RSOFT
Design Group

USER GUIDE

BeamPROP™

THE SOURCE FOR
PHOTONIC & NETWORK
DESIGN SOFTWARE

THIS PAGE IS INTENTIONALLY LEFT BLANK

BeamPROP 8.3

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

ph: 1.914.923.2164
info@rsoftdesign.com

www.rsoftdesign.com

BeamPROP 8.3

Revision 110817

Contents

Preface

Foreword

The BeamPROP simulation engine is a part of the RSoft Photonics Suite, and is based on advanced finite-difference beam propagation (BPM) techniques. It is fully integrated into the RSoft CAD environment which allows the user to define the material properties and structural geometry of a device. It is ideal for the design and modeling of photonic devices and photonic integrated circuits. The benefit of good design and modeling tools is well known in the electronics industry, where both device and circuit simulation programs, such as PICSES and SPICE have been instrumental in advancing the availability and use of integrated electronic circuits. BeamPROP brings this important capability to the photonics area, and can be an extremely useful tool for research and development groups in both university and industrial environments.

To use BeamPROP 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 BeamPROP 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, BeamPROP is designed to work with RSoft's other passive device simulation modules FullWAVE, BandSOLVE, GratingMOD, DiffractMOD, FemSIM, and ModePROP. 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 Unix 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 BeamPROP.

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 BPM algorithm. All new users should read these chapters in their entirety.

Basic BeamPROP Usage

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

Advanced BeamPROP Usage

Chapter 6 covers advanced topics such as polarization effects and algorithm options and can be treated as a reference manual and do not need to be read by all new users.

Tutorials

Chapters 7-10 contain tutorial examples that span both basic and advanced BeamPROP topics, mode solving, and the Multi-Physics Utility. The basic tutorials (Chapter 7) 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 the RSoft Photonics Suite, including the CAD and all simulation modules, is covered in the RSoft Installation Guide.

- *Parameter Scanning/Scripting/Batch Operation*

  These topics are very similar, and are shared by all the simulation modules. They are discussed in Chapter 10 of the CAD manual.

- *Computing the Index Profile*

  Computing the index profile is discussed in Section 3.G of the CAD manual.

- *Pathways*

  Pathways define the location of pathway monitors (used by BeamPROP and FemSIM) and the location and geometry of launch, or initial fields, in BeamPROP, FullWAVE, and ModePROP. They are documented in Section 6.G of the CAD manual.

- *Non-Uniform Grids*

  Non-uniform grids can be used by BeamPROP, FullWAVE, and FemSIM and are documented in Chapter 9 of the CAD manual.

- *Command Line Utilities*

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

- *Mode Solving*

  RSoft software contains a variety of mode solvers. See Chapter 12 in the CAD manual for an overview of the mode solving methods available.

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 BeamPROP, 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 the BeamPROP:

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 BeamPROP:

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 executables for the RSoft CAD, BeamPROP, 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`
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 BeamPROP simulation can be started via either the Graphical User Interface (GUI) or a command line interface. Both of these methods are equally powerful, and are useful in different circumstances. Beginning users are encouraged to use BeamPROP via the GUI interface in order to understand the usage of the program. Advanced users who with to incorporate BeamPROP into scripts/batch operations are encouraged to use the scripting capabilities of BeamPROP.

1.D.1. GUI Program Operation

The GUI operation of BeamPROP 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 a CAD window 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 RSoft CAD manual.

Step 2: Choosing the BeamPROP simulation engine

The first step in initiating a BeamPROP simulation is to ensure that the **Simulation Tool** has been set to *BeamPROP* in the Global Settings dialog box.

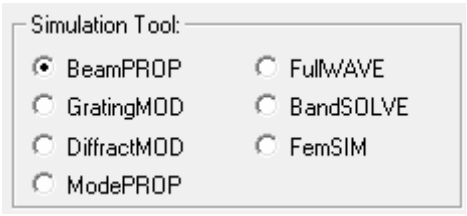

Figure 1-1: The Simulation Tool option as shown in the Global Settings dialog box.

Step 3: Setting up 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 is where the parameters required for a numerical simulation can be entered, and controls other important aspects of the simulation as well. See Chapter 3 for a complete description of this dialog.

The default parameter values can be accepted 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. See Section 3.E for more information about saving simulation data.

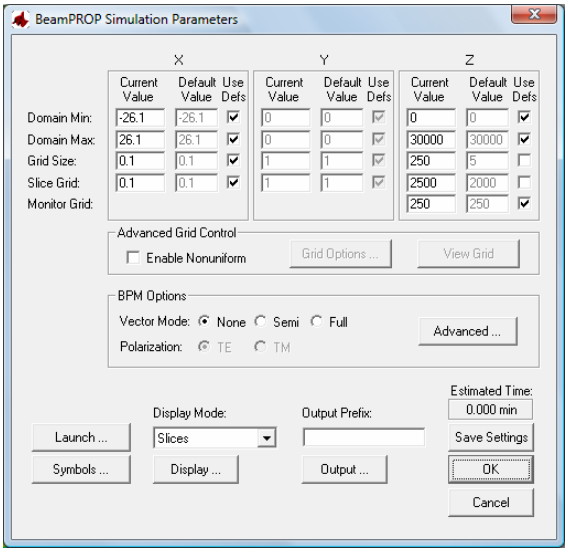

Figure 1-2: The BeamPROP Simulation Parameters window where basic numerical simulation parameters are entered.

Step 4: Running the Simulation

Once the numerical parameters and other options are acceptable, click **OK** to start 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 amplitude and any monitor output is displayed during the calculation. When the simulation is finished, the titlebar 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, and select the desired graph from the dialog that is presented (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.

In addition to simulation results, the program by default will save a *run log,* which is essentially a copy of the `.ind` design file as it existed at the time of the simulation, with default simulation parameters replaced by the specific values used in the run. The run log is stored in the file `prefix.run`, and may be opened by clicking on the open file icon in the top toolbar, selecting *Run Log Files* in the **Files of type** field, and selecting the desired file.

1.D.2. CLI Operation

As an alternative to using the BeamPROP 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 BeamPROP into a larger scripting environment.

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

For a complete discussion of the command line syntax, please see Chapter 10 in the CAD manual.

1.E. Example Files

The subdirectory `<rsoft_dir>\examples\BeamPROP` contains example files for BeamPROP. These files have an extension `*.ind` (named for the refractive index distribution which describes the circuit). A partial listing of these files is as follows:

Example File	Description
`autoscan.ind`	A y-branch setup for automatic parameter scanning
`branch.ind`	A simple y-branch structure
`coupler.ind`	A directional coupler
`fullvect.ind`	A full vector waveguide mode calculation (3D)
`electrod.ind`	A Mach-Zehnder modulator using electrodes
`grating.ind`	A grating structure demonstrating bidirectional BPM
`guass.ind`	A Gaussian beam diffracting in free space
`lens.ind`	A diffracting Gaussian beam focused by a lens
`machzndr.ind`	A Mach-Zehnder modulator in the off state
`modes2d.ind`	A slab waveguide setup for mode calculations (2D)
`modes3d.ind`	An optical fiber setup for mode calculations (3D)
`qcoupl3d.ind`	A pair of coupled waveguides in 3D Fiber or Rib geometry
`switches.ind`	An array of directional coupler switches
`userprof.ind`	A single waveguide with a user-defined index profile
`xcoupl3d.ind`	A crossed pair of waveguides using 3D Multilayer geometry

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 Files

The latest product information can be found in the README files located in the `/readme` subdirectory of the installation directory. These files contain important last minute information about RSoft software that is not contained in other documentation, including new or improved features and options.

1.G. Product Support & 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 on the simulation methods used in BeamPROP.

2.A. Discussion

The objective of BeamPROP is to provide a general simulation package for computing the propagation of light waves in arbitrary waveguiding geometries. This is a complex problem, in general, and several assumptions are made at the outset, many of which are subsequently relaxed. The computational core of the program is based on a finite difference beam propagation method as described in [3,25] and references therein. This technique uses finite difference methods to solve the well-known parabolic or paraxial approximation of the Helmholtz equation. In addition, the program uses "transparent" boundary conditions following [26].

The fundamental physical limitation of the above approach results from the parabolic approximation to the Helmholtz equation, which implies a paraxiality condition on the primary direction of propagation. These limitations can be reduced using more accurate approximations to the Helmholtz equation as outlined in [34]. BeamPROP has the option of implementing this technique and includes (1,0), (1,1), (2,2), (3,3), and (4,4) Padé approximations.

A second limitation of the above approach results from the assumption of scalar waves preventing polarization effects from being considered. BeamPROP has several vector beam propagation techniques to overcome this limitation. These methods are based in part on the approach described in [32,41] and related references.

The third key limitation of the BPM approach described above is that it can not account for backward reflections since the one-way wave equation on which it is based does not admit both positive and negative traveling waves. BeamPROP has a bidirectional BPM algorithm as described in [40], which

considers coupled forward and backward traveling waves, and can account for reflection phenomenon, including resonant effects as found in grating structures.

The physical propagation problem requires two key pieces of information: 1) the refractive index distribution, $n(x,y,z)$ and 2) the input wave field, $u(x,y,z=0)$. From these, the physics dictates the wave field throughout the rest of the domain, $u(x,y,z>0)$. The solution algorithm requires additional input in the form of numerical simulation parameters such as:

- A finite computational domain for x in range (x_{min},x_{max}), y in range (y_{min},y_{max}), and z in range (z_{min},z_{max}).
- The transverse grid sizes, Δx and Δy.
- The longitudinal step size, Δz.

The software attempts to estimate appropriate values for these parameters, but allows the user to override them.

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

For general guidelines on choosing parameters refer to Section 3.G as well as the examples in the tutorial chapter and any other notes given throughout the manual.

In this chapter the concept and capabilities of the Beam Propagation Method, or BPM,[1-3] are reviewed. BPM is the most widely used propagation technique for modeling integrated and fiber optic photonic devices and most commercial software for such modeling is based on it.

There are several reasons for the popularity of BPM; perhaps the most significant being that it is conceptually straightforward, allowing rapid implementation of the basic technique. This conceptual simplicity also benefits the user of a BPM-based modeling tool as well as the implementer, since an understanding of the results and proper usage of the tool can be readily grasped by a non-expert in numerical methods. In addition to its relative simplicity, BPM is generally a very efficient method, and has the characteristic that its computational complexity can, in most cases, be optimal, that is to say the computational effort is directly proportional to the number of grid points used in the numerical simulation. Another characteristic of BPM is that the approach is readily applied to complex geometries without having to develop specialized versions of the method. Furthermore the approach automatically includes the effects of both guided and radiating fields as well as mode coupling and conversion. Finally, the BPM technique is very flexible and extensible, allowing inclusion of most effects of interest (e.g. polarization, nonlinearities) by extensions of the basic method that fit within the same overall framework.

Numerous applications of BPM to modeling different aspects of photonic devices or circuits have appeared in literature. Examples from the authors' own experience include various passive waveguiding devices,[4] channel-dropping filters,[5] electro-optic modulators,[6] multimode waveguide devices,[7,8] ring lasers,[9] optical delay line circuits,[10,11] novel y-branches,[12] optical interconnects,[13] polarization splitters,[14] multimode interference devices,[15-19] adiabatic

couplers,[20] waveguide polarizers,[21] and polarization rotators [22]. Most of the above references involve experimental demonstrations of novel device concepts designed in whole or in part via BPM.

In the following subsections, the basic ideas involved in the BPM as well as the main extensions to the technique are explained and selected theoretical references are given.

2.B. Scalar, Paraxial BPM

BPM is essentially a particular approach for approximating the exact wave equation for monochromatic waves, and solving the resulting equations numerically. In this section the basic approach is illustrated by formulating the problem under the restrictions of a scalar field (i.e. neglecting polarization effects) and paraxiality (i.e. propagation restricted to a narrow range of angles). Subsequent sections will describe how these limitations may be removed.

The scalar field assumption allows the wave equation to be written in the form of the well-known Helmholtz equation for monochromatic waves:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} + k\left(x, y, z\right)^2 \phi = 0 \tag{1}$$

Here the scalar electric field has been written as $E(x,y,z,t) = \phi(x,y,z)e^{-i\omega t}$ and the notation $k(x,y,z) = k_0 n(x,y,z)$ has been introduced for the spatially dependent wavenumber, with $k_0 = 2\pi/\lambda$ being the wavenumber in free space. The geometry of the problem is defined entirely by the refractive index distribution $n(x,y,z)$.

Aside from the scalar assumption, the above equation is exact. Considering that in typical guided-wave problems the most rapid variation in the field ϕ is the phase variation due to propagation along the guiding axis, and assuming that axis is predominantly along the z direction, it is beneficial to factor this rapid variation out of the problem by introducing a so-called slowly varying field u via the ansatz

$$\phi\left(x, y, z\right) = u\left(x, y, z\right) e^{i\bar{k}z} \tag{2}$$

Here k-bar is a constant number to be chosen to represent the average phase variation of the field ϕ and is referred to as the reference wavenumber. Introducing the above expression into the Helmholtz equation yields the following equation for the slowly varying field:

$$\frac{\partial^2 u}{\partial z^2} + 2i\bar{k}\frac{\partial u}{\partial z} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \left(k^2 - \bar{k}^2\right) u = 0 \tag{3}$$

At this point the above equation is completely equivalent to the exact Helmholtz equation, except that it is expressed in terms of u. It is now assumed that the variation of u with z is sufficiently slow so that the first term above can be neglected with respect to the second; this is the familiar slowly varying

envelope approximation and in this context it is also referred to as the paraxial or parabolic approximation. With this assumption and after slight rearrangement, the above equation reduces to:

$$\frac{\partial u}{\partial z} = \frac{i}{2\bar{k}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \left(k^2 - \bar{k}^2\right)u\right) \tag{4}$$

This is the basic BPM equation in three dimensions (3D); simplification to two dimensions (2D) is obtained by omitting any dependence on y. Given an input field, $u(x,y,z=0)$, the above equation determines the evolution of the field in the space $z>0$.

It is important to recognize what has been gained and lost in the above approach. First, the factoring of the rapid phase variation allows the slowly varying field to be represented numerically on a longitudinal grid (i.e. along z) that can be much coarser than the wavelength for many problems, contributing in part to the efficiency of the technique. Second, the elimination of the second derivative term in z reduces the problem from a second order boundary value problem requiring iteration or eigenvalue analysis, to a first order initial value problem that can be solved by simple "integration" of the above equation along the propagation direction z. This latter point is also a major factor in determining the efficiency of BPM, implying a time reduction by a factor of at least of the order of N_z (the number of longitudinal grid points) compared to full numerical solution of the Helmholtz equation.

The above benefits have not come without a price. The slowly varying envelope approximation limits consideration to fields that propagate primarily along the z axis (i.e. paraxiality), and also places restrictions on the index contrast (more precisely, the rate of change of index with z, which is a combination of index contrast and propagation angle). In addition, fields which have a complicated superposition of phase variation, such as exist in multimode devices such as MMI's, may not be accurately modeled if the phase variation is critical to device behavior. A second key issue beyond the above restrictions on the variation of u is that the elimination of the second derivative also eliminates the possibility for backward traveling wave solutions; thus devices for which reflection is significant will not be accurately modeled.

Fortunately, the above issues, which should be considered inherent in the BPM approach, can be eliminated or significantly relaxed in many problems through the use of so-called wide-angle and bi-directional extensions to BPM discussed below. Other restrictions in the above formulation, such as neglect of polarization and simplification of materials properties (e.g. isotropic, linear), are not specific to the BPM approach. Extension of the formulation to address these situations is also considered in subsequent sections. In the following section the numerical solution of the basic BPM equation derived above is considered.

2.C. Numerical Solution and Boundary Conditions

Equation 4 is a parabolic partial differential equation that can be “integrated” forward in z by a number of standard numerical techniques. Most early BPM’s employed a technique known as the split-step Fourier method.[1] Later work demonstrated that for most problems of interest in integrated optics, an

implicit finite-difference approach based on the well-known Crank-Nicholson scheme was superior.[23-25] This approach and its derivatives have become the standard, thus it is reviewed here. It is frequently denoted FD-BPM in the literature, but will be referred to in the following as simply BPM.

In the finite-difference approach, the field in the transverse (xy) plane is represented only at discrete points on a grid, and at discrete planes along the longitudinal or propagation direction (z). Given the discretized field at one z plane, the goal is to derive numerical equations that determine the field at the next z plane. This elementary propagation step is then repeated to determine the field throughout the structure. For simplicity, the approach is illustrated for a scalar field in 2D (xz); extension to 3D is then briefly summarized.

Let u_i^n denote the field at transverse grid point i and longitudinal plane n, and assume the grid points and planes are equally spaced by Δx and Δz apart, respectively. In the Crank-Nicholson method Eq. 4 is represented at the midplane between the known plane n and the unknown plane n+1 as follows:

$$\frac{u_i^{n+1}-u_i^n}{\Delta z}=\frac{i}{2\bar{k}}\left(\frac{\delta^2}{\Delta x^2}+\left(k\left(x_i,z_{n+1/2}\right)^2-\bar{k}^2\right)\right)\frac{u_i^{n+1}+u_i^n}{2} \tag{5}$$

Here δ^2 represents the standard second order difference operator, $\delta^2 u_i = (u_{i+1} + u_{i-1} - 2u_i)$, and $z_{n+1/2} \equiv z_n + \Delta z/2$. The above equation can be rearranged into the form of a standard tridiagonal matrix equation for the unknown field u_i^{n+1} in terms of known quantities, resulting in:

$$a_i u_{i-1}^{n+1} + b_i u_i^{n+1} + c_i u_{i+1}^{n+1} = d_i \tag{6}$$

Expressions for the coefficients in the above are readily derived and can be found in [25]. The tridiagonal nature of Eq. 6 allows rapid solution in order $O(N)$ operations, where is the N is the number of grid points in x.

Since the field can only be represented on a finite computational domain, when the above equation is applied at the boundary points $i = 1$ and N it refers to unknown quantities outside the domain. For these points the above equation must be replaced by appropriate boundary conditions which complete the system of equations. Proper choice of these conditions is critical, since a poor choice can lead to artificial reflection of light incident on the boundary (e.g. radiation) back into the computational domain. For example, simply requiring the field to vanish on the boundary is insufficient since it is equivalent to placing perfectly reflecting walls at the edge of the domain. Several works introduced artificial absorbing material near the edge of the domain, however adjusting the parameters of the absorber to minimize reflection is cumbersome, and artificial reflections persist in many cases since the interface between the problem space and the absorber will also be partially reflective. A commonly used boundary condition is the so-called transparent boundary condition or TBC.[26] The basic approach is to assume that near the boundary the field behaves as an outgoing plane wave, with characteristics (amplitude, direction) that are dynamically determined via some heuristic algorithm. The plane wave assumption allows the field at the boundary point to be related to the adjacent interior point, thus

completing the set of equations. Details on implementation are given in [26]. The TBC is generally very effective in allowing radiation to freely escape the computational domain, however there are problems for which it does not perform well. To address this several other boundary conditions have recently been explored [27-29].

The above numerical solution can be readily extended to 3D, however the direct extension of the Crank-Nicholson approach leads to a system of equations that is not tridiagonal, and requires $O(N_x^2 \cdot N_y^2)$ operations to solve directly which is non-optimal. Fortunately there is a standard numerical approach referred to as the alternating direction implicit or ADI method,[30] which allows the 3D problem to be solved with optimal $O(N_x \cdot N_y)$ efficiency.

In this and the previous section the concept and implementation details of the basic BPM method have been reviewed. In the following sections various methods for extending BPM are summarized, and details of numerical implementation can be found in the corresponding references.

2.D. Including Polarization - Vector BPM

Polarization effects can be included in BPM by recognizing that the electric field E is a vector, and starting the derivation from the vector wave equation rather than the scalar Helmholtz equation.[31,32] In one approach, the equations are formulated in terms of the transverse components of the field (Ex and Ey), and result in the following set of coupled equations for the corresponding slowly varying fields (u_x and u_y):[32]

$$\frac{\partial u_x}{\partial z} = A_{xx}u_x + A_{xy}u_y$$
$$\frac{\partial u_y}{\partial z} = A_{yx}u_x + A_{yy}u_y \qquad (7,8)$$

The A_{ij} are complex differential operators given by:

$$A_{xx}u_x = \frac{i}{2\bar{k}}\left\{\frac{\partial}{\partial x}\left[\frac{1}{n^2}\frac{\partial}{\partial x}\left(n^2 u_x\right)\right] + \frac{\partial^2}{\partial y^2}u_x + \left(k^2 - \bar{k}^2\right)u_x\right\}$$
$$A_{yy}u_y = \frac{i}{2\bar{k}}\left\{\frac{\partial^2}{\partial x^2}u_y + \frac{\partial}{\partial y}\left[\frac{1}{n^2}\frac{\partial}{\partial y}\left(n^2 u_y\right)\right] + \left(k^2 - \bar{k}^2\right)u_y\right\}$$
$$A_{yx}u_x = \frac{i}{2\bar{k}}\left\{\frac{\partial}{\partial y}\left[\frac{1}{n^2}\frac{\partial}{\partial x}\left(n^2 u_x\right)\right] + \frac{\partial^2}{\partial y \partial x}u_x\right\}$$
$$A_{xy}u_y = \frac{i}{2\bar{k}}\left\{\frac{\partial}{\partial x}\left[\frac{1}{n^2}\frac{\partial}{\partial y}\left(n^2 u_y\right)\right] + \frac{\partial^2}{\partial x \partial y}u_y\right\} \qquad (9)$$

The operators A_{xx} and A_{yy} account for polarization dependence due to different boundary conditions at interfaces, and describe such effects as different propagation constants, field shapes, bend loss, etc. for TE and TM fields. The off-diagonal terms involving A_{xx} and A_{yy} account for polarization coupling and hybrid modes due to geometric effects, such as the influence of corners or sloping walls in the cross-sectional structure (effects due to material anisotropy are considered below).

The above equations are generally referred to as describing a full-vectorial BPM. The simplification $A_{xy} = A_{yx} = 0$ gives the important semi-vectorial approximation. In this case the transverse field components are decoupled, simplifying the problem considerably while retaining what are usually the most significant polarization effects. Unless a structure is specifically designed to induce coupling, the effect of the off-diagonal terms is extremely weak and the semi-vectorial approximation is an excellent one.

2.E. Removing Paraxiality – Wide-Angle BPM

The paraxiality restriction on the BPM, as well as the related restrictions on index-contrast and multimode propagation noted earlier, can be relaxed through the use of extensions that have been referred to as wide-angle BPM.[33-35] The essential idea behind the various approaches is to reduce the paraxial limitations by incorporating the effect of the $\partial^2 u/\partial z^2$ term that was neglected in the derivation of the basic BPM. The different approaches vary in the method and degree of approximation by which they accomplish this. The most popular formulation is referred to as the multistep Padé-based wide-angle technique,[34] and is summarized below.

A simple approach for deriving a wide-angle BPM equation is to consider the Helmholtz wave equation written in terms of the slowly varying field (Eq. 3), but before the making the slowly varying envelope approximation by neglecting the $\partial^2 u/\partial z^2$ term. If D denotes $\partial/\partial z$ in this equation then $\partial^2/\partial z^2$ is represented by D^2. Putting aside the fact that D is a differential operator, the equation can now be viewed as a quadratic equation to be solved for D, yielding the following formal solution for a first order equation in z:

$$\frac{\partial u}{\partial z} = i\bar{k}\left(\sqrt{1+P}-1\right)u \tag{10}$$

$$P \equiv \frac{1}{\bar{k}^2}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+(k^2-\bar{k}^2)\right) \tag{11}$$

This equation is referred to as a one-way wave equation, since the first order derivative admits only forward traveling waves (or backward waves if the signs are chosen appropriately, but not both simultaneously). Although restricted to forward propagation, the above equation is still exact in that no paraxiality approximation has been made. The difficulty is that before this equation can be integrated the radical involving the differential operator P must be evaluated. One approach would be to use a Taylor expansion. To first order this leads to the standard paraxial BPM, and to higher order it becomes more accurate and represents one approach to achieving a wide-angle scheme. However expansion via

Padé approximants is more accurate than the Taylor expansion for the same order of terms.[34] This approach leads to the following wide-angle equation:[34]

$$\frac{\partial u}{\partial z} = i\bar{k}\frac{N_m(P)}{D_n(P)}u \tag{12}$$

Here N_m and D_n are polynomials in the operator P, and (m,n) is the order of the approximation. The following table shows several common approximants:

Padé Order (m,n)	N_m	D_n
(1,0)	P/2	1
(1,1)	P/2	1+P/4
(2,2)	P/2+P2/4	1+3P/4+P2/16

When the above equation is employed, larger angles, higher index contrast, and more complex mode interference can be analyzed in both guided wave and free space problems as the Padé order (m,n) is increased.[34,36] Guidelines for using the technique and a discussion of the complex interrelationships between waveguide angle, index contrast, Padé order, reference wavenumber, and grid parameters is discussed in [36].

2.F. Handling Reflections – Bi-directional BPM

While wide-angle BPM allows propagation in a wider cone of angles about the z axis, this cone can only asymptotically approach +/- 90 degrees from the z axis, and can never be extended to handle simultaneous propagation along the negative z axis (i.e. 180 degrees). For this, one must treat the backward traveling waves as a separate, though coupled, part of the problem. Various bi-directional BPM techniques have been considered to address this issue,[37-39] with most focusing on the coupling that occurs through reflection of a wave incident on an interface along z. Here a recent technique that considers multiple interfaces and reflections in a self-consistent and efficient way is reviewed.[40]

In this method, the guided wave propagation problem is divided into regions that are uniform along z, and the interfaces between these regions (problems involving curved sections can be described in this way via a stair-step approximation). At any point along the structure it is considered that both forward and backward waves can exist, which are denoted by $u^+(x,y,z)$ and $u^-(x,y,z)$, respectively. In the uniform regions the forward and backward waves are decoupled, while the interfaces between these regions couple the forward and backward waves due to reflection.

The essential idea in [40] is to employ a transfer matrix approach in which the individual matrices are differential operators. The physical problem generally has the incident (forward) field given at the input of the structure, and the goal is to determine the reflected (backward) field at the beginning and the transmitted (forward) field at the output. The transfer matrix problem, however, is formulated by

assuming that both the forward and backward fields are known at the input of the structure, and an overall transfer matrix, **M**, then describes the system as follows:

$$\begin{pmatrix} u_{out}^{+} \\ u_{out}^{-} \end{pmatrix} = \mathbf{M} \begin{pmatrix} u_{in}^{+} \\ u_{in}^{-} \end{pmatrix} \tag{13}$$

Given incident field (u_{in}^{+}), the above is solved iteratively for reflected field (u_{in}^{-}) such that the backward field at the output is zero (u_{out}^{-}). The transfer matrix **M** describing the entire structure is composed of successive applications of propagation and interface matrices. The propagation matrices describe the uniform regions, and propagate u^{+} and u^{-} independently using normal BPM (either paraxial or wide-angle depending on the situation). The interface matrices relate u^{+} and u^{-} across an interface, and are given by generalized Fresnel formulas involving differential operators employing the Padé approximants used in wide-angle BPM.[40].

2.G. Additional BPM Techniques

There are several additional BPM techniques worth noting. First, while the above discussion has focused on linear, isotropic materials, it is possible to include nonlinear or anisotropic material effects in the BPM as well. Most anisotropic materials are readily dealt with in the context of the full-vectorial BPM described above by extending the definition of the operators A_{ij} to account for the fact that material index is described by a dielectric tensor. [41]

Nonlinear materials can be accommodated by allowing the refractive index appearing in the equations to be a function of the optical field intensity. Only a small adjustment is required in the solution of the resulting finite-difference equations to account for the fact that the index is a function of the unknown field at the next z step. A simple iteration procedure allows a self-consistent solution to the nonlinear difference equations to be obtained, usually in one or two iterations.

Other areas of recent interest in BPM modeling are the use of higher order numerical schemes,[42,43] and the related issue of accurately dealing with dielectric interfaces.[44,45]

2.H. Mode Solving via BPM

Before leaving the subject of BPM, it is worth noting that several useful mode-solving techniques have been developed that are based on BPM; thus a code written to do BPM propagation can be turned into a mode-solver in a relatively straightforward manner. The earliest of these is referred to here as the correlation method, and was used to calculate modes and dispersion characteristics of multimode fibers.[46] More recently, a technique referred to as the imaginary distance BPM has been developed which is generally significantly faster.[47,48] It should be noted that the imaginary distance BPM technique is formally equivalent to many other iterative mode solving techniques;[49,50] the description in terms of BPM is simply a convenience that allows one to leverage existing code and

concepts. The results in [50], which can be duplicated via imaginary distance BPM, have shown excellent agreement with other published data.

In both BPM-based mode-solving techniques a given incident field is launched into a geometry that is z-invariant, and some form of BPM propagation is performed. Since the structure is uniform along z, the propagation can be equivalently described in terms of the modes and propagation constants of the structure. Considering 2D propagation of a scalar field for simplicity, the incident field, $\phi_{in}(x)$, can be expanded in the modes of the structure as

$$\phi_{in}(x) = \sum_m c_m \phi_m(x) \tag{14}$$

The summation should of course consist of a true summation over guided modes and integration over radiation modes, but for brevity the latter is not explicitly shown. Propagation through the structure can then be expressed as

$$\phi(x,z) = \sum_m c_m \phi_m(x) e^{i\beta_m z} \tag{15}$$

In each BPM-based mode-solving technique, the propagating field obtained via BPM is conceptually equated with the above expression to determine how to extract mode information from the BPM results.

As the name implies, in the imaginary distance BPM the longitudinal coordinate z is replaced by $z'=iz$, so that propagation along this imaginary axis should follow

$$\phi(x,z') = \sum_m c_m \phi_m(x) e^{\beta_m z'} \tag{16}$$

The propagation implied by the exponential term in Eq. 15 has become exponential growth in Eq. 16, with the growth rate of each mode being equal to its real propagation constant. The essential idea of the method is to launch an arbitrary field, say a Gaussian, and propagate the field through the structure along the imaginary axis. Since the fundamental mode (m=0) has by definition the highest propagation constant, its contribution to the field will have the highest growth rate and will dominate all other modes after a certain distance, leaving only the field pattern $\phi_0(x)$. The propagation constant can then be obtained by the following variational-type expression:

$$\beta^2 = \frac{\int \phi^* \left(\frac{\partial^2 \phi}{\partial x^2} + k^2 \phi \right) dx}{\int \phi^* \phi \, dx} \tag{17}$$

Higher order modes can be obtained by using an orthogonalization procedure to subtract contributions from lower order modes while performing the propagation.[51] Issues such as optimal choice of launch

field, reference wavenumber, and step size are discussed in [48,50]. Also, an additional correction is added which removes the error due to the fact that we have solved for the eigenvalues of the paraxial

It is important to note that the imaginary distance BPM is not the same as the common technique of performing a standard propagation and waiting for the solution to reach steady state. The latter will only obtain the fundamental mode if the structure is single mode, and generally takes longer to converge. The imaginary distance BPM is closely related to the shifted inverse power method for finding eigenvalues and eigenvectors of a matrix.

In the correlation method, an arbitrary field is launched into the structure and propagated via normal BPM. During the propagation the following correlation function between the input field and the propagating field is computed:

$$P(z) = \int \phi_{in}^{*}(x)\phi(x,z)\,dx \tag{18}$$

Using Eq. 14 and Eq. 15, the correlation function can also be expressed as:

$$P(z) = \sum_{m} |c_m|^2 e^{i\beta_m z} \tag{19}$$

From this expression one can see that a Fourier transform of the computed correlation function should have a spectrum with peaks at the modal propagation constants. The corresponding modal fields can be obtained with a second propagation by beating the propagating field against the known propagation constants via:

$$\phi_m(x) = \frac{1}{L}\int_0^L \phi(x,z)e^{-i\beta_m z} \tag{20}$$

Several corrections to the propagation constants can be made:

- A correction is made which accounts for the error introduced by solving the paraxial equation, and not the exact Helmholtz equation. Further details on the technique are found in [46].
- Second, the imaginary part of the propagation constant can be found by substituting the mode profile in the wave equation and solving for the propagation constant. This not only results in an imaginary value, but a corrected real value as well.

While the correlation method is generally slower than the imaginary distance BPM, it has the advantage that it is sometimes applicable to problems that are difficult or impossible for imaginary distance BPM, such as leaky or radiating modes.

2.I. References

1 M.D. Feit and J.A. Fleck, "Light propagation in graded-index optical fibers", *Appl. Opt*. **17**, 3990 (1978).

2 See references in D. Yevick, "A guide to electric field propagation techniques for guided-wave optics", *Opt. and Quant. Elec.* **26**, S185 (1994).

3 R. Scarmozzino, A. Gopinath, R. Pregla, and S. Helfert, "Numerical Techniques for Modeling Guided-Wave Photonic Devices," *J. Selected Topics in Quantum Electronics* **6**, 150 (2000).

4 L. Eldada, M.N. Ruberto, R. Scarmozzino, M. Levy, and R.M. Osgood, Jr., "Laser-Fabricated Low-Loss Single-Mode Waveguiding Devices in GaAs", *J. Lightwave Tech.* **10**, 1610 (1992).

5 M. Levy, L. Eldada, R. Scarmozzino, R.M. Osgood, Jr., P.S.D. Lin, and F. Tong, "Fabrication of Narrow-Band Channel-Dropping Filters", *Photon. Tech. Lett.* **4**, 1378 (1992).

6 L. Eldada, M.N. Ruberto, M. Levy, R. Scarmozzino, and R.M. Osgood, Jr., "Rapid Direct Fabrication of Active Electro-Optic Modulators in GaAs", *J. Lightwave Tech.* **12**, 1588 (1994).

7 I. Ilic, R. Scarmozzino, R.M. Osgood, Jr., J.T. Yardley, K.W. Beeson, and M.J. McFarland, "Modeling Multimode-Input Star Couplers in Polymers", *J. Lightwave Tech.* **12**, 996 (1994).

8 I. Ilic, R. Scarmozzino, R.M. Osgood, Jr., J.T. Yardley, K.W. Beeson, and M.J. McFarland, and J. Schweyen, "Photopattermed Polymer Multimode 8x8 Star Couplers: Comparative Design Methodologies and Device Measurements", *IEICE Trans. Commun.* **E80-B** (1997).

9 M.C. Shih, M. Hu, M.B. Freiler, M. Levy, R. Scarmozzino, R.M. Osgood, Jr., I.W. Tao, and W.I. Wang, "Fabrication of an InGaAs SQW Circular Ring Laser By Direct Laser Patterning", *Appl. Phys. Lett.* **66**, 2608 (1995).

10 L. Eldada, R. Scarmozzino, R.M. Osgood, Jr., D.C. Scott, Y. Chang, and H.R. Fetterman, "Laser-Fabricated Delay Lines in GaAs for Optically-Steered Phased-Array Radar", *J. Lightwave Tech.* **13**, 2034 (1995).

11 M.H. Hu, Z. Huang, K.L. Hall, R. Scarmozzino, and R.M. Osgood, Jr., "An Integrated Two-Stage Cascaded Mach–Zehnder Device in GaAs", *J. Lightwave Tech.* **16**, 1447 (1998).

12 M. Hu, R. Scarmozzino, M. Levy, and R.M. Osgood, Jr., "A low-loss and compact waveguide y-branch using refractive index tapering", *Photon. Tech. Lett.*, **9**, 203 (1997).

13 R. Scarmozzino, R.M. Osgood, Jr., L. Eldada, J.T. Yardley, Y. Liu, J. Bristow, J. Stack, J. Rowlette, and Y.S. Liu, "Design and Fabrication of Passive Optical Components for Multimode Parallel Optical Links", *SPIE Photonic West Meeting, San Jose, CA*, **3005**, 257 (1997).

14 M. Hu, J.Z. Huang, R. Scarmozzino, M. Levy, and R.M. Osgood, Jr., "Tunable Mach-Zehnder polarization splitter using height-tapered y-branches", *Photon. Tech. Lett.* **9**, 773 (1997).

15 D.S. Levy, Y.M. Li, R. Scarmozzino, and R.M. Osgood, Jr., "A Multimode Interference-Based Variable Power Splitter in GaAs–AlGaAs", *Photon. Tech. Lett.* **9**, 1373 (1997).

16 D.S. Levy, R. Scarmozzino, Y.M. Li, and R.M. Osgood, Jr., "A New Design for Ultracompact Multimode Interference-Based 2x2 Couplers", *Photon. Tech. Lett.* **10**, 96 (1998).

17 J. Z. Huang, M. H. Hu, J. Fujita, R. Scarmozzino, and R. M. Osgood, Jr., "High-Performance Metal-Clad Multimode Interference Devices for Low-Index-Contrast Material Systems", *Photon. Tech. Lett.* **10**, 561 (1998).

18 J. Z. Huang, R. Scarmozzino, and R. M. Osgood, Jr., "A New Design Approach to Large Input/Output-Number Multimode Interference Couplers and Its Application to Low-Crosstalk WDM Routers", *Photon. Tech. Lett.* **10**, 1292 (1998).

19 D.S. Levy, K.H. Park, R. Scarmozzino, R.M. Osgood, Jr., C. Dries, P. Studenkov, and S. Forrest, "Fabrication of ultracompact 3-dB 2x2 MMI power splitters", *Photon. Tech. Lett.* **11**, 1009 (1999).

20 T.A. Ramadan, R. Scarmozzino, and R.M. Osgood, Jr., "Adiabatic Couplers: Design Rules and Optimization", *J. Lightwave Tech.* **16**, 277 (1998).

21 J. Fujita, M. Levy, R. Scarmozzino, R.M. Osgood, Jr., L. Eldada, and J.T. Yardley, "Integrated Multistack Waveguide Polarizer", *Photon. Tech. Lett.* **10**, 93 (1998).

22 J. Z. Huang, R. Scarmozzino, G. Nagy, M. J. Steel and R. M. Osgood, Jr., "Realization of a compact and single-mode optical passive polarization converter", *Photon. Tech. Lett.* **12**, 317 (2000).

23 D. Yevick and B. Hermansson, "Efficient beam propagation techniques", *J. Quantum Electron.* **26**, 109 (1990).

24 Y. Chung and N. Dagli, "An assessment of finite difference beam propagation method", *J. Quantum Electron* **26**, 1335 (1990).

25 R. Scarmozzino and R.M. Osgood, Jr., "Comparison of finite-difference and Fourier-transform solutions of the parabolic wave equation with emphasis on integrated-optics applications", *J. Opt. Soc. Amer. A* **8**, 724 (1991).

26 G.R. Hadley, "Transparent boundary condition for the beam propagation method", *Opt. Lett*. **16**, 624, (1991); G.R. Hadley, "Transparent boundary condition for the beam propagation method", *J. Quantum Electron* **28**, 363 (1992).

27 C. Vassalo and F. Collino, "Highly efficient absorbing boundary condition for the beam propagation method", *J. Lightwave Technol.***14**, 1570 (1996).

28 W.P. Huang, C.L. Xu, W. Lui, and K. Yokoyama, "The perfectly matched layer (PML) boundary condition for the beam propagation method", *Photon. Technol. Lett.* **8**, 649 (1996).

29 Y.P. Chiou and H.C. Chang, "Complementary operators method as the absorbing boundary condition for the beam propagation method", *Photon. Technol. Lett.* **8**, 976 (1998).

30 W.H. Press, B.P. Flannery, S.A. Teuklsky, and W.T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, (Cambridge Univ., New York, 1986).

31 R. Clauberg and P. Von Allmen, "Vectorial beam propagation method for integrated optics", *Electron. Lett.* **27**, 654 (1991).

32 W.P. Huang and C.L. Xu, "Simulation of three-dimensional optical waveguides by a full-vector beam propagation method", *J. Quantum Electron.* **29**, 2639 (1993).

33 D. Yevick and M. Glasner, "Analysis of forward wide-angle light propagation in semiconductor rib waveguides and integrated-optic structures", *Electron. Lett.* **25**, 1611 (1989).

34 G.R. Hadley, "Wide-angle beam propagation using Pade approximant operators", *Optics Letters* **17**, 1426, (1992); G.R. Hadley, "Multistep method for wide-angle beam propagation", *Optics Letters* **17**, 1743 (1992).

35 H.J.W.M. Hoekstra, G.J.M. Krijnen, and P.V. Lambeck, "New formulations of the beam propagation method based on the slowly varying envelope approximation", *Optics Communications*, **97**, 301 (1993).

36 I. Ilic, R. Scarmozzino, and R.M. Osgood, Jr., "Investigation of the Pade approximant-based wide-angle beam propagation method for accurate modeling of waveguiding circuits", *J. Lightwave Technol.* **14**, 2813 (1996).

37 P. Kaczmarski and P.E. Lagasse, "Bidirectional beam propagation method", *Electron. Lett.* **24**, 675 (1988).

38 Y. Chung and N. Dagli, "Modeling of guided-wave optical components with efficient finite-difference beam propagation methods", *Tech. Dig. IEEE AP-S Int. Symp., 1992* **1**, 248 (1992).

39 Y. Chiou and H. Chang, "Analysis of optical waveguide discontinuities using the Pade approximants", *Photon. Technol. Lett.* **9**, 964 (1997).

40 H. Rao, R. Scarmozzino, and R.M. Osgood, Jr., "A bidirectional beam propagation method for multiple dielectric interfaces", *Photon. Technol. Lett.* **11**,830 (1999).

41 C.L. Xu, W.P. Huang, J. Chrostowski, and S.K. Chaudhuri, "A full-vectorial beam propagation method for anisotropic waveguides", *J. Lightwave Tech.* **12**, 1926 (1994).

42 J. Yamauchi, J. Shibayama, and H. Nakano, "Modified finite-difference beam propagation method based on the generalized Douglas scheme for variable coefficients", *Photon. Technol. Lett.* **7**, 661 (1995).

43 G.R. Hadley, "Low-truncation-error finite difference equations for photonics simulation I: beam propagation", *J. Lightwave Technol.* **16**, 134 (1998).

44 H.J.W.M. Hoekstra, G.J.M. Krijnen, and P.V. Lambeck, "Efficient interface conditions for the finite difference beam propagation method", *J. Lightwave Technol.* **10**, 1352 (1992).

45 J. Yamauchi, M. Sekiguchi, O. Uchiyama, J. Shibayama, and H. Nakano, "Modified finite-difference formula for the analysis of semivectorial modes in step-index optical waveguides", *Photon. Technol. Lett.* **9**, 961 (1997).

46 M.D. Feit and J.A. Fleck, "Computation of mode properties in optical fiber waveguides by a propagating beam method", *Applied Optics*, **19**, 1154 (1980).

47 D. Yevick and B. Hermansson, "New formulations of the matrix beam propagation method: Application to rib waveguides", *J. Quantum Electron.*, **25**, 221 (1989).

48 S. Jungling and J.C. Chen, "A study and optimization of eigenmode calculations using the imaginary-distance beam-propagation method", *J. Quantum Electron.* **30**, 2098 (1994).

49 D. Yevick and Witold Bardyszewski, "Correspondence of variational finite-difference (relaxation) and imaginary-distance propagation methods for modal analysis", *Opt. Lett.* **1**7, 329 (1992).

50 G.R. Hadley and R.E. Smith, "Full-vector waveguide modeling using an iterative finite-difference method with transparent boundary conditions", *J. Quantum Electron.* (1995).

51 J.C. Chen and S. Jungling, "Computation of higher-order waveguide modes by the imaginary-distance beam propagation method", *Optical and Quantum Electron.* **26**, S199 (1994).

3

Using BeamPROP

This chapter describes the key features of the BeamPROP simulation engine.

Familiarity with the CAD is assumed; see the CAD manual for instructions on creating design files.

3.A. Computation Domain & Grid

BeamPROP solves for the electromagnetic fields within a given structure within a computation domain on a spatial grid. It is therefore important to define this grid correctly to ensure correct results. The upper portion of the BPM Simulation Parameters dialog allows the user to specify information on the numerical parameters in each spatial direction.

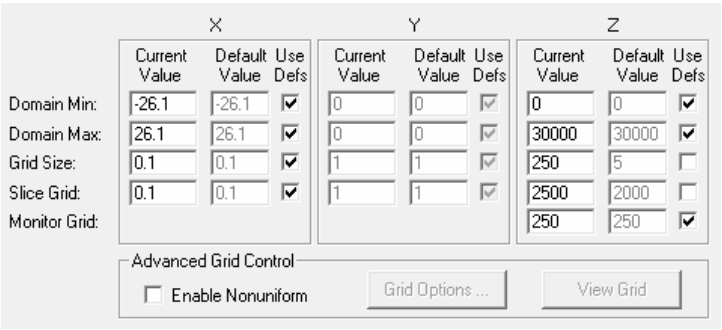

Figure 4-1: The section of the BPM Simulation Parameters window where spatial grid information is set. The non-uniform portion of this dialog is described in Chapter 9 of the CAD manual.

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

Domain Min and **Domain Max**

These fields set 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.

The **Boundary Location** and **Boundary Gap** fields described Section 6.B.3 can also be used determine the default computational domain along X.

Grid Size

These fields set the grid size for each dimension. It is recommended that you perform a convergence study with the grid size to ensure accurate results. When starting a project, you can save time during initial exploratory studies by using a coarse grid and sacrificing accuracy. General results for your structure can often be obtained with remarkably coarse grids. You have not completed your design project until you have confirmed the convergence of your results at a higher resolution.

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

Slice Grid

These fields set the grid on which simulation results are displayed and saved. While using values greater than the **Grid Size** can save time, it can also hide potentially valuable simulation information.

Monitor Grid

This field, which is only set for the Z coordinate, specifies the grid which is used for displaying and saving the pathway monitor results. See Chapter 4 for more information on pathways monitors. While using values greater than the **Grid Size** can save time, it can also hide potentially valuable simulation results.

3.B. Default Boundary Conditions

In addition to the computational domain and grid, BeamPROP also needs to define how the fields behave at the boundaries of the selected domain. By default, BeamPROP uses Full Transparent Boundary Conditions (TBC). This type of boundary is designed to let radiation pass through the boundary without reflection back into the simulation domain. For a description of this boundary condition and it's options as well as other boundary conditions available, please see Section 6.B.3.

3.C. Polarization & Wide-Angle Options

BeamPROP includes many options that control how polarization is handled, including scalar calculations, TE, TM, quasi-TE, quasi-TM, semi-vector, and full-vector options. By default, BeamPROP will perform a scalar calculation which is equivalent to TE in 2D and is valid for low contrast cases in 3D. This default is usually a good starting point for a design, though additional options might be necessary depending on the exact application. See Section 6.A for more details about modeling polarization.

The basic default BPM algorithm does not allow for off-axis propagation. BeamPROP does contain an array of wide-angle options that help relax this restriction and allow for many devices to be successfully simulated. See Section 6.B.2 for a description of these options. A theoretical background for these options can be found in Chapter 2.

3.D. Defining the Launch Field

The launch field is the initial condition for the simulation and is set via the Launch Parameters dialog shown in Fig. 4-2. This dialog can be opened either via the **Launch…** button located in the Simulation Parameters dialog or via the **Edit Launch Field** button in the left CAD toolbar.

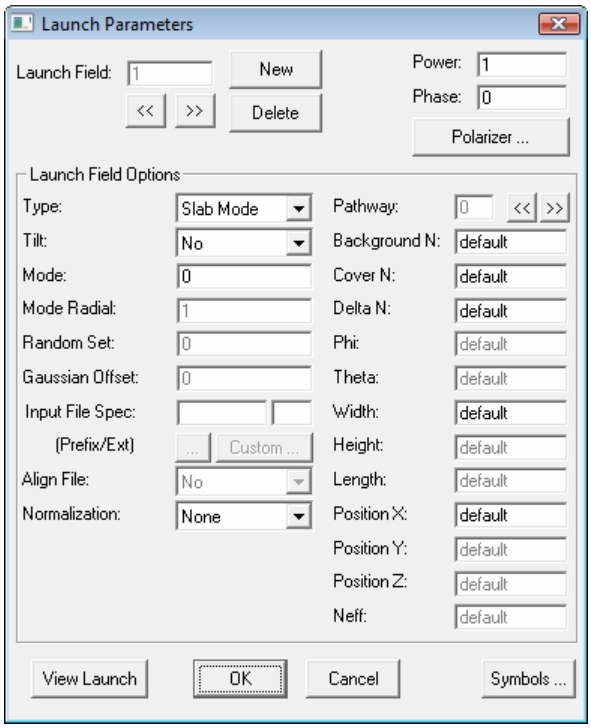

Figure 4-2: The Launch Parameters dialog which selects both the type and the characteristics of the optical input field.

The launch field is represented as a sum of individual excitations and is defined as:

$$\phi_L(\mathbf{r}) = \sum_i \phi_i(x, y, z_0)$$

where each excitation field $\phi_i(x,y,z_0)$ will be referred to as simple $f(x,y)$ through the rest of this section. The launch field has no z-dependence since it is an initial condition for the integration along Z as described in Chapter 2. The function $f(x,y)$ can be divided into two main sections: the desired field profile type, and then optional launch data which describe the desired profile.

3.D.1. Choosing the Field Profile

The options shown in the left column of the dialog in Fig. 4-2 control the type of field profile for the launch field.

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 file(s) are specified via the **Input File Spec** option described below.

- *Computed Mode*

  This option first computes the mode of the input structure using the iterative method, stores the result in a file, and then launches that mode into the structure to perform a regular calculation. More information on this mode solving method can be found in Chapter 5. Note that the step size used for the mode calculation can be set independently from the step size in the propagation simulation using the variable `step_size_idbpm`.

- *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 launch field and is the default. The characteristics of the field are, by default, set via the input component in the CAD, but can be set directly via the Optional 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 input field will be a Gaussian function defined as

  $$f(x,y) = e^{\left(-x^2/a^2\right)} e^{\left(-y^2/b^2\right)}$$

  where $a=w/2$, $b=h/2$, and w and h are the $1/e$ width/height of the Gaussian which are set to the dimensions of the input component in the CAD but can be set directly via 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 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 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 **Mode** field described below.

Tilt

This option enables the launch field to be tilted so as to enter the simulation domain at an angle. By default this option is set to *No* and the optical field is launched along the +Z axis. If this option is set to *Yes*, the optical wavefront is tilted to an angle equal to that of the input component, unless it is overridden by the value in the **Phi** and **Theta** fields described below.

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

Sets the offset from the Gaussian waist for a **Launch Type** of *Gaussian*. A positive value effectively places the waist in front of' the launch plane; a negative value effectively places the waist behind. When using this option the width and 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**, **Polarization**, and **Vector Mode**, 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. See Section 6.A for details. 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**, **Polarization**, and **Vector Mode**. Generally users will only use the major component unless **Vector Mode** has been set to *Full*. See Section 6.A for a complete definition.

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. For an example, see Tutorial 4.

Align File

This option controls alignment of the launch field with the **Pathway** when the input type is *File* or *Computed Mode*. 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.

3.D.2. Using Optional Field Profile Data

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. Phi is measured from the Z axis and is defined in the XZ plane. Theta is measured from the Z axis, is defined in the YZ plane. Both angles have default values of `0` and require the **Tilt** option to be enabled in order to use.

Width and **Height**

These options set the width (measured along X) and height (measured along Y) of the input field to be launched. The default value corresponds to the width of the input component. The **Length** field is not used by BeamPROP.

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

These options set the X and Y position at which the input field is launched. For an input type of *File* this option assumes that the field in the data file is centered at 0 and will effectively offset the launch file data. Position Z is not used by BeamPROP

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.

3.D.3. Polarization Manipulation

BeamPROP allows for the inclusion of polarization effects in a simulation through the use of Vector BPM. See Section 6.A for details.

3.D.4. 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. 4-2. 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.

3.D.5. Viewing Launch Field

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

3.E. Saving Data From a Simulation

BeamPROP includes many options to save data from a simulation. These options are controlled by the Output Options dialog shown in Fig. 4-3 which can be opened by clicking the **Output…** button in the BeamPROP Simulation Parameters dialog.

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

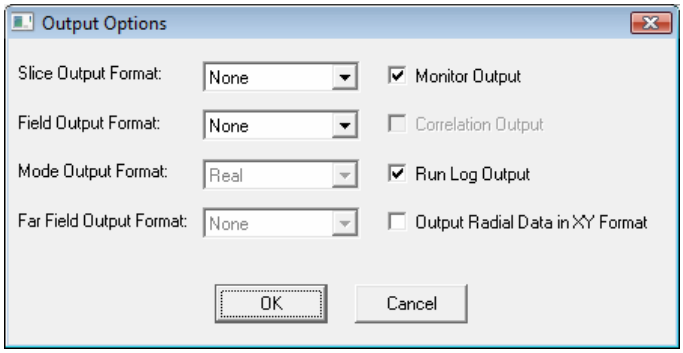

Figure 3-3: The Output Options dialog.

The following options are available:

Slice Output Format

This output saves field data from each Z slice step. The default depends on the simulation options and display mode. It can be useful to set this option to *None* to avoid large amounts of unwanted data to be saved, or at least increase the Z **Slice Step**. Slices are saved with the names `<prefix>_<comp>.###`, where `###` is the slice number, `<prefix>` is the **Output Prefix**, and `<comp>` is a tag (only used when appropriate) indicating the field component. Corresponding WinPLOT command files are also produced for graphing the standard display of slices and any monitor output.

Field Output Format

This output saves the field at the final Z position and has a default value of *None*. The field is saved in a data file `<prefix>_<comp>.fld` with accompanying WinPLOT command file `<prefix>_<comp>.pfd` where `<prefix>` is the **Output Prefix** and `<comp>` is the field component.

The exact components saved depend on the **Model Dimension**, **Polarization**, and **Vector Mode**; see Section 6.A for details.

Mode Output Format

This output sets mode output format saves modes for a mode calculation and has a default value of *Real*. See Chapter 5 for mode solving information. Mode files are saved in data files `<prefix>_<comp>.m##` with accompanying WinPLOT command files `<prefix>_<comp>.p##` where `<prefix>` is the **Output Prefix**, `<comp>` is the field component, and `##` is the mode number. For example the file `<prefix>_ey.m00` contains the E_y component of the fundamental mode. The exact components saved depend on the **Model Dimension**, **Polarization**, and **Vector Mode**; see Section 6.A for details.

Far Field Output

See Section 6.H for details on computing far fields.

Monitor Output

This controls the output of pathway monitor data if pathway monitors have been defined. The raw monitor data is saved in a file `<prefix>.mon`, with an accompanying WinPLOT command file `<prefix>.pmn` where `<prefix>` is the **Output Prefix**. See Chapter 4 for more information on the use of monitors.

Correlation Output

This option enables the output of correlation function data from either a mode spectrum calculation or a mode calculation using the correlation method. Note that this option is only provided for backward compatibility and may be removed in future versions.

Run Log Output

This option instructs the program to save a *run log* which is essentially a copy of the `.ind` file as it existed at the time of the simulation with default simulation parameters replaced by the specific values used in the run. The run log is stored in the file `prefix.run`, and may be opened by clicking on the open file icon in the top toolbar, selecting *Run Log Files* in the **Files of type** field, and selecting the desired file.

Output Radial Data in XY Format

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

3.F. Displaying Data During a Simulation

During a BeamPROP simulation the simulated field along with the pathway monitor results are displayed and updated during the simulation at intervals defined by the Z **Slice Step** described in Section 3.A. This data display can be controlled by the **Display Mode** option in the Simulation Parameters dialog and via the options in the Display Options dialog.

3.F.1. Setting the Display Mode

The **Display Mode** drop down box lists various options for displaying the optical field during the simulation. They are:

Note that some display options do not function in certain simulation modes.

- *None*

  No graphical display.

- *Slices*

  The field amplitude is displayed in a 2D graph showing the transverse profile at several positions along the propagation direction. In bidirectional simulations, the forward and backward fields are displayed independently in blue and green, respectively.

- *3D Slices*

  The field amplitude as a function of x and z is displayed as a series of slices (at different z) in a 3D representation.

- *WireFrame*

  The field amplitude as a function of x and z is displayed as a 3D wire frame graph.

- *SolidModel*

  The field amplitude as a function of x and z is displayed as a 3D solid model with shading to portray an illuminated surface. This option is only available on displays with 256 or more colors.

- *HeightCoded*

  The field amplitude as a function of x and z is displayed as a 3D contour graph with color coding to indicate height (amplitude). The color scale can be selected from among pre-defined or user-defined scales via the **Display…** button. The display options dialog also controls whether or not the color scale is shown along with the graph.

- *ContourMap (XZ)*

  The field amplitude as a function of x and z is displayed as a 2D color-coded contour map. The color scale can be selected from among pre-defined or user-defined scales via the **Display…** button. The display options dialog also controls whether or not the color scale is shown along with the graph, and has an option for displaying the outline of the waveguide circuit. The **Slice Position Y** option in the display options dialog controls the vertical position at which the field is viewed if the simulation is done in 3D. Note that while it is not currently possible to view the XZ and YZ plots simultaneously, it is possible to have both plots stored by setting the variable `slice_output_both` to `1` and setting the **Output Prefix**.

  Finally, in bidirectional calculations, this option displays the sum of the forward and backward fields. At the end of the calculation, the user can select View/Refresh to normalize the plot.

- *ContourMap (YZ)*

  This option is only applicable to 3D simulations, and displays the field amplitude as a function of y and z as a 2D color-coded contour map. The color scale can be selected from among pre-defined or user-defined scales via the **Display…** button. The display options dialog also controls whether or not the color scale is shown along with the graph, and has an option for displaying the outline of the waveguide circuit. The **Slice Position X** option in the display options dialog controls the horizontal position at which the field is viewed. Note that while it is not currently possible to view the XZ and YZ plots simultaneously, it is possible to have both plots stored by setting the variable `slice_output_both` to `1` and setting the **Output Prefix**.

- *ContourMap (XY)*

  This option is only applicable to 3D field simulations, and displays a dynamic 2D color-coded contour map of the transverse field profile as a function of x and y. The color coding is discrete, enabling each grid point to be discerned. A continuous version of this option is used in displaying mode fields. In that case, the color scale can be selected from among pre-defined or user-defined scales via the **Display…** button. The display options dialog also controls whether or not the color scale is shown along with the graph, and has an option for displaying the outline of the waveguide cross-section. Finally, when the simulation program has finished, you can view a horizontal or vertical cut of the field by right-clicking along the y or x axis at the desired position; a second right-click anywhere in the window switches back to the original plot.

  Note that the XY plots do not normally display in a 1:1 aspect ratio, however the display options dialog also contains an option for fixing 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.

3.F.2. The Display Options dialog

Many display options can be set in the Display Options dialog, shown in Fig. 3-4, which can be opened by clicking the **Display…** button in the Simulation Parameters dialog.

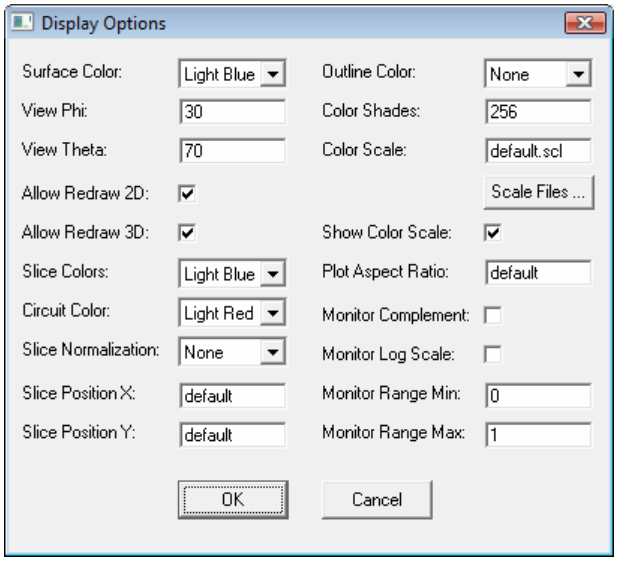

Figure 3-4: The display options dialog which accesses additional display related options.

These options are:

Surface Color

This sets the surface color for plots produced when the **Display Mode** is set to *SolidModel*.

View Phi and **View Theta**

These options set the viewing angles Phi and Theta of the plots in the simulation window when the **Display Mode** is set to either *SolidModel* or *HeightCoded*.

Allow Redraw 2D and **Allow Redraw 3D**

These option indicates that 2D and 3D plots displayed in the simulation window should be stored in memory so that the plots can be redrawn if needed. This option was used in early versions of BeamPROP to optimize memory usage on computer systems with minimal RAM. Today's systems allow for this option to always be left on.

Slice Color

This sets the color of the field slices displayed in the simulation window.

Circuit Color

This sets the color of the circuit displayed behind the simulated fields in the simulation window.

Slice Normalization

This option has two settings: *None* and *Constant Peak*. This choice indicates whether the slices should be normalized such that they all have the same peak or not.

Slice Position X and **Slice Position Y**

These options control the X and Y slice positions when the **Display Mode** is set to *ContourMap (YZ)* and *ContourMap (XZ)* respectively.

Outline Color

This sets the color of the circuit outline to be displayed over the simulated fields. By default, this option is set to *None*.

Color Shades

This option sets the number of color shades used to render plots in the simulation window.

Color Scale

This option sets the color scale used to render plots in the simulation window. Color Scale files can be chosen via the **Color Scales…** button.

Show Color Scale

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

Plot Aspect Ratio

This option sets the aspect ratio of the plot in the simulation window.

Monitor Complement

This option indicates that the monitors should be displayed as the complement of the actual monitor data. The data displayed will equal 1-*data* where *data* is the normal monitor result. This option can be useful when measuring propagation loss for example.

Monitor Log Scale

This option indicates that the monitors should be displayed as the log of the actual monitor data. The data displayed will equal log(*data*) where *data* is the normal monitor result. This option can be useful for certain simulations.

Monitor Range Min and **Monitor Range Max**

These options control the range used for pathway monitor plots and have a default value of `0` and `1` respectively.

3.G. Guidelines for Choosing BeamPROP Simulation Parameters

The options described in this section are described in this chapter as well as in Chapter 6.

Generally, if the estimated time produced by BeamPROP is reasonable, it is recommended to start working with these defaults. If the defaults are not reasonable, use coarser values until the time is

reasonable and run the problem to see if it makes physical sense. Refine the parameters as needed to obtain accuracy or speed.

The guidelines for using BeamPROP presented in this section consider several parameters that can be chosen when performing a simulation and the influences that these parameters have on the results of the calculation. Also noted are notes about the scaling of the simulation time since there is always a tradeoff between accuracy/stability and computational resources.

Effect of Grid Sizes (X and Y)

Smaller grid sizes improve the accuracy of BeamPROP for both propagation analysis and mode calculation. Stability is not affected except in the case of full-vector BPM. Smaller grid sizes tend to produce difficulty with transparent boundary conditions (unwanted reflections) or with fields at interfaces (artifacts such as energy "stuck" near an interface). These problems are alleviated with the use of smaller step size (Δz). Simulation time will scale directly with the number of grid points used.

Effect of Step Size (Z)

For propagation analysis with BeamPROP as well as mode solving via the correlation method, a smaller step size will improve accuracy, will not affect stability, and will also alleviate performance problems of transparent boundary conditions and propagation near interfaces at small grid sizes. Simulation time will scale directly with the number of grid points used. To some extent, the step size can be traded against the reference wavenumber (k-bar), since accuracy is in part related to:

$$\left(\overline{k} - k_z\right) \cdot \Delta z$$

For mode solving via the iterative method, smaller step sizes have no effect on accuracy or stability. A larger step size will improve the rate of convergence, but too large a step size can cause convergence to unwanted modes. The default step size is chosen near this limit; if problems occur, reduce the step size by factors of 2 until the problems resolves.

Effect of Reference Wavenumber (k-bar)

For propagation analysis with BeamPROP as well as mode solving via the correlation method, a reference wavenumber (k-bar) that is close to k_z (the actual behavior of the field) will lead to more accurate results. The best choice for the reference wavenumber (k-bar) is usually βcosθ where β is the propagation constant of the main mode involved in the propagation and θ is the angle of propagation relative to the z axis. To some extent, the reference wavenumber can be traded against the step size, since accuracy is in part related to

$$\left(\overline{k} - k_z\right) \cdot \Delta z$$

Similarly (for propagation only), error in the reference wavenumber (k-bar) can be traded against the Padé Order since higher Padé Order reduces the effect of the above error term.

For mode solving via the iterative method the reference wavenumber (k-bar) has no significant effect since it is internally dynamically updated through the calculation.

Simulation time does not depend on the reference wavenumber (k-bar).

Effect of Padé Order

For propagation analysis with BeamPROP, a higher Padé order will improve accuracy at 'large' angles, 'large' index contrast, and when a 'large' range of propagation constants are involved (e.g. multimode devices). The definition of these three parameters are mixed making it impossible to say what 'large' is in general. The effect of the Padé order on accuracy is intermixed with the effect of the grid and step sizes as well as the reference wavenumber (k-bar). A careful convergence study must be undertaken to optimize all the parameters. The simulation time will scale with the first number of the Padé approximation, that is a simulation with a Padé order of (2,0) will take twice as long as a simulation with a Padé order of (1,0).

For mode calculations, a higher Padé Order has no effect on accuracy and should not be used.

Effect of Computational Domain & Boundary Conditions

The effect of these options is generally less critical since the transparent boundary condition (TBC), when it is working properly, should make the effect of the boundary position immaterial.

When the TBC has difficulty, it can sometimes help to choose a larger computational domain. Even if the TBC is working properly, and the field is accurately being calculated in the domain, if you are computing overlap integrals (e.g. via monitors) with fields that significantly extend beyond the domain, the results may not be accurate. In this case, extend the domain to encompass the important fields.

There are two forms of the TBC, Full and Simple. The default Full TBC attempts to be more intelligent, and can be more accurate, but is sometimes fooled and performs worse than the Simple TBC. When having difficulties, particularly with multimode problems or problems with large amounts of radiation, the Simple TBC should be tried, followed by choosing a smaller step size. If these fail, BeamPROP has several undocumented options which control finer aspects of the transparent boundary conditions; contact RSoft for additional information.

Effect of Polarization Options

The primary polarization control used by BeamPROP is the **Vector Mode** which affects the accuracy and sets which polarization effects are handled. The three modes are:

See Section 6.A for more information about choosing a vector mode.

- *None (Scalar)*

  This mode ignores polarization effects. In 2D, a TE polarization is performed, and in 3D, a scalar simulation is performed. This is appropriate for problems with 'small' index contrasts where polarization effects are weak, or as a starting point to determine whether a device concept (not based on polarization) will work in principle.

- *Semi-Vector*

This mode will include polarization dependent effects, but polarization coupling and hybrid polarization effects are ignored. This is appropriate and sufficient for almost all 'large' index contact problems for which polarization effects are of interest. Simulations are only slightly more complicated and time-consuming than for a vector mode of none.

- *Full-Vector*

  This mode will include all polarization effects, but simulations are significantly more complicated and time consuming than other vector modes. Generally not much more information is obtained except for devices designed to couple polarizations. Stability can be an issue and is controlled by the scheme parameter (see next section).

A general guideline is to start with scalar simulations, even if your problem requires more sophistication, and only after your problem starts making sense, work your way up to semi- or full-vector as needed.

Effect of Scheme Parameter

The scheme parameter controls stability which becomes an issue for 'large' index contrast and/or "small" grid sizes. To overcome the above instability, the scheme parameter must be set to a value greater than its default of 0.5 and less than 1.0. A side effect of using a scheme parameter is artificial numerical dissipation; an eigenmode launched into the structure will attenuate instead of propagating without loss. To minimize this dissipation, choose a scheme parameter as close to 0.5 as possible while maintaining stability. To reduce the dissipation at a given scheme parameter, reduce the step size Δz.

4

Monitoring and Analysis

BeamPROP calculates the steady-state optical field throughout an entire design file. It is frequently useful to analyze this field in standard physical quantities such as the optical power in a particular region of the circuit or the power traveling in a particular mode.

Pathway monitors are what makes these types of measurements possible. Results are output as a function of the propagation direction Z and are measured along a pathway. Monitors can be used to measure the power in the propagating field via overlap integrals or power integrals, measure the phase of a propagating field relative to a test field, and measure other output such as spot size. This chapter will discuss the steps for creating a monitor along with a detailed description of the usage of monitors.

4.A. Defining a Pathway

The first step to create a pathway monitor is to define a pathway via the **Edit Pathways** button located in the left toolbar of the CAD interface. More details can be found in the RSoft CAD manual.

Note that pathways can include inactive components. An inactive component has its **Profile Type** set to *Inactive* as described in the RSoft CAD manual. This can be useful if it is necessary to monitor the power in a region which does not contain or is otherwise different from a true waveguide/fiber region.

4.B. Creating a Pathway Monitor

To add a pathway monitor to a pathway that has just been defined, click on the **Monitors...** button in the **Pathway Edit Mode** toolbar. If the pathway has previously been defined, click on the Edit Pathway Monitors button in the left CAD toolbar. In both cases the Pathway Monitors dialog appears as in Fig. 5-1.

Creating a New Pathway Monitor

To define a new pathway monitor, click the **New** button. The displayed **Monitor Number** will increase by 1. Each monitor requires a **Monitor Type** as well as additional options which are described in the next sections. Additional pathway monitors can be defined by repeating the above procedure.

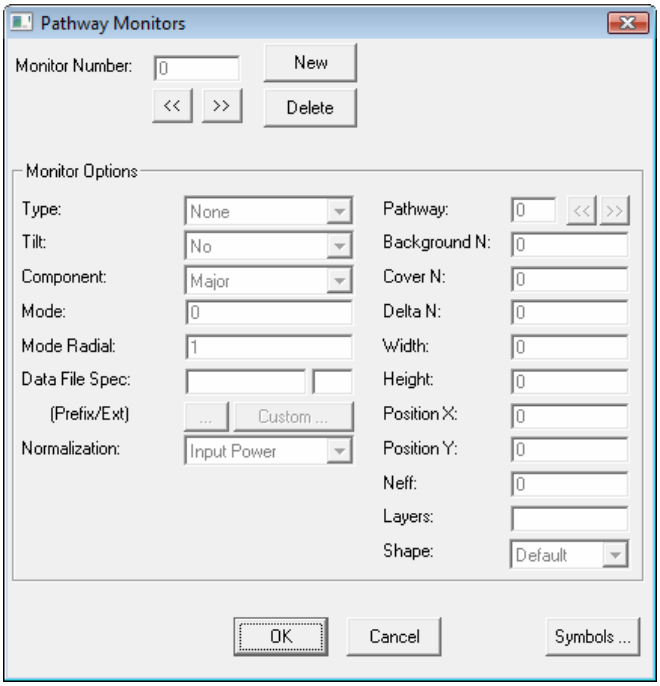

Figure 5-1: The Pathway Monitors dialog which allows various quantities to be measured along optical pathways during a simulation.

Assigning the Pathway Monitor to a Pathway

A pathway monitor can be assigned to a pathway via the **Pathway** option in the properties dialog box. If the properties dialog was entered from **Pathway Edit Mode** the pathway being edited will be used by default, otherwise it is set to 1.

Editing a Pathway Monitor

Once a monitor has been defined, it can be changed by opening the monitor properties dialog and selecting the desired monitor via the **<<** and **>>** buttons. Once the desired monitor has been selected, perform the required changes and then click **OK** to return to the CAD. To delete a pathway monitor, select the monitor using the **<<** and **>>** buttons and click **Delete**.

4.C. Choosing a Pathway Monitor Type

The **Monitor Type** option is the main control for the type of data measured and saved by a pathway monitor. The available choices are:

- *File Power* and *File Phase*

  These options compute the overlap integral between the simulated field at the current Z position and a test field which is specified by a data file and are used to measure the power in or phase of the field profile specified in the data file. The data file to be used for the overlap should be specified in the **Data File Spec** fields described in Section 4.E. See notes on phase monitors in Section 4.D.

- *Slab Mode Power*, *Fiber Mode Power*, *Slab Mode Phase*, and *Fiber Mode Phase*

  These options compute the overlap integral between the simulated field at the current Z position and a test field which is an analytically computed slab/fiber mode of the local position. These pathway monitor types measure the power in or phase of the specified mode. This is the default monitor type. The slab mode options are available for 2D and the fiber mode options are available in 3D. See notes on phase monitors in Section 4.D.

  The characteristics of the analytical slab/fiber mode are, by default, set via the pathway at the current Z position but can be set directly via the settings 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 in the **Mode** and **Mode Radial** options described below to yields an LP_{mn} fiber mode.

- *Gaussian Power* and *Gaussian Phase*

  These options compute the overlap integral between the simulated field at the current Z position and a test field which has a Gaussian profile. See notes on phase monitors in Section 4.D.

  The characteristics of the Gaussian field are, by default, set via the pathway at the current Z position but can be set directly via the settings described in the next section.

- *Launch Power* and *Launch Phase*

  These options compute the overlap integral between the simulated field at the current Z position and a test field equal to the launch field. Before calculated the overlap, note that the input field is translated to align with the center of the pathway. See notes on phase monitors in Section 4.D.

- *Partial Power*

  This option computes the power in the simulated field via a power integral at the current Z position. In 2D, the integral is performed over range +/- ($w/2 + f/\gamma$), where w is the width of the pathway, γ is a characteristic transverse decay constant for the waveguide, and f is a fractional parameter set by the internal variable `monitor_decay_fraction` which defaults to 0. The characteristic decay length is determined from the highest supported slab waveguide mode, but is appropriately limited if that mode is nearly cutoff. In 3D, the same type of calculation is performed using the height of the pathway.

  Additionally, in 3D, the shape of the integration area depends in the types of components that make up the pathway. If the pathway is composed of fiber components, the integration will be over the elliptical (circular if the width and height are equal) cross-section. Otherwise, the integration will be over a rectangular cross section.

In either case the width/height and shape of the integration area can be set manually by the settings discussed in the next section.

The Monitors dialog also has a field for **Shape**, in which you can select *Default*, *Rectangular*, or *Elliptical*. (*Default* is *Rectangular* or *Elliptical* depending on the structure type as noted above).

- *Total Power*

  This option computes the total power in the simulation domain at the current Z position. This type of pathway monitor does not require the use of a pathway.

- *Effective Index*

  This option should only be used within the context of an iterative mode calculation to visualize the convergence process of the effective index. This is an advanced feature which should only be used by experienced users.

- *Field 1/e Width* and *Field 1/e Height*

  These options compute the 1/e width and height of the simulated optical field at the current Z position. These pathway monitor types are independent of the pathway.

4.D. Note for Phase Monitors

The value returned by the phase monitor is the phase of the slowly varying field $u(x,y,z)$ described in Section 2.B and is given in degrees from 0 to 360. In order to derive the phase of the physical field $\phi(x,y,z)$, use the following relation:

$$\phi(x,y,z) = u(x,y,z)\, e^{i\bar{k}z}$$

The phase of the physical field is therefore equal to

$$\text{"Monitor Result"} \; + \; \bar{k}\,\frac{180z}{\pi}$$

since the monitor result is in degrees.

4.E. Monitor Options

Once a **Monitor Type** has been chosen, the relevant monitor options can be set in the Monitor Properties dialog box.

Tilt

Tilts the monitor test field for an overlap monitor to the local component angle.

Component

Selects which field components are used in calculating the monitor result and is only relevant for full-vector calculations and/or bidirectional calculations. See Section 6.A and Section 6.C for more

information about these features. For a full-vector calculation, the choices *Major*, *Minor*, and *Both* correspond to the major, minor, and both major minor components respectively. For a bidirectional calculation, the same options can be used as well as the additional options *Major – Backward*, *Minor – Backward*, and *Both – Backward* which refer to the corresponding component of the backward field.

Mode

Sets the integer mode number of the analytically computed test field for a slab or fiber mode monitor type. For a slab mode monitor, a value of `0` represents the fundamental mode and is the default. For a fiber mode, 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. A value of `*` for both a slab and fiber mode indicates that all supported modes be included in the power calculation.

Mode Radial

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

Data File Spec

These options set the data file(s) to be used for a **Monitor 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 monitor data 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 should 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**, **Polarization**, and **Vector Mode**, 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. See Section 6.A for details. Generally, all necessary components should b_D2HLink_262502e 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**, **Polarization**, and **Vector Mode**. Generally users will only use the major component unless **Vector Mode** has been set to *Full*. See Section 6.A for a complete definition.

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 used for 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 monitor output is normalized. The default choice is *Input Power* which normalizes power monitors to the total power in the launch field. The other choice *Local Power* which normalizes power monitors to the total power in the simulated field at the current Z position. This can be useful to obtain the fraction of the field inside a particular where the total power may be decaying or has otherwise changed from the input power. An example would be to obtain the local confinement factor of a tapered structure.

Background N

This option sets the background index for the slab/fiber mode to be used as a test field. 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 used as a test field. 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 used as a test field. The default value corresponds to the index difference of the pathway at a particular Z position.

Width and **Height**

These options set the width (measured along X) and height (measured along Y) of the analytically computed slab or fiber mode to be used as a test field for a slab or fiber mode monitor. For a partial power monitor, this option sets the integration area along X or Y. The default value corresponds to the width of the pathway at a particular Z position.

Position Y

This option sets the vertical position of the test field for a monitor. The default is at the center of the local pathway structure and is specified by the keyword `default`. Note that the X position of a test field for a monitor is set equal to the center of the local pathway structure and cannot be set directly.

However, it is usually possible to achieve a desired effect through the use of a pathway defined by inactive components along the desired path to be monitored.

Neff

This option sets the effective index corresponding to the propagation constant of a test field for a monitor. For slab or fiber monitors the default value corresponds to the actual propagation constant. For a file monitor the default value is set to the effective index stored in the file. In all other cases, the default is the background index which should be changed appropriately.

Layers

This option can only be used with multilayer structures and with partial power monitors. It specifies a range of layers to be included in the power calculation as either a single layer number or a starting layer number, a hyphen, and an ending layer number (e.g. '`2-5`'). This option, in conjunction with **Normalization**, is useful for determining confinement factors.

Shape

This option selects the 3D shape of the integration area for a partial power monitor. Its default behavior is *Rectangular* for channel structure types, and *Elliptical* for fiber structure types.

5

Mode Solving

BeamPROP includes two fully functional mode solvers that are based on BPM: the iterative method and the correlation method. The mode solvers can be to solve for the propagating modes of a structure with an arbitrary 2D or 3D index cross section. These calculated modes can then be used to determine dispersion relations or used as launch fields for propagation analysis.

The technical background for both of these mode solvers can be found in Section 2.H and is not repeated here.

The RSoft CAD also has additional mode solvers that can also be used including FemSIM (licensed separately) and TmmSIM. Please see the CAD manual for more information on these methods.

5.A. Using the Iterative Method

This section describes how to obtain modal characteristics using the iterative method as described in Section 2.H. The iterative method is a generally significantly faster approach for obtaining the field shape and propagation constant for the fundamental mode as well as higher order modes than the correlation method. The computations are performed using the so-called imaginary distance beam propagation method and is therefore not valid for structures with an imaginary index or for leaky or lossy modes.

To perform the calculation click the **Compute Modes** button in the left CAD toolbar to open the Mode Calculation parameters dialog. To use the iterative method, open the Mode Calculation Options dialog described in Section 5.C and make sure **Method** is set to *Iterative* and any other relevant options have been set appropriately.

5.A.1. Preparing to Find Modes

The structure must be created in the RSoft CAD before a mode can be found. The structure should be uniform in Z and have the desired index profile. Be sure to use the **Display Material Profile** option to check that the profile is indeed correct.

The Launch Field

It is very important to choose a launch field which excites the modes you wish to find which means that in most cases the launch field must excite all the modes supported by a structure. One easy way to do this is to launch a simple Gaussian field off axis (a reasonable displacement is half the component width).

Simulation Domain and Grids

The simulation domain and grid for X, Y, and Z are treated differently.

X and Y Axes

The X and Y simulation domain and grid should generally be chosen such that the domain includes the entire structure, the domain is large enough so that the mode field will be sufficiently close to 0 at the boundaries, and the X and Y grid sizes need to be small enough to resolve the structure.

Z axis

The iterative method will automatically stop the simulation when a mode is found and so the length of the Z-independent structure can be made arbitrary provided it is long enough to converge. For most structures, a length of 1-10 mm should be sufficient. If it fails to converge before the end of the structure a warning is issued and the length should be increased. This warning can be disabled by setting the variable `idbpm_convergence_warning` to `0` although it will still be noted in the log file `log.txt`.
The Z step size used for an iterative mode calculation will be different than for the case of a standard BeamPROP propagation analysis It is recommended to use the maximum stable Z step size that yields the most rapid convergence. However, too large a step size can sometimes cause the iteration to converge to an undesired or unphysical mode. These modes are usually distinguished by unphysical discontinuities in the field pattern, best observed in scalar mode where all fields are continuous. If this occurs, reduce the step size by factors of two until convergence to the desired mode is obtained. Reducing the step size further will provide no accuracy benefit. Since the ideal step size for the iterative mode calculation is usually different from that for BPM propagation, the program maintains both settings for convenience.
The default step size is determined from a theoretical ideal and multiplied by a safety factor controlled by the variable `step_size_idbpm_factor` and has a default value of `0.5`. This factor should have a value less than 1. The advantage of using this method over simply setting the step size for the iterative method is that this automatic setting of the step size will scale properly as other variables such as the index change.

Convergence Tolerance

The default convergence criterion is that the normalized effective index change by less than 10^{-7} between iterations. The **Neff Tolerance** field in the Mode Calculation Options dialog controls this

parameter. Note that in some cases the effective index can oscillate and fail to converge. If this occurs, set the tolerance to a less stringent value, such as 10^{-6}. In other cases, the effective index will converge, but the mode pattern may not. If this appears to be the case, set the tolerance to a more stringent value, such as 10^{-8}.

5.A.2. Computing the Modes

Once all parameters have been set correctly, enter an **Output Prefix** in the Mode Calculation Parameters dialog to enable data to be saved and click **OK** start the simulation. Once the simulation has started the software will first attempt to find the fundamental mode. Once this mode has been found, and if additional modes are to be found, it will solve for each mode one at a time.

At the end of the calculation, the major component of the fundamental mode will be displayed in the simulation window. Other field components and modes can be viewed using the controls in the top toolbar of the simulation window.

5.A.3. Understanding Mode Output

If an **Output Prefix** has been set, any mode files found will be saved to the current working directory. Each mode found will produce two files for each field component: a data file `<prefix>_<comp>.m##` and a corresponding WinPLOT command file `<prefix>_<comp>.p##` where `##` is the mode number, `<comp>` is the field component, and `<prefix>` is the **Output Prefix**. The exact components saved depend on the **Model Dimension**, **Polarization**, and **Vector Mode**; see Section 6.A for details.

The format of the mode data file is discussed further in Appendix B. The effective index of the modes can be found in the file `<prefix>.nef` file. See Appendix B for the format of this file.

5.B. Using the Correlation Method

This section describes how to obtain modal characteristics using the correlation method as described in Section 2.H. The correlation method uses two full BeamPROP propagations to solve for the modes of a structure. In the first propagation, the mode spectrum is found by taking the Fourier transform of a correlation function. From this spectrum the eigenvalues, or neff values, for each mode can be found. In the second simulation, the mode profiles, or eigenfunctions, are found.

To perform the calculation click the **Compute Modes** button in the left CAD toolbar to open the Mode Calculation parameters dialog. To use the correlation method, open the Mode Calculation Options dialog described in Section 5.C and make sure **Method** is set to *Correlation* and any other relevant options have been set appropriately.

5.B.1. Preparing to Find Modes

The structure must be created in the RSoft CAD before a mode can be found. The structure should be uniform in Z and have the desired index profile. Be sure to use the **Display Material Profile** option to check that the profile is indeed correct.

The Launch Field

It is very important to choose a launch field which excites the modes you wish to find which means that in most cases the launch field must excite all the modes supported by a structure. One easy way to do

this is to launch a simple Gaussian field off axis (a reasonable displacement is half the component width).

Simulation Domain and Grids

The simulation domain and grid for X, Y, and Z are treated differently.

X and Y Axes

The X and Y simulation domain and grid should generally be chosen such that the domain includes the entire structure, the domain is large enough so that the mode field will be sufficiently close to 0 at the boundaries, and the X and Y grid sizes need to be small enough to resolve the structure.

Z axis

Since the effective indices are computed from a mode spectrum which is the result of a FFT, standard Fourier sampling principles must be followed and the Z domain and step size must be chosen carefully. The propagation length will determine the resolution within the mode spectrum: a longer propagation length will result in better resolution of mode peaks. The spectral range will be a determined by the Z **Monitor Step**. Therefore a find Z Step Size can be used if needed without increasing the memory and time requirements for the FFT. This is important since the bound eigenmodes are limited in bandwidth (*i.e.* $n_{min}<n_{eff}<n_{max}$), thus there is no need for the frequency range to be infinitely large.

5.B.2. Computing the Mode Spectrum

To compute only the mode spectrum, select the appropriate option in the Mode Calculation Options dialog.

This step is automatically performed if calculating the mode profiles as discussed in the next section.

Open the Mode Calculation parameters dialog and once all parameters have been set correctly, enter an **Output Prefix** to enable data to be saved and click **OK** start the simulation. Once the simulation has started the software will perform a full propagation and display the computed mode spectrum.
The mode spectrum represents the FFT of the correlation function described in Section 2.H and has one or more peaks at the frequencies of the guided modes, each with a height equal to the fraction of power in the input field contained in that mode. The frequency axis in this graph is defined by the mode effective index minus the reference index. The reference index is related to the reference wavenumber (k-bar) described in Section 2.B. The mode spectrum files are discussed in Section 5.B.4. While the search algorithm limits the mode peaks to the frequency range specified by the **Neff Min** and **Neff Max** values, care must be taken to experiment with the propagation length and numerical parameters to ensure that spurious modes are not determined. The search algorithm enumerates peaks above a certain fraction of the maximum peak height. This fraction defaults to 0.01, and can be controlled by the symbol table variable `mode_threshold`.

5.B.3. Computing the Mode Profiles and Propagation Constants

To compute the mode profiles and propagation constants with the correlation method, make sure the **Calculate Spectrum Only** option in the Mode Calculation Options dialog is not selected.

Open the Mode Calculation parameters dialog and once all parameters have been set correctly, enter an **Output Prefix** to enable data to be saved and click **OK** start the simulation. BeamPROP will first compute the mode spectrum as described in the previous section. At the end of this calculation, the mode spectrum will briefly flash on the screen and then a second propagation will start. At the end of the second propagation the major component of the fundamental mode will be displayed in the simulation window. Other field components and modes can be viewed using the controls in the top toolbar of the simulation window.

The Imaginary Propagation Constant

The program will save the imaginary part of the effective index which represents the propagation loss of the mode. See Section 1.B for a description of this parameter.

Corrections Used To Calculate Effective Indices

As discussed in Section 2.H two corrections can be applied for more accurate effective index values. The first correction accounts for the fact that the paraxial equation was solved and not the exact Helmholtz equation. The second correction uses the mode profiles to calculate the propagation constants by substituting the mode profiles into the wave equation. This method not only results in a more accurate effective index, but also allows the imaginary part of the propagation constant to be found. This method is controlled by the variable `neff_method`, and is enabled by default. Note that disabling this variable will result in no imaginary refractive indices.

The effective index values displayed in any output files such as the `.nef` file or any mode files (`.m##`) or plots (`.p##`) uses the corrected value while the mode spectrum will display the uncorrected value. If needed, the three real propagation constant values (uncorrected, first correction only, first and second correction), and the imaginary value can be found in the `.mds` file.

5.B.4. Understanding Mode Output

If an **Output Prefix** has been set, any mode files found will be saved to the current working directory. Each mode found will produce two files for each field component: a data file `<prefix>_<comp>.m##` and a corresponding WinPLOT command file `<prefix>_<comp>.p##` where `##` is the mode number, `<comp>` is the field component, and `<prefix>` is the **Output Prefix**. The exact components saved depend on the **Model Dimension**, **Polarization**, and **Vector Mode**; see Section 6.A for details.

The format of the mode data file is discussed further in Appendix B. The effective index of the modes can be found in the file `<prefix>.nef` file. See Appendix B for the format of this file.

Mode Spectrum Output

If an **Output Prefix** has been set, several files for the mode spectrum will be saved. The file `<prefix>.pft` and `<prefix>.fft` are the WinPLOT command file and data file for the mode spectrum respectively. The output of the file `<prefix>.cor`, which contains the correlation function, can be enabled via the Output Options dialog.

5.C. Setting Mode Calculation Options

The BeamPROP mode solvers have several options that can be set in the Mode Calculation Options dialog shown in Fig. 6-1. This dialog can be opened by clicking the **Mode Options…** button in the Mode Calculation parameters dialog.

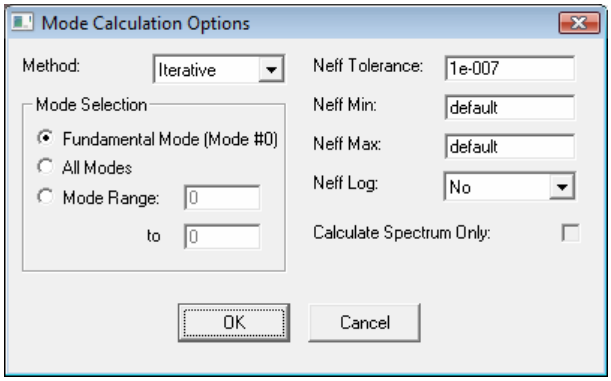

Figure 6-1: The Mode Calculation Options dialog box.

These options are:

Method

This option sets he mode solver to be used for mode calculations.

Mode Selection

These options allow the user to choose the modes to solve for. By default, only the fundamental mode will be found.

Neff Tolerance

This option sets the tolerance for the Neff convergence during an iterative mode calculation.

Neff Min and Neff Max

These options control the range of effective indices that will be found by the mode solver. The default values correspond to the background index and maximum index of the structure. In some cases, this default is not valid and should be changed as necessary.

Neff Log

This option enables the output of the effective index values during an iterative mode calculation and can be useful to study the convergence.

Calculate Spectrum Only

This option is only available when using the correlation mode solving method. When used, it will output only the mode spectrum and will not compute the associated mode profiles.

5.D. Additional Comments

This section contains additional comments about the mode solvers included with BeamPROP.

5.D.1. Finding Asymmetric Higher Order Modes with the Iterative Method

There is an alternative method to compute the first order asymmetric mode of a structure symmetric about x=0. This mode has a null at x=0 and the field for x>=0 can be considered the fundamental mode of the structure in the corresponding half plane which satisfies Dirichlet boundary conditions at the left boundary. We can obtain the desired mode by restrict the domain of the computation to x>=0 by setting the X **Domain Min** field in the simulation parameters dialog to 0 and setting the boundary condition at the left boundary to Dirichlet by setting the variable `bc_type_left` to `BC_DIRICHLET`.

5.D.2. Calculating Modes from the Command Line

The mode solving features can be accessed from the command line using the symbol table variables `mode_method`, `mode_spectrum` and `mode_set`. The `mode_method` variable sets the mode solver to be used: a value of `0` corresponds to the correlation method and `1` corresponds to the iterative method. The `mode_spectrum` variable should be set to `1` to compute the mode spectrum only. The `mode_set` variable can be a single integer to select one mode, a range specification (*low-high*) to select several modes, or an asterisk (`*`) to select all modes.

5.D.3. Setting the Mode Calculation Length

By default, the mode computation will be performed using the Z domain settings, or for the iterative method until a converged result is found. This behavior can be changed via the variable `mode_length`. When this variable is defined, any mode computation will start at the Z domain minimum and continue for a length equal to `mode_length`, or for the iterative method only, until a converged result is found. This option does not require the structure to be defined over the entire length since the program will automatically extend the structure defined at the Z domain minimum. This variable can have several special values:

Value	Description
undefined or `-1`	The values given in **Domain Min** and **Domain Max** determine the length.
`default`	The program will automatically determine a length. For the iterative method, the default is 1000 propagation steps; for the correlation method, a length is chosen to satisfy a tolerance (`1e-4`). The variable `neff_tol_cor` can be used to set this tolerance. Note that this is only one factor determining final accuracy - step size and grid size must be considered by the user as well.
any other value	This sets the mode calculation length.

6

Advanced Features

BeamPROP has several advanced capabilities which extend the basic 2D and 3D BPM 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. Incorporating Polarization Effects (Vector BPM)

As noted in Chapter 2 and elsewhere, BeamPROP has the capability of incorporating polarization effects using vector beam propagation techniques. Here we describe this capability by first noting the relevant simulation options, discussing their effect on 2D and 3D simulations, and then outlining the procedure to modify the polarization to achieve a desired specific polarization type.

6.A.1. Vector BPM Options

The options which control the vector simulation capability of BeamPROP are primarily located in the BeamPROP Simulation Parameters dialog; some features are set elsewhere.

Primary Settings

The primary options controlling the vector simulation capability are located in the BeamPROP Simulation Parameters dialog as shown in Fig. 6-1. These options are discussed in this section.

Figure 6-1: The portions of the BeamPROP Simulation Parameters dialog where the vector BPM options are set.

The relevant fields are as follows:

Vector Mode
This radio button field has three options: *None*, *Semi*, and *Full*. When this field is set to *None*, BeamPROP uses scalar BPM as described in Chapter 2. The *Semi* and *Full* options correspond to the so-called semi-vectorial and full-vectorial methods commonly noted in the literature (see, for example, Ch. 2, Ref. [32]). The precise meaning of these options is described in the following sections.

Polarization
This option has two possible settings: *TE* or *TM*. These describe the polarization of the optical field or predominant polarization in the case of full-vector where the fields are hybrid. The precise meaning of these options is described in the following sections. Depending on the **Vector Mode** and other settings the **Polarization** options may be grayed out as appropriate if they are not applicable or available.

Other Settings
In addition to the polarization options in the BeamPROP Simulation Parameters dialog, BeamPROP has several options which control advanced aspects of the methods used for vector BPM.

- A BPM formulation based on the magnetic field can be used rather than the default electric field. This option is enabled through the **Formulation** option in the Advanced Parameters dialog. See Section 6.B for more details.
- Anisotropic calculations can be enabled via a checkbox in the Advanced Parameters dialog. See Section 6.B for details.
- The **Vector Conserve** option can help conserve power in some cases. See Section 6.B for details.
- The polarization field can be manipulated to achieve a custom polarization for each launch field separately via the **Polarizer...** button in the Launch Parameters dialog. See Section 6.A.3.

6.A.2. Vector BPM – Field Components Simulated
The field components included in and output from a simulation vary depending on the vector BPM settings.

2D Vector BPM
In 2D, the index varies along X, the structure is considered infinite and uniform along Y, and the primary propagation direction is along Z. These conventions, along with Maxwell's equations, imply that TE polarization is along Y and TM polarization is along X.
The nature of Maxwell's equations for a 2D geometry is such that a scalar calculation is equivalent to a TE calculation and there is no difference between semi-vectorial and full-vectorial calculations. Thus the vector modes are defined as:

- *None*

  A TE (E_x) calculation is performed independent of the **Polarization** Setting.

- *Semi* or *Full*

  The **Polarization** setting determines the polarization used in the beam propagation calculation. A setting of TE corresponds to E_y and a setting of TM corresponds to E_x.

3D Vector BPM

In 3D, the index varies along both X and Y and the primary propagation direction is along Z. These conventions, along with Maxwell's equations, imply that quasi-TE polarization is along X and quasi-TM polarization is along Y.

The nature of Maxwell's equations for 3D geometry is such that a scalar calculation is not equivalent to either polarization. However, in the case of a Fiber, it is equivalent to the so-called LP modes. Additionally, while a pure TE or TM wave does not exist, most structures quasi-TE and quasi-TM waves exist in which the most of the field energy is in one polarization (the major component) and only a small fraction (less than one percent) is in the other polarization (the minor component). Thus, the vector modes can be defined as follows:

- *None*

  A scalar calculation is performed which is an accurate approximation for a small index difference.

- *Semi*

  The **Polarization** setting determines the polarization used in the calculation. A setting of TE corresponds to E_x and a setting of TM corresponds to E_y. The minor component of the field is neglected (set to zero) and polarization is partially taken into account by employing the proper field equations for the major component. The corresponding H fields are output, and, if files are present, will be used for the launch field or monitor calculation. If not present, the H field will be calculated from the E field though pre-computed H fields are generally better.

  This approximation is very accurate in many cases since, as noted above, the minor component of the field is frequently negligible. This option can discriminate between quasi-TE and quasi-TM fields and can be used to determine differences in field shapes and propagation constants.

- *Full*

  The **Polarization** setting determines the polarization of the major component used in the beam propagation calculation. A setting of TE corresponds to E_x and a setting of TM corresponds to E_y. Both components of the field, and the coupling between them, are taken into account in the simulation. The corresponding H fields are output, and, if files are present, will be used for the launch field or monitor calculation. If not present, the H field will be calculated from the E field though pre-computed H fields are generally better.

  Generally this option is only needed the amplitude of the minor field component is needed or if there is coupling between polarizations. This option should not be used if not needed since it is significantly slower than the other options. It is also not unconditionally stable with respect to grid size and the **Scheme Parameter** may be needed to ensure stability. See Section 6.B.

6.A.3. Manipulating the Polarization of the Launch Field

These options can only be used when **Vector Mode** is set to *Full*.

Once the **Model Dimension** and **Polarization** have been set 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 can be set with 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

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

  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

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

  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 $\Delta\phi$ 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 which are both measured in degrees. The desired phase difference $\Delta\phi$ 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.

6.A.5. Vector BPM and the Effective Index Calculation Option

When performing an Effective Index Method (EIM) calculation as described in Section 6.G, the Vector Mode is interpreted as described for 2D Vector BPM. In addition to its meaning with regard to vector beam propagation, the **Polarization** setting is used in determining the effective indices used in the

calculation. Thus even if the **Vector Mode** field is set to *None*, polarization is partially accounted for through the effective indices.

6.B. Advanced Numerical Options

Additional options which control the numerical simulation and affect its accuracy and reliability are found in the Advanced Parameters dialog, shown in Fig. 6-2, which can be opened via the **Advanced…** button in the simulation parameters dialog.

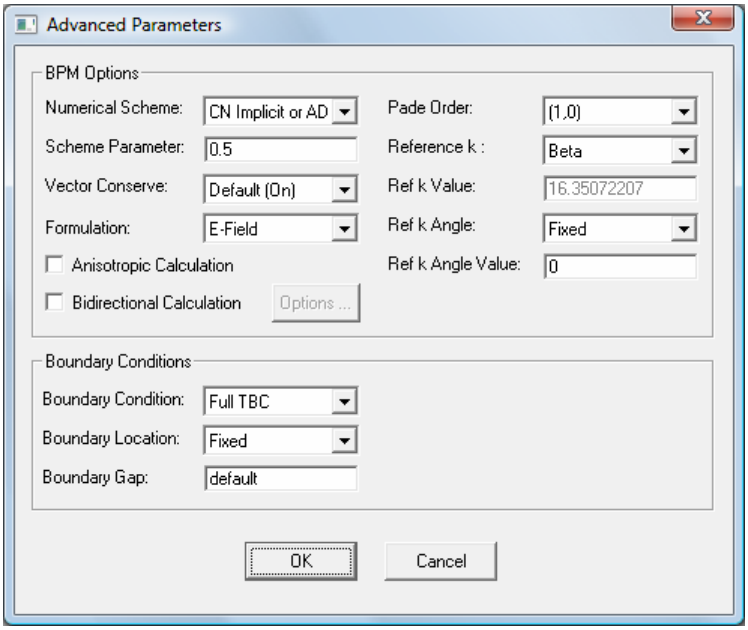

Figure 6-2: The Advanced Parameters dialog.

The options in this dialog can be broken into three groups: Algorithm Options, Wide-Angle Options, and Boundary Condition options.

6.B.1. BPM Options

The BPM options are:

Numerical Scheme

This option sets the numerical scheme that is used in the calculation as described in Chapter 2. The choices are:

- *CN Implicit or ADI*

  This is either the fully implicit Crank-Nicholson scheme for 2D, or the Alternating Direction Implicit (ADI) scheme for 3D (see Ch.2, Ref. [1]). Both schemes are unconditionally stable and are recommended for all problems. This choice is the default.

- *GD (Gen. Douglas)*

This is the Generalized Douglas (GD) scheme which is an alternative to the standard Crank-Nicholson (CN) scheme. The GD scheme can, for some problems, be more accurate than the CN scheme, but is somewhat slower per step. Alternatively, the same accuracy can sometimes be achieved with coarser transverse grid size, leading to greater overall speed. Currently, this scheme is implemented only in 2D.

- *FFT*

  This is the FFT-based BPM method. While generally inferior to the above finite-difference schemes, the FFT method has the advantage that for large regions that are uniform the method can propagate the field in one Z step (provided also that the field does not significantly impinge on the boundary since transparent boundary conditions can not be employed). This method is most useful in conjunction with the simulation region feature discussed in Section 6.D. This method is not recommended except for special circumstances as in the example referred to above.

Scheme Parameter

This sets the scheme parameter for semi/full-vector calculations. In general it should be left at the default value of `0.5` except when needed to ensure stability. Valid values for this parameter are between `0.5` and `1.0`. See Section 6.A for further details on full-vector settings. Refer to the guidelines in Section 3.G for additional comments on use of the scheme parameter. Please contact RSoft with any questions about its use.

Vector Conserve

The basic vector BPM formulation described in Chapter 2 can have difficulty conserving power in some cases such as 2D TM or TE/TM for semi- or full-vector calculations. This option offers an improved algorithm formulation which can in many cases remedy the difficulties with power conservation, at a moderate cost in efficiency. This option is enabled by default, and can be also controlled by setting the variable `vector_conserve` to 0 or 1. This option is not currently enabled for full-vector or nonlinear calculations.

Formulation

This option sets whether the algorithm is based on the electric field or magnetic field. While using the magnetic field formulation can sometimes offers better performance with respect to convergence and boundary conditions, not all capabilities of the program are enabled in this mode. This option should be used by advanced users only.

Anisotropic Calculation

This option enables an anisotropic calculation to be performed. It requires that the **Vector Mode** be set to *Full*. Also, one or more materials with anisotropic properties must be defined and used in the structure. Note that only the diagonal and XY and YX elements of the index will be used since BeamPROP does not compute the E_z field.

Bidirectional Calculation

This option enables the use of the bidirectional feature in BeamPROP and is discussed in Section 6.C.

6.B.2. Wide-Angle Options

Pade Order

This option selects the order of the Padé approximation that is used in the calculation as described in Section 2.E. Higher order approximations increase the accuracy of the results for large angles or large refractive index differences but also increase the computation time in proportion to the first number in the Padé order. A choice of *(1,0)* is equivalent to the standard paraxial approximation while a choice of *(1,1)* generally produces better results than the *(1,0)* case at a negligible increase in computation time. Choices of *(2,2)*, *(3,3)*, and *(4,4)* offer more accuracy at the expense of computation time.
In 3D, the wide-angle feature is generally not currently available. However for scalar 3D calculations an approximate version of the *(1,1)* option, valid for low-index contrast, has been implemented. In some test cases, this option has produced marked improvement over the (1,0) option, but should be used with caution since it can sometimes lead to less accurate results.

Only Padé (1,0) should be used when doing any form of mode calculation.

Reference k

The reference k is the characteristic wavenumber (k-bar) used by BPM as described in Section 2.B. It is critical to choose a reference k value that corresponds to the actual effective index of the simulation. See Section 3.G for details.
This option has several choices. A choice of *k External* corresponds to a k-bar value of $k_0 n_0$, a choice of *k Average* corresponds to a value of $k_0 \langle n \rangle$, and a choice of *Beta* corresponds to an estimate of the propagation constant for the fundamental mode of the structure where k_0 is the k-vector in free space, n_0 is the background index, and $\langle n \rangle$ is the average index in the structure. A choice of Other allows a custom value to be set by the **Ref k Value** option.

Ref k Value

This option displays the current value of the **Reference k** to be used for the simulation. A custom value can be entered here if **Reference k** is set to *Other*.

Ref k Angle and **Ref k Angle Value**

This option corrects the **Reference k** (k-bar) value for propagation at an angle. The k-bar value is multiplied by the cosine of the propagation angle. There are two options:

- *Fixed*

  The reference angle is fixed throughout the calculation and is determined by the value in the **Ref k Angle Value** field which must be non-negative number measured in degrees from the Z axis.

- *Variable*

  The reference angle is allowed to vary throughout the calculation and is determined automatically by a weighted average of the local waveguide/fiber angles present in the circuit. Alternatively the user can control the local reference angle by defining an optical pathway and setting the `kbar_path` variable in the symbol table to be to desired pathway number.

6.B.3. Boundary Conditions

These options control the boundary conditions to be used during a simulation.

Boundary Condition

This option selects the type of boundary condition employed in the calculation. The choices are:

- *Dirichlet*

  The field is required to be zero on the boundary. This effectively reflects radiation back into the computational domain.

- *Simple TBC*

  This is the simple implementation of the transparent boundary condition as described in Ch. 2, Ref. [2]. It effectively lets radiation pass through the boundary and leave the computational domain. This choice is recommended for highly multimode problems.

- *Full TBC*

  This is the full implementation of the transparent boundary condition as described in Ch.2 Ref. [2]. It effectively lets radiation pass through the boundary and leave the computational domain. This choice is the default. However, in problems with a large amount of radiation this option can sometimes behave poorly, and the *Simple TBC* can perform better. For example, the *Simple TBC* is recommended for highly multimode problems.

Boundary Location

This option determines the boundary of the computational domain along the X axis. The options are:

- *Fixed*

  The left and right boundary locations are fixed throughout the calculation, and are determined by the **Domain Min** and **Domain Max** fields described above. The default values for these fields are determined by the extent of the waveguide circuit and the **Boundary Gap** option described below.

- *Variable*

  The left and right boundary locations are allowed to vary throughout the calculation, and are determined by the local extent of the waveguide circuit and the **Boundary Gap** field described below. Alternatively, the user can control the local left and right boundary locations by defining optical pathways and setting the `boundary_path` or `boundary_path_left` and `boundary_path_right` variables in the symbol table. This option is only implemented in 2D.

Boundary Gap

This option sets the gap between the extent of the waveguide circuit and the recommended computational boundary. The default value is estimated by the program and is specified by the keyword `default`.

Other Boundary Types

By default, the boundary condition selected by this option is applied to all boundaries. It can be useful, however, to apply different conditions at different boundaries. BeamPROP has this capability, and it is

accessed through the symbol table variables `bc_type_left`, `bc_type_right`, `bc_type_bottom`, and `bc_type_top`. By setting one or more of these variables to one of the keywords `BC_DIRICHLET`, `BC_STBC`, or `BC_FTBC`, the user can select the condition to be applied to the indicated boundary.

6.C. Bidirectional Calculation Options

The options which control a bidirectional BeamPROP simulation are set in the Bidirectional Calculation Options dialog, shown in Fig. 6-3, which can be opened via the **Options…** button in the Advanced Parameters dialog. The **Bidirectional Calculation** must be enabled to access this feature.

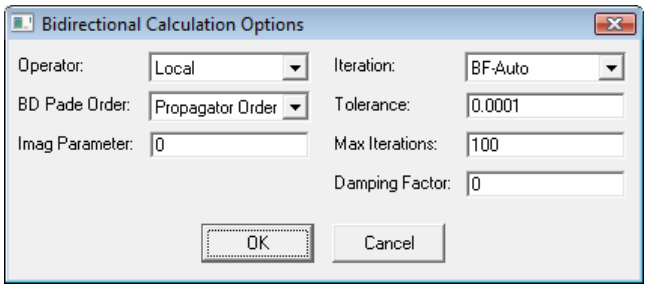

Figure 6-3: The Bidirectional Calculation Options dialog.

BeamPROP's bidirectional calculation feature one of the most complex aspects of the program. Contact RSoft with any questions.

6.C.1. Background

As noted in Chapter 2, BPM techniques generally do not account for backward reflections since the one-way wave equation on which they are based does not admit both positive and negative traveling waves. BeamPROP, however, has a bidirectional BPM algorithm which considers coupled forward and backward traveling waves and can account for reflection phenomena such as resonant effects as found in grating structures.

The Algorithm

The bidirectional algorithm propagates both the forward and backward field through the structure. When an interface is encountered, these fields couple through a differential operator. While the forward field at the input plane is known, the backward field at the input plane must be determined so that the backward field at the output is 0. Satisfying this condition requires an iterative procedure. The interface operator and the iteration procedure are the key elements of the algorithm which must be selected and controlled.

Important Notes

Before discussing the use of the bidirectional feature, it is important to mention several caveats.

- Since this technique is at the forefront of research in this area, its limits have not been established. Thus while many grating structures can and have been modeled, there are potential difficulties with

larger index differences and/or longer lengths, and the technique may or may not fail on any given new problem.

- The technique is not as robust as the usual BPM, and requires somewhat more expertise to use properly.
- While this feature does have its difficulties on some problems, remarkable success on other practical problems warrants its inclusion in BeamPROP since it allows the user to accomplish calculations which would otherwise be possible.
- Normally all BeamPROP simulation results represent the electric field, *E*. However, in the case of bidirectional simulations with TM polarization, the software automatically switches to a mode in which the field being simulated and displayed is the magnetic field, *H*.
- Currently the bidirectional feature is only implemented for 2D geometries.

6.C.2. Creating a Bidirectional BPM Simulation

When using the bidirectional BPM feature, first create the desired structure as usual. If you are creating a grating structure, it is usually easier to use a user-defined taper to create such periodic structures using the built-in `sin(x)` or `step(x)` functions. See the CAD manual for more details.

Once the structure has been created it is recommended to simulate with standard BPM in order to determine reasonable numerical parameters before trying bidirectional BPM. One of the restrictions of the current bidirectional implementation is that transparent boundary conditions are not available; therefore it is important at this stage to set the domain large enough so that the amount of radiation hitting the boundary is small.

To perform a bidirectional simulation, first enable the bidirectional feature by selecting the **Bidirectional Calculation** option in the Advanced Parameters dialog and then run the simulation. For the simplest problems involving small index differences, this is sufficient. More complex problems, however, require setting advanced options in the Bidirectional Calculation Options dialog discussed in the next section.

An Example

Open the `grating.ind` example file, run the simulation, and note the following:

- The **Display Mode** is set to *Slices*, which is the only display available with the bidirectional feature.
- The display contains both forward and backward fields displayed in blue and green respectively.
- The calculation proceeds forward and backward one or more times, reflecting the iterative nature of the technique. Under some conditions, the display may pause for long periods during internal iterations.
- The monitor has been set to display the power in the reflected field by setting the **Monitor Component** option in the monitor properties dialog *Major – Backward*. See Chapter 4.

6.C.3. Bidirectional Parameters

This section documents the option sin the Bidirectional Calculation Options dialog shown in Fig. 7-2.

Operator
This option sets the operator which is used at longitudinal interfaces to couple the forward and backward fields. The default choice is *Local*, which corresponds to an approximation valid for low index contrast structures. The second choice is *Padé*, which uses a Padé technique to achieve a better approximation appropriate to higher index contrast cases. In this case it is important to carefully set the **BD Pade Order** and **Imag Parameter** options as well as the **Pade Order** option Advanced Parameters dialog.

BD Pade Order
This option sets the order of the Padé-based interface operator and is analogous to the **Pade Order** field in the Advanced Parameters dialog. The default choice is *Propagator Order*, which indicates that this option will be set equal to the **Pade Order** setting. The other choices are the same as those described in Section 6.B.2. Generally, for high contrast cases, the Padé technique must be used for both the interface and the propagation but the orders need not be the same. The order required for the propagation is that which is necessary to accurately propagate the phase of the wave between interfaces in order to accurately model resonant reflection effects in a grating for example. The order required for the interface operator is primarily determined by the need to deal with a wide range of k-vectors in the complex plane associated with the **Imag Parameter** field described below.

Imag Parameter
This option sets a parameter associated with controlling the stability of the Padé-based interface operator. When using the Padé operator, artificial evanescent waves are generated during the calculation which can lead to instability. True evanescent waves, however, are also generated which are not problematic. To circumvent this, an imaginary component is added to the reference wavenumber (k-bar) which effectively dampens the artificial evanescent waves. This field controls the stabilization parameter and must be between `0` and `1`. It is best to choose as small a stabilization parameter as possible, since a large value of the parameter affects the true waves in addition to damping the artificial waves. The latter difficulty is circumvented by employing a higher Padé order for the interface operator, which allows accurate computations with a large stabilization parameter. For a high index contrast problem, a typical imaginary parameter is `0.6-0.8`, and an appropriate Padé order is *(2,2)* or *(3,3)*.

Iteration
This option sets the iterative procedure used by bidirectional calculations. The choices are:

- *BF-Fixed*

  BF-Auto

  These choices refer to a proprietary iteration technique. Each option is controlled by a stabilization parameter (see the **Damping Factor** field below), which is held fixed for the *BF-Fixed* option and dynamically adjusted for the *BF-Auto* option. This option generally works best for simpler problems and is currently the default. For high index problems the *Matrix* option described below is recommended.

- *Matrix*

  Matrix-2

Matrix-3

These choices use standard iterative techniques for solving linear systems the details of which are proprietary. These options are generally more robust than the BF options above but can be slower on many problems. For high index problems, use of these options is highly recommended. *Matrix* or *Matrix-3* are generally best. *Matrix-3* is the most robust, but is sometimes slower than *Matrix*. *Matrix-2* is generally not recommended.

Tolerance
This option defines the convergence criterion for the iteration. The iteration stops when the normalized power in the backward field at the output of the circuit is less than the tolerance value set in this field (the backward power is normalized to the forward input power). The default value is `10e-4`. The iteration also terminates after a specified maximum number of iterations, defined in the **Max Iterations** field described below.

Max Iterations
This field limits the number of iterations to the specified value. The default limit is `100` iterations.

Damping Factor
This field defines a stabilization parameter related to the *BF-Fixed* and *BF-Auto* iteration options. The value should be between `0` and `1`, and the default of `0` indicates no damping. In the case of the *BF-Fixed* iteration option, this value defines the damping factor throughout the iteration. For the *BF-Auto* iteration option, this value defines the initial damping factor, and the damping factor is adjusted dynamically during the simulation as required. If the default of `0` yields difficulties, a good alternative starting value is `0.5`.

6.D. Using Simulation Regions

The BPM algorithm gains much of its speed by leveraging the fact that for many photonic device simulations the variation of the structure along the propagation direction is small to moderate. By choosing an appropriate step size, reference wavenumber (k-bar), Padé order, etc. one can get accurate results in a reasonable length of time in such structures. Some structures can have large variations along the propagation direction. An example would be propagation from a waveguide into free space, through a lens, and back into another waveguide. This type of problem has two difficulties.

- There will be reflections at interfaces that must be taken into account. This might be amenable to bidirectional BPM; alternatively in many cases it can be accounted for approximately by a simple Fresnel correction.
- The large change in index means that each region along z has a different set of optimum simulation parameters.

To solve the problem, one can find a compromise for the best reference wavenumber, choose a higher Padé order, and choose a step size small enough to accommodate all regions. While this generally works, the simulation time also increases. A BeamPROP simulation region allows multiple regions along the propagation direction to be defined within which local control of step size and other simulation parameters is possible.

6.D.1. Defining a Simulation Region

To add a simulation region to a design, use the **Simulation Region** button in the left CAD toolbar. It can be drawn into the design similarly as a waveguide/fiber segment. A simulation region is defined only along Z and is therefore infinite along X and Y. Features such as logical attachment, referencing, etc. that are available for positioning waveguide segments are also available for simulation regions.

6.D.2. Creating Local Simulation Parameters

Once the region has been created it is possible to locally define some simulation parameters. Open the simulation region properties by right-clicking on the object in the CAD and click **Local Symbols…** to open up a special symbol table editor. This symbol table functions just like the normal, global symbol table, but variables defined in the local symbol table override those in the global table. The local symbol table will initially be empty. To locally set a parameter, such as the Z **Grid Size**, add the appropriate symbol as a new variable and give it the desired value. Currently, the following symbols may be set:

Variable	Definition
`step_size`	Z **Grid Size**
`kbar`	**Reference k**
`kbar_angle`	**Ref k Angle**
`pade_order`	**Pade Order** (0-4)
`simulated_bend`	Enable/disable Simulated bend (0/1)
`simulated_bend_radius`	Set **Simulated Bend Radius** (can be +/-)
`scheme`	**Simulation Method** (`SCHEME_CN`,…)

In addition to typical simulation parameters such as step size, control over other simulation aspects, such as the simulation method, is possible. For example, propagation through a long free space region can be done via the FFT-based BPM method by setting `scheme` = `SCHEME_FT` in the symbol table. A tutorial example describing the use of the simulation region feature is given in Tutorial 11.

6.E. Anisotropy, Non-Linearity, and Dispersion

BeamPROP supports the following material properties:

A detailed description of how these properties are defined can be found in Chapter 8 of the CAD manual.

- *Anisotropy*

  BeamPROP can incorporate anisotropic materials in order to simulation birefringent effects. Note that the **Vector Mode** must be set to *Full* as discussed in Section 6.A in order to perform an anisotropic simulation.

- *Non-Linearity*

  BeamPROP can incorporate χ^3 or n_2 nonlinearity on a single, monochromatic wave in order to address certain types of guided-wave nonlinear optics problems. Thus, devices such as nonlinear

directional couplers and spatial solitons, or temporal solitons via a change of variables, can be modeled. The CAD manual describes several methods of defining the non-linear parameters.

- *Dispersion*

  BeamPROP performs monochromatic simulations at single wavelengths. Thus, it cannot include dispersive effects within a single simulation. However, by defining the refractive index as a function of wavelength, a parameter scan over wavelength can be used to analyze dispersive materials. The CAD manual describes several methods to accomplish this.

6.F. Radial BPM

BeamPROP includes an option for Radial BPM which allows certain 3D problems with azimuthal symmetry to be simulated as 2D problems. This feature uses a radial version of the Crank-Nicholson scheme for 2D and the propagation is done in cylindrical (r,z) coordinates. Using this feature with problems with circular symmetry in both the structure and the launch field is equivalent to a full scalar 3D BPM, but significantly faster by several orders of magnitude. This feature can be enabled via the **Radial Calculation** option in the Global Settings dialog.

6.F.1 Using Radial BPM

Some simulation options are not available when using Radial BPM.

To use this feature create a 2D structure considering X=0 as the center of symmetry. The X axis will be interpreted as the radial coordinate R. Note that while the structure can be defined for all values of X, which is the optimal method, the problem will only be solved in the half space X>0. This will be automatically enforced by the program.
While it is implied that the structure is azimuthally symmetric, the field can have other implied azimuthal behavior which his controlled by the variable `azimuthal_mode`. 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 this parameter 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.

6.F.2. Displaying 3D Results

The azimuthal pattern is normally not displayed unless a result from a Radial calculation is launched into a 3D calculation. However, by setting **Output Radial Data in XY Format** option in the Output Options dialog, in addition to any 2D radial files that are generated, corresponding 3D results will be generated whose file prefix will be the original file prefix with "`_3d`" appended. Any azimuthal behavior that was implicit in the radial calculation is reflected in the 3D data file and graph.

This option is only for visualization purposes; it is not necessary ‘convert; radial field data can be launched into a 3D calculation automatically.

6.G. Effective Index Calculations

This feature is only supported for the Diffused and Rib/Ridge 3D structures.

The Effective Index Method (EIM) can be used to reduce the computation time and memory requirements of a simulation by converting a 3D structure into an approximate 2D structure. This option only affects the simulation; the definition of the original 3D circuit is unaltered allowing this feature to be safely turned on and off without changing the defined structure. This feature can be enabled by the **Effective Index Calculation** in the Global Settings dialog. The EIM technique greatly reduces the computation time and memory requirements of a simulation however the results will be less accurate than for a full 3D simulation.

6.H. Computing Far-Fields

BeamPROP supports the computation of far-field transformations of near-field outputs. Far-fields can be automatically computed for both ‘Field Output’ (the field at the final Z position) and ‘Mode Output’. To enable far-field output, you must first enable the desired near-field output(s), then set the **Far Field Output** option in the Output Options dialog.

Both the **Far Field Output** option and either the **Field Output Format** and/or **Mode Output Format** options in the Output Options dialog must be enabled to get results. If the phase information is important, which is usually the case for field format but not mode output, either Amplitude/Phase or Real/Imag format must be used for the near field so that phase information is included.

A description of the method used to compute the far-field can be found in the documentation for the utility `bdutil` in Appendix E of the RSoft CAD manual.

6.H.1. Far-Field Types

The **Far Field Output** option has several choices:

- *None*

  This disables the output of far fields and is the default value.

- *Intensity*

  This produces far field plots of intensity vs. angle.

- *Amplitude/Phase*

  This produces far field plots of amplitude and phase vs. angle.

For 'Field Output', the data file produced will have a name of `<prefix>.far` and an accompanying WinPLOT command files `<prefix>.pfr` and `<prefix>_polar.pfr` where `<prefix>` is the **Output Prefix** used. 'Mode Output' will produce far fields for each mode named `<prefix>.f##` with accompany WinPLOT command files `<prefix>.pf#` and `<prefix>_polar.pf#`. For the *Amplitude/Phase* far field output type, WinPLOT command files are also output to display the phase of the far-field.

6.H.2. Additional Far Field Options

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

- 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 have the far-field calculated in another 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, set the variable `farfield_periodic` to `1` which will produce the far-field as a list of plane waves with different amplitudes along each diffraction order.
- The normalization of the output can be controlled via the variable `farfield_normalization`. A value of `0` (the default) leaves the field unnormalized and a value of `1` normalizes the peak field value to 1.

6.H.3. Manually Computing Far Fields

Far field calculations can also be performed on previously generated mode or field files via the command line utility `bdutil` noted in Appendix E of the RSoft CAD manual.

7

Basic Tutorials

This chapter contains several basic tutorial examples which illustrate the basic use of the BeamPROP simulation engine. These tutorials are increasingly complex and build on concepts established in earlier tutorials. It is important that you 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 in this chapter before moving to the advanced tutorials in Chapter 8.

These tutorials do not have associated `.ind` design 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 1.1, and a width of 7 μm. We will calculate the transmitted power along 1000 μm of this waveguide.

This tutorial will introduce basic concepts about the RSoft CAD interface and the BeamPROP 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 toolbars contain icons for standard file, editing, and viewing operations.

- *A toolbar along the left edge of the window*

  The left toolbar contains icons for standard CAD operations such as selection, zooming, and adding different components. Additional icons control 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 see the CAD manual.

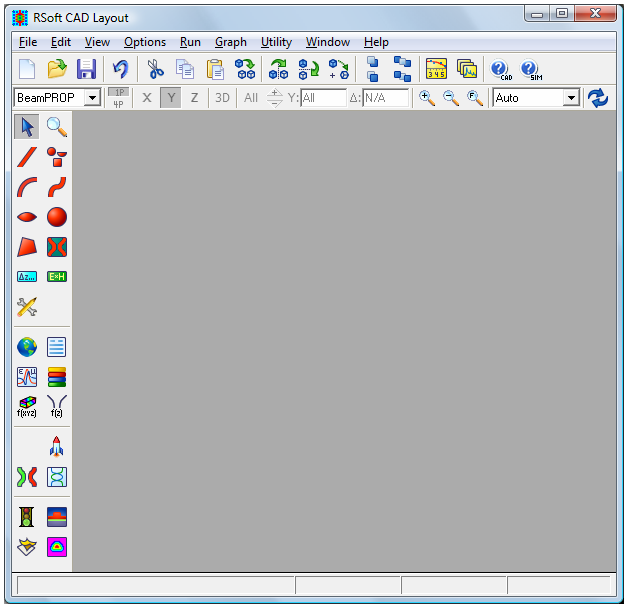

Figure 1: The RSoft CAD program window, showing the menubar at the top, the top and left toolbars, and the status line at the bottom.

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). The startup dialog appears, shown in Fig. 2, where basic information about the circuit to be modeled is entered, 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.

Choosing Initial Settings:

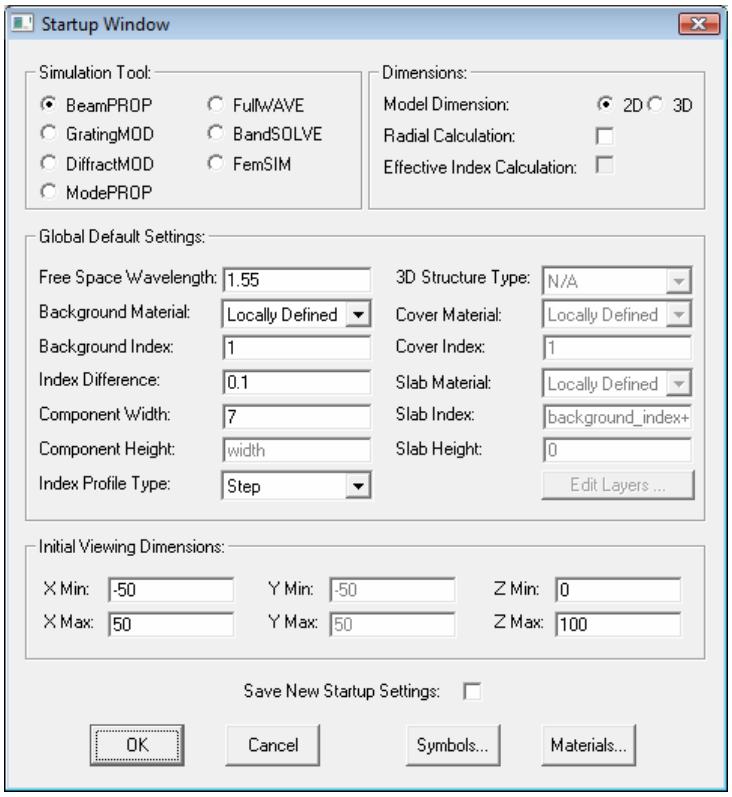

Figure 2: The startup dialog which appears whenever a new circuit is created, and requests basic information about the circuit to be modeled.

For this simulation, we will use the following settings:

Parameter	Value	Description
Simulation Tool	*BeamPROP*	Sets BeamPROP as the simulation engine to be used.
Model Dimension	*2D*	Creates a 2D simulation.
Free Space Wavelength	`1.55`	Sets the wavelength [μm] for the simulation.
Background Index	`1.0`	Represents the real refractive index of the background material where no structure is defined.
Index Difference	`0.1`	Represents the default difference between a component and the background material. In this case our structure will have a default index of 1.1; each component can have a unique value if desired.
Component Width	`7.0`	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 CAD 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].

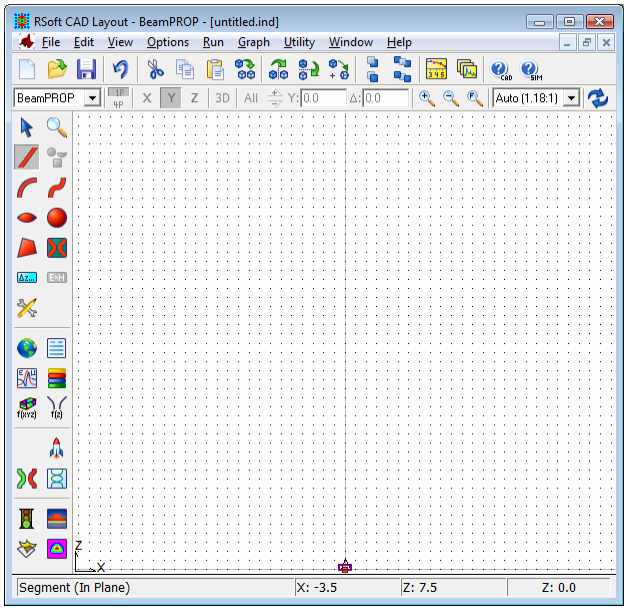

Figure 3: The layout window where waveguide components are added to the circuit.

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 startup dialog box above.

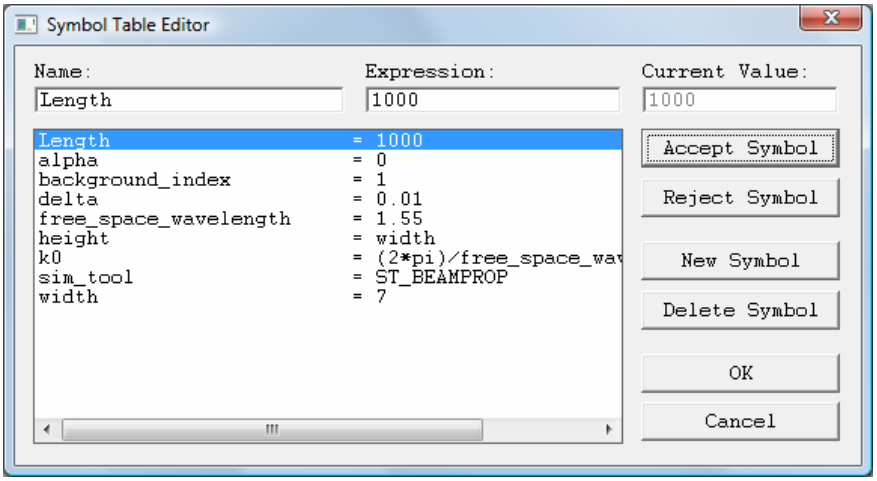

Figure 4: The Symbol Table Editor. The variable `Length` has been defined with a value of `10`.

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 `1000`. 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, with the center line displayed in black, the left and right edges in dark red, and the interior in light red.

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.

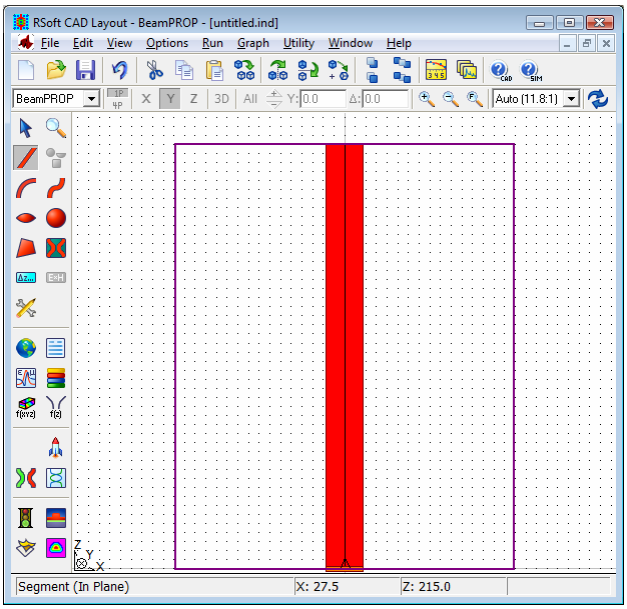

Figure 5: A straight waveguide segment as it appears after drawing, with the center line drawn in black, the left and right edges in dark red, and the interior in light red. The simulation domain and launch field is also shown.

Setting the Length of the Waveguide:

For this example, we are going to set the length of the waveguide to 1000 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.

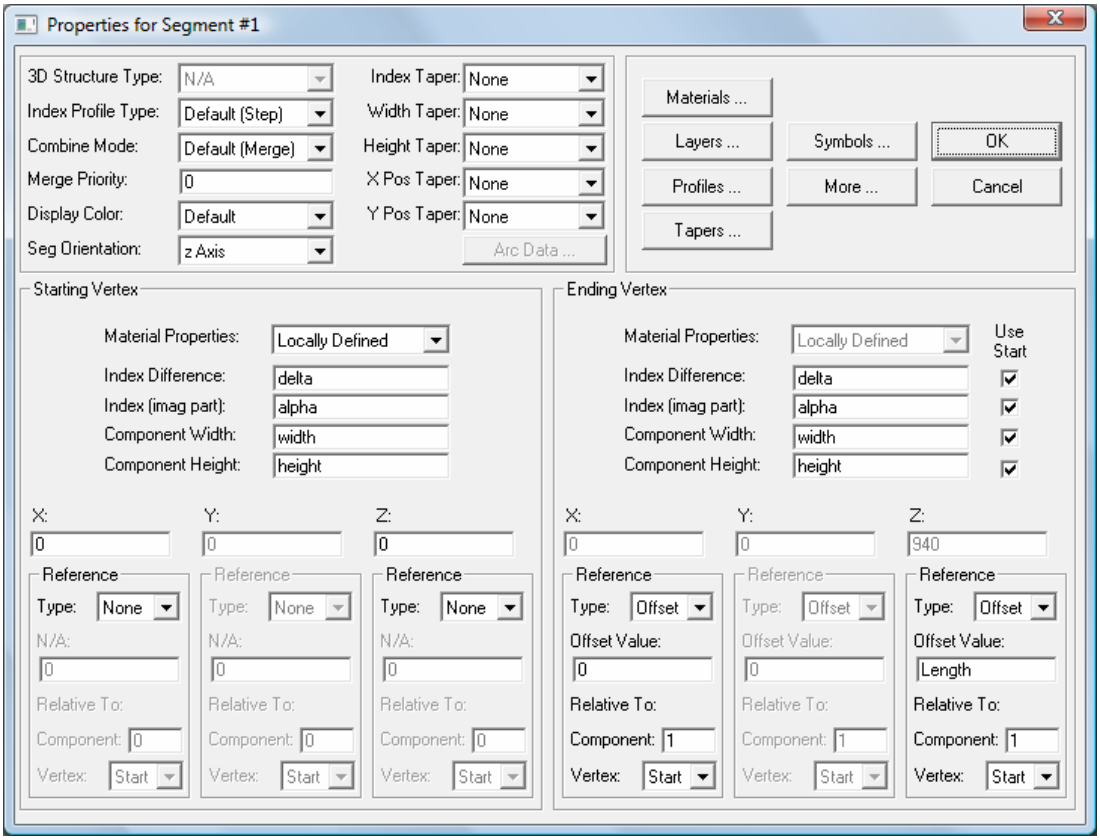

Figure 6: The Segment Properties dialog box for the waveguide segment in this design.

In order to set the length of this segment, make sure the **Reference Type** of the Z coordinate of the Ending Vertex to be *Offset*, and in **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.

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 of the display window. To set the aspect ratio, use the pulldown option in the view toolbar. You may also have to zoom in to see the whole structure. This can be done via the buttons 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 only has a spatial aspect in BPM. 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). This dialog can also be reached via the BeamPROP 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. 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, BeamPROP 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.

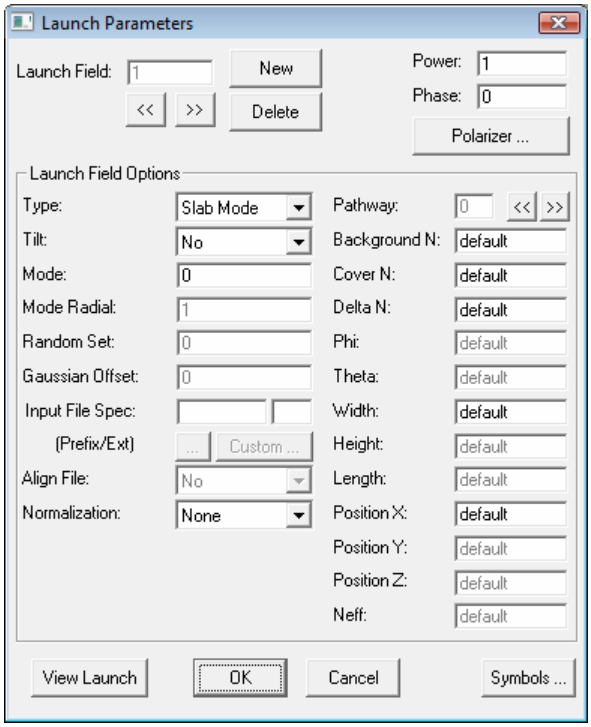

Figure 7: The Launch Parameters dialog.

In this case, the default launch settings are correct. However, it is useful to point out 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, BeamPROP will choose the waveguide with the leftmost position in the design for the launch position (i.e. the segment that has an x position that is the least). 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 can be found in the CAD manual.

A complete description of the options in this dialog can be found in Section 3.D.

Adding a Pathway

As mentioned above, pathways are a way of defining where a launch field is defined. However, in BeamPROP they are also used as part of the monitor analysis. The pathways are used to tell the monitors where to perform computations for the monitor analysis. To add a pathway, press the **Edit Pathways** icon on the left toolbar. The left toolbar will change to list the pathways that are defined. Press the **New** button and then left mouse click on the segment. The segment will turn bright green which means it has been selected as monitor #1 as shown in Fig. 5. Press **OK** to go back to the main CAD window.

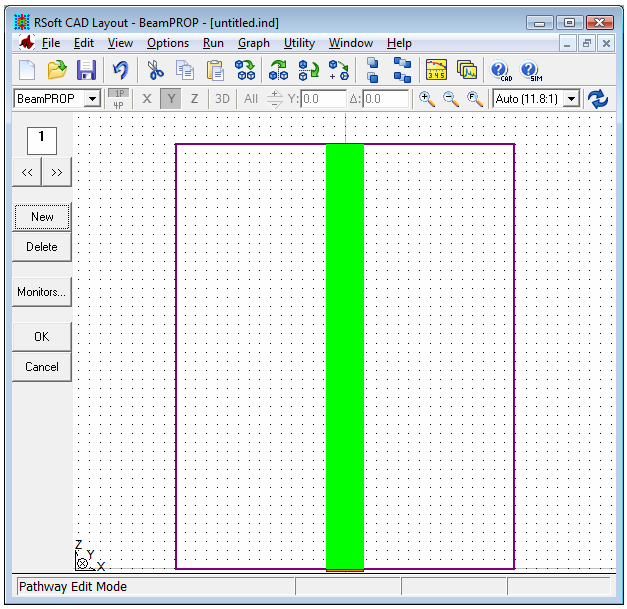

Figure 8: Segment being selected as a pathway.

Adding a Pathway Monitor

Pathway monitors allow measurements to be made during a simulation for viewing or processing. To add a time monitor to the layout, press the **Edit Monitors** button from the left toolbar or the **Monitors** button from the pathway toolbar. The properties dialog for this time monitor will open up. Press the **New** button and the window will be as shown in Fig. 9.

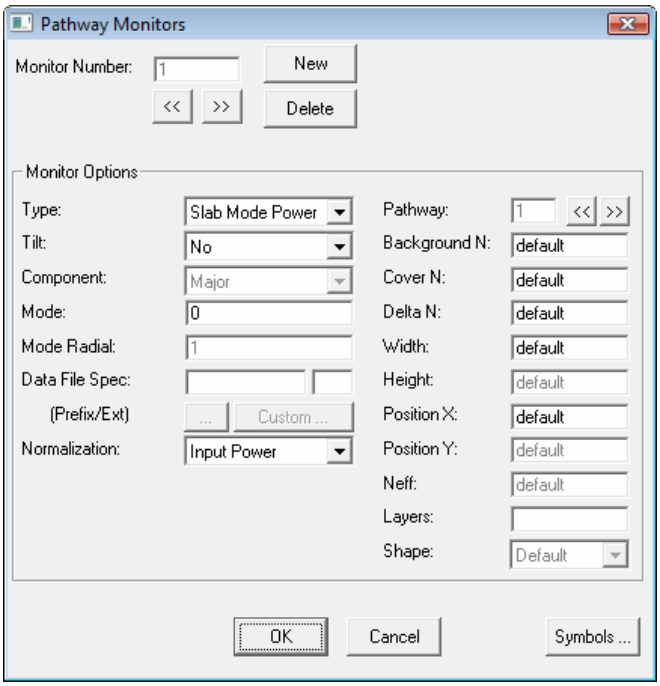

Figure 9: The properties dialog for a monitor.

This dialog allows for many different types of data to be analyzed.

More information on monitors can be found in Chapter 4 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 position, and so the default values shown in Fig. 9 are sufficient.
Click **OK** to return to the CAD window.

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 BeamPROP Simulation Parameters dialog. This dialog allows the basic parameters required for a numerical simulation, such as the spatial 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. We will use the default grid and domain parameters for this simulation. For this tutorial we will change the way the simulation results are displayed: set the **Display Mode** to *ContourMap (XZ)*. 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.

The default values for the rest of the settings are sufficient for this tutorial. Once the simulation 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. 10. The graphical display on the left shows a color-coded contour plot of the optical field which is updated at the spatial interval specified in the **Slice Grid** field in the BeamPROP Simulation Parameters dialog. The graphical display on the right shows the results generated by the monitor we created. 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.

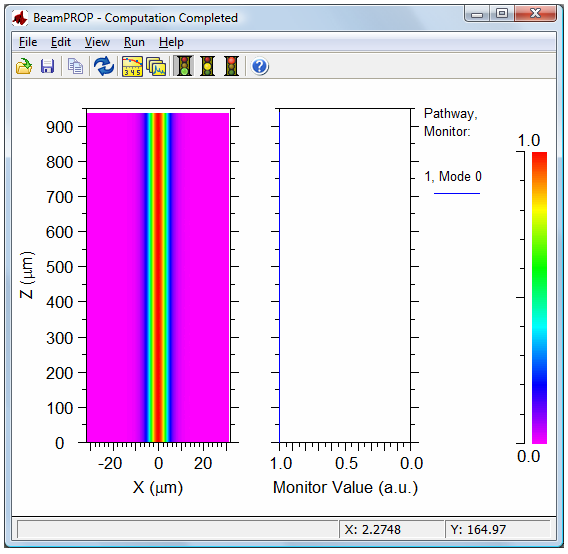

Figure 10: The separate simulation program, after simulating the waveguide structure. The graphical display on the left shows a color-coded contour plot of the optical field. On the right is the monitor result.

Accessing Saved Data

All data saved by BeamPROP 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.pmn`. This file contains the output from the time monitor that we created. Choose this file and click **OK** to open it.

A WinPLOT window which will show the time monitor data. The mouse can be used to zoom in the graph to areas of interest. To zoom out again, simply right-click.

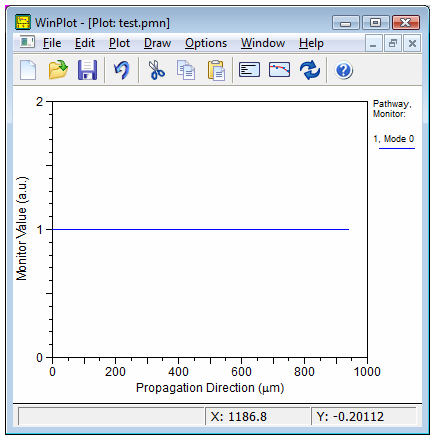

Figure 11: A WinPLOT window which shows the monitor result for this simulation. Since the mode we launched is guided within the waveguide, the result is not very interesting.

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. 12. 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 data before display. Clicking the **View Plot** button returns the view to the plot.

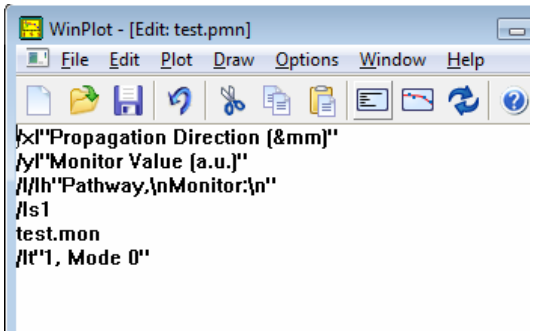

Figure 12: The plotting commands contained in the file `test.pmn` used to display the monitor data stored in the file `test.mon`.

Descriptions for these and other plotting commands can be found in the WinPLOT manual.

Viewing the Raw Data:

The raw data file `test.mon` can be opened in any text editor, and contains two columns: the first shows the z position in units of µm; the second shows the monitor result.

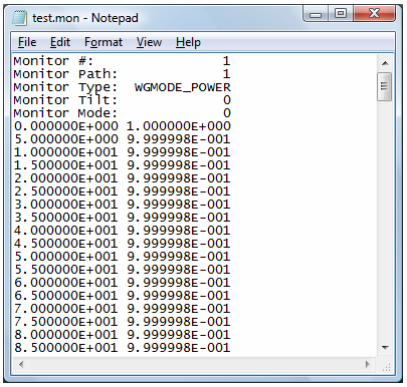

Figure 13: The raw data produced by this simulation.

Viewing other Types of Output:

Most other types of output produced by BeamPROP are controlled by the **Output…** button in the simulation parameters window. A detailed description of the BeamPROP outputs can be found in Section 3.E, and examples of several output types are given in the next tutorials.

Areas for Further Exploration:

Several areas to explore include:

- *Producing Complex Monitor Plots:*

  In the above example we used a simple device with no loss. Try creating a simple splitter structure with branches that come out from the waveguide already created. Create pathways and monitors for all the waveguides you add, and rerun the simulation and see how the monitor results change.

- *Showing the Structure Outline:*

  It can be useful to display the structure outline in the simulation window. To do this, open the BeamPROP 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 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 BeamPROP 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 1.49, a core index of 1.5, and a diameter of 9 μm.

This tutorial will introduce basic concepts for a 3D simulation. Familiarity with the concepts presented in Basic Tutorial 1 is 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. Click the **New Circuit** icon in the top toolbar and make the following settings:

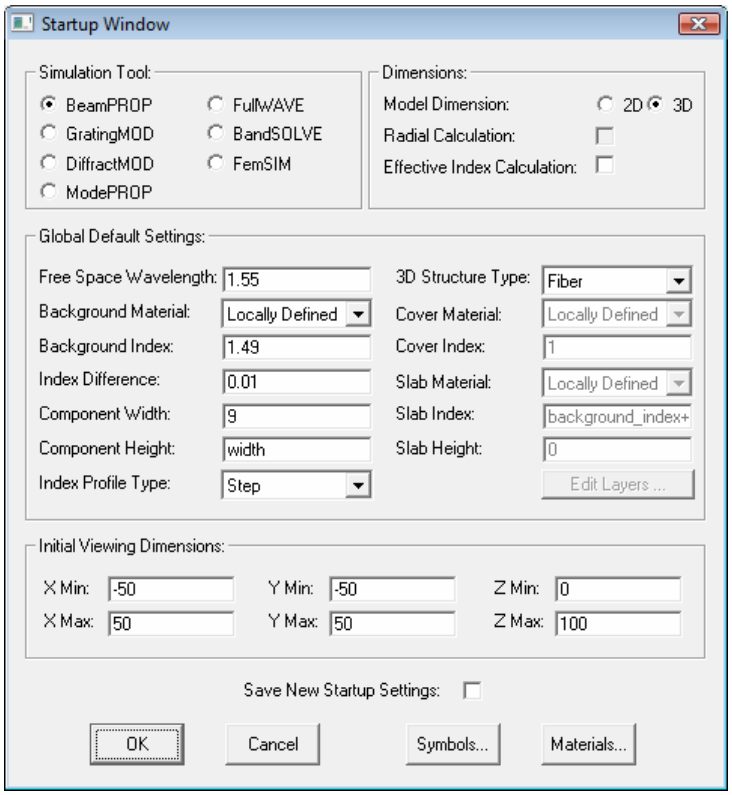

Figure 1: The Startup dialog where which appears whenever a new circuit is created, and requests basic information about the circuit to be modeled. This information can be modified later through the Global Settings dialog.

Parameter	Value	Description
Simulation Tool	*BeamPROP*	Sets BeamPROP as the simulation engine to be used.
Model Dimension	*3D*	Creates a 3D simulation.
Free Space Wavelength	`1.55`	Sets the wavelength [μm] for the simulation.

Background Index	`1.49`	Represents the real refractive index of the background material where no structure is defined.
Index Difference	`0.01`	Represents the default difference between a component and the background material. In this case our structure will have an 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. This choice 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	`9.0`	This represents the default width of a component [μm]; each component can have a unique value if desired.
Component Height	`width`	This represents the default height of a component [μm]; each component can have a unique value if desired. The value here is set to the variable `width` is the built-in variable that 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 500 μm. The symbol table should appear as shown in Fig. 2.

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 BeamPROP can be found in Appendix C this manual.

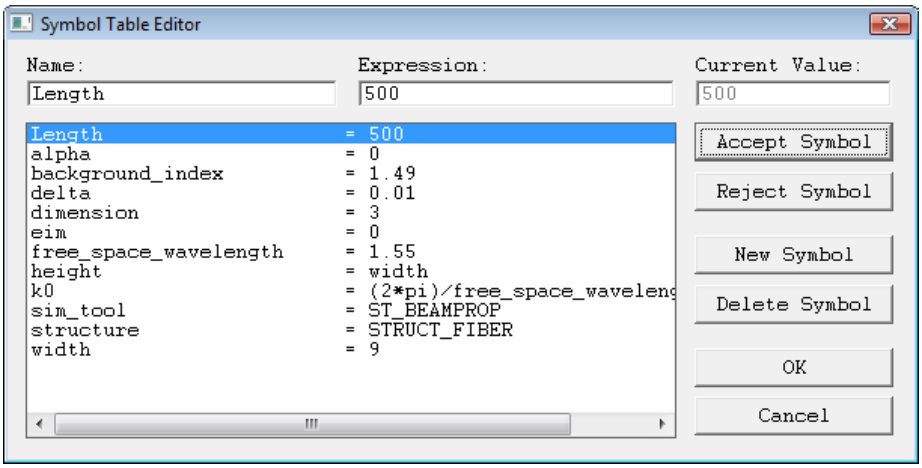

Figure 2: The Symbol Table Editor. The variable Length has been created with a value of 2 μm.

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. Since the CAD presents a top view of the structure, this segment will appear exactly the same as the waveguide segment used in Tutorial 1. The difference in this case is that, because we set the **3D Structure Type** to *Fiber* in the Global Settings dialog, this segment will have, by default, a circular profile.

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` (500 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`).

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.

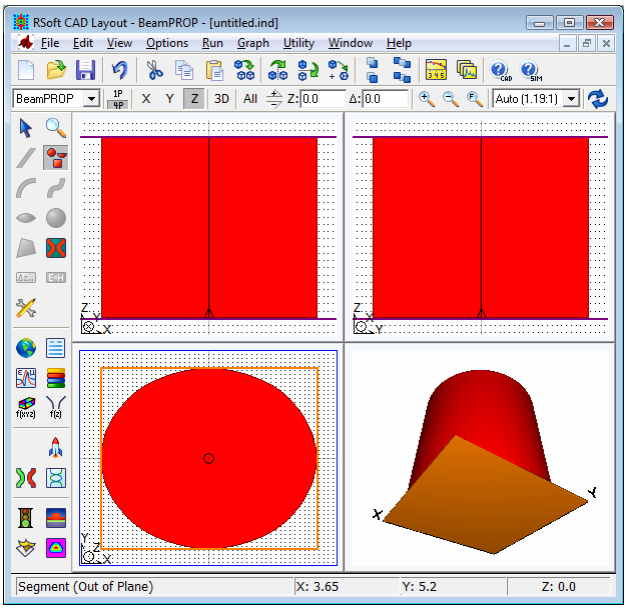

Figure 3: The CAD window in multi-pane mode.

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 Material Profile dialog shown in Fig. 4.

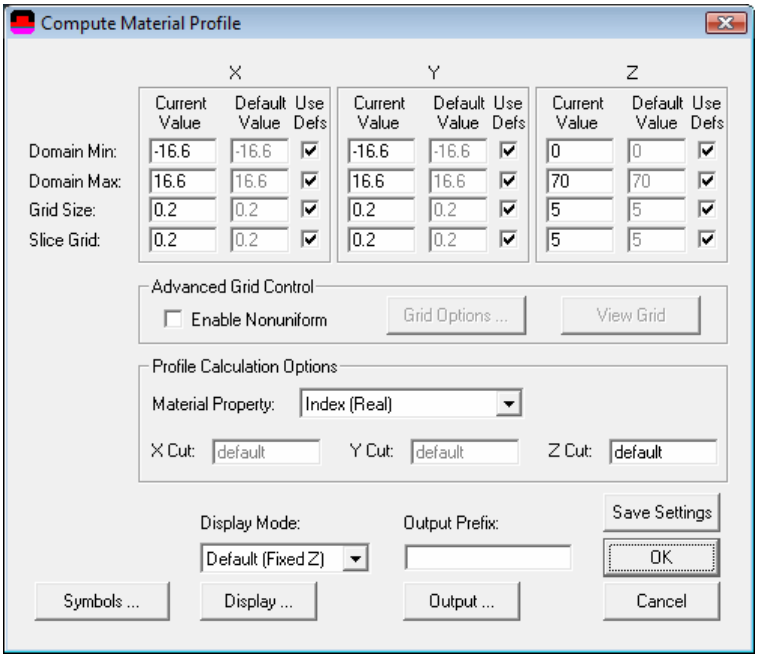

Figure 4: The Compute Material Profile dialog.

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 Grid	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. When completed, the index profile will be shown as in Fig. 5.

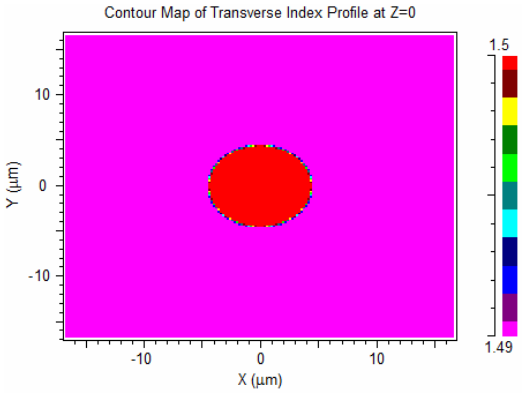

Figure 5: The computed index profile. The fiber cross-section appears elliptical because the aspect ratio of the plot is not set to 1. This can be changed via the **Plot Aspect Ratio** option in the Display Options dialog.

Choosing the Excitation Field

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 Pathway and a Pathway 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 time monitor is very similar to the time monitor used in Tutorial 1.
Add a pathway to the layout and set the segment to be pathway #1. Then create a monitor like in Tutorial 1. Note that the monitor type is now automatically set to be Fiber Mode Power as opposed to Slab Mode Power.

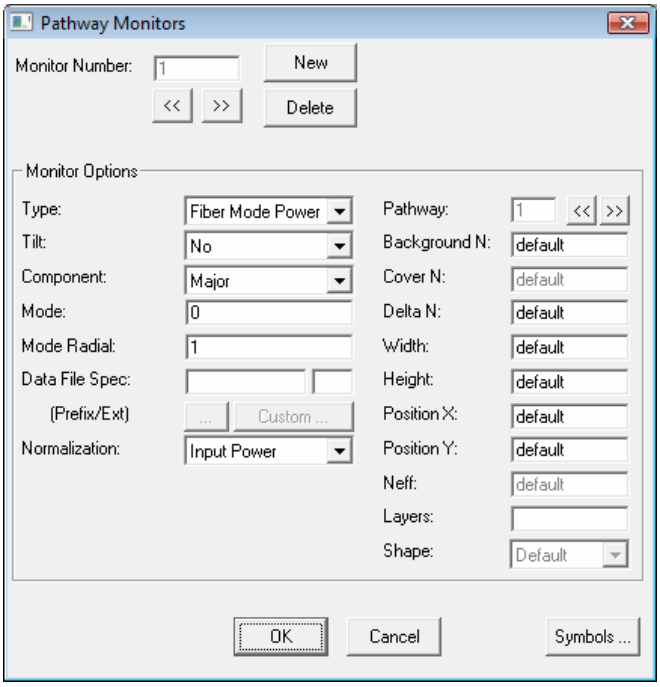

Figure 6: The properties dialog box for the monitor.

Performing the Simulation

To initiate a simulation, click the **Perform Simulation** icon in the left toolbar. Like Tutorial 1, we are going to leave the defaults on.

When performing an actual design, it is important to perform a convergence study with the numerical parameters to ensure that accurate results are obtained.

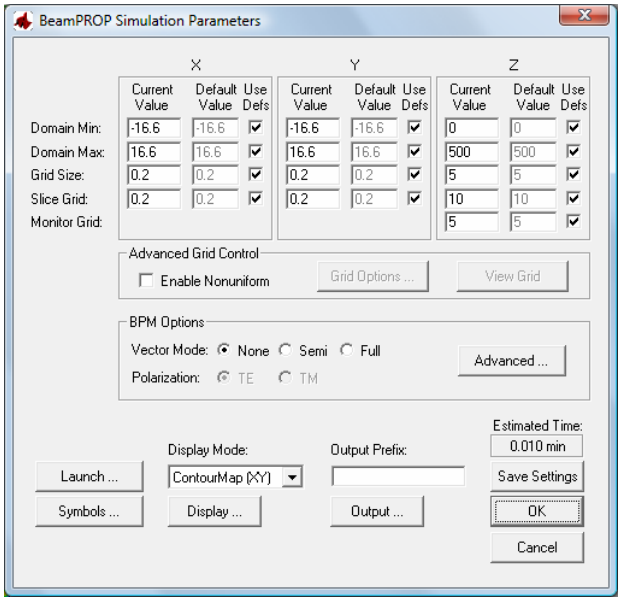

Figure 7: The BeamPROP Simulation Parameters dialog where numeric simulation parameters are set.

Change the **Display Mode** to *ContourMap (XY)* to change the way simulation results are displayed. This choice will display the field slices along the structure as the algorithm integrates along the Z direction. The rest of the parameters are left to their default values.

Selecting Output Types

Different output types can be selected in the Output Options dialog which can be opened by clicking the **Output...** button in the BeamPROP Simulation Parameters dialog. Choose the following output options:

Parameter	Value	Description
Field Output Format	*Amplitude (3D)*	This option enables the output of the field from the final Z position.
Far Field Output	*Intensity*	This option enables the computation of the far field transformation of the field output from the final Z position.

We will create files for the near field at the end of the simulation as well as compute the far field information on this. For more information, please turn to Section 3.E . Press **OK** in this window. Once the numerical parameters and output parameters are set, enter an **Output Prefix** of `test` and click **OK** to begin the simulation. Note that the current Z slice shown is displayed on the bottom of the simulation window in the left portion of the status bar.

Accessing Saved Data

Once the simulation has finished, we can look at the results. Click the **View Graphs** icon and choose the file `test.pfd`. This file contains the simulated field from the final Z position. Next, open the file `test_polar.pfr` which contains the far field transformation of the output field in polar coordinates.

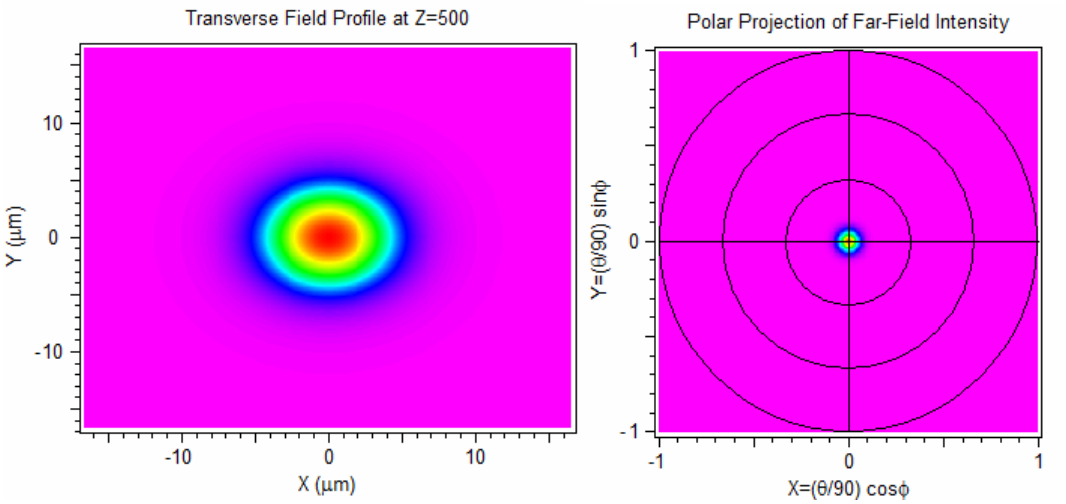

Figure 8: a) The near field from the simulation and b) the far field.

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 outlines the use and meaning of these settings.

- *Change Field 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 Mode** box.

- *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.

- *View the Raw Data for the File Output*

  The field plots opened during this tutorial displayed contour plots of the simulation results. To see the raw data, open the files `test.fld` and `test.far` with any text editor. The data in these files can be used for further processing if needed.

8

Tutorial Examples

This chapter contains several tutorial examples illustrating the use of BeamPROP and its many features for various applications. Corresponding to each example there are one or more `.ind` and/or data files, which are located in the `<rsoft_dir>\examples\BeamPROP\Tutorial\` subdirectory. 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 at info@rsoftdesign.com and we will assist you in finding a solution if it is possible.

Tutorial 1: Simulating Multilayer Structures

This tutorial discusses the setup and analysis of a multilayer structure. Complex waveguide structures, such as those created by etching a ridge into an epitaxial heterostructure, can be realized and analyzed in the BeamPROP package.

This tutorial describes a step by step procedure on how to use the multilayer structure type, define and assign Layer Tables, use overlapping waveguide segments, establish Pathways and Monitors, set appropriate launch fields and simulation parameters, and perform mode calculations and field propagations. Lastly, an analysis of fiber to waveguide coupling is performed.

It is assumed that the user has some familiarity with the RSoft CAD and the multilayer structure. For more on the multilayer structure type, please refer to the RSoft CAD manual.

The associated design file for this example is multilayer.ind and can be found in the `<rsoft_dir>/examples/beamprop/tutorial/tut01/` subdirectory which consists of a multilayer rib structure with an index profile shown in Fig. 1.

Creating the Structure

Open the RSoft CAD and click the **New Circuit** button in the top toolbar. Make the following settings:

Parameter	Value
Simulation Tool	*BeamPROP*
Model Dimension	*3D*
Free Space Wavelength	`1.55`
Background Index	`3.16`
3D Structure Type	*Multilayer*
Cover Index	`1`
Slab Height	`2`

Understanding Multilayer Geometry

A multilayer geometry is characterized by a substrate, a slab, ridges, and a cover. The substrate surface begins at y=0, and is a single index that extends semi-infinitely in the –y direction. The slab extends from y=0 to y=**Slab Height**, and is treated as infinite in x and z. Each ridge is represented by a waveguide segment, and obeys the **Component** (y=0 to y=`height`) and **Component Width** parameters.

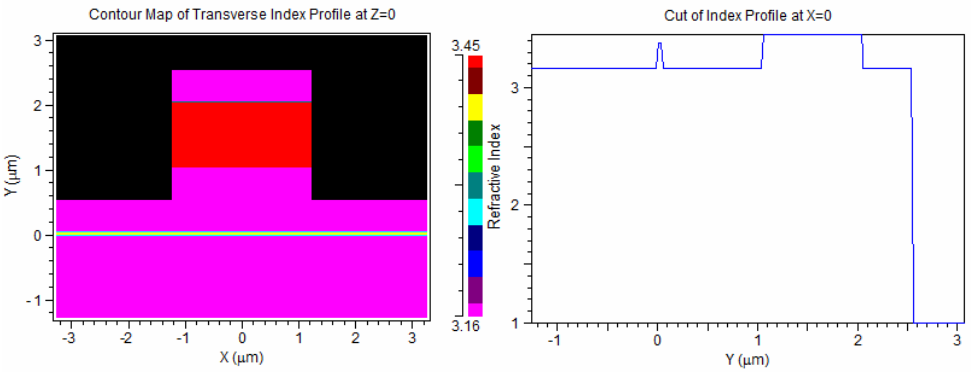

Figure 1: a) The index profile for the multilayer rib structure used in this example. The black region consists of air. b) a vertical cross-section of this profile that clearly shows the index structure.

Layer tables describe the thickness and complex index of each layer in a heterostructure. The total height of the cumulative layers in a table should be equal to or greater than the height of the ridge(s) or slab which refers to it. Any additional thickness is truncated. For more on this please refer to the RSoft CAD manual.

Defining Layer Tables

Click the **Edit Layers…** button to open the Layer Table editor. Click the **Symbols…** button and create the following variables:

Parameter	Value
`N1`	`3.38`
`N1_Slab`	`N1`

```
N2                      3.16
N2_Slab                 N2
N3                      3.45
N4                      3.16
T1                      0.05
T1_Slab                 0.05
T2                      1.0
T2_Slab                 0.5
T3                      1.0
T4                      0.5
X_Pos                   0.0
Y_Pos                   1.5
```

Press **OK** to return to the layer table.

First Layer Table #0 will be defined which sets the index structure of the slab and is also the default for any ridge regions. To add a new layer, click the **New Layer** button, setting the **Height**, **Index(real)**, and **Index(imag)** values, and clicking the **Accept Layer** button. Repeat this for both layers shown in Fig. 2. While this table could be used for the ridge as well, a different layer table will be used for illustrative purposes. To create a new Layer Table (#1), press the **Duplicate** button and create/modify layers as shown in Fig. 3. Click **OK** to exit the Layer Table editor and return to the Global Settings dialog. Finally, define the **Component Width** to be 2.5, the **Component Height** to be `T1+T2+T3+T4`, and the **Slab Height** to be `T1_Slab+T2_Slab`. Click on **OK** to create the design file.

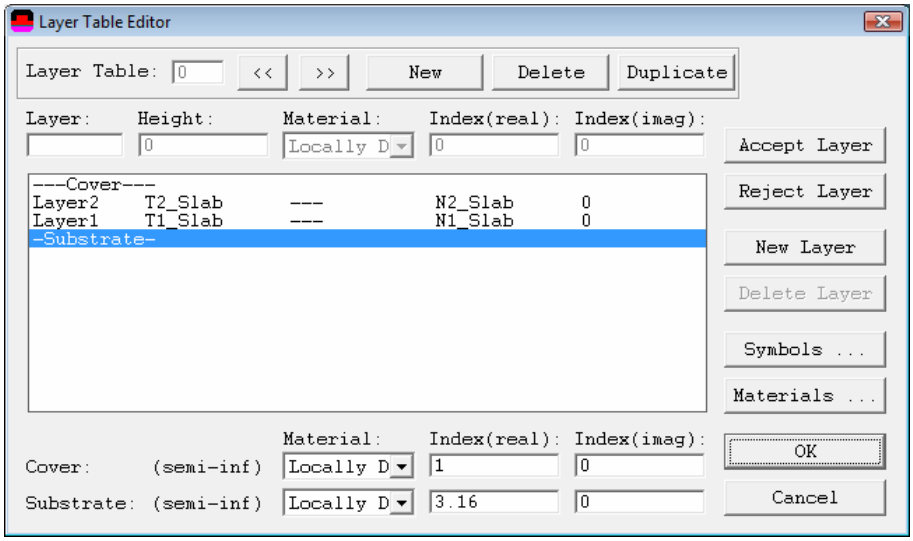

Figure 2: Layer Table 0 which will set the slab information.

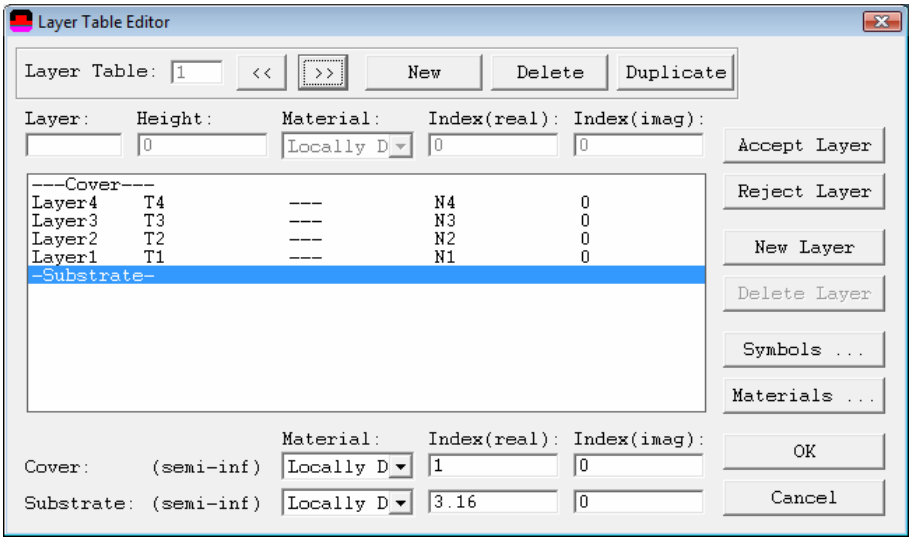

Figure 3: Layer Table #1 which will set the ridge information.

Drawing the Waveguide

To draw the waveguide, press and hold the left mouse button, drag, and then release. Once the waveguide is drawn, right-click on it to open its Component Properties dialog. To assign the Layer Table #1 to this waveguide, set the **Layer Table** for the starting vertex to `1`. Make sure the waveguide goes from (X=0, Y=0, Z=0) to (`0,0,1000`) as shown in Fig. 4 and click **OK** to return to the CAD.

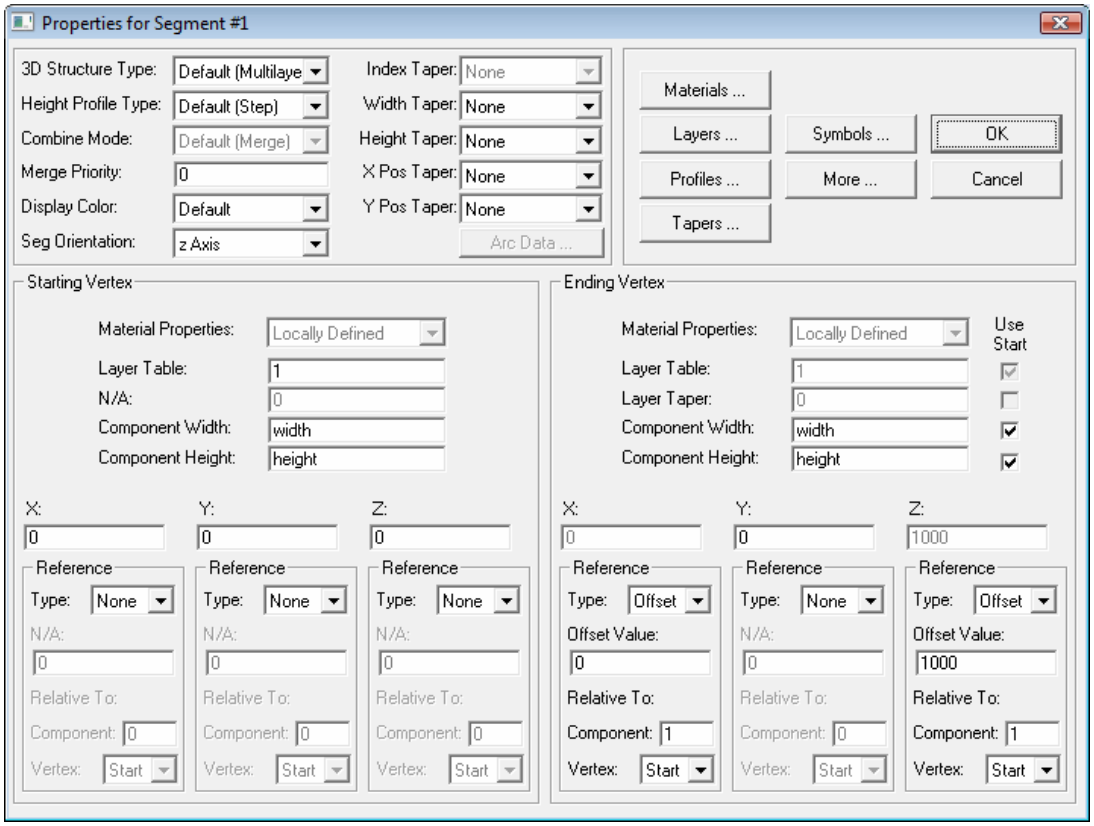

Figure 4: The Segment Properties dialog box where individual segment properties are set.

Viewing the Index Profile

To view the see the resulting index profile, click the **Display Material Profile** button in the left toolbar and click **OK** (the default settings are ok for this case). The profile shown in Fig. 1a should be shown. To see a cross section (like the one shown in Fig. 1b), press the right mouse button on either the X or Y axis (right-click again to return to the full plot). Fig. 1b shows the cross-section at X=0.

Computing Modes

In this section, the supported modes of this rib waveguide will be computed and saved to file(s). See Chapter 5 for more details on mode computations.

Launch Setup

Click the **Edit Launch Field** icon in the left toolbar to open the Launch Parameters dialog and set **Type** to *Gaussian*, **Width** and **Height** to `6.0`, and **Position X** and **Position Y** to `0.2`. The X and Y offsets are important as they will allow for asymmetrical modes to be found. Press **OK** to return to the CAD

Performing the Mode Computation

Click the **Compute Modes** button in the left toolbar to open the Mode Calculation Properties dialog. Click the **Mode Options…** button and set the **Mode Selection** to the *Range* `0` to `3`. Finally, change the Z **Grid Size** to `0.1` and the **Output Prefix** to `mode`. Press **OK** to run the mode computations. When the simulation is complete, the fundamental mode will be displayed. Use the toolbar buttons in the top of the simulation window to view the other modes.

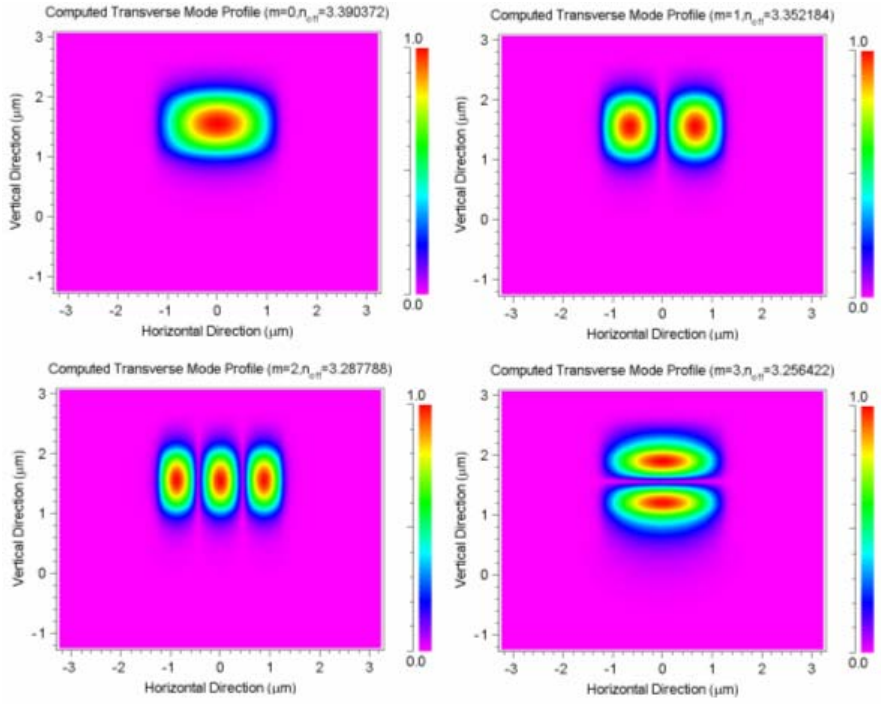

Figure 5: Computed modes for the multilayer structure.

Fiber to Waveguide Analysis

Once the structure has been defined and its supported modes computed, the fiber-waveguide coupling analysis can be performed.

Configuring the Launch Field

While the user can use a data file for a fiber mode launch, it is usually not necessary since these modes can be easily analytically calculated. For this simulation, a fiber mode with a cladding index of 1.49, a core index of 1.5, and a diameter of 9 μm will be used. Press the **Edit Launch Field** button in the left toolbar set **Type** to *Fiber Mode*, **Background N** to `1.49`, **Delta** to `0.01`, **Width** and **Height** to `9.0`, **Position X** to `X_Pos`, and **Position Y** to `Y_Pos` where `X_Pos` and `Y_Pos` are symbols that need to be created in the symbol table and set to a value of `0` and `1.5` respectively. Press **OK** to return to the CAD window.

Creating a Pathway

Click the **Edit Pathways** icon on the side toolbar and click the **New** button. Select the waveguide segment; it should turn bright green. Click **OK** in the left toolbar to exit pathway mode.

Creating the Monitors

The last step before a simulation can be performed is to create monitors so that the coupling can be measured. Click the **Edit Monitors** button in the left toolbar to open the Monitors dialog. For this example, two monitors will be used: the first will compute the overlap integral of the simulated field and the fundamental mode of the waveguide, and the second will compute the total power traveling within the waveguide.

To define the first monitor, click the **New** button to create the first monitor and set **Type** to *File Power*. Set the **Data File Spec** to the fundamental mode previously computed (`mode.m00`). This can easily be done by selecting this file via the **...** button which automatically sets the **Prefix** to `mode` and the **Extension** to `m00`. To define the second monitor, click the **New** button again and set the **Type** to be *Partial Power*. Set the **Width** to `6`, the **Height** to be `3`, and the **Position Y** to be `Y_Pos`. This defines a rectangular cross-section within which the program will compute the power. Press **OK** to return to the CAD window.

Performing a Single Simulation

Click the **Perform Simulation** button in the left toolbar to open the Simulation Parameters dialog. Change the Y **Grid Size** to 0.01 and the **Display Mode** to *ContourMap (XZ)*. Click the **Display...** button and set the **Slice Y Position** to be `Y_Pos` so that the simulation will view the field within the central part of the structure. Click **OK** to return to the simulation parameters dialog and click **Advanced** button and set the **Boundary Condition** to be *Simple TBC*. This is done since we will have a lot of initial energy incident on the boundary, and this setting handles this type of case more easily.

Click **OK** to return to the Simulation Parameters dialog and click the **Output...** button and set the **Field Output Format** to *Amp/Phase (3D)*, and click **OK** to return again to the Simulation Parameters dialog. Enter an **Output Prefix** of `sim` and click **OK** to start the simulation.

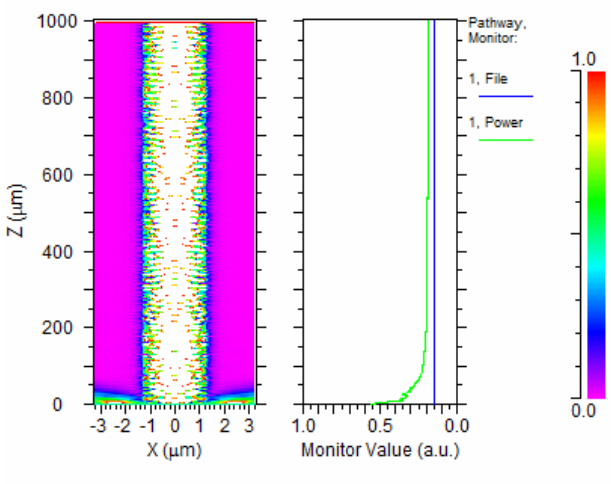

Figure 6: An example of the propagating fields seen during the simulation.

The results (Fig. 6) it can be seen that the total power being guided in the waveguide is about 18% of what was launched. In addition, about 14% of this is in the fundamental mode. We can also see the beating of the individual modes in this plot. Press the **View Graphs** button and open the plot `sim.pfd` to view the output field of this device.

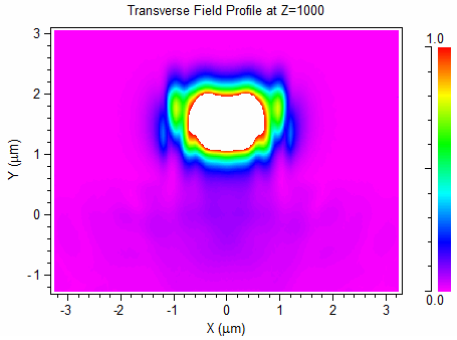

Figure 7: Output near field of the device

Running a Parameter Scan

A parameter scan can be used to see how the device performs as one or more parameters are varied. In this example, RSoft scanning tool MOST will be used to study the effect of the X launch position will be varied to see how the power coupled changes.

Click the **Launch MOST** button in the left toolbar to open icon to open up the MOST dialog. Under **Available Symbols** select `X_Pos` and then press **Add**. Set **Low** to `0`, **High** to `1.5`, and set **Steps** to `11`. This will create a scan where the variable `X_Pos` is varied between 0 and 1.5 in 11 steps.

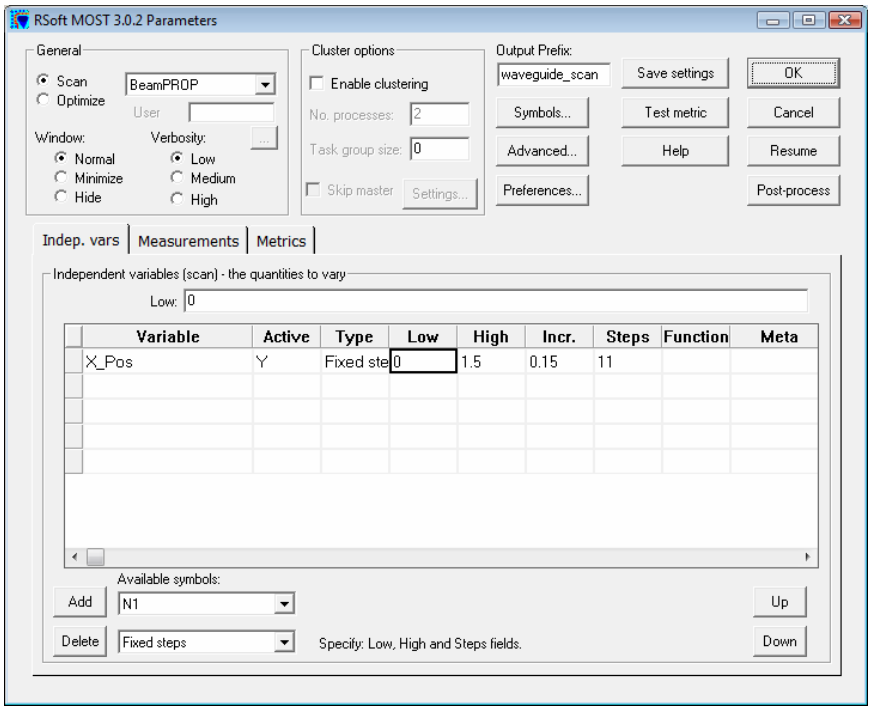

Figure 8: MOST variables setup.

Next, click on the **Measurements** tab in MOST. The default options should be sufficient, but note that the active measurements include the last value of the two monitors.

Finally, enter an **Output Prefix** of `waveguide_scan` and then press **OK**. The MOST window will open and the simulations will be performed. When the scan has finished, click the **Open DataBROWSER** button and open the plot `waveguide_scan_bp_mon_1_last.pcs`. As expected, the power coupling drops off as the misalignment increases.

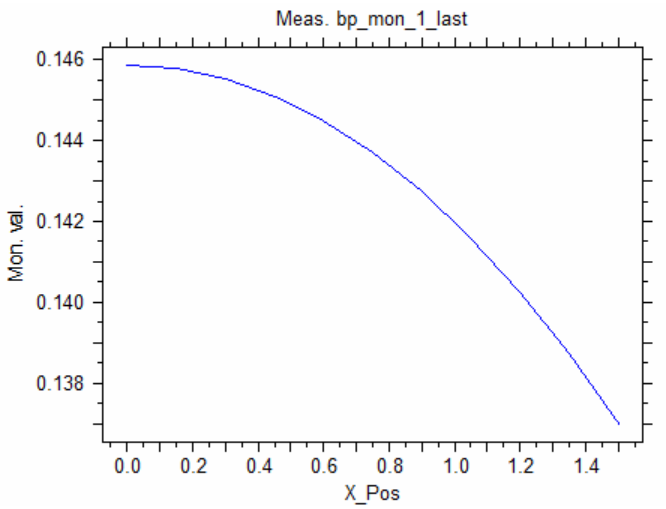

Figure 9: Results of MOST scan for multilayer device.

Areas for Further Exploration

- Use the Symbol Table to change the values of `T3` and `N3`. Observe the effects of these changes on the index profile and on the resulting modes.
- Use the vector BPM options to study the effects of polarization for this example. If this is done, it might be necessary to recompute the mode(s) used for the monitor.
- Compute more modes of the device to see how they operate. Also, create additional monitors to study these modes.
- Perform scans to view the coupling efficiency from fiber to waveguide and vice versa for other parameters such as index, wavelength, and thickness of the device.
- Use the Partial Power monitor to measure the power in just one layer of the waveguide.
- Perform a convergence study on the simulation parameters.
- View the simulation in the XY plane by setting the **Display Mode** to *ContourMap (XY)*.
- Setup an example where waveguide-fiber coupling is analyzed. This can be done by changing the monitor to have a **Type** of *Fiber Mode Power* and then using the correct settings. Also, the launch should be setup to have the modes traveling down the waveguide.

Tutorial 2: Using Simulated Bends

Direct simulation of curved waveguides with large angles can be difficult, even with the Padé-based wide-angle capability available in BeamPROP. The simulated-bend method makes use of a coordinate transformation to map a curved waveguide onto a straight waveguidem, and so is not limitated by the paraxial limitation of normal BPM. It is accurate when $w<<R$, where w is the width of the waveguide and R is the radius of curvature. See the CAD manual for more information about this feature.

This tutorial describes the use of simulated bends for mode calculations and field propagations, and will then compare a simulation of a simulated bend with an actual bend. In all cases, a simple 2D slab waveguide will be used.

The associated file for this example can be found in `bend_mode.ind, actual_bend.ind, mode_match.ind, and sim_bend.ind` located in;

```
<rsoft_prefix>\examples\beamprop\tutorial\tut02\
```

Computing Modes of Simulated Bends

The correlation method is the preferred method for computing bending modes since it can handle leaky and lossy modes. See Chapter 5 in this manual as well as the Mode Tutorials in Chapter 9.

For this tutorial, it is assumed that the user has some familiarity with the correlation method and the use of it. For more on using the correlation method mode computation, please go to the mode tutorials found in this manual.

Creating the Structure

Open the CAD interface, click the **New Circuit** icon on the top toolbar and set the following parameters:

Parameter	Value
Simulation Tool	*BeamPROP*
Model Dimension	*2D*
Free Space Wavelength	`1.55`
Background Index	`1.5`
Index Difference	`0.015`
Component Width	`5`

Click **OK** to create the design file.

Figure 1 shows the setup in the global settings. Press **OK** to open the main CAD window. Press the **Edit Symbols** button to open the Symbol Table, and create/define the following variables:

Variable	Value	Description
`Step`	`.2`	Step to be used for correlation method mode computation.
`L`	`2^12*Step`	Length of the device.
`Radius`	`1500`	Radius of the arc.

Draw a waveguide segment in the CAD and right-click on it to open its Segment Properties dialog. Make sure the waveguide goes from (x=0, y=0, z=0) to (0, 0, L). To enable the Simulated Bend feature, click the More... button and set **Simulated Bend** to *Yes*, and the **Simulated Bend Radius** to `R`. Click **OK** twice to return to the CAD window.

Computing the Modes

In this example three modes will be computed, one for a positive bending radius, one for a negative radius, and one for a straight waveguide. To compute the mode at the current radius (set by the variable `Radius = 1500`), click the **Compute Modes** button in the left toolbar and click the **Mode Options...** button to open the Mode Options dialog. Set the **Method** to *Correlation* and click **OK**. Set the Z **Domain Max**, Z **Grid Size**, and Z **Monitor Step** to `L`, `Step`, and `Step` respectively. Set the **Output Prefix** to `bend_mode_positive` and click **OK** to compute the mode.

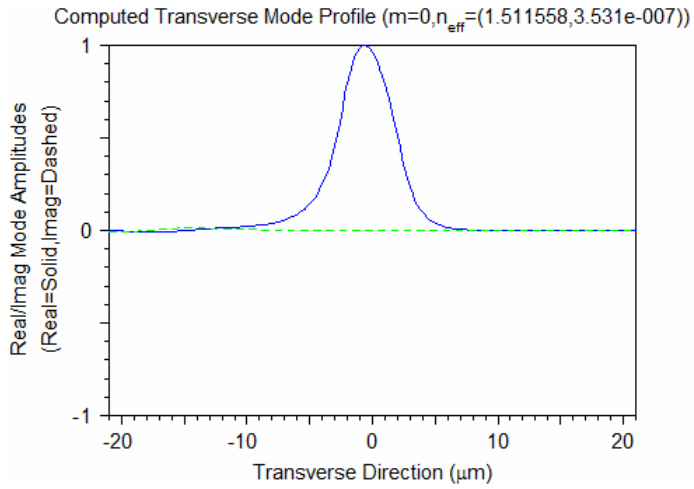

Figure 1: Computed mode for a positive bending radius.

The propagation constant for this mode, which is displayed on the top of the computed mode, is complex. The imaginary effective index indicates the loss of the mode.

Rerun the mode computation with the simulated bend feature off (the straight waveguide) and an **Output Prefix** of `bend_mode_straight`, and again with the simulated bend feature on and `Radius` set to `-1500` and an **Output Prefix** of `bend_mode_negative`. Using RSoft's WinPLOT, it is possible to combine the three modes together on one plot.

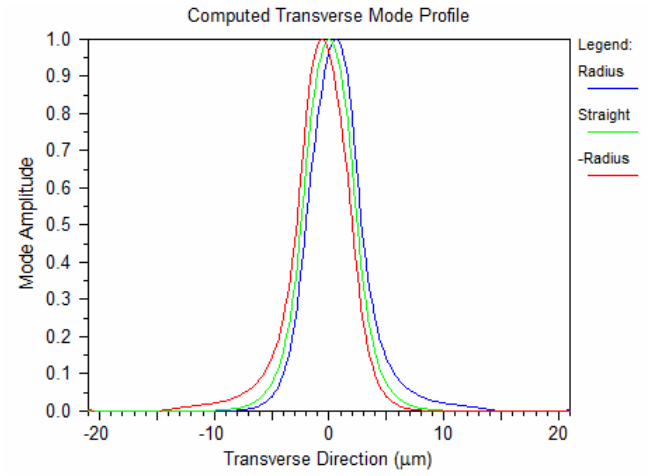

Figure 2: Combined plot of bending modes.

We can see that the negative bending case is shifted to the left, while the positive case is shifted to the right.

Computing Optimal Offsets for an S-Bend

An s-bend segment can be built using the bent segments studied in the previous section. In order to maximize transmission, it is critical to find the optimal offsets between each segment. This can be done by at each joint by computing the overlap integral between mode of the input segment with the mode of the output segment.

To do this in BeamPROP, the mode of the input segment will be used as the launch field and propagated one step in the structure. A monitor will then be created to measure the overlap of this propagating field (really the launch field since it is just one step) with the mode of the output segment. By scanning over the launch position, the optimal offset value will be found. Since the s-bend has three segment junctions, this process will be performed three times.

Computing the First Offset

The first segment junction occurs between the straight input waveguide and the bent waveguide with a negative radius. To recreate this, set the launch field to the fundamental mode of the straight input waveguide previously found (`bend_mode_straight_ex.m00`). This can be done by clicking the **Edit Launch…** button in the left toolbar, set the **Type** to *File*, and choosing this file via the **Input File Spec** setting. To do this, click … and select the file.

Open the Segment Properties of the waveguide and create a variable named `Offset1` and set it be equal to `0`. Set the X Starting Position of the waveguide to `Offset1`. Also, be sure that the variable `Radius` is set to `-1500` to represent negative radius. This creates a situation where, when this variable is scanned, the waveguide will move relative to the launch position.

Next, click the **Edit Pathways** icon, and set the entire circuit to be a part of pathway 1. Press the **Edit Monitors** icon and create a new monitor with a **Type** of *File Power*. Set the **Data File Spec** to use the file `bend_mode_negative_ex.m00` which is the mode previously found for a negative bending radius.

Click **Perform Simulation** and set the Z **Domain Max** equal to `0`. This ensures that the simulation is only performing a mode-overlap calculation since the propagation length is `0`. Clicking **Save Settings** and then press **Cancel** to return to the CAD without performing a simulation.
Click the **Launch Most** icon to open the MOST dialog to create the parameter scan. Create a scan of the variable Offset1 over a range of -1 to 2 with 41 steps and click **OK** to start the scan. The scan results, shown in Fig. 4, show an optimal value of 0.5.

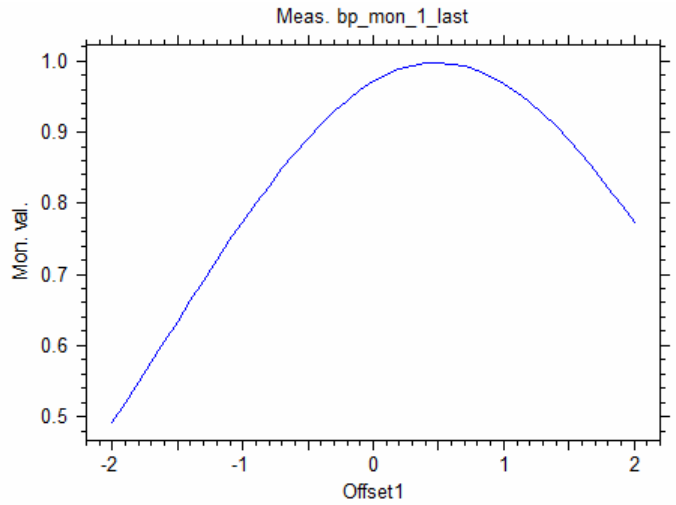

Figure 3: The scan results used to find the optimal value for `Offset1`.

This process should be repeated twice more for the remaining junctions in the structure. The variable `Offset2` is the value of the offset between the two arc waveguides, and the variable `Offset3` is the offset between the second arc waveguide and the straight output waveguide. For each case, both the launch file and monitor file need to be set to the correct file. The optimal values of the offsets are found to be `Offset1=0.5`, `Offset2 = -1`, `Offset3 = 0.5`, respectively.

Comparison between actual S-Bend and Simulated S-Bend

Once the optimal offsets have been found, a simulation of the entire bending structure can be performed. First, the transmission of an actual s-bend will be performed, then of a simulated s-bend.

Using an Actual S-Bend

Open the file `<rsoft_prefix>\examples\BeamPROP\tut02\actual_bend.ind`. This file contains an actual s-bend

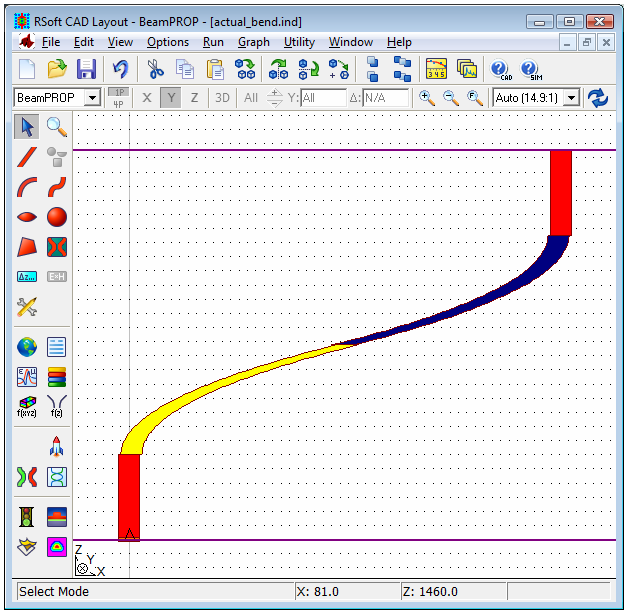

Figure 4: The layout of an actual s-bend in the CAD.

Before running the simulation, note the following things about this structure:

- *Arc Segments*

  This s-bend is made of two straight segments and two arc segments. Right-click on one of the arc segments and note that it has an **X Position Taper** of *Arc*. Click the **Arc Data…** button to see how this arc is defined.

- *Offsets at Segment Junctions*

  Each segment junction is offset by the values found in the previous section to achieve maximum coupling.

- *Launch Field*

  The launch field should be the fundamental slab mode of the straight input waveguide. Since this mode is easy to find analytically, the launch **Type** is set to *Slab Mode*.

- *Pathway and Monitor*

  The entire structure is placed on one pathway and a single monitor with a **Type** of *Slab Mode Power* has been created to measure the power in the fundamental mode along the length of the structure. The Tilt option has been enabled to give more accurate results in the bending sections, though since it is not using the bending mode, will not be as accurate as possible. This is not a problem however as we are only interested in the total transmission.

- *Wide-Angle Settings*

  The field in this structure will be propagating off-axis as it goes through the s-bend, and so the **Pade Order** in the Advanced Simulation Options dialog has been set to *(1,1)*.

Click the **Perform Simulation** button in the left toolbar and click **OK** to start the simulation.

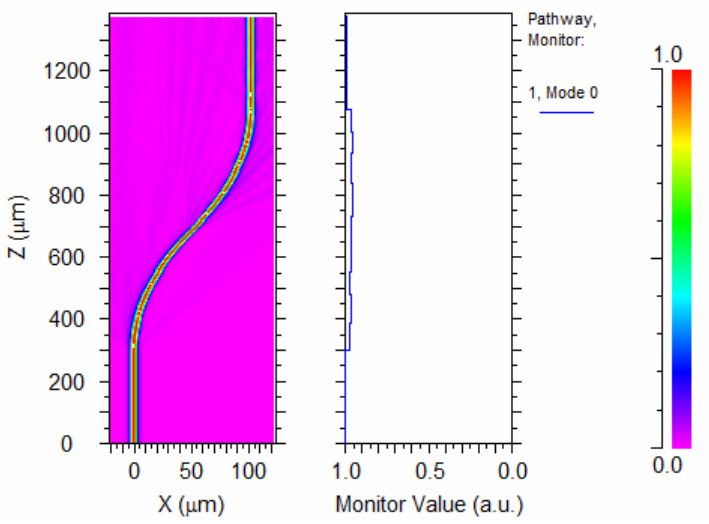

Figure 5: The simulation results for the actual s-bend.

The simulation results show approximately 98% transmission. Note that since we did not use the correct modes in the bent waveguides, the monitor results are not sensible, though the total transmission is still accurate.

Using Simulated S-Bend

Open the file `<rsoft_prefix>\examples\BeamPROP\tut02\sim_bend.ind`. This file contains an s-bend composed of simulated bends.

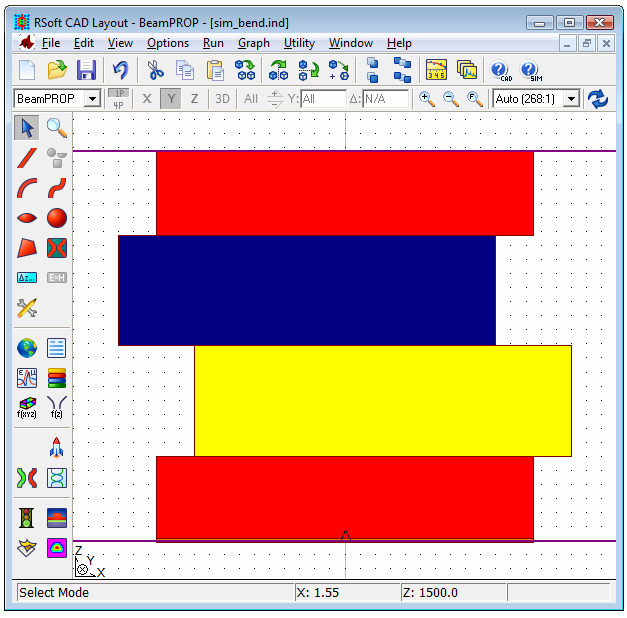

Figure 6: The layout of an actual s-bend in the CAD.

Before running the simulation, note the following things about this structure:

- *Arc Segments*

Again, this s-bend is made of two straight segments and two arc segments. Right-click on one of the arc segments (yellow of blue segments) and note that it has a simulated bend defined.

- *Offsets at Segment Junctions*

  Each segment junction is offset by the values found in the previous section to achieve maximum coupling.

- *Launch Field*

  The launch field should be the fundamental slab mode of the straight input waveguide. Since this mode is easy to find analytically, the launch **Type** is set to *Slab Mode*.

- *Pathway and Monitor*

  Again, a single pathway and monitor are used to measure the power in the fundamental mode of the straight waveguide through the entire structure. Since only the transmission is of interest, this is not a problem.

- *Wide-Angle Settings*

  The field in this structure does not propagate off-axis as it goes through the s-bend, and so no wide angle settings are necessary.

Click the **Perform Simulation** button in the left toolbar and click **OK** to start the simulation.

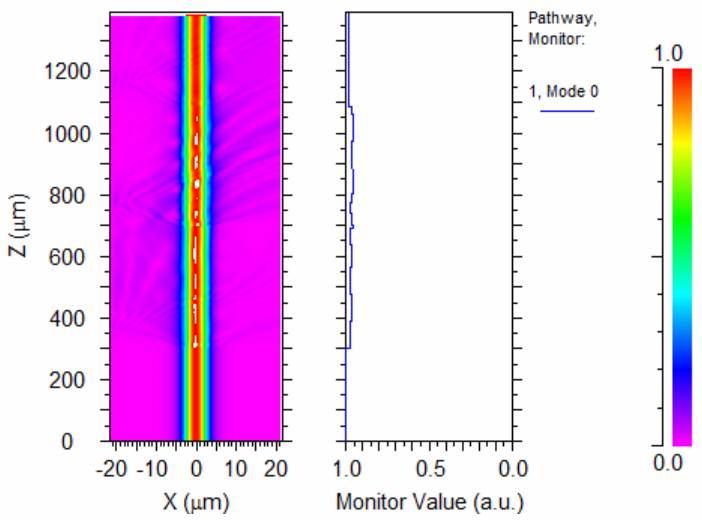

Figure 7: The simulation results for the simulated s-bend.

Again, the simulation results show approximately 98% transmission.

Areas for Further Exploration

- This example was in 2D. Try performing a 3D mode computation for a simulated bend and analyze the results.
- Change the wide-angle settings such as the Pade order for the actual s-bend and see how this affects the results.

- Rerun the simulations of the simulated bend and s-bend using additional pathways and monitors to measure the actual power in the correct modes for the bending regions. These modes are computed in the first part of the tutorial.

Tutorial 3: Multimode Interference (MMI) Devices

This tutorial describes the layout and simulation of a 1x3 MMI coupler in BeamPROP. Multi-mode interference (MMI) couplers utilize a specific integral relationship among the propagation constants of different modes to achieve the self-imaging of the input field along the length of MMI couplers. MMI couplers have been implemented for a variety of optical signal processing functions, such as splitting, switching and routing. This tutorial is based on the following reference:

1 L. B. Soldano and E. C. M. Pennings, "Optical multi-mode interference devices based on self-imaging: Principles and applications," *J. Lightwave Technol.*, **13**, 615 (1995)

The associated file for this example is located in

```
<rsoft_dir>\examples\beamprop\tutorial\tut03\mmi.ind.
```

Creating the Structure

In this section we will create the device in the CAD; first the input waveguides, then the multimode section, and finally the output waveguides. Finally pathways and monitors will be created.
Open the CAD and click the **New Circuit** button in the top toolbar. Make the following changes and then click **OK** to create the design file.

Variable	Value
Simulation Tool	*BeamPROP*
Model Dimension	*2D*
Free Space Wavelength	`1.0`
Background Index	`1.0`
Index Difference	`0.1`
Component Width	`5`

Defining Symbols

It is always suggested to use symbols as much as possible so that the design is organized and easier to modify for simulation purposes. Press the **Edit Symbols** button in the left toolbar and create the following symbols.

Variable	Value	Description
`Lmmi`	`1500`	Length of MMI section.
`Lin`	`200`	Length of the input waveguide.
`Lout`	`Lin`	Length of the output waveguides.
`Wmmi`	`50`	Width of MMI section.

At this point, since the optimized imaging length is unknown, we can pick an arbitrary value for `Lmmi`. From the reference, we know that based on the MMI imaging formula the characteristic imaging length is:

$$L_\pi = \frac{4 n_f W_{mmi}^2}{3\lambda_0}$$

In this example, n_f is 1.1, W_{mmi} is 50 μm , and the 1x3 center input imaging length is $L\pi/4$. Thus the estimated length is 916.7 μm. For this example a slightly longer length has been set the actual optimal length will be determined through simulation.

Drawing the Structure

For references, the final MMI structure is shown in Fig. 1. Draw the input waveguide and right click on it to open its Segment Properties dialog. Make sure that the segment goes from (x=0,z=0) to (0, `Lin`).

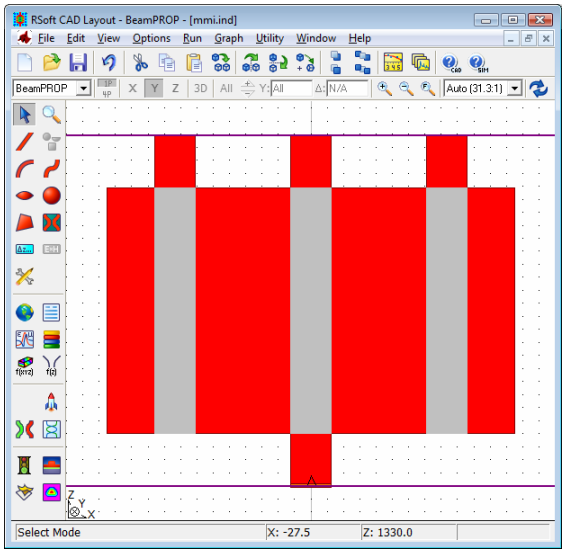

Figure 1: The final MMI structure as seen in the CAD. Each segment will drawn one at a time. The gray segments are inactive and are for analysis.

The large multimode segment can now be drawn. When drawing it, start near the ending vertex of the input segment so that they are logically linked via references (if not, it can be changed later). Set the width and length of this segment to `Wmmi` and `Lmmi` respectively as shown in Fig. 2.

After drawing this segment, it might be necessary to resize the viewing window in the CAD. Click the **Zoom Full** button on the top toolbar.

The output waveguides will be drawn at the 3-fold imaging X positions: `-Wmmi/3`, `0`, and `Wmmi/3`. We also set the length to `Lout`, which we have defined to equals `Lin`. Draw these segments as shown in Fig. 1, making sure that they are defined to have a Z offset of 0 from the ending vertex of segment 2 (the mmi segment). This is so that the output segments are logically connected to the multimode segment and will always stay "connected."

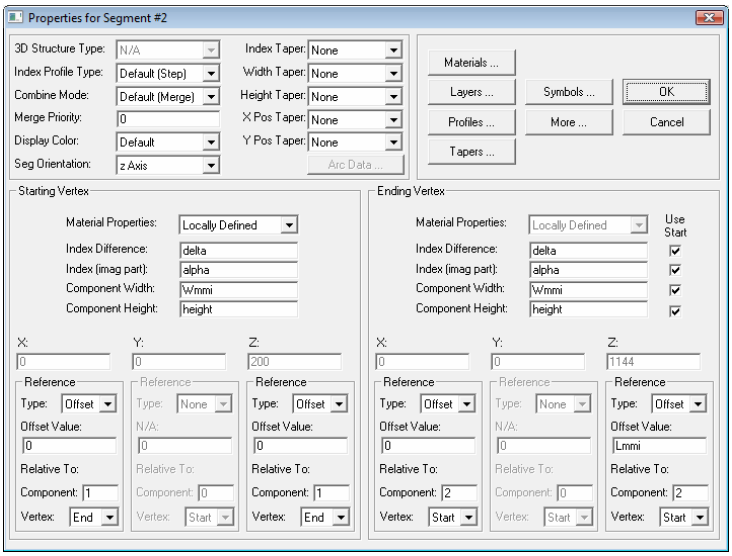

Figure 2: The Segment Properties dialog for the multimode segment. Note that the starting vertex is set to be offset by a value of 0 from the input segment (segment 1).

Analysis Setup

Once the structure has been defined, analysis options such as pathways, monitors, and a launch field can be defined.

Using Inactive Waveguides

In order to determine the optimized 1x3 splitting length, three inactive segments will be used to define the pathways for monitors. Inactive elements are a useful way to setup pathways that are used for monitoring, but do not effect the propagation of the fields. When used in this way, the monitors will measure the output power in each output waveguide as if the multimode region were truncated at that point.

To create the first inactive waveguide, create a new segment and set its starting vertex to be offset by (x=`-Wmmi/3`, z=`0`) from the starting vertex of segment 2 (the multimode segment). Set the ending vertex to be offset by (`0`, `Lmmi`). Finally, set the **Index Profile Type** to be *Inactive*. Repeat this twice more, except change the starting X position to be `0` for the second segment and `Wmmi/3` for the third. The structure should now appear as shown in Fig. 1.

Creating Pathways and Monitors

Three pathways will be created; one for each inactive segment. Monitors will be created to measure the power in the fundamental mode in each pathway. Click the **Edit Pathways** button in the left toolbar, and then click **New** and select the left-most inactive segment and output waveguide as shown in Fig. 3. Next, click the **Edit Monitors** button and then click **New**. Set the monitor type to Slab Mode Power and click **OK** to return to pathway mode.

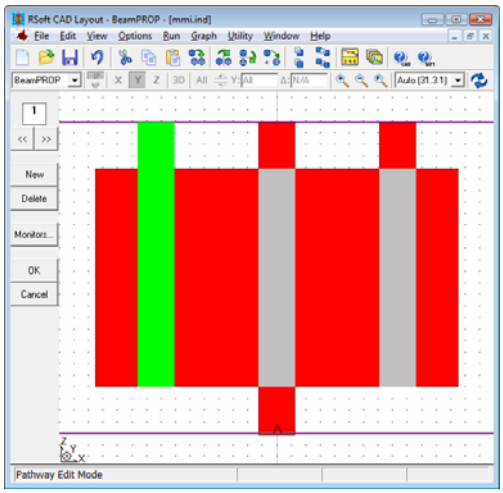

Figure 3: The definition of the first pathway in the CAD window.

Next, follow the same procedure for the second and third pathways.

Defining the Launch Field

The launch field should be the fundamental mode of the input waveguide. While the default values for the launch type are sufficient in this case, the launch position will have to be set. This is because, by default, the launch field is launched into the first pathway. In this case, the first pathway corresponds to the left output port. To remedy this, click the **Edit Launch Field** button in the left toolbar and either set the launch **Pathway** to `2` or set the launch **Position X** to `0`.

Performing the Simulation

Click the **Perform Simulation** button in the left toolbar to open the BeamPROP Simulation Parameters dialog. Enter an **Output Prefix** of `mmi_run` and click **OK** to begin the simulation.

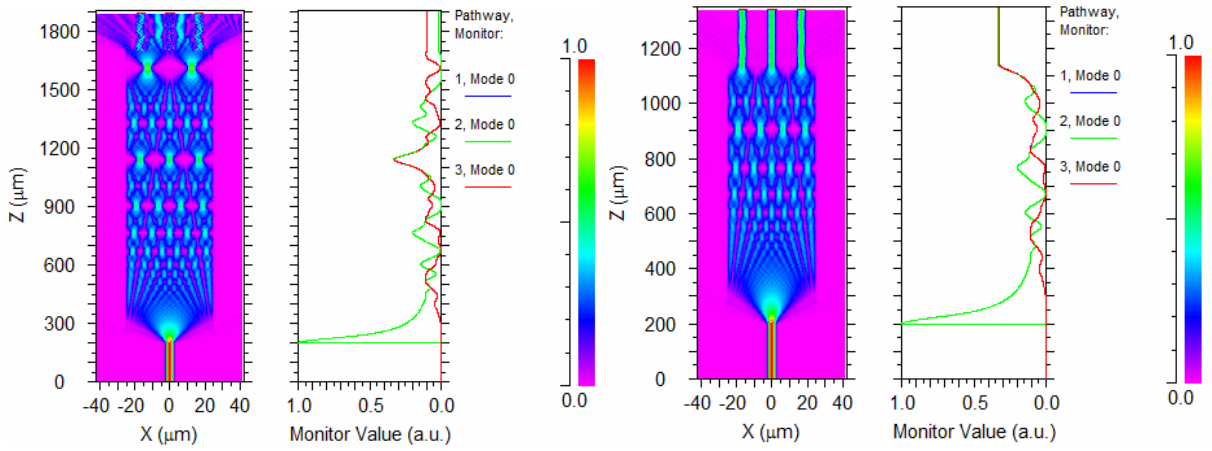

Figure 4: a) The simulation results for the MMI device. Note that the best imaging point is where the monitor plots share a maximum value around Z=1150 µm. b) Simulation results at the optimal length.

Determining the Optimal Imaging Length

Click the yellow **View Graphs** button and examine file file `mmi_run.pmn`. The best imaging point in terms of low loss and imbalance is around Z=1144 μm, where the three power monitors have almost the same local maxima value. The optimal value of `Lmmi` is therefore `944` since `Lin` is `200`.
Change the value of Lmmi to `944` and rerun the simulation. If all the segments to not automatically adjust positions to the new length, then the offsets were not defined quite right during the creation of the device.

Areas for Further Exploration

- Perform a convergence study on this device based on simulation parameters and advanced numerical methods.

Tutorial 4: Nonlinear Coupler

BeamPROP can simulate the effect of the n_2 or χ^3 non-linear parameters within a structure. The effects included are for monochromatic cases where self-focusing or self-phase modulation is concerned. The associated file for this example is found in `<rsoft_dir>\examples\BeamPROP\tutorial\tut04` directory and is based on this reference:

1 K. Yasumoto, "Coupled-Mode Analysis of an Asymetric Nonlinear Directional Coupler," *J. Lightwave Tech.*, **14** 628 (1996)

See Chapter 8 in the CAD manual for information on the definition of non-linear materials.

Structure Overview

This structure consists of a waveguide coupler where the coupling is a function of the input power due to a non-linearity. Without the non-linearity, the evanescent field will freely couple the power from one waveguide to another. However, with the non-linearity, as the input power increases, the index of the input waveguide also increases, making the evanescent field smaller and lowering the coupling.
Open the file `nonlinear_coupler.ind` and note the following about the design:

- *Structure*

  This device consists of two slab waveguides embedded in a uniform background material. The left waveguide has a width of 1.9 μm and the right waveguide has a width of 2.0 μm. Both waveguides are 2100 μm long.

- *Materials*

  Three materials are used to define the refractive indices used: the background region is specified by the material *Background* which has an index of 1.53 and the waveguides are defined by the materials *WaveguideA* and *WaveguideB* which have a linear refractive index of 1.55.

  The left waveguide (material *WaveguideA*) has a non-linear setting for χ^3 set to `Chi3` which corresponds to the parameter α in Ref. [1]. Several adjustments are made to this factor to convert it to the units used by BeamPROP:

  - The parameter in the reference is given in $[m^2/V^2]$ and so a factor of 10^{12} is included to convert to $[\mu m^2/V^2]$.
  - A factor of 4/3 is included since BeamPROP uses a strict interpretation of non-linear parameters.
  - The χ^3 factor can be inversely scaled with the launch power without changing the simulation results. Though not necessary, this technique can be used to produce field values that are near unity and therefore easier to view in the field display in the BeamPROP simulation window. The variable `ScaleFactor` has already been set up for this purpose. It has a value of `1e-6` which scales χ^3 and the launch power to produce field values near unity; setting it to a value of `1`

will remove all scaling. To more easily toggle between these two settings, another variable `Scale` has been created which, when given a non-zero value, will enable the scaling.

- *Launch Field*

  The launch field is the fundamental slab mode of one of the waveguides at a wavelength of 1.064 µm. When working with non-linear materials it is critical to correctly set the launch power and normalization since non-linear effects are directly related to these parameters. In this case, to match Fig. 1 in Ref. [1], we want to launch power values from 0 to 15 [W/m]. To accomplish this, we set the launch **Normalization** in the Launch Properties dialog to *Unit Power* and set the variable `launch_power_unit = 1` so that the specified power is interpreted in [W/µm]. The launch **Power** is set to `2*max(P,1e-10)*1e-6/ScaleFactor` where the built-in `max()` function is used so the launch power is always greater than 0, `P` is the power as defined in the reference [W/m] and is multiplied by 1e-6 to convert to [W/µm]. The factor of `2` is present since we are setting the peak power and not the averaged power. Finally, the previously described variable `ScaleFactor` is also used.

- *Pathways and Monitors*

  There are two pathways defined, and two monitors (one per pathway) set to measure the power in the fundamental mode.

Performing a Simulation

Click the **Perform Simulation** icon click **OK** to start the simulation.

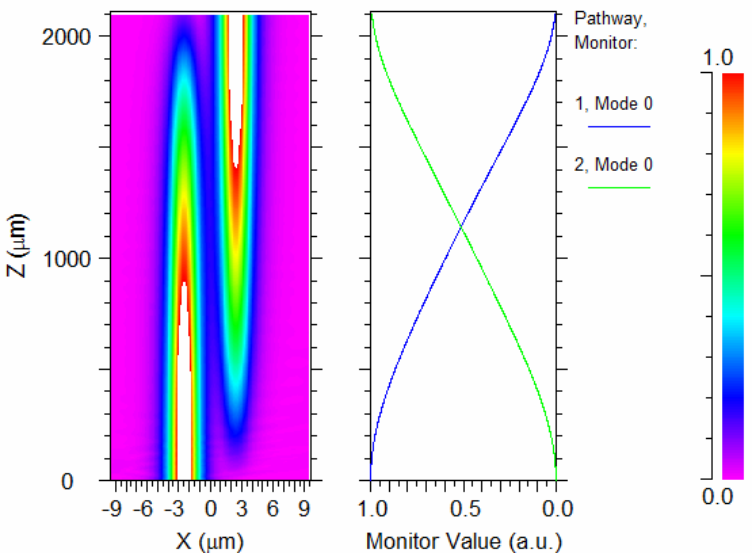

Figure 1: Single Simulation results.

At the current value of `P` (`2.2`) there is complete coupling between the waveguides at a length of 2100 µm. We can use MOST to vary the launch power and see how the coupling changes. Click the **Launch MOST** button in the left CAD toolbar to open the MOST dialog. Note that a scan over the variable `P` is

already created and click **OK** to start the scan. At the end of the scan, open up DataBROSWER and look at the plots that have been created.

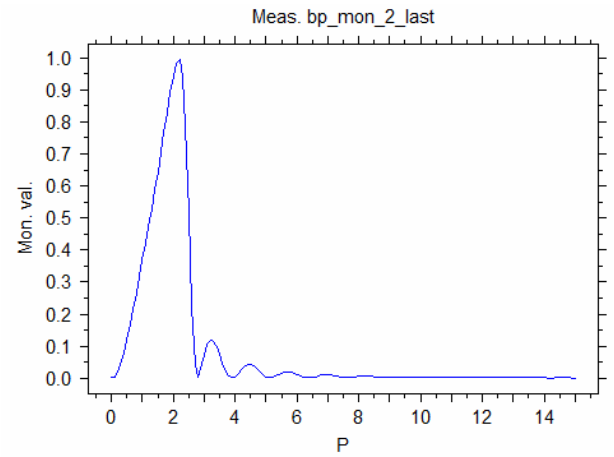

Figure 2: Output power in the right waveguide as a function of the input power `P`. This figure matches Fig. 1 in Ref. [1].

These plots show the dependency of the coupling on the input power. Try running the simulation and scan again with the scaling turned of (i.e. set `Scale` to `0`) and see how the monitor results do not change but the field display is blank. This is because the field values are on the order of 10^{-6} and are too small to show.

Tutorial 5: Using Multiple Launch Fields

It is sometimes necessary to launch multiple launch fields into a simulation with different relative powers and phases. This example illustrates how to use multiple launch fields.
The associated file for this example is located in the directory:

```
<rsoft_dir>\examples\beamprop\tutorial\tut05\.
```

Structure Overview

Open the file `multi_launch.ind` in the CAD interface.

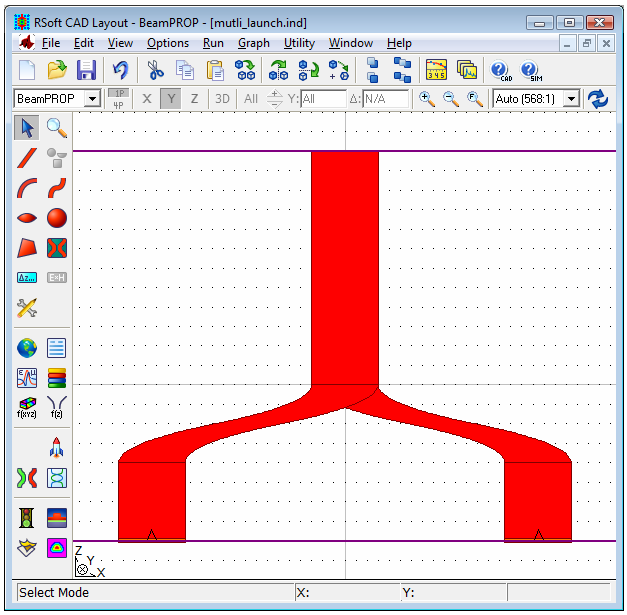

Figure 1: The CAD file to be used in this example.

Note the following about this design:

- *Structure*

  The structure for this example consists of two input waveguides which are connected via two s-bends into one output waveguide.

- *Pathways and Monitors*

  This structure has three pathways, two for the launch field definition, and one for a monitor. The monitor will measure the power in the fundamental mode of the output waveguide.

- *Launch Fields*

  This example has two launch fields, one for each pathway. The **Power** for each launch field is set to `0.5`. The **Phase** of the first launch field is 0, and the second has been set to the variable `Phase`.

Simulating with Multiple Launch Fields

This simulation will scan over the variable `Phase` to see how the launch fields interfere. Click the **Launch MOST** button on the left toolbar to open the MOST window, click **OK** to start the scan.

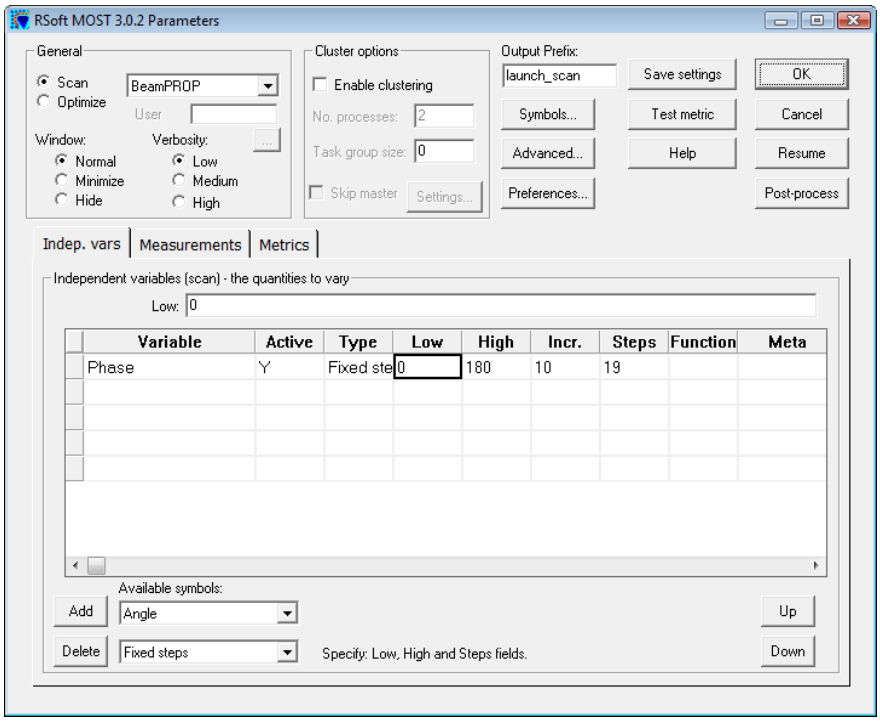

Figure 2: The MOST dialog box.

While the parameter scan is running, you should be able to see the power start to more and more destructively interfere. After running the parameter scan, you should obtain the results shown in Fig. 3.

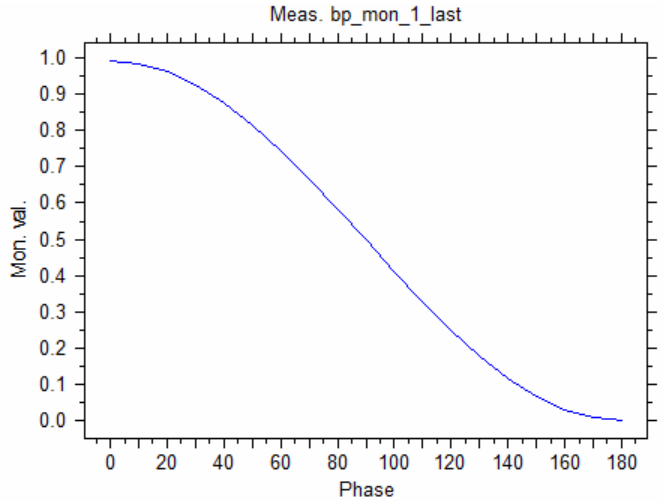

Figure 3: The results of the parameter scan.

The effect of launching two identical waves with different phases is clearly seen.

Tutorial 6: Simulating Anisotropic Devices

This tutorial discusses the setup and analysis of an anisotropic rotator device. Complex anisotropic situations can be realized and analyzed in the BeamPROP package, and require the use of BeamPROP's full-vectoral capabilities. This tutorial will demonstrate two examples: the first will be a faraday rotator, and the second will be a half wave plate.
The associated files for these examples are located in the `<rsoft_dir>\examples\BeamPROP\tutorial\tut06\` directory.

Faraday Rotator Example

In this section, we will study a faraday rotator. Open the file `rotator.ind`. Note the following about the design:

- *Structure*

  This device is implemented as a fiber structure.

- *Anisotropic Material*

  The Faraday Effect is created by defining an anisotropic material in the Material Editor. Open the Material Editor and look at the *Faraday* material to see how this is done. The properties of the waveguide are set to this material in the Segment Properties dialog. Also, note that the **Anisotropic Calculation** option is enabled in the Advanced Simulation Options dialog.

- *Full-Vector BPM*

  Since this simulation will incorporate anisotropic material, it must use the full-vector BPM formulation. This is set in the *BeamPROP* Simulation Parameters dialog.

- *Pathways and Monitors*

  One pathway has been defined. Two monitors have been defined with a **Type** of *Partial Power* (to measure the power within the fiber core). One monitor will measure the *Major* **Component** (E_x in this case); the other will measure the *Minor* **Component** (E_y). These monitors will allow measurements to be made that show the power rotating between the two polarizations.

- *Launch Field*

  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.

Click the **Perform Simulation** button in the left toolbar and click **OK** to start the simulation.

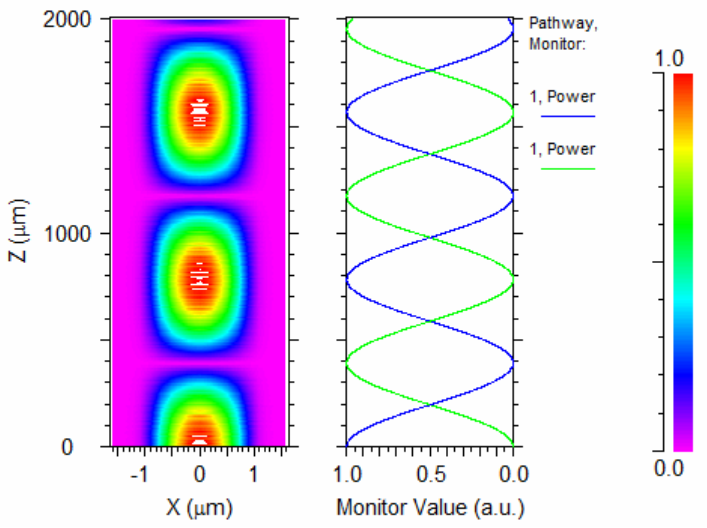

Figure 1: Simulation results of rotator device

The monitor plot on the right clearly shows the power of the system coupling between E_x and E_y. For a 180-degree polarization rotation, the characteristic length for this structure is therefore about 390 µm.

Half Wave Plate Example

In this example, we will create a half wave plate and perform an analysis as to what its characteristic length is. Open the file `calcite_bp.ind`. Note the following about the design:

- *Structure*

  This device is implemented as a bulk region in such a way that there is a constant refractive index at all points within the simulation domain. The green segments are monitors for use with other simulation tools such as FullWAVE, DiffractMOD, and ModePROP. See 'Areas for Further Exploration' section below.

- *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, the **Anisotropic Calculation** option in the Advanced Options dialog is selected.

- *Full-Vector BPM*

  Again, since this simulation will incorporate anisotropic material, it must use the full-vector BPM formulation. This is set in the *BeamPROP* Simulation Parameters dialog. Also, the **Reference K** is set to *Other* and the **Ref K Value** is set to be `k0*(Ne+N0)/2` in the Advanced Options dialog.

- *Pathways and Monitors*

  One pathway has been defined. Three monitors have been defined. Two of these have a **Type** of *Launch Phase* and a **Component** of *Major* and *Minor* respectively (to measure the relative phase of the E_x and E_y fields). The third monitor has a **Type** of *Launch Power* and is set to measure the power in both components.

- *Launch Field*

  The launch field has been set to a rectangular beam to approximate a plane wave at a wavelength of 590 nm. Furthermore, its polarization has been rotated by 45 degrees using the Launch Polarizer (click **Polarizer…** in the Launch Parameters dialog to see these settings).

Simulation of Wave Plate

Click the Perform Simulation button in the left toolbar, set the **Output Prefix** to `waveplate`, and click **OK** to start the simulation.

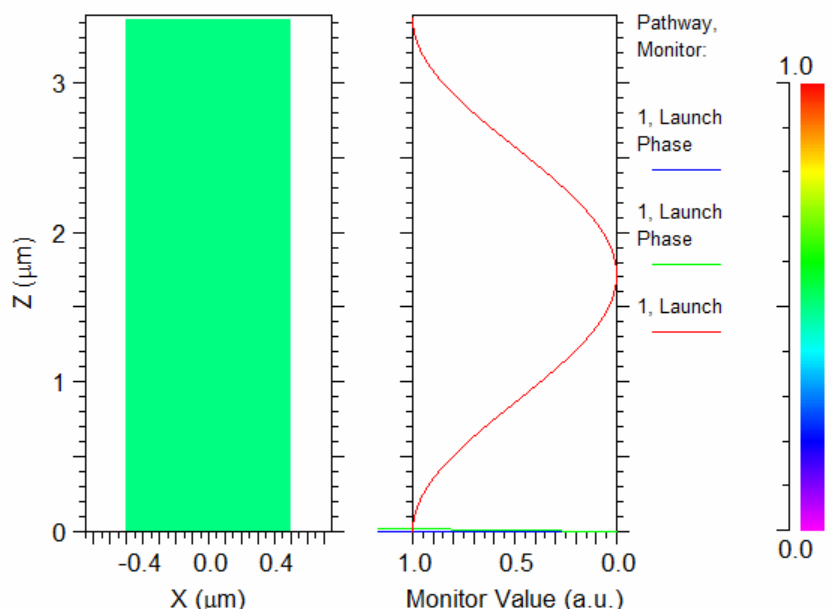

Figure 2: Simulation results for wave plate. The field display shows a constant value since the amplitude of the field does not change.

The monitor result shown in Fig 2. shows the overlap of the launch field with the propagating field and clearly shows the beating pattern. Open the monitor results file `waveplate.pmn` to see the results of the phase monitors.

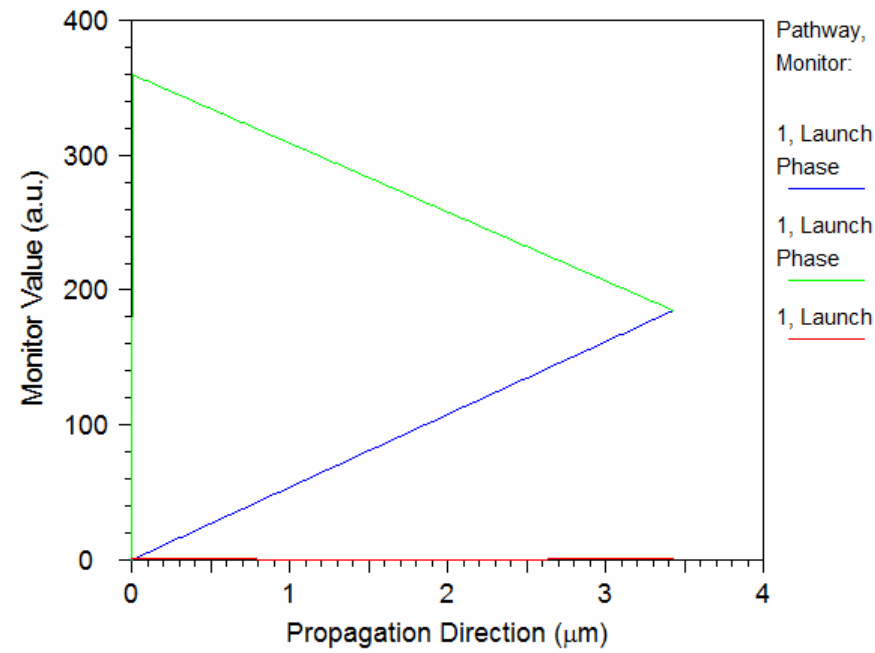

Figure 3: Zoomed in view of the monitor results.

The beating length agrees with the analytically calculated value `L_pishift`.

Areas for Further Exploration

- Run the simulation with the **Display Mode** set to *ContourMap (XY)* so you can see the polarization states couple back and forth.
- Perform a convergence study.
- For the wave plate example, simulate a similar device in a guided waveguide structure.
- *Use a different simulation tool for the wave-plate example!*

  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 7: Using Wide-Angle BPM

The basic BPM algorithm is fundamentally limited for paraxial propagation, which is typically considered to be up to 10 to 15 degrees from the optical axis. However, many waveguide structures require propagation at angles larger than this. As discussed in Section 2.D, BeamPROP incorporates several enhancements to the standard BPM algorithm for wide-angle propagation. This technique uses Padé Approximants to relax, to varying degrees, the paraxial approximation inherent in the BPM without sacrificing its numerical advantages.

This tutorial explores BeamPROP's wide-angle parameters necessary to account for wide-angle propagation. It will use a simple straight 2D waveguide that extends at an angle from the optical axis. Various simulations will be conducted to demonstrate the effects of wide-angle propagation and how these effects can be negated via the use of the Z grid size, the reference k value, and higher Padé orders. A convergence study will be performed to determine optimal values for these parameters.

The associated file for this example is

```
<rsoft_dir>\examples\beamprop\tutorial\tut07\wideangle.ind.
```

Device Layout

This structure will consist of a simple 2D tilted waveguide. The launch field will be the fundamental mode of the waveguide, and a monitor will be used to measure the transmission along the waveguide. Open the CAD interface and click the **New Circuit** button in the top toolbar. For the purposes of this example, the default values for these parameters are sufficient; click **OK** to create the design file.

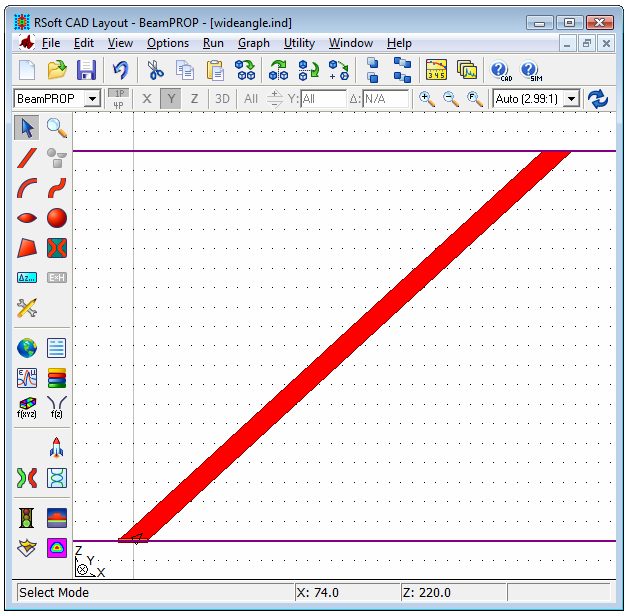

Figure 1: The completed structure in the CAD interface.

Creating the Waveguide

Draw a waveguide in the design, and right-click on it to open its Segment Properties dialog. Rather than define the properties of this waveguide exactly, it can be benificial to use variables so that the design can easily be changed later or to simplify a parameter scan. Click the **Symbols…** button and define a symbol `Angle` with a value of `20`. This symbol will be used to set the off-axis angle of the waveguide. Define the waveguide to have a starting vertex located at (x=0,z=0) and a ending vertex to have an X reference **Type** of *Angle* at an **Angle Value** of `Angle`, and a Z reference **Type** of *Offset* and an **Offset Value** of `200`. In this way, the waveguide will extend for 200 μm along Z and at an angle of `Angle` from the Z axis. Click **OK** to return to the CAD window. It might be necessary to click the **View Full** button in the top toolbar to see the entire structure.

Defining a Pathway, Monitor, and Launch Field

Define one pathway and add the entire structure (just one waveguide!) to it. Also define one monitor with a **Type** of *Slab Mode Power* and be sure to set the monitor **Tilt** option to *Yes*. Define a launch field with a **Type** of *Slab Mode* and make sure that launch **Tilt** is set to *Yes*. The default launch **Angle** default is sufficient since it is determined by the angle of the specified pathway.

Exploring Wide-Angle Propagation

Click the **Perform Simulation** icon to bring up a Simulation Parameters dialog, and click the **Advanced…** button and set the **Ref k Angle Value** is set to `Angle`, which is the optimal value for this case). Also note that the default **Pade Order** is *(1,0)*, which corresponds to standard paraxial BPM. We will experiment with this parameter later on, so the default value is sufficient for now.

Using the Default Parameter Values

Leave all the other parameters as default, and click **OK** again to run the simulation.

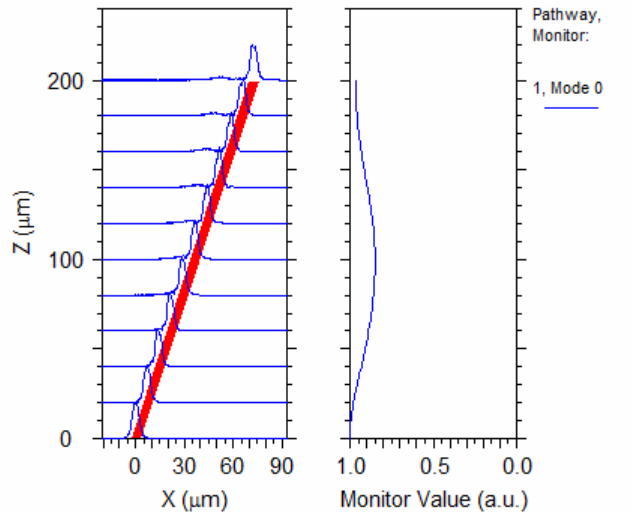

Figure 2: The simulation results using the default simulation parameters.

The dip in the middle of the monitor is unphysical and is due to the angled of the waveguide. It can usually be reduced below an acceptable level through the proper choice of **Pade Order** and X and Z **Grid Size**.

Convergence Study of Varying the X and Z Grid Sizes

Before changing the Padé Order, a convergence study on the X and Z grid sizes will be performed. First, run the simulation three times with the Z **Grid Size** set to `1.0`, `0.5`, and `0.25`. A combined plot of all three results is shown in Fig. 3. Note how the monitor oscillations decrease with the X grid size, and how the value `0.5` represents the point of diminishing returns. Further decreasing this simulation parameter has no significant effect on reducing the error.

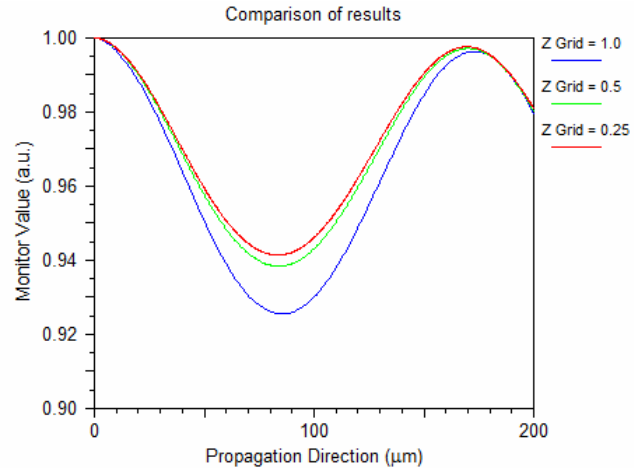

Figure 3: The simulation results found when varying the X **Grid Size**. Additional accuracy cannot be achieved through a further reduction of this parameter.

Next, experiment with reducing the X **Grid Size** while keeping the Z **Grid Size** set at `0.5`. Try values of `0.2`, `0.1`, `0.05`, and `0.025`. A combined plot of the results is shown in Fig. 4.

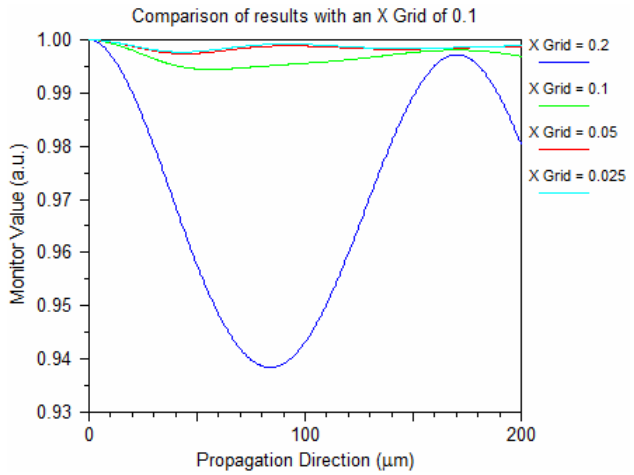

Figure 4: The simulation results found by reducing the X **Grid Size** to `0.1`.

Convergence Study of Increasing the Pade Order

As the X and Z grid sizes are reduced, the simulation results have converged to some degree. The next simulation parameter to look at is the Padé Order. Repeat the above simulations again, but set the **Pade Order** to *(1,1)* in the Advanced Options dialog. A typical result (X grid = 0.1, Z grid = 0.25) is shown in Fig. 5. Notice that the monitor oscillations are no longer visible on this scale.

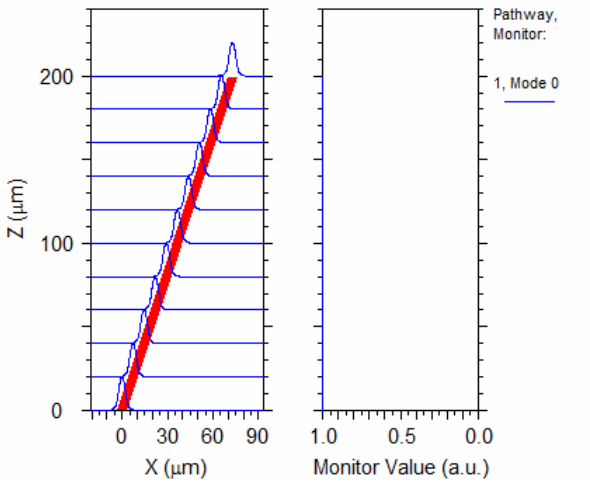

Figure 5: The simulation results obtained using a **Pade Order** of *(1,1)*.

Note that with the **Pade Order** set *(1,1)*, the values of the X and Z grid sizes can be made somewhat larger without sacrificing accuracy.

Closing Remarks

While the benefit of using a smaller grid size is obvious, it also lengthens the simulation time. Wide-angle problems typically are associated with large domains, making the simulation take longer. There is always a trade-off in numerical calculations of speed vs. accuracy. In such cases, choosing a higher **Pade Order** may give a bigger benefit than reducing the grid sizes. In other cases, choosing higher **Pade Order** might have no effect, and reducing the grid sizes may be required.

The concept of a convergence study, which was outlined in this example, is a methodical reduction of the numerical parameters to determine their optimal values so as to maximize accuracy and minimize computation speed. This example also illustrates the fact that many simulation parameters are related to each other, and the optimal value of one parameter depends on the values of the other parameters.

Areas for Further Exploration

- Use larger grid and step sizes with an increased Padé Order to see the tradeoff between Padé Order and the other simulation parameters.
- Don't set the **Ref K Angle Value** and see what type of convergence is possible by changing just the grid size, step size, and Padé Order.
- Investigate larger off-axis angles by changing the value of the variable `Angle`.

Tutorial 8: Modeling Gratings Using Bidirectional BPM

Many waveguide structures, such as Bragg Gratings, contain perturbations along the direction of propagation (z direction). The standard unidirectional BPM algorithm will fail to account for the coupling caused by these perturbations between the forward and backward propagating fields. To accommodate these situations, several algorithms for Bidirectional BPM are implemented in BeamPROP. These techniques take into account wave propagation in both directions and provide a means for determining the steady state field. This feature is currently only available for 2D problems, and it works best for, but is not restricted to, situations in which the index contrast (perturbation) is small and the paraxiality condition is satisfied (both forward and backward waves propagate along, or nearly along the z axis). The Bidirectional BPM feature is documented in Section 6.C.

This tutorial describes the simulation of a grating using the bidirectional feature that will entail a scan over the free space wavelength so as to determine the grating spectrum.

The associated file for this example is located in the

```
<rsoft_dir>\examples\BeamPROP\Tutorial\tut08\
```

directory.

Device Overview

Open the file `grating.ind` in the CAD interface.

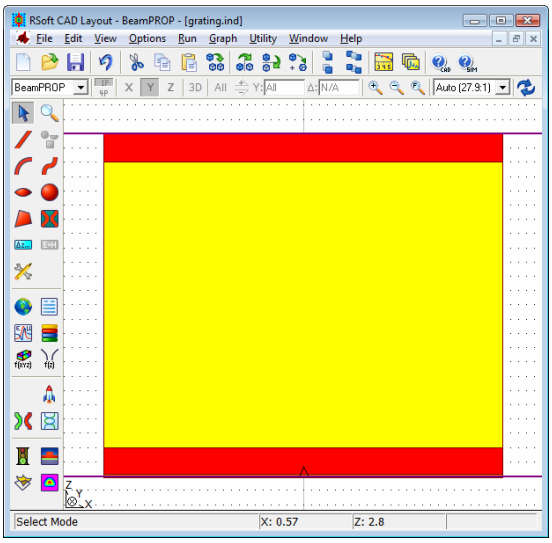

Figure 1: The grating structure in the CAD interface. The yellow segment has an index taper that varies the refractive index along the Z axis of the structure.

Note the following about the design:

- *Structure*

  The grating structure is defined by three segments: an input segment, a yellow grating segment, and an output segment.

- *Grating Definition*

  The grating segment has an index taper which varies the index along the propagation direction. Open the Segment Properties dialog and look at the **Index Difference** of the starting and ending vertices, as well as the **Index Taper** (*User 1*). Click the **Tapers…** button and look at the taper function. It is defined using the built-in `step()` function, and defines a rectangular grating profile with `M` periods. Also, note that the **Points** is set to `10*M+1` so that enough points (10 per period) are used to evaluate the grating function. See the CAD manual for more details on tapers and built-in functions. Finally, the length of the segment is set to `M*Period` which indicates that there are exactly `M` periods of the grating.

- *Final Index Profile*

  To view the grating structure, check the index profile. Click the **Display Material Profile** button in the left toolbar and note that the Z **Grid Size** is set to `Period/8` ; change the Z **Slice Grid** to the same value so that enough points are used to see the grating structure. Click **OK** to see the index profile. It is also possible to see a cross-section through the grating by right-clicking on the X axis at X=0 (see Fig. 2).

- *Pathways and Monitors*

  There is one pathway defined, and two monitors. Both monitors have a **Type** of *Slab Mode Power*, one with a **Component** of *Major* the other with a **Component** of *Major – Backward*. One monitor will measure the forward traveling field (as usual), the other will measure the backward traveling field computed by the bi-directional BPM feature.

- *Launch Field*

  The launch field is defined to be the slab mode of the input waveguide at a wavelength of 1.485 μm.

- *Bidirectional Options*

  The bi-directional option in the Advanced Options dialog has been enabled, and for purposes of this tutorial, the associated options use default values.

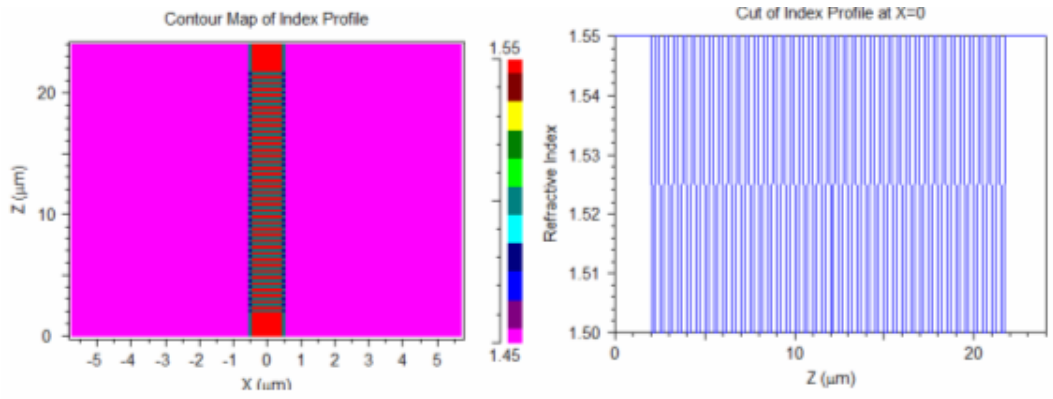

Figure 2: a) The index profile of the grating structure, and b) a cross-section at X=0.

Performing a Simulation

Click the **Perform Simulation** icon to open the Simulation Parameters dialog box. Set the **Slice Grid** Z to the default (if changed earlier to view the index profile) and click **OK** to start the simulation. The field will propagate back and forth along the waveguide until it reaches steady state.

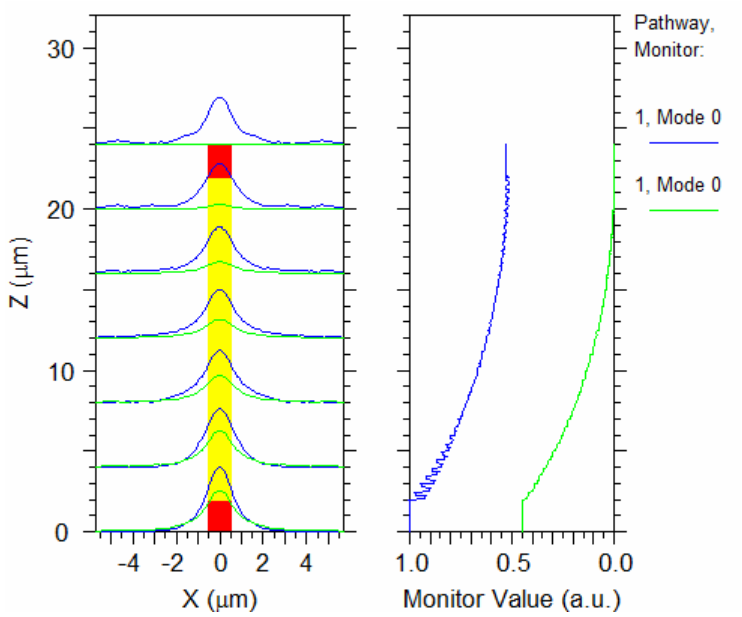

Figure 3: The simulation results at a single wavelength.

At the set wavelength, the transmitted (blue monitor at Z=24) and reflected power (green monitor at Z=0) are both about 50%.

Scanning over the Wavelength

It is frequently desirable to know the response of the grating at all the wavelengths of interest, which can be achieved, in BeamPROP, via a parameter scan.

Click the **Launch MOST** button in the left toolbar to open the MOST dialog. Note the settings that define a scan over the wavelength. Also note that a custom user-measurement has been created to correspond to the reflected field.

The reason a user-measurement is required is that the built-in monitor measurement records the last monitor value (in Z) which, because we are using bi-directional BPM, corresponds to the input field at the transmission plane which is an intermediate result and isn't important. In this case, we want the first monitor value which corresponds to the output field at the reflection plane, or the reflection. See the MOST manual for more information about creating user-measurements.

Click **OK** to start the scan. Each simulation will open on the screen and run, and when complete, the final result can be found.

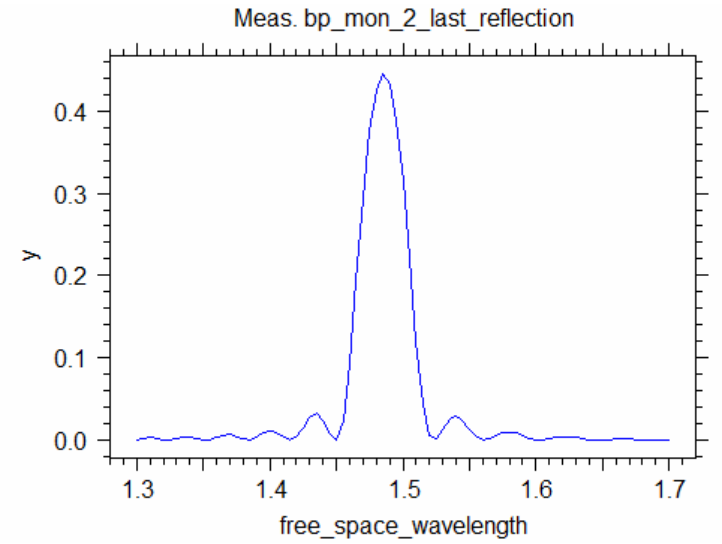

Figure 4: The completed scan results.

For this example, a sharp resonance exists at a wavelength of 1.485 µm.

Areas for Further Exploration

- Use the Symbol Table to change the value of `M` and rerun the scan. Observe the change in the width and height of the resonance.
- Use bi-directional BPM to simulate the effects of a waveguide/air interface.

Tutorial 9: Using the Ray-Tracing Interface

This tutorial illustrates the use of RSoft's Ray-Tracing Interface which converts data files from Zemax and CodeV formats to RSoft's field format and vice versa. Zemax and CodeV are registered trademarks of and can be purchased from ZEMAX Development Corporation (www.zemax.com) and Optical Research Associates (www.opticalres.com) respectively. This tutorial will not attempt to illustrate the use of these software packages; some familiarity is therefore assumed.

This tutorial uses BeamPROP as a representative simulation tool from RSoft's suite; any other simulation tool that shares the RSoft CAD interface could also be used. Familiarity with the RSoft CAD and BeamPROP is assumed.

The combination of ray-tracing simulation tools with field simulation tools is very useful. This is true for mutli-stage problems where one part of the system is more efficiently simulated by ray tracing tools and electro-magnetic propagating tools become more useful in other stages. This tutorial will illustrate this concept through the modeling of the simple case of coupling light from a ball lens into a small silica fiber. First, the output from the will first create output information from the ray tracing software. Then we will convert these files into field files that can be used for a BeamPROP simulation. This process will be repeated for both CodeV and Zemax.

Using CodeV

This section of the tutorial illustrates the conversion of CodeV data into an RSoft field for use in a BeamPROP propagation.

Extracting Data from CodeV Example

The ball lens example we will use for this example is one of the standard CodeV examples. Open the CodeV package, select the 'Open File' icon, move to the lens example directory (by default `c:\codev##\lens` where `##` is the version number), and open the file `ballcouple.len`. Click the 'Quick 2D –Labeled' icon to view the ball lens as shown in Fig. 1.

The data needed from CodeV for the conversion to RSoft's format is the intensity and wavefront information after the rays have propagated through the lens. Select 'Analysis/Diagnostics/Pupil Map' and select the 'Output Controls' tab to create the wavefront file (in CodeV's INT file format). Type `wavefront.wfr` in the 'Write PMA Output to Interferogram File' box. We recommended creating a new folder as your working directory and save the file to this folder. Click 'OK' to create the file.

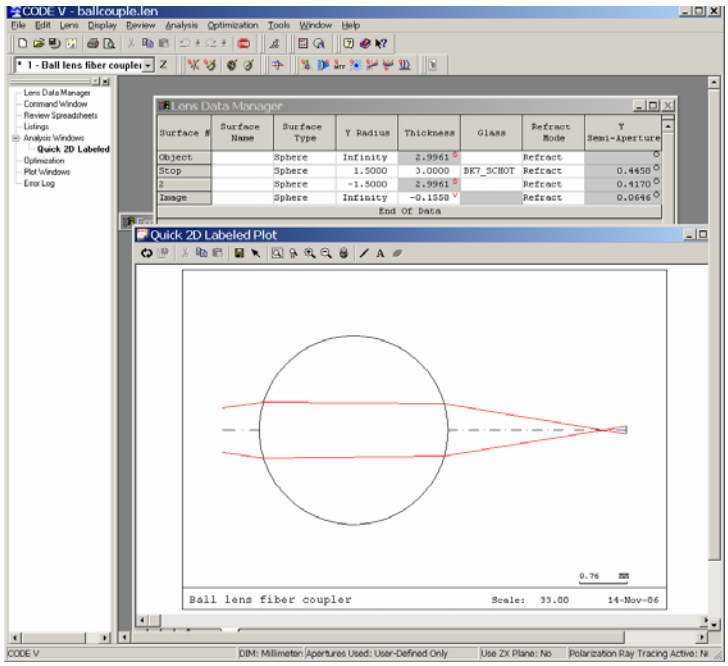

Figure 1: Ball coupler lens as seen in CodeV.

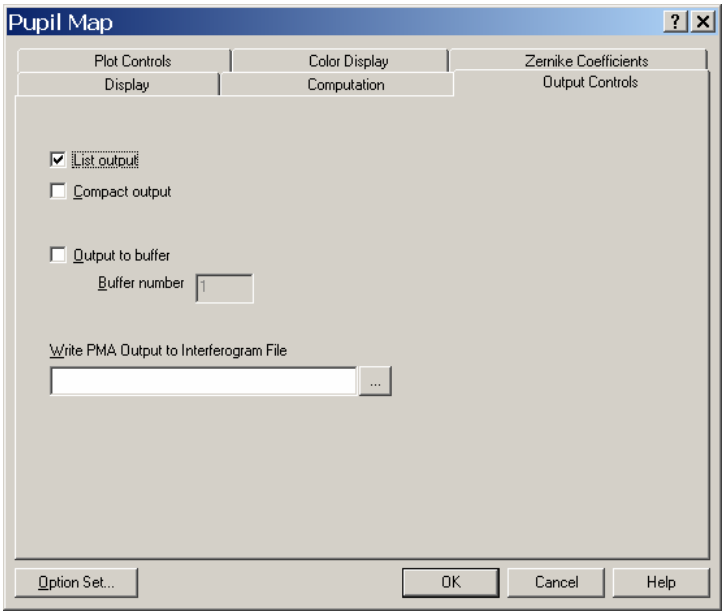

Figure 2: Pupil Map Output Controls Setting

Next, select 'Analysis/Diagnostics/Pupil Map' again and select the 'Display' tab. Set 'Pupil Output Selection' to `Intensity` and select the 'Output Controls' tab again. Type `intensity.fil` in the 'Write PMA Output to Interferogram File' box. Make sure to save this file to your working directory. Click 'OK' to save the file.

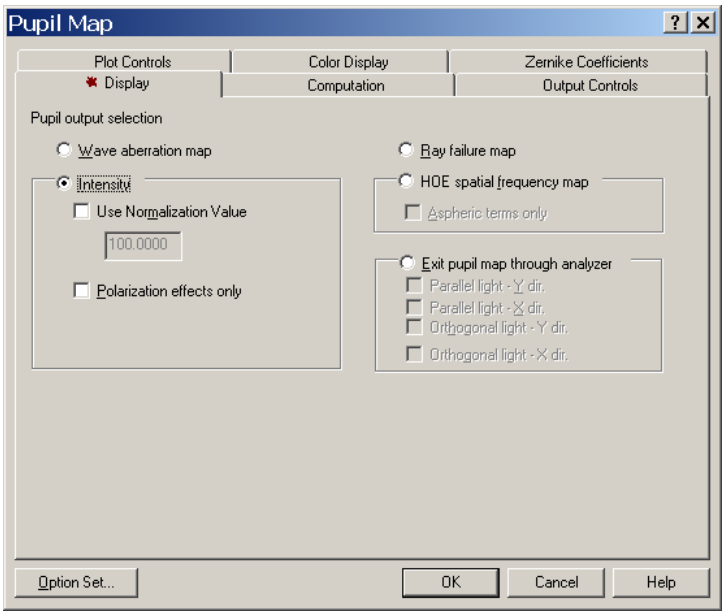

Figure 3: Pupil Map Display Settings

Once complete, the CodeV window should have the plots shown in Fig. 4.

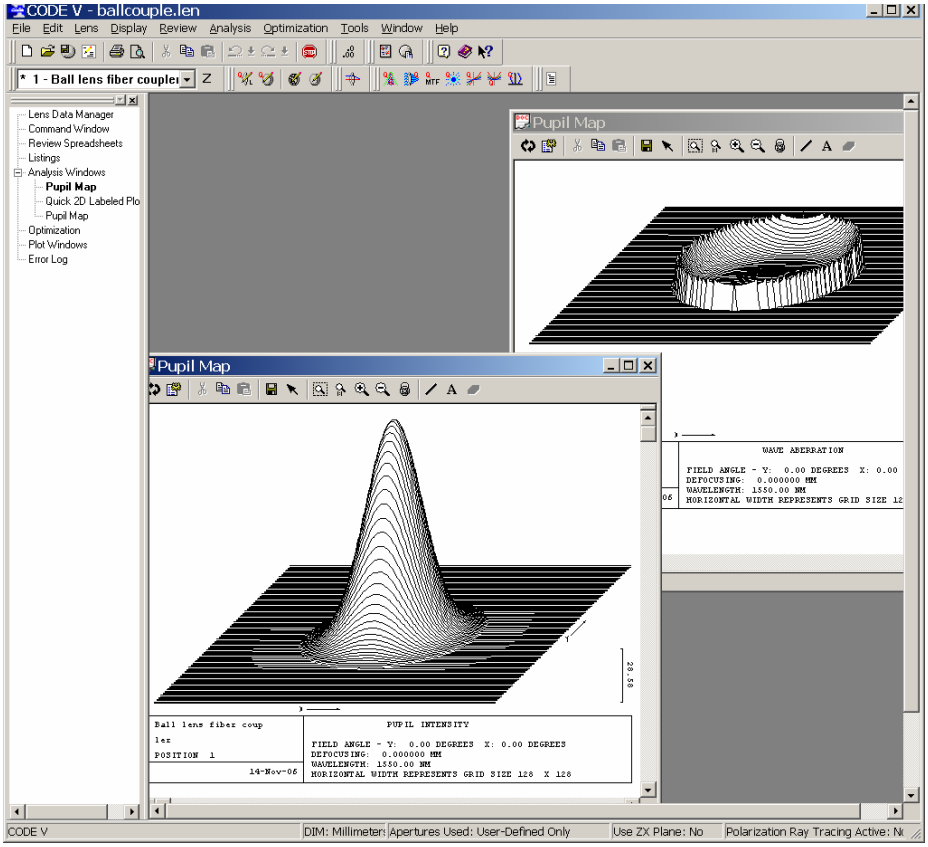

Figure 4: Intensity and Wave Aberration Output

Creating the RSoft .ind File

Open the RSoft CAD and click the **New Circuit** icon. Set **Simulation Tool** to *BeamPROP*, the **Model Dimension** to *3D*, the **Free Space Wavelength** to `1.55`, the **Index Difference** to `0.5`, and the **Component Width** and **Component Height** to `1`. Click **OK** to create the file.

The structure will consist of a simple fiber component 100 μm long. Draw a straight waveguide component and right-click on it to set its properties. Set the starting vertex (X, Y, Z) to (`0`, `0`, `0`) and set the ending vertex to (`0`, `0`, `100`) offset from the starting vertex. We will also create a simple monitor to measure the power propagating in the fiber. Click the **Edit Pathways** icon, press the **New** button, and then select the fiber segment. Press the **Monitors** button and then the **New** button in the Monitor Properties dialog. Set the **Type** to *Total Power* and click **OK** in this dialog and left toolbar to return to the CAD.

Save the file to your working directory.

Converting the CodeV Output Files to RSoft's Field Format

The launch field we will use in the BeamPROP simulation will be created by converting the CodeV output. Open RSoft's Ray-Tracing Converter using the Utility menu item in the CAD.

Make sure the two CodeV files and the `.ind` design file are in the same directory before proceeding.

Select *Code-V to RSoft* from the top menu and enter the CodeV `.fil` and `.wfr` files. Enter an **Output Prefix** of `bpfield` as shown in Fig. 5 and click **OK** to convert the files.

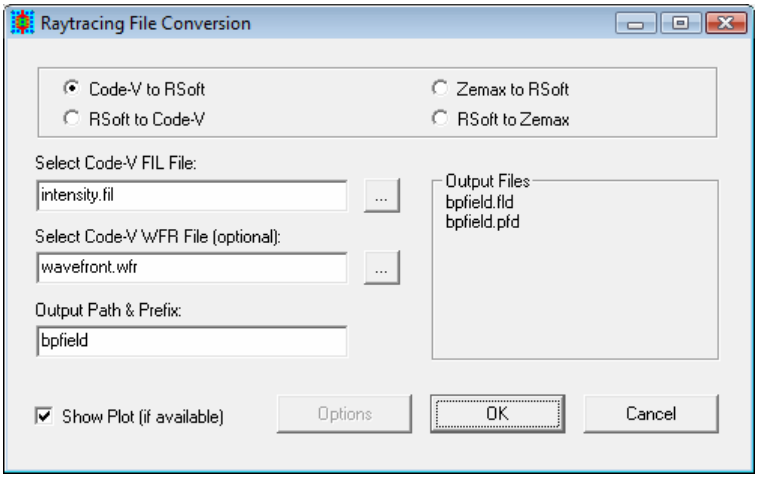

Figure 5: Converting CodeV data using RSoft's Ray-Tracing Utility.

A plot of the converted field (Fig 6a) will open automatically in WinPLOT. This data is complex valued and includes phase information. To view the phase (Fig 6b) click the **View Editor** button in the top WinPLOT toolbar and add the command `/zphase` to the bottom of the listed commands. Click the **View Plot** button to return to the plot.

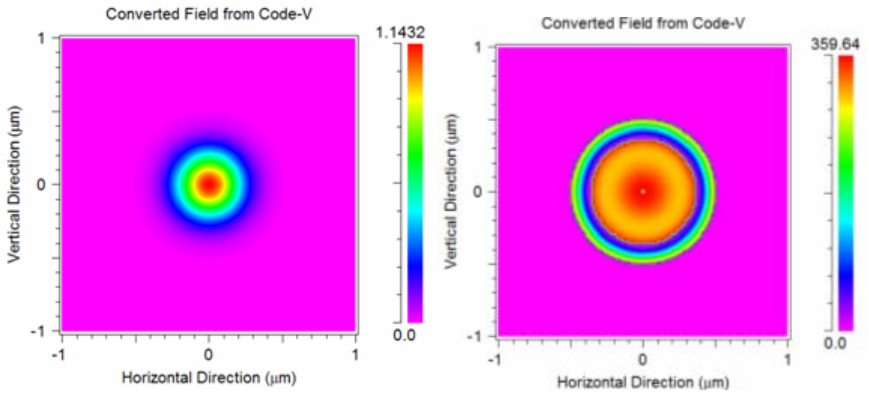

Figure 6: The converted data: a)amplitude, and b) phase.

Setting up the Launch Field

We now have to set the converted file as the launch field in BeamPROP. Click the Edit Launch button in the left CAD toolbar and set the **Type** to *File* and set **Input File Spec** to `bpfield` and `fld` (**Prefix** and **Extension**).

Running the Simulation

Click the **Perform Simulation** icon in the left CAD toolbar and set the **Display Mode** to *Contour Map (XZ)* and click OK to start the simulation.

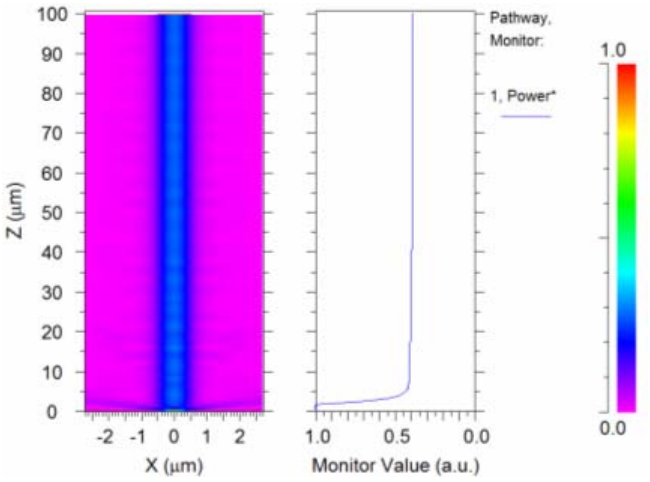

Figure 7: Simulation results using a launch field generated by CodeV and converted to the RSoft field format.

The results shown in Fig. 7 show that this particular structure achieves approximately 39% coupling.

Using Zemax

This section of the tutorial illustrates the conversion of ZEMAX data into an RSoft field for use in a BeamPROP propagation.

Extracting Data from ZEMAX Example

Open the ZEMAX software package, select File/Open from the top menu, move to the `Samples/Physical Optics` folder, and open the file `Fiber Coupling.zmx`.

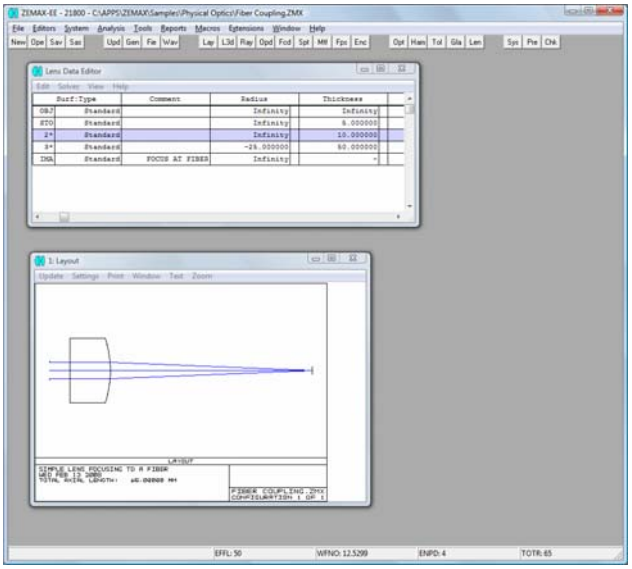

Figure 8: ZEMAX setup for fiber coupling lens

To view the beam at the focal point, select 'Analysis/Physical Optics/Physical Optics Propagation'.

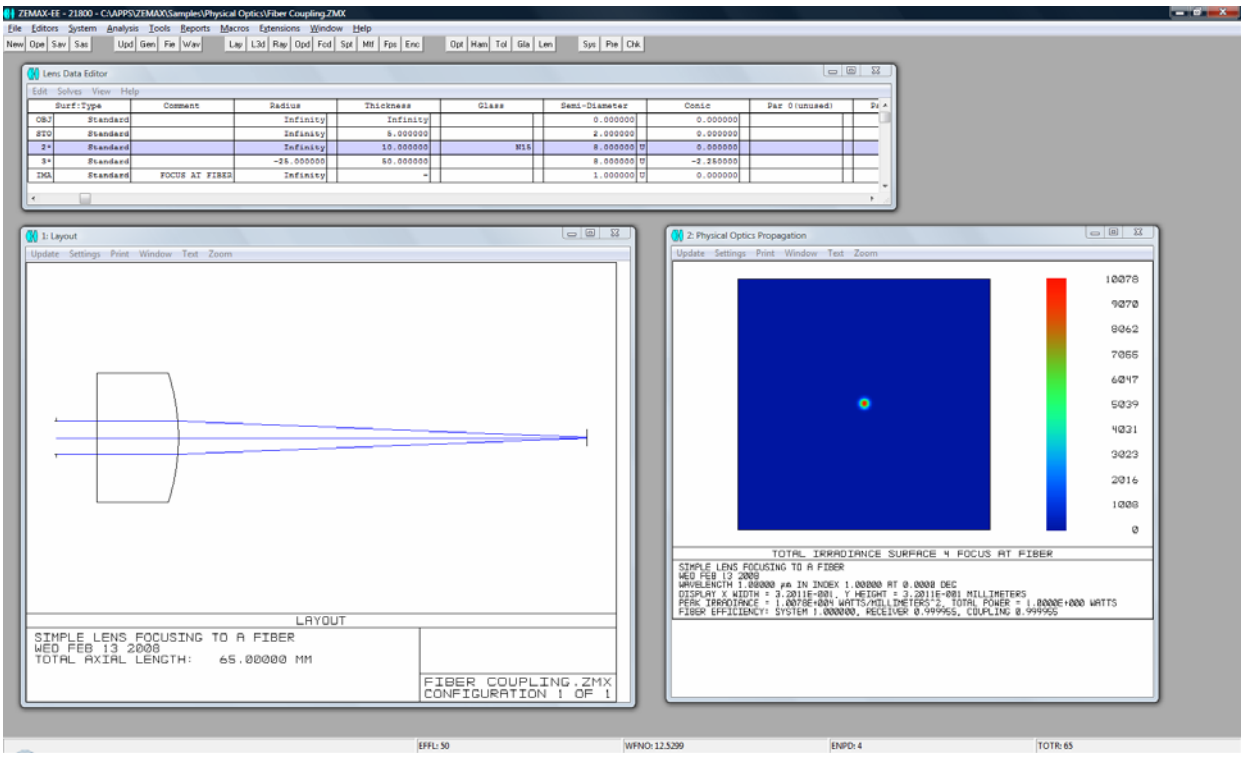

Figure 9: ZEMAX screen with field pattern.

To instruct ZEMAX to output the necessary files for conversion, click the 'Settings' option on the top of the physical optics propagation window, select the 'Display' tab, and click on the 'Save Output Beam To:' option. Set the name of the file to be `Fiberoutput.zpf` and click 'OK'.

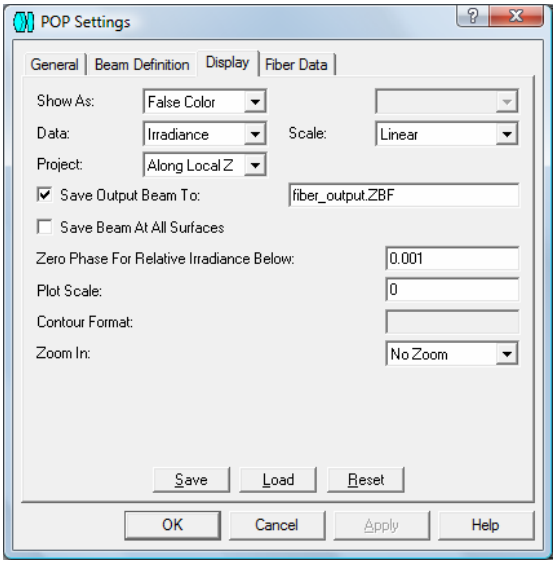

Figure 10: Display settings for physical optics propagation.

The output files are usually saved in the `ZEMAX/POP/Beamfiles` directory. We recommended creating a new folder as your working directory and save the file to this folder. Move the `Fiberoutput.zpf` file just created to your working directory.

Creating the RSoft .ind File

Open the RSoft CAD and click the **New Circuit** icon. Set **Simulation Tool** to *BeamPROP*, the **Model Dimension** to *3D*, the **Background Index** to `1.49`, the **Index Difference** to `1.4934-background_index`, and the **Component Width** and **Component Height** to `10`. Click **OK** to create the file.

The structure will consist of a simple fiber component 10,000 μm long. Draw a straight waveguide component and right-click on it to set its properties. Set the starting vertex (X, Y, Z) to (`0`, `0`, `0`) and set the ending vertex to (`0`, `0`, `10000`) offset from the starting vertex. We will also create a simple monitor to measure the power propagating in the fiber. Click the **Edit Pathways** icon, press the **New** button, and then select the fiber segment. Press the **Monitors** button and then the **New** button in the Monitor Properties dialog. Set the **Type** to *Total Power* and click **OK** in this dialog and left toolbar to return to the CAD.

Save the file to your working directory.

Converting the ZEMAX Output Files to RSoft's Field Format

The launch field we will use in the BeamPROP simulation will be created by converting the ZEMAX output. Open RSoft's Ray-Tracing Converter using the Utility menu item in the CAD.

Make sure the ZEMAX file and the `.ind` design file are in the same directory before proceeding.

Select *Zemax to RSoft* from the top menu and enter the ZEMAX `.zbf` file. Enter an **Output Prefix** of zemax_`bpfield` as shown in Fig. 11 and click **OK** to convert the files.

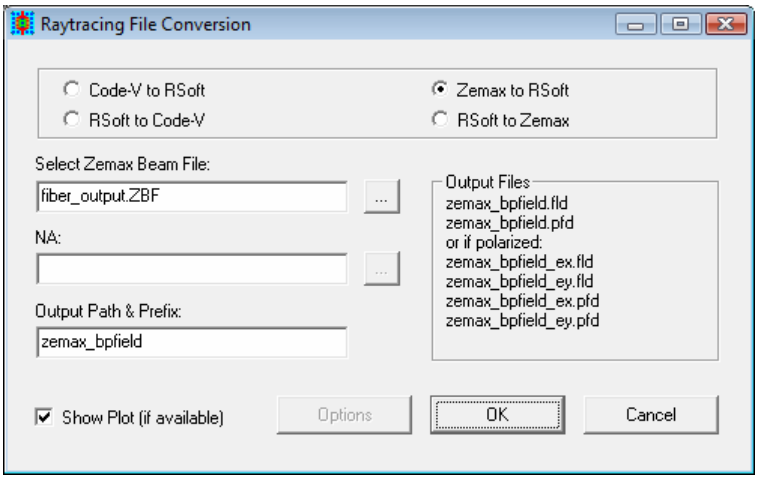

Figure 11: Converting CodeV data using RSoft's Ray-Tracing Utility.

A plot of the converted field (Fig 12a) will open automatically in WinPLOT. This data is complex valued and includes phase information. To view the phase (Fig 12b) click the **View Editor** button in the top WinPLOT toolbar and add the command `/zphase` to the bottom of the listed commands. Click the **View Plo**t button to return to the plot.

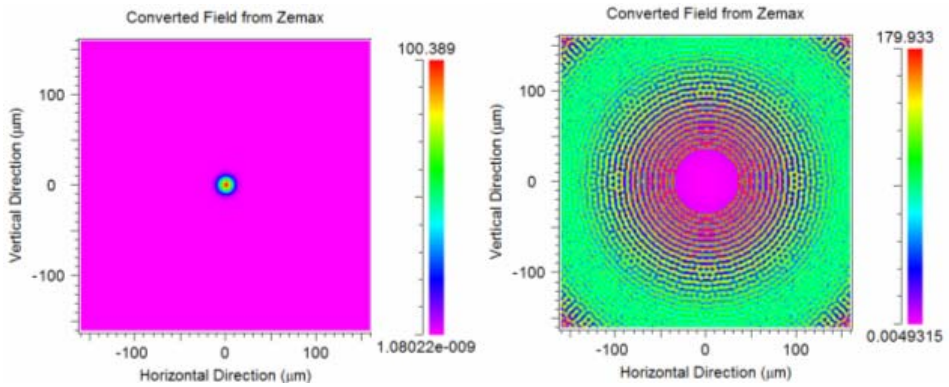

Figure 12: The converted data: a)amplitude, and b) phase.

Setting up the Launch Field

We now have to set the converted file as the launch field in BeamPROP. Click the Edit Launch button in the left CAD toolbar and set the **Type** to *File* and set **Input File Spec** to `zemax_bpfield` and `fld` (**Prefix** and **Extension**).

Running the Simulation

Click the **Perform Simulation** icon in the left CAD toolbar and set the **Display Mode** to *Contour Map (XZ)* and click OK to start the simulation.

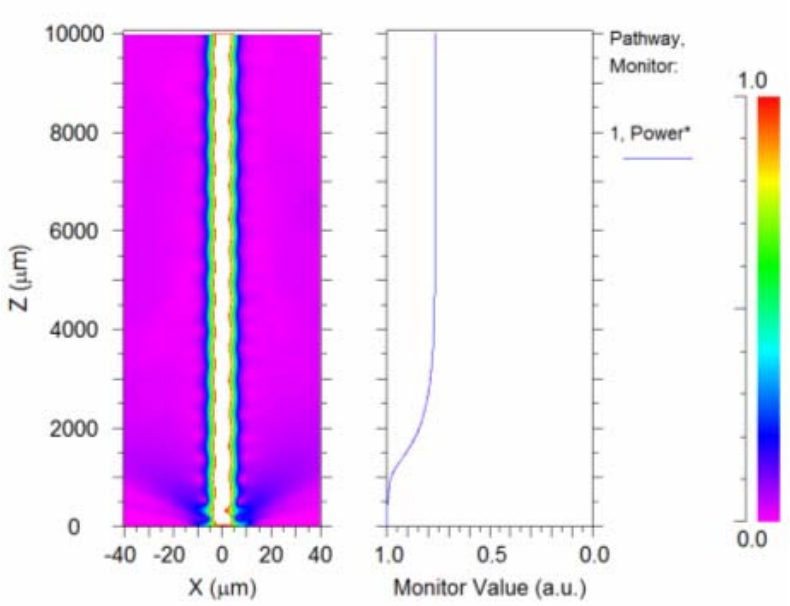

Figure 13: Simulation results using a launch field generated by ZEMAX and converted to the RSoft field format.

The results shown in Fig. 13 show that this particular structure achieves approximately 77% coupling.

Tutorial 10: Designing a Mode Converter

One of the primary advantages of the BPM algorithm is the simulation of relatively large photonic integrated circuits. In this example a mode converter will be studied to measure the coupling efficiency. The mode converter will be designed to convert a large fiber mode to that of a much smaller SOI waveguide. This is accomplished through the use of tapers which slowly convert the modes so that a very low loss and high conversion can be achieved.
The associated file for this is tutorial is located in the

```
<rsoft_dir>\examples\BeamPROP\tutorials\tut10\
```

directory.

Device Overview

Open the file `converter.ind` in the CAD interface.

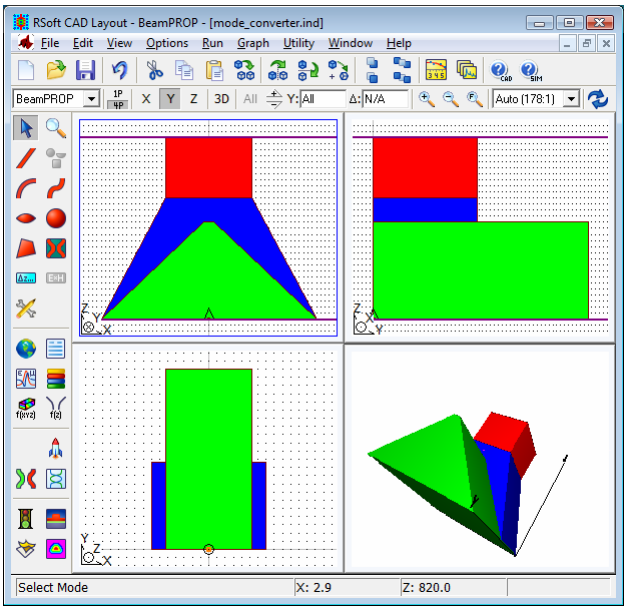

Figure 1: The mode converter as seen in the CAD interface.

Note the following about the structure:

- *Structure*

  The structure is defined using the *Multilayer* **3D Structure Type**. Only one layer table is defined, and is used for the background region (with a non-zero slab height) as well as for each segment. Several overlapping segments are used to produce the desired geometry. See Fig. 2 for several cross-sections along the structure.

- *Tapers*

  Two segments have width tapers to help adiabatically convert the input mode to the output mode. These segments are defined with a different starting and ending vertex and a linear taper function.

- *Launch Field*

  The launch field consists of the mode of the input cross-section and is shown in Fig. 3a.

- *Pathways and Monitors*

  One pathway is defined, and three monitors are defined to measure the power in the input mode (`input_mode_ex.m00`), output mode (`output_mode_ex.m00`), and the total power in the simulation domain. It is expected that the power in the input mode be at a maximum near the input of the structure, and then power in the output mode be at a maximum near the output of the structure. The total power monitor will show how much field is lost to radiation as the field propagates through the structure.

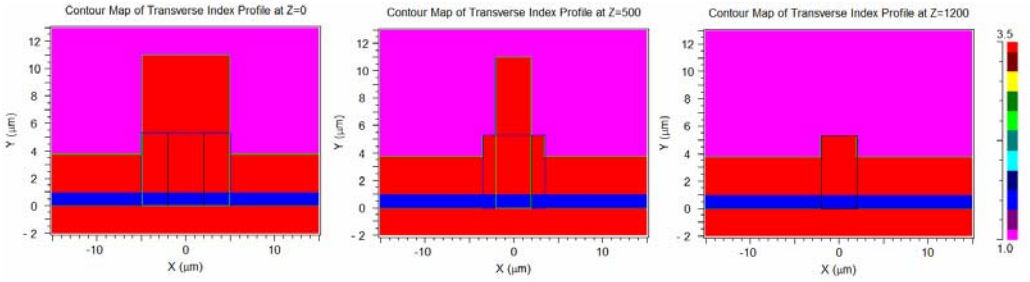

Figure 2: The refractive index profile at various positions along the structure: a) Z = 0 µm (input), b) Z = 500 µm, and Z = 1200 µm (output). The SOI structure can clearly be seen.

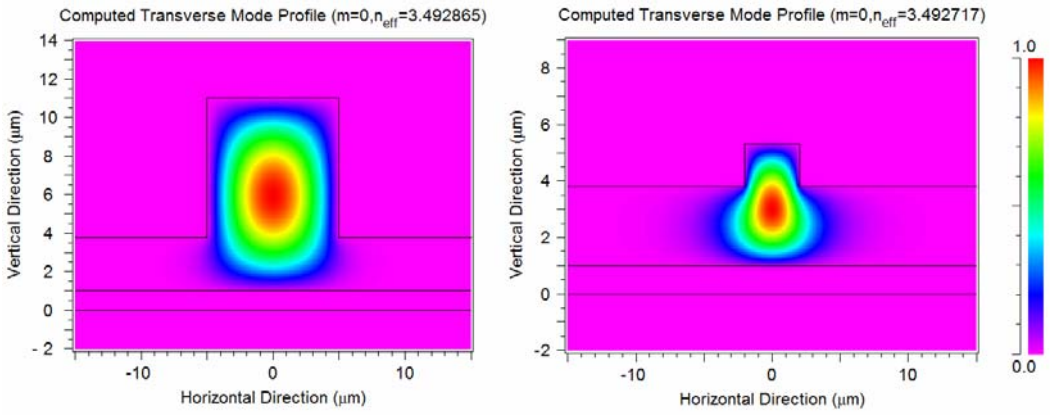

Figure 3: The E_x mode profiles of the a) input mode, and b) output mode. These modes can be seen by opening the files input_mode_ex.p00 and output_mode_ex.p00 respectively.

Performing a Simulation

Click the **Perform Simulation** button in the left toolbar and click OK to start the simulation.

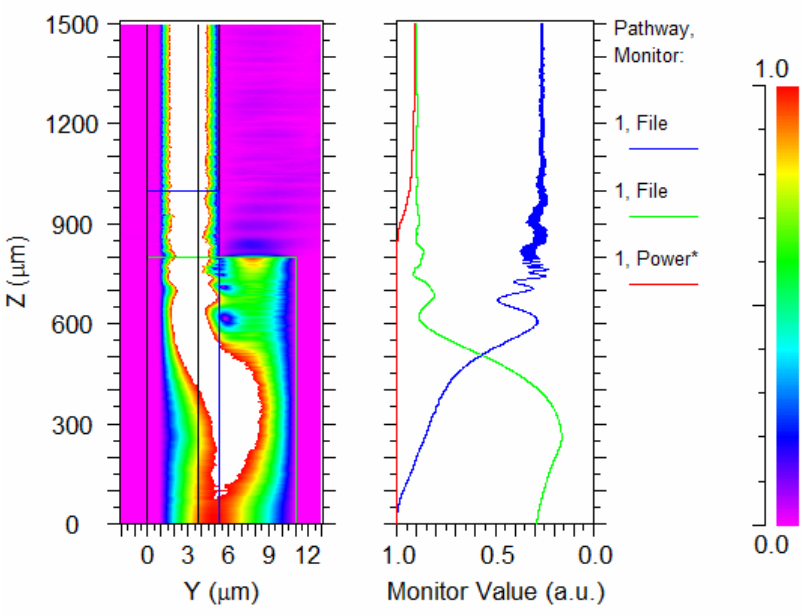

Figure 4: Simulation Results of Mode Converter

The left portion of the simulation results (shown in Fig. 4) shows the YZ cross-section of the amplitude of the field in the structure. The right portion shows the monitor results. The blue line shows the power in the input mode, which, since the launch field was that mode, has zero insertion loss. The green line shows how much of the field is converted into the output mode. As the field propagates through the structure, this increases as expected. The monitor results can be looked in more detail by opening the monitor results.

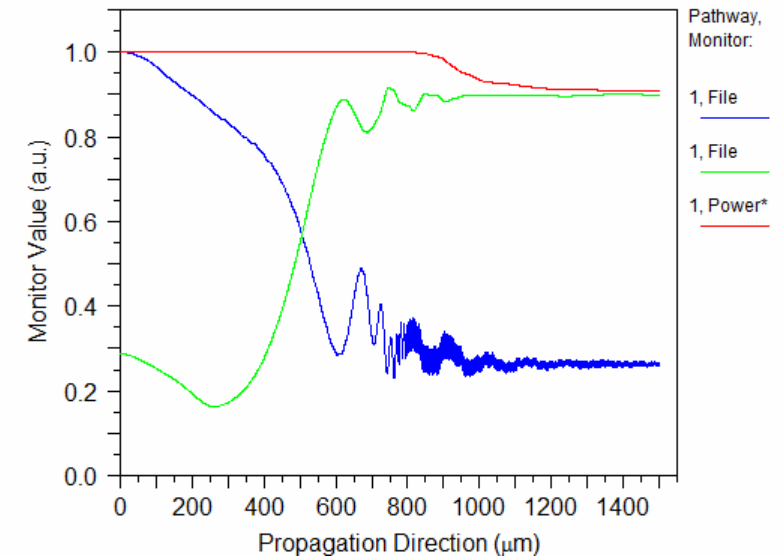

Figure 5: The monitor results.

Areas for Further Exploration

- Performa full convergence study on the simulations parameters such as the X, Y, and Z grid sizes. Keep in mind that the modes might have to be recomputed.

Tutorial 11: Using the Simulation Region Feature

The following tutorial explores the use of a simulation region to greatly reduced simulation time while obtaining accurate simulation results. A simulation region allows the user to locally control select simulation parameters such as the Z grid size. This type of calculation is useful for structures where certain regions along Z require a small grid size where others do not. See Section 6.D for more details on the use of Simulation Regions.

The associated files for this example are located in the

```
<rsoft_dir>\examples\BeamPROP\Tutorial\tut11\
```

directory.

Structure Overview

Open the file `air_gap.ind` in the CAD interface and note the following about the design:

- *Structure*

  This represents a silica waveguide (in red) with an air-gap in the middle (in gray).

- *Pathways and Monitors*

  One pathway is been defined (it includes just the two waveguide segments), and one monitor is defined to measure the power in the fundamental mode of these waveguides.

- *Launch Field*

  The launch field is the fundamental mode of the silica waveguide at a wavelength of 1.55 μm.

Performing a Simulation

Three simulations will be performed to understand how the simulation parameters affect the simulation results.

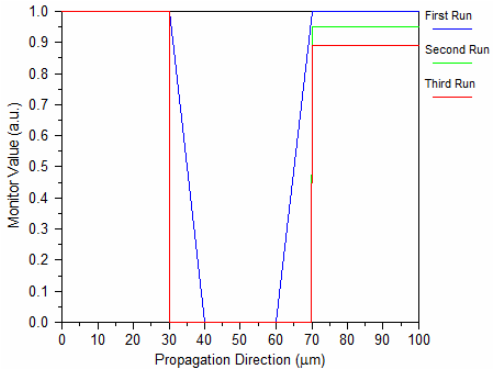

Figure 1: The simulation results of the transmission through the air gap. Results from all three simulation runs are shown.

- *First Run*

In this simulation run, all the simulation defaults will be used. The results (shown in Fig. 1) show almost 100% transmission is found. This is because the **Reference k** (`kbar`) value is not optimized for each section. The default value for `kbar` is β which his roughly equal to `1.445*k0`. In the air gap, however, the best choice for `kbar` is `1.0*k0`.

- *Second run*

  In the second simulation run, change the Z **Grid Size** to 0.1 μm, and set the Reference k (in the Advanced Options dialog) to an average value of `(1.445+1)*k0`. Also set the **Pade Order** to *(1,1)*. The results for this simulation are shown in Fig. 1.

- *Third run*

  In the third simulation run, change the Z **Grid Size** to `0.005` μm and set the **Pade Order** to *(3,3)*. The results for this simulation are also shown in Fig. 1.

When viewed together, it can be seen that the simulation results are converging to a more accurate value. However, the simulation time has also increased.

Performing a Simulation with a Simulation Region

Now the same structure will be studied but with a simulation region in the air gap to locally control the **Reference k** value and **Pade Order**. Open the file `air_gap_sr.ind` and note the Simulation Region (the light blue segment) that sits in the air gap.

Right click on the simulation region to see its properties, and click **Local Symbols…** button and note the following settings:

Variable Value	Value
`free_space_wavelength`	`1.55`
`k0`	`(2*pi)/free_space_wavelength`
`kbar`	`k0`
`pade_order`	`1`
`step_size`	`1`

This symbol table is local only to this Simulation Region, and can therefore have different values for these parameters than the rest of the file. This file has been set up to use all the default parameters like just like the first run above.

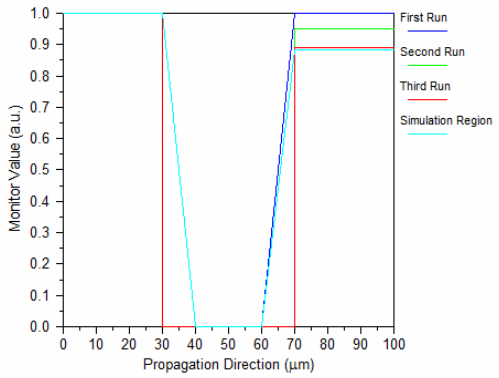

Figure 3: Combined results from the Simulation Region run with the previous three runs.

The results found when using the simulation region with the same numerical parameters as the first run without the region are very close. As an extended exercise, perform a convergence study with and without the simulation region to show these results can be made equal, but that the use of the simulation region drastically reduces the simulation time.

Areas for Further Exploration

- Run a parameter scan on either example to see the different responses for each case versus a design parameter.
- Try creating a similar example that has a lens object in free-space. Use a Simulation Region to locally use a larger step size and the FFT-BPM algorithm in the air regions.

9

Mode Solving Tutorials

This chapter contains several tutorial examples which illustrate the use of BeamPROP's BPM-based mode solvers. Corresponding to each example there is one or more `.ind` and/or data files, which are located in this folder:

```
<rsoft_dir>\examples\BeamPROP\tutorial\Mode_Calculation
```

It is recommended that users work through the simple tutorial in the CAD manual before starting these tutorials. Users should also work through the first two mode tutorials in this chapter before proceeding to the others. It is also recommended that users read through the mode-solving technical background found in Chapter 2.H, and the descriptions in Chapter 5.

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 at info@rsoftdesign.com and we will assist you in finding a solution if possible.

Mode Tutorial 1: Using the Iterative Method

This tutorial discusses the basic use of the iterative method to compute modes in both 2D and 3D. Convergence and proper use of the mode simulation parameters will also be discussed.
The iterative method, which is based on imaginary distance BPM, is very useful for computing modes of well-confined devices. This method is only valid for non-lossy structures. The technical background for this method can be found in Chapter 2.H, and the use of this method is described in Chapter 5. Structures with lossy materials, or that have leaky modes, require the use of the correlation method described in Mode Tutorial 2.

Modes of a Simple 2D Slab

We will create a simple 2D slab in this section and then compute the modes it supports.

Creating the Structure

Click the **New Circuit** button in the CAD window and set the following parameters and then click **OK**.

Parameter	Value	Description
Free Space Wavelength	`1.55`	Sets the simulation wavelength.
Background Index	`1.0`	Represents the real refractive index of the background material where no structure has been defined.
Index Difference	`0.2`	Represents the default difference between a component and the background material; for this case the index will be 1.2.

In the CAD, draw a straight waveguide segment from (X,Z) = (0,0) to (0,1000). The structure length, in this case 1000 μm, need only be long enough to allow the mode solver to converge. In this case a warning will be displayed and the user should lengthen the structure and retry the simulation.
The index profile of the structure can be computed by click the **Display Material Profile** in the CAD.

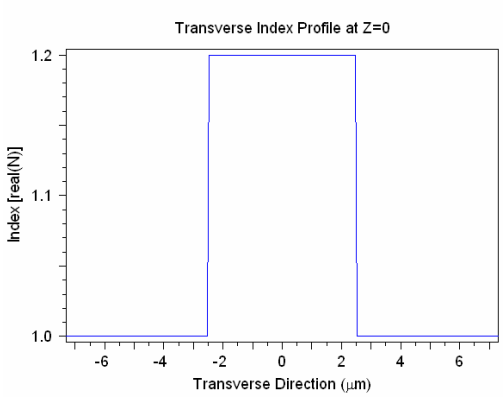

Figure 1: The computed index profile of the simple 2D slab.

Launch Settings

The use of an appropriate launch field is critical when computing modes. It is important to excite all the modes of the structure in order to properly compute them. If a particular launch field does not excite a particular mode, the mode solvers will not compute it. As a general rule, a offset Gaussian launch field is usually sufficient to excite a large continuum of modes, including any anti-symmetric modes.
To use an offset Gaussian launch field, click the **Edit Launch Field** icon in the CAD window. Set the **Type** to *Gaussian* and the **Position X** to be `0.5`. Press **OK** to return to the CAD.

Computing the Fundamental Mode

To compute the fundamental mode, click the **Compute Modes** icon on the left toolbar in the CAD window. This dialog allows the user to set the computation parameters for the mode calculation. The mode method, which is the iterative method described in this tutorial by default, can be chosen via the

Mode Options… button. For this example use the default settings and press **OK** to start the calculation. When the simulation completes, the computed fundamental mode profile will be displayed.

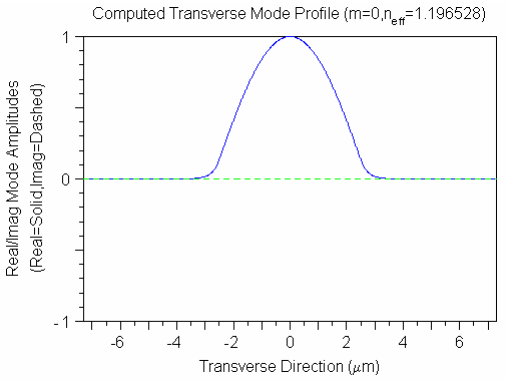

Figure 2: Fundamental mode of a simple 2D slab waveguide. Note that the mode number and effective index are displayed in the top right corner of the plot.

While the effective indices are conveniently displayed in the top of every mode plot, this may not be useful for post-processing purposes. The text file `<prefix>.nef` , where `<prefix>` is the **Output Prefix**, contains a list of the modes found and corresponding effective index values.

Computing Higher Order Modes

To compute higher order modes, open the Mode Calculation Parameters dialog and click **Mode Options** and select *All Modes*. Click **OK**, set the **Output Prefix** to `2D`, and click **OK** to start. When the simulation has completed, the fundamental mode will be displayed in the simulation window. To view any of the modes computed, including higher order modes, click the **View Graphs** icon in the CAD window. The plot files for the computed modes will be named `2D.p##`, where `##` denotes the mode number. These files can be opened to view the computed modes.

Multiple files are output for every mode computed: the `<prefix>_<comp>.p##` files are text files that contain plotting commands, and the `<prefix>_<comp>.m##` files are text files that contain the actual mode data where `<prefix>` is the **Output Prefix** given, `<comp>` is the field component (if applicable), and `##` is the mode number. These files, along with the file `<prefix>.nef` described earlier can be opened in any text editor for further processing.

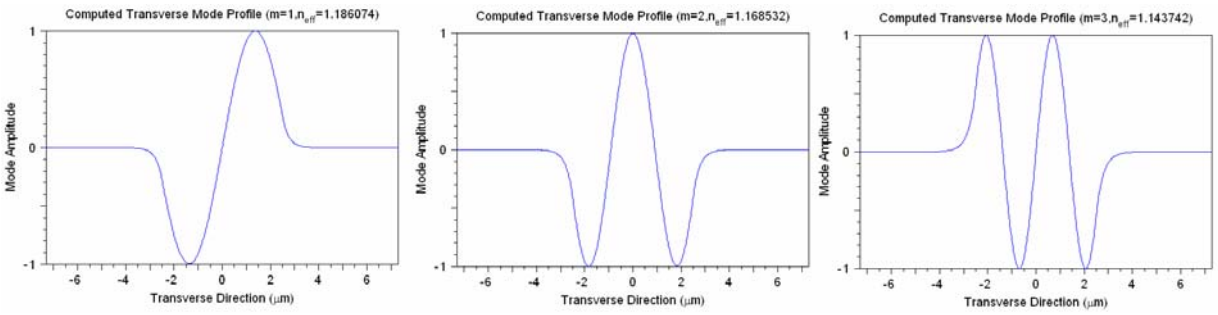

Figure 3: Some of the higher order modes supported by a simple 2D slab waveguide.

Modes of a Simple 3D Buried Channel

We will create a simple 3D buried channel in this section and then compute the modes it supports. We will also perform a convergence study to check the accuracy of the results as well as explore several polarization options.

Creating the Structure

Press the **New Circuit** icon, and set the following parameters:

Parameter	Value	Description
Model Dimension	*3D*	This indicates that a 3D simulation should be performed.
3D Structure Type	*Channel*	Indicates that the default structure type should be channel.
Free Space Wavelength	`1.55`	Sets the simulation wavelength.
Background Index	`1.0`	This represents the real refractive index of the background material where no structure has been defined.
Index Difference	`0.5`	This represents the default difference between a component and the background material. In this case this means that the structure will have an index of 1.5.
Component Width	`2`	This represents the default width of a component in μm.
Component Height	`1`	This represents the default height of a component in μm.

Once these settings are made, click **OK** to continue.
In the CAD, draw in a straight waveguide segment from (X,Z) = (0,0) to (0,1000). The structure length, in this case 1000 μm, need only be long enough to allow the mode solver to converge. In this case a warning will be displayed and the user should lengthen the structure and retry the simulation.
The index profile of the structure can be computed by click the **Display Material Profile** in the CAD.

Launch Settings

As with the 2D case, the launch field must excite all the modes of the structure, and so an offset Gaussian launch will be used. Click the **Edit Launch Field** icon in the CAD window and set the **Type** to *Gaussian*, the **Position X** to be `0.1`, and the **Position Y** to be `0.05`. Press **OK** to return to the CAD.

Computing the Fundamental Mode

To compute the fundamental mode, click the **Compute Modes** icon on the left toolbar in the CAD window. For this example use the default settings and press **OK** to start the calculation. When the simulation completes, the computed fundamental mode profile will be displayed.

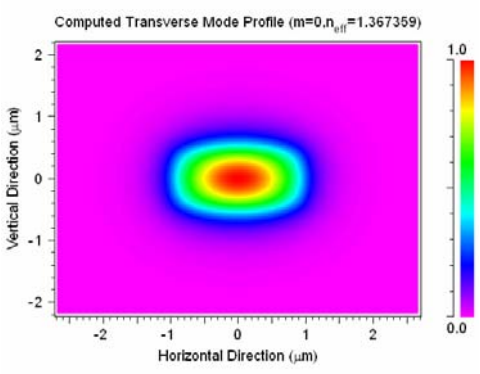

Figure 4: Fundamental mode of a simple 3D buried channel waveguide. Note that the mode number and effective index are displayed in the top right corner of the plot.

Performing Convergence Studies

This section will study the effect of several simulation parameters on the computed mode. As an example, it will use the simple 3D channel structure described above. For each parameter, a convergence study will be done that illustrates how the parameters affect the simulation accuracy.

Studying the Effect of the Z Grid Size

Since the BPM algorithm integrates along the Z axis, the Z grid size has a different effect than the X and Y grid sizes. A convergence study can be used to ensure that a accurate result is obtained. Generally, a smaller Z grid size can provide greater accuracy and faster convergence, as well help provide a proper solution. However, a small grid size can result in a longer simulation. It is critical to perform a convergence study on the Z grid size to optimize the tradeoff between speed and accuracy when computing modes via the iterative method.

Rerun the mode computation for the simple 3D buried channel structure for Z **Grid Size** values of `1`, `0.5`, `0.25`, `0.125` and note the effective indices found for each case. The results you find should be similar to those in the following table:

Z Grid	Effective Index	Comments
`1`	`1.293697`	The simulation will not converge. This can be fixed by lengthening the structure or by reducing the Z grid size.
`0.5`	`1.367333`	
`0.25`	`1.367358`	
`0.125`	`1.367359`	This result should basically be the same as the previous result.

By consistently reducing the Z grid size, the results converge to a particular solution.

Studying the Effect of the X and Y Grid Sizes

As stated before, the BPM algorithm integrates along the Z axis, and so the Y and Y grid sizes have a different effect than the Z grid size. A convergence study ensures that accurate results are obtained.

Generally, smaller X and Y grid sizes can provide greater accuracy. However, a small grid size also results in a longer simulation. It is critical to perform a convergence study on the X and Y grid sizes to optimize the tradeoff between speed and accuracy.

To perform this convergence study, we will use the scanning capabilities of MOST, RSoft's scanning and optimization tool. While we could vary the X and Y grid sizes linearly (i.e. values of `0.5`, `0.4`, `0.3`, etc.), it is more efficient to study a wider parameter space by scanning exponentially (i.e. values of `0.5`, `0.25`, `0.125`, etc.). This can easily be done by representing the grid size as $0.5/2^N$ where 0.5 is a suitable starting grid size. To scan exponentially, we can now simply scan linearly over the parameter *N* from a value of 0 to an arbitrary value.
Open the Symbol Table, click the **Edit Symbols** button, and create/modify the following variables:

Variable	Value	Comments
`N`	`0`	
`grid_size`	`0.5/2^N`	This is the built-in variable for the X grid size.
`grid_size_y`	`grid_size`	This is the built-in variable for the Y grid size.
`step_size`	`0.125`	This sets the Z grid size. This value was shown by the previous convergence study to provide accurate results.

These grid size variables, like any other built-in variables, only appear in the symbol table if a non-default grid size is to be used. If these variables do not exist in the symbol table, they should be created.

Open MOST by clicking the **Launch MOST** button in the CAD. To enable a scan of mode computations, choose *Scan* and *BP Mode Solver* in the top left portion of the dialog.
To scan over the variable `N`, choose *N* under **Available Symbols**, select *Fixed Increments*, and click **Add**. The variable `N` will be added to the list of variables to be scanned. To scan from a value of 0 to 6 with a step of 1, set **Low** to `0`, **High** to `6`, and **Incr.** to `1`. Set **Output Prefix** to `converge`, and press **OK**.

Since this is a mode computation, we do not have to define measurements or metrics. The effective indices of the computed modes will be automatically saved for each point in the scan.

When the scan is complete, click the **Open DataBROWSER** button in the MOST simulation window, and open the file `converge_bp_mode_neffc.pcs`.

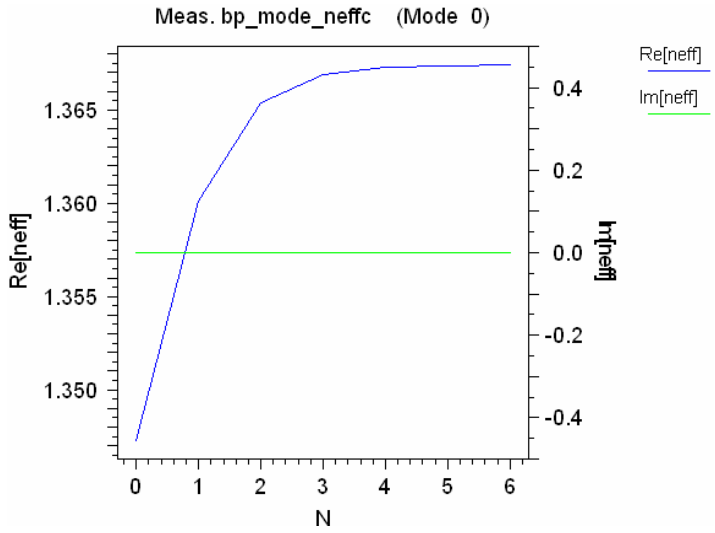

Figure 5: Convergence results.

The scan results show that as `N` increase, which corresponds to a decreasing grid size, the results converge to a particular solution. Of course, the smaller the grid size the longer the simulation times. The user needs to determine what order of accuracy is desired. Also, in this case, the X and Y grid sizes were set to the same value; other cases might require separate convergence studies of these parameters.

Note that some BPM simulation parameters are related to other parameters. Once optimal values for the X and Y grid sizes are found, it may be necessary to study the Z grid size again. It is important to study how one simulation parameter affects another so that a coherent result is produced.

Studying the Effect of Polarization Options

All the modes computed so far in this tutorial have been scalar: they have no polarization information. We will now study the effect of different polarization options on mode computations. The main option studied here will be the BPM **Vector Mode**. This option is described in more detail in Section 6.A. For purposes of this tutorial, we will assume that `N = 4` provide the needed accuracy in the previous convergence study. Open the Symbol Table and set this value.

Semi-Vector Mode

In the previous sections, we have used a **Vector Mode** of *None*, which corresponds to a scalar calculation. Click the **Compute Modes** button in the CAD and set **Vector Mode** to *Semi*. Press **OK**.

These calculations are for the TE mode; TM modes can be found by changing the **Polarization** setting.

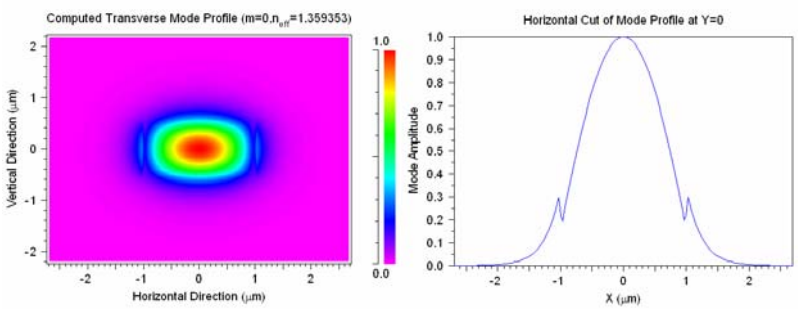

Figure 6: a) Semi-vector mode computation results. b) Horizontal cut of the mode profile at Y=0 showing discontinuities on the X boundaries of the waveguide. The cross-cut can be seen by right-clicking on either the X or Y axis of the mode profile.

The results show two important things. First, the effective index of the mode has changed in the third digit. Second, the discontinuities on the X boundaries of the waveguide are clearly seen. This indicates that the semi vectoral mode is necessary for this structure.

Full-Vector Mode

Set the **Vector Mode** to *Full* and rerun the mode computation.

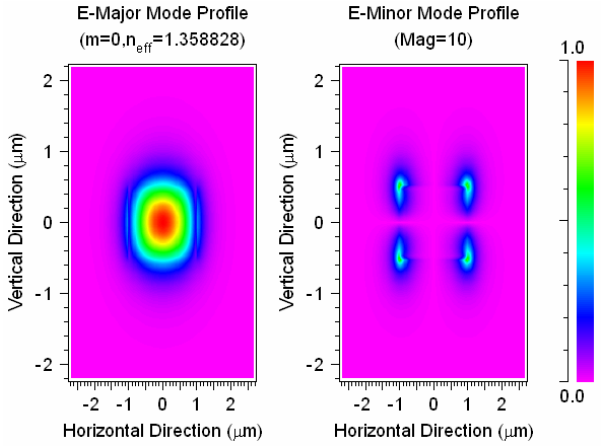

Figure 7: Full-vector mode computation results. The left plot shows the major mode profile (Ex in this case) and the right plot shows the minor mode profile (Ey in this case) of this quasi-TE mode. The right plot has been magnified 10 times for display purpose.

The mode profile from the full-vector computation appears similar to the semi-vector results, except the effective index differs in the fourth or fifth digit. If this accuracy is necessary, then a full vector simulation is necessary. Otherwise, a semi-vector simulation, which is faster and sometimes more robust, is sufficient.

While not explored here, it is typically a good idea to again study the grid size after enabling polarization options to ensure that an accurate result is obtained.

Areas for Further Exploration

- *Convergence studies and Polarization in 2D*

This tutorial performed convergence studies and explored polarization options in 3D only. Try performing similar studies in 2D

- *Perform additional convergence studies*

  This tutorial did not perform a comprehensive convergence study for X, Y, and Z grid sizes and polarization options. Try these on your own and see how each option affects the other.

Mode Tutorial 2: Using the Correlation Method

This tutorial discusses the basic use of the correlation method to compute modes in both 2D and 3D. The correlation method, which is based on the BPM algorithm, is very useful for computing modes of highly multimode as well as leaky and/or lossy devices. The technical background for this method can be found in Chapter 2.H, and the use of this method is described in Chapter 5. When compared with the iterative method described in Mode tutorial 1, the correlation method can take longer to run and be more sensitive to simulation parameters.

Modes of a Simple 2D Slab

We will create a simple 2D slab in this section and then compute the modes it supports.

Creating the Structure

Click the **New Circuit** button in the CAD window and set the following parameters:

Parameter	Value	Description
Free Space Wavelength	`1.55`	Sets the simulation wavelength.
Background Index	`1.0`	This represents the real refractive index of the background material where no structure has been defined.
Index Difference	`0.2`	This represents the default difference between a component and the background material. In this case this means that the structure will have an index of 1.2.

Once these settings are made, click **OK** to continue. Open the symbol table and create/modify the following variables:

Variable	Value	Comments
`step_size`	`0.2`	This is the built-in variable for the Z grid size.
`Length`	`2^9*step_size`	This variable will be used to represent the length of the waveguide.

This grid size variable, like any other built-in variables, only appear in the symbol table if a non-default grid size is to be used. If these variables do not exist in the symbol table, they should be defined.

In the CAD, draw a straight waveguide segment from (X,Z) = (0,0) to (0, `Length`). The structure length will then be defined by the variable `Length`, which is set to a power of 2 times the Z grid size. We will start with a 2^9 steps, but as we will see later, it is important to perform a convergence study on this parameter to see how it affects the simulation results.
The index profile of the structure can be computed by clicking the **Display Material Profile** button in the CAD.

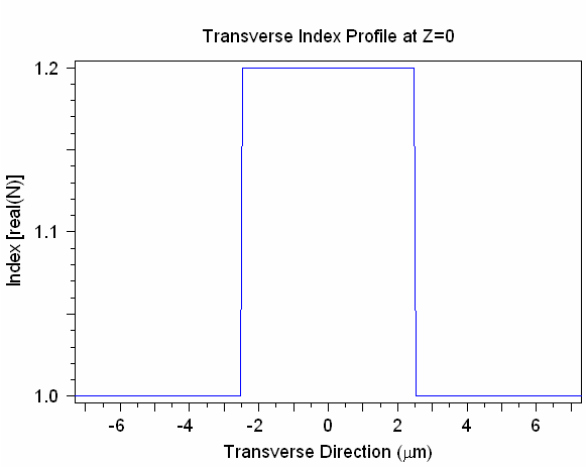

Figure 1: The computed index profile of the simple 2D slab.

Launch Settings

The use of an appropriate launch field is critical when computing modes. It is important to excite all the modes of the structure in order to properly compute them. If a particular launch field does not excite a particular mode, the mode solvers will not compute it. As a general rule, an offset Gaussian launch field is usually sufficient to excite a large continuum of modes, including any anti-symmetric modes.
To use an offset Gaussian launch field, click the **Edit Launch Field** icon in the CAD window. Set the **Type** to *Gaussian* and the **Position X** to be 2. Press **OK** to return to the CAD.

Computing the Fundamental Mode

To compute the fundamental mode, click the **Compute Modes** icon on the left toolbar in the CAD window. This dialog allows the user to set the computation parameters for the mode calculation. The mode method, which is not the correlation method described in this tutorial by default. Click the **Mode Options…** button and set **Method** to *Correlation*. For this example the rest of the default are sufficient.

It is recommended to check that the Monitor Step equals the Z grid size when using the correlation method since the number of grid points was specifically chosen.

Click **OK** to start the calculation. When the simulation completes, the computed fundamental mode profile will be displayed.

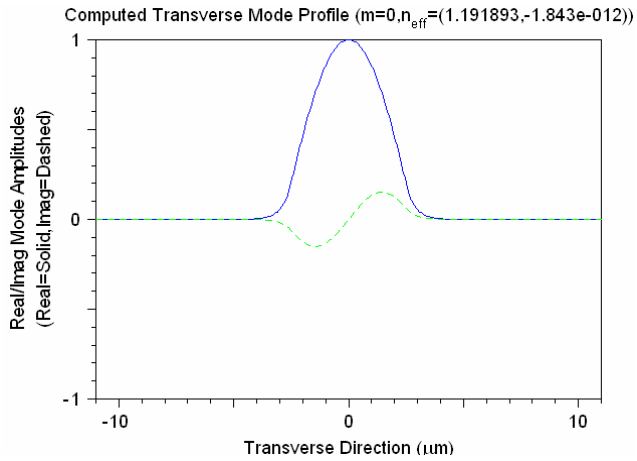

Figure 2: Fundamental Mode of simple slab waveguide. Note that the mode number and effective index are displayed in the top right corner of the plot.

During the simulation, two simulations are actually being performed: the first computes the mode spectrum and looks effective index values that have peaks; the second computes the mode profiles at the effective indices found. Also, unlike the iterative method, an imaginary effective index corresponding to loss is displayed with the simulation results.

While the effective indices are conveniently displayed in the top of every mode plot, this may not be useful for post-processing purposes. The text file `<prefix>.nef` , where <prefix> is the **Output Prefix**, contains a list of the modes found and the complex effective index values.

Computing Higher Order Modes

To compute higher order modes, click the **Mode Options…** button in the Mode Calculation Parameters dialog. Select *All Modes* and click **OK**. This will open a similar dialog as used above. Set the **Output Prefix** to `2D` and then press **OK** to start the simulation.

When the simulation has completed, the fundamental mode will be displayed in the simulation window. To view any of the modes computed, including higher order modes, click the View Graphs button in the CAD window. The plot files for the computed modes will be named 2d.p##, where ## denotes the mode number. These files can be opened to view the computed modes.

Multiple files are output for every mode computed: the `<prefix>_<comp>.p##` files are text files that contain plotting commands, and the `<prefix>_<comp>.m##` files are text files that contain the actual mode data where `<prefix>` is the **Output Prefix** given, `<comp>` is the field component (if applicable), and `##` is the mode number. These files, along with the file `<prefix>.nef` described earlier can be opened in any text editor for further processing.

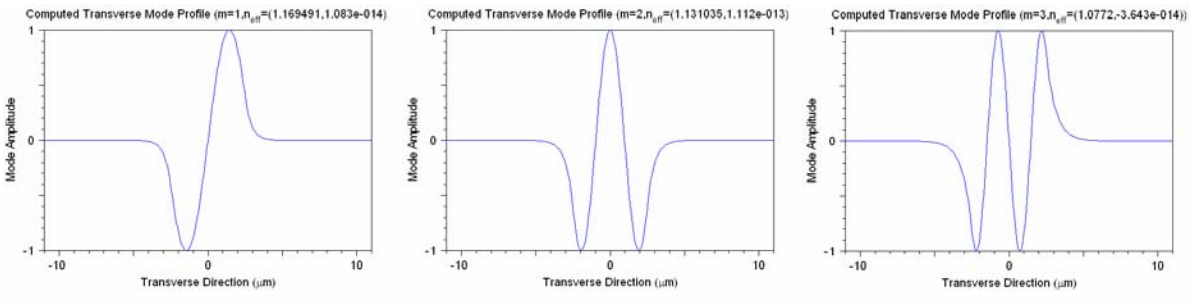

Figure 3: Some of the higher order modes supported by a simple 2D slab waveguide.

Viewing the Mode Spectrum

The correlation method uses an FFT of a correlation function to compute the mode spectrum in order to determine the modes that a structure supports.

The mode spectrum can be manually computed by selecting the **Compute Spectrum Only** options in the Mode Options dialog. This will compute only the mode spectrum; the mode profiles will not be solved for.

This mode spectrum is contained in the file `2d.pft`, which can be opened via the **View Graphs** button in the CAD window.

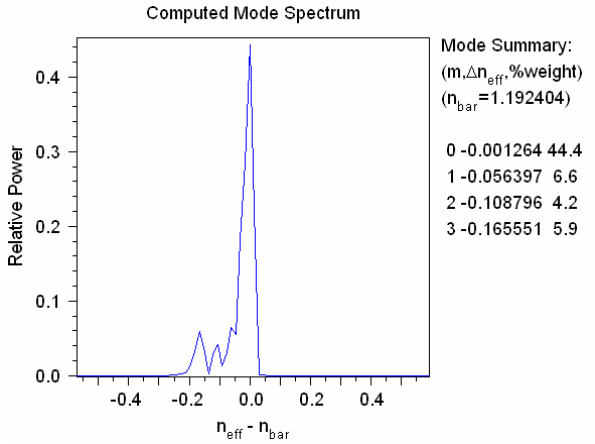

Figure 4: The Fourier transform result from the mode computation. Four modes were found, and are described in the right portion of the plot.

Since this result relies a Fourier transform, the length of the monitor data will control the data resolution in the mode spectrum. The accuracy of this spectrum can therefore be increased by increasing the length of propagation and thus the number of monitor data points.
Change the value of the variable `Length` to `2^13*Step` and rerun the mode computation with the same **Output Prefix**. Reopen the mode spectrum and note the greater data resolution.

The number of points in the monitor data, which is directly related to the length of the computation, is critical when performing a correlation mode computation.

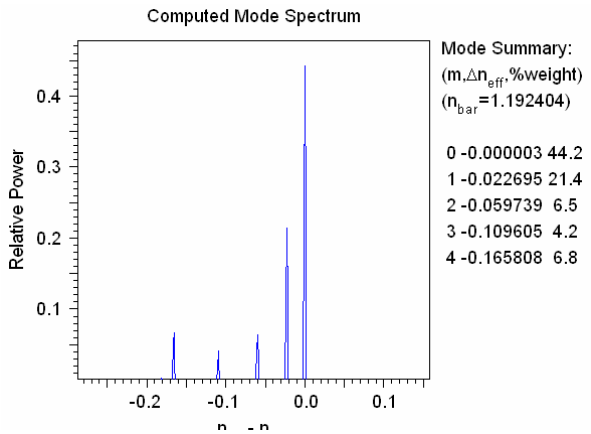

Figure 5: The mode spectrum found using a longer simulation length. The peaks are now very well defined.

The mode solver is choosing peaks that are above a certain threshold in the mode spectrum plot. Advanced users are encouraged to read Section 5.B.2 for more details about how to set this threshold.

Modes of a Simple Fiber Structure

We will create a simple 3D fiber structure in this section and then compute the modes it supports.

Creating the Structure

Press the **New Circuit** icon, and set the following parameters:

Parameter	Value	Description
Model Dimension	*3D*	This indicates that a 3D simulation should be performed.
3D Structure Type	*Fiber*	Indicates that the default structure type should be fiber.
Free Space Wavelength	`1.55`	Sets the simulation wavelength.
Background Index	`1.5`	This represents the real refractive index of the background material where no structure has been defined.
Index Difference	`0.1`	This represents the default difference between a component and the background material. In this case this means that the structure will have an index of 1.6.
Component Width	`6`	This represents the default width of a component in μm.

Once these settings are made, click **OK** to continue. Open the symbol table and create/modify the following variables:

Variable	Value	Comments
`step_size`	`0.2`	This is the built-in variable for the Z grid size.
`Length`	`2^12*step_size`	This variable will be used to represent the length of the waveguide.

In the CAD, draw a straight waveguide segment from (X,Z) = (0,0) to (0, `Length`). As with the 2D case, the structure length is set to a power of 2 times the Z grid size to maximize the efficiency of the FFT.
The index profile of the structure can be computed by clicking the **Display Material Profile** button in the CAD.

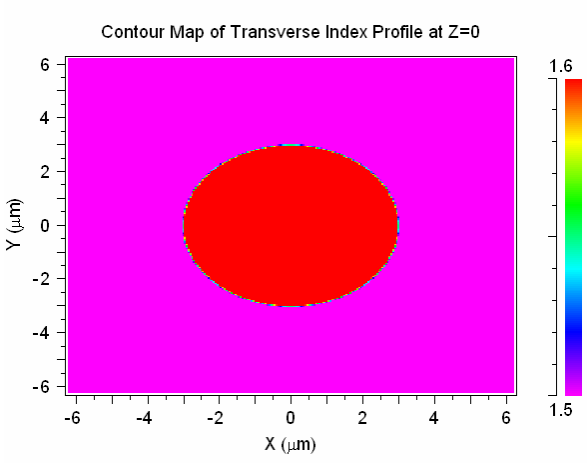

Figure 6: The computed index profile of the simple 3D fiber structure.

Launch Settings

As with the 2D case, the launch field must excite all the modes of the structure, and so an offset Gaussian launch will be used. Click the **Edit Launch Field** icon in the CAD window and set the **Type** to *Gaussian* , the **Position X** to be `0.5`, and the **Position Y** to be `0.5`. Press **OK** to return to the CAD.

Computing the Modes

To compute all the modes, use the Run/Compute Modes/All option in the CAD menu. Enter an **Output Prefix** such as *3D* and click **OK** to start the simulation.
When the simulation completes, the fundamental mode will be displayed. All the computed modes can be viewed via the **View Graphs** button in the CAD window. The Fourier spectrum can also be viewed.

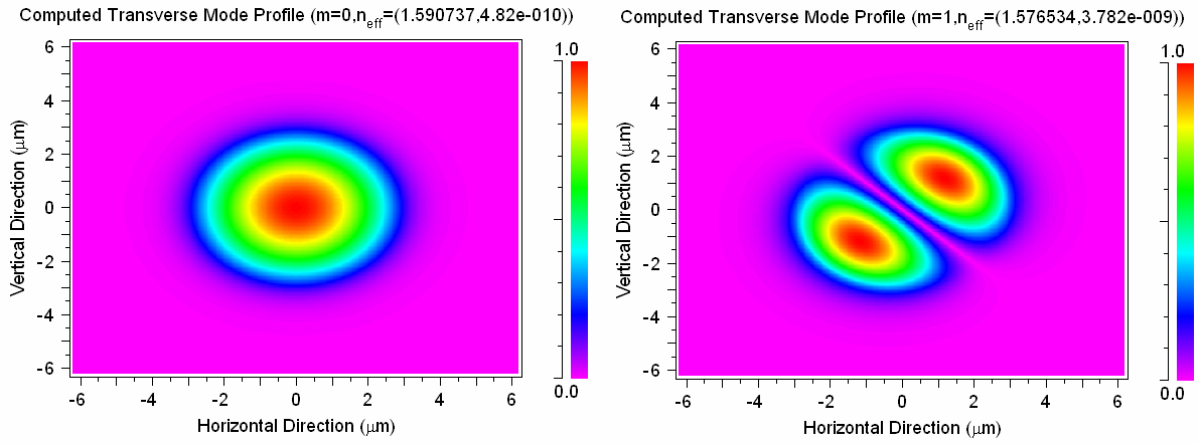

Figure 7: Some of the modes supported by a simple 3D fiber structure.

Areas for Further Exploration

- *Perform convergence studies*

  Mode Tutorial 1 described convergence studies for the grid sizes as well as polarization options. Try running similar studies with the correlation method and see how each parameter affects the simulation results. A convergence study can also be performed on the simulation length to see how increased data resolution in the mode spectrum can affect simulation results.

Mode Tutorial 3: Computing the Mode Cutoff

This tutorial illustrates the computation of the mode cutoff at which a structure becomes single-mode. This tutorial uses the correlation mode solving method, but the iterative method could be used as well. The modes of a channel structure will be studied as the waveguide dimensions are changed. This type of analysis could also be done for any other parameter including wavelength and refractive index.

User familiarity with the correlation method is assumed. See Mode Tutorial 2 for more details on the correlation method.

Creating the Structure

Press the **New Circuit** icon and set the following parameters:

Parameter	Value	Description
Model Dimension	`3D`	Enables a 3D simulation.
3D Structure Type	*Channel*	Sets the default structure type to channel.
Free Space Wavelength	`1.55`	Sets the simulation wavelength.
Background Index	`1.48`	This represents the real refractive index of the background material where no structure has been defined.
Index Difference	`0.02`	This represents the default difference between a component and the background material. In this case this means that the structure will have an index of 1.5.
Component Width	`7`	Sets the width of the channel segment.

Once these settings are made, click **OK** to continue. Open the symbol table and create/modify the following variables:

Variable	Value	Description
`step_size`	`1`	Sets the Z grid size.
`Length`	`2^10*step_size`	This will be used to set the length of the segment.

This step variable, like any other built-in variables, only appear in the symbol table if a non-default grid size is to be used. If this variable does not exist, it should be created.

In the CAD, draw a straight waveguide segment from (X,Z) = (0,0) to (0, `Length`). The structure length will then be defined by the variable `Length`.

Setting Simulation Parameters

This tutorial will use the correlation method. Click the **Compute Modes** button in the left toolbar and then click the **Mode Options…** button. Set **Method** to *Correlation* and press **OK** twice.
To set the launch parameters, click the **Edit Launch Field** button on the CAD toolbar and set the following parameters:

Parameter	Value	Description
Type	*Gaussian*	Indicates that a Gaussian launch should be used.
Width	`width*2`	Sets the width of the Gaussian launch.
Height	`height*2`	Sets the height of the Gaussian launch.
Position X	`width/3`	Sets the X position of the Gaussian launch.
Position Y	`height/3`	Sets the Y position of the Gaussian launch.

To indicate the modes which should be solved for, we will use the `mode_set` variable. This variable is equivalent to choosing the appropriate option in the Mode Options dialog. Open the Symbol Table and set `mode_set` to be `0-1` so that the first two modes will be found.

Parameter Scan for Mode Cutoff

To scan over the waveguide width, open MOST by clicking the **Launch MOST** button in the CAD. Set MOST to *Scan* using the *BP Mode Solver*. Next, under **Available Symbols**, select *width* and *Fixed Increments* and press **Add**. Set the **Low** to `3`, **High** to `8`, and **Incr.** to `.5`. Set **Output Prefix** to be `cutoff`.

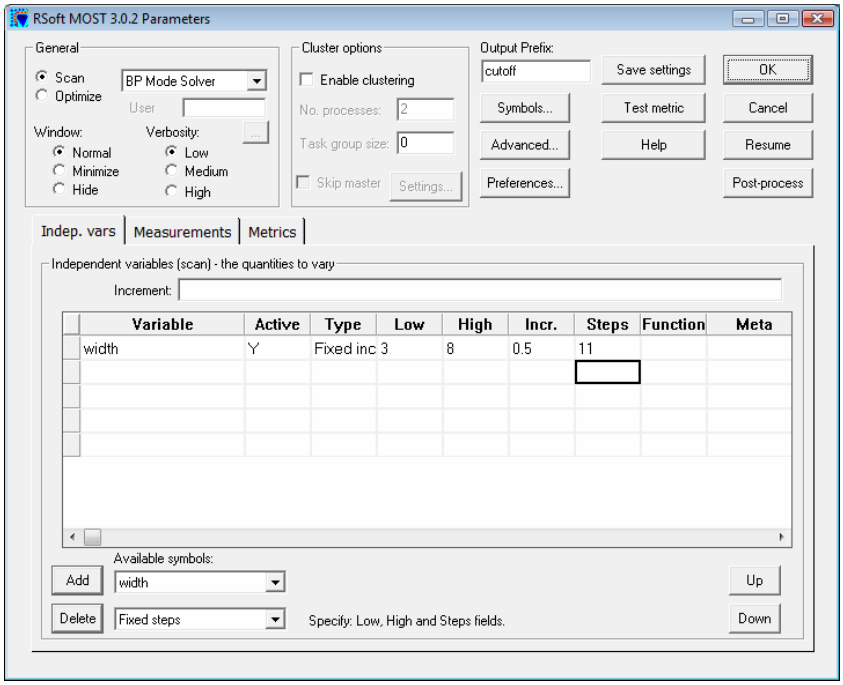

Figure 1: The MOST dialog where the variables to be scanned are set.

Next, click the **Measurements** tab. Under **Available** select *bp_mode_neffr* and then press **Add**. This enables the output of the real effective indices. Remove all other measurements and then click **OK** to start the scan.

When the scan is complete, click the **Open Databrowser** button on the MOST toolbar and open the plot `cutoff_scan_bp_mode_neffr.pcs` to view the results.

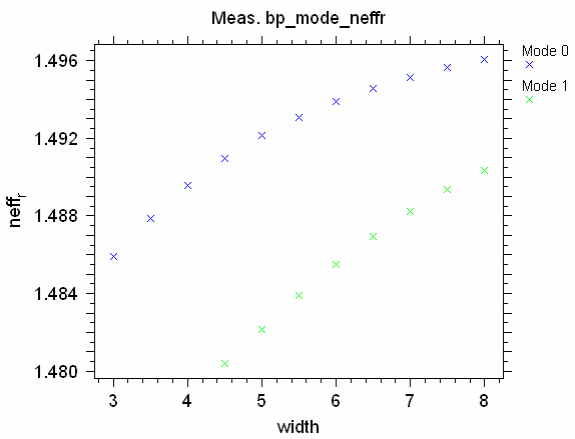

Figure 2: The results of the scan.

These results show that the waveguide is single-mode for values of width below 4.5.

Areas for Further Exploration

- *Scan over other parameters*

  This example scanned over the waveguide dimensions. Try scanning over other parameters such as the wavelength or refractive index.

- *Determine the cutoff point more accurately*

  This scan used a coarse step of 0.5. Try rerunning the scan to find the cutoff point more accurately.

Mode Tutorial 4: Dispersion in Single Mode Silica Fibers

Dispersion is a very important issue in optical communications, and so it is very advantageous to incorporate it into simulations. This tutorial covers the inclusion of dispersive properties of a typical single-mode silica fiber with a 0.035% core into a simulation.

This tutorial will first compute the dispersion curve of this structure by computing the optical modes at several wavelengths. Then several common dispersive parameters such as derivatives of the propagation constant β vs. wave vector **k**, group index, n_g, and dispersion parameter D will be computed using the `disperse` utility that is included with RSoft software.

For more on MOST and the `disperse` utility, please see the MOST and CAD manuals respectively. Also, user familiarity with the iterative method is assumed. See Mode Tutorial 1 for more details on the iterative method.

The material properties of the fiber are based on information from:

Fiber-Optic Communication Systems, Govind P. Agrawal, 2nd edition, Ch.2.3. P41-45.

Creating the Structure

Press the **New Circuit** icon in the CAD window and set the following parameters:

Parameter	Value	Comment
Model Dimension	*3D*	Indicates that a 3D simulation will be performed.
3D Structure Type	*Fiber*	Indicates that the default structure type should be fiber.
Component Width	`8`	This represents the default width of the fiber in μm.
Component Height	`width`	This represents the default height of the fiber in μm, and is set equal to the width.

Once these settings are made, click **OK** to continue.

Defining Dispersion

Material dispersion occurs because the material's refractive index changes as a function of the optical wavelength. This refractive index change can be well approximated by the Sellmeier equation:

$$n^2(\lambda) = 1 + \sum_{i=1}^{M} \frac{A_i \lambda^2}{\lambda^2 - \lambda_i^2}$$

where λ is the resonance wavelength and *Ai* is the oscillator strength.

For this tutorial, we will use three resonances to create this dispersion relation for pure silica in the symbol table. Open the symbol table and create/modify the following variables:

Variable	Value	Description
`A1`	`0.6961663`	Strength of resonance 1
`A2`	`0.4079426`	Strength of resonance 2
`A3`	`0.8974794`	Strength of resonance 3

L1	0.0684043	Location of resonance 1 (in μm)
L2	0.1162414	Location of resonance 2 (in μm)
L3	9.896161	Location of resonance 3 (in μm)
L	free_space_wavelength	The built-in variable free_space_wavelength defines the simulation wavelength. This definition defines the variable L as a convenient way to reference this wavelength.
R1	A1*L^2/(L^2-L1^2)	Resonance 1
R2	A2*L^2/(L^2-L2^2)	Resonance 2
R3	A3*L^2/(L^2-L3^2)	Resonance 3

Once these parameters have been defined, we can define the index of the fiber. For this fiber, we will define the cladding index equal to the equation given above, and the index of the core will be 0.035% higher. Modify the following variables in the symbol table:

Variable	Value	Description
background_index	sqrt(1+R1+R2+R3)	Sets the index of the fiber cladding. Note that in this example, we are only considering the core/cladding interface, which is generally a very good approximation since the modes are well-guided.
delta	.035*background_index	Sets the core index as a function of the cladding index.

The definition of the material index (background_index, delta) is now logically tied to the simulation wavelength (free_space_wavelength): when the value of the wavelength is changed, the index values will change according to the equation above.

While in this case we used an equation to define the material dispersion, the user can also use data files or the Material Editor to define material dispersion. See the CAD manual for more details.

Once these variables have been defined, click **OK** to return to the CAD.

Creating the Fiber

In the CAD, draw in a straight segment from (X,Z)= (0,0) to (0, 1000). The structure length, in this case 1000 μm, need only be long enough to allow the iterative mode solver to converge.

Computing Dispersion with MOST

Once the fiber has been created so that its refractive index is a function of wavelength, we can use MOST to scan over the wavelength and compute the fiber mode. MOST can also be used to perform post-processing with the disperse utility to compute several common dispersion parameters. Click the **Launch MOST** button in the left CAD toolbar to open the MOST dialog.

Creating the Parameter Scan

We are going to compute the mode of the fiber over a wavelength range of 1.1 μm to 1.8 μm with an increment of 0.01. Create a scan with these parameters in the MOST dialog, including an **Output Prefix** of `fiber`.

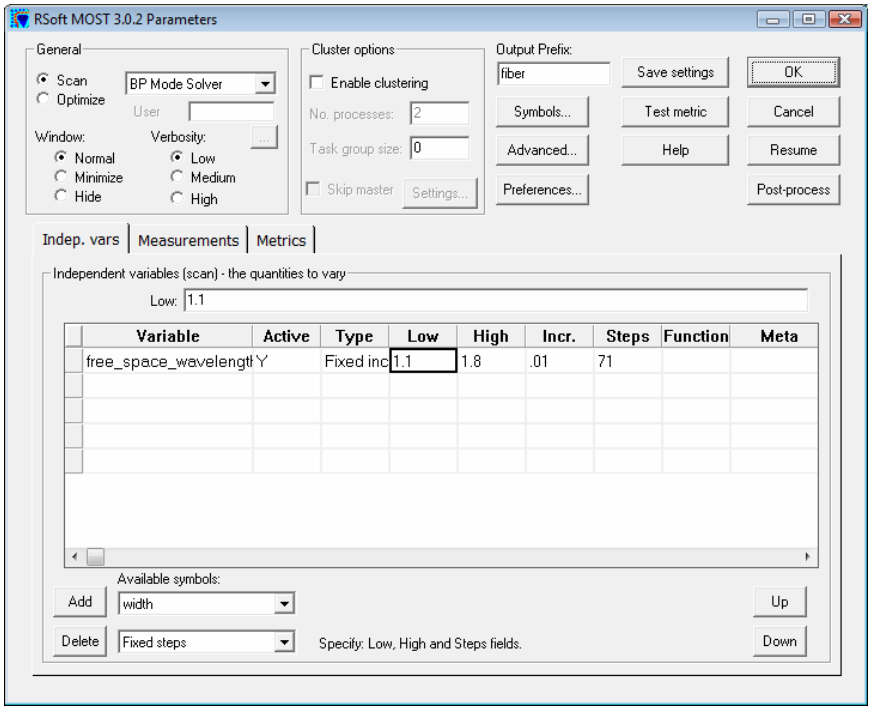

Figure 1: MOST Scanning Setup

Enabling Post-Processing

Click the Measurements tab and ensure that the measurement `bp_mode_neffr` is enabled. This will output the real effective index which we post-process using the `disperse` utility. The disperse utility takes the computed dispersion relation from the scan, and then outputs the requested dispersion parameters. When run from the command line, the command to do this is:

```
disperse -g -p
.\[most_prefix]_work\results\[most_prefix]_bp_mode_neffr.dat
```

where `[most_prefix]_bp_mode_neffr.dat` will be the file produced by the MOST scan, and `[most_prefix]` is the **Output Prefix** used in the MOST scan. For Linux simply reverse the slash. To automatically run this command at the end of the scan, click the **Advanced** button and enable the **Execute this command after the entire scan in the launch directory** option and set the value to the appropriate command listed above. Once these settings are made, enter an **Output Prefix** of `fiber` and click **OK** to start the scan.

Once the scan has completed, click the **Open DataBROWSER** icon in the MOST simulation window. Open the file `fiber_bp_mode_neffr.pcs`.

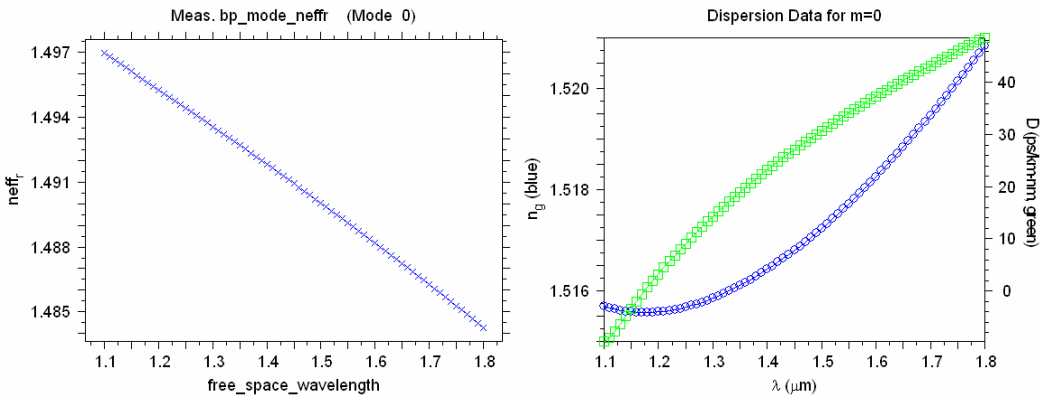

Figure 2: Results of a) MOST scan with the real effective index versus the wavelength, and b) computed dispersion parameters found via the disperse.exe utility.

Next, open the file `fiber_bp_mode_neffr.pde`, which may have already opened automatically at the completion of the scan. This plot (Fig. 3b) shows the dependence of n_g and D with wavelength. These results have excellent agreement with the reference given above.
Please note that these results include the combination of waveguide dispersion and material dispersion. The waveguide dispersion can be isolated by solving for modes at various wavelengths at a fixed index value. Additionally, other important parameters, such as the group velocity v_g, can also be obtained with the results offered by the `disperse` utility.

Areas for Further Exploration

- *Study multimode structures*

  Try increasing the structure dimensions so that it becomes multimode and rerun the calculations.

- *Perform a convergence study*

  Check the convergence of the mode computations performed above.

Mode Tutorial 5: Index Guided Photonic Crystal Fibers

Photonic crystal fibers (PCFs) rely on various repeated defects in order to confine light in a core. The BPM mode solvers have the capability of solving for the modes of index guided PCFs since the confinement is due to Bragg reflection effects. Air guided PCFs, however, must be solved by other tools, such as RSoft's FemSIM and BandSOLVE packages.

In this tutorial, we will solve for the modes of a PCF at different ratios of the period to wavelength. We will then determine ratios at which the fiber is single-moded for all wavelengths. It is based on:

> T.A. Birks, J.C. Knight, and P. St. J. Russell, *"Endlessly single-mode photonic crystal fiber,"* Optics Letters, **22**, 961 (1997)

This tutorial uses the iterative method. However, when looking to compute the modes of lossy PCFs, the correlation method should be used. User familiarity with the iterative method is assumed. See Mode Tutorial 1 for more details on the iterative method.

Creating the PCF Structure

The creation of periodic arrays is greatly simplified through the use of the Array Layout Utility. Choose Utility/Array Layout from the CAD menu and set the following parameters:

Parameter	Value	Comment
Dimensions	*2D (XY)*	This indicates that a 2D array should be created. In this array, the objects will represent air holes.
Lattice type	*Hexagonal Rings*	This indicates that a hexagonal ring lattice should be used.
L	`0`	
M	`4`	
Contents	*Ellipse*	Indicates that a elliptical object should be used in each unit cell. These objects will represent air holes.
Output Prefix	`pcf`	The name of the `.ind` file to be created by the utility.

Click **OK** to close the utility and open the new design file. Open the symbol table and create/modify the following variables:

Variable	Value	Comment
`Q`	`1`	This variable will be used to easily scan logarithmically over the wavelength.
`R`	`0.3`	This variable will be used to set the ratio of the Period to hole diameter.
`Period`	`2.3`	
`width`	`R*Period`	Sets the diameter of the air holes.
`height`	`width`	
`free_space_wavelength`	`Period/(10^Q)`	The simulation wavelength. The variable `Q` is

		used here so that we can scan over the wavelength in steps equal to a power of ten.
`background_index`	`1.45`	Index of core/cladding.
`delta`	`1-background_index`	Index difference of air holes such that they have an index of 1.0.

When these settings have been made, click **OK** to return to the CAD. The structure is now complete, and its index profile can be computed by clicking the Display Index Profile button in the CAD.

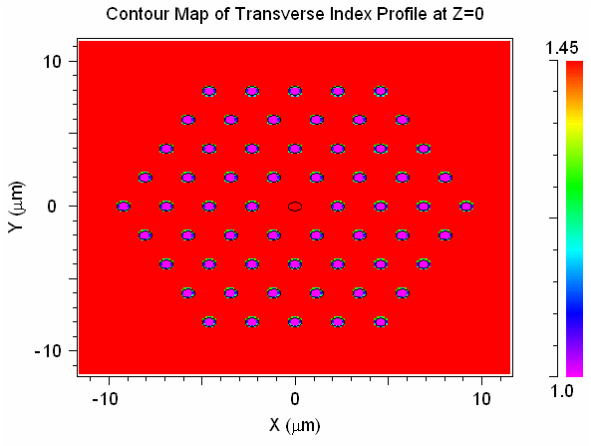

Figure 1: The index profile of the PCF structure.

Note that the central segment, which lies in the middle of the core, has an index difference set to 0. The index of this segment is set by a special symbol, `DeltaCore`, that the Array Layout created.

Computing the PCF Mode

Press the **Compute Modes** icon, and then press **OK**. When finished, you should see Fig. 2.

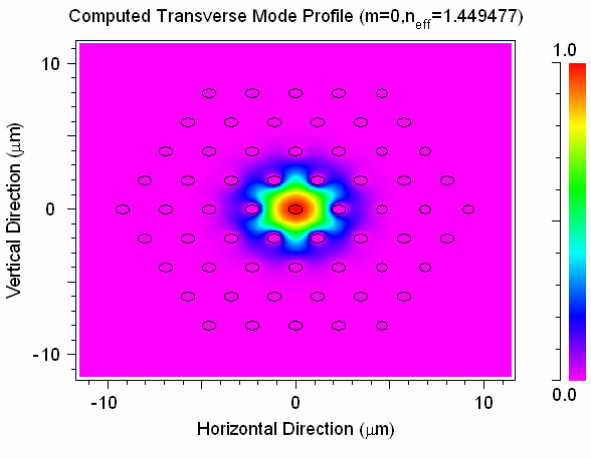

Figure 2: Fundamental Mode Computation

For this situation, the mode does appear to be well guided. However, as the wavelength gets much larger, a lot of the field will no longer be as confined to the core. We could use the correlation method

with a large domain, but this could be very computationally prohibitive. Instead we will use some of the advanced features found in BeamPROP to compute the modes of this device.

Exploring the Single-Mode Condition

In order to explore the single-mode condition, we need to solve for the fundamental mode over a wide range of wavelengths. In a conventional fiber, the number of bound modes is governed by the *V* number, which increases without limit as the wavelength decreases. The reference above shows that it is possible to define an effective *V* number for a PCF that indicates reasonably accurately whether or not a fiber is single-moded:

$$V_{eff} = (2\pi\Lambda/\lambda)\sqrt{n_o^2 - n_{eff}^2}$$

where Λ is the lattice period (`Period`), *n0* is the glass index and *neff* is an "effective cladding index." The effective cladding index can be considered as the effective index of the first radiation state, which is equivalent to finding the lowest mode in the band structure of the plain lattice. The reference shows that in contrast to step-index fibers, the *Veff* for a PCF converges to a finite value as the wavelength decreases. If the finite value is less than approximately 2.405, the PCF is single-moded.

Setting Simulation Parameters

In order to compute the parameter neff in the above equation, we will solve for the mode of the plain lattice with no defect. This can be done by choosing boundary positions such that only a single unit cell lies within the domain, the use of symmetric boundary conditions, and a plane wave launch.
To do this, we will set all the simulation parameters via the symbol table. Open the symbol table and create/modify the following variables:

Variable	Value	Description
`boundary_min`	`-Period/2`	Sets the minimum X boundary location.
`boundary_max`	`Period/2`	Sets the maximum X boundary location.
`boundary_min_y`	`-PeriodY`	Sets the minimum Y boundary location.
`boundary_max_y`	`PeriodY`	Sets the maximum Y boundary location.
`bc_type_bottom`	`BC_SYMMETRIC`	Sets the Y minimum boundary type to symmetric.
`bc_type_top`	`BC_SYMMETRIC`	Sets the Y maximum boundary type to symmetric.
`bc_type_left`	`BC_SYMMETRIC`	Sets the X minimum boundary type to symmetric.
`bc_type_right`	`BC_SYMMETRIC`	Sets the X maximum boundary type to symmetric.
`DeltaCore`	`delta`	The variable created by the Array Layout to set the index of the central air hole. This value enables the presence of the central air hole.
`grid_size`	`width/16`	Sets the grid size in X direction
`grid_size_y`	`width/16`	Sets the grid size in Y direction
`launch_type`	`LAUNCH_RECTANGLE`	Enables a rectangular launch which will be used to represent a plane wave.
`launch_width`	`inf`	Launch size to excite mode.
`launch_height`	`inf`	Launch size to excite mode.

`step_size_idbpm`	`0.005`	Sets the grid size in the Z direction for the iterative method.
`neff_tol`	`1e-009`	Sets the **Neff Tolerance** used by the iterative mode method.

Once the simulation parameters have been set, it is beneficial to look at the index profile to see the new simulation domain.

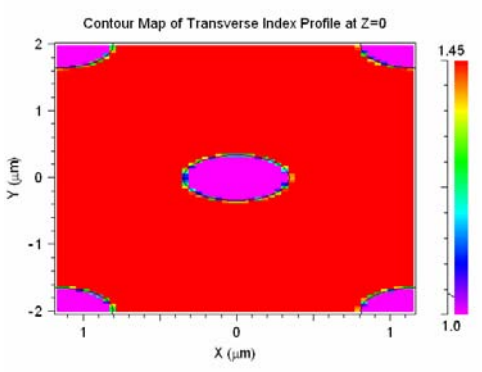

Figure 2: The reduced simulation domain for one unit cell.

Performing a Parameter Scan over Wavelength and Hole Size.

Two variables, `Q` and `R`, have been defined in the symbol table to represent the wavelength and hole size at a fixed period. These variables can be scanned to study the single-mode condition.

Click the **Launch Most** button in the CAD toolbar to open the MOST dialog. We are going to compute the mode and scan over the variable `Q`, which represents the wavelength, from -0.5 to 1.5 with an increment of 0.1, and `R` from 0.15 to 0.45 with an increment of 0.15. Create a scan with these parameters in the MOST dialog, including an **Output Prefix** of `pcf`.

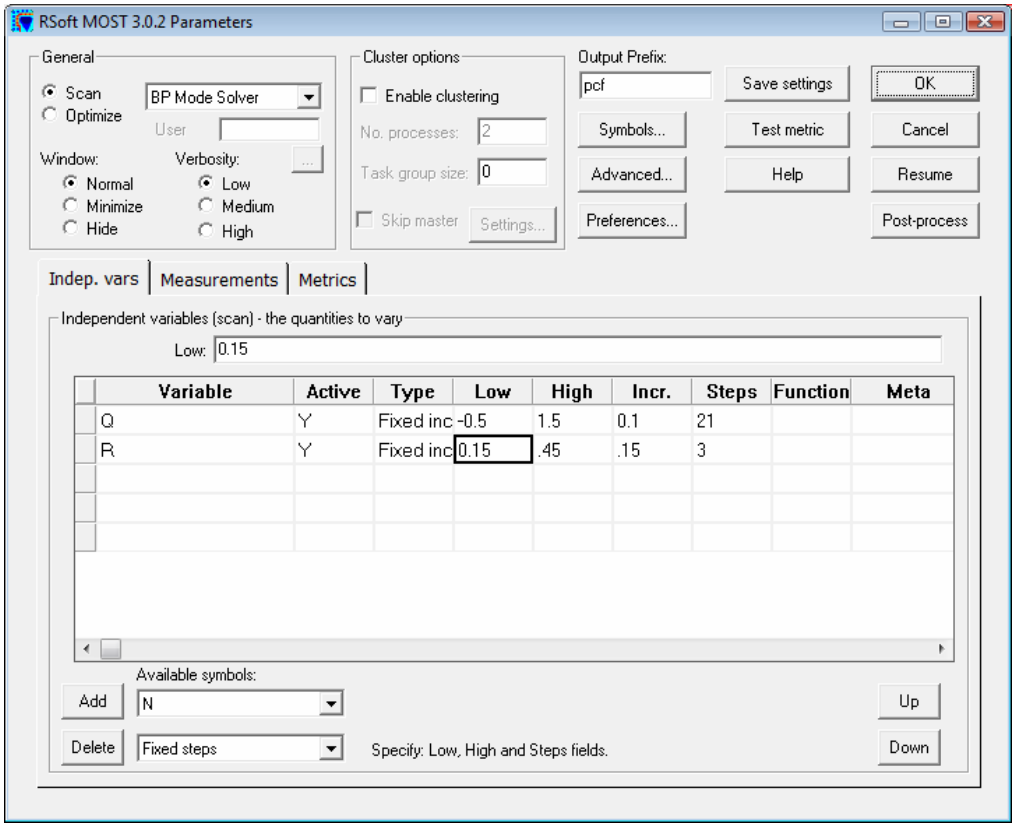

Figure 3: MOST scanning dialog with variables defined.

Click the Measurements tab and ensure that the measurement *bp_mode_neffr_0* is enabled. This measurement will output the real effective index of the computed modes.
Next, click the Metrics tab. Here we will define a metric to automatically compute the parameter V described above. Create a metric called `V` and set it equal to:

```
k0*Period*sqrt(background_index^2-bp_mode_neffr_0^2)
```

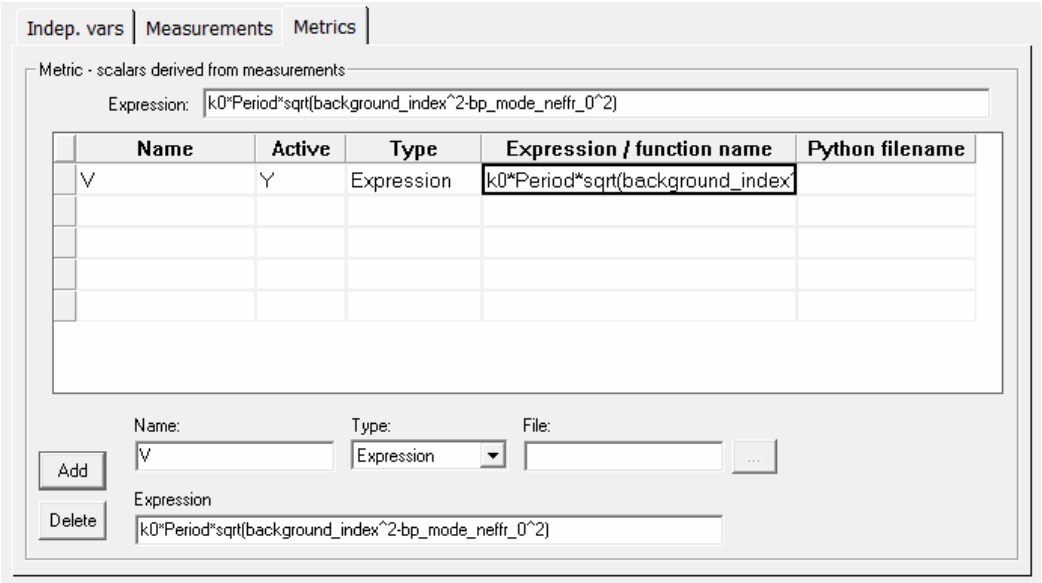

Figure 4: The portion of the MOST dialog where metrics are set.

Once the metric has been defined, click **OK** to start the scan. When it is complete, click the **Open DataBROWSER** button in the MOST simulation window and open the file `pcf_met_V.pcs`.

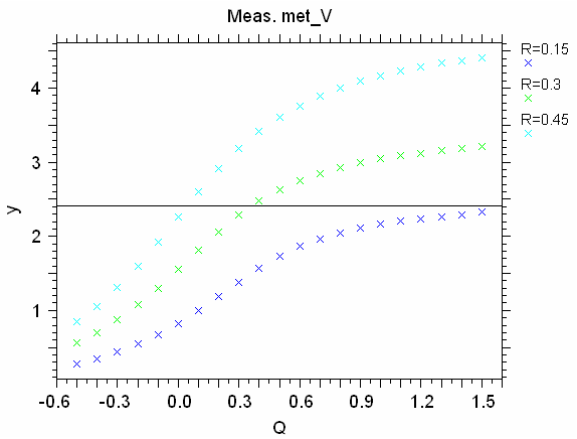

Figure 5: Metric plot of V parameter with a line drawn in at V=2.405, which was drawn with the /ln command. Also remember that

From the reference, we know the device is single mode if V is less then 2.405. The fiber geometries that lie below the line in Fig. 5 are single mode, and a fiber with `R`=0.15 will be endlessly single-moded for all wavelengths.

Areas for Further Exploration

- *Perform a convergence study*

  Check the convergence of the mode computations performed above.

Mode Tutorial 6: Leaky Modes of a Rib Waveguide

The iterative mode computation method, while it is very useful for fast mode computations of well-defined modes, it is not capable of finding lossy or leaky modes. This tutorial illustrates the use of the correlation method to solve for the leaky modes of a high index contrast rib waveguide with a thin silica buffer layer between the core and substrate.

User familiarity with the correlation method is assumed. See Mode Tutorial 2 for more details on the correlation method.

Creating the Structure

Click the **New Circuit** button in the CAD toolbar, and set the following parameters:

Parameter	Value	Comment
Model Dimension	*3D*	This indicates that a 3D structure will be created.
3D Structure Type	*Multilayer*	Indicates that the default structure type should be multilayer.
Free Space Wavelength	`1.55`	Sets the simulation wavelength.
Background Index	`3.5`	For the case of a multilayer structure type, this sets the substrate index.
Index Difference	`0.45`	
Component Width	`5`	This sets the width of a waveguide segment.
Component Height	`1.75`	This sets the height of a waveguide segment.
Cover Index	`1.0`	This sets the cover index.
Slab Height	`1.0`	This sets the height of the slab around the waveguide.

For more information about how a multilayer structure is defined, please see the CAD manual.

Next, press the **Edit Layers...** button and modify Layer Table 0 as shown in Fig. 1.

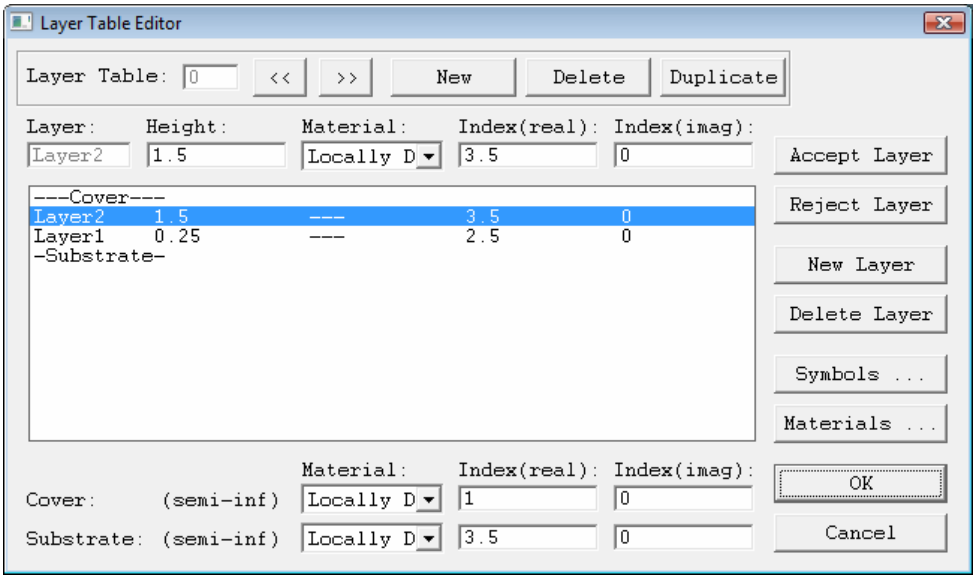

Figure 1: Layer Table 0 settings.

Press **OK** in this window and the next to open the CAD window.
Click the **Edit Symbols** button and create/modify the following variables:

Variable	Value	Comment
`step_size`	`0.05`	This built-in variable sets the Z grid size.
`Length`	`2^15*step_size`	This variable will be used to set the length of the multilayer waveguide. This value has been chosen to maximize the efficiency of the FFT used by the correlation mode method.

In the CAD, draw a straight waveguide segment from (X,Z) = (0,0) to (0, `Length`). The segment will use the default parameters defined above, and the index cross-section can be computed by clicking the Display Index Profile button in the CAD.

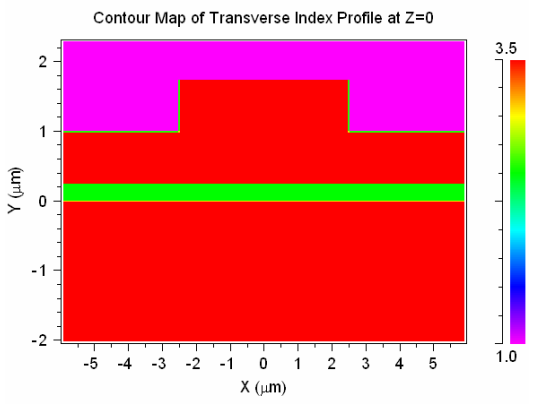

Figure 2: The computed index profile of the multilayer waveguide structure.

Computing the Mode

In this section we will define the launch field, and then illustrate one incorrect method to compute the mode, and then the correct method.

Defining the Launch Field

In order to compute the mode of this structure, the launch field must be set correctly. It is necessary to excite the mode in order to compute it. Press the Edit Launch Field button in the CAD and set the following parameters:

Parameter	Value
Type	*Gaussian*
Position X	`0.5`
Position Y	`1.0`

Setting the Polarization

The polarization controls are in the Mode Calculation Parameters dialog. Click **Compute Modes** and set **Vector Mode** to *Semi* and the **Polarization** to *TE*.

Calculating the Mode Incorrectly

Even though this structure is leaky, and therefore the correlation method must be used to solve for the mode, we will first use the iterative method for illustration purposes. Since the iterative method is enabled by default, simply click **OK** in the Mode Calculation Parameters dialog.
During the mode computation, the entire field will quickly 'leak' into the substrate. This is a result of how the iterative method works: it seeks the highest index modes.

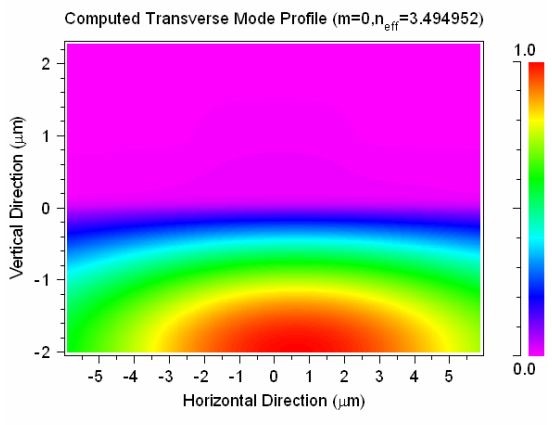

Figure 3: Incorrect mode found via the iterative method.

Computing the Mode Correctly

As seen above, the correlation method must be used for this structure. Click the **Compute Modes** button in the CAD, then click the **Mode Options...** and set the **Method** to *Correlation* and click **OK**. Set the following options:

Parameter	Value	Comment
Y **Domain Min**	`-2`	This sets the domain minimum slightly lower so that the leaky field will be visible in the computed mode profile.
Y **Grid Size**	`0.02`	This value is chosen so that the simulation will run faster.
Z **Slice Grid**	`0.1`	

Click **OK** to start the mode computation.

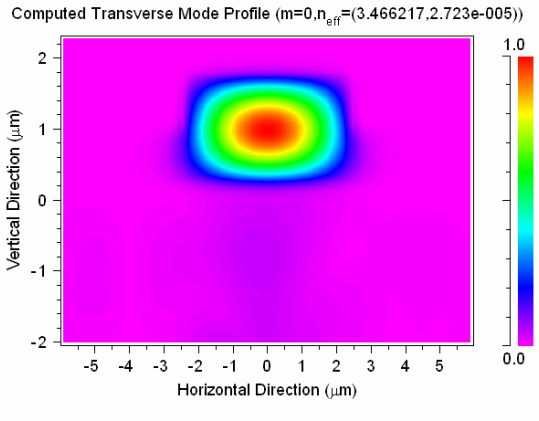

Figure 4: Computed mode of leaky multilayer waveguide. The leaky field can be seen below the waveguide.

The results clearly show a clear waveguide mode. The imaginary part of the effective index, which is displayed above the mode profile, is somewhat high indicating the lossiness of the device.

Areas for Further Exploration

- *Perform a convergence study*

  The example in this tutorial was run with slightly coarse simulation parameters in order to speed up the simulations. Try refining the parameters via a convergence study and see if the results change.

Mode Tutorial 7: Modes with a Non-Uniform Grid

This tutorial describes the use of a non-uniform grid with BeamPROP to compute modes, 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.

User familiarity with the iterative method is assumed. See Mode Tutorial 1 for more details on the iterative method. For more information on non-uniform grids, see Chapter 9 in the CAD manual.

The files associated with this tutorial can be found in

```
<rsoft_dir>\examples\BeamPROP\Tutorial\Mode_Calculation/Tut7.
```

Structure Overview

Open the file `nonuniform-bpm.ind`. The structure is a multilayer rib with two thin quantum well structures. The layer structure can be viewed in the Layer Table, and to view the index profile of the structure, use the **Display Material Profile** option in the CAD.

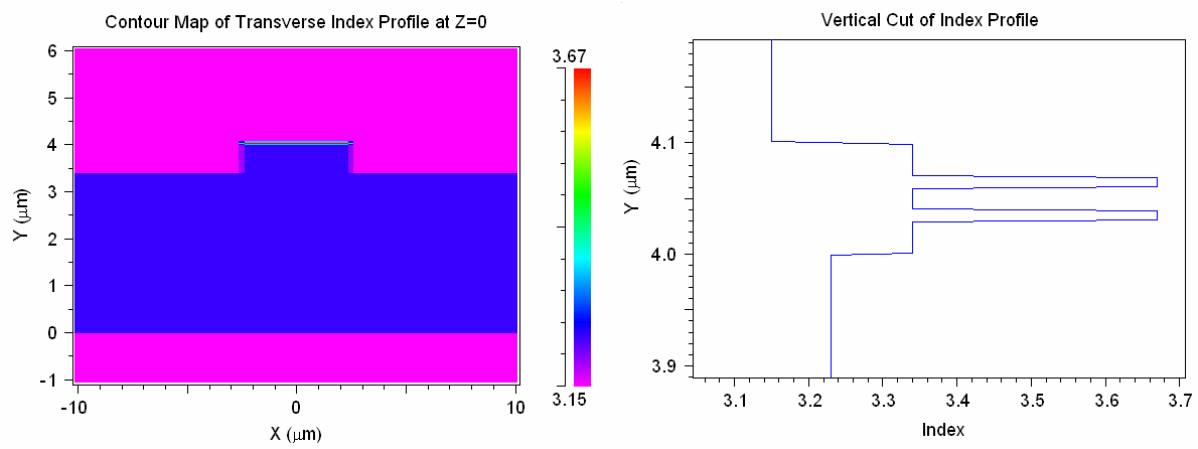

Figure 1: a) The index profile of this structure and b) a vertical cut through the center of the rib that shows the quantum wells.

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: ultra thin (sub-wavelength) quantum wells in a relatively larger waveguide structure. In order to correctly resolve these small structures, several grid points would be needed within the quantum wells. When using a uniform grid, this fine grid size would have to be used throughout the simulation domain, even in the other regions. This would be an unnecessary waste of computational resources as the other regions can be resolved with a much coarser grid size. The use of a non-uniform simulation

grid removes this extra computational burden 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.

Creating a Non-Uniform Grid

Enable the use of a non-uniform grid by selecting **Enable Nonuniform** in the BeamPROP 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 BeamPROP Simulation Parameters dialog.

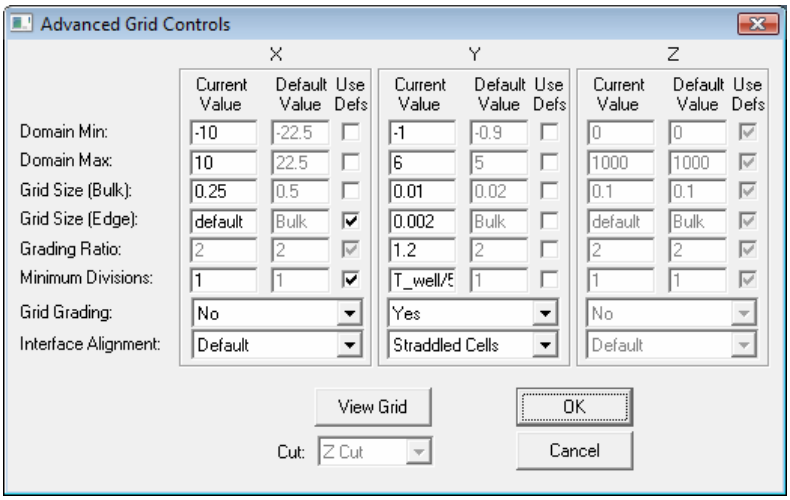

Figure 2: The Advanced Grid Controls dialog where the options that control a non-uniform grid are set.

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 equivalent to the options located in the BeamPROP 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)*

The **Grid Size (Bulk)** is set to the relatively coarse values of 0.25 and 0.1 respectively for X and Y. These values correspond to the values which would be used if the thin quantum well structures were not present.

- *Grid Size (Edge)*

  The **Grid Size (Edge)** options 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 Y it is set to a value that accurately resolves the thin quantum well regions.

- *Minimum Divisions*

  The **Minimum Divisions** setting 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 5 cells across each quantum well.

These settings ensure that a fine grid size of 0.002 μm along Y 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.

Furthermore, the Y **Interface Alignment** has been set to *Straddled Cells* so that grid points are not located on material interfaces.

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 BeamPROP Simulation Parameters dialog.

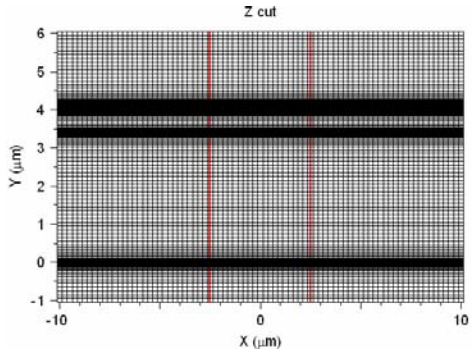

Figure 3: The non-uniform grid for the metal slit structure. Note the increased density of grid points for better resolution in the quantum well region.

Modes with a Non-Uniform Grid

In this simulation, the fundamental mode of this structure will be computed. A off-axis Gaussian launch field has been configured to excite the structure.

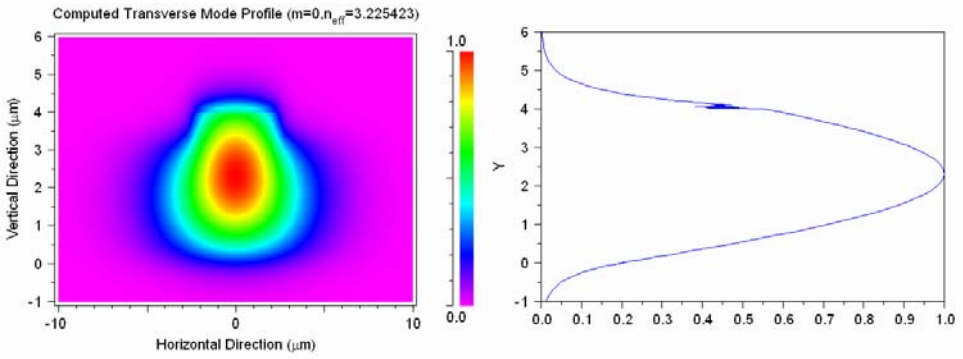

Figure 4: a) The mode computed using a non-uniform grid. b) A cross-section through the mode.

To compute the mode, enter an **Output Prefix** of `nonuniform_mode` and click **OK**. When the simulation has completed, Windows users can open the Windows Task Manager to see how much system memory the program uses, as well as the simulation time. The Task Manager can be opened by right-clicking on the Start bar and choosing 'Task Manager'. Linux users can use the `ps` command to get similar results.

The computer used to create this example took ~15 minutes to complete this simulation and used about ~14 MB of simulation memory. 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 resulting mode shows an effective index of 3.225423.

Modes 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 BeamPROP 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 Y **Grid Size** to `0.002`.

To run the simulation, enter an **Output Prefix** of `uniform_mode` and click **OK**. The simulation will run, though much slower than the non-uniform case. Again note the memory used and simulation time.

The computer used to create this example took ~1 hour to complete this simulation and used about ~30 MB of simulation memory. 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 resulting mode shows an effective index of 3.225434.

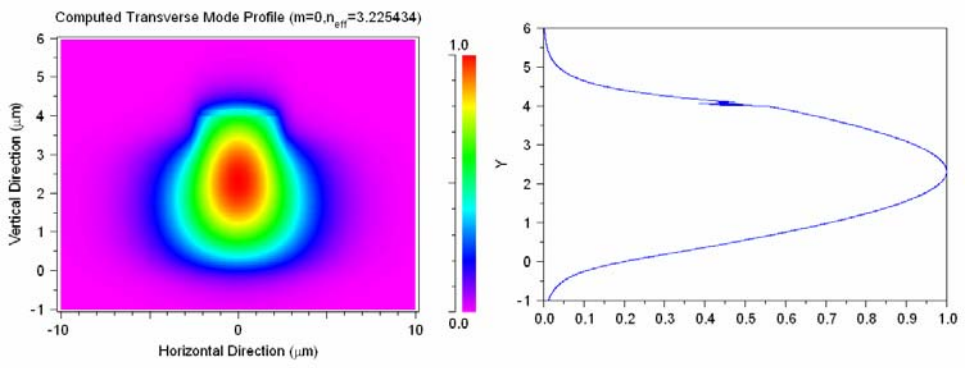

Figure 4: a) The mode computed using a uniform grid. b) A cross-section through the mode.

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 60 minutes to 15 minutes, resulting in a simulation that is about 4 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 34 MB to 14 MB. This is because the non-uniform grid uses less grid points than a uniform grid.

- *Accuracy of the result:*

  The effective indices of the modes differ on the 7th digit, even though the non-uniform grid was coarse in the bulk regions. Furthermore, the mode profiles are very similar as can be seen by comparing Fig. 3 and Fig. 4.

10

Multi-Physics Tutorials

This chapter contains several tutorial examples which illustrate the use of RSoft's Multi-Physics Utility. This utility is shared by several RSoft simulation tools these tutorials have been placed in a separate chapter to make them more accessible. Corresponding to each example there is one or more `.ind` and/or data files, which are located in the `<rsoft_dir>\examples\Utitiles\MultiPhysics` subdirectory.

While the examples in these tutorials use BeamPROP, any of RSoft's simulation tools that share the CAD interface can be used with the Multi-Physics Utility. Before working through these tutorials, it is recommended that users work through at least one of the basic tutorials in this manual before proceeding.

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 at info@rsoftdesign.com and we will assist you in finding a solution if possible.

User familiarity with the CAD and the simulation capabilities of BeamPROP is assumed.

Tutorial 1: Simple Electro-Optical Example

Electro-optic effects can be modeled in steady state in BeamPROP by defining materials with electro-optic parameters in the Material Editor, and then defining electrodes in the CAD interface. This example studies a simple SOI rib waveguide with electrodes. We will use the Multi-Physics Utility to study the index perturbation caused by the electrodes.

The associated file for this tutorial is located in the `<rsoft_dir>\examples\Utilities\MultiPhysics\electrode\` directory.

Structure Overview

Open the file `soi_electrode.ind` in the CAD interface.

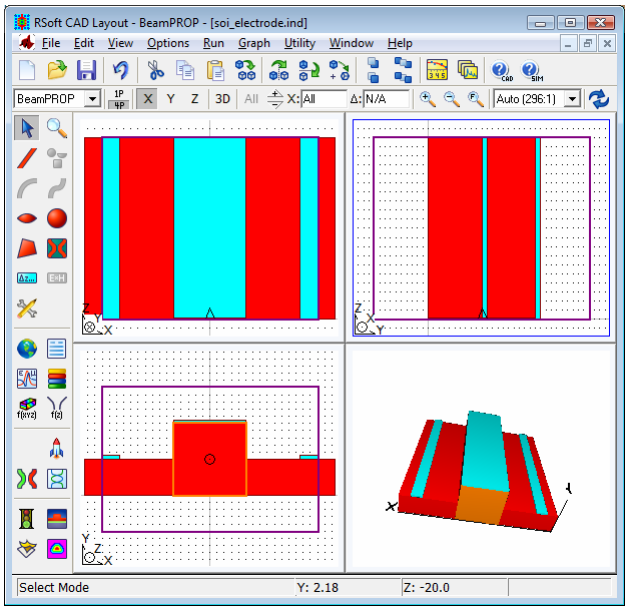

Figure 1: The simple SOI rib waveguide in the CAD interface. The light blue regions are electrodes.

Note the following about the design:

- *Structure*

  This structure is created using the built-in Multilayer structure type. It consists of a 2 x 2 μm Si rib structure, a Si slab with a height of 1 μm, and a SiO2 cover and substrate. A layer table is used to define the Silicon rib and slab.

- *Material*

  The material properties of the waveguides in this design are defined by two materials: *Si_ntype* and *Si_ptype*. Both of these materials are, for the purposes of this tutorial, equivalent. This same structure is used in Tutorial 5 where the Carrier Effects effect is discussed and the n- and p- doping is important. Both materials have a real refractive index of 3.476, as well as a electro-optic **DC Dielectric Constant** equal to the 11.8 and **Electro-Optic Coefficients** equal to the variables `RX` and `RY`. The values of these parameters have been chosen to correspond to strained Silicon. These settings can be seen in the Material Editor by selecting these materials and looking at the Eps Linear and Electro-Optic tabs.

  These parameters can be viewed using the **Display Material Profile** button on the left toolbar. Be sure to set **Material Property** to either *Index (Real)*, *DC Dielectric Constant* or *Electro-Optic Coefficients* to see each property.

- *Electrodes*

  Three electrodes have been defined in this simulation, and are displayed as light blue segments. Note the geometry and position information for each electrode, and that the **Index Profile Type** is set to *Inactive* so that they only are used to compute the DC electric field, and do not directly contribute to the index profile of the structure. Furthermore, look in the Additional Segment Options and note that the **Electrode/Heater Type** is set to *Electrode*, and the **E/H Control Parameter** is set to the applied voltage (`0` for the outer electrodes, and `V` for the inner electrode). For more information on defining electrodes, see the CAD manual.

Using the Electrodes

We will now determine the effect that the electrodes will have on the refractive index. Choose Utility/Multi-Physics from the CAD menu to open the Multi-Physics Utility. Make sure the **Physical Effect** is set to *Electro-Optic* and enter an **Output Prefix** such as `test` and press **OK**. When the calculation is complete, the results will be displayed in DataBROWSER.

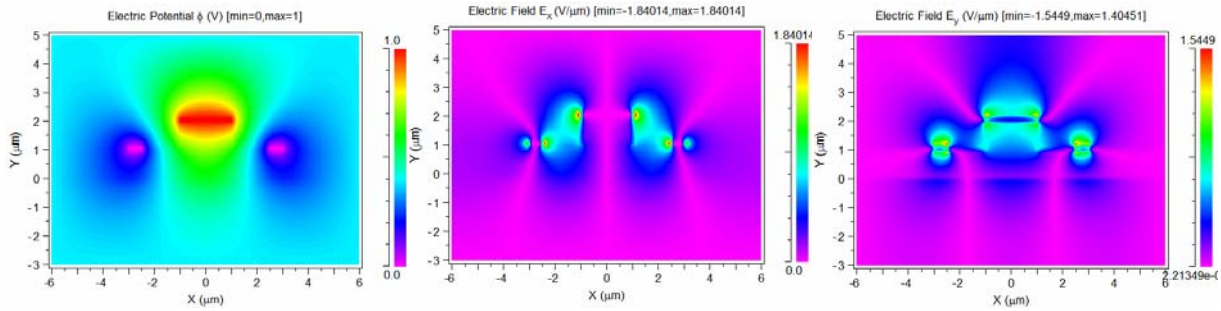

Figure 2: The computed a) electric potential for the electrodes, and b) E_x, and c) E_y fields produced by the electrodes.

Several plots are output. Fig. 2 shows the electric potential as well as the E_x and E_y electric fields. The file `test.ppf`, shown in Fig. 3, shows the absolute index perturbation that is generated by these electric fields via the electro-optic coefficients defined in the Material Editor. The exact formulation of how the index perturbation is computed from the electric fields and material properties can be found in the CAD manual and is included in any simulations run.

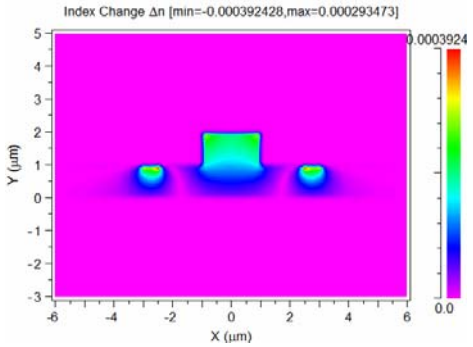

Figure 3: Computed index change for electro-optic effect.

Areas for Further Exploration

- Create a MOST scan over the applied voltage (variable `V`) and see how the effective index of the fundamental mode changes. Be sure to set the simulation tool in the MOST dialog to *BP Mode Solver* and be sure to choose the *bp_mode_neffr* measurement which corresponds to the real part of the effective indices.

Tutorial 2: Mach-Zhender Electro-Optical Modulator

Electro-optic effects can be modeled in steady state in BeamPROP by defining materials with electro-optic parameters in the Material Editor, and then defining electrodes in the CAD interface. This example studies a Mach-Zehnder electro-optical modulator.

The associated file for this tutorial is located in the `<rsoft_dir>\examples\Utilities\MultiPhysics\electrode\` directory.

Structure Overview

Open the file `eo_modulator.ind` in the CAD interface.

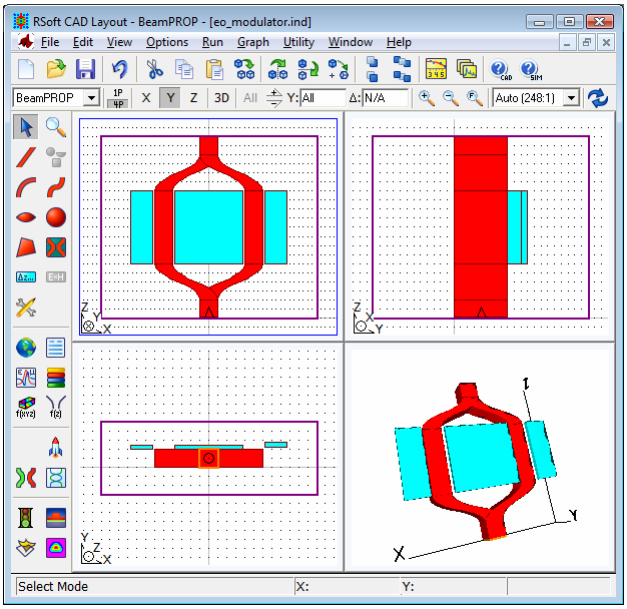

Figure 1: The Mach-Zhender structure in the CAD interface. The light blue regions are electrodes.

Note the following about the design:

- *Structure*

  This structure is created using the built-in Multilayer structure type. It consists of a 4 x 4 μm buried channel structure with an index of 1.01 in a uniform background of air.

- *Material*

  The material properties of the waveguides in this design are set to a material named *ElectroOptic*, which has a real refractive index of 1.01, as well as a electro-optic **DC Dielectric Constant** equal to the variable `Eps` and **Electro-Optic Coefficients** equal to the variables `RX` and `RY`. These settings can be seen in the Material Editor by selecting this material and looking at the Eps Linear and Electro-Optic tabs.

These parameters can be viewed using the **Display Material Profile** button on the left toolbar. Be sure to set **Material Property** to either *Index (Real)*, *DC Dielectric Constant* or *Electro-Optic Coefficients* to see each property.

- *Electrodes*

  Three electrodes have been defined in this simulation, and are displayed as light blue segments. Note the geometry and position information for each electrode, and that the **Index Profile Type** is set to *Inactive* so that they only are used to compute the DC electric field, and do not directly contribute to the index profile of the structure. Furthermore, look in the Additional Segment Options and note that the **Electrode/Heater Type** is set to *Electrode*, and the **E/H Control Parameter** is set to the applied voltage (`0` for the outer electrodes, and `V` for the inner electrode). For more information on defining electrodes, see the CAD manual.

- *Pathways and Monitors*

  Two pathways and two monitors are defined, one to measure the power in the launch mode in each arm.

- *Launch Field*

  The launch **Type** is *Computed Mode*, which indicates that the mode of the input waveguide at a wavelength of 1.0 μm should be computed whenever a simulation is run.

Testing the Electrodes

Once the structure is defined, it can be useful to determine the effect that the electrodes will have on the refractive index. Choose Utility/Multi-Physics from the CAD menu to open the Multi-Physics Utility. Make sure the **Physical Effect** is set to *Electro-Optic* and enter an **Output Prefix** such as `test` and press **OK**. When the calculation is complete, the results will be displayed in DataBROWSER.

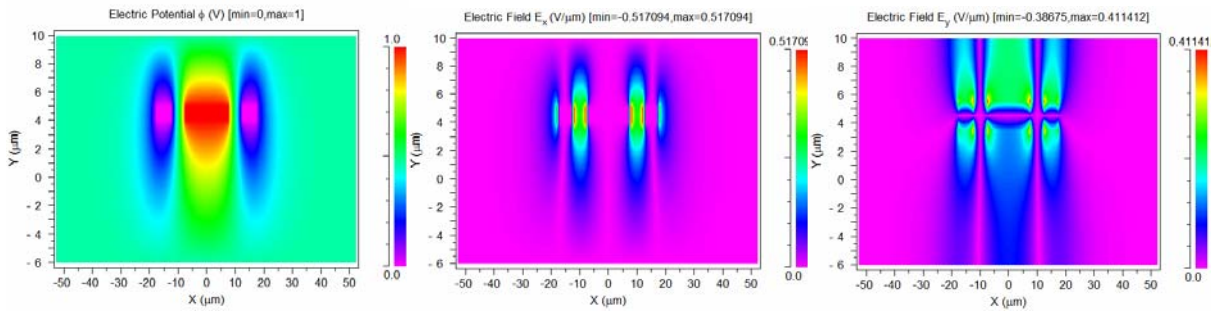

Figure 2: The computed a) electric potential for the electrodes, b) E_x, and c) E_y fields produced by the electrodes.

Several plots are output. Fig. 2 shows the electric potential as well as the E_x and E_y electric fields. The file `test.ppf`, shown in Fig. 3a, shows the absolute index perturbation that is generated by these electric fields via the electro-optic coefficients defined in the Material Editor. Figure 3b shows the signed data that shows that the index rises in the left waveguide and is reduced in the right waveguide. This change will create a phase difference between the propagating fields in the two waveguides and creates

constructive or destructive interference depending on the length of the arms and the magnitude of the index difference. The exact formulation of how the index perturbation is computed from the electric fields and material properties can be found in the CAD manual and is included in any simulations run.

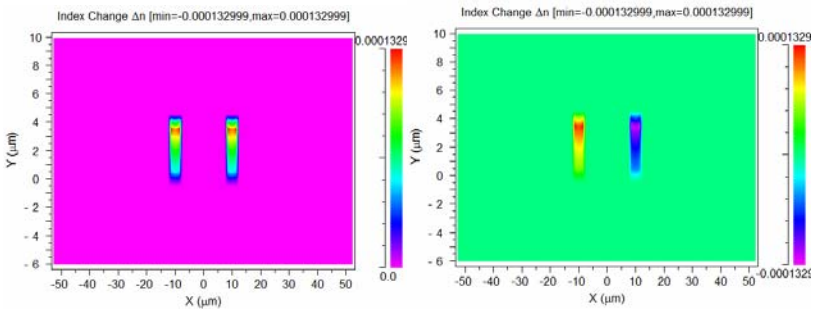

Figure 3: Computed index change for electro-optic effect. a) default absolute value plot, and b) modified plot to show signed data. The color scale was changed and the plot command "`/absy`" was removed.

Performing a Simulation

In this section, we will run several simulations to characterize the performance of the device. Click the **Perform Simulation** button in the left toolbar and click **OK** to start a simulation.

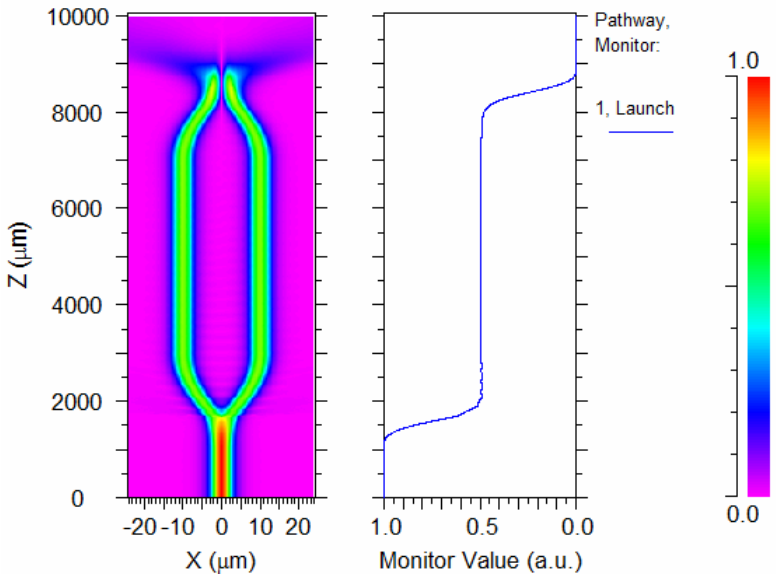

Figure 4: The completed simulation results at `V = 1`.

These results are at a single voltage point. More interesting results can be found by scanning over the applied voltage. Click the **Launch MOST** button in the left toolbar to open the MOST dialog, and note that a scan over the voltage `V` has been created. Click **OK** to start the scan. The scan results are shown in Fig. 5.

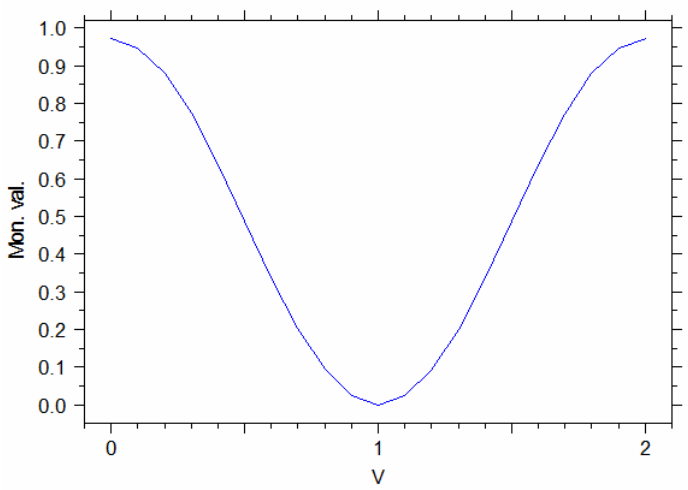

Figure 5: The variable scan results which show the output power vs. applied voltage.

Areas for Further Exploration

- Look at scans of the results of the output power versus `Larm`.
- Modify the material properties in the Material Editor to understand how they affect the simulation results.

Tutorial 3: Thermo-Optic Example

Thermo-optic effects can be modeled in steady state in BeamPROP by defining materials with thermo-optic parameters in the Material Editor, and then defining heaters in the CAD interface. This example studies a simple rib waveguide with a heater. We will use the Multi-Physics Utility to study the index perturbation caused by the heater.

The associated file for this tutorial is located in the `<rsoft_dir>\examples\Utilities\MultiPhysics\thermal\` directory.

Structure Overview

Open the file `soi_thermal.ind` in the CAD interface.

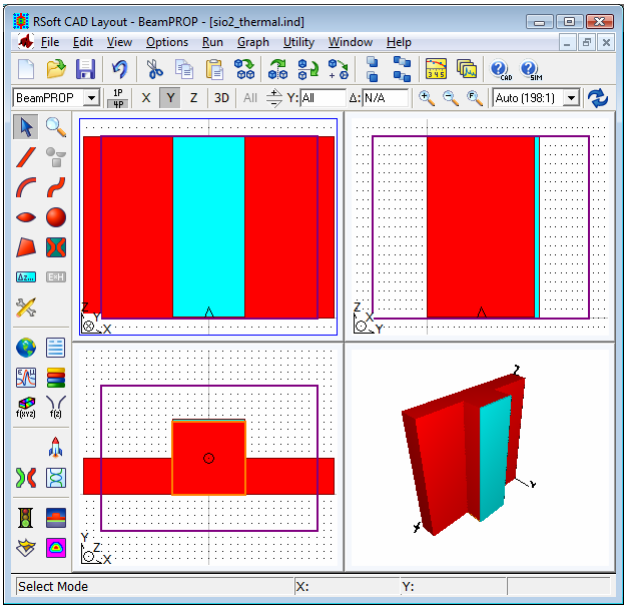

Figure 1: The simple SOI rib waveguide in the CAD interface. The light blue region is a heater.

Note the following about the design:

- *Structure*

  This structure is created using the built-in Multilayer structure type. It consists of a 2 x 2 μm SiO_2 rib waveguide, an SiO_2 slab with a height of 1 μm, a substrate, and an air cover. A layer table is used to define the rib and slab.

- *Material*

  The material properties of both the waveguide and slab are defined by the material *SiO2*, which has the optical and thermal properties of SiO_2. The cover material is *Air* and the substrate is defined via the material *Substrate* which has a **Thermo-Optic Coefficient** of `0` so it does not contribute to the

index change. These settings can be seen in the Material Editor by selecting these materials and looking at the Eps Linear and Thermo-Optic tabs.

These parameters can be viewed using the **Display Material Profile** button on the left toolbar. Be sure to set **Material Property** to either *Index (Real)*, *Thermal Conductivity,* or *Thermo-Optic Coefficient* to see each property.

- *Heater*

  One heater has been defined in this simulation, and is displayed as a light blue segment. Note the geometry and position information for the heater and that the **Index Profile Type** is set to *Inactive* so it is only used to compute the temperature distribution and does not directly contribute to the index profile of the structure. Furthermore, look in the Additional Segment Options and note that the **Electrode/Heater Type** is set to *Temperature* and the **E/H Control Parameter** is set to the temperature rise in degrees. For more information on defining heaters, see the CAD manual.

Using the Heater

We will now determine the effect that the heater will have on the refractive index. Choose Utility/Multi-Physics from the CAD menu to open the Multi-Physics Utility. Make sure the **Physical Effect** is set to *Thermo-Optic* and enter an **Output Prefix** such as `test` and press **OK**. When the calculation is complete, the results will be displayed in DataBROWSER.

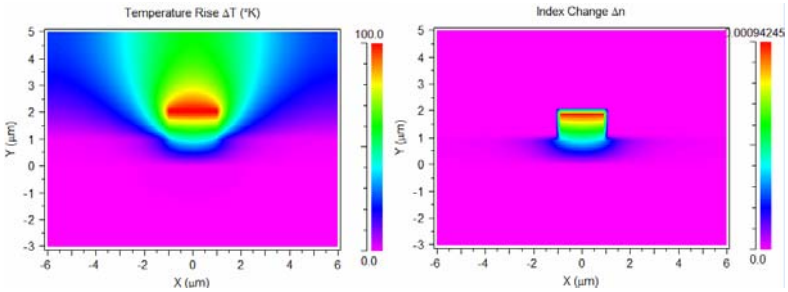

Figure 2: The computed a) temperature rise, and b) index change for the thermo-optic effect.

Several plots are output. Fig. 2a shows the temperature rise computed by the utility and the file `test.ppf`, shown in Fig. 2b, shows the absolute index perturbation that is generated by this temperature rise via the thermo-optic coefficients defined in the Material Editor. The exact formulation of how the index perturbation is computed from the temperature rise and material properties can be found in the CAD manual and is included in any simulations run.

Performing a Simulation

In this section, we will run several mode calculations to study how the effective index of the fundamental mode varies with the applied temperature. Click the **Launch MOST** button in the left toolbar to open the MOST dialog, and note that the simulation engine has been set to *BP Mode Solver* and a scan over the voltage `T` has been created. Click **OK** to start the scan. The scan results are shown in Fig. 5.

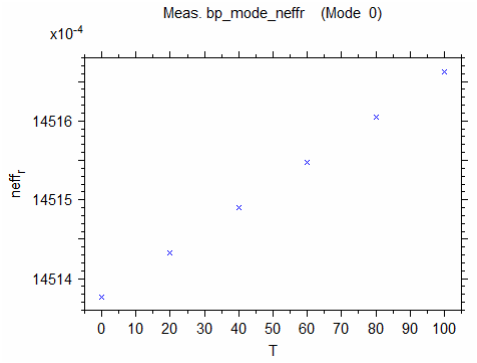

Figure 3: The variable scan results which show the effective index vs. applied temperature.

Areas for Further Exploration

- Create a multi-variable MOST scan over the temperature rise (variable `T`) as well as another structural parameter (such as the width or height of the waveguide).

Tutorial 4: Simple Stress-Optic Example

Stress-optic effects can be modeled in steady state in BeamPROP by defining materials with stress-optic parameters in the Material Editor. This example studies a simple fiber where the core and cladding have different stress parameters. We will use the Multi-Physics Utility to study the index perturbation caused by the stress.

The associated file for this tutorial is located in the `<rsoft_dir>\examples\Utilities\MultiPhysics\stress\` directory.

Structure Overview

Open the file `fib_stress.ind` in the CAD interface.

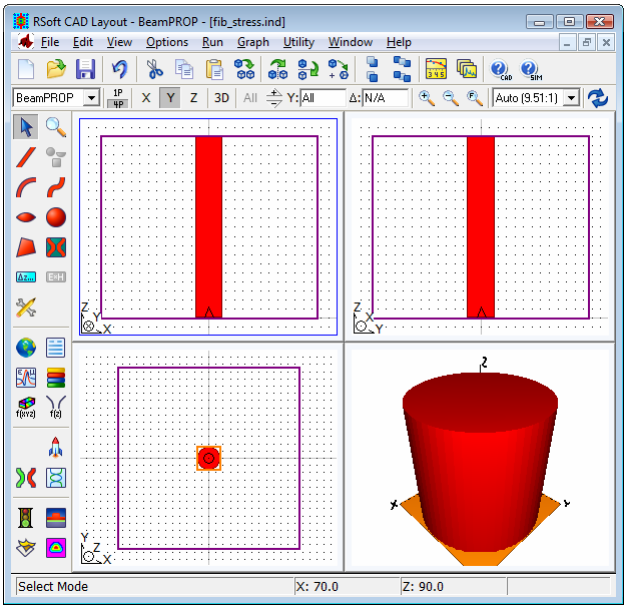

Figure 1: The simple SOI rib waveguide in the CAD interface. The light blue regions are electrodes.

Note the following about the design:

- *Structure*

  This structure is created using a single fiber component which represents the core embedded in a uniform background material which represents the cladding. The core has a diameter of 15.6 μm.

- *Material*

  The material properties of the core and cladding are defined by two materials: *SiO2_Core* and *SiO2_Clad*. The real refractive index of both of these materials is 1.51 and 1.5 respectively. Both of these materials have different stress parameters as well. These settings can be seen in the Material Editor by selecting these materials and looking at the Eps Linear and Stress-Optic tabs.

You can use the **Display Material Profile** button on the left toolbar to see cross-sections of these parameters. Be sure to set **Material Property** to either *Index (Real)* or *Stress Parameters* to see each property.

Viewing Effect of Stress on the Refractive Index

We will now determine the effect that the stress will have on the refractive index. Choose Utility/Multi-Physics from the CAD menu to open the Multi-Physics Utility. Make sure the **Physical Effect** is set to *Stress-Optic* and note the **Temperature Change** of `-1000` which indicates that the structure has cooled 1000 degrees since fabrication. Enter an **Output Prefix** such as `test` and press **OK**. When the calculation is complete, the results will be displayed in DataBROWSER.

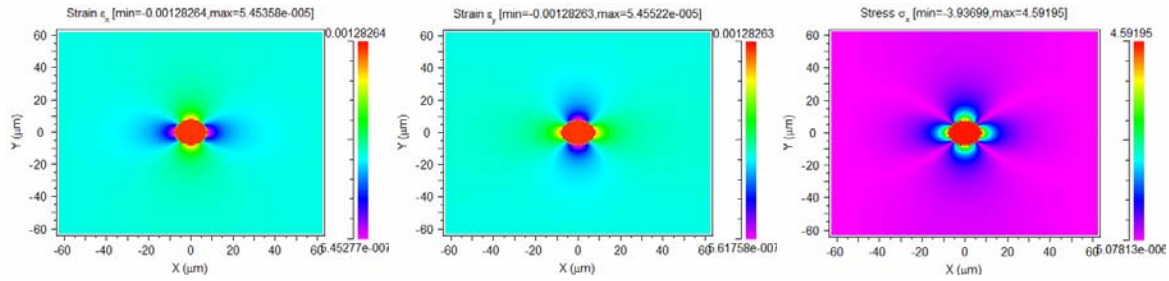

Figure 2: The computed a) X component of the computed strain, b) Y component of the computed strain, and c) X component of the computed stress.

Several plots are output and a select few are shown in Fig. 2. The file `test.ppf`, shown in Fig. 3a, shows the absolute index perturbation that is generated by the stress. The exact formulation of how the index perturbation is computed can be found in the CAD manual and can automatically be included in any simulations.

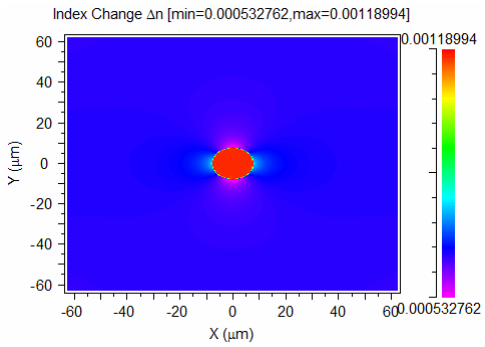

Figure 3: Computed index change for stress-optic effect.

Areas for Further Exploration

- Try computing the modes of the fiber and note how the stress-induced index perturbation changes the effective index of the modes.

Tutorial 5: Stress Effects on Birefringence

Stress-optic effects can be modeled in steady state in BeamPROP by defining materials with stress-optic parameters in the Material Editor, and then defining electrodes in the CAD interface. This example studies the effects of stress caused by cooling during fabrication on the birefringence of a simple SOI rib waveguide.

The associated file for this tutorial is located in the `<rsoft_dir>\examples\Utilities\MultiPhysics\stress\` directory.

The carrier-effects portion of the Multi-Physics Utility is licensed separately. Contact RSoft for details.

Structure Overview

Open the file `soi_stress.ind` in the CAD interface.

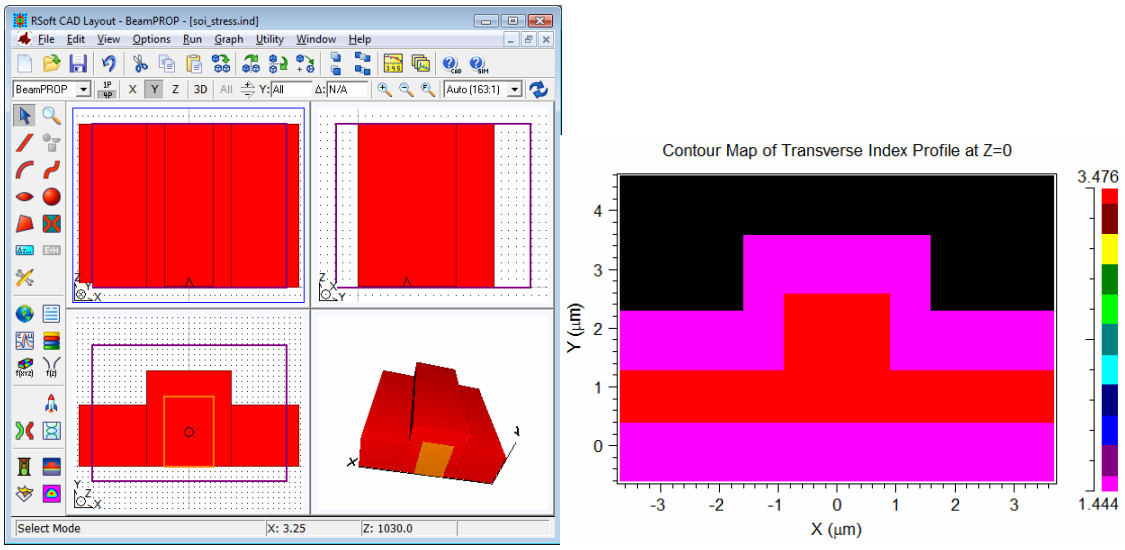

Figure 1: a) The simple SOI rib structure in the CAD interface, and b) the index profile of the waveguide.

Note the following about the design:

- *Structure*

  This structure is created using the built-in Multilayer structure type. Two segments are used, one to represent the rib and one to represent a slab. The result is the SOI structure as illustrated in Figure 1b. The etch depth is controlled by the variable `D`.

- *Material*

  The material properties of this structure are defined by four materials: *Air*, which is used for the cover, *Si* and *SiO2* which are used for the rib and slab regions, and *Substrate*, which is used for the substrate and has the optical properties equivalent to *SiO2* and the stress properties of *Si*. These

settings can be seen in the Material Editor by selecting these materials and looking at the Eps Linear and Stress-Optic tabs.

You can use the **Display Material Profile** button on the left toolbar to see cross-sections of these parameters. Be sure to set **Material Property** to either *Index (Real)* or *Stress Parameters* to see each property.

Computing Stress and Viewing Index Perturbation

Open the Multi-Physics Utility (choose Utility/Multi-Physics from the CAD menu) to compute the stress parameters within the waveguide as well as the stress-induced index perturbation. Make sure the **Physical Effect** is set to *Stress-Optic* and enter an **Output Prefix** such as `test` and press **OK**. When the calculation is complete, the results will be displayed in DataBROWSER. A select few of the plots are shown in Fig. 2.

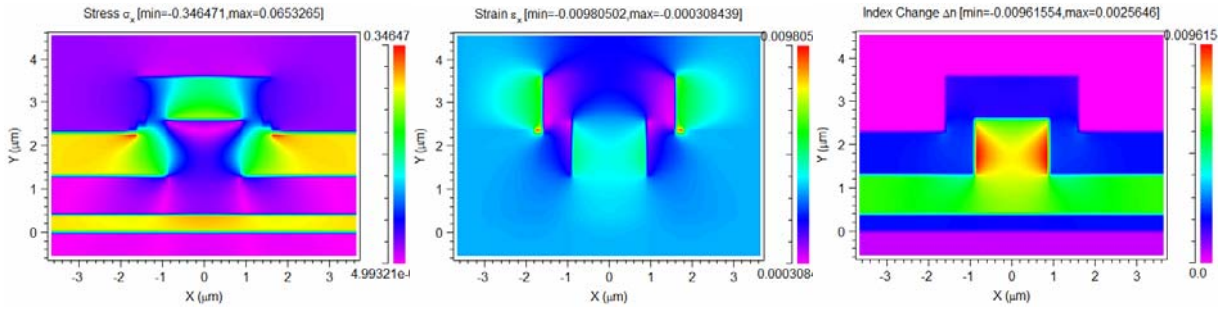

Figure 2: The computed a) x component of the stress, b) x component of the strain, and c) index perturbation.

The exact formulation of how the index perturbation is computed from the stress/strain can be found in the CAD manual and is included in any simulations run.

Studying Birefringence

We wish to study how the stress affects the birefringence of the waveguide, specifically as a function of the rib height controlled by the variable `D`. We could do this by performing several simulations at various values of `D` both with and without the stress-induced index perturbation and then collate all the results to produce plots of birefringence vs. `D`. We can, however, use MOST to automate this task. MOST normally only performs a single simulation at each scan point, but in this case we need 4 simulations: we need to solve for modes for both TE and TM polarizations with and without the stress-induced index perturbation. To accomplish this, we will use MOST's concept of a 'User Simulator' which is a script that tells MOST what steps to perform at each scan step, including the creation of custom outputs (in our case this would be the birefringence). We will then define custom MOST measurements to read in our custom outputs back into MOST so that nice scan plots can be automatically produced at the end of the scan. See the MOST manual for more information on using User Simulator scripts as well as the creation of user measurements.

Verifying Simulation Output

Before looking at the scan, it is important to verify that single simulations are giving reasonable results as well as to think through the algorithm needed to create the User Simulator in MOST.

- First, we will disable the inclusion of the stress-induced index perturbation in simulations to see the birefringence caused solely by the geometry of the waveguide. Open the Multi-Physics Utility and set the Control Parameter **Apply Index Perturbation** to *Never*. Click **Save Settings** and then **Cancel** to return to the CAD. Click the **Compute Modes** button in the left CAD toolbar, note that the **Polarization** is set to *TE*, and click **OK** to compute the fundamental mode. Note the resulting effective index and recomputed the mode for a *TM* polarization. The difference between these two numbers represents the geometric birefringence of the waveguide.
- Next, enable the inclusion of the stress-induced index perturbations in simulations by setting **Apply Index Perturbation** to *Default (Sim only)*. Rerun the same two mode calculations as before and note the change in the effective indices which represents the full birefringence of the waveguide. The difference between the geometric and full birefringence is then the stress birefringence.

These steps first show us that the results of the mode calculations are reasonable as well as give us an outline for the creation of a MOST User Simulator.

Creating the User Simulator Script

In order to codify the steps taken above to find the birefringence, we need to know the variables used to control the mode calculation, polarization, and whether or not the index perturbation is included in the simulation. These variables are `mode_set=0`, `polarization` with values of `0` and `1` for TE and TM, and `electrode_enable` with values of `0` or `1`. If you did not previously know the names of these variables, you could either look in the appendices of the appropriate simulation tool (BeamPROP in this case) or by simply making a setting in the GUI and then examining the Symbol Table for the appropriate variable and value.

Open the file `stressbiref.py` in your favorite text editor. This is a previously written cross-platform Python User Simulator script we can use to automate the algorithm outlined above on both Windows and Linux machines. We will not go through this in great detail (see the MOST manual for a complete description of User Simulator scripts) but we will mention a few important aspects here:

- The first lines parse the incoming command line into important parts using the RSoft Python library `rsoft.rspytools`. These lines are standard for a User Simulator. The executable name of BeamPROP is also determined as BeamPROP has a different name on Windows and Linux.
- The next lines set the base part of the simulation command which is common for all BeamPROP mode calculations. The `mode_set` variable indicates that the fundamental mode will be solved for.
- The next lines create four commands using the variables `polarization` and `electrode_enable` and run them using the `rspy.spawn()` command from the RSoft Python library. The results are stored to files `<prefix>_geom_TE`, `<prefix>_geom_TM`, `<prefix>_full_TE`, and `<prefix>_full_TM`.
- The final lines construct three `mathmat` commands to find the geometric, full, and stress birefringence values and pass them to the operating system for execution. See Appendix E in the

CAD manual for a description of `mathmat`. The output is saved in three files named `<prefix>_geom.biref`, `<prefix>_geom.biref`, and `<prefix>_geom.biref`. We will use a User Measurement in MOST to read these files and create nice scan plots.

Note that in this entire discussion we have not mentioned the variable `D`, which we will be scanning over, once. This is because the User Simulator script is independent of the variables scanned; they are automatically passed in the arguments to the script and, therefore, passed to each simulation command. This script could, if needed, be used to scan over any number of variables if needed with MOST.

Open the MOST dialog by clicking the **Launch MOST** button in the left CAD toolbar and note that the simulation tool is set to *User Simulator* and the `stressbiref.py` file has been specified.

Creating User Measurements

MOST automatically knows about standard outputs from RSoft simulators, but in our case, we have created three custom outputs: the geometric, full, and stress birefringence. We will have to therefore use three User Measurements to load these outputs back into MOST so nice scan plots can be created.

Click the Measurements tab in the MOST dialog and note the three measurements present: *Full_biref*, *Geometric_biref*, and *Stress_biref*. Click on *Full_biref* and note the information displayed. It is defined as a Scalar measurement that corresponds to the [0,0] data value in the file `<prefix_full.biref`, which is the file created by our User Simulator script. The other measurements are similarly defined to correspond to the appropriate data file.

Performing the Scan

A scan over `D` is set up in the MOST dialog. Enter an **Output Prefix** and click **OK** to start the scan.

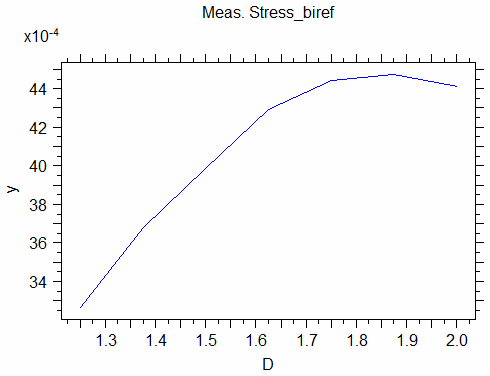

Figure 3: The scan results showing the change in birefringence due to stress as a function of `D`.

Areas for Further Exploration

- Modify the material properties in the Material Editor to see how they affect the simulation results.
- Scan over the waveguide width, then over the width and `D`. The script does not have to change.

Tutorial 6: Carrier Effects in an SOI Waveguide

This example demonstrates the use of the Multi-Physics Utility to simulate the carrier-induced index effects in a SOI ridge waveguide.

The associated file for this tutorial is located in the directory:

```
<rsoft_dir>\examples\Utilities\MultiPhysics\carrier
```

The carrier-effects portion of the Multi-Physics Utility is licensed separately. Contact RSoft for details.

Structure Overview

Open the file `soi_carrier.ind` in the CAD interface.

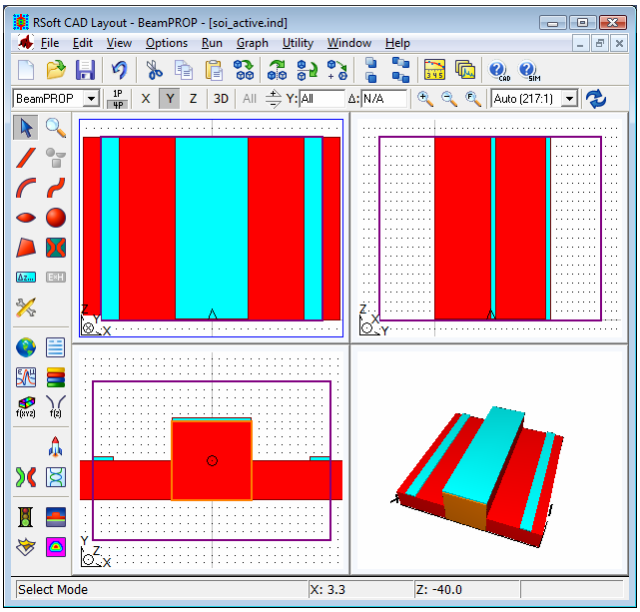

Figure 1: The SOI structure shown in the CAD interface. The light blue regions are electrodes.

Note the following about the design:

- *Structure*

  This structure is created using the built-in Multilayer structure type. It consists of a 2 x 2 μm rib waveguide with a slab height of 1 μm in a background of SiO_2. Note that an inactive segment has also been drawn to represent the slab. This is not necessary for the BeamPROP part of the simulation, but is necessary when using the carrier-effects portion of the Multi-Physics Utility.

- *Materials*

  The waveguide and ridge are composed of two layers that have been defined using the materials *Si_ntype* and *Si_ptype* which correspond to n-type and p-type Silicon respectively. These materials were defined using the Material Editor in the CAD. They are based on the built-in *Si* material and the only modification from default values comes from the doping choice. Open the Material Editor,

choose one of the Si materials, move to the Semiconductors tab, click the **Edit Param…** button, and note the doping concentration setting: Donors for n-type silicon, and Acceptors for p-type silicon.

- *Electrodes*

  Three electrodes have been defined in this simulation, and are displayed as blue segments. Note the geometry and position information for each electrode, and that the **Index Profile Type** is set to *Inactive*. Furthermore, look in the Additional Segment Options and note that the **Electrode/Heater Type** is set to *Electrode*, and the **E/H Control Parameter** is set to the applied voltage (`0` for the outer electrodes, and `V` for the inner electrode). For more information on defining electrodes, see the CAD manual.

- *Pathways and Monitors*

  One pathway and two monitors are defined using the rib segment.

Setting the Multi-Physics Utility Parameters

Once the structure is defined, the Multi-Physics Utility must be configured. Choose Utility/Multi-Physics from the CAD menu to open the Multi-Physics Utility. Make sure the **Physical Effect** is set to *Carrier-Effects* and enter an **Output Prefix** such as `test` and press **OK**.

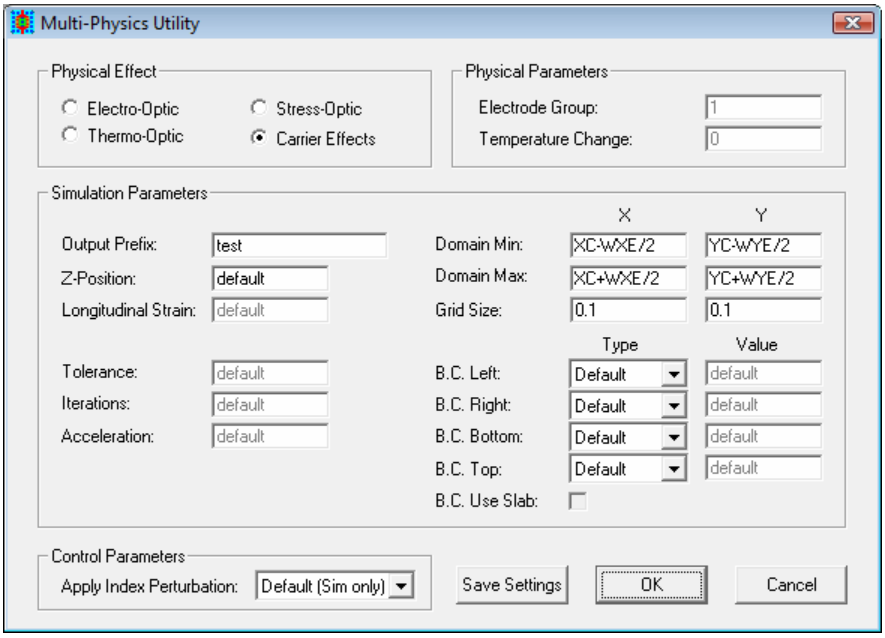

Figure 2: The Multi-Physics Utility.

The LaserMOD simulation engine will be called to determine the electronic transport within the device and the results are shown in Fig. 3. The resulting electron and hole distributions will be used to update the refractive index profiles for use in the optical simulation since **Apply Index Perturbation** is set to *Default (Sim only)*. They will not, however, be included in any index profile calculations unless the appropriate setting is made.

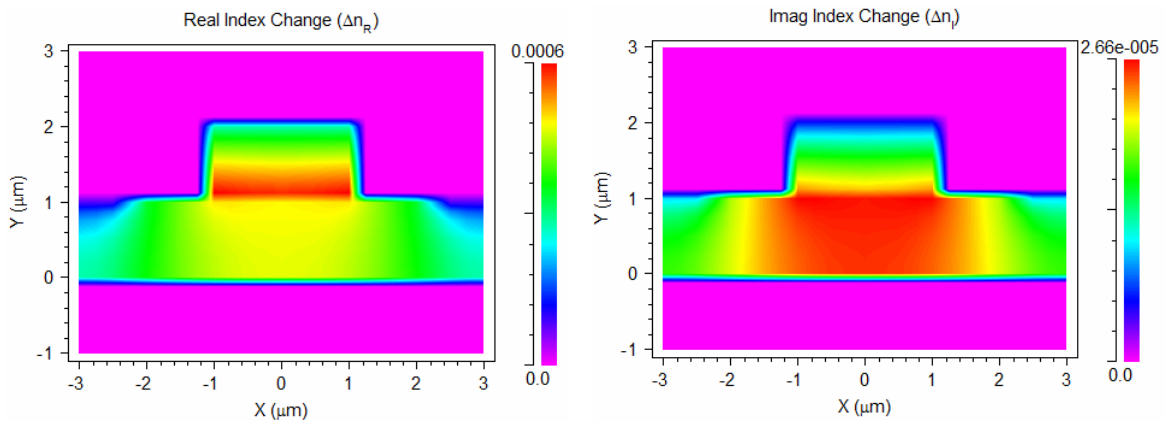

Figure 3: The computed index perturbation: a) real index change, and b) imaginary index change.

Performing a Simulation

Finally, the optical simulation will be performed. For this example we will study the change in the effective index of the waveguide mode that results from the inclusion of carrier effects. For most modulator problems, this is the key point of interest as the change in effective index with applied bias determines the optical path length change in the device, and hence its modulation characteristics. Click the **Compute Modes** button on the let CAD toolbar to open the Mode Calculation Parameters dialog. Note that the simulation domain and grid sizes have been previously set and click OK to star the simulation. The LaserMOD simulation engine will once again be invoked, the index perturbations will be determined and added to the refractive index profile of the circuit, and then the mode calculation will begin automatically. The result of the mode calculation for an applied voltage of `V = 1` is shown in Fig. 4a with an effective of 3.440083.

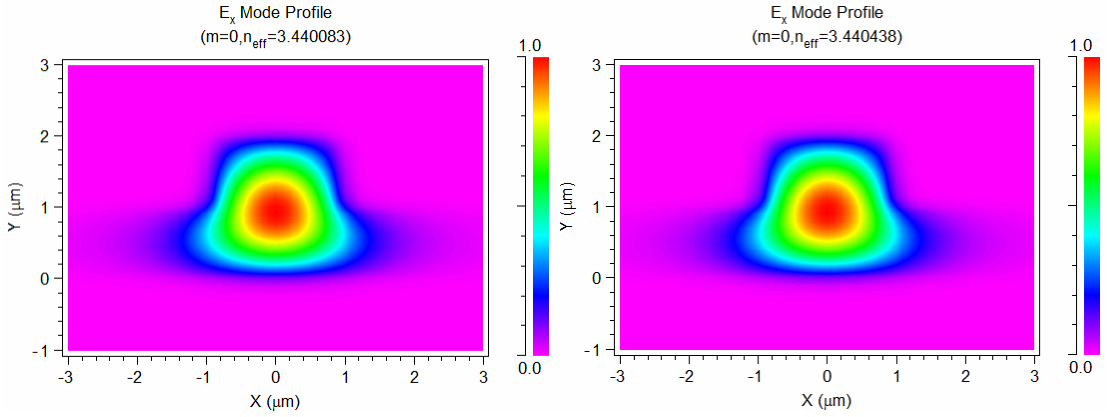

Figure 4: Computed mode for applied bias of a) 1V and v) 0V.

Rerun the simulation with `V` set to `0`. Rerun the mode calculation. The results, shown in Fig. 4b, show an effective index of 3.440438 which is a relatively small change and therefore requiring a long waveguide to effect modulation. Through the use of the Multi-Physics Utility, device performance can be optimized.

Appendix A

Tips and Traps

This appendix contains some advice on good habits and sources of confusion for novices and experienced BeamPROP users alike.

A.A. Common BeamPROP mistakes

- *Choosing large grid/step sizes results in inaccurate results.*

  The grid sizes used for a simulation should be small enough to resolve the fields present in the simulation. Since BeamPROP essentially solves for a steady state solution at one frequency, the smallest field feature is not the wavelength, but usually given by the index distribution and geometry of your structure. For instance, a large index contrast indicates sharp field features which then have to be resolved via the simulation grid sizes. A convergence study should be used to determine an optimal value of a simulation parameter.

- *Misuse of advanced features.*

  The advanced features in BeamPROP must be used properly in order to achieve correct results. As discussed in Chapter 2, many of the advanced features extend the basic BPM algorithm to include additional physical effects. If these effects are of interest to you, these advanced features are necessary. However, if these effects are not of interest to you, these features add unnecessary complications to the computation and increase simulation time. These features should not be used unless necessary.

- *Not setting the reference k correctly.*

  The **Reference k** value sets the characteristic wavenumber for the simulation, and must be set to an accurate number to get accurate results.

A.B. Some Good BeamPROP 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.*

  In order to obtain the desired results from a simulation, it is important to use the correct launch condition.

- *Check the monitor configurations.*

  Again, in order to obtain desired results from a simulation, the monitors need to be configured properly. If they are not, the results they return will of course be incorrect.

- *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. 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.

- *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.

- *Increase the slice time for long simulations.*

  It is usually unnecessary to have BeamPROP continually update the graphical display of the fields during a long simulation. While it can be beneficial to watch the fields evolve for short simulations, the continual update of the simulation window steals valuable CPU time from the actual simulation. Increase the value of the **Slice Grid** in X, Y, and/or Z in order to avoid making a long simulation even longer. Of course, if you need to save the fields, you cannot do this.

- *Learn the scripting capabilities of BeamPROP.*

  While BeamPROP 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. BeamPROP's scripting capabilities are based on the fact that simulations can be run from the command line. Because of this unique capability, BeamPROP can utilize any native scripting language within the OS you are using. For instance, Windows users can use simple DOS batch files to run BeamPROP, while advanced users who may have standard scripting languages installed on their systems 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 scales 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.

Appendix B

BeamPROP File Formats

This appendix documents the file formats used in BeamPROP. 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, mode profiles, index profiles, and other output types. This format is described in detail in the Appendices of the CAD manual.

B.B. Pathway Monitor File Format

Monitor files are generated by the simulation program when monitors have been defined and monitor output has been selected, which is the default. These files have the extension `*.mon`. The data they contain describes the defined monitor quantities as a function of propagation distance (Z position). The first five lines can be considered comments and ignored. The rest of the file consists of columns of numbers. The first column contains the Z position, and subsequent columns contain each monitor value.

B.C. Correlation File Format

Correlation files are generated by the simulation program when a mode spectrum is calculated, provided that correlation output has been enabled in the Output Options dialog. These files have the extension `*.cor`. They contain the overlap integral of the input field with the computed field as a function of propagation distance.
The correlation function is a 1D complex valued function, and is in the `OUTPUT_REAL_IMAG` format. The first non-directive line contains seven fields which are:

- The number of data points in the z direction
- The Z coordinate of the first data point
- The Z coordinate of the last data point
- The free space wavenumber
- The reference refractive index ($n_{bar}=k_{bar}/k_0$)
- The minimum refractive index
- The maximum refractive index

This line is followed by the real and imaginary parts of the correlation as in the `OUTPUT_REAL_IMAG` file format.

B.D. Mode Spectrum File Format

Mode spectrum files are generated by the simulation program when a mode spectrum is calculated. These files have the extension `*.fft`. They contain the Fourier transform of the correlation function, with the frequency defined by the effective index minus the reference index.
Their format is similar to the field file format described above for the `OUTPUT_AMPLITUDE` case, except that the units are power not amplitude, and that the first non-directive line contains seven fields which are:

- The number of data points in the "frequency" (n_{eff}-n_{bar}) direction
- The "frequency" coordinate of the first data point
- The "frequency" coordinate of the last data point
- The free space wavenumber
- The reference refractive index ($n_{bar}=k_{bar}/k_0$)
- The minimum refractive index
- The maximum refractive index

This line is followed by the power spectrum as in the `OUTPUT_AMPLITUDE` file format.

B.E. Mode Results (Effective Index) File Format

A mode result file (`<prefix>.nef`) is produced by BeamPROP whenever a mode computation is performed. It contains a list of the modes found, and includes the mode number, the real part of the effective index, and the imaginary part of the effective index. Note that these values are the corrected values discussed in Section 5.B.3.

B.F. *.mds File Format

A `*.mds` file contains raw data generated by the simulation program whenever any mode calculation is performed. The first non-directive line contains five fields which are:

- The number of modes found
- The free space wavenumber
- The reference refractive index ($n_{bar}=k_{bar}/k_0$)
- The minimum characteristic refractive index.
- The maximum characteristic refractive index

This line is followed by a list of the mode information, each line corresponding to a mode. Each line has eight columns which represent the following data:

- The mode number, which simply enumerates the peaks from high to low frequency
- The mode effective index, given relative to the reference index (*i.e.* n_{eff}-n_{bar}). This is the uncorrected value.
- The mode effective index, obtained by simply adding the reference index to the above. This again is an uncorrected value.
- The mode effective index, obtained more accurately through the 'first correction' discussed in Section 5.B.3. This field is the same as the above field for the iterative method.
- The normalized mode effective index (*i.e.* the so-called *b* parameter)
- The mode weight, given as a fraction of power in the input field (0 for iterative method)
- The imaginary part of the mode effective index. This is not valid in all cases and be set to `0`. For example, the imaginary part is not calculated if only the mode spectrum is calculated as the method requires the mode profiles themselves be computed. Also, imaginary effective index output is disabled when the variable `neff_method` is set to `0`.
- The real part of the mode effective index as obtained via the 'second correction' described in Section 5.B.3. For the iterative method, this is just a duplicate of the 3rd and 4th columns. For the correlation method, this is the value found by substituting the mode profiles in the wave equation.

The values displayed in any output data or plot files will correspond to the values in the 7th and 8th columns.

B.G. Mode Field File Format

Mode field files are generated by the simulation program when one or more modes are calculated. These files have the extension `*.m##`, where `##` is the mode number. They are in the Standard RSoft File Format described above. Depending on the simulation settings, multiple components may be output.

Note that for mode field files, the optional fields in the field file format documented above are used and contain the real and imaginary parts of the mode effective index. The effective index information is also stored and displayed in the corresponding `*.p##` graph file that are generated.

B.H. Refractive Index and Loss Profile File Formats

Refractive index or loss profile files are generated by the simulation program when the index or loss profile is calculated. These files have the extensions `*.ipf` and `*.lpf`, respectively. They contain the real or imaginary part of the refractive index as a function of the transverse coordinates. They are in the Standard File Format described above.

Appendix C

Symbol Table Variables

In this appendix we document the symbol table variable names which control the internal operation of BeamPROP. Most of these variables correspond to dialog fields which have already been documented. The following table lists the variable names corresponding to these dialog fields.

C.A. BPM Simulation Parameters

The following variables correspond to the fields found in the BPM Simulation Parameters dialog box:

Field Name	Variable
X Domain Min	`boundary_min`
X Domain Max	`boundary_max`
X Grid Size	`grid_size`
X Slice Grid	`slice_grid_size`
Y Domain Min	`boundary_min_y`
Y Domain Max	`boundary_max_y`
Y Grid Size	`grid_size_y`
Y Slice Grid	`slice_grid_size_y`
Z Domain Min	`domain_min`
Z Domain Max	`domain_max`
Z Grid Size	`step_size`
Z Slice Grid	`slice_step_size`
Z Monitor Grid	`monitor_step_size`
Display Mode	`slice_display_mode`
Output 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. Advanced Numerical Parameters

The following variables correspond to the fields found in the Advanced Parameters dialog box:

Field Name	Variable
Numerical Scheme	`scheme`
Boundary Condition	`bc_type`
Boundary Location	`boundary`
Boundary Gap	`boundary_gap`
Scheme Parameter	`scheme_parameter`
Padé Order	`pade_order`
Reference k	`kbar`
Reference k Value	`kbar`
Reference k Angle	`kbar_angle`
Reference k Angle Value	`kbar_angle`

C.D. Bi-Directional Parameters

The following variables correspond to the fields found in the Bi-Directional Parameters dialog box:

Fieldname	Variable
Operator	`bd_operator`
BD Pade Order	`bd_pade_order`
Imag Parameter	`bd_imag_parameter`
Iteration	`bd_iteration`
Tolerance	`bd_tol`
Max Iterations	`bd_max_iterations`
Damping Factor	`bd_damp_factor`

C.E. Display Options

The following variables correspond to the fields found in the Display Options dialog box:

Field Name	Variable
Slice Colors	`color_slices`
Circuit Color	`color_circuit`
Slice Normalization	`slice_normalization`
Slice Position X	`slice_position_x`
Slice Position Y	`slice_position_y`
Monitor Complement	`monitor_complement`
Monitor Log Scale	`monitor_log_scale`
Monitor Range Min	`monitor_min`
Monitor Range Max	`monitor_max`
Index Min	`index_min`
Index Max	`index_max`
Surface Color	`color_surface`
Outline Color	`color_outline`
View Phi	`view_phi`
View Theta	`view_theta`
Color Shades	`color_shades`
Color Scale	`color_scale`
Show Color Scale	`color_scale_display`
Allow Redraw 2D	`allow_redraw_2d`
Allow Redraw 3D	`allow_redraw_3d`
Plot Aspect Ratio	`plot_aspectratio`

C.F. Output Options

The following variables correspond to the fields found in the Output Options dialog box:

Field Name	Variable
Slice Output Format	`slice_output_format`
Field Output Format	`field_output_format`
Mode Output Format	`mode_output_format`
Far Field Output	`farfield_output`
Monitor Output	`monitor_output`
Correlation Output	`correlation_output`
Run Log Output	`run_output`

C.G. 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
Launch Pathway	`launch_pathway`
Launch Type	`launch_type`
Launch Tilt	`launch_tilt`
Launch Polarization	`launch_polarization`
Launch Mode	`launch_mode`
Launch Mode Radial	`launch_mode_radial`
Random Set	`random_set`
Input File (E-Major)	`launch_file`
Input File (E-Minor)	`launch_file_minor`
Launch Normalization	`launch_normalization`
Launch Background N	`launch_background_index`
Launch Delta N	`launch_delta`
Launch Angle	`launch_angle`
Launch Width	`launch_width`
Launch Height	`launch_height`
Launch Position X	`launch_position`
Launch Position Y	`launch_position_y`
Launch Neff	`launch_neff`
Launch Align File	`launch_align_file`
Launch Power	`launch_power`
Launch Phase	`launch_phase`

C.H. Mode Solving

The following variables correspond to the mode solving features discussed in Chapter 5.

Mode Calculation Options

The following variables correspond to the fields in the Mode Calculation Options dialog:

Field	Variable
Method	`mode_method`
Neff Log	`neff_log`
Neff Tolerance	`neff_tol`
Neff Min	`char_nmax`
Neff Max	`char_nmin`

Advanced Mode Solving Features

The following variables correspond to advanced mode-solving features discussed in Chapter 6:

Variable	Description
`mode_threshold`	The minimum threshold above which modes peaks are found from a mode spectrum.
`idbpm_convergence_warning`	Toggles on/off the warning that an iterative mode calculation has not converged.
`mode_spectrum`	Indicates that a mode spectrum should be computed.
`mode_set`	Indicates the modes to be solved for.
`idbpm`	Computes fundamental mode.
`step_size_idbpm`	Z Compute Step for mode calculations.
`step_size_idbpm_factor`	The safety factor used to determine the default step sized used for iterative mode solving calculations. This factor should normally be less than one, and defaults to `0.5`.
`mode_length`	Sets the domain length for a mode calculation.
`neff_tol_cor`	Sets the desired tolerance for the effective indices for a correlation mode solving calculation.
`neff_scan_output`	Sets the type of output for a mode scan. Setting this parameter to `0` disables output, `1` enables output of real indices, and `2` enables output of real and imaginary indices.

C.I. Boundary Conditions

The following variables correspond to different boundary conditions:

Variable	Description

`boundary_path_right`	Identical to the `boundary_path` variable above, but for only for the right boundary.
`boundary_path_left`	Identical to the `boundary_path` variable above, but for only for the left boundary.
`bc_type_left`	Sets the boundary condition for the minimum X boundary.
`bc_type_right`	Sets the boundary condition for the maximum X boundary.
`bc_type_bottom`	Sets the boundary condition for the minimum Y boundary.
`bc_type_top`	Sets the boundary condition for the maximum Y boundary.

C.J. Additional Features

The following variables correspond to the advanced features discussed (mainly) in Chapter 6:

Variable	Description
`hfield`	Toggles between the use of the E and H fields.
`vector_conserve`	Can sometimes remedy difficulties with power conservation.
`kbar_path`	Sets the pathway to be used to calculate the Reference K Angle.
`electrode_enable`	Includes the effects of electrodes in a calculation.
`azimuthal_mode`	Controls the azimuthal mode number for a 2D Radial BPM simulation.
`monitor_decay_function`	Sets the characteristic transverse decay constant for a WG Power monitor.

Appendix D

Release Notes

This appendix summarizes the changes from previous versions of BeamPROP. Additionally, since the documentation for the RSoft CAD Layout tool was found in the BeamPROP manual prior to version 5.1, changes in the CAD are also found in this section.
Within each section, the first subsection discusses new or improved features, and the second subsection covers compatibility issues.

Changes from Version 8.2 to Version 8.3

Note that the change log which details the history of changes from Version 8.2 to Version 8.3 is available in the file `READMEBP83.TXT`.

- Modified field and mode file output names to include the field component contained in the file.
- New launch file and pathway monitor data file specification to now accept launch files based on a file prefix and extension instead of providing individual files for each necessary field component. This feature works well with the previous feature.
- Added a default output prefix for BeamPROP simulations.
- New controls added to the top of the mode simulation window to select the field component and mode displayed.
- The far-field output has been modified to output the results as a true far-field intensity profile utilizing all required field components, including implicit field components that are not normally calculated directly. Additionally, plot plots of the far-field are also output. The results should match those of FullWAVE.
- Improved FFTs which are more efficient and handle non-power-of-2 data lengths to avoid anti-aliasing effects. This effects mode solving via the Correlation method, FFT-based BPM, and far-field calculations.

- Pathway monitors now pay attention to rotation out of the XZ plane (theta).
- New 'View Launch' button in the Launch Properties dialog.
- Major improvement to the 3D semi-vector algorithm which provides better conservation of power, especially in cases involving high index contrast. These simulations now require H field input; .ind files created with older versions that use data files for launch or overlap monitors may be affected. In these cases, recompute any mode/field data files used as the H field is now automatically output. Contact RSoft with any problems.

Changes from Version 8.1 to Version 8.2

Note that the change log which details the history of changes from Version 8.1 to Version 8.2 is available in the file `READMEBP82.TXT`.

- There has been a naming change in the monitors that should provide better clarity to their actual operation. These analysis objects were previously referred to as "Monitors"; henceforth they would be called "Pathway Monitors".
- 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.
- A new Gaussian offset feature which allows users to launch the beam off its waist.

Changes from Version 8.0 to Version 8.1

Note that the change log which details the history of changes from Version 8.0 to Version 8.1 is available in the file `READMEBP81.TXT`.

New Capabilities and Improvements to the Program

- Improved Material Editor. See CAD manual for more details.
- New far-field option for periodic structures.

Significant Changes in Program Behavior

- New labeling of mode and field plots when using polarization. The actual field component is now displayed in the plots.
- New aximuthal mode numbering to be consistent with other RSoft products.

Changes from Version 7.0 to Version 8.0

Note that the change log which details the history of changes from Version 7.0 to Version 7.1 is available in the file `READMEBP80.TXT`.

New Capabilities and Improvements to the Program

CAD:

- Improved CAD interface. See CAD manual for more details.

Significant Changes in Program Behavior

- Added a control for the H-field formulation in the Advanced Parameters dialog.
- Moved polarization and vector mode options to the Simulation Parameters dialog.
- The bidirectional options are now accessible from the Advanced Options dialog.

Changes from Version 6.0 to Version 7.0

Note that the change log which details the history of changes from Version 6.0 to Version 7.0 is available in the file `READMEBP70.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.
- Integrated BeamPROP with the Material Editor.

Simulation:

- Improved the convergence of the iterative mode solver in some cases.

Other:

- Improved dialog for mode computation.

Significant Changes in Program Behavior

- Addition of new output file for mode calculations which simplifies the output of mode effective indices. The new file is named `<prefix>.nef`.
- Changed the default number of grid points used for the slice grid.

Changes from Version 5.1 to Version 6.0

Note that the change log which details the history of changes from Version 5.1 to Version 6.0 is available in the file `README60.TXT`.

New Capabilities and Improvements to the Program

CAD:

- Correction to index profile for overlapped channel waveguides with negative side slope. Previously, in some cases, this combination could yield cutoff portions of the channel. The problem has been corrected.

Simulation:

- A new simulation variable, `step_size_idbpm_factor`, has been added. For the iterative mode calculation method, the default step size is determined from a theoretical ideal, and multiplied by a safety factor (which should normally be less than one). The safety factor was fixed at 0.5, but can now be changed.
- A new simulation variable, `mode_length`, has been added. This variable sets the length for mode calculations independently of Z Domain Max.
- A new feature to output the imaginary part of the effective index calculated during a mode scan. Normally, only the real part of neff is output. By setting the variable `neff_scan_output` to the value `2` in the symbol outputs, the imaginary part will be output as well.

Analysis/Results:

- Extension of far field output option to include phase.
- New option to output the H field when doing mode calculations via the variable `hfield_output`.
- A change to the output of the `.mds` file which accomplishes several things. First, the precision of some columns is improved. Second, the file now regularly includes the 7th column for the imaginary part. Note, however, that this will not be valid in some cases and will just contain 0. Third, the file now includes an 8th column which is the real part of neff.
- Enabled output of imaginary part of index profile for Anisotropic case. Note that to enable output of all tensor components, `index_profile_all=1` must be set.

Other:

- A small correction to the `stargen` utility for AWG layout.
- Corrected an error in the `disperse` utility which produced incorrect results for some combinations of options.

Significant Changes in Program Behavior

- The `vector_conserve` option, introduced in Version 5.0, has been enabled by default and added to the Advanced Parameters dialog. The default for this option may be changed by the user by setting `VectorConserve=1` in the [BSIM] section of the `BCADW32.INI` file.
- A new method for overlap power calculations in 3D vector BPM has been introduced. This method applies a more rigorous vector formulation to determine overlap integrals, and is enabled by

default. If necessary, this method may be disabled by setting `overlap_eh=0` in the symbol table for an individual `.ind` file, or setting `OverlapEH=0` in the [BSIM] section of the `BCADW32.INI` file to disable it by default.

- A change in power calculations for full vector BPM. The new method calculates power more accurately for large minor component, and separates major and minor components of the power in a more useful way.
- Added an option to exclude the boundary points in power or overlap calculations. Rigorously, the theorems for power conservation in BPM exclude the boundary points, however in practice the normally do not contribute significantly. However, in some rare cases, these points can diverge briefly, and the power reported by BeamPROP demonstrates a spike. The new option avoids this problem by excluding these points, and is enabled by default. To disable it, you can set `overlap_exclude_boundary=0` in the symbol table of an individual `.ind` file, or `OverlapExcludeBoundary=0` in the [BSIM] section of the BCADW32.INI file to turn it off by default.
- Added an option to improve mode convergence in some cases. Previously, in rare cases, the iterative method would prematurely converge and yield an unphysical value for the effective index. The new option `neff_clip=1`can be used to avoid this problem. Note that it should only be used if there appears to be a problem, since it can exclude valid modes in some cases.

Changes from Version 5.0 to Version 5.1

Note that the change log which details the history of changes from Version 5.0 to Version 5.1 is available in the file `README51.TXT`.

New Capabilities and Improvements to the Program

CAD:

- Improved export of components employing the new Segment Orientation feature. Previously, these were only exported correctly if the Export Polygon Mode was Raw; otherwise they were exported more or less as if the Segment Orientation had been ignored. Now they are exported correctly in all cases, however Raw export is always a truer representation but may exceed the 200 point GDS-II polygon limit.
- A new GDS-II import option. Note that the importer does not currently support PATH and NODE elements.
- Support for LWPOLYLINE in DXF import and an improvement allowing import of closed polylines which are missing the final point.
- Improved clipping of objects in the CAD an simulation windows.
- A feature for easier editing of 3D structures involving overlapping components in different y planes using the variable `cad_yselect_size`.

- Export of concave lens objects has been corrected.
- Updated the Symbol Table Editor dialog to have a wider window for displaying the current value. This avoids cutting off the display of very small or large numbers that need to be shown in scientific form.
- New functions that can be used in expressions for generating random sequences and lines.

Simulation:

- Performing a Mode Spectrum scan has been altered to allow the range of modes (i.e. Mode Set) to be selected. When this is done, only these modes will be output to the `.nes/.pne` file for the effective index data.

Analysis/Results:

- A new option in the scan variable dialog called "Plot Dynamically". This option causes the results of a single variable scan to be displayed and updated dynamically as each run is completed. Depending on user feedback, this option may become the default.
- The WDM Router Simulation dialog now generates additional scripts.
- Added rotation option to `BDUTIL.EXE`.

Other:

- Updated the XZ and XYZ array generators to enable producing layouts centered at z=0 rather than starting at z=0.
- Increased the maximum allowed characters per line in WinPLOT to 80,000 from 16,000. This will allow larger data files to be opened in WinPLOT.
- The EXAMPLES directory has been restructured so that it no longer contains any files directly.
- RSoft has developed data file converters between BeamPROP and Zemax, similar to the existing converters between BeamPROP and CODEV. While these will be officially released in an upcoming version, any user requiring these converters may contact RSoft to obtain them by email.

Significant Changes in Program Behavior

- Faster calculation for 3D simulations with large numbers of fiber or channel structures (e.g. PBG fibers).
- A new option which allows mode calculations via the correlation method to calculate the imaginary part of the effective index if the mode is leaky. Even for normal guided modes, the new option generally results in slightly more accurate values for the effective index. The new option is enabled by default, but can be controlled by the variable `neff_method`.
- For radial simulations, updated the 1/e monitor to calculate the width of the field in full Cartesian coordinates, and not in radial coordinates. The monitor will now give values which are double the

values obtained with the previous versions. If you experience any difficulty with this new method, please contact RSoft.

- Improvements to the licensing mechanism, especially in the case of Network licenses which now allow multiple copies of the CAD program to be opened on the same machine, even when only one license is available.
- Changed the behavior of the SaveAs dialog to change the working directory to the specified directory.
- The electronic licensing mechanism has been updated. Now all licensing is done through license files which must be obtained by email from RSoft (support@rsofdesign.com).
- Distribution of new code based on DLL's. This is being done to enable more dynamic updates of dialogs for addons such as BandSOLVE and GratingMOD.
- A major update to the launch capabilities has been made.
- An error which would cause the program to crash if the number of modes found in the iterative method exceeded the maximum mode limit has been corrected. The program will now simply terminates gracefully after finding the largest mode allowed, and prints a warning message to the log file. If more modes are desired, you can change the limit as noted above.
- A change in the behavior of the scan dialog to warn if the scan variable does not exist (or is not an internal variable). This warning can be disabled by setting `scan_variable_warning=0`.

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 `README40.TXT`.

New Capabilities and Improvements to the Program

CAD:

- Added a major new feature to allow BeamPROP's generalized segment (including most tapers and arcs) to have a reference orientation along the axis of the segment. Use of this feature provides a means of achieving geometries or effects which were previously awkward or impossible to obtain, including arbitrary segment rotation, endfaces perpendicular to the segment, width tapers referenced to the segment axis, and arcs up to 90 degrees which maintain shape and endface orientation.
- Implemented use of the above in the layouts generated by the WDM Router (AWG) module.
- Eliminated arbitrary limitations on the number of components (previously limited to 20,000).
- 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

- The number of user tapers/profiles has been increase to 16 each, and the CAD interface for accessing them has been improved.
- The number of symbols has been increased from 2048 to 3000, and it use configurable if necessary.
- The Polygon and Lens dialogs have been expanded to allow additional control over these objects.
- Added CAD import options for DXF and CIF files.

Simulation:

- Implemented a major new feature for "Simulation Regions", which allows, among other things, local control of many simulation options along z. These include step size, reference k value and angle, Pade order, simulated bend options, and propagation scheme. This feature is discussed in the BeamPROP manual.
- Added an FFT-based BPM which can be used in conjunction with the above for certain special circumstances.
- Added nonlinear capability to 3d for Fiber/Channel structures.
- Added a new option which enables an improved vector BPM that handles a wider range of problems.
- Improved and extended the `WGMODE` command line program for analyzing simple waveguide structures.
- Improved robustness of the programs parameter estimation.
- Improved logic of the program if the user specifies a domain and grid which are not evenly divisible.
- Several limitations of the `PHASECOR` component of the `AWG` design module were removed.
- Corrected a problem in the AWG design capability which occurred when the user incorrectly selected TM in the global settings to attempt to account for polarization. As noted in the AWG documentation, the correct approach is to use the “design” variables described there.
- The command line length allowed in script processing has been increased from 256 to 1024.
- The "radial" variable has been added as an alternate way of specifying Radial BPM (for consistency with FullWAVE).
- Added a new option to disable convergence warnings during iterative mode calculation.
- Extended the Combine Mode Add option to allow the addition of standard index profiles for Fiber, Channel, and Diffused structures to any other structure (previously only user profiles could be added). This allows, for example, more convenient mixing of diffused and multilayer structures.
- Added a new capability to allow fiber and channel components to have their own “delta tensor”. This feature is accessed via the More... dialog of the component properties dialog. Coupled with Combine Mode Add and user profiles, by overlapping multiple waveguides this feature allows different index profiles for different components of the index tensor.
- Improved the calculation of diffused index profiles at substrate/cover interfaces.

- The `bdutil` BeamPROP data file manipulation utility has been updated to include a new option (`-k`) for calculating coupling coefficients. Type "`bdutil`" at the command line to see the new command line options. This feature is also accessed via the new grating analysis feature that is accessed via the Utility/Grating... menu item.
- Added a new utility, `grutil`, for performing simple grating analysis. Type "`grutil`" at the command line to see the usage for this utility. This feature is also accessed via the Utility/Grating... menu item.
- General speed improvements.

Analysis/Results:

- Added a capability to the WG Power monitor option in 3D to allow the user to choose whether the power should be integrated over a rectangular or elliptical cross-section.
- Improved power and overlap calculation for 3D vector field calculations.
- Added an option to output Radial BPM results as 3D data files and graphs.
- Added a new option to enable storing of both XZ and YZ data simultaneously when doing a 3D simulation and saving results.
- Added a fixed aspect ratio option to various plots. 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.
- The dispersion utility has been updated with new options to compute derivatives of Beta instead of Neff and with respect to wavenumber instead of wavelength, as will as an option to compute group velocity and dispersion as a function of wavelength. Type "`disperse`" at the command line to see the new command line options.

Other:

- Updated the main documentation and included a new consolidated and improved version of the AWG Utility documentation.
- Included a new version of WinPLOT (2.5) which, among other things, has a toolbar which allows convenient switching between the edit and plot windows.

Significant Changes in Program Behavior

- The method for calculating powers and overlap integrals for 3D vector field calculations has been improved. This may cause some change in results depending on the problem. Contact RSoft with any questionable change in behavior.
- The default behavior of the WG Power monitor option has been changed in the case of Fiber structures. Previously, the WG Power monitor would integrate the power over a rectangular cross section, independent of the waveguide geometry. Now, if the geometry of the waveguide in the

pathway being monitored is a Fiber, the program will integrate over an ellipse (or circle if the width and height are equal). This change in behavior will cause different results, so if you have files which use Fibers and WG Power, they may need to be changed if you prefer the old definition. The monitor dialog now has an option called Monitor Shape to control this.

- The method for calculating diffused index profiles at substrate/cover interfaces has been improved, however it should be noted that this may cause previous results to change if the grid size used was coarse (if the grid size used was sufficiently fine, the results should be essentially the same). While the new behavior should be more accurate, the old behavior can be recovered by setting `diffusion_intavg` to `0` in the symbol table.
- The old method for selecting Radial BPM, namely through the Numerical Scheme in the Advanced Parameters dialog, has been removed. Files which use this option will automatically be converted where appropriate to use the new "`radial`" variable introduced in 4.0f, which is also selected in the Global Settings.
- 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.
- The default color map for WinPLOT line drawings has been changed. The new map avoids the use of harder to see colors and addresses other issues. The old map can be recovered via the `/lcm` option in the `plot.cfg` file.
- Internal changes to script processing were made for general improvement. This may cause problems if DOS batch files or rarely used commands were called from BeamPROP scripts. If this is the case, contact RSoft to see if there is a workaround.

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 `README30.TXT`.

New Capabilities and Improvements to the Program

- The electrode/heater feature has been extended to recognize anisotropic dielectric constants or thermal conductivities.
- An option (`electrode_enable`) has been added to allow control the automatic electrode calculation feature.
- The electrode/heater feature has been extended to partially allow for tilted/curved structures.
- The wide-angle feature has been partially enabled for 3D.
- New iteration schemes have been added to the bidirectional calculation feature.

- The ContourMap(XZ) display option has been enabled for the bidirectional calculation feature, and displays the sum of the forward and backward fields.
- The precision of the scan output files has been increased.
- A new utility, `DISPERSE.EXE`, has been added to calculate dispersion from the results of an effective index versus wavelength scan.
- The behavior of the Run menu items and green light icon have been improved to retain default behavior that has been explicitly selected by internal options such as `idbpm=1`.
- The storage of data for 3D graphics has been turned on by default.
- A new diffused waveguide option has been added, `diffusion_gamma`, which provides additional control over the diffusion profile shape.
- A new option, Options/Import-Symbols, has been added to allow symbol table information from another file to be imported into the current circuit. One use of this feature is to allow construction of material databases that can be imported into a circuit. An example has been created for TiLiNbO3 in `TiLiNbO3.ind`, which is documented in `TiLiNbO3.doc` in the `examples` directory.
- The Launch Computed Mode option has been corrected to function with the scan feature.
- The scan feature has been extended to automatically produce neff versus the scan parameter when a mode calculation option is selected. The results are stored in `prefix.nes` (data) and `prefix.pne` (plot).
- The scan dialog now has an option to select the run type (e.g. propagation or mode calculation).
- An option, `electrode_a`, has been added to control the acceleration parameter in the electrode/heater feature calculation.
- The router design feature has been updated with a new interface. See `Router.doc` in the `router` directory for a description of the new procedure.
- The Run/Run-Script option has been added to allow running BeamPROP script files from inside the CAD program in addition to from the command line.
- Improvements to the nonlinear calculation option at high intensities, including a new `nl_iterations` option to control the iteration procedure in nonlinear calculations.
- An option for saturable nonlinearities has been added under the More... dialog of the component properties dialogs.
- An option for locking the aspect ratio has been added under View/Set-View-Parameters.
- A field in the symbol table editor has been added which shows the current value of an expression.
- The labels for the imaginary part of the index of refraction have been changed from "Loss Coefficient" to "Index (imag)" to clarify their meaning.
- Access to the More... dialog has been added to the Polygon and Lens elements.

- The maximum number of pathways and monitors has been increased from 32 to 200. The maximum number of components in a pathway has been reduced from 1000 to 200.
- The maximum number of characters in an expression has been increased from 64 to 80.
- Several new function calls have been added. See Appendix for more details.
- The constants "`pi`" and "`e`" have been added for use in expressions.
- A radial BPM option has been implemented, which is selected under the Numerical Scheme in the Advanced... section of the simulation dialog. This option can be used to simulate circularly symmetric structures, such as fibers, in 2D rather than 3D. In conjunction with the bidirectional option it can be used to simulate properties of some fiber bragg gratings.
- The Generalized Douglas (CN) scheme, which can sometimes lead to higher accuracy than the Crank-Nicholson (CN) scheme, has been implemented in the 2D, TE case.
- All of the LP fiber modes have been implemented in the launch and monitor dialogs.
- The multimode launch option now functions in 3D, and launches a random combination of LP fiber modes.
- The automatic far field option now applies to the field output in propagation mode as well as in mode calculations.
- The accuracy of far field calculations has been improved, and a new option, `farfield_da`, has been added to control the accuracy.
- Redraw of circuits with complicated formulas, such as router layouts, has been improved.
- Calculation speed has been improved for various specific structure types.
- The program now stores different step sizes for propagation (`step_size`) and iterative mode calculation (`step_size_idbpm`).
- A warning has been added when the iterative mode calculation does not converge.
- New icons have been added in the simulation part of the toolbar.
- Tooltips have been added to the icons in the toolbars.
- In Run Logs, the Ref k Value is now stored as `kbar_value`, while `kbar` is retained as originally set.
- A warning has been added to remind users to edit tapers and profiles via the Edit/Tables/... menu.
- For convenience the variable "`k0`" is now predefined in new circuits to be "`(2*pi)/free_space_wavelength`". This is mostly of use when defining a Ref k Value to correspond to an effective index as in "`k0*Neff`".
- A new option has been added under the Edit menu to allow selection/editing of a component by its number, which is useful for components that are "buried" under other overlapping components.
- An option has been added to allow 2D mode fields (coming from radial BPM for example) to be used as launch fields for 3D calculations.

- When an iterative mode calculation is performed, the imaginary part of the index is reported if present.
- The mode effective index is now stored in mode field files, and is used, for example, when a mode field is launched from a file at a tilted angle.
- The length of script files is now unlimited.
- The maximum number of symbols has been increased to 2048.
- The maximum number of electrode groups has been increased to 32.
- Deletion of components now maintains component ordering (component references were always maintained, but so-called depth-ordering would be modified).
- Options have been added under Edit/Change-Order to change the so-called depth-ordering of components.
- Several options have been added to provide an interface to FullWAVE, most notably the FDTD calculation in the startup dialog, and the corresponding simulation dialog that is invoked by the green light when in FDTD calculation mode. In addition, the Options/Insert/Time-Monitor menu item has been added.
- An option to produce registration marks for CAD export has been added under Options/Insert/Mark.
- The Channel structure now has an option (under the More... dialog) for obtaining sloped sidewalls.
- Various improvements and new features to the router design have been made. Refer to `Router.doc` in the `router` directory for more details.
- Several new utilities have been added under the Utility menu for creating arrays of waveguides in (XZ) or (XY) for use with photonic bandgap crystals and photonic bandgap fibers.
- A new ContourMap(YZ) display option has been added.
- The display of index profiles has been modified so that if an option other than Fixed at Z-Min is selected, the index instead of the height is plotted. A new option SurfaceRelief recovers the previous behavior.
- Several new color scales have been added.
- Several new options to WinPLOT have been added. These are documented in the `README.RPL` file.
- The default output option for storing slices has been changed to reflect more common usage. Now, storing of slices is disabled by default in the case of either mode calculations or when using the ContourMap(XY) display mode. Storing slices can be re-enabled via the output options dialog.
- The launch capabilities have been extended to allow multiple launch fields.

- The default vertical positioning of the monitor field has been changed in the case of a monitor type of File so as to attempt to automatically align with the pathway being monitored. Previously, only the horizontal position was automatically aligned.
- An option has been added (`launch_align_file=1`) to force alignment of the launch field with the pathway being launched into in the case of a launch type of File or (Computed Mode since it is based on File). Other launch types have always been and continue to be automatically aligned, independent of the setting of this new option.
- The launch and monitor polarization options (which only affect 2D slab modes) have been extended to include a Default option, which automatically adapts to the polarization setting in the global settings dialog. This changes the default behavior and may effect some existing `.ind` files if they were improperly set to begin with.
- The default options for bidirectional simulation in the TM case have been changed to use the undocumented `hfield` option, as is required for proper simulation in the bidirectional TM case.
- A new arc type has been added. The new Final Angle option complements the operation of the previously existing Initial Angle option.
- A new option controlling whether the lens object represents a cylindrical lens (the default) or a spherical lens, has been added to the lens properties dialog.

Significant Changes in Program Behavior

- The default mode calculation option has been changed from the correlation method to the iterative method. The correlation method can be selected explicitly via Run/Compute-Mode/Options and will be retained when the `.ind` file is saved.
- The Windows version of RPlot, `WINPLOT.EXE`, has been changed so that viewer mode is now the default and the `/v` option is unnecessary. To run the program in edit mode use the new `/e` option as in "`winplot /e`".
- The initialization file has been renamed from `BCADWIN.INI` to `BCADW32.INI`
- In the startup dialog the Save Settings box is now off by default.
- The option to show color scales has been turned on by default.
- The default coarse color scale has been changed to have 10 levels instead of 11, and more accurately correspond to the default continuous scale.
- The default waveguide display has been changed to Both from Center/Outer (see Options/Preferences dialog).
- The Run Log Output has been turned on by default.
- The default output option for storing slices has been changed to reflect more common usage. Now, storing of slices is disabled by default in the case of either mode calculations or when using the ContourMap(XY) display mode. Storing slices can be re-enabled via the output options dialog.

- The default vertical positioning of the monitor field has been changed in the case of a monitor type of File so as to attempt to automatically align with the pathway being monitored. Previously, only the horizontal position was automatically aligned.
- The launch and monitor polarization options (which only affect 2D slab modes) have been extended to include a Default option, which automatically adapts to the polarization setting in the global settings dialog. This changes the default behavior and may effect some existing `.ind` files if they were improperly set to begin with.
- The default options for bidirectional simulation in the TM case have been changed to use the undocumented `hfield` option, as is required for proper simulation in the bidirectional TM case.
- The behavior of user profiles has been changed. Previously, when a user profile was used which had higher priority than other waveguides, it would dominate the index profile, causing other waveguides to disappear. Now, only if the value of the user function is nonzero will it dominate. This new behavior matches what is most likely intended, however it may cause some existing `.ind` files to function differently. If this is the case, contact RSoft for a workaround.
- The interpretation of data files in the simulation program and various utilities has been modified so that outside the domain of definition, the function is defined to be zero, rather than an extension of the boundary data. In most cases, the data in a file would be near zero anyway and this will not matter, however we find this new behavior to be beneficial in many instances (for example if a mode is calculated on a domain which is not quite large enough, stored to a file, and then launched into the program). Contact RSoft if you notice any difficulties which appear to be related to this new behavior.
- The step size used for propagation and the step size used for iterative mode calculation have been fully separated. Previously, if the step size for iterative mode calculation had not been changed from the default, but the step size for propagation had, the propagation step size would be used for the iterative mode calculation. Now, if the step size for iterative mode calculation has not been changed from the default it will use the proper default, independent of the setting for the propagation step size.

Changes from Version 2.1 to Version 3.0

New Capabilities and Improvements to the Program

- A major new feature for bidirectional BPM, which accounts for reflections, has been added. A new example file, `GRATING.IND`, is included.
- A major new feature for nonlinear BPM has been added.
- A major new feature for calculating the effects of electrodes/heaters on the index profile has been added. A new example file, `ELECTROD.IND`, is included.
- A new launch option for automatically computing a mode and launching it into a circuit (Launch Computed Mode).

- A new launch option for controlling the normalization of the input field.
- A new monitor type for calculating the overlap with the input field.
- A new monitor type for calculating the mode effective index when doing an iterative mode calculation. This is useful when doing a parameter scan to get a plot of the effective index versus some parameter.
- Two new monitor types for calculating the 1/e width and height of the field.
- A new monitor option for controlling monitor normalization.
- A new monitor option for monitoring the power in a selected range of layers in a Multilayer structure.
- Improved expression handling with regard to operator precedence.
- Two new functions, `step(x)` and `step2(x,a)`, which are useful for defining grating structures or segmented waveguides.
- A new option for controlling the vertical profile for Diffused structures (`diffusion_shape`).
- When selecting the RPlot graph icon in BeamPROP, the Files of type field lists the available graph types.
- The simulation program now indicates the coordinates of the cursor in its titlebar, which is useful for determining monitor values or other coordinates.
- The Outline Color option in the display options dialog has been expanded to show the circuit outline in the ContourMap(XZ) display mode, as well as the cross-section outline in the ContourMap(XY) display mode.
- A new feature for displaying an "isometric" plot of the index profile.
- A new display mode, 3D Slices, has been added.
- An option for viewing horizontal and vertical cuts of contour plots.
- An option for enabling/disabling the saving of settings in the startup dialog.
- The scan variable feature has been extended to allow negative going scans.
- The scanning of two variables has been improved so that separate `.scn` files are generated for each value of the outer iteration variable.
- A new conversion utility, `mat2bp`, has been added, and the previous conversion utility, `bdconv`, has been expanded.
- A new utility, `bdutil`, is included which can perform power calculations, overlap integrals, field size measurements, and far field calculations.
- The iterative mode solver has been extended for higher order modes.
- A new mode options dialog has been added to allow control over several mode calculation options.

- An output option for automatically computing the far field after a mode calculation has been added.
- The Multilayer option has been enhanced to allow arbitrary tapering of individual layers.
- A new option for duplicating Multilayer layer tables and layer taper tables has been added.
- The priority level option has been extended to different structure types.
- The waveguide display color option has been extended to use the display color in the CAD program as well as the simulation program.
- A new option to allow different waveguide components to be assigned different mask layers when exporting to a standard CAD file.
- The maximum number of components allowed in the CAD tool has been increased to 20,000. Note that the simulation tool is still limited to 1024 components.
- The RPlot graphing tool, WinPLOT, has been upgraded to 32-bit, supports long filenames (note that spaces in file names are still not supported), and includes other miscellaneous improvements.
- Enhancements to the WDM router design package, including an interface from the main program, an option for generating the full CAD layout, the ability to control the taper shape of the input/output arms, and several smaller improvements. These features are documented in `ROUTER.DOC`.

Significant Changes in Program Behavior

- The DOS command line version of RPlot, `PLOT.EXE`, has been removed. Placing the command "`DOSKEY plot=winplot /v $*`" in the `AUTOEXEC.BAT` file will create an alias which functions similarly, but employs the more complete Windows version of RPlot.

Changes from Version 2.0 to Version 2.1

New Capabilities and Improvements to the Program

- Major new utilities and examples for designing Arrayed-Waveguide-Grating WDM Routers (see `ROUTER.DOC` in `C:\BEAMPROP\ROUTER` directory)
- Major new feature for anisotropic or birefringent media (see Global Settings)
- Ability to export the CAD layout in GDS-II as well as AutoCAD DXF format (see File/Export in the CAD program)
- Ability to launch and monitor the minor field component for full-vector simulations
- An optional fast-redraw mode in the simulation program and WinPlot (select View/Settings/Fast-Redraw or Plot/Settings/Fast-Redraw)
- Improved speed for many types of simulations

- Ability to set the display color of individual waveguides (see Segment Properties / More...)
- Options for local waveguide priority and effective background index (see Segment Properties / More...)
- Retaining Scan Variable information in `.ind` files
- Polygon components can specify a local rotation angle (see Polygon Properties)
- Extended flipping options and new rotation option (Edit menu)
- New convert-to-polygon option (Edit menu)
- Extended simulated bend feature allows different bend radii for different sections of waveguide (see Segment Properties / More...)
- Extended batch/scripting language
- Ability to allow individual waveguide components to act as perturbations (see Segment Properties / Combine Mode)
- Ability to apply different boundary conditions to different sides of the computational domain
- Control over the vertical position in the Fiber and Channel structures (see Segment Properties / More...)
- Extension of user-defined profiles to allow analytic functions of both x and y
- Control over the number of points and range in user-defined functions
- New functions `u(x)` and `clip(x)` available in expressions
- More control over display of the index profile
- Fields in the Launch and Simulation Parameters dialog now take formulas

Significant Changes in Program Behavior

- The rib-loaded structure has been removed, as it is a subset of the general multilayer option
- The DOS command line simulation program `BEAM.EXE` has been removed, as the extended batch/scripting capabilities of the graphical simulation program eliminate the need for this option.

Changes from Version 1.1 to Version 2.0

New Capabilities and Improvements to the Program

- Major new capability for incorporating polarization via vector beam propagation
- New feature for rapid mode calculation via imaginary distance beam propagation
- New features for dynamic reference wavenumber and dynamic boundary location
- Major new capability for user-defined index profiles and geometric tapers

- Several new pre-defined geometric tapers
- Major new capability for user-defined monitors
- Several new pre-defined monitors
- Enhanced formula capability
- Improved calculation with large index differences
- Major new feature for automatic parameter scanning
- Expanded toolbar access to common features
- Access to RPlot from the CAD program for convenient display of saved results

Significant Changes in Program Behavior

- The format for mode effective index files (`.mds` files) has been changed to provide more information.
- The default launch height and monitor height have been changed to reflect the local waveguide height. This may cause previous files to produce somewhat different results. The old behavior can be restored by appropriate choice of launch or monitor height in the respective dialogs.
- The default boundary gap and default monitor decay length have been changed. This may cause previous files to produce somewhat different results. Previously, the scale length was based on the fundamental mode. This was appropriate for single mode structures, but not for multimode devices. The default scale length is now based on the highest and next to highest modes, and is appropriate for both single and multimode devices.
- The behavior of the Full TBC (transparent boundary condition) has been changed to be more robust. This may cause previous files to produce somewhat different results. Also note that there are several undocumented options which control finer aspects of the TBC; contact RSoft for additional information.
- The index averaging procedure employed at waveguide boundaries has been changed. This may cause previous files to produce somewhat different results. Previously, the refractive index profile was averaged over a computational cell to produce more accurate results. Now the default is to average the square of the refractive index profile as this is more accurate for large index discontinuities. The previous behavior can be recovered by defining the symbol table variable `index_averaging` and setting its value to `1`.
- The interface to the user-defined index profile. feature introduced in Version 1.1 has been changed to allow more flexibility. Files using the old approach will automatically be converted to the new format. The old format will not be supported in future versions.
- The rib-loaded structure will be removed in the next release, since it is can be accomplished equally well via the multilayer option. If there are any questions regarding this, please contact RSoft.

Changes from Version 1.0 to Version 1.1

New Capabilities and Improvements to the Program

- Multiplatform support for DOS/Windows, Windows 95 and NT, OS/2, and UNIX/Motif
- Major new feature for computing modal properties (eigenvalues and eigenfunctions) of any 2D or 3D structure supported by BeamPROP
- Major new feature for defining multilayer structures with up to 256 layers, including the ability to have different layer tables for different waveguide components
- New capability for incorporating user-defined index profiles
- Addition of a user-defined polygon component type
- Addition of a lens component type
- Addition of cosine-based s-bend
- Enhanced formula capability allowing nested variable definitions
- Full support for complex refractive indices in all geometries
- Ability to graph the index or loss profile used in the calculation
- Ability to model effects of finite lithographic resolution
- Ability to model effects of waveguide edge roughness. More flexible multimode launch option
- Improved modeling of diffused waveguides (see notes below)
- Improved algorithm for generating AutoCAD DXF files

Significant Changes in Program Behavior

- The default for the Multimode launch option is now to launch all supported modes instead of only even modes. The previous launch option may be obtained using the new capability for selecting the mode set to be launched.
- The accumulation method for Gaussian profile types has been changed. The default is now to sum index or height changes for Gaussian profiles rather than take the local maximum as was previously done. This feature is controlled by the `gaussian_accum` variable, which defaults to `1`, and which when set to `0` recovers the previous behavior.
- The model for diffused waveguides has been altered significantly. These changes are based on field experiences, and lead to a more accurate model for these problems. Note that you may need to modify certain parameters in pre-existing index files to obtain results consistent with the previous model. The previous model can be restored via new options, however the default is to use the new model.
- The most drastic change from the previous model is that the normalization has been changed. The previous model contained the factor $1/\mathrm{erf}[w/(2*h_x)]$ in the normalization. With this factor, the

index difference that the user specifies is interpreted as the peak index difference at the top-center of the diffused region. While this normalization may be convenient, it leads to incorrect modeling of structures with varying width, if care is not taken to also modify the index difference. For example, in a taper structure, the above model would yield the same index difference at the top-center all along the taper. However, if the taper were fabricated by standard diffusion techniques, the peak index difference should vary along the taper. By removing the above factor in the normalization, the peak index difference is calculated correctly and automatically. Thus the new normalization does not contain the above factor. The index difference that the user specifies is now interpreted as the peak index difference that would be obtained for a one-dimensional diffusion (i.e. infinite width w). The variable `diffusion_norm` has been introduced to control this behavior, and a value of `0` indicates that the above factor should not be present (the new model), while a value of `1` will include the above factor (the old model). Note that for consistency with previous results, you should multiply any values for the index difference that you obtained previously by the above factor when using the new model. Also note that h_x in the above represents the diffusion length in the horizontal direction.

- The second significant change in the diffusion model relates to the method for calculating the net index change when multiple waveguides are present and closely spaced (as in a coupler). In the previous model, the index change at any point was obtained as the maximum index change due to any waveguide component. This approach works well for step-index structures and effectively deals with overlapping waveguides, however for diffused waveguides it is incorrect, as the local dopant concentration must be calculated as the sum of contributions from each waveguide due to the linearity of the diffusion equation. Thus in the new model, for diffused profile types the index changes from multiple waveguides are added, rather than the maximum being taken. The variable `diffusion_accum` has been introduced to control this behavior, and a value of 0 indicated that index changes should not be summed (the old model), while a value of 1 will accumulate index changes (the new model). Note that in order to deal with overlapping components, which would be handled incorrectly if their contributions were independently accumulated, the program now merges overlapping components internally so that they appear as one diffused region with a wider width. This new component has the properties of the leftmost component that has been merged. Also note that merging takes place only if the Global Settings for either Profile Type or 3D Structure Type are set to Diffused.
- The third change to the diffusion model is really an addition, namely the ability to specify different diffusion lengths in the horizontal and vertical directions. The fourth and final change to the modeling of diffused waveguides concerns the application of the effective index method. Previously, when reducing a 3D diffused structure down to a 2D one, only the peak effective index was calculated, and a pseudo-diffused index profile was used in the 2D calculation. Now, the full $n_{eff}(x)$ profile is used, and this can be significantly more accurate in many cases.

Index

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X

Y

Z

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