements.

Areas for Further Exploration:

- *Experiment with the grid options*

  Try changing the grid options to see how the resulting grid changes.

- *Perform a convergence study with the grid options*

  While the simulation results found above were close, it is possible to make them closer by tweaking the grid options. Try to improve the accuracy of the non-uniform grid without substantially increasing the simulation time.

Tutorial 9: Modelling Materials with Optical Gain

This example illustrates the definition and usage of materials with optical gain in FullWAVE and is based on a problem described in the following reference:

> A Taflove, *Computational Electrodynatmics: The Finite-Difference Time-Domain Method*, Third Edition (Artech House, Norwood, MA, 2005). Chapter 9, Section 9.7.

Structure Overview

Open the file `<rsoft_prefix>\examples\FullWAVE\Tut9\gain-tf1.ind`. The reference describes a homogeneous GaAs gain material characterized b a single Lorentzian gain resonance $\lambda_0 = 0.89$ μm, a relaxation time constant T_2=0.07ps, refractive index n=3.9, and $\Delta\sigma_{0g}$=-5000S/m. The material is assumed to be only dispersive in gain, and the standard linear ε is considered to be independent of frequency.

Gain Material Definition

To view the material definition, open the Material Editor and look at the material GaAs.

Additional details about the definition of materials can be found in Chapter 8 of the CAD manual.

Since this material is only dispersive in gain, **Epsilon at infinity** has been set to `3.9*3.9`. The gain dispersion parameters as defined in the reference needed to be converted from SI units into RSoft units as follows. Firstly, the input parameters are converted from SI units to RSoft units. Since time in RSoft units is given in μm, we have $T_{2(RSoft)} = c^* T_{2(SI)}$. Also after some manipulations:

$$\Delta\sigma_{0g(RSoft)} = \frac{\Delta\sigma_{0g(SI)}}{c\varepsilon_0}$$

The dispersion parameters for the gain material are then given as follows:

Parameter	Value
$\Delta\sigma_{0g(RSoft)}$	$\Delta\sigma_{0g(SI)}/c\varepsilon_0$
a_k	1
b_k	$2/T_{2g(RSoft)}$
c_k	$(2\pi/\lambda_0)^2$

Also note that the **gain term** option has been selected for this resonance.

Performing a Simulation

Run the example and open the FFT monitor results. The amplification spectrum shown in Fig. 1 (modified to change frequency units and display amplitude) agrees well with the results in the reference.

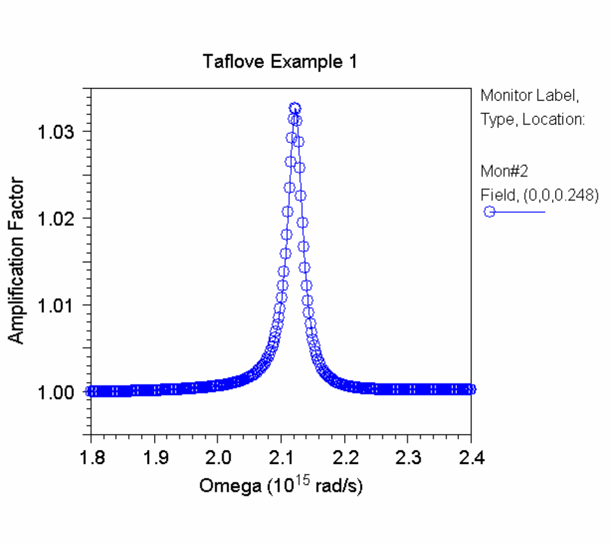


Figure 1: The obtained amplification spectrum.

Tutorial 10: Simulating Surface Plasmon Effects

Surface plasmons are a unique characteristic of optical propagation along the boundaries of metals. In certain instances, an optical field incident on a metal induces what is known as a polariton which can lead to the induction of an electrical field in the metal. The photonic and electronic field can then propagate along the surface of the metal inducing each other along the length of the material. Surface plasmons are of interest due to their high carrier rate (100 THz or so) on metal wires, which could lead to higher speeds on electrical devices.

In this tutorial we will set up two examples of utilizing FullWAVE to simulate a surface plasmon device. Familiarity with the CAD tool and material dispersion is assumed.

Example 1: Simple Plasmonic Waveguide

This example consists of a simple 2D plasmon waveguide and is loosely based on:

> R. Charbonneau, N. Lahoud, G. Mattiussi, and P. Berini, “Demonstration of integrated optics elements based on long-ranging surface plasmon polaritons,” Optics Express **13** 977 (2005)

While the reference is based on a full 3D device, this section will be based on a 2D structure for purposes of simplicity.

Structure Overview

Open the file `<rsoft_prefix>\examples\FullWAVE\tut10\waveguide.ind`.

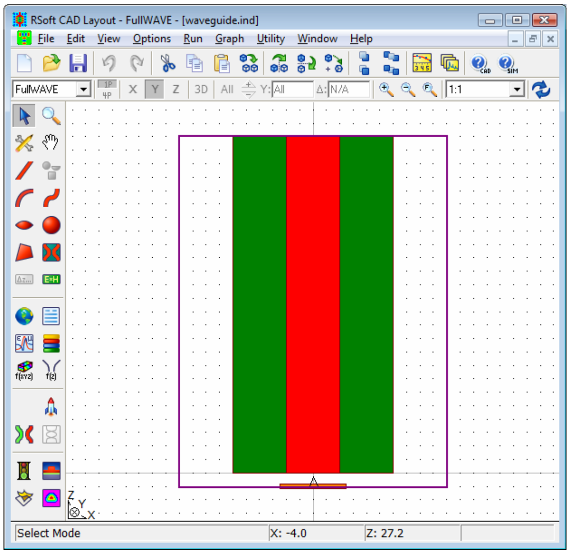

RSoft CAD Layout - FullWAVE - [waveguide.ind]
File Edit View Options Run Graph Utility Window Help
FullWAVE
X Y Z 3D All
Y: All
Δ: N/A
1:1
Select Mode
X: -4.0
Z: 27.2

Figure 1: The simple 2D metallic structure in the CAD.

Note the following about the design:

- *Waveguide & Background*

  The waveguide has a width of 4 μm and a length of 25 μm. Its material properties have been set to Gold as defined in the Material Editor. The background has been defined with an index of 1.444.

- *Monitor*

  A monitor has been defined to record the final steady-state field for later analysis using the DFT frequency analysis option.

- *Launch Field*

  The launch field has a Gaussian profile and a width of 5µm. Also, a small gap has been placed between the launch field and the waveguide so the field can couple into the surface plasmon mode.

Running the Simulation

Open the FullWAVE Simulation Parameters dialog, define an **Output Prefix** such as `run`, and press **OK** to start the simulation. The propagation of the bound surface plasmon field along the metal-dielectric interface is clearly seen.

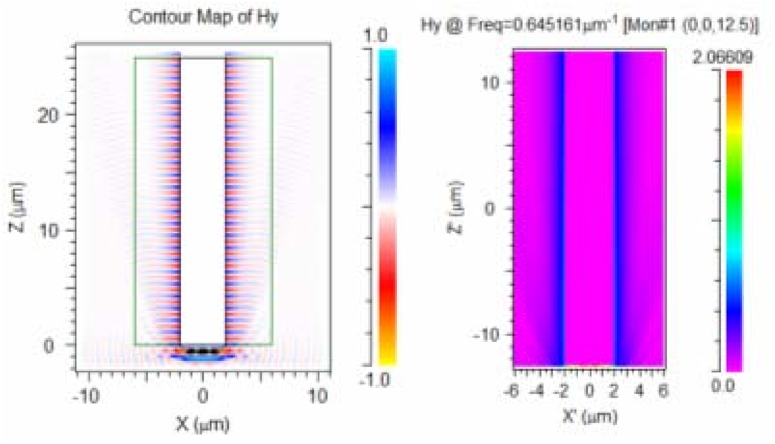


Figure 2: a) Finished simulation for the simple 2D plasmonic waveguide, and b) DFT-Spatial response of plasmon waveguide.

The steady-state field information (Fig. 2b) can be seen by opening the file `run.m2_f1_hy.pcs`, and again the plasmon field is clearly traveling along the metal interface.

Example 2: Propagation through a Periodic Array of Metal Slits

This example will study propagation through a metal slit. This is a very common application for filters and the transmission through these apertures has been an active area of research. It is based on:

> Y. Xie, A. Zakharian, J. Moloney, and M. Mansuripur, "*Transmission of light through a periodic array of slits in a thick metallic film*," Optics Express **13**, 4485 (2005)

In this reference, it was found that surface plasmons can be a dominant effect and will substantially reduce transmission through the aperture. This example will first determine the transmission at a single wavelength (1 µm), and then study the response of the system as a function of the periodicity and metal thickness at this wavelength.

Structure Overview

Open the file `<rsoft_prefix>\examples\FullWAVE\tut10\slit.ind`.

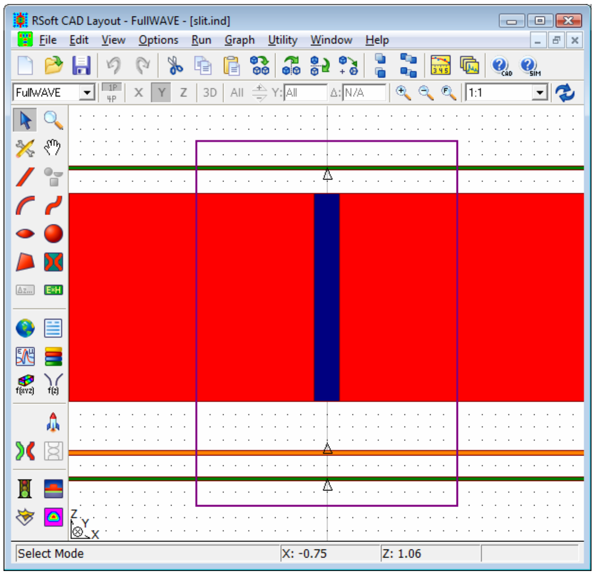

RSoft CAD Layout - FullWAVE - [slit.ind]
File Edit View Options Run Graph Utility Window Help
FullWAVE
X Y Z 3D All Y: All Δ: N/A
1:1
Z
Y
X
Select Mode
X: -0.75
Z: 1.06

Figure 3: The metal slit structure in the CAD.

Note the following about the design:

- *Geometry*

  The slit is cut into a silver layer that is 0.8 μm thick, and its material properties have been set to Silver as defined in the Material Editor. The slit itself is 0.1μm thick, and has an index of 1 (air). The background has also been defined with an index of 1 (air). Also, this structure represents a periodic array of slits. This is accomplished by setting the X domain equal to one period (1 μm), and using periodic boundary conditions along X.

- *Monitors*

  Two monitors have been defined: one to measure the transmitted power, one to measure the reflected power.

- *Launch Field*

  The launch field is a plane wave and is positioned between the metal plate and the monitor used to measure reflection.

Running the Simulation

Open the FullWAVE Simulation Parameters dialog, define an **Output Prefix** such as `run`, and press **OK** to start the simulation. The propagation of the surface plasmon modes through the slit are clearly seen.

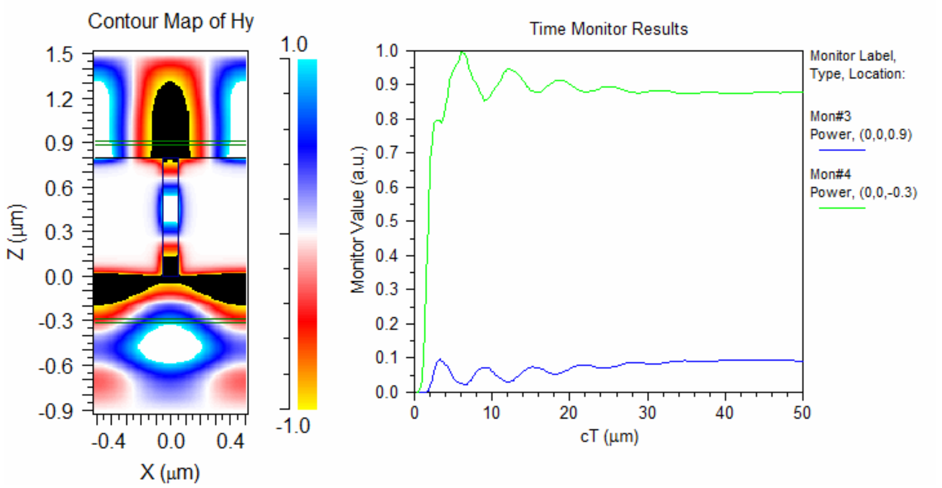


Figure 4: Simulation results: a) field profile at end of simulation, and b) time monitor results which show the transmission and reflection as a function of time. The field results show monitor results show that the simulation has reached steady-state.

Performing a Parameter Scan

As reported in the reference, the transmission characteristics of this device can be found to be periodic as a function of the periodicity and thickness. RSoft's scanning tool *MOST* can be used to perform a CW scan to see how this device operates. Click the **Launch MOST** button in the left toolbar and note that a scan over the variable `Period` and `Thickness` has been set up. Press **OK** to start the scan.

This scan has many data points and might take a long time complete. It might be best to run this overnight and check the results in the morning.

When the scan is complete, the results from the transmission monitor agree well with the reference.

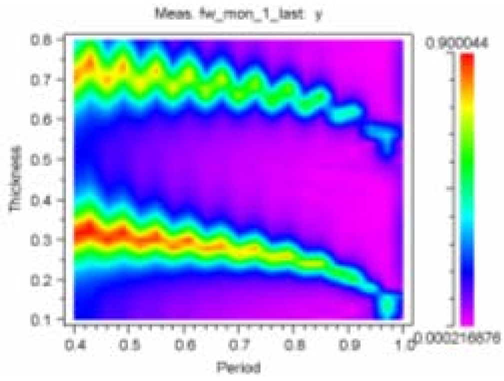


Figure 5: Scan results of periodic array of holes in a metal.

When the period of the device matches that of an integer-multiple of the surface plasmon resonance, the transmission goes to zero. Thus, the surface plasmon effect is parasitic in nature for this device.

Tutorial 11: Anisotropic Wave-Plate

Anisotropic materials affect propagating electromagnetic fields by shifting the phase between the TE and TM polarizations. This tutorial demonstrates this effect using the birefringent material calcite. The associated file for this tutorial is `calcite.ind` and is located in the subdirectory:

```
<rsoft_dir>\examples\tutorial\FullWAVE\Tut11\
```

Structure Overview:

Open the file `calcite_fw.ind` in the CAD interface.

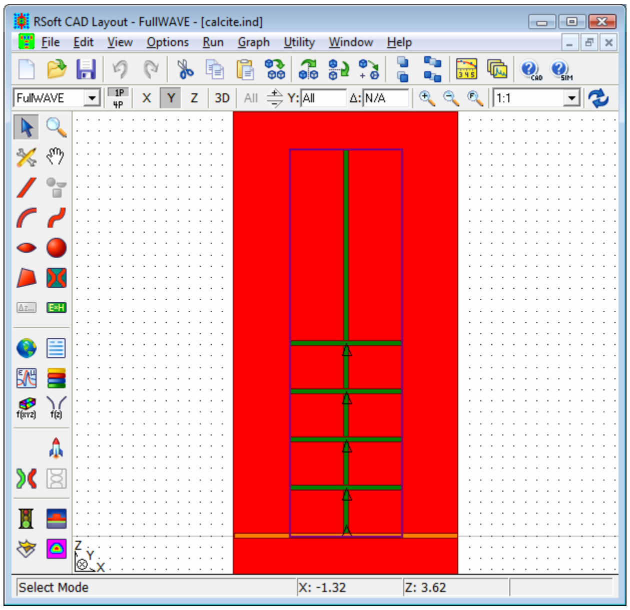


Figure 1: The completed structure in the CAD interface. The red segment represents calcite while the green segments are monitors used to make measurements during the simulation.

Note the following about the design:

- *Structure*

  This structure represents a bulk slab of calcite. The simulation domain is smaller than the calcite segment, and so FullWAVE will interpret this segment as infinite along both X and Z.

- *Material Definition*

  The segment's material properties are set to the custom material *Calcite*. This material has been defined to be anisotropic with an extraordinary (N_e) and ordinary (N_o) refractive index defined by the symbols `Ne` and `No` respectively. For convenience and measurement purposes, the variable `L_pishift` has been defined and set to $\lambda/(N_o\text{-}N_e)/2$ which is the analytically predicted length at which the phase difference between the two polarizations will be 180°.

  In order to include anisotropic materials in the simulation,

- *Launch Condition*

  The launch field has been set to the fundamental fiber mode for this structure at a wavelength of 590 nm. The launch polarizer has been used to rotate the launch polarization 45 degrees so that both Ex and Ey are equally excited. Click the **Polarizer...** button in the Launch Properties dialog to see this setting.

- *Monitor Definition*

  There are a total of five monitors defined in the design file. The four monitors that are perpendicular to the propagating direction (i.e. horizontal in Fig. 1) have the *Overlap* **Single-Value** outputs enabled. They have been positioned at Z values of `L_phishift/4`, `L_phishift/2`, `3*L_pishift/4`, and `L_pishift` to record the overlap integral between the launch field and the local field at these positions. These positions have been chosen to illustrate the phase difference between the E_x and E_y field components; for example at Z = `L_pishift`, the overlap integral should be 0 since the polarizations (field components) are completely out of phase.

  The monitor that is parallel to the propagating direction (i.e. vertical in Fig. 1) has the *Ex* and *Ey* **Spatial Output** options enabled. The **Spatial Format** is set to *Amp/Phase* and will be used to record the phase of the field components along the propagation direction.

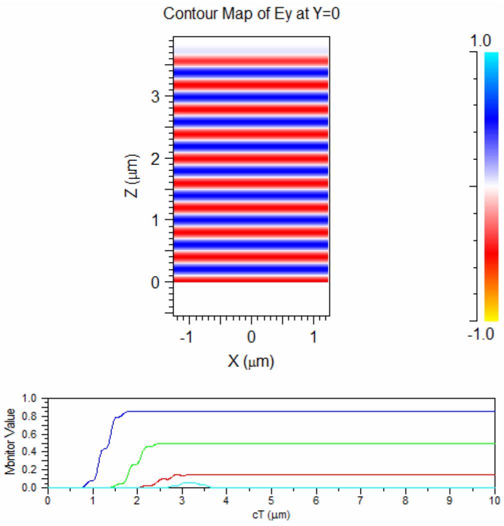


Figure 2: The simulation results.

Performing a Simulation

Click the **Perform Simulation** button in the left toolbar and note the settings. Enter an **Output Prefix** of `waveplate` and click **OK** to start the simulation.

As the simulation progresses, the the results of the overlap monitors are displayed in the simulation window (Fig. 2). As expected, after a propagation length equal to the analytically calculated pi-shift length (`L_pishift`) the overlap is indeed 0.

Examining the Phase

To view the phase along the propagation direction, we will use the spatial output of the vertical monitor. Click the **View Graphs** button and select the file `waveplate_m2_ex.pcs`. This file contains the amplitude and phase of the E_x field component along the monitor. Click the **Edit Plot** button in the top toolbar and change the plotting commands to be:

```
        /tt"Ex @ [Mon\#2 (0,0,1.71512)]"
        /xl"Z' (&mm)"
        /yl""
        /ls1
        waveplate_m2_ex.dat /i0
        waveplate_m2_ey.dat /i0
```

which will plot the phase of both E_x and E_y on the same graph.

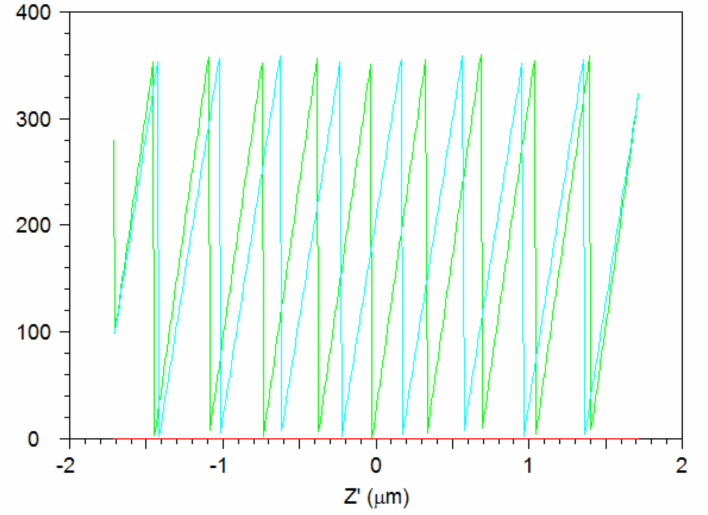


Figure 3: Plots of the phase of E_x and E_y along the propagation direction. The beating caused by the phase shift is clearly visible.

Areas for Further Exploration

Several areas to explore further are:

- *Use a different simulation tool!*

  This file can be used with either RSoft's BeamPROP, FullWAVE, DiffractMOD, or ModePROP to perform the same simulation. Users who are licensed for any of these tools can simply change the **Simulation Tool** in the Global Settings dialog and rerun the simulation. See the appropriate simulation manuals for a corresponding tutorial discussion.

Tutorial 12: Negative Refractive Index

In all known naturally occurring materials, the real part of the index of refraction is positive. However, as shown in [1], this is not a fundamental restriction. It is possible to create negative refraction with left handed materials – materials whose electrical permittivity and magnetic permeability are simultaneously negative in certain frequency bands. Recently, several artificially structured left-handed meta-materials have been introduced [2] and analytical analysis has shown surprising results [1,3]. The properties of these new materials are of some considerable interest.

FullWAVE is a very suitable tool for modeling these materials: the requirement for both a negative ε and μ implies a frequency-dependent dispersion model for both properties, which FullWAVE can include with ease. This tutorial will investigate the transmission characteristics of a finite beam incident on both a negative prism and lens with $\varepsilon = \mu = -1$.

References:

1 V. G. Veselago "The electrodynamics of substances with simultaneously negative values of ε and μ" Soviet Physics USPEKI **10**, 509 (1968).

2 D. R. Smith, Willie J. Padilla, D. C. Vier, S. C. Nemat-Nasser, S. Schultz, "A composite medium with simultaneously negative permeability and permittivity," Phys. Rev. Lett. **84**, 4184 (2000).

3 J. B. Pendry, "Negative refraction makes a perfect lens," Phys. Rev. Lett. **85**, 3966 (2000).

Example 1: Negative Prism

This example consists of a simple 2D prism, or wedge, with a negative index of refraction.

Structure Overview

Open the file `<rsoft_prefix>\examples\FullWAVE\tut12\wedge.ind` and note the following about the design:

- *Structure Geometry*

  The structure consists of two components to create a prism-like geometry. The material properties of both components are defined by the material *NegIndex* described below, and the background material is air.

- *Negative Index of Refraction Definition*

  The material *NegIndex*, which has been defined in the Material Editor, is assigned to both components in the design. This material consists of the same dispersive model for both Linear Epsilon and Linear Mu. These models can be seen on their respective tabs in the Material Editor, and are defined such that at the value of the **Free Space Wavelength**, both ε and μ are equal to -1.

  If needed, spend some time looking at the definitions in the Material Editor, the Symbol Table, and the discussion in the CAD manual on how linear dispersion is defined. To view the actual index used, click the **Display...** button on the appropriate tab in the Material Editor.

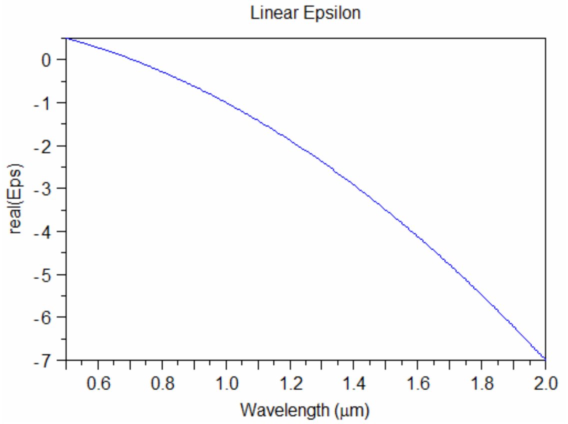


Figure 1: At a wavelength of 1 µm, ε = -1. The linear µ properties are exactly the same.

- *Using Magnetic Materials*

  To enable the use of magnetic materials in a FullWAVE simulation, note that the Use Magnetic Materials option in the Advanced Options dialog is enabled.

- *Launch Field*

  The launch field has a Gaussian profile and a width of 2 µ and has a wavelength of 1 µm.

Running the Simulation

Open the FullWAVE Simulation Parameters dialog and note that the simulation domain has been defined such that the negative material does not touch the PML. Click **OK** to start the simulation.

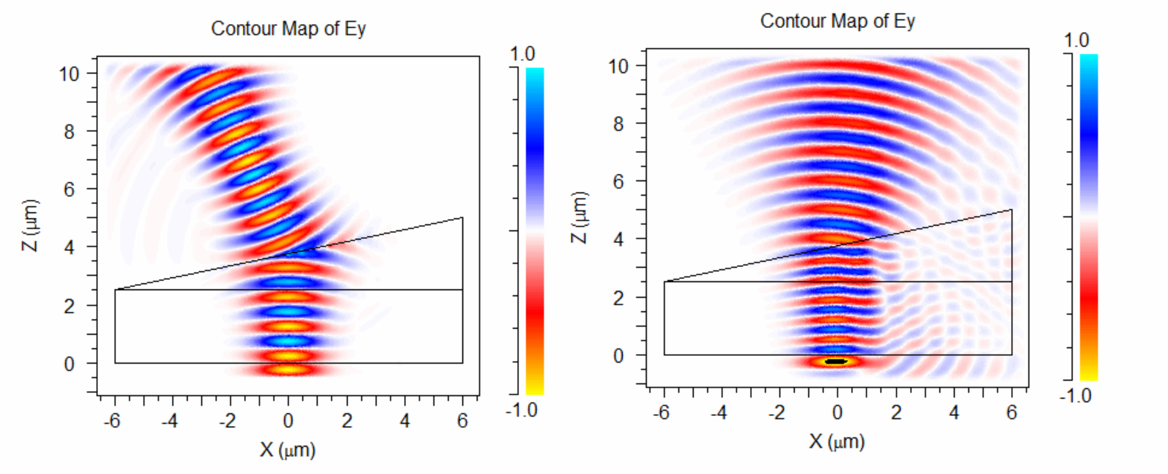


Figure 2: Simulation results for a) the negative prism, and b) the 'normal' prism.

In order to compare the results from the negative prism shown in Fig 2a to positive refraction, change the **Material Properties** for both components in the wedge to *Locally Defined*. The positive index in this case has been already defined to be ε = 2.2 and µ to be 0. Rerun the simulation to get results like those shown in Fig. 2b.

As expected, the refraction angle for the negative index case is tilted to the opposite side than that of the positive index material. Further comparison between the two results reveals some interesting phenomena. In the negative index case (Fig. 2a), the energy is forward while the field is moving toward the back of the material. Furthermore, note that the Gaussian beam disperses as expected after leaving the prism in the positive index case (Fig. 2b). However, in the negative index case (Fig. 2a), the field is focused after leaving the prism. Also note that the special case of $\varepsilon = \mu = -1$ represents a perfect impedance match ($Z = +1$) with free space, and, for all incident angles, the field pattern will exhibit minimal reflection at the interfaces.

Example 2: Negative Lens

This example will study the propagation of light through a lens composed of negative index material.

Structure Overview

Open the file `<rsoft_prefix>\examples\FullWAVE\tut12\lens.ind` and note the following about the design:

- *Structure Geometry*

  The structure consists of a concave lens object. Its material properties have been set by the material *NegIndex* described below, and the background material is air.

- *Negative Index of Refraction Definition*

  The material *NegIndex* is used to define the material properties of the lens. This is the same material that was used in Example 1 of this tutorial. Please see the discussion above for details.

- *Launch Field*

  The launch field has a Gaussian profile and a width of 4 µm and has a wavelength of 1 µm.

Running the Simulation

Open the FullWAVE Simulation Parameters dialog and click **OK** to start the simulation. Again, in order to compare the results from the negative lens shown in Fig 3a to positive refraction, change the **Material Properties** for both components in the wedge to *Locally Defined.* The positive index in this case has been already defined to be $\varepsilon = 3$ and μ to be 0. Rerun the simulation to get results like those shown in Fig. 3b.

As expected, the field is clearly focused by the negative concave lens but not by the positive lens.

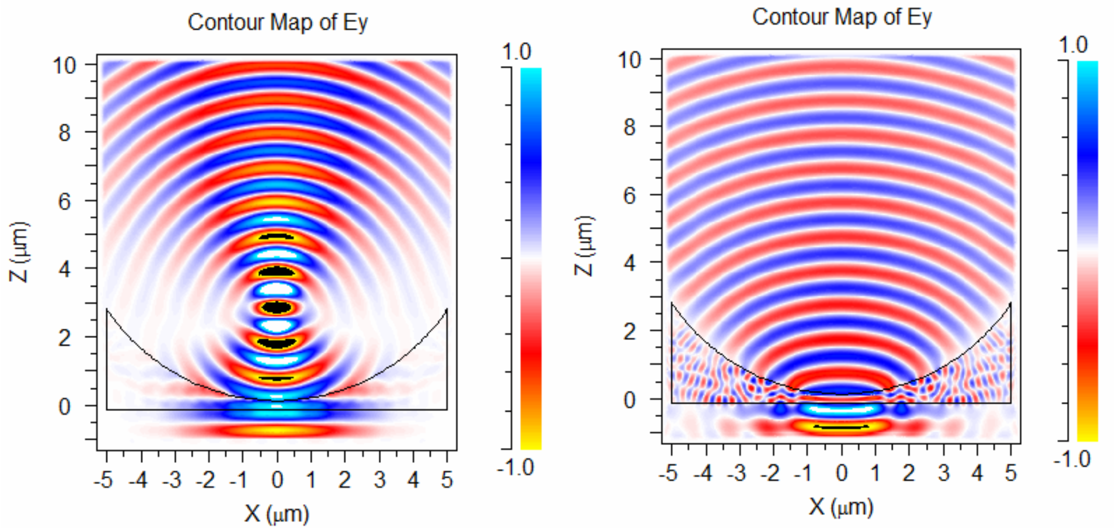


Figure 3: The simulation results for a) the negative lens, and b) the positive lens.

Appendix A

Tips and Traps

This appendix contains some advice on good habits and sources of confusion for novices and experienced FullWAVE users alike.

A.A. Common FullWAVE mistakes

- *Choosing an unphysically large time step results in an unstable simulation.*

  In order to satisfy the Courant condition given the desired grid sizes, the time step must be below a certain threshold. If the time step chosen is larger than this threshold, FullWAVE will return an error and return the threshold value. Either use the automatic time step or simply change the time step to be equal or less than the stability limit in order to ensure a stable simulation. Of course, you may have to reduce the time step further in order to achieve accurate results.

- *Attempting to perform a simulation in FullWAVE which exceeds the computational resources of your computer system.*

  The FDTD algorithm is quite rigorous, and solves for all six field components at all positions in space at all values of time. This is obviously advantageous as this information is usually interesting to the user, and also results in accurate simulations. However, it can result in a large drain on computational resources, as especially for 3D simulations. It is important to keep the computation limits of your computer system in mind when performing FullWAVE simulations. FullWAVE will prompt the user when a simulation is started if the simulation will require more system memory than is present on your system.

- *Misuse of the dispersion feature.*

  The dispersion feature in FullWAVE must be used properly in order to achieve correct results. A haphazard choice of the dispersion parameters can result in an unphysical material definition, and therefore an unstable simulation. Careful attention should be made to the choice of these parameters in order to ensure the material being simulation corresponds to the actual material. Furthermore, it may be necessary to reduce the time step to achieve a stable simulation.

- *Misunderstanding the output units.*

  It is important to keep in mind the units used by FullWAVE to store simulation data. This is important to keep in mind, especially when using power monitors. In this case, the displayed data is in units of power, but the stored data is the square root of this quantity. The data needs to be squared before using it in other calculations.

- *Properly understanding black/white areas in field plots.*

  The black and/or white regions in the field display in the simulation window indicate that the field values are above (white) or below (black) the indicated color scale. This can happen for two very different reasons. First, it can happen if the field is focused on itself, or if more than one beam of light cross (or one beam is reflected back on itself). This occurs since the field display is normalized to the input field. The second case this happens in is when the simulation goes unstable. The most common cause of this is that dispersive materials are used but the **Dispersion/Nonlinearity** option is not enabled. Simply enabling this option should resolve this issue. This can also occur with dispersive materials if the option is enabled but the time step is too large (see Section 2.B.2). Reduce the time step and retry the simulation. Finally, this can also be caused by the buildup of evanescent waves in a simulation where periodic boundary conditions are used. Use the PML Kappa options described in Section 6.C.1 to alleviate this problem.

A.B. Some Good FullWAVE habits to learn

Developing the following habits will save you trouble and time in the long run.

- *Use the* **Display Material Profile** *option frequently.*

  When setting up a structure, it is easy to make mistakes with geometry and refractive index information. Many apparently bizarre results are simply the result of simulating the wrong structure.

- *Check the launch condition and excitation type.*

  In order to obtain the desired results from a simulation, it is important to use the correct launch condition and excitation type.

- *Use enough grid points to resolve the smallest feature in the fields.*

  In order to effectively simulate the propagation of electromagnetic fields in your structure, you must choose grid sizes such that the smallest features in the fields. For most structures, this simply means that you must choose a grid size such that the wavelength of the field *in the structure* is resolved. However, if the structure has relatively small features, such as curved waveguides or thin films, you may have to reduce the grid sizes even further in order to resolves these features.

- *Use too few grid points at first; use too many at the end.*

General results for your structure can often be obtained with remarkably coarse grids (*which of course satisfy the above condition*). You can save time when doing initial exploratory studies by not worrying too much about accuracy. At the same time, when a structure includes fine features, the results can be quite inaccurate with an overly coarse grid. You have not completed your design project until you have confirmed the convergence of your results at a higher grid resolution.

- *Learn how to use the command line utilities included with FullWAVE*

  Since post-processing with these tools can be automated, you can achieve a lot of tedious work by writing a single command. You may even find uses for these utilities in your other work. These utilities are documented in the appendices of the CAD manual.

- *Learn the scripting capabilities of FullWAVE*

  While FullWAVE is designed to be used via a graphical interface (GUI), a long series of simulations can be tedious to perform through the GUI. These types of calculations can easily be automated through the use of a script. FullWAVE's scripting capabilities are based on the fact that simulations can be run from the command line. Because of this unique capability, FullWAVE can utilize the any native scripting language within the OS you are using. For instance, Windows users can use simple DOS batch files to run FullWAVE, while advanced users who may have standard scripting languages installed on their system such as PERL can write powerful scripts in these languages.

- *Read the WinPLOT manual.*

  WinPLOT's rather terse command syntax can be daunting at first but allows complex plotting features to be described quickly in just a few lines. Being able to quickly launch plots with old command files from the command line can be a huge time saver.

- *Use color scales to produce attractive plots.*

  WinPLOT is capable of producing many types of plots beyond the standard output usually displayed. Color scale files can be found in the installation directory, and are used to change the default scale used when displaying contour plots. These files are easy to create, and provide a quick way to customize the display of data for presentation purposes.

A.C. Tips for Improving Simulation Speed

These tips can, in some cases, considerably improve FullWAVE simulation speed. They can loosely be divided into two categories: using optimal simulation parameters such as grid sizes and domain settings and reducing the refresh and output rate of simulation data.

- *Use the single precision for numerical bookkeeping.*

  By default, FullWAVE uses single precision for all numerical bookkeeping during a simulation. While it is also possible to use double precision, which might be necessary in some cases, the default choice of single precision is recommended and will result in a reduction of memory used and therefore an increase in simulation speed.

- *Don't use an excessively large computational domain.*

  Nothing can be gained by simulating a part of your structure where no field is present. You can cut down on simulation time by making an intelligent choice for your computational domain. Also, make full use of the different boundary conditions in FullWAVE in order to maximize computational efficiency. For instance, you might be able to reduce the size of the computational domain through the use of periodic boundary conditions.

- *Use symmetric boundary conditions when possible.*

  Closely related to the above tip, it is highly beneficial to use symmetric boundary conditions when possible to reduce the size of the simulation domain. See Section 6.C.3 for more information about choosing the correct symmetric boundary conditions.

- *Use optimal values for the simulation grid steps..*

  While it is absolutely necessary to validate your results using a fine grid, it is not necessary to use an even finer grid for simulation. The largest grids that produce accurate results should be used. Using fine grid sizes that do not result in higher accuracy only create longer simulations.

- *Increase the update time for long simulations.*

  It is usually unnecessary to have FullWAVE update the graphical display of the fields during a long simulation. While it can be beneficial to watch the fields evolve in time for short simulations, the continual update of the simulation window steals valuable CPU time from the actual simulation. Increase the value of the **Update Time** in order to avoid making a long simulation even longer.

- *Disable Energy Density and Power Monitors When Not Needed*

  These types of monitors require post-processing of field information for each point output. If these results are not specifically needed, don't output them to save computation time.

- *Output Spatial Field Slices Sparingly*

  Output only the spatial field slices that you need. Saving slices that you don't need results in a slower simulation.

- *Output spatial fields on a coarser grid than the simulation grid if possible.*

  When saving spatial field slices, the simulation grid may provide more resolution than necessary. Depending on the intended use of the data, it may be possible to use a coarser grid which will increase the simulation speed.

A.D. Using the Command Line

FullWAVE, as well as all other RSoft products, are designed to be used graphically. Most of the time, all work can be done inside the graphical interface making the tools easy to learn and use. Nevertheless, in any serious numerical work, there comes a time when there is no substitute for access to a command line or "DOS prompt". For example, we can usually gain a quick initial view at a data file can be found by typing a simple WinPLOT command from the command line. Typing a single command can be faster than using the mouse-based interface for WinPLOT. Other uses for the command line may include the utilities described in the CAD manual, or examining the actual values of a data file in a simple text editor as opposed to a full spreadsheet program which may be slower or inconvenient, or absent. In addition, as described in the CAD manual, FullWAVE can be run completely from the command prompt.

On Linux systems, the use of the command line is second nature, but it can be less familiar under Windows-based systems. Unfortunately, the standard command prompt provided with Windows operating systems is poor in both commands and editing features. It is sufficient to carry out all the commands described in this manual, but after working on your own projects for awhile, you may find its shortcomings irritating.

Another solution to this problem is the "Cygwin" package. This is a replacement command line for Windows containing a full set of UNIX utilities and more powerful editing facilities. Cygwin is free to download and use, and is relatively easy to install, though we would only recommend its use for advanced computer users. It is available at www.cygwin.com. As always, software you obtain from sources other than RSoft are installed and used at your own risk.

Appendix B

FullWAVE File Formats

This appendix documents the file formats used in FullWAVE that are specific to this package. Except where noted, all files are in ASCII text format. For further information, contact RSoft.

B.A. Standard RSoft File Format

This is the main file format which is used to represent functions of one, two, or three variables, including field profiles, index profiles, and other output types. This format is described in detail in the Appendices of the CAD manual.

B.B. Monitor File Formats

If a design file contains monitors, an **Output Prefix** has been specified, and the appropriate output type has been enabled, FullWAVE will generate one or more data files and graphs. These files can be split into two groups: single-value monitor output and spatial output.

B.B.1. Single-Value Output File

The results of single-value monitor output are displayed during a simulation in the simulation window below the field display. If an **Output Prefix** has been set, the data will also be saved in a file `<prefix>.tmn` with corresponding WinPLOT command file `<prefix>.ptm`. The `.tmn` file is organized into several columns. The first column contains the time in units of cT in FullWAVE's standard unit of µm. Each subsequent column contains a monitor value corresponding to a monitor in the layout. There is one column for each single-value output type enabled.

Units

The units of single-value output are different for each output type and can be found in the descriptions of each type in Section 5.C. Note that generally the data shown in plots and the simulation window are the squared values of the data in the `.tmn` file. This is done to show the most interesting quantity while preserving as much information as possible about the output.

B.B.2. Spatial Output Files

The results of spatial monitor output will be saved to individual files during a FullWAVE simulation if an **Output Prefix** is set. Two files for each spatial output type selected will be produced: A data file, and a corresponding WinPLOT command file. The data files are in the standard field format discussed in the previous section. The coordinate system used within the output file is the time monitor's local coordinate system described in Section 5.B.1.

Output File Naming Convention

The output files are saved with names in the form `<prefix>_m#_t#_montype.dat` where "`_m#`" is the monitor number, "`_t#`" is the value of the simulation time at which the output was saved in units of cT, and "`montype`" indicates the type of spatial output contained in the file. The monitor types are

Value	Spatial Output Type
`ex`	**Ex**
`ey`	**Ey**
`ez`	**Ez**
`hz`	**Hx**
`hy`	**Hy**
`hz`	**Hz**
`pow`	**Power**
`eden`	**E Density**
`hden`	**H Density**
`poy`	**Poynting Vector**
`absorption`	**Absorption**

For example, the file `run1_m3_t10_pow.dat` contains spatial output of type **Power** from monitor #3 at t=10 cT from a simulation that was run with an **Output Prefix** of `run1`. The corresponding WinPLOT command file for this example data file will be named `run1_m3_t10_pow.pcs`.

Units

The units of single-value output are different for each output type and can be found in the descriptions of each type in Section 5.D. The data shown in plots is exactly the same as the data saved in the output files.

B.B.3. FFT Output Files

FFT output is enabled by selecting **Wavelength Output**, **Frequency Output**, or both in the FDTD Output Options dialog box in addition to appropriate monitor settings. Section 3.E.2 and Section 5.E contain more information on these options.

Frequency Analysis

FFT results can be saved as a function of frequency via the appropriate option in the FDTD Output Options dialog box. The frequency results consist of two files: a data file `<prefix>.fmn`, and a WinPLOT command file `<prefix>.pfm`. Note that if the **Padding Factor** option is used, the data files will have extensions .fm2, .fm4, etc, and if the **Preliminary Output Factor** option is used, the data files will have extensions .fmn_1, .fmn_2, etc. In both of these special cases, only one plot file is produced which corresponds to the final results. Additionally, an additional file <prefix>.smn is produced which contains the source spectrum.

In all these data files, the first column contains the frequency in inverse microns. Each subsequent column contains the FFT results for each output selected. The units of the data contained in these files and displayed is discussed in Section 5.E.2.

Wavelength Analysis

FFT results can be saved as a function of wavelength via the appropriate option in the FDTD Output Options dialog box. The wavelength results consist of two files: a data file <prefix>.wmn, and a WinPLOT command file <prefix>.pwm. Note that if the **Padding Factor** option is used, the data files will have extensions .wm2, .wm4, etc, and if the **Preliminary Output Factor** option is used, the data files will have extensions .wmn_1, .wmn_2, etc. In both of these special cases, only one plot file is produced which corresponds to the final results.

In all these data files, the first column contains the wavelength in microns. Each subsequent column contains the FFT results for each output selected. The units of the data contained in these files and displayed is discussed in Section 5.E.2.

B.B.4. DFT Output Files

DFT output is enabled by selecting the appropriate monitor settings discussed in Section 5.F. The DFT results for single-value output types will be saved in data files <prefix>_m#.dmn for each monitor, where _m# is the monitor number, with a single accompanying plot file <prefix>.pdm. The DFT results for selected spatial output types will be saved in data files <prefix>_m#_f#_type.dat for each monitor and frequency value, where _m# is the monitor number, _f# is the frequency at which the DFT was taken, and 'type' is the monitor type. For a list of the supported spatial output types, please see the previous section.

The simulation time at which the DFT is taken is included in the file name if the **Spatial Time Step** is less than the **Stop Time**.

For example, the file run1_m3_f064516_pow.dat contains spatial output of type **Power** from monitor #3 at a frequency of 0.64516 μm^{-1} from a simulation with an **Output Prefix** of run1. The corresponding WinPLOT command file for this example data file will be named run1_m3_f064516_pow.pcs.

The units of the data contained in these files and displayed is discussed in Section 5.F.2.

Appendix C

Symbol Table Variables

In this appendix we document the symbol table variable names which control the internal operation of FullWAVE. Most of these variables correspond to dialog fields which have already been documented. Note that some of these options are shared with RSoft's BPM based simulation engine BeamPROP, but are repeated here for completeness.

The following table lists the variable names corresponding to these dialog fields:

C.A. FDTD Simulation Parameters

The following variables correspond to the fields found in the FDTD Simulation Parameters dialog box:

Field Name	Variable Name
X Domain Min	`boundary_min`
X Domain Max	`boundary_max`
X Compute Step	`grid_size`
X PML Width	`fdtd_pml_width_x`
Y Domain Min	`boundary_min_y`
Y Domain Max	`boundary_max_y`
Y Compute Step	`grid_size_y`
Y PML Width	`fdtd_pml_width_y`
Y PML Reflectivity	`fdtd_pml_refl_y`
Z Domain Min	`domain_min`
Z Domain Max	`domain_max`
Z Compute Step	`step_size`
Z PML Width	`fdtd_pml_width_z`
Z PML Reflectivity	`fdtd_pml_refl_z`
Time Step	`fdtd_time_step`

Stop Time	`fdtd_stop_time`
Slice Time	`fdtd_slice_time`
Update Time	`fdtd_update_time`
Monitor Time	`fdts_monitor_time`
Excitation Type	`fdtd_excitation_type`
Ramp/Pulse Time	`fdtd_ramp_time`
Output File Prefix	`prefix`

C.B. Advanced Grid Parameters

The following variables correspond to the fields found in the Advanced Grid Parameters dialog box that have not documented in the last section:

Field Name	**Variable Name**
X Grid Size (Bulk)	`grid_size`
X Grid Size (Edge)	`grid_edge_x`
X Griding Ratio	`grid_ratio_x`
X Minimum Divisions	`grid_mindiv_x`
X Grid Grading	`grid_bulk_nonuniform_x`
X Interface Alignment	`grid_align_x`
Y Grid Size (Bulk)	`grid_size_y`
Y Grid Size (Edge)	`grid_edge_y`
Y Griding Ratio	`grid_ratio_y`
Y Minimum Divisions	`grid_mindiv_y`
Y Grid Grading	`grid_bulk_nonuniform_y`
Y Interface Alignment	`grid_align_y`
Z Grid Size (Bulk)	`step_size`
Z Grid Size (Edge)	`grid_edge_z`
Z Griding Ratio	`grid_ratio_z`
Z Minimum Divisions	`grid_mindiv_z`
Z Grid Grading	`grid_bulk_nonuniform_z`
Z Interface Alignment	`grid_align_z`

C.C. Launch Parameters

The following variables correspond to the fields found in the Launch Parameters dialog. Note that this is only valid for the specification of one launch field. Multiple launch fields are specified directly in the `*.ind` file.

Field Name	**Variable**
Power	`launch_power`
Phase	`launch_phase`

Launch Type	launch_type
Tilt	launch_tilt
Mode	launch_mode
Mode Radial	launch_mode_radial
Random Set	random_set
Input File (E-Major)	launch_file
Input File (E-Minor)	launch_file_minor
Align File	launch_align_file
Normalization	launch_normalization
Pathway	launch_pathway
Background N	launch_background_index
Delta N	launch_delta
Phi	launch_angle
Width	launch_width
Height	launch_height
Length	launch_length
Position X	launch_position
Position Y	launch_position_y
Position Z	launch_position_z
Neff	launch_neff
Temporal Excitation Type	fdtd_excitation_type
Wavelength	fdtd_wavelength
Ramp/Pulse Time	fdtd_ramp_time
Delay Time	fdtd_delay_time
Shutoff Time	fdtd_shutoff_time
Chirp Coefficient	fdtd_chirp_coef
Spatial Excitation Type	fdtd_excitation_spatial
Current Direction	fdtd_currentdirection_auto
Direction Vector (X)	fdtd_currentdirection_x
Direction Vector (y)	fdtd_currentdirection_y
Direction Vector (Z)	fdtd_currentdirection_z

C.D. Display/Output Options

The following variables correspond to the fields found in the Display and Output Options dialog boxes:

Field Name	Variable Name
Slice Output	fdtd_slice_output
Field Output	fdtd_field_output
BPM Field Output	bpm_field_output
BPM Field Offset	bpm_field_offset

Far Field Output	`farfield_output`
Time Monitor	`fdtd_tmonitor_output`
Frequency Monitor	`fdtd_fmonitor_output`
Wavelength Monitor	`fdtd_wmonitor_output`
Padding Factor (2^(N-1))	`fdtd_fft_maxpow2f`
Preliminary Output Factor (2^-N)	`fdtd_fft_minpow2f`
Field Component	`fdtd_view_component`
Cross Section	`fdtd_cut_direction`
Slice Position X	`slice_position_x`
Slice Position Y	`slice_position_y`
Slice Position Z	`slice_position_z`
Outline Color	`color_outline`
Color Shades	`color_scale`
Color Scale	`color_shades`
Show Color Scale	`color_scale_display`
Plot Aspect Ratio	`plot_aspectratio`
Display Whole Domain on Master Node	`fdtd_mpi_wholedisplay`
Display Grid Factor	`fdtd_mpi_wholedisplay_grid`
Display Height	`fdtd)mpi_wholedisplay_height`

C.E. Advanced Parameters

The following variables correspond to the FDTD Advanced Parameters dialog:

Field Name	**Variable Name**
X Boundary	`fdtd_bc_x`
Y Boundary	`fdtd_bc_y`
Z Boundary	`fdtd_bc_z`
Boundary type at minimum X	`fdtd_bc_left`
Boundary type at maximum X	`fdtd_bc_right`
Boundary type at minimum Y	`fdtd_bc_bottom`
Boundary type at maximum Y	`fdtd_bc_top`
Boundary type at minimum Z	`fdtd_bc_front`
Boundary type at maximum Z	`fdtd_bc_back`
X ka/pi Value	`fdtd_bc_kax`
Y ka/pi Value	`fdtd_bc_kay`
Z ka/pi Value	`fdtd_bc_kaz`
Automatic ka/pi values	`fdtd_bc_ka_auto`
Index value for ka/pi	`fdtd_bc_ka_n0`
X PML Refl.	`fdtd_pml_refl_x`
Y PML Refl.	`fdtd_pml_refl_y`

Z PML Refl.	fdtd_pml_refl_z
X PML Kappa	fdtd_pml_kappa_x
Y PML Kappa	fdtd_pml_kappa_y
Z PML Kappa	fdtd_pml_kappa_z
Repeat Source	fdtd_source_repeat
Use Single Precision	fdtd_float
Compute Mode	fdtd_compute_mode
Continue Simulation	fdtd_continue

C.F. Advanced Features

The following variables correspond to the advanced features discussed mainly in Chapter 6, but also throughout other parts of the manual.

Variable Name	Description
radial	Radial FDTD
azimuthal_mode	Sets the azimuthal mode number for a Radial FDTD calculation
output_as_3D	Produce 3D results from 2D radial simulation
fdtd_dispersion	Enable Dispersion/Non-Linearity
background_n2_index	Background Real N2 parameter
background_n2_alpha	Background Imag N2 parameter
fdtd_dispersion_epsinf	Value of epsinf for dispersion model.
fdtd_continue	Enable the continuation of a completed simulation
fdtd_state_output	Enable output of simulation state
fdtd_excitation_file	Set initial state in order to restart simulation
fdtd_append_tmonitor	Append time monitors when restarting a completed simulation
fdtd_nx_master	Set to 0 to use master node during a cluster simulation for collecting and analyzing data, and not for simulation.
fdtd_magnetic	Enables the use of magnetic materials.
fdtd_density_normalization	Toggles the normalization of E and H density monitors to the input power.
fdtd_monitor_update_time	Sets the update time for the monitor display in the simulation window.
fdtd_launch_pml	Sets the behavior of the launch field in the PML.
fdtd_launch_pml_factor	Sets the gain/loss factor used when `fdtd_launch_pml` is set to `2`.
launch_phase_ref	Controls the phase reference for the rectangular phase launch type.
fdtd_launch_direction	Sets the launch plane to be used for the primary launch field.
fdtd_launch_direction_#	Same as `fdtd_launch_direction`, except for a specific launch field number (#).
fdtd_bc_ka_auto	Automatically sets the phase differences across periodic

	boundary conditions for a tilted plane wave launch.
`launch_power_unit`	Sets the unit for the launch power.
`fdtd_pml_index`	Controls the scheme used to calculate the index in the PML region.
`fdtd_complex`	Enables the use of a complex FDTD algorithm.
`launch_random_spatial_phase`	Enables a random phase for a current profile source.
`Launch_random_spatial_dir`	Enables a random direction for a current profile source.
`launch_random_spatial_corlen`	Controls the randomness for a current profile source.
`launch_random_spatial_seed`	Controls the randomness for a current profile source.
`fdtd_launch_enclosed`	Enables the use of an enclosed plane wave launch.
`fdtd_farfield_output`	Enables the output of the far-field in all directions.
`fdtd_farfield_offset`	Offset for far-field calculation.
`farfield_periodic`	Enables the computation of the far-field for a periodic structure.

Appendix D

Units & Nonlinearity

In this appendix we explore the relation between real world SI units, and the units used within FullWAVE calculations. This is particularly important when doing nonlinear calculations.

Nonlinear Response and Units

This general section does not use FullWAVE-specific units.

We begin by deriving some expressions for optical power, intensity and nonlinear coefficients. We assume harmonic time dependence and write the electric and magnetic fields as

$$\mathcal{E}(\mathbf{x},t)=\frac{1}{2}\left(\mathbf{E}_0\mathrm{e}^{-i\omega t}+\mathbf{E}_0{}^{*}\mathrm{e}^{i\omega t}\right),$$
$$\mathcal{H}(\mathbf{x},t)=\frac{1}{2}\left(\mathbf{H}_0\mathrm{e}^{-i\omega t}+\mathbf{H}_0{}^{*}\mathrm{e}^{i\omega t}\right). \tag{1}$$

We take a mode polarized along the *y* axis and propagating along *z*:

$$\mathbf{E}_0=f\left(x,y\right)\mathrm{e}^{ikz}\hat{\mathbf{y}}$$

where $\mathbf{k}$=$nL\omega/c$, with n_L the effective index of the mode. From Faraday's law

$$\nabla\times\mathbf{E}=-\mu_0\frac{\partial}{\partial t}\mu\mathbf{H},$$

we find

$$\mathbf{H}_0=\left(\frac{-n_L}{Z_0}f\left(x,y\right)\hat{\mathbf{x}}+\frac{1}{i\omega\mu_0}\left(\frac{\partial f}{\partial x}-\frac{\partial f}{\partial y}\right)\hat{\mathbf{z}}\right)\mathrm{e}^{ikz}$$

where the impedance of free space

$$Z_0 = \sqrt{\mu_0 / \varepsilon_0} \approx 376.7\ \Omega$$

For the special case of a plane wave with *f*(*x*) = 1,

$$\mathbf{H}_0 = \frac{-n_L}{Z_0} \mathrm{e}^{ikz} \hat{\mathbf{x}}$$

The intensity *I*, measured in Wm^{-2}, is the time-averaged value of the *z*-component of the Poynting vector **S**:

$$\begin{aligned} I &= \langle S_z \rangle = \left\langle (\boldsymbol{\mathcal{E}} \times \boldsymbol{\mathcal{H}})_z \right\rangle \\ &= \frac{1}{4} \left\langle \left(\mathbf{E}_0 \times \mathbf{H}_0^{\,*} + \mathbf{E}_0^{\,*} \times \mathbf{H}_0 \right)_z \right\rangle \\ &= \frac{n_L}{2Z_0} |\mathbf{E}_0|^2 \hat{\mathbf{z}} \\ &= \frac{n_L}{2Z_0} f(x,y)^2 \hat{\mathbf{z}}, \end{aligned} \tag{2}$$

which is equivalent to a form commonly found in textbooks:

$$I = \varepsilon_0 n_L{}^2 v_p |E_0|^2 / 2$$

where v_p is the phase velocity c/n_L. Finally, the optical power is the integral of the intensity over the transverse plane of the waveguide:

$$\begin{aligned} P &= \iint \mathrm{d}x\mathrm{d}y\, I(x,y) \\ &= \iint \mathrm{d}x\mathrm{d}y \frac{n_L}{2Z_0} |\mathbf{E}_0|^2 \\ &= \iint \mathrm{d}x\mathrm{d}y \frac{n_L}{2Z_0} f(x,y)^2. \end{aligned}$$

Calculations in 2D

The above analysis applies for 3D calculations where the incoming mode has a 2D profile *f*(*x,y*). For 2D calculations, where the mode profile is a function of one variable *g*(*x*), the analysis is identical except that the optical power is expressed as power per unit vertical length and measured in W/m:

$$\begin{aligned} \hat{P} &= P/(\text{unit height}) \\ &= \int \mathrm{d}x \frac{n_L}{2Z_0} |\mathbf{E}_0|^2 \\ &= \int \mathrm{d}x \frac{n_L}{2Z_0} g(x)^2. \end{aligned}$$

The Nonlinear Polarization

Neglecting dispersion and assuming linearly polarized fields, the dielectric polarization including second and third order nonlinearities is given by

$$\mathcal{P}_x = \varepsilon_0 \left[\left(1+\chi^{(1)}\right) \boldsymbol{\mathcal{E}}_x + \chi^{(2)} \boldsymbol{\mathcal{E}}_x \boldsymbol{\mathcal{E}}_x + \chi^{(3)} \boldsymbol{\mathcal{E}}_x \boldsymbol{\mathcal{E}}_x \boldsymbol{\mathcal{E}}_x \right]$$

Here

$$\varepsilon = n_L{}^2 = \left(1+\chi^{(1)}\right)$$

is the dielectric constant and $\chi^{(2)}$ and $\chi^{(3)}$ are the quadratic and cubic susceptibilities, with units of m/V and m^2/V^2 respectively. Technically, the susceptibilities are tensor quantities, but we assume isotropic materials here. The issues of field normalization do not change significantly when dealing with anisotropic materials. Considering only the cubic nonlinearity, the nonlinear refractive index or Kerr effect is associated with the part of the polarization that oscillates at the driving frequency ω. From Eq. (1), this is found to be

$$P_{NL} = \frac{3}{8} \varepsilon_0 \chi^{(3)} \left|\mathrm{E}_0\right|^2 \mathrm{E}_0 \mathrm{e}^{-i\omega t} \hat{\mathbf{x}}$$

The total polarization is thus

$$\begin{aligned} P_{\text{all}} &= \frac{\varepsilon_0}{2} \left[n_L{}^2 + \frac{3\chi^{(3)}}{4} \left|E_0\right|^2 \right] E_0 \mathrm{e}^{-i\omega t} \hat{\mathbf{x}}, \\ &= \frac{\varepsilon_0}{2} \left[n_L{}^2 + n^{(2)} \left|E_0\right|^2 \right] E_0 \mathrm{e}^{-i\omega t} \hat{\mathbf{x}}, \end{aligned}$$

where the nonlinear coefficient

$$n^{(2)} = \frac{3}{4} \chi^{(3)}$$

It is very important that the nonlinear coefficient $n^{(2)}$ is not confused with the nonlinear refractive index n_2, which is defined in terms of the intensity as

$$\Delta n_{NL} = n_2 I \tag{3}$$

This value of n_2 is determined by writing

$$\begin{aligned} \varepsilon_{\text{tot}}\left(E_0\right) &= \left(n_L + \Delta n_{NL}\right)^2 \\ &\simeq n_L{}^2 + 2 n_L \Delta n_{NL}, \end{aligned}$$

so that we find

$$\Delta n_{NL} = \frac{3\chi^{(3)}}{8 n_L} \left|E_0\right|^2$$

and finally

$$n_2 = \frac{3\chi^{(3)}}{4{n_L}^2} Z_0 \tag{4}$$

where it is easily checked that n_2 has the units m^2/W. Experimental reports almost always quote values for either $\chi^{(3)}$ or n_2, but rarely $n^{(2)}$. The latter may be encountered in theoretical works, and in particular $n^{(2)}$ is the quantity used in the popular texts by G. P. Agrawal (as *n2*), so it is critical to check the definitions carefully and make any necessary conversions to $\chi^{(3)}$ before attempting simulation.

Dimensions of Electromagnetic Quantities in FullWAVE

In SI units, the Maxwell curl equations (in the absence of free currents) take the form

$$\nabla \times \mathbf{E} = -\mu_0 \frac{\partial}{\partial t} \mu \mathbf{H},$$
$$\nabla \times \mathbf{H} = \varepsilon_0 \frac{\partial}{\partial t} \varepsilon \mathbf{E}.$$

The free space permittivity ε_0=8.854 x 10^{-12} [Fm^{-1}] and permeability μ_0=4π x 10^{-7} [NA^{-2}] are both very small quantities in SI, which is inconvenient for numerical simulation. Hence FullWAVE works in a set of normalized units in which both quantities are set to 1 and the unit of time is the same as length (cT). For linear calculations, we normally work without worrying about the units of the field since all results can be scaled to match the power in experiments. However, in nonlinear work we need to know how to convert back to standard SI quantities. This is conceptually straight-forward but must be done carefully to avoid scaling errors.

First, we note that the unit of length in FullWAVE is 1µm. FullWAVE's electric field quantity **E** can then be naturally interpreted in units of V/µm. However in the normalized units, the magnetic field **H** in FullWAVE is also expressed in V/µm rather than the standard units of A/µm^3. This means that any derived quantity that depends on **H** must be scaled appropriately to recover values in SI. The simplest way to accomplish this is to divide each appearance of **H** by the impedance of free space Z_0.

For example, for the magnetic field, optical power, and magnetic energy density, we have

$$\mathbf{H}_{\mathrm{SI}} = \frac{1}{Z_0} \mathbf{H}_{\mathrm{FW}},$$
$$P_{\mathrm{SI}} = \frac{1}{Z_0} P_{\mathrm{FW}},$$
$$U^m{}_{\mathrm{SI}} = \frac{1}{{Z_0}^2} U^m{}_{\mathrm{FW}}.$$

In particular, the optical power monitor in FullWAVE is interpreted in units of W·Ω^2 or simply V^2.

Field Normalization Options

The absolute intensity of a launch field in FullWAVE is influenced by the following fields and options in the launch dialog

- The **Launch power** [launch_power] field at the top of the dialog.
- The choice of normalization of any mode files if the launched field is from a file.
- The **Launch Normalization** option [launch_normalization] at the bottom left of the dialog.
- The value of the symbol launch_power_unit if **Launch Normalization**=*Unit Power*.

The resultant field also depends on whether the launch source is of *Field* or *Current* type. The following analysis is for *Field* launch types exclusively.

The total propagating power is of course a combination of the peak intensity and the transverse mode profile. In the following, we use the symbol P_0 for the value of the **Launch power** dialog field and we write the real transverse mode profile as $f(x,y)$. This is an arbitrary function with no particular normalization specified. In addition, we define the scaled profile $f_s(x,y)$ by $f(x,y) = Af_s(x,y)$ where $1/A = \max[f(x,y)]$ so that $f_s(x,y)$ has the maximum value of unity. We also define a normalized mode area we define as

$$\alpha = \iint \mathrm{d}x\mathrm{d}y\left[f_s\left(x,y\right)\right]^2$$

When the spatial profile *f(x,y)* is one of the standard built-in types such as *Slab Mode* or *Gaussian*, then FullWAVE uses the convention A=1 and the value of α depends on the function. However, if the launched mode is from a file, these quantities may have arbitrary positive values. For instance, a common alternative would be to have a launch file normalized to unit area, so that $\alpha = 1$.
There are four schemes for field normalization in FullWAVE, as follows, remembering that P_0 is the value of the **Launch Power** field in the launch dialog.

- **Launch Normalization** = *Unit Peak*

  The peak electric field in the launched field is given by $E_0 = \text{sqrt}[P_0]$ regardless of the normalization of the mode profile.

- **Launch Normalization** = *None*

  The peak electric field in the launched field is given by $E_0 = A\text{·sqrt}[P_0]$

- **Launch Normalization** = *Unit Power* with launch_power_unit=0

  The maximum *instantaneous* power of the launch field (equal to the intensity integrated over the launch profile) is given by P_0 measured in units of V^2. The time-averaged power is given by $P_0/2$.

- **Launch Normalization** = *Unit Power* with launch_power_unit=1

  The maximum *instantaneous* power of the launch field (equal to the intensity integrated over the launch profile) is given by P_0 measured in units of W. The time-averaged power is given by $P_0/2$.

Note that the power flowing through a cross-sectional plane varies over an optical cycle, and for a mode, the average power is half the peak power. It is very important to remember this when using a time-averaged power monitor in FullWAVE. In time-averaged mode, the power monitor compensates for this factor of one-half by doubling the true time-averaged power. Thus if you use a time-averaged power monitor and need to know the physical average power, you must remultiply the monitor value by 0.5. Note that no such scaling is applied for a power monitor that is not in time-averaged mode. As a result, when **Launch Normalization** = *Unit Power*, it is convenient to specify a launch power of $2P_0$, so that P_0 represents the average power, which is usually the quantity of interest.

The following table shows the relations between all the important quantities for a launched mode

$$\mathbf{E}(\mathbf{x},t)=\frac{1}{2}\left(A\,f_s(x,y)\mathrm{e}^{i(kz-\omega t)}\hat{y}+c.c.\right)$$

in the four different unit schemes.

Launch normalization	**None**	**Unit peak**	**Unit power units=V²**	**Unit power units=W**
FW Launch power dialog value	P_0	P_0	$2P_0$	$2P_0$
FW Launch power dialog units	$V^2/\mu m$	$V^2/\mu m$	V^2	W
max[E_y] (V/ μm)	sqrt(P_0A)	sqrt(P_0)	sqrt($2P_0/n_{eff}\alpha$)	sqrt($2P_0Z_0/n_{eff}\alpha$)
max[H_x] (V/ μm)	n_{eff} sqrt(P_0A)	n_{eff} sqrt(P_0)	sqrt($2neffP_0/\alpha$)	sqrt($2neffP_0Z_0/\alpha$)
max[Intensity] (W/μm²)	$n_{eff}P_0A^2/2Z_0$	$n_{eff}P_0/2Z_0$	$P_0/\alpha Z_0$	P_0/α
Average Power (V²)	$n_{eff}P_0A^2\alpha/2$	$n_{eff}P_0\alpha/2$	P_0	P_0Z_0
Average Power (W)	$n_{eff}P_0A^2\alpha/2Z_0$	$n_{eff}P_0\alpha/2Z_0$	P_0/Z_0	P_0

The final two lines in the table above refer to the physical power launched into the simulation, and not the value that would be reported by a time-averaged power monitor (which would be twice as large).

Note that while we have a choice of unit schemes for the launched field, the monitors *always* use the same set of rationalized units: ie the measured **E** and **H** are in V/μm and the measured power is in V^2. The last section in this appendix discusses examples of each of the different field normalizations and demonstrates the consistency of the various field quantities.

Setting FullWAVE's Nonlinear Parameters

Understanding how to convert the nonlinear response coefficients $\chi^{(2)}$, $n^{(2)}$ or $\chi^{(3)}$ is the trickiest part of performing nonlinear calculations that can be directly compared to experiment.

We now proceed to calculating the values for the nonlinear parameters in FullWAVE's Material Editor. We assume that the material properties are known as the value of either $\chi^{(2)}$ for quadratic nonlinearities, or $n^{(2)}$ or $\chi^{(3)}$ for cubic nonlinearities. The first step is to find the physical susceptibility $\chi^{(2)}$ or $\chi^{(3)}$ in units of μm/V or μm²/V² respectively, using Eq. (4) if necessary. Using Eq. (4) requires a knowledge of the effective index n_L of the incoming waveguide mode, which can be found by modal solution with BeamPROP or FemSIM or simply by estimation. In this case, comparison with experiment might require a fitting parameter associated with factors such as the fraction of power coupled in to the waveguide or the mode profile. The relation between the physical susceptibility and the quantities entered as FullWAVE's **chi2** or **chi3** settings, denoted here as $\chi^{(3)}$ and $\chi^{(3)}$, and launch power setting, denoted P_{FW}, depends on the choice of field normalization.

There are several equivalent choices. The one chosen is a matter of personal preference and convenience depending on which physical quantities are best known for your system.

Method 1

Launch normalization=*Power*, `launch_power_unit=1` (Power measured in Watts)

With this normalization, we need to know the required input power P. We may know this directly. Commonly however, we instead know the required peak intensity or the required peak nonlinear phase shift. In such cases, we can find P by calculating or estimating the mode area α and using relations in the field normalization section above. There are then two choices:

- Fix the nonlinear coefficients and vary the launch power:

$$P_{FW} = 2P[\mathrm{W}]$$
$$\chi^{(3)}_{FW} = \chi^{(3)}\left[\mu\mathrm{m}^2/\mathrm{V}^2\right]$$
$$\chi^{(2)}_{FW} = \chi^{(2)}\left[\mu\mathrm{m}/\mathrm{V}\right]$$

- Fix the launch power and vary the nonlinear coefficients:

$$P_{FW} = 2$$
$$\chi^{(3)}_{FW} = P[\mathrm{W}]\,\chi^{(3)}\left[\mu\mathrm{m}^2/\mathrm{V}^2\right]$$
$$\chi^{(2)}_{FW} = \sqrt{P[\mathrm{W}]}\,\chi^{(2)}\left[\mu\mathrm{m}/\mathrm{V}\right]$$

The first approach is the most direct, but the second has the advantage that the field amplitudes remain of order unity and are better displayed by the graphics system. The factor of 2 in the launch power accounts for the fact that the FullWAVE quantity corresponds to the peak instantaneous power rather than the average power.

Method 2

Launch normalization=*Power*, `launch_power_unit=0` (Power measured in V^2)

This method is essentially the same as Method 1, except that a factor Z_0 must be included in the nonlinear coefficients to account for the scaled units of power. Using this normalization means that the fields are smaller by sqrt[Z_0] than in Method 1, and so perhaps better displayed by the graphics, but there is no advantage over the second scheme of Method 1, which is normally the preferred approach. The two scaling choices are

- Fix the nonlinear coefficients and vary the launch power:

$$P_{FW} = 2P[\mathrm{V}^2]$$
$$\chi^{(3)}_{FW} = Z_0\,\chi^{(3)}\left[\mu\mathrm{m}^2/\mathrm{V}^2\right]$$
$$\chi^{(2)}_{FW} = \sqrt{Z_0}\,\chi^{(2)}\left[\mu\mathrm{m}/\mathrm{V}\right]$$

- Fix the launch power and vary the nonlinear coefficients:

$$P_{FW} = 2$$
$$\chi_{FW}^{(3)} = Z_0\, P\left[\mathrm{V}^2\right] \chi^{(3)}\left[\mu\mathrm{m}^2/\mathrm{V}^2\right]$$
$$\chi_{FW}^{(2)} = \sqrt{Z_0\, P\left[\mathrm{V}^2\right]}\; \chi^{(2)}\left[\mu\mathrm{m/V}\right]$$

Method 3

Launch normalization=*Unit Peak*

This method is useful if the maximum desired electric field is known. This might be known directly or might be obtained from knowing the desired peak nonlinear index change, and thus peak intensity, giving the peak electric field by Eqs. (2) and (3).

The two scaling options are then:

- Fix the nonlinear coefficients and vary the launch power:

$$P_{FW} = \left(E_0[\mathrm{V}/\mu\mathrm{m}]\right)^2$$
$$\chi_{FW}^{(3)} = \chi^{(3)}\left[\mu\mathrm{m}^2/\mathrm{V}^2\right]$$
$$\chi_{FW}^{(2)} = \chi^{(2)}\left[\mu\mathrm{m/V}\right]$$

- Fix the launch power and vary the nonlinear coefficients:

$$P_{FW} = 1$$
$$\chi_{FW}^{(3)} = \left(E_0\left[\mathrm{V}/\mu\mathrm{m}\right]\right)^2 \chi^{(3)}\left[\mu\mathrm{m}^2/\mathrm{V}^2\right]$$
$$\chi_{FW}^{(2)} = E_0\left[\mathrm{V}/\mu\mathrm{m}\right] \chi^{(2)}\left[\mu\mathrm{m/V}\right]$$

Note that the factor of 2 in the launch power is not required in this mode of operation, since the specified quantity, the maximum electric field is the peak instantaneous value rather than an averaged value.

One could also perform nonlinear calculations using **Launch normalization**=*None* in a similar fashion to Method 3, but we recommend against this since the dependence of the field strength on the amplitude of the mode profile introduces a likely source of scaling errors. Method 3 is a safer scheme and no less convenient.

Application Note 1: FullWAVE Field Normalization

This note quantitatively explores the field normalization issues discussed in above. You should thoroughly review the material in that section before proceeding with the following.

The commands to run the examples in this note are quite complicated and to save typing they have been expressed in Unix bash files. On Unix/Linux they can be run directly at the command line. Under Windows, a Unix interpreter is required. The free *Cygwin* environment provides a convenient, professional and comprehensive Unix command line under Windows. It is easy to install and may be downloaded from http://www.cygwin.com/. Independent of these examples, a powerful command line is indispensible in advanced simulation work, and we encourage users to consider obtaining Cygwin. However, RSoft can not provide support for installing or using Cygwin and is not responsible for any damage or data loss due to its use. Alternatively, Windows users can simply open the Unix scripts in a text editor and copy the individual lines to a Windows command terminal. This will work but is rather less convenient.

The example files for this note are contained in `<rsoft_dir>/examples/FullWAVE/ApplicationNotes/Normalization` and subdirectories.

Please note that these example files have the option `fdtd_p0_index = 0` so that the power monitor data is unscaled and absolute.

We explore the three different forms of launch field normalization: *Unit Peak*, *Unit Power* and *None*.

Normalization = *Unit Peak*

- Open and examine `norm_unit_peak/fwnorm_unit_peak.ind`.

The structure is a 2D slab waveguide of width 1 μm, background index 1, index contrast 0.5 and wavelength 1.55 μm. The fundamental mode of this waveguide has an effective index of 1.4029. The file is configured to launch the slab mode in CW mode.
A number of monitors are configured. In order they are monitors of E_y, H_x, H_z, power, time-averaged power and the spatial profile of E_y. All the monitors are located at the center of the slab waveguide. In the following, all of these settings remain the same. The only changes are to the normalization of the launch field and the resolution of the simulation grid.

- Open the launch dialog and observe that **Launch power**=`E0*E0`, and **Normalization**=*Unit Peak*.

Now run the calculations in `runner.sh`. Under Unix or Cygwin, this can be done simply by typing '`sh runner.sh`' at a command line. To use the MS-DOS command line, the commands must be copied and entered line by line.
The script runs four *FullWAVE* simulations with different grid resolution and settings for the launch field amplitude `E0`. After each simulation, a Python program `get_power.pyc` calculates a number of quantities from the monitor output created by *FullWAVE*. The calculated quantities have the following meanings:

- `E0`: Launch field amplitude in *Unit Peak* and *None* modes.
- `P0`: Launch field power in *Unit Power* modes.
- `Ey_max_theory`: The peak amplitude of the electric field according to the table above.
- `Ey_max_mon`: The peak amplitude of the electric field measured by the E_y monitor.
- `Hx_max_theory`: The peak amplitude of the magnetic field according to the table above.
- `Hx_max_mon`: The peak amplitude of the magnetic field measured by the H_x monitor.
- `Error_Hx`: The percentage error in the theoretical and measured values of H_x.

- `P[fw_mon]`: The time-averaged power in V^2/μm measured by the power monitor.
- `P[\int I(x,y)]`: The time-averaged power in V^2/μm according to the table above.
- `P[\int I(x,y)]`: The time-averaged power in W according to the table above.

The power values are quoted in V^2/μm and W/μm because for 2D calculations, the power is normalized per unit vertical length out of the screen.

Note that the theoretical values taken from the *None* column of the table above agree well with the measured values and that the agreement improves as the grid resolution increases. Secondly, when the launch amplitude `E0` increases by a factor of 2, the power is increased by a factor 4 as expected.

Normalization = *None*

- Change to the directory `norm_none` and open `fwnorm_none.ind`.

This file is identical to the previous one except that the normalization setting is changed to *None* and the launch profile is taken from a file rather than a pre-calculated slab mode.

- Run the script file `runner.sh`.

This script first solves for the mode profile using BeamPROP to obtain the profile `bpmode.m00`. By default this file is normalized to have a peak amplitude of unity. This file is then manipulated to create `bpmode_bytwo.m00` which has the same profile with a peak amplitude of 2.0. These two files are then used as launch fields into the FullWAVE calculation. Observe from the results, that if `bpmode_bytwo.m00` is launched, the peak electric field is twice that when `bpmode.m00` is launched, and the power correspondingly changes by a factor 4. Also, the effect of doubling the dialog launch amplitude `E0` is combined with the doubled amplitude of the launch file to increase the measured field by a factor of 4 and the power by a factor of 16.

If the normalization mode is changed to *Unit Peak*, then the two launch files would produce identical results, because the increased amplitude in `bpmode_bytwo.m00` is scaled away by FullWAVE before launching.

Normalization = *Unit Power*

- Change to the directory `norm_unit_power` and open `fwnorm_unit_power.ind`.

In this version of the problem, we use the default slab launch profile, but change the normalization mode to *Unit Power*, and set the **Launch power** to 2*`P0`. Recall that the factor 2 here means that `P0` then specifies the average rather than peak power. As discussed above, we also have to choose between power normalization in V^2 (`launch_power_unit`=0) or W (`launch_power_unit`=1).

- Run the script file `runner.sh`.

Two calculations are run, the first with the launch power express in W and the second with the launch power express in V^2. Observe that the calculated power display the correct values in the two cases and that they differ by a ratio of Z_0=376.7 Ω. Similarly, between the two cases the fields change by a factor of sqrt[Ω].

Summary

The above analysis is without doubt tedious. However, the calculations demonstrate how to check that the fields and powers are precisely as you expect. Getting these values correct is essential for nonlinear work, and the ideas above will help you to check quickly and efficiently that your configuration is correct.

Appendix E

Release Notes

This appendix summarizes the changes from previous versions of FullWAVE. Within each section, the first subsection discusses new or improved features, and the second subsection covers compatibility issues.

Changes from Version 6.2 to Version 6.3

Note that the change log which details the history of changes from Version 6.2 to Version 6.3 is available in the file `<rsoft_dir>/README/readmefw.txt`.

New Capabilities and Improvements to the Program

- New launch file specification to now accept launch files based on a file prefix and extension instead of providing individual files for each necessary field component.
- The same file specification used for launch fields is also used for custom data files for 'overlap' single-value monitor output.
- Added a default output prefix for FullWAVE simulations.
- Added a new feature to automatically choose an optimal time step which is the new default.
- Added bias control for the 'vm_weight' feature for power/energy/absorption monitors.
- Improved the 'Steady-state Field' and corresponding 'Far-Field' option when used with a pulsed excitation to output results at multiple frequencies.
- New option to output Overlap DFT monitors in 'field' units instead of 'power' units.
- New option to control the format of the source spectrum.
- Improved FFT's which are more efficient and handle non-power-of-2's to avoid aliasing effects.

- New ‘View Launch’ button in the Launch Properties dialog that displays the launch field for diagnostic purposes.
- New PEC material in the Material Editor library.
- Increased the maximum number of dispersion terms supported in the Material Editor from 6 to 24.
- New PEC boundary condition.
- Expanded the Polarizer controls to allow for any angle.
- New panning options in the CAD. See CAD manual for more details.

Significant Changes in Program Behavior

- Introduced new defaults for the Monitor Time and Update Time.
- Renamed the ‘Excitation Time Envelope’ to ‘Excitation Time Waveform’ which is more accurate. To output the envelope, set the variable `fdtd_stm_output_envelope` to `1`.
- The auto direction option for current launches now enforces Ey for TE and Hy for TM. The direction vector is now disabled by default, but advanced users can enable it by setting `fdtd_allow_curdir2d` to `1` or globally in the INI file by setting AllowCurrentDir2D to 1 in the [FullWAVE] section.
- Renamed the far-field “Intensity/Phase” option to “Amp/Phase” which is more accurate.
- Improvements to the normalization options for point sources. Normalization settings of None (the default) and Unit Peak are now identical and set the peak current to 1. Specifically, J*dx*dz (2D) and J*dx*dy*dz (3D) are set to 1 where J is the current density and dx, dy, and dz are the grid sizes. The old behavior can be recovered by setting `fdtd_current_norm_mode` to `2`. The Unit Power normalization has been modified to set the emanating power from the unbound source to unity. Backward compatibility can be has by setting `fdtd_current_norm_power` to `0`. Lastly, the source spectrum that is used to normalize the results in a pulsed case (when ‘Source Norm’ is set in the Output Options dialog) so that the additional frequency factor that results since the current spectrum is Gaussian, not he power spectrum is included so that the results are independent of frequency. To recover the old behavior, set `fdtd_current_norm_spec` to `0`. Note that these scalings are only for point sources, not extended sources. This will be addressed in the future.
- Single-plane far-field has been modified to produce a true far-field intensity profile. All field components are now combined into a single intensity plot. If all significant field on the output plane of the simulation, the results should match the multi-plane far-field output. In the periodic case, this option now produces diffraction efficiency vs. order to match DiffractMOD output.
- The ‘Monitor Format’ option in the Monitor Properties dialog has been renamed to ‘FFT/DFT Format’ to reflect that the option only affects the frequency analysis output. This option does not apply to memory intensive Power/Density FFT/DFT monitors which can be set by the variable `fdtd_powdens_spectrum_format`.

Changes from Version 6.1 to Version 6.2

Note that the change log which details the history of changes from Version 6.1 to Version 6.2 is available in the file `<rsoft_dir>/README/archive/readmefw61.txt`.

New Capabilities and Improvements to the Program

- Improved the calculation of E-Density/T-density and Absorption monitors spectrum. Previously, the dielectric constant at the center frequency was used in the calculation. Now, the spectrum calculation uses the full dispersive dielectric constant.
- Added new options to allow the output of far-fields for periodic structures.
- Added new option to allow for field component output for far-field (all directions).
- A new Gaussian offset feature which allows users to launch the beam off its waist.

Significant Changes in Program Behavior

- Changed the format of state files (`.fst`).
- Changed the nomenclature used to describe monitors and several related options. What used to be called 'Time Monitors' are now simply called 'Monitors', and the output options in the monitor properties dialog that used to be called 'Temporal Outputs' are now called 'Single-Value Outputs' and labeled as to whether they will be output vs. time or vs. frequency depending on the frequency analysis options chosen.
- User-specified launch width and height values now set the perpendicular width/height to the launch direction; the default values (when '`default`' keyword is used) already does this and so did not change. Contact RSoft with any problems.
- Any monitor names set in the Additional Options dialog are now used in monitor plots as well as output filenames. To disable this behavior, set the variable `monitor_labeling` to `0`.

Changes from Version 6.0 to Version 6.1

Note that the change log which details the history of changes from Version 6.0 to Version 6.1 is available in the file `<rsoft_dir>/README/archive/readme60.txt`.

New Capabilities and Improvements to the Program

CAD:

- Introduction of a new material editor where complex materials can be defined and shared between different design files and users. See CAD manual for more details.

Simulation:

- Added a new Current Profile excitation option useful for modeling incoherent sources, such as an LED.

- A major new feature for modeling gain has been added.
- A new enclosed launch option for launching a plane wave inside an enclosed area/volume. This feature is useful for scattering calculations.
- New launch behavior for radial FDTD calculations: when the azimuthal mode number is set to 0, a pure TE or TM mode is created corresponding to the selected polarization.

Analysis/Results:

- Improved the calculation of E energy density monitors to account for dispersion.
- A new far-field output has been added to compute the far-field in all directions from the simulation domain.
- The Poynting Vector spatial output now also include each component of the Poynting Vector as well as vector plot.
- Time monitors can now have multiple temporal outputs per monitor.
- New far-field option for periodic structures.
- New option to automatically compute the far-fields for DFT spatial monitors.
- New absorption monitor to compute the power absorbed within a volume.

Significant Changes in Program Behavior

- Overlap monitors have been changed to separate forward and backward components.
- The naming of MOST measurements has changed to better reflect the fact that multiple temporal outputs are possible from one time monitor.
- Azimuthal mode numbering for radial FDTD calculations has been made consistent across all of RSoft's passive device tools.
- Changed the default interface alignment for non-uniform grids.

Changes from Version 5.0 to Version 6.0

Note that the change log which details the history of changes from Version 5.0 to Version 6.0 is available in the file `<rsoft_dir>/README/archive/readme60.txt`.

New Capabilities and Improvements to the Program

CAD:

- Improved CAD interface. See CAD manual for more details.

Significant Changes in Program Behavior

- The Slice Time control has been moved to the Output Options dialog.

- The Output Options dialog has been expanded with new options
- New options have been added to the Advanced Parameters dialog.

Changes from Version 4.0 to Version 5.0

Note that the change log which details the history of changes from Version 4.0 to Version 5.0 is available in the file `<rsoft_dir>/README/archive/readme50.txt`.

New Capabilities and Improvements to the Program

CAD:

- Introduction of an entirely new index/grid/mesh generation algorithm which improves accuracy and allows more flexible layout.
- Added support for non-uniform grids. This can significantly improve accuracy while improving simulation speed.

Simulation:

- An option has been introduced to smoothly ramp down CW excitations.
- A new feature allows a hyperbolic secant (sech) time envelope for the launch pulse which can be useful for nonlinear soliton-forming simulations.
- Linear anisotropic simulations can now be combined with chi2 nonlinearity simulations. While this might cause small changes in the results for linear anisotropic simulations, the new results are more accurate.
- Added option (`fdtd_stm_output`) to output the time sequence of the launch source.
- Added advanced option to improve the performance of the Perfectly Matched Layer (PML) boundary in certain cases by absorbing evanescent waves inn the PML. This option should only be used in cases where the simulation is unstable in the PML region and increasing the PML width doesn't help.

Analysis/Results:

- Improved averaging of power monitors by enabling signed averaging.
- Added option (`monitor_update_time`) to control the redraw time for monitors. The use of this option can improve speed in rare cases.
- Added option to output 3D spatial field data for 3D time monitor spatial output. The output files are in a VTK format, and can be opened via the RSoft DataBROWSER.
- Introduced new utility Q-Finder which automates the calculation of cavity modes and Q-factors.
- Added new power monitor option (`fdtd_total_power_monitor`) to compute the total power flux through the a 3D power time monitor. This option is useful when using Q-Finder.

- Added a new total density (T Density) option which computes the sum of the E and H Density. This option is useful when using Q-Finder.
- Added ability to use arbitrary data files for overlap time monitors.
- Added option to control how time monitors are normalized. The variable `fdtd_p0_index` can be used to choose the launch field to use for the normalization.
- Added GUI for option to normalize frequency and wavelength output.
- The desired frequency for a cavity mode can now be set via the variable `fdtd_modefrequency_real`. The imaginary part of the mode frequency (for mode attenuation) can be set by the variable `fdtd_modefrequency_imag`.

Other:

- Improved the `output_as_3d` option for radial simulations to output all six field components (Ex, Ey, Ez, Hx, Hy, and Hz). Additionally, this option now generates 3D plots of all field output, including monitor output.

Significant Changes in Program Behavior

- The default implementation for clustering on Windows has been changed from MPICH to MPICH2 for improved stability/performance.
- The behavior of the Shutoff Time option has been changed to allow a more consistent meaning with other options. It now represents the total time from T=0 to the time the launch shuts off.
- The behavior of the Launch Normalization option for "Unit Power" has been changed. Previously, this option normalized the integral of E^2 to be normalized to unity, but now it properly causes the integral of ExH to be unity.
- The interpretion of n2 values entered in the additional segment properties dialog now include the 4/3 factor resulting from the first Born approximation making the dialog consistent with BeamPROP. This will change power scales by the same factor, and can be disabled for compatibility with the old behavior by setting `fdtd_n2_addbornfactor = 0`, although this is not recommended.

Changes from Version 3.0 to Version 4.0

Note that the change log which details the history of changes from Version 3.0 to Version 4.0 is available in the file `<rsoft_dir>/README/archive/readme40.txt`.

New Capabilities and Improvements to the Program

CAD:

- General improvements to the CAD program which are useful for FullWAVE. See the CAD manual for more information.

Simulation:

- A new command line utility `fwmpirun` for the simple execution of cluster calculations.
- A previously undocumented feature has been enabled which allows the linear anisotropic simulation to be combined with chi2 nonlinearity (isotropic or anisotropic).
- Added a new Material Editor which simplifies the definition of dispersive, non-linear, and magnetic materials.
- Added an option (`fdtd_bc_ka_auto`) to automatically set the periodic boundary condition ka/pi values appropriately when launching tilted plane waves. Note this option may not always be able to detect the correct parameters, and should be used with caution.
- Added an option in the simulation dialog to enable/disable cluster simulation on a per-file basis. In addition, for convenience, the global cluster setting for number of processes has been added to the main simulation dialog.

Analysis/Results:

- Added a feature for the cluster version (`fdtd_nx_master` = 0) which enables the master node to be excluded from the calculation, and only used for collecting, analyzing, displaying, and storing monitor information.
- Extension of far field output option to include phase.
- A new implementation for overlap monitors which allows arbitrary rotation. The new implementation also provides better normalization, a more accurate vector calculation of the overlap, and a better correlation with other power calculations by combining overlap integrals with both the E and H fields. In some cases, the overlap may yield noticeable differences from prior versions, but in general the new method should give more meaningful results. If it is desired to match prior results, the combined E/H overlap may be disabled by setting `fdtd_overlap_eh`=0 in the `*.ind` file, or setting `OverlapEH` = `0` in the `[FULLWAVE]` section of `BCADW32.INI` to disable it by default, although neither is recommended.
- New options to assist in calculating Q for resonant cavities, as well as other problems.

Other:

- Improved ability to shut down cluster simulations.

Significant Changes in Program Behavior

- Improved handling of dispersive, non-linear, and magnetic materials.
- The default field components when launching in the y direction (e.g. phi around 0 and theta around +/- 90) have been changed to better match user expectations. If you have previously used this feature, please contact RSoft with questions.
- Improved accuracy for launching 3D fields into structures. The new option may be disabled by setting `fdtd_etoh_vector` = `0`, although this is not recommended.

- Implementation of an important memory and time optimization option which uses "float" instead of "double" arithmetic. This option requires approximately half the memory, and is also significantly faster in many cases. Note that the use of this option may cause results to change slightly (e.g. in the sixth digit), but not significantly. This option can be switched off in the Advanced Parameters dialog if needed.
- Altered definition of E and H density monitors to be "normalized" by input power which is advantageous when using symmetric boundary conditions. The behavior can be turned off, and the old behavior recovered, by setting `fdtd_density_normalization` = `0`.
- Altered behavior of symmetry boundary conditions in 3D/TE case to yield the expected behavior. The adjustment can be disabled, and the old behavior recovered, by setting `fdtd_flipsym` = `0`.
- Added new options to improve the launching of plane waves directed along a primary axis.
- The launch field is no longer truncated in the PML by default. This option is controlled by `fdtd_launch_pml`. A value of 0 truncates the field and does not launch in the pml (the old behavior). Setting to `1` launches the field unchanged in the PML (the new default behavior).
- Changed the definition of the E and H Density monitors to include a factor of 1/2 in order to make them more directly interpretable in terms of Poynting's theorem.

Changes from Version 2.0 to Version 3.0

Note that the change log which details the history of changes from Version 2.0 to Version 3.0 is available in the file `<rsoft_dir>/README/archive/readme30.txt`.

New Capabilities and Improvements to the Program

CAD:

- As noted in the BeamPROP readme, several updates to the array generation utilities have been made.
- Added new functions which generate random sequences and lines.

Simulation:

- Release of the clustering feature for FullWAVE, which allows FullWAVE simulations to employ multiple CPU's distributed over a network.
- Implemented the Drude dispersion model.
- Added support for anisotropic materials.
- Added support for periodic boundary conditions.
- Enabled dispersion in the PML.
- Enabled Dispersion for the 2D TM case (previously it was only enabled for 2D TE or full 3D).
- Extended the cavity mode solver to 3D.

- Extended the continue simulation option to 3D.
- Extended the BPM Field Output option to operate with the clustering feature above, as well as additional settings to save all field components.
- Added limited support for the saturation coefficient and the imaginary part of the nonlinear index.

Analysis/Results:

- Added major new analysis features for the time monitors which allow the output of all field components and other common electromagnetic quantities.
- Enabled output of data from a time monitor at arbitrary angles.
- Added DFT frequency analysis to the above time monitors.
- Enabled the output of field data from a clustered simulation to be consistent with the non-clustered case.
- Enabled the display of the whole field on the master node during a clustered simulation.

Other:

- Introduced a new GUI interface for the FullWAVE simulation program which is common to both Windows and UNIX. The new interface has a new status line which is enabled under View/Settings/Enable-Status-Line.
- New organization of tools, documentation, etc. for the products in the RSoft Photonics CAD Suite. Please refer to `README.TXT` for information.
- Introduced new installation process.

Significant Changes in Program Behavior

- Improved handling of interfaces in dispersion and 3D codes. This results in improved accuracy and stability.
- Automatic adjustment of the computational domain to match the grid has been made consistent with BeamPROP.
- Improvement to the PML algorithm in both 2D and 3D which improves both accuracy and stability.
- Improved the logic used when launching a field in a free space region. The field is now decomposed into a plane wave solution of Maxwell's equations resulting in a realizable field.
- The FFT frequency output is now displayed versus a frequency scale in units of f/c to be consistent with the DFT output.

Changes from Version 1.0 to Version 2.0

Note that the change log which details the history of changes from Version 1.0 to Version 2.0 is available in the file `<rsoft_dir>/README/archive/readmefw10.txt`.

New Capabilities and Improvements to the Program

CAD:

- General improvements to the CAD program are described in the release notes section of the BeamPROP manual.
- Added improvements to the XZ array layout utility, particularly useful for PBG applications in FullWAVE
- Introduced a new XYZ array layout utility, particularly useful for PBG applications in FullWAVE Simulation:
- Added major new capabilities for dispersion and nonlinearity in both 2D and 3D.
- Added a capability to deal with metals with negative real part of the dielectric constant.
- Added a Radial FDTD option for structures with azimuthal symmetry.
- Added a feature which can improve the frequency resolution when generating a spectrum.
- Added a capability to calculate cavity modes (2D only).
- Improved control over the phase in time waveforms used in the excitation of the field.
- Added a new option to control the pulse delay.
- Added a new Impulse excitation type (2D only).
- Added new options to restart the simulation from a file (2D only).
- General speed and accuracy improvements.

Analysis/Results:

- Added a new option for time averaging of monitors, useful for cw simulations, which is available in the time monitor properties dialog. The scan variable feature has been extended to show the final time monitor versus the scan parameter(s).
- Added a feature to save the current view after the simulation is completed. By switching field views under the view menu, different field components or cross-sections can be selected, and then saved via File/Save-View.
- Added additional options to allow export of FullWAVE results to BeamPROP, and calculation of farfield patterns from the exported results.
- Added options to extend the time window used for FFT's which can sometimes yield greater resolution.

- Added a fixed aspect ratio option to various plots. The option is available under the Display... button. The option is available under the Display... button. For stored plots generated with previous versions, or generated without having set this option, the aspect ratio can be changed in WinPlot under Options/Axes/Aspect-Ratio... or via the `/aspect` option.

Other:

- Updated and expanded the documentation, including a new chapter containing several tutorial examples.
- Included a new version of WinPLOT, as noted in the release notes section of the BeamPROP manual.

Significant Changes in Program Behavior

- The log file has been renamed to `log.txt` for compatibility with other RSoft programs.
- The location of the INI file holding the window size and other settings has been changed to the users home directory under Windows NT/2000/XP. This avoids permission problems that sometimes occurred with this file and also allows different users to have different preferences.

Index

#

# Processes, 85

<

<<, 33, 43

>

>>, 33, 43

1

11, 147

2

2D, 109, 125, 131
2D XZ, 147

3

3D, 125
3D Output Format, 21
3D Structure Type, 125

A

a, 165
Absorption, 202
Accept Symbol, 109
Advanced..., 18, 43, 75, 88, 150
Ag, 162
Align File, 30
All Components, 23
Amp/Phase, 187
Amplitude, 66, 67
Amplitude/Phase, 72
Angle, 41
Anti-Symmetric, 75, 78, 151
Aspect Ratio, 147
Auto, 29, 30, 44, 55, 57, 77, 135
Auto Step, 19
Azimuth, 42
Azimuthal Mode #, 82

B

b, 165
Back Radius, 132
Background Index, 109, 125, 131
Background N, 33
Black, 136
boldface, 2

C

c, 165
CAD Display, 51, 153
Cancel, 7
Center, 100, 170
Chi2, 211
Chi3, 165, 211
Chirp Coefficient, 37, 38, 40
Circle, 147
Circular Left, 41, 42
Circular Right, 41, 42
Cluster Options, 104
Cluster Settings, 106
color scale, 24, 109
Color Shades, 24
Component, 109, 132, 135
Component Height, 51, 125
Component Name, 52
Component Width, 51, 109, 125, 131, 132
Compute Mode, 96, 150, 153
Computed Mode, 30
Continue Simulation, 80, 81, 92, 96, 97
ContourMap(XY), 134
ContourMap(XZ), 147
Controller, 42
Cover N, 33
Cross Section, 24
Current, 29
Current Direction, 29
Custom, 30, 41, 42, 55
Custom..., 30, 55
CW, 37, 67, 109, 139

D

Data File, 55
Decay Length, 101
Default, 23, 66, 67, 109, 147
Default Field, 53, 59, 67, 96
Delay Time, 38, 40
Delete, 43
delta, 165
Delta N, 33
DFT, 65, 67
Dimensions, 147
Direction Vector, 29
Dispersion/Nonlinearity, 18, 19, 71, 157, 158, 160, 163, 195
display, 162, 165
Display Color, 52, 132
Display Grid Factor, 88
Display Height, 88
Display Material Profile, 67, 125, 134, 147, 196
Display Mode, 125, 134, 147
Display Type, 125
Display Whole Domain on Master Node, 88
Display..., 24, 109, 147, 191
Domain Cut Direction, 88
Domain Max, 18, 147, 165, 176
Domain Min, 18, 33, 67, 125, 147, 151, 165, 176
Duplicate Segment, 132
Duplicate Selection, 132
DX, 52

E

E Density, 55, 61, 202
Edit Global Settings, 139
Edit Launch Field, 28, 109, 135
Edit Materials..., 162
Edit Pathways, 135
Edit Plot, 188
Edit Symbols, 109, 132, 147
Edit Table..., 58
Ellipticity, 42
Enable, 85
Enable clustering, 104
Enable for Power/Density monitors, 64
Enable Nonuniform, 176, 178

Enable Single-Value Output, 21
Enclosed Launch, 29, 45
Ending Frequency, 65
Ending Vertex Z, 132
Environment, 85
Eps Linear, 162
Epsilon at Infinity, 165, 181
Ex, 53, 59, 187, 202
Excitation, 62, 65, 67, 96, 109, 136, 139, 150, 165
Excitation Time Waveform, 23
Excitation Type, 146
Extension, 30, 55
Ey, 53, 59, 187, 202
Ez, 202

F

far field, 23, 72
Far Field for DFT Monitors, 23, 72
FDTD Continue, 151, 170
FDTD Options, 109
FFT, 62, 135
FFT Resolution, 100
FFT/DFT Format, 63, 64, 65, 66, 67
FHA if possible, 170
fiber, 125
Fiber Mode, 30, 33, 82
Field, 29
Field Component, 24
File, 29, 30, 33
File Spec, 55
Files of type, 7
Final Field, 23
Find Mode Volume, 103, 105, 170
Fixed at Z-Min, 125
Frac. tolerance, 100
Free Space Wavelength, 19, 39, 40, 57, 65, 66, 67, 96, 109, 125, 131, 139, 150, 151, 191
Frequency Analysis, 65, 67, 135
Frequency Monitor, 62
Frequency Output, 21, 63, 66, 136, 202
Frequency Spacing, 65
Front Radius, 132
FullWAVE, 109, 125, 131
FullWAVE Q-Finder, 171
FullWAVE Window, 104

G

Gain Term, 181
gaussian, 30
Gaussian Offset, 30
Global Monitor Settings, 52
Grid Grading, 176
Grid Options..., 176
Grid Ratio, 176
grid size, 18, 39, 109, 125, 134, 147, 178
Grid Size (Bulk), 176
Grid Size (Edge), 176
Grid Step, 147

H

H Density, 55, 61, 202
Height, 29, 33, 45, 51, 54, 55, 58, 59, 72
Hexagonal, 147
Hide, 104, 171
Host Machines, 85
Hx, 202
Hy, 53, 59, 202
Hz, 202

I

Ignore old .fst files, 103
Impulse, 39, 96, 150, 153
Index (imag part), 162
Index Difference, 109, 125, 131, 132, 162
Input File Spec, 30
Intermediate Field Slices, 23

K

ka/pi Value, 44, 77
Keep Window, 85

L

L, 147
Lattice Size, 147
Lattice Type, 147
Launch DataBROWSER, 125
Launch Field, 43
Launch Mode, 30
Launch MOST, 158, 160, 186
Launch Neff, 39
Launch Normalization, 165, 211
Launch Pathway, 135
Launch Polarizer, 41, 42
Launch Power, 211
Launch Type, 30, 82, 135
Launch..., 20, 28, 150
Launcher, 85
Length, 29, 33, 39, 45, 51, 54, 55, 58, 59, 72, 153
linear, 41, 42
Locally Defined, 191, 193
Lock for first N syms, 100

M

M, 147
Material Properties, 162, 191, 193
Materials..., 165
Max, 75
Max., 75, 100
Max. tolerance, 100
Measure Q by energy decay, 101, 172
Merge Priority, 132
Min, 75
Min., 75
Minimize, 104
Minimum Divisions, 176
Mode, 30, 55, 109
Mode Radial, 30
Mode slice command, 103
Mode slice densbox, 103
Model Dimension, 30, 41, 55, 109, 125, 131
Monitor, 48, 135
Monitor Time, 19, 21, 48, 53, 64, 65, 136
More..., 52, 58
MPI Implementation, 85
MultiMode, 30
Multi-Plane Far Field, 23, 74

N

Name, 109
Neff, 33
New, 43
New Circuit, 109, 125, 131
New Pathway, 135
New Symbol, 109
No, 30, 57, 80
No processes, 104
None, 30, 41, 42
Normal, 104
Normalization, 30, 43, 165, 211
Number of Processes, 85

O

Offset, 109, 125, 132, 135
Offset (from Z Max), 23
Offset Value, 109, 132

OK, 7, 104, 109, 125, 132, 134, 135, 136, 139, 142, 146, 147, 150, 158, 160, 163, 170, 171, 177, 178, 184, 185, 186, 188, 191, 193
Open DataBROWSER, 158, 160
Options..., 21
Outline Color, 24, 109, 136, 147
Output Prefix, 7, 21, 48, 59, 60, 61, 62, 72, 74, 80, 81, 96, 104, 109, 125, 136, 139, 142, 147, 150, 151, 153, 163, 167, 170, 171, 177, 178, 184, 185, 188, 201, 202, 203
Output Radial Data in XY Format, 23, 82
Output..., 7, 52, 109, 136
Overlap, 55, 109, 125, 135, 187
Overlap Data..., 55
Overlap Type, 55

P

Padding Factor, 21, 64, 202
Pathway, 30, 109
PEC, 75
Perform Simulation, 7, 109, 125, 136, 139, 150, 157, 163, 188
Periodic, 75, 77
Phase, 43
Phi, 33, 35, 39, 45, 51, 55, 72
Plot Aspect Ratio, 24, 109, 125
PML, 75
PML Kappa, 71, 76
PML Refl., 76
PML Width, 19, 76, 150
polarization, 18, 30, 41, 55
Polarizer Mode, 41, 42
Polarizer..., 41, 187
Position X, 33, 150
Position Y, 33
Position Z, 33, 150
Power, 43, 67, 202, 203
Poynting Vector, 202
Prefix, 30, 55
Preliminary Output Factor, 21, 202
Propagation lengths, 100, 170
Propagation lengths (2^N*dt), 101, 170
Pulse, 136, 165

R

Radial Calculation, 82
Ramp Off Time, 37, 40
Ramp/Pulse Time, 37, 38, 40, 165
Random Set, 30
Rectangle, 29, 30, 44, 45, 165
Reference Type, 109, 135
Relative To, 109, 135
Repeat in Each Unit Cell (non-cubics), 43
Restart, 80
Rotation, 41

S

Save Settings, 7, 109
Scale Files..., 24, 109
Seed center frequencies in scan, 171
Seed center frequencies in scans, 100
Segment Mode, 109, 125, 132
select mode, 132
Settings..., 85, 104
Show Color Scale, 24
Show Window, 85
Shutoff Time, 37, 38, 40
Simulation Tool, 7, 109, 125, 131, 190
Single Precision, 81
Single Value Output, 109, 125, 153
Single-Value, 187
Single-Value Output, 135
Skip final mode profile calculation, 101, 103, 170
Skip Master, 104
Slab Mode, 30, 33, 109, 135
Slice Grid, 147
Slice Grid (X & Y), 134
Slice Position X, 24
Slice Step, 125
Slice Time, 23, 59, 65
Source Norm, 21, 63, 66
Spatial Excitation Type, 29
Spatial Format, 62, 65, 66, 67, 187
Spatial Output, 187
Spatial Output Time, 59
Spatial Time Step, 21, 203
Spectral analysis method, 103, 170
Starting Frequency, 65
Starting Vertex X, 132
Starting Vertex Z, 132
Steady State Field (at Fixed Z), 23, 66, 72
Steady State Field (Fixed Z), 72
Stop Time, 19, 109, 136, 139, 150, 203
Storage Grid, 52
Symmetric, 75, 78, 151

T

T Density, 153
TE, 18
Temporal Excitation Type, 37, 40
Temporal Output, 136
Test..., 85
Theta, 33, 35, 39, 45, 51, 55, 72
Tilt, 30, 35
Time Average, 57, 109, 135, 153
Time Monitor, 153
Time Step, 19, 39, 101, 136, 170
TM, 18
Tolerance, 100
Transformer, 41
Type, 29, 45, 109, 165

U

Undo Last Change, 109
Unit Cell Contents, 147
Unit Peak, 30, 165
Unit Power, 30, 43
Update Time, 19, 24, 109, 136, 197
Use Anisotropic Materials, 71
Use Complex Implementation, 44, 83
Use Master Node, 88
Use Volume Monitor Weight Profile, 58
Utility/Terminal, 153

V

Value, 109
View Editor, 109
View FST, 81
View Graphs, 109, 136, 150, 163, 188
View Grid, 176
View Launch, 37
View Plot, 109

W

Wavelength, 37, 38, 40
Wavelength Monitor, 62, 96
Wavelength Output, 21, 63, 136, 202
Width, 29, 33, 45, 51, 54, 55, 58, 59, 72, 147, 150, 153, 165
Window, 171

X

X, 135

X Domain Max, 134
X Domain Min, 134
X Min, 151

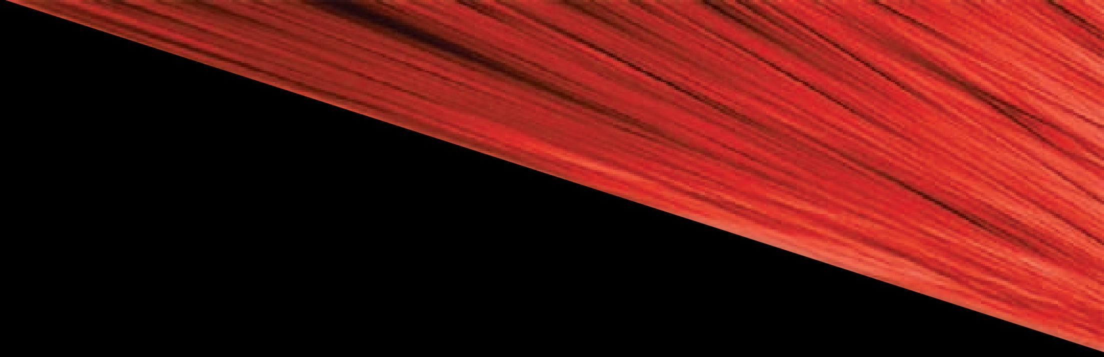
Y

Y, 24
Yellow, 132
Yes, 30, 57, 80, 96, 109, 153

Z

Z, 24, 135
Z Domain Max, 134
Z Domain Min, 134
Z Min, 151
Z Offset Value, 135

THIS PAGE IS INTENTIONALLY LEFT BLANK



RSOFT
Design Group

RSoft Design Group is the worldwide leader in photonic design automation software and serves several industries including optical communication, optoelectronics and semiconductor manufacturing. Within optical communications, RSoft is the only company to provide a full range of design, optimization and network design and planning tools. RSoft also provides award-winning photonic design and simulation software for passive and active optoelectronics components and subsystems as well as advanced electro-magnetic modeling software for optical metrology and lithography.

RSoft Design Group is a privately held corporation with global marketing operations worldwide and offices in the US, Japan and Europe. For the latest information and product updates, visit *www.rsoftdesign.com*.

UNITED STATES
Corporate Headquarters

RSoft Design Group, Inc.
400 Executive Boulevard,
Ste. 100
Ossining, NY 10562, USA

PHONE: 1.914.923.2164
EMAIL: info@rsoftdesign.com
WEB: www.rsoftdesign.com

JAPAN

RSoft Design Group JapanKK
Matsura Building 2F,
1-9-6 Shiba Minato-ku,
Tokyo, 105-0014 Japan

PHONE: + 81.3.5484.6670
EMAIL: info@rsoftdesign.co.jp
WEB: www.rsoftdesign.co.jp

EUROPE

RSoft Design UK, Ltd.
11 Swinborne Drive
Springwood Industrial Estate
Baintree, Essex CM7 2YP

PHONE: 44 (0) 1376 528556
EMAIL: info@rsoftdesign.co.uk
WEB: www.rsoftdesign.co.uk