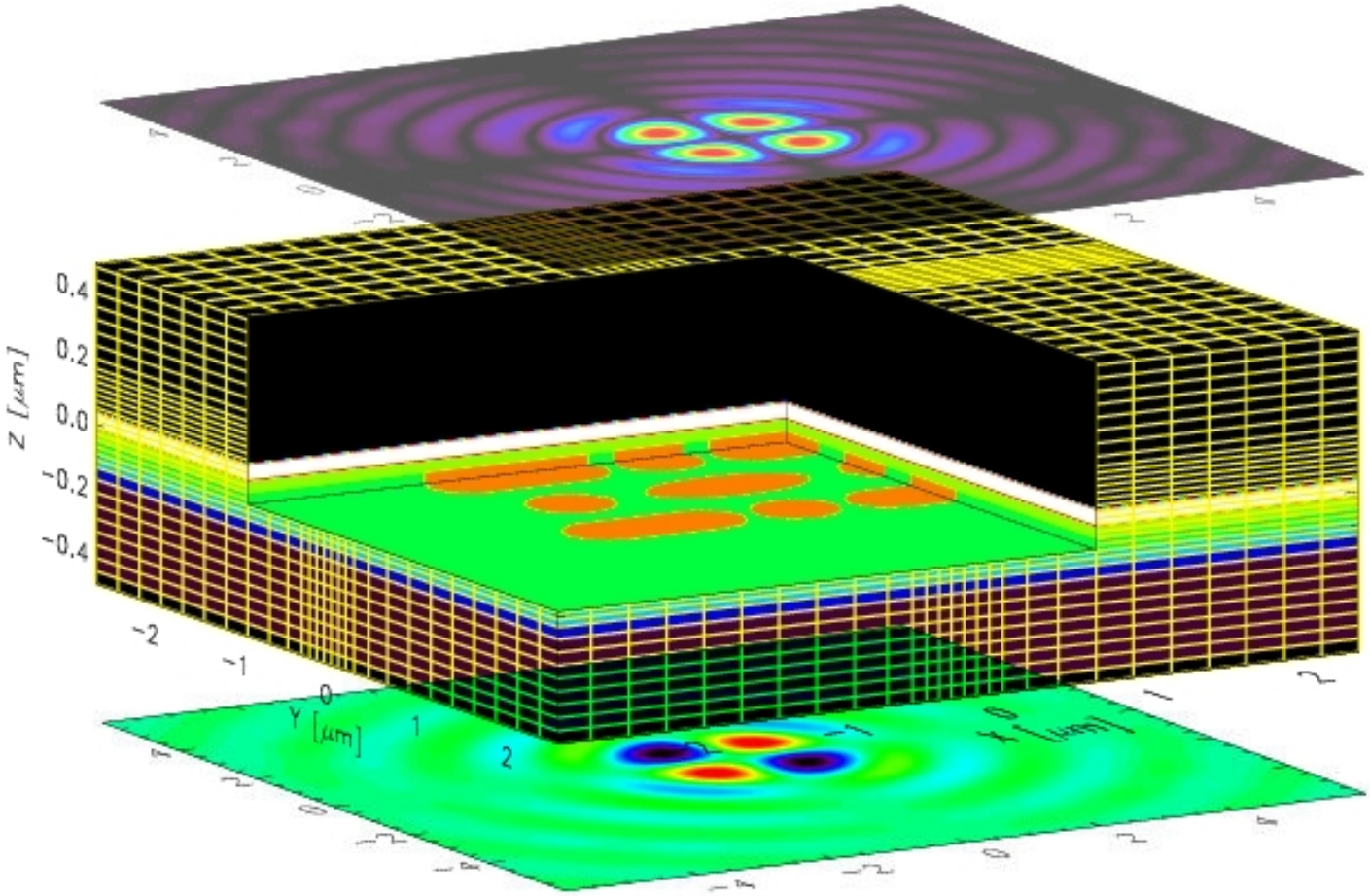
Sim3D_Max™



User's Manual

Version 2.17

Contents

Sim3D_Max^{TM} FDTD code User Manual

November 29, 2009

This document describes the user interface to a software module for solution of the time dependent vector Maxwell equations in three dimensions using the Finite Difference Time Domain (FDTD) method [1]. The main features of the program are:

- full vector field description in 3-D Cartesian geometry
- computation speed-up with Acceleware^{TM} hardware
- compatibility with DIFFRACT^{TM} software
- non-uniform grid support
- non-dispersive and dispersive material models
- UPML absorbing boundary conditions
- arbitrary geometry input
- parallel implementation for multi-processor platforms, based on the Message Passing Interface (MPI) [2] standard

Maxwell's equations, along with the constitutive relations, are discretized in space and time using FDTD method based on the second-order accurate, staggered central-difference operators. In MKS units the relevant equations can be written as follows:

$$\frac{\partial \mathbf{D}}{\partial t} = \nabla \times \mathbf{H} - \sigma \mathbf{E}, \qquad \nabla \cdot \mathbf{D} = 0,$$
$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}, \qquad \nabla \cdot \mathbf{B} = 0,$$
$$\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}, \quad \mathbf{B} = \mu_0 \mathbf{H}, \quad \mathbf{J}_p = \frac{\partial \mathbf{P}}{\partial t},$$

where $\mathbf{E}$ [Volts/m] is the electric field, $\mathbf{D}$ [Coulombs/m^2] - electric flux den-

sity, **H** [Amperes/m] - magnetic field, **B** [Webers/m^2] - magnetic flux density, $\mathbf{J}_p$ [Amperes/m^2] - electric polarization current density and **P** [Coulombs/m^2] is the electric polarization vector. Material properties are defined by electric conductivity σ [1/(Ohm m)], permittivity $\epsilon = \epsilon_0\epsilon_r$ and polarization **P**, where $\epsilon_0 = 8.854 \times 10^{-12}$ [Farads/m] and $\mu_0 = 4\pi \times 10^{-7}$ [Henrys/m] are free-space electric permittivity and magnetic permeability, and ϵ_r is the relative permittivity.

The version of the program described in this document allows computation of the scattered field for a given structure and a given incident field distribution created in DIFFRACT$^{\text{TM}}$ [3]. The input and output file formats are compatible with the file formats used by the FDTD interface option of the DIFFRACT$^{\text{TM}}$ software.

Sections 1 and 2 provide information on the system requirements and installation. Section 3 describes command line arguments of the program. In section 4 the structure of the input parameter file is given. Boundary conditions, material model and geometry input are described in sections 5-7. The last section provides some example problems along with the corresponding input files.

1 System requirements

Binary executables of the program are available for IA32 and EM64T/AMD64 platforms running Windows NT/2000/XP or XP x64 operating systems. Hardware acceleration is supported for Windows 2000/XP Pro/XP Pro x64 operating systems and requires a PCI Express x16 slot(s) for the Acceleware Accelerator$^{\text{TM}}$ card(s).

Table 1: Supported architectures and Windows$^{\text{TM}}$ operating systems.

ABI/OS	NT,2000	XP Pro	XP Pro with Acceleware$^{\text{TM}}$	Compute Cluster Server 2003 x64
×32	●	●	●	●
×64	●	●	●	●

An Intel Pentium$^{\text{TM}}$ 4, AMD Opteron (or equivalent) 1GHz or better processor and 1Gbyte or more of random access memory are recommended. The lower bound on the memory (in bytes) required by any given problem, can be estimated with the following formula: $M = N_x \times N_y \times N_z \times N_v \times N_f$,

where N_x, N_y, N_z are the number of points along the x, y, z axis and N_v is the number of variables stored at each point. The value of N_v is 6 (for E and H vector fields) if only dielectric material model is used and $N_v = 6 + 3p$ (E and H plus polarization vectors) if a dispersive Debye material model with p poles is used, or $N_v = 9 + 6p$ for the dispersive materials described by the Lorenz or Drude model with p poles (see section 6). The number of variables doubles when the Floquet boundary conditions are used. The factor N_f in the formula equals 4 bytes per float number for single precision and $N_f = 8$ for double precision. For example, an application that uses only a dielectric material model, boundary conditions other than Floquet and $200 \times 200 \times 100$ grid points, will require about 200Mbytes of memory.

Binary executables of the program for parallel computations take advantage of the many CPUs in the multiple-processor workstations, or a cluster of workstations connected by a high speed communication network. The execution time and memory requirements per node are reduced by distributing the computation across many nodes. For both serial and parallel execution of the program, the MPI libraries must be installed on the system. Support for the Microsoft™ MPI based High Performance Compute Cluster systems (Compute Cluster Server 2003 x64) as well as for the freely available MPICH2 based systems, is included. A freely available implementation of the MPI standard for Windows NT4/2000/XP Professional/XP Professional x64 or Server can be found on the installation disk or can be downloaded from the Argonne National Laboratory web site [2].

2 Installing the FDTD program to hard disk

To install the Finite Difference Time Domain (FDTD) program Sim3D_Max™ on hard disk, load the distribution CD-ROM. Double-click on Setup to start the installation (requires administrator priviledges). In addition to the setup of the Sim3D_Max™ package, the Setup utility will also invoke the installers for the following packages[1]:

- Message Passing Interface MPICH2™ system;
- Acceleware™ drivers (if applicable);

[1] Installation of the Message Passing Interface system is required for single- as well as multi-processor systems.

• SafeNet Sentinel™ USB license key drivers.

Choose "Reboot later" option during installation of each of the above components, and reboot only once, after the Setup finishes. Upon successful completion the installation creates a directory named
<NLCST\Sim3D_Max\Version.Number\>
under <Program Files> folder, and places a shortcut to the Sim3D_MaxGUI executable on the Desktop. Within the installed directory you will find the following folders: <GUI>, <Bin>, <Docs>, <Examples>.

<GUI> - supporting files for the Graphical User Interface of the program
<Bin> - Sim3D_Max™ executables and supporting files
<Docs> - manuals, publication reprints, and other documentation
<Examples> - folders with sample input files and test cases.

Follow instructions in `Readme.pdf` to complete the installation. For the computations that use the DIFFRACT™ source, the input beam, i.e. the complex light amplitude distribution that enters the FDTD mesh, must be generated by DIFFRACT™ (using the menu option FDTD in Export mode), then imported to the `Sim3D_Max.exe`. All output distributions computed by `Sim3D_Max.exe` will be stored on hard disk in files within the working directory specified in the parameter input file of `Sim3D_Max.exe`. Subsequently, these output files may be imported by DIFFRACT™ (using the menu option FDTD in Import mode) and displayed (using the option PLOT) or processed as any other beam cross-section is processed in DIFFRACT™.

3 Command line arguments

For serial computations the executable program, called `Sim3D_Max`, can be invoked with a single argument – the name of a text file containing simulation parameters, e.g.

```
Sim3D_Max parameters.input
```

If parameter filename is not specified, the program will prompt the user to enter it. To do many runs in sequence, batch files (in Windows) or shell script files (in Unix) can be used. For example, in Windows a text file called, e.g. "simulate.bat" containing lines

```
Sim3D_Max parameters1.input -b
Sim3D_Max parameters2.input -b
   ...
Sim3D_Max parametersN.input
```

can be executed to perform unattended `N` simulations with different parameters. The switch `-b` at the end of the command lines prevents the batch job from stopping at the end of each run or in case error conditions are encountered.

For parallel runs the program is invoked using MPI launcher and takes, in addition to the input parameter file name, an optional list of three integer numbers. These numbers specify the desired number of processing elements (CPUs or threads) per x, y, z dimension of the computational domain (see Appendix A Figure 35 for details). For example to start a run on six processors, and default distribution of processing elements, use

```
mpiexec.exe -np 6 Sim3D_Max.exe parameters.input
```

To assign one processor along the x-axis, three - along the y-axis and two - along the z-axis, use

```
mpiexec.exe -np 6 Sim3D_Max.exe parameters.input 1 3 2
```

The choice of processor grid, load balancing issues and parallel performance on different platforms are discussed in Appendix B.

The Acceleware^{TM} hardware-accelerated runs can be launched by specifying on the command line the option "-acceleware auto", e.g.:

```
Sim3D_Max.exe C:\somedir\parameters.input -b -acceleware auto
```

This executes computations using Acceleware^{TM} hardware that speeds up the computations, with automatic selection of the processing option for optimal performance.

4 Parameter File Description

The input parameter file ("parameters.input" in the above example) has a predefined structure. The order in which parameters appear in the file and the number of entries on each line should conform to the description given below. The number of white space characters before, after or between the

entries on each line and the number of newline characters between lines can be arbitrary. The *entry* here means a sequence of characters not separated by a white space, tab, newline or carriage-return character. For example, if the manual shows `x0,y0,z0`, then DO NOT use `x0, y0, z0` instead.

4.1 Simulation name

The first line should be a line consisting of three arbitrary words. It can be used to describe the file or simulation. Example:

```
FDTD Input Parameters
```

4.2 Time control

Time control specifies the start and finish times for the simulation in nanoseconds and sets the time step, e.g.:

```
Start-stop and timestep:
 tmin  [nanoseconds]  0.0
 tmax  [nanoseconds]  20.0e-6
 delta_t  automatic-with-CFL 0.4
```

The time step, `delta_t`, can be set in two ways. The first format,

`delta_t automatic-with-CFL CFL_NUMBER`

specifies that the time step should be computed from the spatial grid cell sizes, $\Delta x, \Delta y, \Delta z$ according to,

$$dt = CFL\_NUMBER \times min(\Delta x_i, \Delta y_j, \Delta z_k)/c,$$

where c is the speed of the light in a vacuum and CFL number should be a number less than the stability limit, $CFL\_NUMBER < 1/\sqrt{3}$.
The second format,

`delta_t [nanoseconds] 1.0e-8`

sets the value of the time step explicitly, in nanoseconds.

For problems in which time-harmonic (continuous-wave) solution is sought, the convergence of the solutions can be verified by increasing the value of t_{max} with all other problem parameters unchanged, and repeating the simulation. Similarly, convergence of the solution with respect to the value of

the time step can be verified by decreasing the time step Δt (or, equivalently, decreasing the CFL_NUMBER). See section 4.3 for accuracy and convergence dependence on the spatial grid cell size.

4.3 Spatial grid specification

A uniform grid can be specified by setting the number of cells and computational domain size as follows, e.g.:

```
Uniform Grid:
   nx      [cells]      120
   ny      [cells]      150
   nz      [cells]      200
   xmin    [micron]    -3.0
   xmax    [micron]     5.0
   ymin    [micron]    -6.0
   ymax    [micron]     4.0
   zmin    [micron]     0.0
   zmax    [micron]     15.0
```

The cell size along x axis will then be a constant $\Delta x = (x_{max} - x_{min})/n_x$, and similarly for y and z axis.

Alternatively, a non-uniform grid can be set up in one of the two ways. In the first approach, appropriate for grids symmetric in the XY-plane, the user specifies the sizes (in micron) of three regions, w_1, w_2, w_3 along the x (or y) axis, followed by the cell sizes in each of these regions $\Delta_1, \Delta_2, \Delta_3$, see Figure 1a. The x and y axis are treated identically and the generated grid is symmetric with respect to the center of the x and y axis.

The cell sizes are interpolated at the boundaries of the regions with different cell sizes to provide a grid with gradually changing cell size. Unlike the uniform grid case, for a non-uniform grid, the resulting computational domain size and number of cells are computed by the program, and the coordinate origin is positioned at the center of the computational domain. The x_{max}, x_{min} and y_{max}, y_{min} values are set by the program to $\pm L_x/2, \pm L_y/2$ respectively, where L_x, L_y are the total domain size computed from w_1, w_2, w_3. Similarly, three regions h_1, h_2, h_3 along the z axis are specified, followed by the cell sizes in each of these regions, $\Delta z_1, \Delta z_2, \Delta z_3$, Figure 1a. The z_{min}, z_{max}

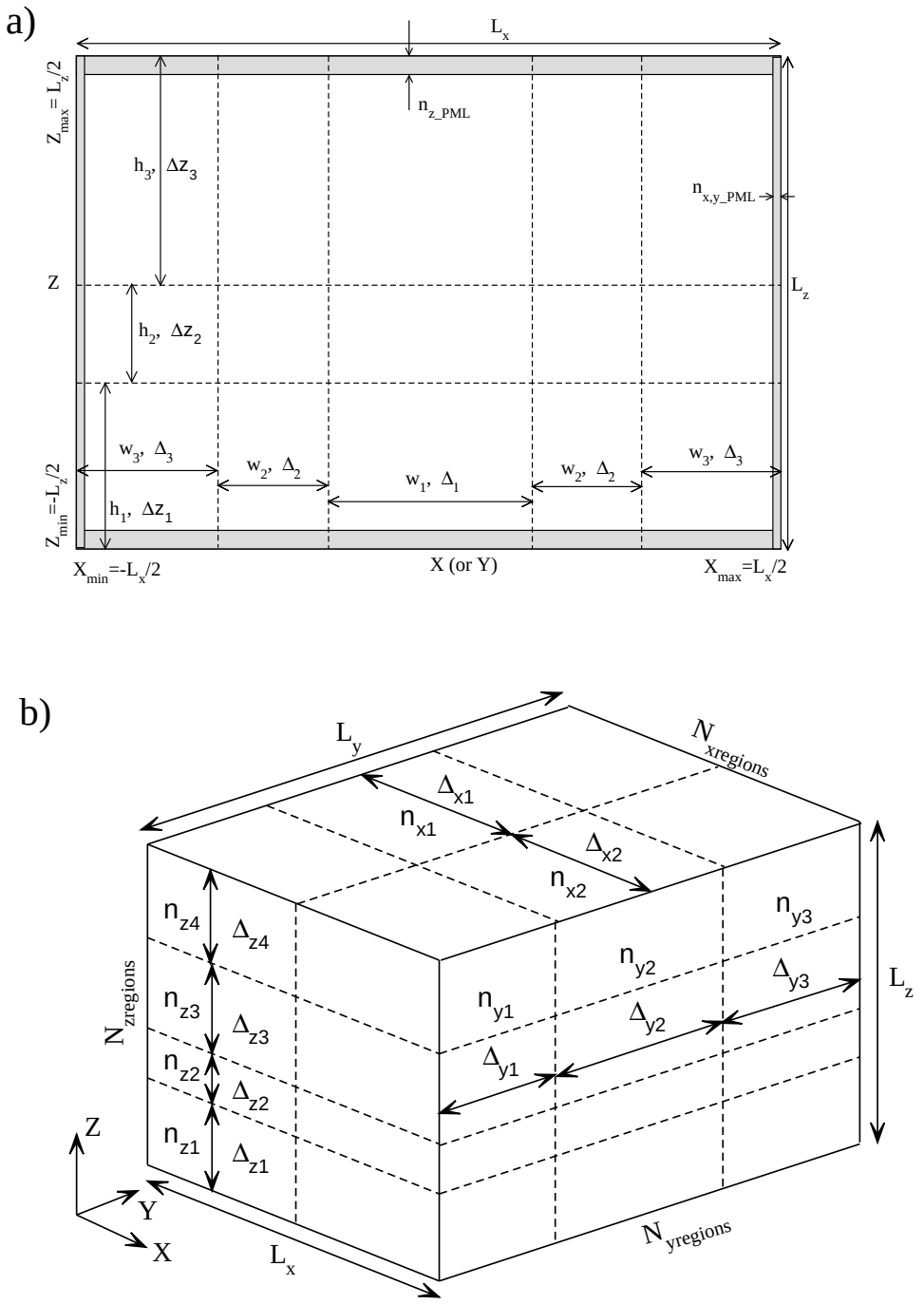


Figure 1: **a)** Computational domain and non-uniform grid definition for the `Non-Uniform Grid1` input. The grid is symmetric in the XY plane and consists of three regions with different cell sizes along either x, y or z-axis. **b)** Computational domain and non-uniform grid definition for the `Non-Uniform Grid2` input. The grid consists of arbitrary number of regions with different cell sizes along the x, y and z-axis. In the example shown $N_{xregions} = 2, N_{yregions} = 3, N_{zregions} = 4$. The total number of cells along any axis is equal to the sum of the specified number of cells in each region of that axis. When absorbing BCs are set, PML regions are always counted as part of the total length of the domain.

are computed and set by the program in the same way as x_{min}, x_{max}. An example of non-uniform grid specification is given below:

```
Non-Uniform Grid1:
   w1        [micron]  500e-3
   w2        [micron]  200e-3
   w3        [micron]  2040e-3
   delta_1   [micron]  10e-3
   delta_2   [micron]  20e-3
   delta_3   [micron]  30e-3

   h1        [micron]  160e-3
   h2        [micron]  140e-3
   h3        [micron]  300e-3
   deltaz_1  [micron]  10e-3
   deltaz_2  [micron]  5.0e-3
   deltaz_3  [micron]  10e-3
```

Note, that due to the smoothing of the cell size done by the program at the interfaces between regions with different resolution, the resulting domain will have a total length larger (by a few cells) than the sum of lengths of individual grid regions. A second way of setting up a non-uniform grid allows direct specification of an arbitrary number of regions with a certain number of cells of a given size, separately for each axis.

```
Non-Uniform Grid2:
   N_xregions [integer]   2
   deltax_1   [micron]    10e-3
       nx_1   [cells]     50
   deltax_2   [micron]    10e-3
       nx_2   [cells]     50

   N_yregions [integer]   5
   deltay_1   [micron]    10e-3
       ny_1   [cells]     185
```

```
deltay_2    [micron]    5e-3
    ny_2    [cells]     10
deltay_3    [micron]    2e-3
    ny_3    [cells]     5
deltay_4    [micron]    1e-3
    ny_4    [cells]     150
deltay_5    [micron]    2e-3
    ny_5    [cells]     5

N_zregions  [integer]   1
deltaz_1    [micron]    10e-3
    nz_1    [cells]     400
```

In the above example the grid consists of 2 regions along the x-axis, of 5 regions along the y-axis, and along the z-axis the grid has just 1 region (uniform). For any of the axes, the total number of cells is a sum of the specified number of cells for each region. The x_{min}, x_{max}, are computed as $x_{min} = -L_x/2$, $x_{max} = L_x/2$, where $L_x = \sum_{i=1}^{N_{xregions}} n_{xi}\Delta_{xi}$ (Figure 1). Similarly for y_{min}, y_{max},z_{min}, z_{max}. Unlike the non-uniform grid option `Non-Uniform Grid1`, no grid smoothing is used in the case of the grid set with `Non-Uniform Grid2` entry.

Some general considerations for setting up the grid:

When the PML boundary condition is set for any of the axes, the computational domain size input by the user will include a Perfectly Matched Layer (PML) region at the boundaries of that axis. For example, if there are n_x cells set for an x-axis, with total length L_x, and nx_{PML} cells set for the PML region at each end of the domain, then there will be $nx - 2nx_{PML}$ non-PML points (over the length $L_x - nx_{PML}(dx_{top} + dx_{bot})$) inside of the domain, where dx_{top}, dx_{bot} is the cell size in the PML regions. Inside the PML layers the nonuniform grid **should not have** any variation of the cell size in the direction normal to the PML layer. To change the boundary conditions or modify the default number of PML points, see section 5.

Large cell-size ratio $r = \Delta_1/\Delta_2$ in neighboring regions of the non-uniform grid along any axis can have a negative impact on the accuracy of the solu-

tion. Gradual change of the cell size must be used when a ratio of cell size $r > 2$ is used for grid refinement. Additionally, large cell-size aspect ratios Δ_u/Δ_v, $u, v = x, y, z$ must be avoided to maintain accuracy.

The grid cell-size along any axis and within a region of material with refractive index n_{ref}, can be estimated as $\Delta \leq \lambda/(N_{ppw} n_{ref})$, where λ is the free space wavelength of interest, and N_{ppw} is the number of cells per wavelength in the medium. Typically, for errors in the solution to be less than a few percent, $N_{ppw} > 20-30$ must be used. The computation time step is proportional to the smallest cell size found in the computational domain. Convergence of the solutions can be verified by decreasing the cell sizes, typically by a factor of 2, with all other problem parameters unchanged, and repeating the simulation.

4.4 Computations in 2D

The default computation mode is 3D. In order to switch to two-dimensional computations, either uniform grid or second approach to the non-uniform grid specification must be used to set the number of points in the x-direction to `nx=1` and the boundary condition for the `xaxis` must be defined as `[periodic]` in the boundary conditions file, described in section 5. One of the 2D sources must be specified for (Hx_Ey_Ez) or (Ex_Hy_Hz) mode computation (sources are described in subsection 4.13).

With `nx` set to 1, the 2D computational domain is the YZ-plane and the 2D modes are the transverse electric $TE_x \rightarrow (H_x, E_y, E_z)$, and transverse magnetic $TM_x \rightarrow (E_x, H_y, H_z)$ modes, with field components depending on time and y, z coordinates: $\mathbf{E} = \mathbf{E}(t, y, z)$ and $\mathbf{H} = \mathbf{H}(t, y, z)$. Hence, this manual uses the convention, in which TE_x (or TM_x) signifies the mode with electric (or magnetic) field transverse to the x-axis.

For 2D computations the domain in the x-direction is just one cell long and can have arbitrary coordinates x_{min}, x_{max} of the computational domain. A convenient choice of x_{min}, x_{max} usually is $x_{min} = 0$ and x_{max} equal to the largest cell size along the y or z axis. All source, monitor, geometry, etc. objects, that require input of an x-coordinate must use values within the $x_{min} < x < x_{max}$ range, if positioning in the YZ-plane of the computational domain is desired.

4.5 Working Directory

The working directory is a full pathname specifying a directory where the program will look for all input (geometry, material, boundary conditions, source) files, and where it will write the output files, e.g.,

```
   Working directory:     /home/username/Maxwell/FDTD/
or
   Working directory:     C:\username\Maxwell\FDTD\
```

For all input files the program looks for the file (e.g. file `material.input` described in the next subsection) in the directory specified under `Working directory:`. However, if the filename includes explicitly the path,

`C:\User\myfiles\material.input` or `.\material.input`

then the filename is used as specified. Directory and file names containing the space character must be enclosed by double-quote characters: "C:\User Name\My Data Files\".

4.6 Material Definition File

The "Material Definition File" sets a filename of a file containing material model specifications (see section 6), e.g.

```
   Material Definition Filename:    material.input
```

4.7 Geometry Definition File

The "Geometry Definition File" sets the filename of a file containing geometry specifications (see section 7), e.g.

```
   Geometry Definition Filename:    geometry.input
```

Same rules regarding filename apply as specified above.

4.8 Boundary Conditions File

Boundary conditions can be set using the file specified under the `Boundary Conditions Filename:` entry (see section 5). For example:

```
   Boundary Conditions Filename:    boundaries.input
```

4.9 Material index output

The material index output provides an option to write to a file a 3D spatial distribution of the material logical index set up from the material and geometry files. Example:

```
Material index:
   Write to file?        no
   Filename              mindex.out
```

`Write to file?` must be followed by `yes` or `no`. The resulting file is in ASCII format. It contains the total number of cells n_x, n_y, n_z, followed by the logical enumeration value of the material for each cell i, j, k of the computational grid. This logical value is simply a number corresponding to the order in which materials are defined in the material definition file. The order of the cell output is one, in which k changes first, then j, then i, as illustrated in the following pseudo-code segment:

```
for i=1 to nx
  for j=1 to ny
     for k=1 to nz
       write/read material enumeration value in cell i,j,k
```

The same file structure is used in the geometry object defined in 7.1. The coordinates of the cells are written into ASCII files "coordx", "coordy","coordz" in microns, i.e. the cell i, j, k has cell-center coordinates $x[i], y[j], z[k]$ found on lines i, j, k in files "coordx", "coordy" and "coordz" respectively. These files can be used to plot and verify the material and geometry setup specified in the input files. Sample plots are shown in Figures 2 and 3.

4.10 Field output

Spatial distribution of the **E** and **H** fields can be optionally written into files at specified equal time intervals:

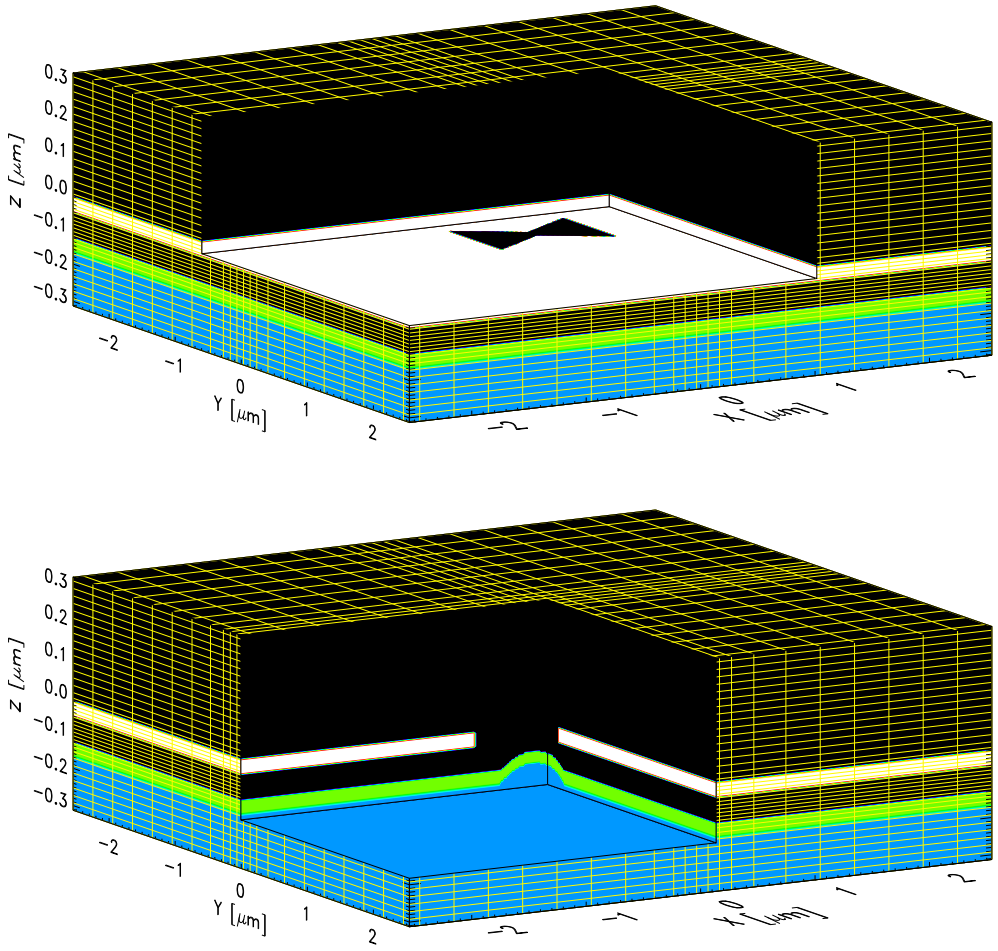


Figure 2: Sample geometry set-up for a near field antenna over a bump. The transmitter consists of a bow-tie aperture in a metal layer (cross-section) and a sphero-cylindrical bump in another layer (bottom cross-section view).

```
Fields:
  NumberOfOutputs    2
  WriteEx,Ey,Ez?     no yes yes
  WriteHx,Hy,Hz?     yes no no
```

These lines can be followed by an optional line requesting output of the Poynting vector field as well, e.g: `WriteSx,Sy,Sz?  no yes no`. Field output can be disabled by setting `Number of outputs` to 0. If `Number of outputs` is $N > 0$, then fields are written into files N times, at $t = t_{max}/N$, $2t_{max}/N$, ..., t_{max}.

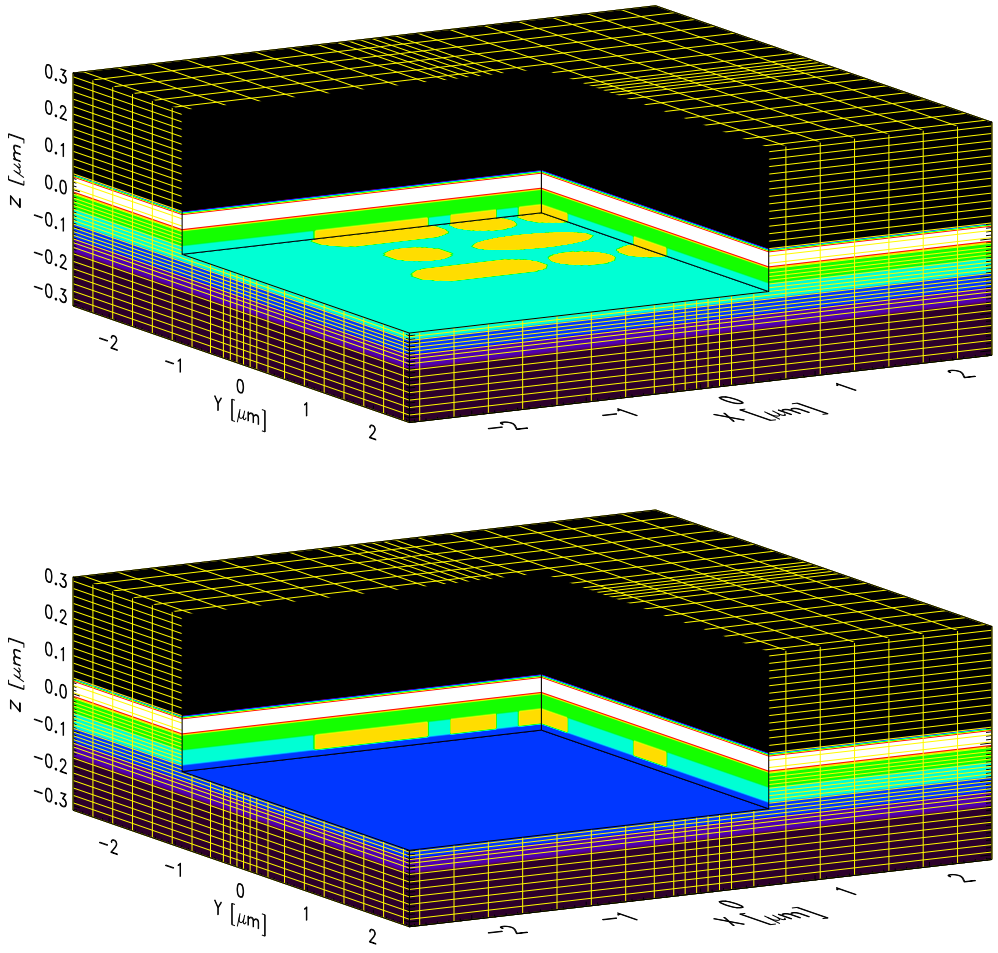


Figure 3: Sample geometry set-up of a multi-layer stack with an array of elliptic and capped-rectangle marks similar to those used in the optical data storage media.

Arbitrary times for output can be set as follows:

```
NumberOfOutputs 4
OutputTimes [user_defined]
Time_1          [nanoseconds] 0.8e-6
Time_2          [nanoseconds] 1.07e-6
Time_3          [nanoseconds] 1.35e-6
Time_4          [nanoseconds] 1.8e-6

WriteEx,Ey,Ez?    yes yes yes
WriteHx,Hy,Hz?    no no no
```

The number of `Time_i` lines must be equal to `NumberOfOutputs`. When

requested, the **E** fields are written into files "Ex.bin.out","Ey.bin.out" and "Ez.bin.out". The **H** fields are written into files "Hx.bin.out", "Hy.bin.out", "Hz.bin.out". During the computation each time-snapshot of the spatial dis-

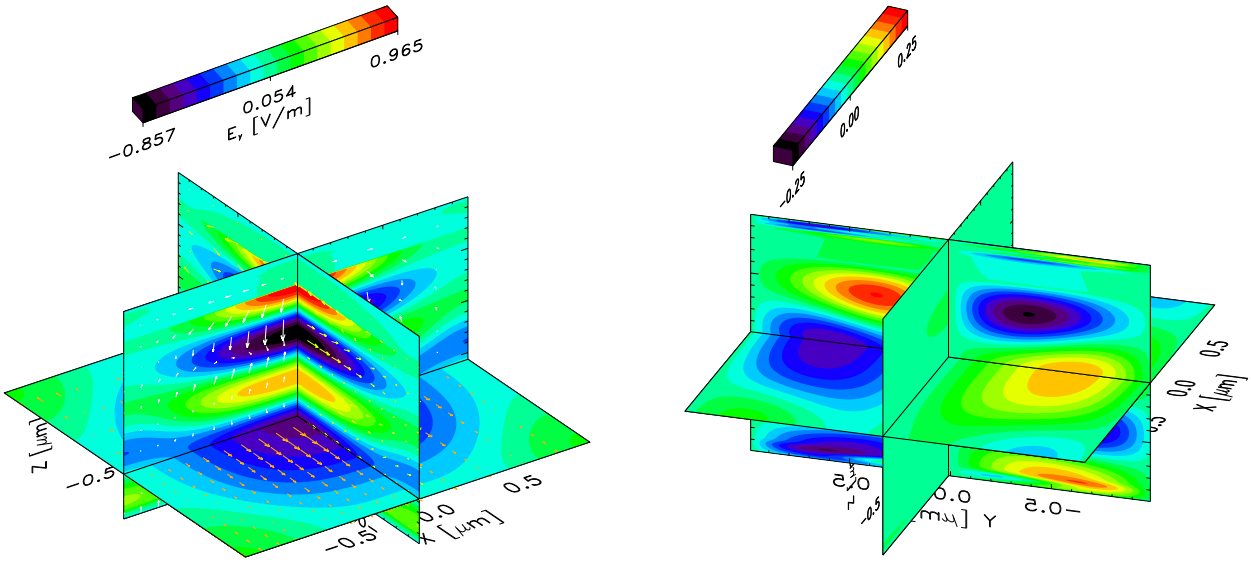


Figure 4: E_y (left) and E_z (right) components of a y-polarized 3D Gaussian beam sourced in the xy-plane with periodic boundary conditions, at $z_{min} = z_{max} = 0.7\mu m$ with z-parameter set to 0, beam amplitude FWHM= $1.2\mu m$, $\lambda_0 = 0.8\mu m$, $n_{ref} = 1.0$, and direction set to -1.

tribution of the fields is appended to the end of the corresponding file. When Poynting vector output is requested, the last output to the "Sx.bin.out", "Sy.bin.out" and "Sz.bin.out" files is the time average of **S** over one period of the source frequency, instead of a time-snapshot. The files can be read and visualized (Figures 4-5) after each output, while the computation is in progress. When new simulation is started, the output files are **overwritten**.

The fields are written into files as binary (unformatted) data. The order of output of the fields is one in which index k along the z-axis changes first, then index j along the y-axis, and last - index i along the x-axis. Each point (k,j,i) is written as a 4 byte floating point number. Hence, if N outputs are requested, and the number of points in the computational domain is n_x, n_y, n_z, the size of each output file will be $4N \times n_x \times n_y \times n_z$ bytes.

If any of the fields are requested to be written, the material layout binary file "Ml.bin.out" and ASCII text files "coords" and "coordx", "coordy", "coordz" are also created. The material layout file "Ml.bin.out" contains the refractive index n_{ref} distribution in the computational domain. For the ma-

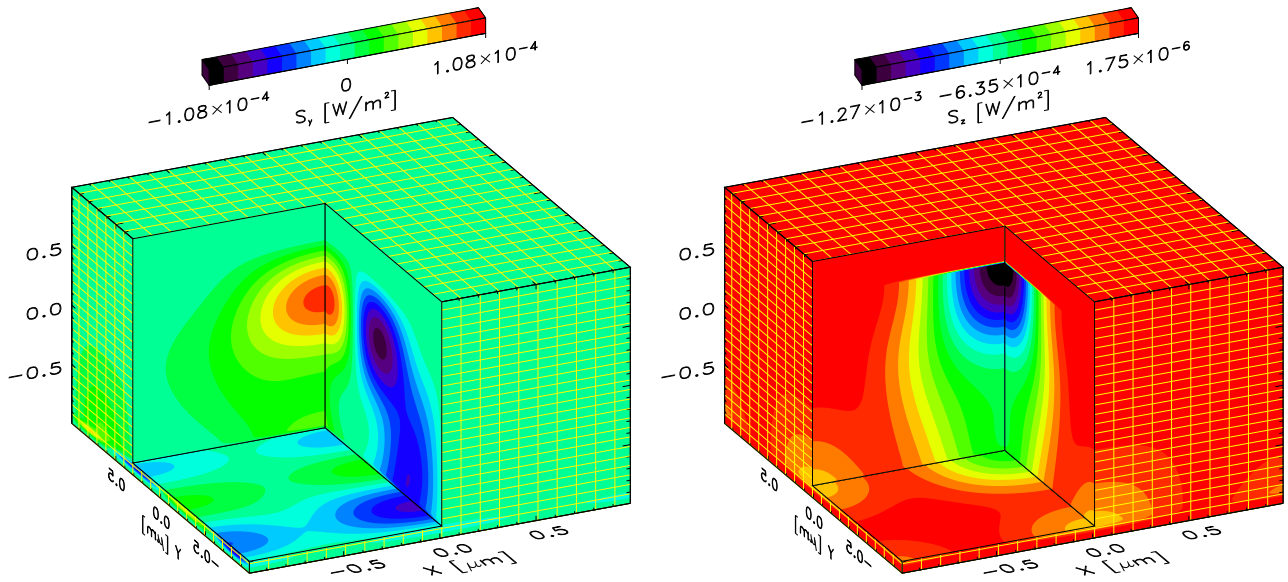


Figure 5: S_y,S_z components of a 3D Gaussian beam from Fig. 4.

terials based on Lorentz and Debye material models, instead of n_{ref} a value $-m$ is written into "Ml.bin.out" file, where m is the order number in which the material is defined in the material definition file. For the materials of type `Debye(x,y)` the conductivity (in MKS units) is used in the "Ml.bin.out" file for each point of the domain occupied by the `Debye(x,y)` materials. The binary data format of the material layout file is the same as that of the field files.

The file "coords" contains on the first line the number of points, n_x, n_y, n_z, followed on the second line by the number N of outputs done, and the computation output times, in seconds, e.g. $t = t_{max}/N$, $2 * t_{max}/N$, ..., t_{max}, on the third and following lines. The files "coordx", "coordy", "coordz" contain a single column corresponding to the coordinates of each cell in the x,y,z direction.

4.11 Checkpoint files

The `CheckpointFile:` block sets the file output and input options for saving and restarting the computation. `RestartFromCheckpointFile` option allows the computation to be continued from a previously saved checkpoint file. `WriteCheckpointFile` requests that a checkpoint file be created at the

end of simulation.

```
CheckpointFile:
  RestartFromCheckpointFile    no
  WriteCheckpointFile          yes
```

Each processor creates one checkpoint file in the working directory under filename "chkpLxMxN", where L,M,N are integers identifying the processor. If the checkpoint file already exists, it is **overwritten**.

When simulation is restarted from the checkpoint file, the program assumes that the material and geometry definition files are the same as those used in the run that created the checkpoint file. Also, start time *tmin* is reset to the simulation time read from the checkpoint file, and *tmax* is set to the new *tmin* plus the difference between *tmax* and *tmin* appearing in the input parameters file. All other time dependent entries in the parameter file are used as specified.

When monitors or ExportReflected/Transmitted planes are used with the checkpoint option, the Fourier transforms and monitored fields are not saved to the checkpoint files, and hence they do not persist from one checkpoint to another. Instead, they are computed anew in every run. Therefore, for example, the data in the Export plane files from any run is valid only if the simulation time $tmax - tmin$ of that particular run is large enough to sample the minimum required number of periods of the source, otherwise a warning message is issued by the program.

4.12 Source time profile

The `TimeProfile:` block is optional in the input file. When present, it specifies the modulation $f(t)$ of the time dependent source $f(t)\sin(\omega t + \phi_{ceo})$:

```
TimeProfile:
  f(t) [superGaussian]
  t0   [nanoseconds]   400.0e-6
  FWHM [nanoseconds]   80.0e-6
  n                    2.0
```

Valid options for the `f(t)` entry are `[sech]`, `[superGaussian]`, `[tanh]` or `[file]`. The functional dependence and parameter values for these op-

tions are given in the Table 2. The default time profile is a `tanh` function, with `t0` set to 3 fs, and `tau` - to 1 fs. In the case of a user defined time profile,

```
TimeProfile:
  f(t)       [file]
  Filename   [string]  user_time_profile.input
```

the specified ASCII file is expected to contain on the first line the number of points, N_t, present in the file, followed by a single column of the values of the function $f(t)$ sampled at each computation time step $t = i \cdot dt$, $i = 1, 2, ...$ During the simulation, for $t > N_t \cdot dt$, the $f(t)$ is set to the value on the last line of the input file.

Table 2: Source time profile parameters

f(t)	Function	Parameter 1	Parameter 2	Parameter 3
`[sech]`	$\frac{2}{exp[-(t-t_0)/\tau]+exp[(t-t_0)/\tau]}$	t_0	`FWHM` $= 2\tau \ln(2+\sqrt{3})$	—
`[superGaussian]`	$exp[-\{(t-t_0)/\tau\}^n]$	t_0	`FWHM` $= 2\tau(\ln 2)^{1/n}$	n
`[tanh]`	$\frac{1}{2}(1+tanh[(t-t_0)/\tau])$	t_0	`tau`=τ	—
`[file]`	user defined	filename	—	—

For any choice of the profile, the amplitude of the time profile function can be set by specifying an optional entry in the following form:

```
  amplitude  [V/m]  2.0   -or-  amplitude  [A/m]  2.0
```

If this line is omitted, a default value of 1.0 is assumed for the amplitude.

The default value of the phase-shift between the envelope function $f(t)$ and the carrier wave $\sin(\omega t + \phi_{ceo})$ is zero, $\phi_{ceo} = 0$. The carrier-envelope offset ϕ_{ceo} can be changed with an optional entry:

```
carrier-envelope offset  [degrees] -90.0
```

4.13 Sources

One of the following sources must be specified in the input parameters file:

- source from the input DIFFRACT^{TM} file
- point source

- planewave source
- Gaussian beam source
- source from the input file in 2D
- planar waveguide source in 2D

Sub-sections 4.14-4.19 describe input required for each of these sources.

4.14 DIFFRACT^{TM} source specification

To use a DIFFRACT^{TM} source the user must set the wavelength (in microns), the filename (see 4.5 for input filename rules) of a complex-valued source amplitude distribution for **E** and **H** fields, the format of the file and the grid type. For a beam propagating in DIFFRACT^{TM} along the positive direction of the z-axis, **H**-fields must be sampled on a plane positioned a distance $\Delta_z/2$ (half of a FDTD grid cell) behind the plane where **E**-fields are sampled, Figure 6. Therefore, negative value of $-\Delta_z$ (not $-\Delta_z/2$) must be specified when the source file is produced by the FDTD export option of DIFFRACT^{TM}. The content and format of the file are described in detail in section 4.20 and in the manual for the DIFFRACT^{TM} software [3]. Either `Source:` or `DiffractSource:` may be used to specify a source distribution created with DIFFRACT^{TM}. The `File Format` can be one of `ascii`, `fortran_binary`, `c_binary`. The choice `ascii` is used for files in a text format as produced by FORTRAN for type COMPLEX.

```
DiffractSource:
   Wavelength [micron]   0.65
   Filename              diffract_output.dat
   File Format           ascii
   Grid Type             staggered
   z-location [micron]   0.1
```

The `fortran_binary` format corresponds to unformatted output produced by FORTRAN and is the same as the binary format used in DIFFRACT^{TM}. The option `c_binary` is for the binary output produced for/by C programs. The later is different from FORTRAN in that it does not have data tags, and elements of the arrays are written in the row-major order. Binary format allows smaller file size and faster data input or output, than does ASCII.

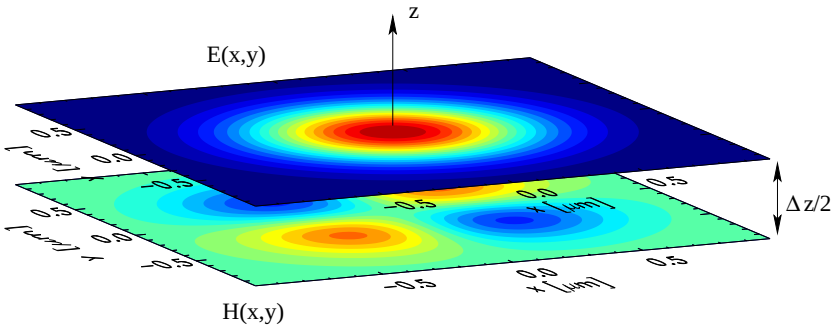


Figure 6: In Diffract source file **E**-fields must be sampled on an xy-plane at some position z, with **H**-fields sampled on an xy-plane at position $z - \Delta z/2$.

The `Grid Type` can be specified as `staggered` or `collocated`, to indicate the logical location on the computation mesh of the **E** and **H** fields. The `staggered` corresponds to a staggered positioning of **E** and **H** fields, as defined in the FDTD method. The `collocated` option corresponds to all field components defined at the cell center. The grid type should be set to be the same as the grid type used when creating the source file with FDTD Export option in DIFFRACT^{TM}.

The last line is optional and specifies position, along the z-axis, of the XY-plane in which the source is excited. When this line is not present, the source plane position defaults to the top of the computational domain, just before the PML layer, at $z = z_{max} - (n_{z\,PML} + 3)\Delta z_{top}$, where Δz_{top} is the grid spacing used in the PML region at the top of the computational domain.

The sourced field distribution imported from DIFFRACT^{TM}, creates a beam propagating in the FDTD grid along the negative direction of the z-axis. Due to the numerical approximation of the source distribution, small amplitude (about -30 db) residual waves propagating in the opposite direction will be generated at the source plane. The magnitude of these waves can be evaluated by launching the beam into a uniform medium, and monitoring fields behind the source plane. The magnitude of the residual waves can be reduced by refining the computation grid.

4.15 Point source specification

One or more point sources can be specified by their position `x0,y0,z0`, single field component to be sourced, and the switch on/off times.

```
PointSource:
  Wavelength  [micron]      0.65
  x0,y0,z0    [micron]      0.0 0.0 100e-3
  E-field     [x]
  t_on,t_off  [nanoseconds] 1.0e-6 6.0e-6
  phase       [degrees]     90
```

The phase parameter is optional. It specifies the constant phase shift ϕ_0 to be added to the time harmonic dependence of the point source, $\omega t + \phi_0$. When not specified, the phase shift defaults to zero.

The valid options for the field component entry are `E-field` followed by one of the `[x]`, `[y]` or `[z]` for computations in 3D. For 2D Hx_Ey_Ez mode, `E-field` components `[y]` or `[z]`, or `H-field` component `[x]` can be specified. For 2D Ex_Hy_Hz mode, `H-field` components `[y]` or `[z]`, or `E-field` component `[x]` can be specified. At each time step of the computation the value of the specified field component at the source location (single grid point $\mathbf{r}_{ps} = (x_0, y_0, z_0)$) is set to a time-harmonic $\sin(\omega t)$ dependence with specified `TimeProfile`. This results in a non-transparent ("hard") point source.

Instead of the electric field component, a polarization vector component can be specified. For example `P-field [x]` can be set for 3D or 2D Ex_Hy_Hz computation mode. In this case the point source is transparent, and is implemented by adding (at a single grid point) a time derivative of the polarization vector $\mathbf{P}_{ps}$ of the point source to the time derivative of the rest of the displacement field:

$$\frac{\partial \mathbf{D}}{\partial t} = \nabla \times \mathbf{H} - \frac{\partial \mathbf{P}_{ps}}{\partial t} \delta(\mathbf{r}_{ps})$$

On the computational grid the point source is representative of a volume of the single grid cell, and hence the amplitude of the point source will depend on the grid resolution.

In the case of multiple point sources, `PointSource` entries in the parameter file must be specified one after another.

4.16 Planewave source specification

A planewave source is set by the following input:

```
PlaneWaveSource:
  Wavelength   [micron]      0.65
  Mode         [3D]
  theta        [degrees]     30.0
  phi          [degrees]     45.0
  polarization [degrees]     90.0
  TFSF-boundary [top]
```

The `Mode` can be either `[3D]`, or one of the `[Ex_Hy_Hz]` or `[Hx_Ey_Ez]` for computations in 2D.

The propagation direction of the planewave is along the wavevector $\mathbf{k} = (k_x, k_y, k_z)$, specified by the angles θ and ϕ, Figure 7. The angle `theta` ($\theta \in [0°, 180°]$) is defined with respect to the negative k_z component, for the planewave propagating along the negative z-direction. The angle $\phi \in [0°, 360°]$ is measured with respect to the positive direction of the x-axis. Specification of the angle ϕ is optional, when omitted its value defaults to $\pi/2$.

Two orthogonal polarizations of the planewave can be specified by setting the polarization angle to 0° (**E**-field in the plane defined by the **k** and k_z) or 90° (**E**-field normal to the plane defined by the **k** and k_z vectors). Specification of the polarization angle is optional, when omitted its value defaults to 0°.

In 2D computations only the angle $\theta \in [-90°, 90°]$ has an effect, since the wavevector is in the YZ-plane, $\mathbf{k} = (0, k_y, k_z)$, and hence $\phi = \pi/2$, while the polarization angle is determined by the `Mode` parameter: `[Hx_Ey_Ez]` mode corresponds to 0° polarization angle, and `[Ex_Hy_Hz]` mode – to the 90°.

The planewave source can be used together with the PML, periodic or Floquet boundary conditions, also known as Bloch periodic boundary conditions. The PML and periodic boundary conditions can be combined for different axis, as described below. The Floquet boundary condition is appropriate for simulating periodic structures, and always implies PML boundaries along the z-axis and Floquet boundary conditions along the x- and y-axis.

The planewave source is implemented using the Total Field/Scattered Field (TFSF) formulation, with the TFSF boundaries defined two points away from the PML absorbing boundary. The TFSF boundary consists of 6 planes (4 line segments in 2D) that make up a surface of a cube in 3D (a

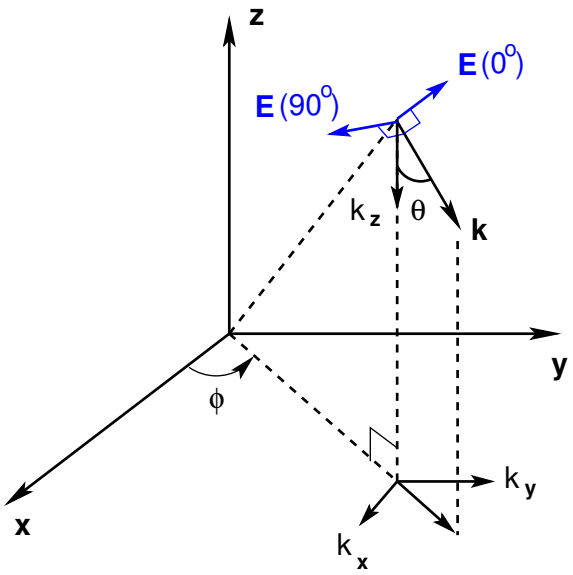


Figure 7: Definition of the parameters for the planewave source. Incidence direction **k** is specified by the angles θ and ϕ, while the polarization angle can be set to the one of the two orthogonal directions: in the plane, or orthogonal to the plane defined by the k_z and **k** vectors.

rectangle in 2D) (Figures 8-9) and provide the means to truncate in space the infinitely extended planewave. Figure 8 shows the YZ-cross-section of the computational domain, with an example of a planewave that propagates in the negative z- and positive y-direction.

When a periodic boundary condition is specified for any of the axis, then there will be no TFSF-boundary plane normal to that axis, Figure 9 . For example, if periodic boundary conditions are used for the x and y-axis, the only relevant TFSF boundaries are the top and bottom XY-planes. When a periodic boundary condition is set for some axis $w \in x, y, z$, then in order for the planewave to be a valid solution, the domain size L_w along the w-axis and the planewave wavevector $k_w(\theta, \phi)$ must satisfy the condition $L_w = n \times 2\pi/k_w$, where $n = 1, 2, ...$

The `TFSF-boundary` entry in the planewave specification is optional. It can be used in conjunction with the periodic boundary conditions for the x and y-axes to specify that only the top TFSF-boundary XY-plane should be used for a planewave source, as is the case in Figure 9 (right).

In the TFSF formulation, the planewave source is applied at the TFSF boundary, and the planewave propagates only in the Total Field region. Outside of the TFSF boundary only the Scattered Field is present. Simulations that use TFSF formulation for the planewave source, must have a refractive

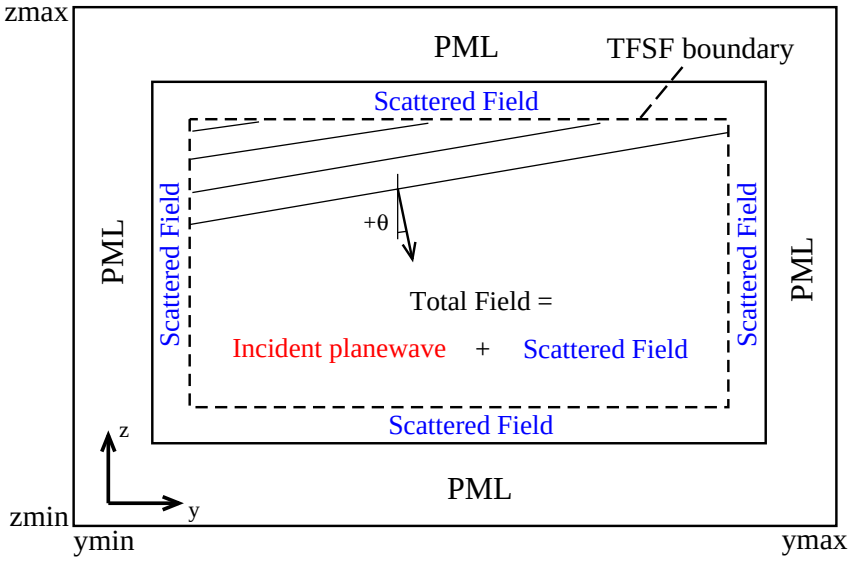


Figure 8: YZ-cross-section of the computational domain with Total Field/Scattered Field (TFSF) formulation for the planewave source. The refractive index must be the same at all sides of the TFSF boundary.

index distribution in the computational domain, such that the refractive index n_{ref} is same at all TFSF boundaries. Hence, the TFSF planewave source is appropriate for problems that involve interaction of a planewave with isolated (or periodic) objects embedded in a uniform medium, for example, scattering of a planewave from a chain of metallic nanospheres embedded in a glass substrate.

The Floquet boundary condition allows simulation of a planewave incident at an arbitrary angle on a structure that is periodic along the x- and/or y-axis. With this boundary condition the L_x and L_y can be arbitrary and do not have to be integer multiples of $2\pi/k_x$, $2\pi/k_y$. Therefore a single unit cell of the periodic structure can be modeled.

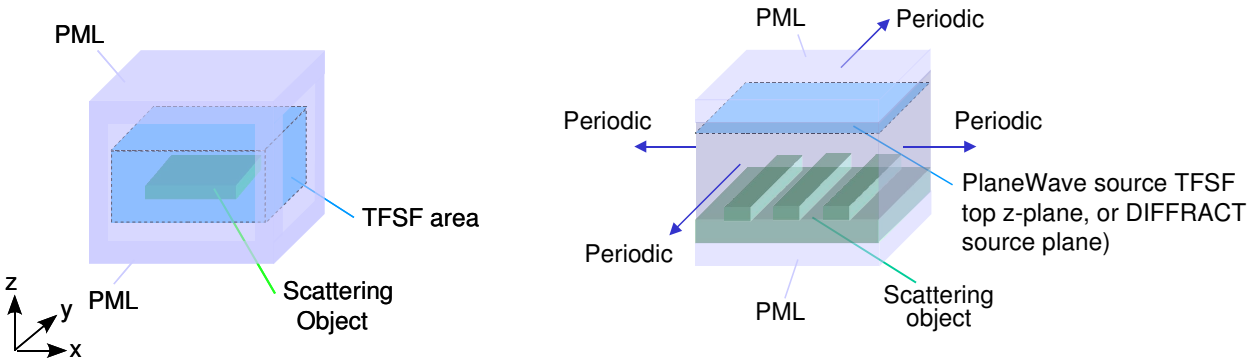


Figure 9: Left: computational domain with PML boundaries along all axis and Total Field/Scattered Field (TFSF) formulation for the planewave source. The scattering object is enclosed by the TFSF surface, the refractive index must be the same at all sides of the TFSF boundary. Right: computational domain with PML boundaries along the z-axis, and periodic (or Floquet) boundary conditions along the x- and y-axis. It is assumed in this plot that the `TFSF-boundary [top]` option is in effect for the planewave source.

4.17 Gaussian beam source specification

One or more gaussian beams can be specified for 2D or 3D computations by using a `GaussianBeamSource:`

```
GaussianBeamSource:
  Wavelength          [micron]       0.633
  Mode                [Hx_Ey_Ez]
  Exyz-component      [y]
  FWHM                [micron]       2.0 2.0
  beam-waist-offset   [micron]       0.0
  direction           [plus/minus]   1
  phase               [degrees]      90.0
  x0,y0,z0            [micron]       0.0 0.1 0.0
  xmin,xmax           [micron]       0.0 0.0
  ymin,ymax           [micron]      -5.0 4.0
  zmin,zmax           [micron]       0.2 0.2
```

The `Mode` can be `[Hx_Ey_Ez]`, `[Ex_Hy_Hz]` or `[3D]`. For the `[Hx_Ey_Ez]` mode, the polarization component can be set to `[y]` (for propagation along z) or `[z]` (for propagation along y). For the `[Ex_Hy_Hz]` mode, polarization is `[x]`, and for 3D computations it can be `[x]` or `[y]`. In 2D computations (in the YZ-plane) the beam propagates by default along the positive direction of the y or z-axis. In 2D the line segment, along which the source is ap-

plied, is specified by the `ymin,ymax` and `zmin,zmax` options, and `xmin,xmax` is ignored. Only one of the `ymin,ymax` or `zmin,zmax` pairs can have distinct values, hence specifying a line segment aligned with the y or z-axis. In the example above the source is applied from $y = -5\mu m$ to $y = +4\mu m$ along the y-axis, at $z = 0.2$, as specified by `zmin,zmax`. The `x0,y0,z0` specifies the beam center. In 2D only `y0` or `z0` is used to set the beam center position.

The `FWHM` specifies two numbers (in micron) for the amplitude full-width half-max of the beam width at $z = 0$. For the 3D case the amplitude FWHM values are related to w_{0x} and w_{0y} by FWHM=$w_{0x/y}2\sqrt{\ln 2}$. In 2D case w_{0x} and w_{0y} must be set to the same number and represent an amplitude FWHM equal to $w_0 2\sqrt{\ln 2}$. The w_0, w_{0x}, w_{0y} correspond to the formulas described below. The parameter `beam-waist-offset` sets an initial shift of the beam-waist plane from the source plane, and corresponds to the variable z in equations (1)-(3). The parameter `direction` is optional. It sets the direction of propagation along (+1) or against (-1) the positive direction of the y or z-axis in 2D (Figure 10), and along or against the positive direction of the z-axis in 3D. The default propagation direction is +1 in 2D computations.

```
GaussianBeamSource:
  Wavelength          [micron]    0.633
  Mode         [3D]
  Exyz-component      [y]
  FWHM                [micron]    1.5 1.5
  beam-waist-offset   [micron]    0.0
  x0,y0,z0            [micron]    0.0 0.0 0.0
  xmin,xmax           [micron]   -7.0 7.0
  ymin,ymax           [micron]   -7.0 7.0
  zmin,zmax           [micron]    0.4 0.4
```

In 3D computations the beam propagates by default along the z-axis in the negative direction, similar to the `DiffractSource`.

The phase parameter is also optional. It specifies the constant phase shift ϕ_0 to be added to the time harmonic dependence of the beam source, $\omega t + \phi_0$. The default value of the phase shift is zero.

The default propagation direction can be changed as discussed above. In 3D only `x0,y0` is used to set beam center, and `zmin,zmax` must both be equal and set to the desired location of the source plane along the z-axis.

The source is computed according to the following formula, $\mathbf{E}(x,y,z,t) = \hat{s}\exp\left[i(kz-\omega t)\right]\hat{E}$, with complex-valued $\hat{E}$ envelope given in 3D by:

$$\begin{aligned}\hat{E}(x,y,z) &= f(z)e^{-i[\phi_x(z)+\phi_y(z)]/2}e^{-x^2/w_x^2(z)-y^2/w_y^2(z)} \times \\ &\times \; e^{ik[x^2/(2R_x(z))+y^2/(2R_y(z))]}, \end{aligned} \tag{1}$$

where $f(z) = \sqrt{w_{0x}w_{0y}/[w_x(z)w_y(z)]}$, and

$$\phi_x(z) = atan\left(\frac{z}{l_{Dx}}\right), \quad w_x^2(z) = w_{0x}^2\left[1+\left(\frac{z}{l_{Dx}}\right)^2\right], \quad R_x(z) = z\left[1+\left(\frac{l_{Dx}}{z}\right)^2\right] \tag{2}$$

$l_{Dx} = \pi w_{0x}^2/\lambda$, and similar definitions apply to functions with y-label. In 2D ($\partial\hat{E}/\partial x \equiv 0$) the corresponding formulas are:

$$\hat{E}(y,z) = f(z)e^{-i\phi(z)/2}e^{-\rho^2/w^2(z)}e^{ik\rho^2/(2R(z))}, \tag{3}$$

$f(z) = \sqrt{w_0/w(z)}$, $\rho^2 = y^2$ *or* $\rho^2 = z^2$, and

$$\phi(z) = atan\left(\frac{z}{l_D}\right), \quad w^2(z) = w_0^2\left[1+\left(\frac{z}{l_D}\right)^2\right], \quad R(z) = z\left[1+\left(\frac{l_D}{z}\right)^2\right] \tag{4}$$

and $l_D = \pi w_0^2/\lambda$. In the equations, k is the wavenumber in the incidence medium, and λ is the wavelength in the incidence medium. The unit vector

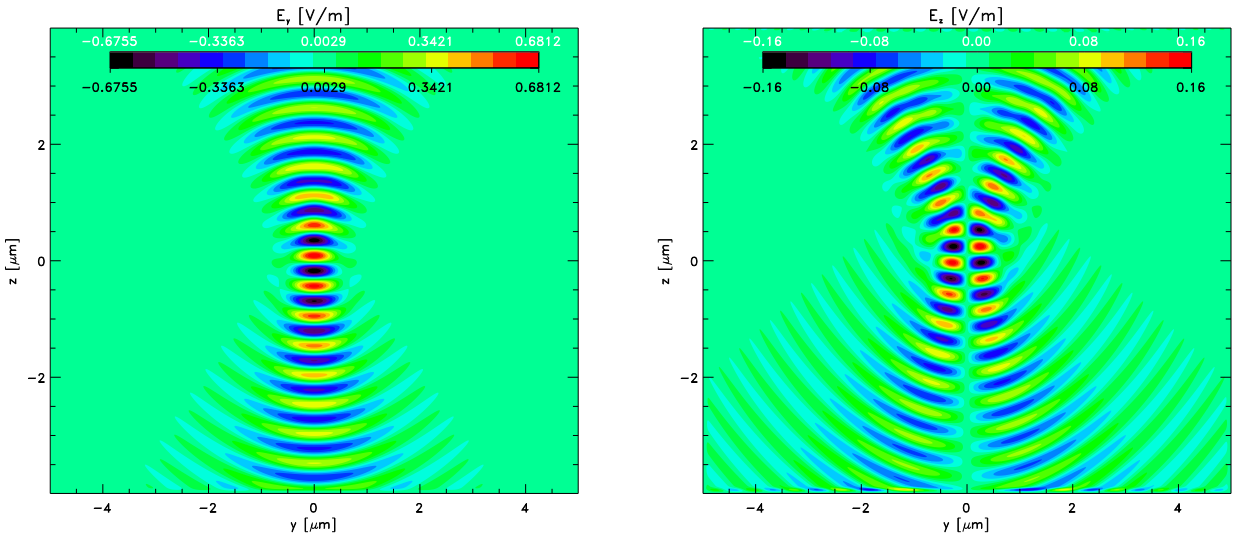


Figure 10: E_y and E_z components of a 2D Gaussian beam sourced along the y-axis at $z_{min} = z_{max} = 3.5\mu m$ with beam amplitude FWHM= $0.5\mu m$, $\lambda_0 = 0.65\mu m$, $n_{ref} = 1.3$, and direction set to -1. The z-parameter is $+4\mu m$, resulting (for a beam that propagates in the $-z$ direction), in initial focusing.

along the linear polarization direction is denoted by $\hat{s}$. Formulas (1)-(3)

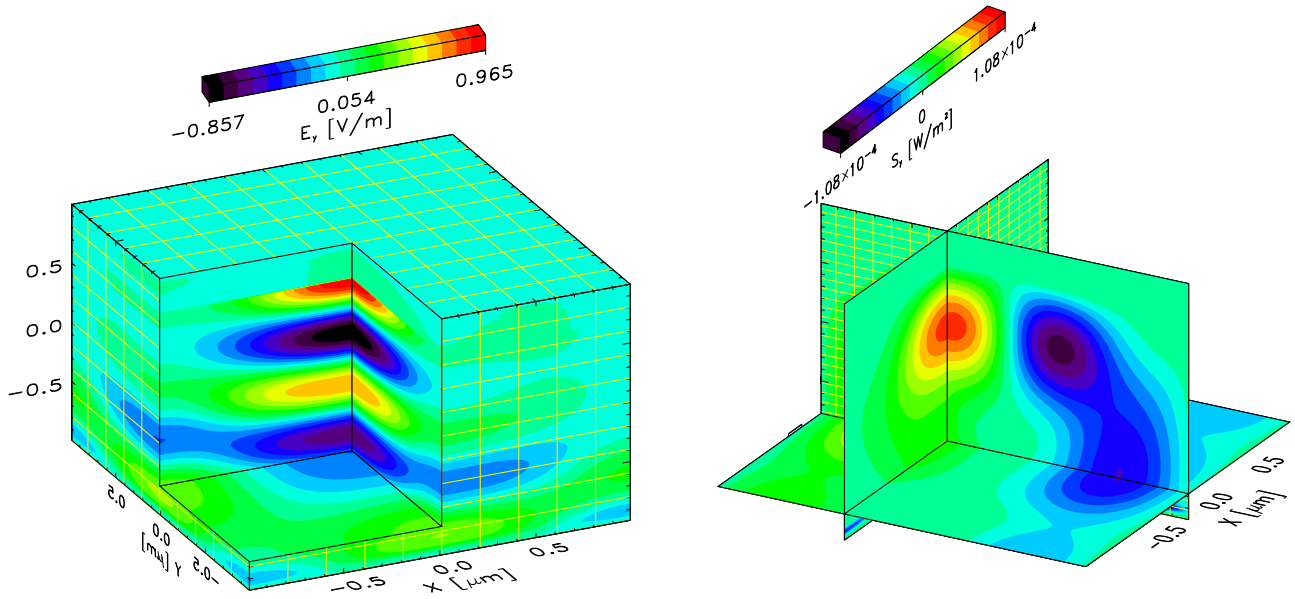


Figure 11: E_y and S_y components of a 3D Gaussian beam.

assume that z-axis is the direction of propagation, and are an exact solution to the Schrödinger equation,

$$\nabla^2 \hat{E}(\mathbf{x}) = -2ik\frac{\partial \hat{E}(\mathbf{x})}{\partial z} \tag{5}$$

derived from Maxwell's equations, assuming that $\frac{\partial^2 \hat{E}(\mathbf{x})}{\partial z^2}$ is small.

In the case of multiple beam sources, `GaussianBeamSource:` entries in the parameter file must be specified one after another.

4.18 2D source specification from the file

One or more arbitrary source distributions can be specified as a file input for two-dimensional computations. Example:

```
UserSource:
  Wavelength [micron]      0.9
  Mode       [Hx_Ey_Ez]
  Filename   [string]       usersource.dat
  ymin,ymax  [micron]      -1.5 1.5
  zmin,zmax  [micron]       1.0 1.0
```

Parameter `Mode` can be either `[Hx_Ey_Ez]` for a (H_x, E_y, E_z) 2D mode or

`[Ex_Hy_Hz]` for (E_x, H_y, H_z) mode. The `ymin,ymax, zmin,zmax` specify the extent of the source. The source is applied along the y or z aligned coordinate lines, so either `ymin,ymax` or `zmin,zmax` must be equal.

The ASCII text file is expected to contain on the first line the number of points present in the file. The following lines must contain 5 columns: coordinate of the point in microns, real part of Field1, imaginary part of Field1, real part of Field2, imaginary part of Field2. The fields Field1 and Field2 are defined as:

for `[Hx_Ey_Ez]` mode
Field1 is H_x
Field2 is E_z, if `ymin` equals `ymax`, or
Field2 is E_y, if `zmin` equals `zmax`,

for `[Ex_Hy_Hz]` mode
Field1 is E_x
Field2 is H_z if `ymin` equals `ymax`, or
Field2 is H_y, if `zmin` equals `zmax`.

The range of coordinates of the source points in the file may be a subset or superset of the range specified by `ymin,ymax` or `zmin,zmax`. The complex valued field amplitudes are read and interpolated into the FDTD grid. The interpolated values are written to an output file with the same name as the input source file plus a ".interpN" extension, where N is the rank of the processor that created the file. The content of the ".interpN" file is the same as that of the input source file, with two exceptions: the first line specifying the number of points is omitted, and there is an additional field (last entry) on each line, identifying material logical index along the source line.

The **E** and **H**-fields are sourced in time with time-harmonic ansatz $e^{-i\omega t}$ as $Re[H_x(y,z)e^{-i\omega t}]$, $Re[E_y(y,z)e^{-i\omega t}]$, etc. Appendix C Figure 38 gives a detailed description of the required staggered **E** and **H** field positioning of the user-defined source for the 2D Hx_Ey_Ez and Ex_Hy_Hz computations. In the case of multiple user defined sources, `UserSource` entries in the parameter file must be specified one after another.

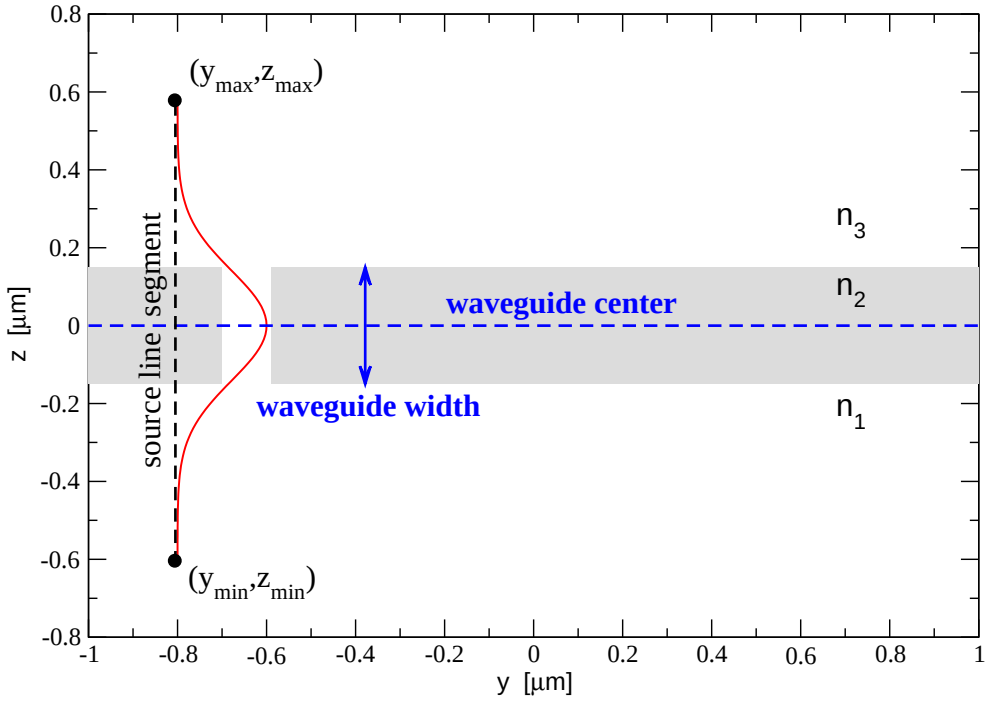


Figure 12: Example of a waveguide running along the y-axis, with the source applied on a line-segment along the z-axis, at $y_{min} = y_{max} = -0.8\mu m$. The refractive index distribution, waveguide width and center-line position are specified as part of the WaveguideSource definition, and should correspond to the geometry and material properties specified in the geometry and material definition files.

4.19 2D planar waveguide source specification

A planar waveguide source for Ex_Hy_Hz or Hx_Ey_Ez mode is specified by its effective index `n_eff` $= \beta/k$, where β is a mode propagation constant, and $k = 2\pi/\lambda$ is a free space wavenumber. The refractive index is $n2$ for the waveguide, and $n1, n3$ for the cladding layers. The position of the center and width of the waveguide, and refractive index values $n1$,$n2$,$n3$ must correspond to the structure and material properties specified in the geometry and material definition files. The sourced mode has a wavevector along the positive direction of the y- or z-axis.

```
WaveguideSource:
  Wavelength [micron]    0.65
  Mode       [Hx_Ey_Ez]
  ModeNumber    0
  n_eff,n1,n2,n3    1.49797 1.0 1.75 1.0
  center,width  [micron]   3.0 0.3
  ymin,ymax  [micron]    0.0 6.0
  zmin,zmax  [micron]    0.3 0.3
```

The `ymin,ymax, zmin,zmax` specify the extent of the source. The source is applied along the y or z aligned coordinate lines, so either `zmin,zmax` or `ymin,ymax` must be equal, Figure 12. For example, for a waveguide running along the z-axis, the waveguide mode is sourced across the waveguide along the y-axis, so `zmin` must be equal to `zmax` and equal to the desired z-location of the source line. The range of `ymin,ymax` should be large enough to cover the regions where the source field amplitudes are not negligible. If the `ymin,ymax,zmin,zmax` extend outside of the computational domain, they are reset to the edge of the computational domain or PML region.

4.20 File format version compatibility with DIFFRACT™

This entry is optional in the input file. It specifies the version of DIFFRACT™ software with which the DiffractSource and export files should be compatible:

```
ExportFileFormat [version] 8.2
```

The export files are produced by the "ExportReflected" and "ExportTransmitted" entries described in subsections 4.22.1-4.22.2.

If the file format version is not specified, the default is compatibility with DIFFRACT™ versions 8.4 and up. When using source files (subsection 4.14) created with DIFFRACT™ versions 8.3 and lower, it is **mandatory** to explicitly specify the file format version, otherwise the source distribution will be incorrect. When DIFFRACT™ source file is used, the ExportFileFormat is set to the same format as the source. The different file formats are as follows:

For DIFFRACT™ versions **less than 8.2**:

the file contains on the first line the refractive index n_r (assumed to be constant in the sampling plane), followed on the second line by N_x, N_y, L_x, L_y and complex-valued field distributions $E_x(x,y)$, $E_y(x,y)$, $E_z(x,y)$, followed by δ and $H_x(x,y)$, $H_y(x,y)$, $H_z(x,y)$. The fields are sampled on a uniform grid with steps L_x/N_x and L_y/N_y in x and y-directions. The variable δ is equal to the FDTD grid cell size in a direction normal to the plane, and at the position of the plane, on which E and H fields are sampled. Negative values of δ specify that $H_x(x,y), H_y(x,y)$ fields are sampled on a plane located along the z-axis a distance of $|\delta|/2$ behind the plane on which $E_x(x,y), E_y(x,y)$ fields are sampled (Figure 6), while positive δ corresponds to H-fields sampled on a plane $\delta/2$ ahead of the E-field plane. The δ and domain sizes L_x,L_y are in units of λ/n_r. The **E** and **H** fields are in normalized units defined in the DIFFRACT$^{\text{TM}}$ manual. For a beam propagating in DIFFRACT$^{\text{TM}}$ along the positive direction of the z-axis, the **H**-fields must be sampled half a cell behind **E**-fields, hence negative $\delta = -\Delta z$ must be specified in the FDTD/Export option of DIFFRACT$^{\text{TM}}$ when creating a file for DIFFRACT$^{\text{TM}}$ source option described in section 4.14.

For DIFFRACT$^{\text{TM}}$ versions **8.2 and 8.3**:

the file contains on the first line the refractive index n_r, the wavelength in the vacuum λ, the units of the wavelength (cm,mm,um,nm), followed on the second line by N_x, N_y, L_x, L_y and the field distributions with intervening δ as above. The domain sizes L_x and L_y, and δ are in units of the wavelength λ in the vacuum.

For DIFFRACT$^{\text{TM}}$ versions **8.4 and above** (default):

the same as the format of versions 8.2 and 8.3, but the **E** and **H** fields are in physical units of V/m and A/m.

4.21 Coordinate system transformation

In Diffract the beam nominally propagates along the positive z-axis. In Sim3D_Max the incident DiffractSource beam cross-section is intended for propagation along the negative z-axis, Figure 13. Hence, a coordinate transformation $z \to -z$, $y \to -y$ is applied to the $\mathbf{E}(x,y)$ and $\mathbf{H}(x,y)$ fields read from the Diffract source file. Similarly, the transmitted field obtained with

the ExportTransmitted entry in Sim3D_Max nominally propagates along the negative z-axis in the FDTD grid, and coordinate system rotation $-z \rightarrow z$, $-y \rightarrow y$ is applied to it upon export, so that the data can be imported into Diffract for propagation along the positive z-axis. The reflected field, obtained with ExportReflected entry in Sim3D_Max, propagates along the positive z-axis, and no coordinate system rotation is applied to it upon export. Its direction coincides with that obtained in DIFFRACT for the reflected beam that propagates in the $+z$ direction, Figure 13.

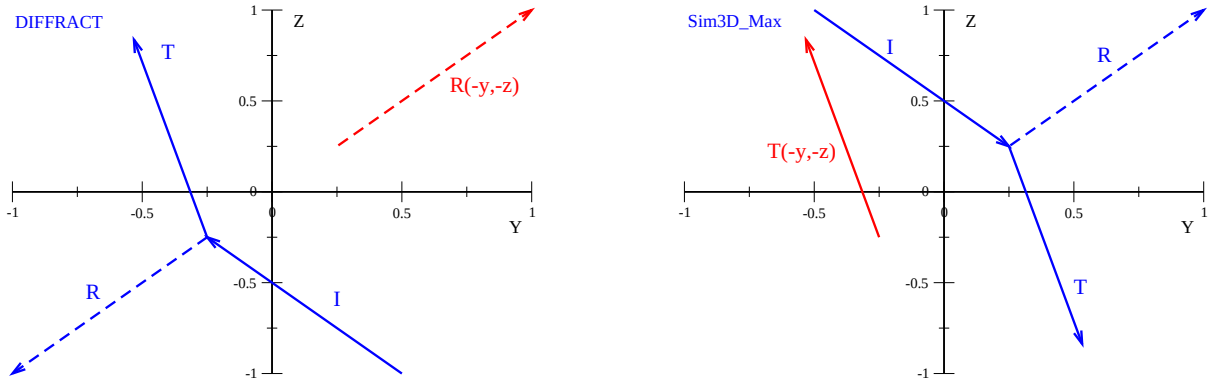


Figure 13: In DIFFRACT the incident and transmitted beams (solid blue lines) propagate along the $+z$ direction, hence the reflected beam (dashed blue line) propagates along the $-z$. Upon reflection, in DIFFRACT a coordinate system rotation is applied in order for the reflected beam to propagate along the $+z$ as well (dashed red line). In Sim3D_Max the coordinate system is rotated, so that the incident and transmitted beams propagate along the $-z$ direction, and reflected beam along the $+z$. When the CoordinateSystem [Diffract] option is used in Sim3D_Max, the coordinate system rotation is applied to the transmitted beam, in order for it to propagate along the $+z$, in agreement with the convention used in DIFFRACT.

The coordinate system rotations described above are applied by default to the field distributions of the source and ExportTransmitted XY-planes (but not XZ or YZ-planes) when Diffract input file is used as a source. The default can be changed with an optional entry:

```
CoordinateSystem [code]
```

where `code` can take a value of `Sim3d_Max` (no rotation) or `DIFFRACT` (coordinate system rotation is applied).

4.22 Export file specification

The spatial distribution of the amplitude and phase of the **E** and **H** fields on a specified plane, and at the frequency of the source, can be obtained using ExportReflected and ExportTransmitted entries. These objects can be used with any source, and produce files in a format compatible with the FDTD import/export interface of the DIFFRACT^{TM} software.

The difference between ExportReflected and ExportTransmitted entries is that the reflected field is sampled on the XY-plane at a predefined position, while the ExportTransmitted can specify any coordinate plane with arbitrary location. The ExportReflected plane will be positioned just above the source plane, when used with a PlanewaveSource or DiffractSource at its default location, so that only the reflected field is sampled.

Because of the possibility to arbitrarily place the source and the ExportTransmitted sampling planes, the meaning of "reflected" or "transmitted" may be lost, depending on the relative location of the sampling plane positions and the source plane location.

4.22.1 Export of the reflected field

This item is optional in the input file. It can be used to obtain complex-valued distribution of the fields in the XY-plane. The specified output file conforms to the file structure used in the FDTD import/export option of DIFFRACT^{TM}:

```
ExportReflected:
   Filename          fdtd.export.reflected
   File Format       ascii
   Grid Type         collocated
   NX,NY             256 256
```

The `File Format` can be one of `ascii`, `fortran_binary`, `c_binary`, and `Grid Type` can be `staggered` or `collocated` as described in sub-section 4.14. A collocated grid and ASCII or Fortran binary format must be used when generating files intended for input into DIFFRACT^{TM} via FDTD Import option. The `NX,NY` option specifies the desired number of points along the x and y axis. The line specifying `NX,NY` can be omitted. In that case the number of points will be set to the N_x, N_y of the DIFFRACT^{TM} input source

file 4.14, or, if another source is used, N_x, N_y are set to the number of points in the XY-plane of the FDTD grid. The scattered field complex-valued amplitude is computed in the xy plane at $z = z_{max} - (n_{z\,PML} + 2)\Delta z_{top}$ via the Discrete Fourier Transform of the time dependent solution, applied in the time interval $[t_{max} - 4\lambda/c, t_{max}]$[2], where Δz_{top} is the grid spacing used near the PML region at the top of the computational domain. There can be only one `ExportReflected:` entry in the input file, and its position on the grid is always at the top of the domain as specified above, regardless of the position along the z-axis of any of the sources. When used with DiffractSource or PlanewaveSource at their default locations, the ExportReflected plane will be just above the source plane and will sample only the reflected fields.

4.22.2 Export of the transmitted field

If `ExportReflected:` entry described above is specified, it must also be followed by at least one `"ExportTransmitted:"` entry. This entry is similar to the entry for the reflected field, but its plane has a user-defined, rather than default, position. If an `"ExportTransmitted:"` entry is not necessary in the computation, its position can be set to be outside of the computational domain, and it will be ignored.

The `ExportTransmitted:` entry sets the filename of the export file into which to write complex-valued amplitude distribution of the E_x, E_y, E_z and H_x, H_y, H_z fields computed at the specified plane. The file content and parameters are the same as for the reflected field described above.

```
ExportTransmitted:
   Filename              fdtd.export.transmitted
   File Format           fortran_binary
   Grid Type             collocated
   z-location [micron]   -50e-3
   NX,NY                 512 512
```

In addition, `z-location` specifies the position of the plane along the z-axis, in micrometers. If this position is outside of the computational domain bounds, no output file will be produced. The line `z-location` corresponds to fields sampled in the XY-plane. Similarly, `y-location` or `x-location`

[2]Note, that changing z_{max} will change the Diffract source and export reflected field plane positions. Also, t_{max}, is set independently of z_{max}, and should be chosen large enough to get time-harmonic converged solution.

can be used to sample fields in the XZ or YZ planes. The line specifying the number of points in the sampling plane (`NX,NY` for the XY-plane, `NX,NZ` for the XZ-plane, or `NY,NZ` for the YZ-plane) can be omitted, in which case the number of points will default to the values from the DIFFRACT^{TM} source file 4.14, or, if DIFFRACT^{TM} source is not used, to the number of points in the corresponding plane of the FDTD grid.

After the line specifying the number of points, an optional specification of the sampling region can follow. For example in the YZ-plane one can specify an area centered on the point (0,0) as `LY,LZ [micron] 2.4 1.0`. When not specified, the sampling region will be set to the computational domain size for the corresponding cross-section. One exception is when DIFFRACT^{TM} source is used: then the sampling region in the XY-plane is set to L_x, L_y values read from the DIFFRACT^{TM} source file.

The reflected or transmitted field magnitude and phase data can be requested as follows:

```
magnitude,phase  Ex,Ey,Ez?  yes yes yes
magnitude,phase  Hx,Hy,Hz?  yes yes yes
Poynting vector  Sx,Sy,Sz?  yes yes yes
current density  Jx,Jy,Jz?  yes yes yes
```

All of the above lines are optional, and when omitted, or when set to "no", the corresponding output files are not generated. When specified, a folder is created in the working directory into which the files are written under the names "mEx.dat", "mEy.dat", etc. for the magnitude data, and "pEx.dat", "pEy.dat", etc. for the phase data (in radians). For the real-valued Poynting vector only amplitude data "Sx.dat", etc., is generated. The files are written in a text (ASCII) format and in the same "xy" order as the data in the export file. The magnitude and phase folder has the same name as the export file, but without the filename extension. If the export filename does not have an extension, a suffix "-mp" is added to it to create the folder name.

There can be multiple "ExportTransmitted:" entries, specified one after another, for sampling the computational domain with different planes and at various locations. Note however, that when an export plane is at a location such that it samples a region with refractive index variation in that plane, then the refractive index value stored in the export file is not well defined,

since it will represent only one value of the refractive index distribution in the export plane. In such cases magnitude and phase distributions of the exported fields are not suitable for propagation in the DIFFRACT^{TM} software, since it requires constant refractive index in the export plane. However, the data can be visualized or processed otherwise.

4.23 Monitor header comment

A sequence of characters can be added at the beginning of the first line of the Monitor files described in section 4.24. For example,

```
MonitorFileComment  [string]   #
```

will put # as the first character on the first line (which describes the monitor type, number of points, etc.) in all Monitor files. This entry is optional in the input file.

4.24 Monitor specification

Monitors are optional in the input file. They can be used to sample fields at specified points and within a specified time interval. Parameter `Mode` can be either `[3D]`, `[Hx_Ey_Ez]` for a (H_x, E_y, E_z) 2D mode or `[Ex_Hy_Hz]` for (E_x, H_y, H_z) 2D mode. The `Type` can be either `[fourier-transform]` or `[time-history]`. The output file specified under `Filename` is an ASCII text file.

```
Monitor:
  Mode        [Hx_Ey_Ez]
  Type        [fourier-transform]
  Filename    [string]       monitor1.out
  xmin,xmax   [micron]        0.0  0.0
  ymin,ymax   [micron]       -1.0  1.0
  zmin,zmax   [micron]        0.0  0.0
  t_on,t_off  [periods]       8
  bandwidth   [frequency-interval]
  fmin,fmax   [THz]           285.0 315.0
```

Another monitor example:

```
Monitor:
  Mode        [Ex_Hy_Hz]
  Type        [time-history]
  Filename    [string]        monitor2.out
  xmin,xmax   [micron]         0.0  0.0
  ymin,ymax   [micron]        -1.0  1.0
  zmin,zmax   [micron]         0.0  0.0
  t_on,t_off  [nanoseconds]   10e-6 20e-6
  sampling-factor [integer] 5
```

During parallel computation with N_p CPUs, each processor writes to its own output file, so processor rank (an integer number, between 0 and $N_p - 1$) is appended to the output filename, e.g. `monitor1.out3`. The content of the output file depends on the `Mode` entry, and is described below. With each new simulation, the existing monitor files with the same name as specified under the `Filename` (with appended processor numbers), are **removed**.

The `xmin,xmax, ymin,ymax, zmin,zmax` specify the extent of the monitor. The `t_on,t_off` sets the monitor switch-on and switch-off times, and can be specified either as

```
t_on,t_off [nanoseconds]     10e-6 20e-6
```

or

```
t_on,t_off [periods]     10
```

In the case when number of `[periods]` P is specified, the sampling in time is done from $t_{max} - P \times T$ to t_{max}, where $T = \lambda/c$ is the period of the source.

For the `[fourier-transform]` monitor, the `bandwidth` entry must be present with one of the options:

`bandwidth [source-frequency]`, `bandwidth [frequency-interval]` or `bandwidth [wavelength-interval]`.

The first option, `[source-frequency]`, specifies that Fourier Transform should be evaluated at a single point in the frequency space - the input source frequency.

The `[frequency-interval]` (or `[wavelength-interval]`) option allows Fourier Transform to be computed for a range of frequencies (or wavelengths), specified on the next line with an entry `fmin,fmax` (or `lmin,lmax`),

for example: `fmin,fmax [THz] 285.0 315.0` or `lmin,lmax [nm] 850.0 920.0`.

The sampling-factor specification is optional. It can be used to set the decimation factor for sampling fields in time. For example `sampling-factor [integer] 5` sets monitor data processing to occur only every 5th time step. The default sampling factor is 1.

The output of the monitor of time-history of fields on a specified area (which also generates time history of the area integral of the Poynting vector component $\mathbf{S_n}$ normal to the plane) can be controlled by specifying on the last line of the monitor entry whether to create files with time-history of **E,H** and **S** (`output [distribution]`) or only time-history of the integral of $\mathbf{S_n}$ (`output [integral]`), or both (`output [all]`). When output type is not specified, the default is `output [all]`.

4.24.1 Monitors for computations in 2D

For TE_x/TM_x **computations in 2D**, the `xmin,xmax` values are ignored, and only one of the `ymin,ymax` or `zmin,zmax` pairs must have distinct min/max values. The monitor then specifies a line segment aligned with the y or z-axis. Along this line either the time history or Discrete Fourier Transform of the $\mathbf{E}, \mathbf{H}$ fields and the Poynting vector $\mathbf{S} = \mathbf{E} \times \mathbf{H}$ are sampled and written to the specified file.

For a `[time-history]` monitor, the output file contains on the first line the computation mode, monitor type, number of monitor points in space and number of points sampled in time, e.g. `[Ex_Hy_Hz] [time-history] Ny Nz Ntime`.

The rest of the file consists of one line for each point in space and time with the following 8 columns: time in nanoseconds, y,z-coordinates of the point in microns, Field1, Field2, Field3, Field4, Field5. The fields Field1 through Field5 are defined as:

for `[Hx_Ey_Ez]` mode
Field1 is H_x, Field2 is E_y, Field3 is E_z,
Field4 is S_y, Field5 is S_z,

for `[Ex_Hy_Hz]` mode

Field1 is E_x, Field2 is H_y, Field3 is H_z,
Field4 is S_y, Field5 is S_z.

For a `[fourier-transform]` monitor, the file contains on the first line the computation mode, monitor type, number of monitor points in space and number of points sampled in frequency, e.g.
`[Ex_Hy_Hz] [fourier-transform] Ny Nz Nfreq`.
If `bandwidth [source-frequency]` option was selected, the `Nfreq` will be equal to 1.

The rest of the file consists of one line for each point in space and frequency with the following 11 columns: frequency value in THz, y,z-coordinates of the point in microns, Field1, Field2, Field3, Field4, Field5, Field6, Field7, Field8. The fields Field1 through Field8 are defined as:

for `[Hx_Ey_Ez]` mode
Field1, Field2 are $Re(H_x), Im(H_x)$,
Field3, Field4 are $Re(E_y), Im(E_y)$,
Field5, Field6 are $Re(E_z), Im(E_z)$,
Field7 is real $< S_y >$, Field8 is real $< S_z >$,

for `[Ex_Hy_Hz]` mode
Field1, Field2 are $Re(E_x), Im(E_x)$,
Field3, Field4 are $Re(H_y), Im(H_y)$,
Field5, Field6 are $Re(H_z), Im(H_z)$,
Field7 is real $< S_y >$, Field8 is real $< S_z >$.

The complex valued amplitudes of the **E** and **H** fields are computed via time Discrete Fourier Transform of the real valued **E** and **H** FDTD variables. The DFT is evaluated at the source frequency if `bandwidth` option is set to `[source-frequency]`. The real valued components of the time averaged Poynting vector are computed from complex valued **E** and **H** fields, as $< \mathbf{S} > = \frac{1}{2} Re(\mathbf{E} \times \mathbf{H}^*)$. This formula assumes a continuous wave time-harmonic dependence of fields.

4.24.2 Monitors for computations in 3D

For **3D computations**, parameter `Mode` is set to `[3D]`, and `xmin,xmax`, `ymin,ymax` or `zmin,zmax` can be used to specify a line segment or a plane section along which the monitor is applied. For a line segment, only one of the min/max coordinate pairs can have distinct values, so that the line segment is aligned with one of the x,y or z-coordinate directions. For a plane section, only two of the min/max coordinate pairs can have distinct values, so that the section is in the XY, YZ or XZ plane.

In 3D, for a monitor type `[time-history]`, the first line specifies the 3D computation mode, monitor type, number of monitor points in space and number of points sampled in time, e.g. `[3D] [time-history] Nx Ny Nz Ntime`. The output file contains one line for each point in space and time with the following 13 columns: time in nanoseconds, x,y,z-coordinates of the point in microns, Ex,Ey,Ez, Hx,Hy,Hz, Sx,Sy,Sz.

For a 3D `[fourier-transform]` monitor, the output file contains one line for each point in space with the following 19 columns: frequency value in THz, x,y,z-coordinates of the point in microns, $Re(Ex)$, $Im(Ex)$, $Re(Ey)$, $Im(Ey)$, $Re(Ez)$, $Im(Ez)$, $Re(Hx)$, $Im(Hx)$, $Re(Hy)$, $Im(Hy)$, $Re(Hz)$, $Im(Hz)$, Sx, Sy, Sz.

For a plane monitor in 3D, an additional file is generated, containing the area integral of the Poynting vector component normal to the plane as a function of time or frequency. The filename of this file is the same as that of the monitor file, but with extension ".is" added before the processor rank.

An integral of the electromagnetic energy $(\mathbf{E}\cdot\mathbf{D}+\mathbf{H}\cdot\mathbf{B})/2$ over a specified volume containing only dielectric materials, can be monitored as a function of time with an `[energy]` monitor. A volume is specified with all min/max coordinate pairs having distinct values. Detailed summary of the monitor file formats is given in Appendix D.

There can be multiple "Monitor:" entries in the parameter file, specified one after another.

5 Boundary Conditions File Description

The file specified under the `Boundary Conditions Filename:` entry of the parameter file (see 4), contains a definition of the BCs used in the simulation.

Currently, boundary types `[PEC]`, `[periodic]`, `[Floquet]` or `[PML]` can be used. The selected boundary condition applies to both top and bottom boundaries of the specified axis.

- The Perfect Electric Conductor (PEC) boundary condition sets the tangential electric field components to zero on the boundary.
- The periodic boundary condition allows simulation of the periodic systems in 2D or 3D.
- The Floquet boundary condition is used with the planewave source (sec. 4.16) to model periodic systems in 2D or 3D, with arbitrary planewave incidence angles.
- The Perfectly Matched Layer (PML) option sets up an absorbing layer that simulates an open boundary condition.

An example of the content of the boundary conditions file, that sets the x-axis to a 15 point PML with default PML parameters for σ and κ, y-axis to periodic and z-axis to PEC:

```
BoundaryConditions:
  x-axis    [PML]
    nz_pml          15
    sigma,kappa     0.0 0.0
  y-axis    [periodic]
  z-axis    [PEC]
```

If the desired boundary type is PML for any of the axes, the next line after `...-axis [PML]` must specify for that axis the number of points in the absorbing layer, followed by `sigma,kappa` - the values of σ_{max} and κ_{max} at the end of the absorbing layer. A polynomial of order $m_{pml} = 3$ is used in computing the grading of the PML layers. If σ_{max} or κ_{max} are set to zero, they will assume their respective default values of $\kappa_{max} = 1.0$ and $\sigma_{max} = 8 \times (m_{pml} + 1)/(n_{pml}\Delta\sqrt{\mu_0/\epsilon_0})$, where Δ is the cell size in the PML region. Inside the PML layers the material refractive index and grid cell sizes **should not have** any variation in the direction normal to the PML layer.

If the boundary specification file is empty, the default boundary condition is PML absorbing boundary for all axes, with a default number of points in the PML layers set to $n_{x,y\,PML} = 10$ and $n_{z\,PML} = 15$. This results in

approximately -40 db reflection for dielectric media. The default settings can be modified using a specification of the boundary type for each axis.

The Floquet boundary condition, used with the planewave source, can be set with an entry `x-axis [Floquet]` or `y-axis [Floquet]` for the x- or y-axis.

6 Material File Description

The file specified under the `Material Definition Filename:` entry of the parameter file (see 4), contains a definition of the material properties used in the simulation. The file can contain an arbitrary number of material definitions. Each material can be of one of the predefined types: `dielectric`, `dielectric(x,y)`, `Debye`, `Lorentz`, etc. Each material declaration starts with two lines:

```
Material Label  [string]       YourLabel
    model       [MaterialType]
```

where `YourLabel` is a unique arbitrary word (except for the reserved word `Vacuum`) chosen by the user to identify each material. The material label `Vacuum` is predefined as a dielectric with permittivity ϵ_0 and permeability μ_0. `Material Type` is one of the available types: `dielectric`, `Debye`, `Lorentz`, etc. For example,

```
Material Label  [string]       SiO2
    model       [dielectric]
```

```
Material Label  [string]       Gold
    model       [Debye]
```

```
Material Label  [string]       AluminumOxide
    model       [Lorentz]
```

These two lines are followed by a set of parameters specific to the material type.

6.1 Dielectric materials

For the type `dielectric`, the material properties can be entered in any one of the following equivalent formats:

```
Material Label       [string]          GaAs750nm
    model            [dielectric]
    sqrt Re(eps)     [dimensionless]   3.6986
    conductivity     [1/(ohm*m)]       16455.9
```

```
Material Label      [string]          GaAs750nm
  model             [dielectric]
  n-ki              [dimensionless]   3.7-0.1i

 Material Label     [string]          GaAs750nm
  model             [dielectric]
  eps               [relative]        13.68-0.74i
```

When used with a *monochromatic* source, materials with a complex refractive index $n - ki$ can be modeled by the material type `dielectric` if $n^2 - k^2 > 1$ (in the case of arbitrary n and k refer to the Debye material model). The refractive index $n - ki$, permittivity ϵ and conductivity σ are related by $\epsilon = \epsilon_r + i\epsilon_i = (n - ki)^2$, $\sqrt{Re(\epsilon)} = \sqrt{n^2 - k^2}$ and $\sigma = 2nk\epsilon_0\omega$, where ω is the frequency of the source.

Arbitrary continuous variation of the refractive index and conductivity in the XY-plane can be realized with material model `dielectric(x,y)`. This model is useful for simulating optical disk data storage components in which optical constants are continuous functions of the temperature in the plane of the disk, $T = T(x, y)$. For the material type `dielectric(x,y)`, there is one input parameter: the name of the file with the distribution of the optical constants, e.g.

```
Material Label       [string]          GST
  model              [dielectric(x,y)]
  filename           [string]          GST.input
```

The data file is an ASCII file containing on the first line a single word, e.g. `Dielectric(x,y)`, followed on the second line by:
Lx_{min} Lx_{max} Ly_{min} Ly_{max} Nx Ny (n_{bg} σ_{bg})

where Lx_{min}, Lx_{max}, ... specify the domain size (in nanometers) in which the data is defined, Nx, Ny are the number of rows, columns in the file, and $(n_{bg}\ \sigma_{bg})$ are the background refractive index and conductivity. Subsequent lines must contain entries $(n_{ij}\ \sigma_{ij})$ for each point i, j of the grid defined on the first line. The units of n and σ must be the same as in the material model `[dielectric]` described above. The data in the file is read in the order indicated by the following pseudo-code segment:

```
for i=1 to Nx
  for j=1 to Ny
     read (n[i][j] sigma[i][j])
```

Then the data is interpolated to the FDTD computational grid. The grid points that are outside of the range specified by Lx_{min}, Lx_{max}, Ly_{min}, Ly_{max}, are assigned background optical constant values $(n_{bg}\ \sigma_{bg})$.

A more general distribution of the refractive index and conductivity, dependent on all of the coordinates, can be realized with the material model `dielectric(x,y,z)`.

```
Material Label       [string]      GST-T
  model              [dielectric(x,y,z)]
  filename           [string]      epsxyz.dat
```

The input data file is an ASCII file containing on the first line a single word, for example `Dielectric(x,y,z)`, followed by $n_x \times n_y \times n_z$ entries $(n_{ijk}\ \sigma_{ijk})$ for each point i, j, k of the grid defined in section 4.3. The data is read from the file in the following order:

```
for i=1 to nx
  for j=1 to ny
    for k=1 to nz
       read (n[i][j][k] sigma[i][j][k])
```

6.2 Debye model

The parameters for a dispersive medium based on the single-pole `Debye` model correspond to a complex-valued frequency-domain susceptibility func-

tion $\chi(\omega)$ and permittivity function $\epsilon(\omega)$:

$$\chi(\omega) = \frac{\Delta\epsilon}{1 + i\omega\tau}, \qquad \epsilon(\omega) = \epsilon_\infty + \chi(\omega) - i\frac{\sigma}{\omega\epsilon_0} \tag{6}$$

with τ - the pole relaxation time, ϵ_∞ - the relative permittivity at infinite frequency, $\Delta\epsilon = \epsilon_s - \epsilon_\infty$, where ϵ_s is the static or zero-frequency relative permittivity. An example set of parameters for gold ($n = 0.16 - 3.95i$) at $\lambda = 700nm$:

```
Material Label      [string]       Gold
    model           [Debye]
    tau             [femtosec]     4.84371
    eps_inf         [relative]     1.0
    delta_eps       [relative]    -2832.73
    conductivity    [1/(ohm*m)]    5.17818e6
```

The Debye model can be used to simulate materials which have a dispersion relation that can be approximated by the expression (6) over some frequency band.

The Debye model can also be used to represent materials with complex-valued refractive index $n - ki$ at a specified frequency. In such cases, given two parameters, n and k, the choice of four Debye model parameters is not unique, Figure 14. At any given frequency ω the real and imaginary parts

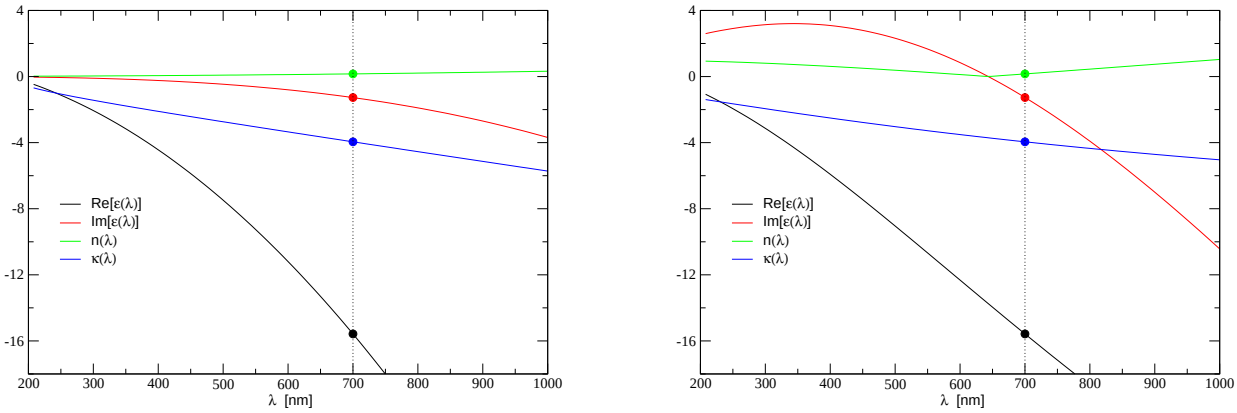


Figure 14: Dependence of complex-valued permittivity and refractive index on the wavelength $\lambda = 2\pi c/\omega$ computed using single-pole Debye model (6) with two sets of model parameters for gold ($n = 0.16 - 3.95i$) at $\lambda = 700nm$.

of the equation $\epsilon = (n - ki)^2$ provide two conditions for the choice of Debye parameters:

$$n^2 - k^2 = \epsilon_\infty + \frac{\Delta\epsilon}{1 + \omega^2\tau^2},$$
$$-2nk = -\frac{\Delta\epsilon\omega\tau}{1 + \omega^2\tau^2} - \frac{\sigma}{\omega\epsilon_0},$$

which can be rewritten in the following form:

$$\Delta\epsilon = (1 + \omega^2\tau^2)(n^2 - k^2 - \epsilon_\infty), \tag{7}$$
$$-2nk + \frac{\sigma}{\omega\epsilon_0} = -\omega\tau(n^2 - k^2 - \epsilon_\infty), \tag{8}$$

By setting $\epsilon_\infty = 1$, parameters σ and τ are determined by equation (8), linear in both unknowns, with $\Delta\epsilon$ following from equation (7). The Debye model corresponding to a given permittivity at source frequency can be realized by specifying $n - ki$ or ϵ in the material definition file, for example:

```
Material Label        [string]       Aluminum
  model               [Debye]
  n-ki                [dimensionless] 2-7i

Material Label        [string]       Gold
  model               [Debye]
  eps                 [relative]    -15.0-1.2i
```

A multi-pole Debye model corresponding to the relative permittivity

$$\epsilon(\omega) = \epsilon_\infty + \sum_{p=1}^{N_p} \frac{\Delta\epsilon_p}{1 + j\omega\tau_p} - j\frac{\sigma}{\omega\epsilon_0}$$

can be set by specifying the number of poles N_p, followed by the list of parameters for each pole, in the format shown below:

```
Material Label       [string]      M1
    model            [multipole-Debye]
    poles            [integer]     2
    eps_inf          [relative]    ...
    conductivity     [1/(ohm*m)]   ...
    tau0             [femtosec]    ...
    delta_eps0       [relative]    ...

    tau1             [femtosec]    ...
    delta_eps1       [relative]    ...
```

Similarly to the dielectric material model, arbitrary continuous variation of the single-pole Debye model parameters in the XY-plane can be realized with material type `Debye(x,y)`. This model may be used, for example, to simulate optical disk data storage components in which material parameters are continuous functions of the temperature in the plane of the disk, $T = T(x, y)$. For the material type `Debye(x,y)`, there is one input parameter: the name of the file with the distribution of Debye model parameters, e.g.

```
Material Label       [string]          PtOx
  model              [Debye(x,y)]
  filename           [string]          PtOx.input
```

The data file is an ASCII file containing on the first line a single word, e.g. `Debye`, followed on the second line by:
Lx_{min} Lx_{max} Ly_{min} Ly_{max} Nx Ny (τ_{bg} $\epsilon_{\infty\, bg}$ $\Delta\epsilon_{bg}$ σ_{bg})
where Lx_{min}, Lx_{max}, ... specify the domain size (in nanometers) in which the data is defined, Nx, Ny are the number of rows, columns in the file, and the rest specify background Debye model parameters. Subsequent lines must contain entries (τ_{ij} $\epsilon_{\infty\, ij}$ $\Delta\epsilon_{ij}$ σ_{ij}) for each point i, j of the grid defined on the first line. The units of parameters must be the same as in the material model `[Debye]` described above.

The data is read from the file in the same order as for the `dielectric(x,y)` material described above, and interpolated to the FDTD computational grid. The grid points that are outside of the range specified by Lx_{min}, Lx_{max}, Ly_{min}, Ly_{max}, are assigned background values (τ_{bg} $\epsilon_{\infty\, bg}$ $\Delta\epsilon_{bg}$ σ_{bg}).

6.3 Lorentz model

The parameters for a dispersive medium based on the single-pole `Lorentz` model correspond to a complex-valued frequency-domain susceptibility function:

$$\chi(\omega) = \frac{\Delta\epsilon\omega_0^2}{\omega_0^2 + 2j\omega\delta - \omega^2}, \qquad \epsilon(\omega) = \epsilon_\infty + \chi(\omega) - j\frac{\sigma}{\omega\epsilon_0}$$

where ω_0 - is the pole frequency, ϵ_∞ - the relative permittivity at infinite frequency, $\Delta\epsilon = \epsilon_s - \epsilon_\infty$, ϵ_s is the static or zero-frequency relative permittivity, δ is the damping coefficient. Example parameters for aluminum ($n = 2 - 7i$) at $\lambda = 650nm$:

```
Material Label      [string]      Aluminum
    model           [Lorentz]
    omega0          [rad/s]       23.536118e14
    delta           [rad/s]       2.9504974e14
    eps_inf         [relative]    1.81
    delta_eps       [relative]    32.802
    conductivity    [1/(ohm*m)]   0.0
```

Multi-pole Lorentz model corresponding to the relative permittivity

$$\epsilon = \epsilon_\infty + \sum_{p=1}^{N_p} \frac{\Delta\epsilon_p\omega_p^2}{\omega_p^2 + 2j\omega\delta_p - \omega^2} - j\frac{\sigma}{\omega\epsilon_0}$$

can be set by specifying the number of poles N_p, followed by ϵ_∞ and σ, and the list of parameters for each pole, in the format shown below:

```
Material Label      [string]      M2
    model           [multipole-Lorentz]
    poles           [integer]     2
    eps_inf         [relative]    ...
    conductivity    [1/(ohm*m)]   ...
```

```
    omega0          [rad/s]       ...
    delta0          [rad/s]       ...
    delta_eps0      [relative]    ...

    omega1          [rad/s]       ...
    delta1          [rad/s]       ...
    delta_eps1      [relative]    ...
```

6.4 Drude model

The parameters for a dispersive medium based on the single-pole `Drude` model correspond to a complex-valued frequency-domain susceptibility function:

$$\chi(\omega) = \frac{\omega_p^2}{2j\omega\delta - \omega^2}, \qquad \epsilon(\omega) = \epsilon_\infty + \chi(\omega)$$

where ω_p - is the plasma frequency, ϵ_∞ - the relative permittivity at infinite frequency, δ is the damping coefficient. Example input parameters for Drude model:

```
Material Label      [string]      Metal
  model             [Drude]
  delta             [rad/s]       1e13
  eps_inf           [relative]    1.0
  omega_p           [rad/s]       5e12
```

6.5 Magnetic material model

The `[magnetic*]` material type allows specification of magnetic permeability and electric permittivity for materials with constant $\mu \neq \mu_0$ or with magnetic dispersion. Example input parameters for constant ϵ and μ:

```
Material Label      [string]   MagMat
  model             [magnetic1]
  permittivity      [relative]    2.25
  permeability      [relative]    2.0
```

Drude model is used to model electric and magnetic dispersion of magnetic materials. Drude model parameters $\epsilon_\infty, \delta, \omega_p$ can be specified instead of the constant permittivity, resulting in $\epsilon(\omega)$ frequency dependence given in subsection 6.4. Similarly, $\mu_\infty, \delta_m, \omega_m$ can be specified instead of the constant permeability, with frequency dependent $\mu(\omega)$ given by:

$$\mu(\omega) = \mu_\infty + \frac{\omega_m^2}{2j\omega\delta_m - \omega^2}.$$

Example input parameters for dispersive magnetic material:

```
Material Label      [string]   MetaMat
  model             [magnetic2]

  eps_inf           [relative]    1.0
  delta             [rad/s]   1.25e14
  omega_p           [rad/s]   8.0e15

  mu_inf            [relative]    1.0
  delta_m           [rad/s]   2.5e14
  omega_m           [rad/s]   6.0e15
```

Debye(x,y), dielectric(x,y),dielectric(x,y,z) and magnetic material models currently can be used only in 3D computation mode, and should not extend into PML layers.

7 Geometry File Description

The file specified under the `Geometry Definition Filename:` entry of the parameter file (see 4), contains definitions of the structures to be set up in the computational domain. The default (i.e. empty geometry definition file) is a free space computational domain occupied by the predefined material `Vacuum`, see section 6. More complex structures may be simulated by adding any number of geometric objects to this free space computational domain. Currently defined basic geometric primitives are: layer, cube, ellipsoid, sphere; triangular, rectangular, circular, elliptical, capped-rectangular apertures or marks of finite thickness; and different lattice types. Also predefined are geometric objects commonly used in optical data storage research: bumps and pits, grooves, conformal layers.

7.1 Geometry specification from an input file

An arbitrary distribution in the computational domain of a finite number of materials, can be imported from a file using the following entry in the geometry definition file:

```
ReadGeometryFile
  Filename [string] user_geom.dat
```

The specified file has the same structure as the mindex output file described in 4.9. It must be an ASCII text file and is expected to contain on the first line the number of points n_x, n_y, n_z, which should match the number of points specified in the input parameters file. The rest of the input file should contain the logical enumeration value, m, of the material for each cell i, j, k of the computational grid. This logical value is simply an integer number ($m \geq 0$) corresponding to the order in which materials are defined, one after another, in the material definition file. Materials are counted in the material definition file starting from 1. The value 0 corresponds to the predefined material $Vacuum$. The order of the cell input is one, in which k changes from 1 to n_z first, then j from 1 to n_y, then i from 1 to n_x. The material assignment for each cell, read from the file, overwrites any geometry defined by entries specified before the `ReadGeometryFile`. Other geometry entries (basic objects or another `ReadGeometryFile`) can follow `ReadGeometryFile`

entry, and will modify the setup accordingly.

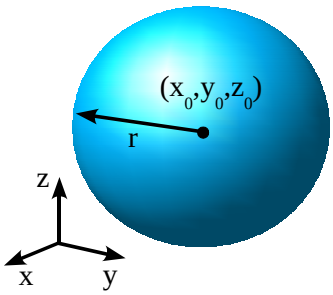
7.2 Basic Geometric primitives

Each geometric primitive is associated with a particular material via user defined material labels as they appear in the material definition file, e.g. the following adds a layer, uniform in x and y, of aluminum, 50nm in thickness, from $z = 10nm$ to $z = 60nm$ to the default background of `Vacuum`:

```
AddLayer
   material    [string]  Aluminum
   z_min       [micron]  10e-3
   z_max       [micron]  60e-3
```

The material `Aluminum` is assumed to be defined in the material definition file. The values z_{min}, z_{max} are arbitrary and don't have to be inside the computational domain boundaries. Values that lie outside the bounds of the computational domain are truncated to the edges of the computational domain. Each geometric primitive overwrites any pre-existing definition at the points where it is defined, so if two layers (or any other objects) physically overlap in space, the one defined latest in the file will take precedence at the points of overlap. To change the background material of the entire computational domain a layer of the desired material can be added with z_{min}, z_{max} of the layer correspondingly less than and greater than the computational domain boundaries z_{min}, z_{max} (Figure 1).

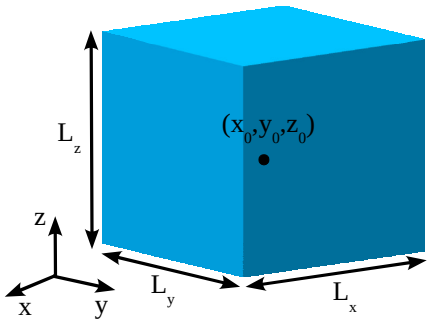
7.2.1 Sphere




```
AddSphere
  material    [string]   Gold
  center      [micron]   0.0 0.0 0.0
  radius      [micron]   1.7
```

Figure 15: Definition of parameters for sphere: center x_0, y_0, z_0, radius r.

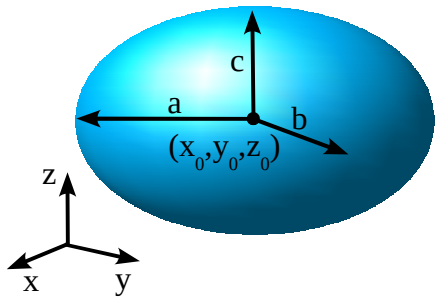
7.2.2 Cube




```
AddCube
  material    [string]   Aluminum
  center      [micron]   0.0 0.0 0.0
  Lx,Ly,Lz    [micron]   1.0 1.5 0.8
```

Figure 16: Definition of parameters for cube: center x_0, y_0, z_0, size L_x, L_y, L_z.

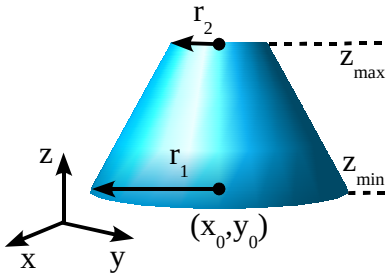
7.2.3 Ellipsoid




```
AddEllipsoid
  material    [string]   Cobalt
  center      [micron]   0.0 0.0 0.0
  a,b,c       [micron]   1.0 1.5 0.8
```

Figure 17: Definition of parameters for ellipsoid: center x_0, y_0, z_0, semi-major axis a, b, c.

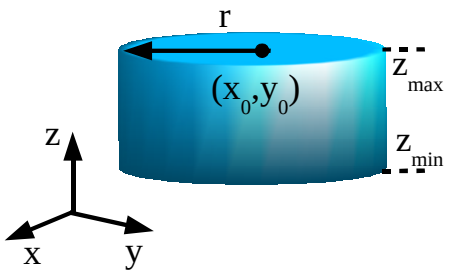
7.2.4 Cone




```
AddCone
  material    [string]   AlAs
  x,z-center  [micron]   0.0 0.0
  r1,r2       [micron]   0.2 0.4
  y_min/max   [micron]  -0.5 0.5
```

Figure 18: Definition of parameters for cone: center x_0, y_0, radii r_1, r_2, thickness z_{min}, z_{max}.

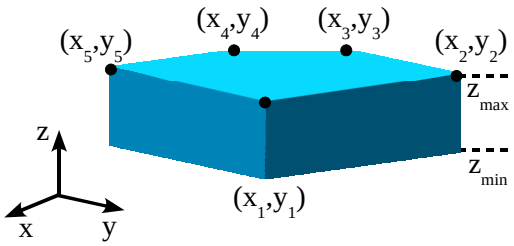
7.2.5 Disk




```
AddDisk
  material    [string]   Vacuum
  x,y-center  [micron]   0.0 0.0
  radius      [micron]   1.0
  z_min/max   [micron]   -450e-3 -70e-3
```

Figure 19: Definition of parameters for disk (cylinder): center x_0, y_0, radius r, thickness z_{min}, z_{max}. Alias AddCylinder can be used instead of AddDisk.

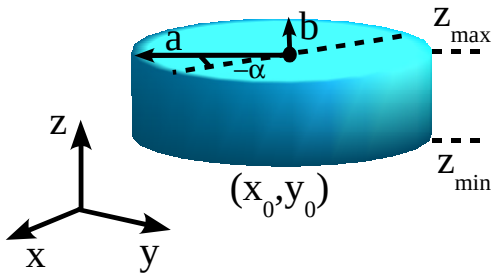
7.2.6 Polygon




```
AddConvexPolygon
  material    [string]   AlOx
  vertices    [number]   5
  x1,y1       [micron]  -0.25 -0.5
  x2,y2       [micron]   0.25 -0.5
  x3,y3       [micron]   0.5   0.0
  x4,y4       [micron]   0.25  0.5
  x5,y5       [micron]  -0.25  0.5
  z_min/max   [micron]  -0.25 0.25
```

Figure 20: Definition of parameters for convex polygon: number of vertices, list of vertex coordinates x_i, y_i, thickness z_{min}, z_{max}.

7.2.7 Ellips




```
AddEllips
  material    [string]   Silver
  x,z-center  [micron]   0.0 0.0
  a,b         [micron]   1.0 1.5
  angle       [degrees]  45.0
  y_min/max   [micron]   -450e-3 -70e-3
```

Figure 21: Definition of parameters for ellipse: center x_0, y_0, semi-major axis a, b, angle of rotation $-\alpha$ ($\alpha > 0$), thickness z_{min}, z_{max}.

Definitions for other geometric objects:

```
AddCappedRectangle
  material   [string]   SiO2
  y,z-center [micron]   0.0 0.0
  radius     [micron]   1.0
  length     [micron]   0.5
  x_min/max  [micron]  -450e-3 -70e-3
```

```
AddQuadrilateral
  material  [string]    GaAs
  x1,z1     [micron]   -0.5  -0.5
  x2,z2     [micron]   -0.25 -0.5
  x3,z3     [micron]    0.0  0.0
  x4,z4     [micron]   -0.25 0.0
  y_min/max [micron]   -0.25 0.25
```

```
AddTriangle
  material   [string]   Vacuum
  x1,y1      [micron]  -1.0 -1.0
  x2,y2      [micron]   1.0 -1.0
  x3,y3      [micron]   0.0 1.0
  z_min/max  [micron]  -450e-3 -70e-3
```

The lengths, sizes and coordinates in three-space dimensions are always specified in the order x,y,z. The lengths, sizes and coordinates in the plane are always specified in the order x, y, or x, z, or y, z. For all objects specified with the `min/max` entries, the minimum and maximum coordinate values determine the thickness of the object (`max-min`). These objects, (such as a layer, cone, triangle, etc) can be specified to have an extent (thickness) along one of the axis, e.g. x, y or z. Corresponding axis labels for other parameters must be specified accordingly. For example if the `Cylinder` object is set to have a `y_min/max` extent, then its center position should be set to be in the XZ-plane by using `x,z-center`.

For `AddEllipsoid` and `AddEllips`, the `a,b` and `c` parameters are semi-major axis values. The `angle` option for the `AddEllips` object specifies the angle of rotation (counter-clockwise for positive values) of the ellipse with respect to the positive direction of the x-axis. The `AddCappedRectangle`

type represents an object of finite thickness and the same shape as the XY cross-section shown in Figure 25b, with `radius`$= w/2$.

Triangular objects are specified by three points in the XY plane. The quadrilateral and polygon objects are specified by the coordinates of their vertices in a plane, listed in clockwise, or counter-clockwise order.

7.2.8 Lattice

A two-dimensional lattice of rectangular, circular, elliptic and triangular rods of finite thickness can be added using `AddLattice` option. The following example creates a honeycomb lattice of Aluminum rodes with elliptic cross-sections, in xy plane, and with rod lengths along the z-axis from -70nm to -20nm:

```
AddLattice
  material          [string]      Aluminum
  plane             [XY]
  lattice-type      [honeycomb]
  lattice-constant  [micron]      0.85
  lattice-center    [micron]      0.0 0.0
  M,N               [number]      11 11
  vertical_min/max  [micron]     -70e-3 -20e-3
  unit-element      [ellips]
    a,b             [micron]      0.2 0.15
  angle             [degrees]     0.0
```

The `plane` option can take values `[XY]`, `[XZ]`, `[YZ]`, to set the plane of the lattice. Then `vertical_min/max` extent option will then apply, correspondingly in z,y and x axis. The `lattice-type` can be of types `[square]`, `[triangular]` and `[honeycomb]`, with a corresponding `lattice-constant`, Figure 22. The lattice is centered at `lattice-center` in the specified `plane` and consists of `M` by `N` unit elements. `unit-element` can be a `[circle]`, `[ellips]`, `[rectangle]` or equilateral `[triangle]`. The next lines specify the properties of the unit element: radius, semi-major axes, or side lengths: Example definitions for unit element:

```
unit-element      [circle]
  radius          [micron]      0.15
unit-element      [ellips]
  a,b             [micron]      0.15 0.2
unit-element      [rectangle]
  Lx,Ly           [micron]      0.3 0.2
unit-element      [triangle]
   side           [micron]      0.3
```

The last line is optional. It specifies rotation angle of the lattice in the

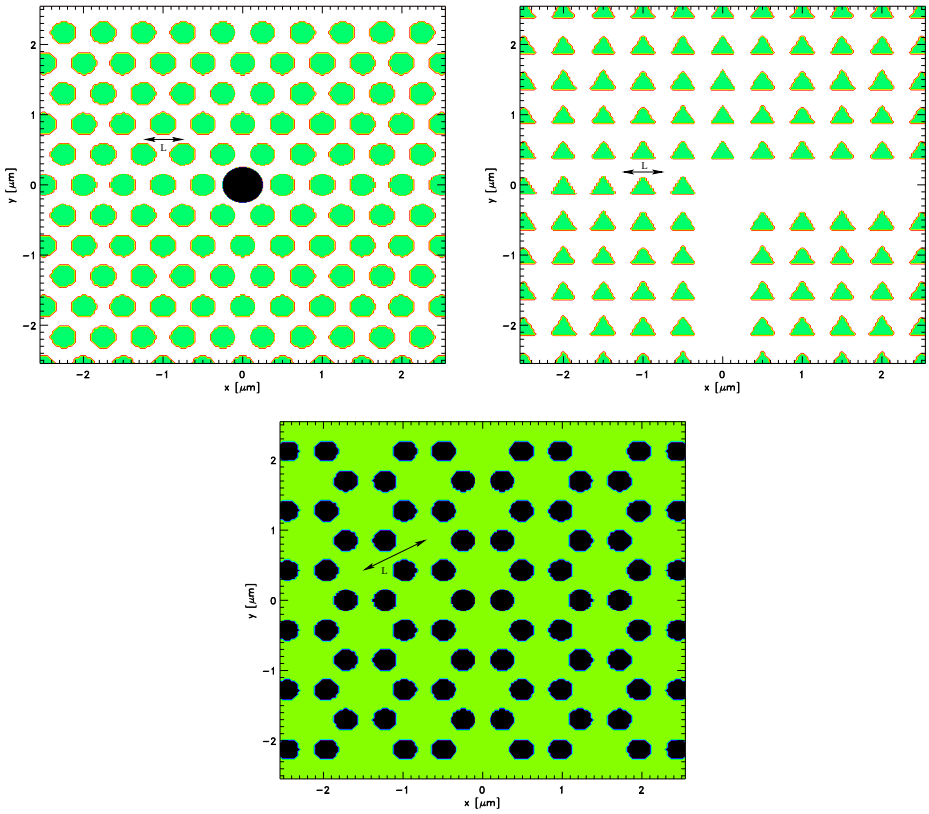


Figure 22: Triangular lattice of circular rods with a point defect. Rectangular lattice of triangular rods with line defects. Honeycomb lattice of circular holes.

plane selected by the `plane` entry.

7.2.9 Pattern

A periodic pattern that has a unit element consisting of a number of other geometric objects, can be setup using an `AddPattern` object:

```
AddPattern
  pattern-center [micron] 0.0 0.0 0.0
  pattern-size   [micron] 1.4 1.4 0.5
  M,N,P          [number] 4 4 1
  ConvexPolygon
    material  [string] GaAs
    vertices  [number] 8
    x1,y1     [micron] -0.25 -0.5
    x2,y2     [micron]  0.25 -0.5
    x3,y3     [micron]  0.5  -0.25
    x4,y4     [micron]  0.5   0.25
    x5,y5     [micron]  0.25  0.5
    x6,y6     [micron] -0.25  0.5
    x7,y7     [micron] -0.5   0.25
    x8,y8     [micron] -0.5  -0.25
    z_min/max [micron] -0.25 0.25
  ConvexPolygon
    material  [string] GaAs
    vertices  [number] 4
    x1,y1     [micron] -0.25 -0.75
    x2,y2     [micron]  0.25 -0.75
    x3,y3     [micron]  0.25  0.75
    x4,y4     [micron] -0.25  0.75
    z_min/max [micron] -0.25 0.125
  ConvexPolygon
    material  [string] Vacuum
    vertices  [number] 8
    x1,y1     [micron] -0.125 -0.25
    x2,y2     [micron]  0.125 -0.25
    x3,y3     [micron]  0.25  -0.125
    x4,y4     [micron]  0.25   0.125
    x5,y5     [micron]  0.125  0.25
    x6,y6     [micron] -0.125  0.25
    x7,y7     [micron] -0.25   0.125
    x8,y8     [micron] -0.25  -0.125
```

```
    z_min/max [micron] -0.125 0.125
  ConvexPolygon
    material  [string] Vacuum
    vertices  [number] 4
    x1,y1     [micron] -0.125 -0.75
    x2,y2     [micron]  0.125 -0.75
    x3,y3     [micron]  0.125  0.75
    x4,y4     [micron] -0.125  0.75
    z_min/max [micron] -0.125 0.125
  Disk
    material   [string] GaAs
    x,y-center [micron]  0.7 0.7
    radius     [micron]  0.15
    z_min/max  [micron] -0.125 0.125
```

This object allows a set of other basic geometric objects (four convex poly-

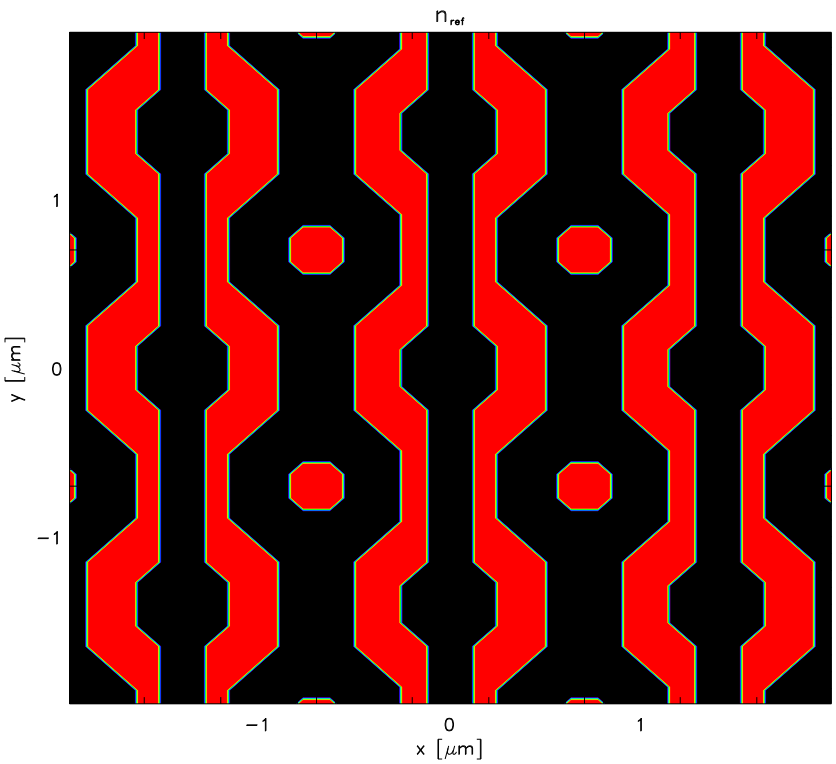


Figure 23: XY-cross-section of an example periodic pattern from the text, setup with four ConvexPolygons and one Cylinder object.

gons and a disk in the above example) to be repeated in space on an M by N by P lattice. The `unit-center` specifies the position of one of the unit elements of the pattern, and `unit-size` sets the size of the unit element along the x, y and z-axis. The parameters of the geometric objects that constitute a unit element are the same as those used when specifying a single object by itself, but the name of the object is used without the `Add` prefix. The pattern objects can be nested, as in the following example:

```
AddPattern
  unit-center [micron] 0.0 0.0 0.0
  unit-size   [micron] 1.0 1.0 0.8
  M,N,P       [number] 3 3 1
  Cone
    material   [string] GaAs
    x,y-center [micron]  0.0 0.0
    r1,r2      [micron]  0.4 0.3
    z_min/max  [micron] -0.6 0.6
    Pattern
      unit-center [micron] 0.15 0.15 0.0
      unit-size   [micron] 0.3 0.3 0.3
      M,N,P       [number] 2 2 2
      Disk
        material   [string] SiO2
        x,y-center [micron]  0.0 0.0
        radius     [micron]  0.1
        z_min/max  [micron] -0.1 0.1
```

7.3 Geometric objects for optical data storage media modeling

7.3.1 Bumps/pits

A bump (or a pit) can be added to a layer. Three types of bump objects are defined: [sphero-cylindrical-cap] (corresponds to the type "Round" defined in DIFFRACT^{TM} software), [sphero-cylindrical-stadium], and [elliptical-stadium] (corresponds to the type "Flat" defined in DIFFRACT^{TM} software).

For all bump types `material` specifies the material of the layer on which bump is put, while `substrate` specifies the material under the bump. The

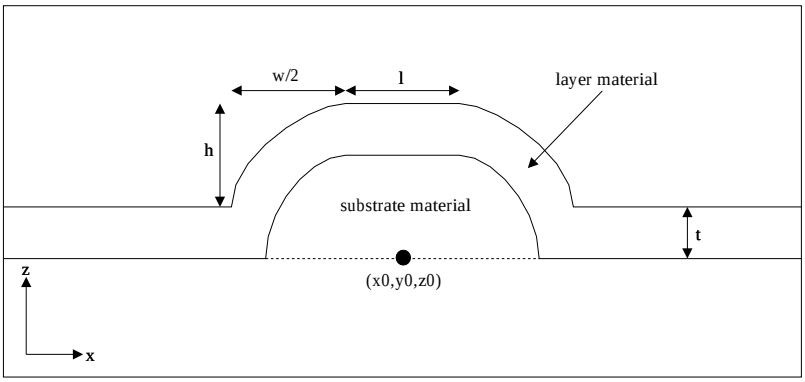


Figure 24: Definitions of parameters for bump type `[sphero-cylindrical-cap]`. The edges of the bump are defined by two spherical shells matched to a cylindrical shell of length l with the cylinder axis directed along the x-axis. The shells have a constant thickness t when measured along the z-axis.

bump position is specified by a central point x_0, y_0 of the bump in the x, y plane, and the coordinate z_0 of the bottom of the layer on which the bump is put (z_0 of the bump is equal to the z_{min} of the layer). A pit can be set up by specifying a negative value for the `height`. For the pit, z_0 still signifies z_{min} of the layer in which the pit is made, while `substrate` specifies the material inside the pit.

The width w, height h, length l, and layer thickness t for the `[sphero-cylindrical-cap]` type are defined on Figure 24. The length is applied only in x coordinate, so the bump (pit) is elongated only along x. If the length is zero, the bump is circular in the xy plane. Example:

```
AddBump
   bump_type  [sphero-cylindrical-cap]
   material   [string]   Aluminum
   substrate  [string]   Vacuum
   x_0        [micron]   0.0
   y_0        [micron]   0.0
   z_0        [micron]  -140e-3
   width      [micron]   400e-3
   height     [micron]   60e-3
   length     [micron]   200e-3
   thickness  [micron]   50e-3
```

For the bump types `[sphero-cylindrical-stadium]`,

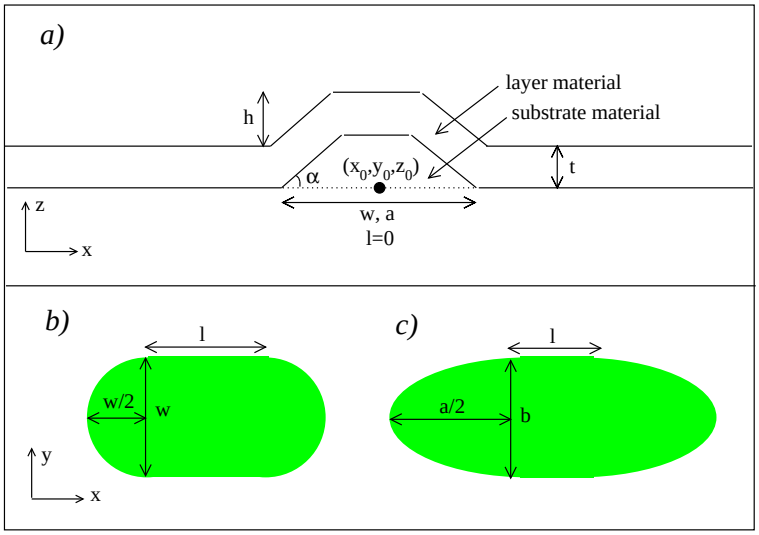


Figure 25: Definitions of parameters for the bump types `[sphero-cylindrical-stadium]` and `[elliptical-stadium]`. Angle α determines the steepness of the walls which have a constant thickness t when measured along the local unit vector normal to the surface. a) XZ cross-section is shown for the case of $l = 0$ (zero elongation), b) XY cross-section of the sphero-cylindrical bump and c) XY cross-section of the elliptical bump.

`[elliptical-stadium]` the parameter `wall_angle` specifies angle α defined in Figure 25. For the bump type `[elliptical-stadium]`, instead of a `width` parameter, the major axis `a,b` of the elliptical cross-section of the bump are specified, see Fig. 25. Examples of the "stadium" type bump/pit setup:

```
AddBump
  bump_type  [sphero-cylindrical-stadium]
  material   [string]   Aluminum
  substrate  [string]   SiO2
  x_0        [micron]   1.5
  y_0        [micron]   1.5
  z_0        [micron]  -70e-3
  width      [micron]   400e-3
  height     [micron]   60e-3
  length     [micron]   500e-3
  thickness  [micron]   50e-3
  wall_angle [degrees]  60.0
```

Any of the bump types can have an optional last line specifying the an-

gle (measured counter-clockwise from the x-axis) of the bump rotation in the XY-plane: `angle [degrees] 60`. The default value for this rotation angle is zero.

```
AddBump
  bump_type  [elliptical-stadium]
  material   [string]   Aluminum
  substrate  [string]   SiO2
  x_0        [micron]  -1.5
  y_0        [micron]  -1.5
  z_0        [micron]  -70e-3
  a,b        [micron]   500e-3 750e-3
  height     [micron]  -65e-3
  length     [micron]   0.5
  thickness  [micron]   50e-3
  wall_angle [degrees]  45.0
  angle      [degrees]  60.0
```

The bump types [sphero-cylindrical-stadium] and [elliptical-stadium] have a constant thickness t of the layer, when the thickness is measured along the local normal to the surface, whereas [sphero-cylindrical-cap] bumps have constant thickness of the layer, when the thickness is measured along the z-axis.

7.3.2 Grooves

The `AddGrooves` option allows user to set up grooves with a trapezoidal shape. In the following example, first a layer of `SiO2` is created and then grooves are added to it:

```
AddLayer
  material    [string]   SiO2
  z_min       [micron]  -250e-3
  z_max       [micron]  -70e-3

AddGrooves
  substrate      [string]   SiO2
  A,B,C,D,zeta   [micron]   300e-3 800e-3 1100e-3 1800e-3 70e-3
  angle          [degrees]  60.0
  x0,y0          [micron]   0e-3 0e-3
```

where `substrate` specifies material of the layer in which grooves are to

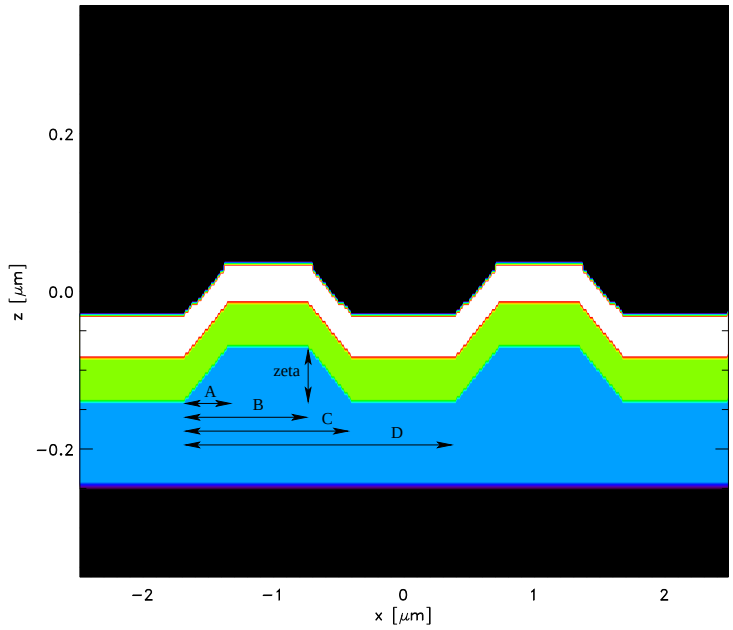


Figure 26: XZ-plane cross-section of the multilayer grooved stack with definitions of the input parameters for the groove geometry.

be made, `A,B,C,D,zeta` set groove parameters (Figure 26) and `angle` specifies angle (measured counter-clockwise from the x-axis) of the grooves in the

XY-plane. The `x0,y0` specify the shift of the position of the center-line of a groove with respect to the center of the computational domain. This can be used, for example, to direct either groove, or land, or edge through the center of the computational domain.

In the above example, if there is more than one `SiO2` layer already setup, the grooves are applied to the layer with largest `z_max`.

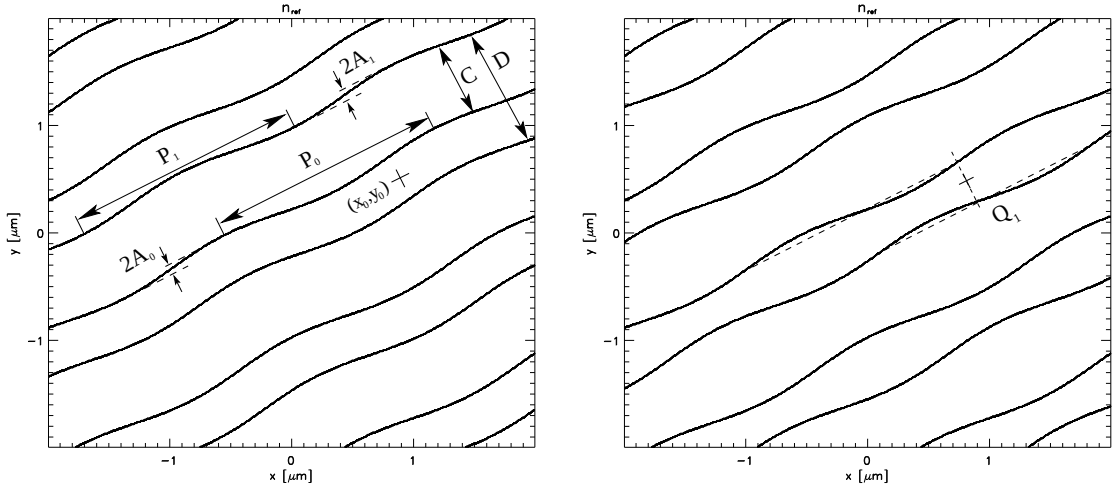


Figure 27: Wobbled grooves on an Optical Disk Surface, using parameters from the example in the text with $Q_1 = 0°$ (left), and $Q_1 = 180°$ (right).

Groove-width modulation, groove radial-position modulation and similar effects can be modeled using the `AddWobbledGrooves` geometry object:

```
AddWobbledGrooves
  substrate       [string]   Aluminum
  A,B,C,D,zeta    [micron]   100e-3 660e-3 760e-3 1100e-3 80e-3
  angle           [degrees]  30.0
  x0,y0           [micron]   860e-3 500e-3
  A0,P0           [micron]   50e-3 2000e-3
  A1,P1           [micron]   50e-3 2000e-3
  Q1              [degrees]  0.0
```

The `AddWobbledGrooves` object specifies grooved structure in the same way as the `AddGrooves` object. The additional parameters, `A0,P0` and `A1,P1, Q1`, set the wobble amplitude, period and relative phase of the opposite groove edges, as shown in Figure 27. The groove-edges have variation of the form $A_0 \sin(2\pi x/P_0)$ and $A_1 \sin(2\pi x/P_1 + Q_1)$. The groove-edge variation defined

by the A_0, P_0 parameters has 0° phase with respect to the (x_0, y_0) point.

7.3.3 Conformal layer

Once some structures are specified in the computational domain, a conformal layer can be added on top of the existing structures using `AddConformal Layer` definition:

```
AddConformalLayer
  material          [string]      Aluminum
  add-on-top-of     [string]      SiO2
  thickness         [micron]      50e-3
```

where in this example, a layer of `Aluminum` 50nm thick, is added on top of the structures made of `SiO2`. If there are `SiO2` structures at more than one z-coordinate (for example two layers of `SiO2`, separated by some other material), then the conformal layer will be added on top of the `SiO2` layer with largest `z_max`.

The following sequence of structure definitions will produce a multilayer-grooved stack shown in Figure 26:

```
AddLayer
  material    [string]   SiO2
  z_min       [micron]  -250e-3
  z_max       [micron]  -70e-3

AddGrooves
  substrate      [string]  SiO2
  A,B,C,D,zeta   [micron]  300e-3 800e-3 1100e-3 1800e-3 70e-3
  angle          [degrees] 60.0
  x0,y0          [micron]  0e-3 0e-3
```

```
AddConformalLayer
  material         [string]  Aluminum
  add-on-top-of    [string]  SiO2
  thickness        [micron]  50e-3

AddConformalLayer
  material         [string]  Gold
  add-on-top-of    [string]  Aluminum
  thickness        [micron]  50e-3
```

7.3.4 Sine-layer

The `AddSinLayer` option allows user to set up a sinusoidally modulated layer. The `direction` entry can take values `[X],[Y]` or `[Z]` and sets the axis, w $(= x, y, z)$, along which $h \sin(2\pi w/p)$ variation is applied.

```
AddSinLayer
  material   [string]   Gold
  direction  [Z]
  x0,y0,z0   [micron]   0.0 50e-3 1000e-3
  pitch,height,thickness [micron] 0.36 50e-3 50e-3
```

The `pitch` of the variation is p, the amplitude of the sine, h, corresponds to the `height` parameter, and the thickness of the layer is given by the `thickness` entry.
Depending on the `direction`, the `x0,y0,z0` are used as follows:

When $w = x$, the sine runs along x, with layer modulated in z, uniform along y, and the layer has one of its minima in the XZ-plane at `x0,z0`.

When $w = y$, the sine runs along y, with layer modulated in z, uniform along x, and the layer has one of its minima in the YZ-plane at `y0,z0`.

When $w = z$, the sine runs along z, with layer modulated in y, uniform along x, and the layer has one of its minima in the YZ-plane at `y0,z0`.

7.4 Dielectric Material Interfaces

The interfaces between different dielectric media by default are treated as discontinuous, step-function transitions of the permittivity, e.g. from ϵ_1 to ϵ_2. An entry in the geometry definition file,

```
AverageDielectricInterfaces
```

can be used to create a distribution of ϵ in which the permittivity at the interfaces between two dielectric materials is replaced by $\bar{\epsilon} = (\epsilon_1 + \epsilon_2)/2$. The averaging applies to all dielectric material-interfaces found in the computational domain, separately along each of the coordinate directions, and is valid only for materials with real-valued ϵ.

8 Comments in the input files

In the material and geometry input files C-style comments /* */ (but no nested comments) can be used to comment out one or more material or geometry definition blocks. For example the following blocks,

```
/*
AddConvexPolygon
  material  [string] SiO2
  vertices  [number] 8
  x1,y1     [micron] -0.25 -0.5
  x2,y2     [micron]  0.25 -0.5
  x3,y3     [micron]  0.4  -0.2
  x4,y4     [micron]  0.4   0.2
  x4,y4     [micron]  0.25  0.5
  x5,y5     [micron] -0.25  0.5
  x7,y7     [micron] -0.4   0.2
  x8,y8     [micron] -0.4  -0.2
  z_min/max [micron] -0.25  0.25

AddLayer
  material   [string]   SiO2
  y_min      [micron]   750e-3
```

```
  y_max       [micron]   1000e-3
*/
```

are commented out, and will have no effect on the geometry setup. Successive commented out blocks (/* ... */ /* ... */) must be separated by at least one space or newline character.

9 Application Examples

This section works through simple validation cases and example input files set to simulate scattering of a laser beam from a mark similar to those found on the optical disk surface; and an imaging problem with a partially coherent light source. References [4]-[8] and articles listed in Appendix E illustrate more application examples: dependence of the reflected signal on beam-center position with respect to the sub-wavelength marks; modeling of push-pull tracking signal from a grooved optical disk surface; and light transmission through small elliptical apertures in a thin metal film.

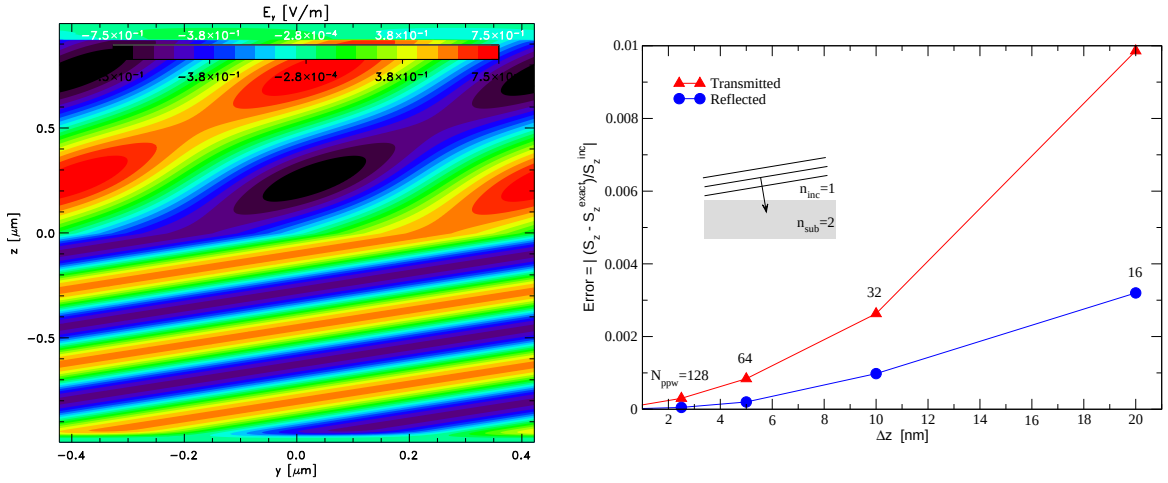


Figure 28: Rate of convergence for a problem of a planewave scattering from a a dielectric interface. The error in the numerical solution decreases as $O(\Delta z^2)$ with increasing number of points per wavelength, $N_{ppw} = \lambda_0/(n_{sub}\Delta z)$.

9.1 Order of convergence

Figure 28 shows a time-snapshot of the distribution of the y-component of the **E** field and the computed error as a function of the grid step-size for

a reflection/transmission problem. A TE_x (E_y, E_z, H_x) polarized planewave with $\lambda_0 = 650nm$ is incident at an angle of 50° on an air/glass interface. The Poynting vector is computed along the z-axis at $y = 0$ with a monitor at source frequency. The difference between computed and exact S_z normalized by the incident energy flux, indicates second-order $O(\Delta z^2)$ convergence of the numerical solution to the exact solution (R=0.0268, T=0.9732). In the problems that have material interfaces not aligned along the grid lines, the staircased approximation of the curved material interfaces on the finite-difference grid in general will reduce the order of convergence to the exact solution to $O(\Delta z)$.

9.2 Reflection from a bi-layer

In this example we compute reflection of a planewave and of a laser beam from a two-layer stack embedded in a medium with refractive index n=1.55. The stack consists of materials, and has layer thicknesses, similar to those commonly used in optical data storage media: a layer of $ZnS - SiO_2$, extending from $z_{max} = 50nm$ to $z_{min} = 0nm$, followed by a 50nm layer of GST from $z_{max} = 0nm$ to $z_{min} = -50nm$. In the case of a planewave source, the light has a unit elictic field amplitude, free-space wavelength of $\lambda_0 = 405nm$, and is incident normally onto the surface of the stack. The computational grid has $\Delta z = 5nm$ in the region occupied by the two layers, and $\Delta z = 10nm$ elsewhere. Two sets of material refractive index values corresponding to temperatures of $T = 300K$ and $T = 400K$ are used, and Debye material model is employed to represent the GST layer. The exact and computed reflection and transmission characteristics are shown in Table 3. Both amplitude coefficients R and T and phase difference $\Delta\phi = \phi_{300} - \phi_{400}$ of the reflected and transmitted waves converge to the exact solution when grid cell size Δz is reduced by a factor of two throughout the domain. Only the phase differences, and not absolute value of the phase, can be used for comparison to the analytic results, since the phases computed in FDTD have initial phase shifts due to start time of the the source and offset time of the field sampling for Fourier Transform.

For the same materials, geometry, wavelength, and FDTD grid parameters, we also compute reflected fields for the case of an incident laser beam. In DIFFRACT a circularly polarized Gaussian beam is brought to focus in

Table 3: Comparison of exact and numerical solutions for a planewave incident on a bi-layer in a medium with refractive index $n = 1.55$, layer thicknesses $l_{ZnS-SiO_2} = l_{GST} = 50nm$ and layer refractive indecies $n_{ZnS-SiO_2}(300K) = 2.32$, $n_{GST}(300K) = 1.753 + 3.248i$ at $T = 300K$, and $n_{ZnS-SiO_2}(400K) = 2.1$, $n_{GST}(400K) = 1.3 + 4i$ at $T = 400K$.

	R_{300}	T_{300}	R_{400}	T_{400}	$\Delta\phi_R$	$\Delta\phi_T$
Exact	0.655	0.0913	0.751	0.0553	37.4°	41.7°
FDTD, $\Delta z_{min} = 2.5nm$	0.657	0.0915	0.752	0.0560	37.3°	42.0°
FDTD, $\Delta z_{min} = 5.0nm$	0.660	0.0921	0.756	0.0565	37.0°	41.0°

air by a focal lens with $NA = 0.85$ and focal length $f \approx 4346\lambda_0$. After focusing, the beam is transferred into a medium with n=1.55, and exported into a file for use as a source in the FDTD simulations. The source distribution is read and placed into the FDTD grid, on an XY-plane positioned 35nm above the surface of the stack, with incident beam propagating in the negative z-direction. The reflected beam in FDTD propagates in a positive z-direction and is computed just above the source plane. To have a comparable computation performed in DIFFRACT alone, without FDTD, first the beam is propagated in DIFFRACT 35nm to the surfce of the stack, then the field reflected from the stack is computed and propagated 35nm back to where the incident focused beam started inside the n=1.55 medium.

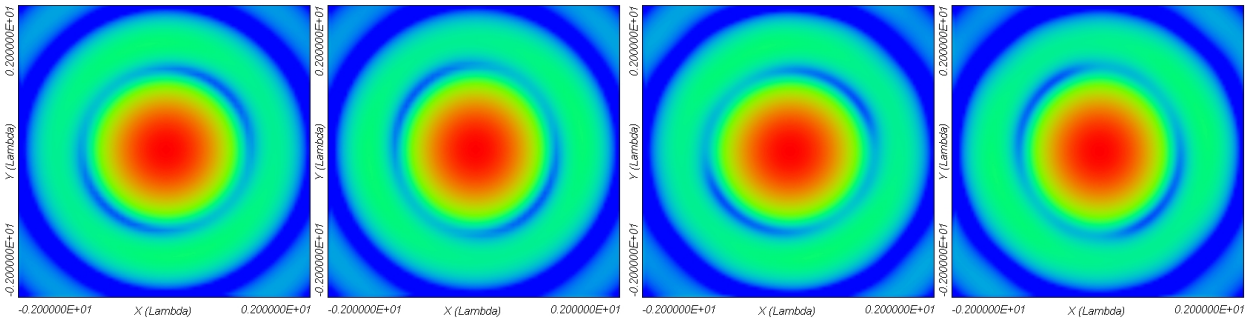


Figure 29: Left two: log_intensity_3 scale plots I_{x300} and I_{y300} of the reflected wave computed with DIFFRACT alone. Right two: I_{x300} and I_{y300} computed with DIFFRACT/FDTD.

The results of computations for material parameters corresponding to temperatures of $T = 300K$ and $T = 400K$ are shown in Table 4 for the intensities I_x and I_y of the x- and y-components of the reflected field, indicating good agreement between intensities and phase difference $\Delta\phi_R = \phi_{400} - \phi_{300}$, from computations performed with DIFFRACT/FDTD and DIFFRACT alone.

Table 4: Comparison of numerical solutions using DIFFRACT and DIFFRACT/FDTD for a focused beam incident on a bi-layer with parameters specified in the caption of Table 3

	I_{x300}	I_{y300}	I_{x400}	I_{y400}	$\Delta\phi_R$
DIFFRACT	0.1792	0.1792	0.238	0.238	39.84°
DIFFRACT/FDTD, $\Delta z_{min} = 5.0nm$	0.1803	0.1814	0.240	0.241	39.56°

9.3 Scattering of a planewave from a sphere

In this test case example we compute in three space dimensions the problem of scattering of a planewave from small dielectric and metal spheres. Figure 30 shows exact solutions computed using Mie-scattering theory, and corresponding numerical solutions. The incident planewave propagates along the negative direction of the z-axis. A uniform grid cell size of $\Delta = 10nm$

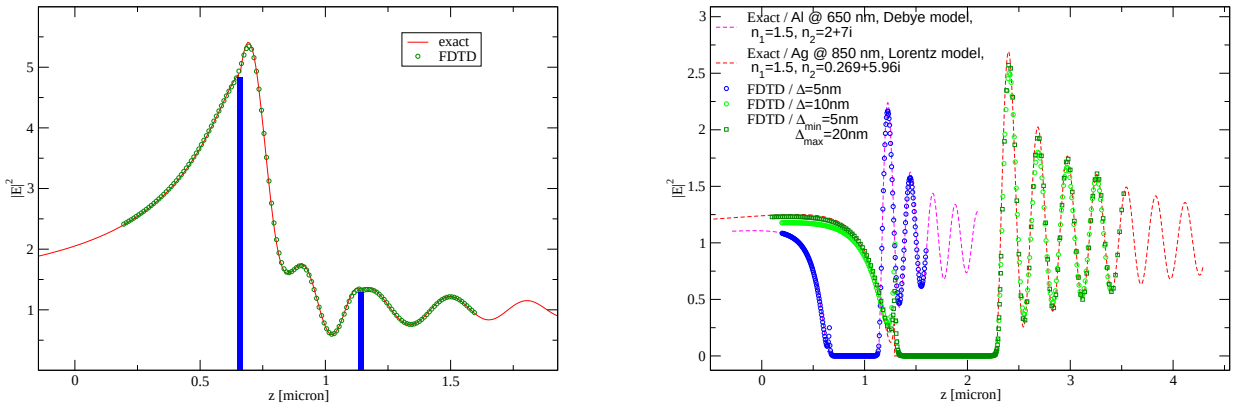


Figure 30: Comparison of exact (lines) and FDTD (symbols) solutions in terms of total electric field magnitude variation along the light incidence axis passing through the center of the sphere. Left: the dielectric sphere of radius $0.24\mu m$ and refractive index $n_2 = 1.54$ is illuminated by the planewave with wavelength $\lambda = 0.6\mu m$ in a medium with $n_1 = 1.0$. Vertical lines mark the boundaries of the dielectric sphere. Right: the metal spheres are embedded in a dielectric with $n_1 = 1.5$ and have radii $r_{Al} = 0.24\mu m$ and $r_{Ag} = 0.5\mu m$.

was used for the dielectric sphere, $\Delta = 5nm$ - for the Al sphere, and both $\Delta = 10nm$ uniform grid and $\Delta_{min} = 5nm$, $\Delta_{max} = 20nm$ non-uniform grid for the Ag sphere.

The following material model parameters were used for aluminum at $\lambda = 0.65\mu m$ and silver at $\lambda = 0.85\mu m$:

```
Material Label          [string]        Aluminum
```

```
  model               [Debye]
  tau                 [femtosec]    1.95595
  eps_inf             [relative]    1.0
  delta_eps           [relative]   -1522.91
  conductivity        [1/(ohm*m)]   7.40866e6

Material Label        [string]      Silver
  model               [Lorentz]
  omega0              [Hz]          1.63991e15
  delta               [Hz]          4.18345e13
  eps_inf             [relative]    3.0
  delta_eps           [relative]    32.0
  conductivity        [1/(ohm*m)]   0.0
```

9.4 Laser beam scattering from a mark

In this subsection we discuss input required for computation of scattering of a focused beam from a single pit formed in a $50nm$-thick layer of aluminum coated on a dielectric substrate. To simulate the focused beam distribution the Diffract source option is used in the input parameter file. The "diffract_source.dat" file, created with DIFFRACT software, contains $\mathbf{E}$ and $\mathbf{H}$ field distributions in the XY-plane, obtained by bringing to a focus a beam of light with wavelength $\lambda_0 = 650nm$. The focusing lens has a numerical aperture $NA = 0.6$ and focal length $5000\lambda_0$.

To adequately resolve the pit, a non-uniform grid is used, with resolution of $5nm$ in the z-direction and $10nm$ in the x and y directions at the position of the pit. The input parameter file:

```
PIT SIMULATION PARAMETERS

Start-stop and timestep:
   tmin      [nanoseconds]      0.0
   tmax      [nanoseconds]      20.0e-6
   delta_t   automatic-with-CFL 0.4

Non-Uniform Grid1:
      w1     [micron]     500e-3
```

```
      w2     [micron]     200e-3
      w3     [micron]     2040e-3
   delta_1   [micron]     10e-3
   delta_2   [micron]     20e-3
   delta_3   [micron]     30e-3

      h1     [micron]     160e-3
      h2     [micron]     140e-3
      h3     [micron]     300e-3
   deltaz_1  [micron]     10e-3
   deltaz_2  [micron]     5.0e-3
   deltaz_3  [micron]     10e-3

Working directory:                C:\username\Maxwell\FDTD\
Material Definition Filename:     pit_materials.input
Geometry Definition Filename:     pit_geometry.input
Boundary Conditions Filename:     boundaries.input

Material index:
  Write to file?           no
  Filename                 mindex.out
Fields:
  NumberOfOutputs   0
  WriteEx,Ey,Ez?    no no no
  WriteHx,Hy,Hz?    no no no
CheckpointFile:
  RestartFromCheckpointFile    no
  WriteCheckpointFile          no

DiffractSource:
  Wavelength [micron]      0.65
  Filename                 diffract_source.dat
  File Format              ascii
  Grid Type                staggered

ExportReflected:
```

```
  Filename                  fdtd.export.r
  File Format               ascii
  Grid Type                 collocated

ExportTransmitted:
  Filename                  fdtd.export.t
  File Format               fortran_binary
  Grid Type                 collocated
  z-location  [micron]      -10000e-3
```

In this example the output of the material index, field and checkpoint files is switched off. The desired output from the simulation is obtained with the *ExportReflected* option and represents the distribution of the reflected light sampled in the XY-plane. The *ExportTransmitted* sampling plane is also specified, but its location along the z-axis is set out of bounds of the computational domain, so it will be ignored.

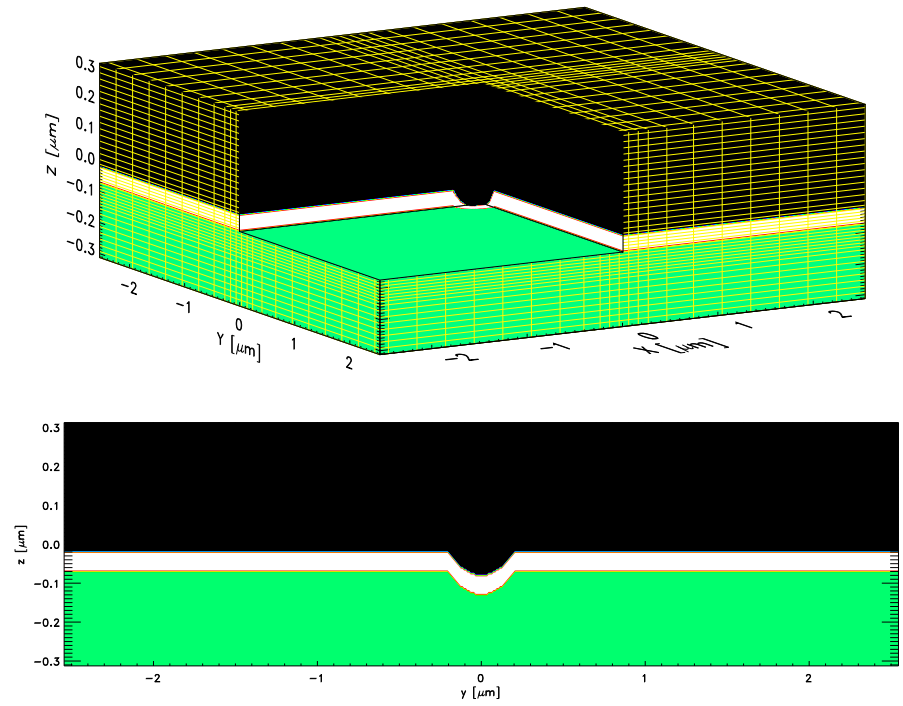


Figure 31: Computational domain with a non-uniform grid refined in the center, at the location of the pit. The sphero-cylindrical pit made in 50nm aluminum layer is 400nm wide, 600nm long and 60nm deep. The 2D plot shows a zy-cross section of the 3-D domain.

PML absorbing boundary conditions are set for all axes in the boundary

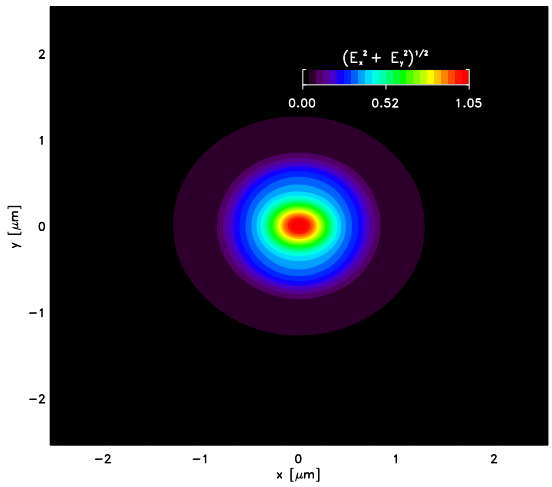


Figure 32: Transverse amplitude $\sqrt{E_x^2 + E_y^2}$ distribution of the light reflected from a pit in aluminum layer for a geometry shown in Figure 31. Scattering of the light from the walls of the pit "focuses" it toward the center of the pit.

conditions input file:

```
BoundaryConditions:
       x-axis [PML]
       nx_pml 15
  sigma,kappa 0 0
       y-axis [PML]
       ny_pml 15
  sigma,kappa 0 0
       z-axis [PML]
       nz_pml 15
  sigma,kappa 0 0
```

The input material definition file defines a transparent dielectric substrate material, called in this example `SiO2`, and material `Aluminum`, modeled using Lorentz model. Parameters for the Lorentz model are set to result in $n - ki = \sqrt{\epsilon} = 2 - 7i$ at the wavelength $\lambda_0 = 650nm$ of the incident light.

```
Material Label        [string]       SiO2
  model               [dielectric]
  refractive index    [dimensionless] 1.5
  conductivity        [1/(ohm*m)]     0.0
```

```
Material Label        [string]       Aluminum
  model               [Lorentz]
  omega0              [Hz]           23.536118e14
  delta               [Hz]           2.9504974e14
  eps_inf             [relative]     1.81
  delta_eps           [relative]     32.802
  conductivity        [1/(ohm*m)]    0.0
```

In the input geometry definition file first a substrate layer is set to extend from the bottom of the computational domain to $z = -70nm$, then a $50nm$-thick layer of aluminum is added on top of the substrate. A pit (bump with a negative height) in the aluminum layer is placed in the center of the computational domain $x_0 = y_0 = 0$.

```
AddLayer
  material    [string]    SiO2
  z_min       [micron]   -450e-3
  z_max       [micron]   -70e-3

AddLayer
  material    [string]    Aluminum
  z_min       [micron]   -70e-3
  z_max       [micron]   -20e-3

AddBump
  bump_type   [sphero-cylindrical-cap]
  material    [string]    Aluminum
  substrate   [string]    Vacuum
  x_0         [micron]    0.0
  y_0         [micron]    0.0
  z_0         [micron]   -70e-3
  width       [micron]    400e-3
  height      [micron]   -60e-3
  length      [micron]    200e-3
  thickness   [micron]    50e-3
```

Figures 32, 31 show the grid and material layout corresponding to the above example input files, and the computed reflected transverse $\mathbf{E}$ field amplitude.

The reflected light field distribution can be imported back into DIFFRACT software for further processing and propagation through various optical elements.

9.5 Imaging problem

We consider an imaging problem with the following setup of the numerical experiment: a partially coherent source with wavelength $\lambda_0 = 250nm$ illuminates a sample, and the reflected light is propagated a total distance of $5000\mu m$ to the entrance pupil of a collimating lens (focal length equal to $5.0mm$, $NA = 0.8$), then focused to the final image plane by a focusing lens having $f = 40mm$, $NA = 0.1$. The magnification M of this system is the ratio of the two focal lengths, namely, $M = 40/5 = 8$. The simulation is done in four steps:

1. Data sets representing partially coherent source are created in DIFFRACT, and stored in the files to be used as input source in FDTD computations.
2. For each of the source files, an FDTD simulation is performed to obtain the light distribution reflected from the sample, and the reflected fields are stored for import back into DIFFRACT software.
3. Each reflected field distribution is imported into DIFFRACT, and propagated through the collimating and focusing lenses to the image plane, where intensity of the light distribution is recorded.
4. The intensities from each computation in step 3 are added to obtain the total image.

The partially coherent illumination is modeled by creating in DIFFRACT uniform beams and using C_{11} and ϕ_{11} options of the beam Distortion entry to assign a tilt to the beam, via polar $\theta = \theta(C_{11})$ and azimuthal $\phi = \phi_{11}$ angles. A $5\mu m \times 5\mu m$ square mask is applied to the beam and the distribution is exported to a file. To reduce diffraction at the edges of the beam, the top-hat shape of the square beam is smoothed using shape softening option Alpha of the Mask entry. Example of DIFFRACT commands used to create square beam with $C_{11} = 5$ and $\phi_{11} = 0$:

```
$_____________________________________________________________
 Remarks:

  Vacuum wavelength (nm):    250.0000          NVIRON: 1.000000
_____________________________________________________________
```

```
    Initial distribution: BEAM               (Length_Units: um)
 Type: UB/SG/GG/LG/HG/LD: UB
                 BCX,BCY:    0.000000  0.0000
      Radius of aperture:    4.000000
             Aberrations: Seidel
           Spherical C40:    0.000000
          Coma C31,Phi31:    0.000000  0.0000
   Astigmatism C22,Phi22:    0.000000  0.0000
           Curvature C20:    0.000000
    Distortion C11,Phi11:    5.000000  0.0000
    Polarization RHO,ETA:    0.000000  0.0000
               NMAX,NMAY:    512   512
               LMAX,LMAY:    25.00000  25.000
-------------------------------------------------------------

    Amplitude/phase mask: MASK               (Length_Units: um)
                   Shape: Rectangle
                 MCX,MCY:    0.000000  0.0000
      Length,Width,Alpha:    5.000000  5.0000 0.2000000
 Orientation angle Theta:    0.000000
  Inside amplitude,phase:    1.000000  0.0000
 Outside amplitude,phase:    0.000000  0.0000
-------------------------------------------------------------

          FDTD Interface: FDTD               (Length_Units: um)
           Export/Import: Export
                  NX,NY:    256   256
                  LX,LY:    6.000000  6.0000
                 DeltaZ:   -0.004000
    Staggered mesh (Y/N): N
                Filename: SP01.DAT
         ASCII or Binary: Binary
$_____________________________________________________________
```

In the computations described below, the following sampling of angles was used:

$$C_{11}^0 = 0.0, \quad \phi_{11}^0 = 0.0,$$

plus twelve pairs of $C_{11}^i,\ \phi_{11}^{ij}$:

$$C_{11}^i = 5.0, 3.5, 1.5,\ \ i = 1, 2, 3; \quad \phi_{11}^{ij} = \phi^j + (i-1) \times 30^\circ,\ \ j = 1, 2, 3, 4;$$

with $\phi^j = 0^\circ, 90^\circ, 180^\circ, 270^\circ$.

Each of the input source files is used in the Diffract source option of the FDTD input parameter file setup to compute the reflected fields:

```
-------------------------------------------------------------

FDTD INPUT VALUES

Start-stop and timestep:
                          tmin         [nanoseconds] 0
                          tmax         [nanoseconds] 30E-006
                       delta_t   automatic-with-CFL 0.4
                 Uniform Grid:
                            nx                [cells] 800
                            ny                [cells] 800
                            nz                [cells] 250
                          xmin              [micron] -2.8
                          xmax              [micron]  2.8
                          ymin              [micron] -2.8
                          ymax              [micron]  2.8
                          zmin              [micron] -0.6
                          zmax              [micron]  0.4
            Working directory: imag/
 Material Definition Filename: material.input
 Geometry Definition Filename: geometry.input
 Boundary Conditions Filename: boundaries.input
Material index:
                Write to file? No
                      Filename mindex.out
Fields:
               NumberOfOutputs 0
                WriteEx,Ey,Ez? no no no
                WriteHx,Hy,Hz? no no no
CheckpointFile:
```

```
     RestartFromCheckpointFile no
           WriteCheckpointFile no

DiffractSource:
                     Wavelength [micron] 0.25
                       Filename SP01.DAT
                    File Format fortran_binary
                      Grid Type collocated

ExportReflected:
                       Filename sr01.dat
                    File Format fortran_binary
                      Grid Type collocated
                          NX,NY 256 256
ExportTransmitted:
                       Filename st01.dat
                    File Format fortran_binary
                      Grid Type collocated
                     x-location             [micron] 10e3
                          NY,NZ 256 100
-------------------------------------------------------------
```

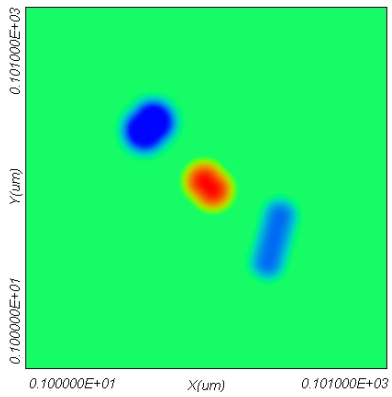


Figure 33: Imaged sample comprised of three randomly placed marks. Center: round [sphero-cylindrical-cap] bump with length=1000nm, width=800nm, height=60nm. Top-left: flat [elliptical-stadium] pit with length=1200nm, width=1000nm, depth=60nm. Bottom-right: round [sphero-cylindrical-cap] pit with length=1500nm, width=700nm, depth=60nm.

The imaged sample consists of three marks (pits and bumps) placed arbitrarily in a layer of material with high reflection coefficient, ($n_1 = n - ki =$

$0.4 - 4.5i$, $r = |(1 - n_1)/(1 + n_1)| = 0.9633$) similar to the Optical Disk surface. The input geometry (Figure 33):

```
-------------------------------------------------------------

AddLayer
  material    [string]    Aluminum
  z_min       [micron]   -260e-3
  z_max       [micron]   -100e-3

AddBump
  bump_type   [sphero-cylindrical-cap]
  material    [string]    Aluminum
  substrate   [string]    Vacuum
  x_0         [micron]    0.0
  y_0         [micron]    0.0
  z_0         [micron]   -260e-3
  width       [micron]    800e-3
  height      [micron]    60e-3
  length      [micron]    200e-3
  thickness   [micron]    160e-3
  angle       [degrees]  -45

AddBump
  bump_type   [elliptical-stadium]
  material    [string]    Aluminum
  substrate   [string]    Vacuum
  x_0         [micron]   -1.0
  y_0         [micron]    1.0
  z_0         [micron]   -260e-3
  a,b         [micron]    1200e-3 1000e-3
  height      [micron]   -60e-3
  length      [micron]    0e-3
  thickness   [micron]    160e-3
  wall_angle  [degrees]   60
  angle       [degrees]   45
```

```
AddBump
  bump_type  [sphero-cylindrical-cap]
  material   [string]   Aluminum
  substrate  [string]   Vacuum
  x_0        [micron]   1.1
  y_0        [micron]  -0.9
  z_0        [micron]  -260e-3
  width      [micron]   700e-3
  height     [micron]  -50e-3
  length     [micron]   800e-3
  thickness  [micron]   160e-3
  angle      [degrees]  75
-------------------------------------------------------------
```

Absorbing boundary conditions are applied along the x, y and z-axis. The Diffract source is applied near the top of the computational domain, at $z = 0.33\mu m$. The reflected field is sampled just above the source at $z = 0.338\mu m$ and the top surface of the layer in which marks are made is at $z = -0.1\mu m$. Hence the reflected beam propagates $0.438\mu m$ inside FDTD grid before being saved to a file for later import into DIFFRACT.

After reflected fields are computed in FDTD for all incident beams, they can be imported into DIFFRACT and propagated to the image plane:

```
$_____________________________________________________________
 Remarks:

  Vacuum wavelength (um):    1.000000          NVIRON: 1.000000
-------------------------------------------------------------

          FDTD Interface: FDTD                (Length_Units: um)
           Export/Import: Import
                Filename: sr01.dat
-------------------------------------------------------------

    Amplitude/phase mask: MASK                (Length_Units: um)
                   Shape: Rectangle
                 MCX,MCY:    0.000000  0.0000
```

```
      Length,Width,Alpha:    5.000000  5.0000  0.000000
 Orientation angle Theta:    0.000000
  Inside amplitude,phase:    1.000000  0.0000
 Outside amplitude,phase:    0.000000  0.0000
-------------------------------------------------------------

          Spatial Filter: FLTR               (Length_Units: um)
      Computation Method: DFT
             CSX,CSY,S0:    0.000000  0.0000  1.000000
        Inside (A0,Phi0):    1.000000  0.0000
       Outside (A1,Phi1):    0.000000  0.0000
      New Mesh NMAX,NMAY:    512   512
      New Mesh LMAX,LMAY:    25.000000 25.000
-------------------------------------------------------------

Propagate in environment: PROP               (Length_Units: um)
    Propagation distance:    4999.562
Multiply curvature (Y/N): N
   Reposition beam (Y/N): N
      Propagation regime: FRNHF
                    Smax:   0.010000
     Scalar/Quasi-vector: SC
-------------------------------------------------------------

                    Lens: LENS               (Length_Units: um)
                    Type: COLL
                  LCX,LCY:    0.000000  0.0000
                    NA,FL:    0.800000  5000.0
              Aberrations: None
     Scalar/Quasi-vector: QV
-------------------------------------------------------------

                    Lens: LENS               (Length_Units: mm)
                    Type: PFOC
                  LCX,LCY:    0.000000  0.0000
                    NA,FL:    0.100000  40.000
```

```
             Aberrations: None
     Scalar/Quasi-vector: QV
      Calculation method: APRX
    Propagation distance:    40.00000
-------------------------------------------------------------

       Plot distribution: PLOT               (Length_Units: um)
                    Type: Intensity
       Logarithmic SCALE:    4.000000
               Xmin,Xmax:   -30.00000  30.000
               Ymin,Ymax:   -30.00000  30.000
     Color or Gray-scale: C
       Z-component (Y/N): N
   Save data files (Y/N): Y
         File identifier:  0
-------------------------------------------------------------

File management/Graphics: FMAN
                  Action: L
Data-file loaded to TEMP: ITOT.DAT
                  Action: A
 Data-file added to TEMP: IX00.DAT
           Weight Factor:    1.000000
                  Action: D
     Color or Gray-scale: C
                  Action: F
Saving TEMP in data-file: ITOT.DAT
                  Action: Q
-------------------------------------------------------------
```

Following import of the reflected field file, a square mask is applied to the distribution to cancel the fields induced by the FDTD absorbing boundary conditions near the edges of the domain. Then a DFT filter is applied to remove non-propagating evanescent fields ($S_0 > 1$) and to re-sample the distribution into a larger $25\mu m \times 25\mu m$ mesh, required for better sampling in the k_x, k_y wavevector space. The beam is propagated $4999.562\mu m$ (the difference between the focal length of $5mm$ and the $0.438\mu m$ already propagated

in the FDTD grid) to the collimating lens, then through the focusing lens to the image plane. The computed intensity distribution at the image plane is added to the file ITOT.DAT. After all reflected fields are propagated to the image plane, this file will contain the total sum of intensities.

The images of a flat, unmarked layer, and two sets of three randomly placed marks are shown in Fig 34. The images have different relative intensity scales. As expected, the image from a flat layer has uniform intensity distribution. The low intensity region in the middle of the central image is contributed by the central bump, which scatters the light, while the two pits on each side have higher intensity due to reflection of the light from pit walls toward the pit center. Similar effect is evident in the image of three circular pits.

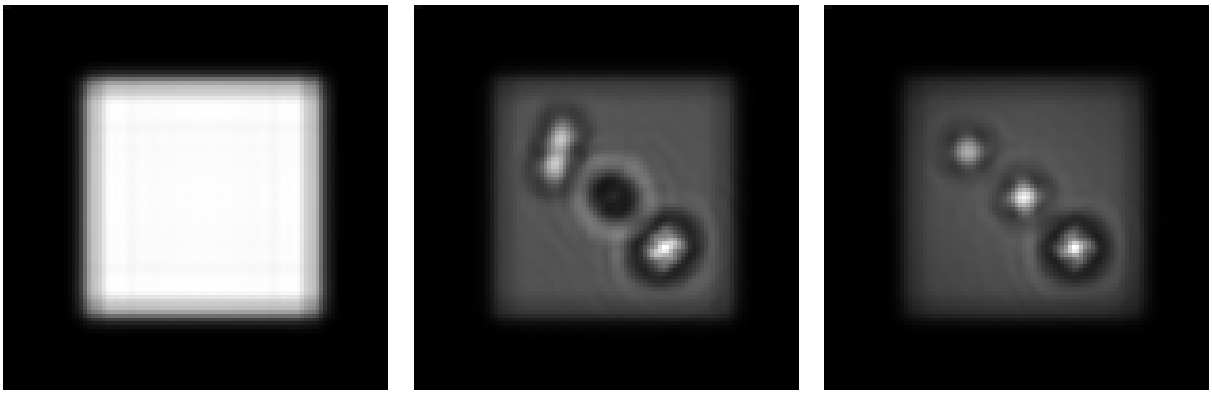


Figure 34: Left: image of a flat, unmarked layer, obtained using five beams with $(C_{11} = 0, \phi_{11} = 0)$, and $(C_{11} = 1.5, \phi_{11} = 60°, 150°, 240°, 330°)$. Center: image of three randomly placed marks with geometry shown in Fig 33, and source sampling described in the text. Right: image of three circular pits with the following parameters (top-left to bottom-right): width=700nm, 800nm, 1000nm, depth=50nm, 60nm, 60nm. The image on the bottom is that of a flat [elliptical-stadium] pit, the other two – of round [spherocylindrical-cap] pits.

References

[1] K.S. Yee, *IEEE Trans. Antennas and Prop.,* vol.14, 1966 pp.302-307.

[2] MPICH-A Portable Implementation of MPI, *http://www-unix.mcs.anl.gov/mpi/mpich/*

[3] DIFFRACT^{TM} software, MM Research, Inc., *http://www.mmresearch.com/*

[4] M. Mansuripur, A.R. Zakharian, and J.V. Moloney, "Interaction of Light with Subwavelength Structures", *Optics and Photonics News* **13** 3 (2003) 56-61.

[5] A.R. Zakharian, J.V. Moloney and M. Mansuripur, "Computer simulations of the near field effects in high-density optical disk data storage", *Computing in Optics*, a special issue of *Computing in Science and Engineering* **5**(6) (2003) 15-21.

[6] M. Mansuripur, A.R. Zakharian, and J.V. Moloney, "Transmission of Light through Small Elliptical Apertures", (Part - I), *Optics and Photonics News*, **15** (3) 38-43 (2004).

[7] M. Mansuripur, A.R. Zakharian, and J.V. Moloney, "Transmission of Light through Small Elliptical Apertures", (Part - II), *Optics and Photonics News*, **15** (4) 44-48 (2004).

[8] A.R. Zakharian, M. Mansuripur and J.V. Moloney, "Transmission of Light Through Small Elliptical Apertures", *Optics Express* **12**(12) (2004) 2631-48.

[9] Y. Xie, A.R. Zakharian, M. Mansuripur and J.V. Moloney, "Transmission of Light Through Slit Apertures in Metallic Films", *Optics Express* **12**(25) (2004) 6106-6121.

[10] M. Mansuripur, A. R. Zakharian, Y. Xie, and J. V. Moloney, "Light transmission through subwavelength slits and apertures", **(invited paper)** *4th Asia Pacific Data Storage Conference*, Taiwan, September (2004), *IEEE Trans. Magnetics* **41** (2) (2005) 1012-1015.

[11] T. Liu, A.R. Zakharian, R. Rathnakumar, M. Fallahi, J. V. Moloney, and M. Mansuripur, "Applications of photonic crystals in optical data

storage", *Proceedings of SPIE Optical Data Storage Conference* 5380, 430-438 (2004).

[12] T. Liu, A.R. Zakharian, M. Fallahi and M. Mansuripur, "Multimode Interference-Based Photonic Crystal Waveguide Power Splitter", *Journal of Lightwave Technology*, **22**(12) (2004) 2842-2846.

[13] T. Liu, A.R. Zakharian, M. Fallahi and M. Mansuripur, "Design of a Compact, Photonic-crystal-based Polarizing Beam-splitter", *Journal of Lightwave Technology*, to appear (2005).

[14] Y. Xie, A.R. Zakharian, M. Mansuripur and J.V. Moloney, "Transmission of light through a periodic array of slits in a thick metallic film", *Optics Express*, **13**(12) (2005) 4485.

[15] A.R. Zakharian, M. Mansuripur and J.V. Moloney, "1. Surface plasmon polaritons on metallic surfaces", *Optics Express* **15**(1) (2007) 183-197.

[16] A.R. Zakharian, M. Mansuripur and J.V. Moloney, "2. Surface plasmon polaritons on metallic surfaces", *IEEE Tran.on Magnetics* **43**(2) (2007) 845-850.

Appendix A

Computational domain decomposition for parallel processing

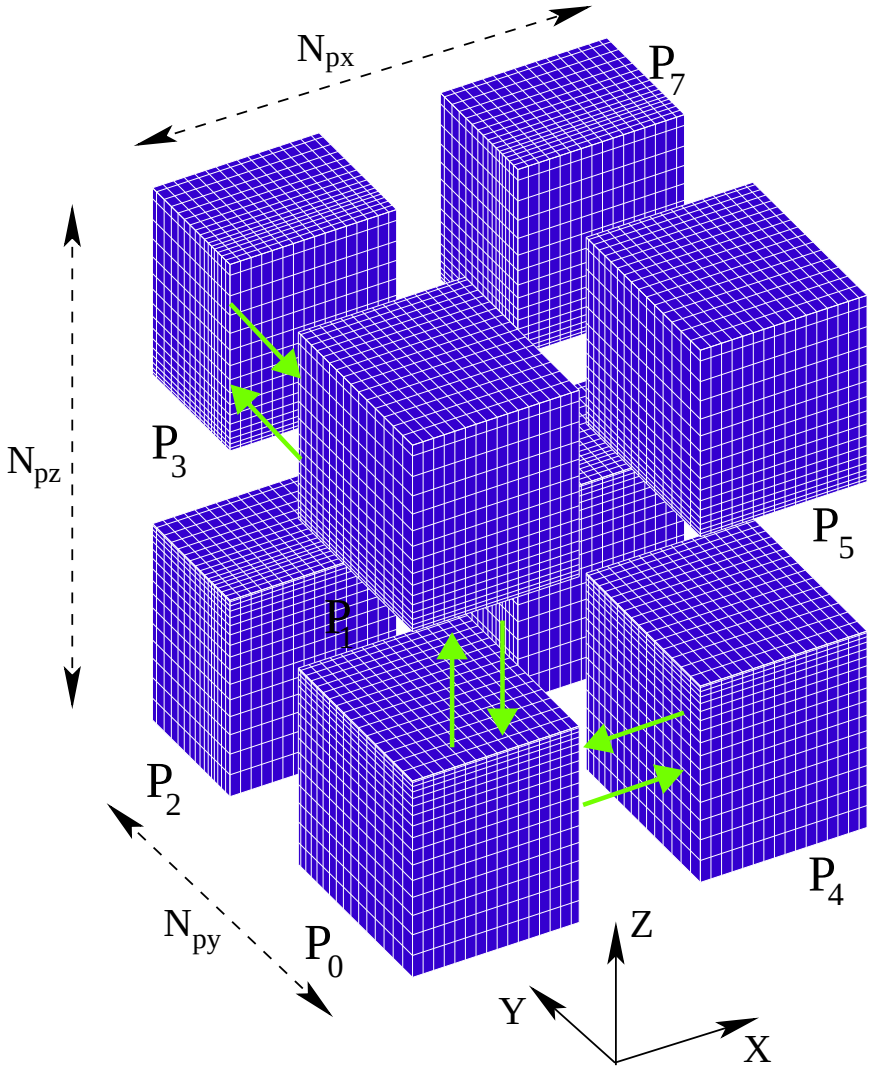


Figure 35: Non-uniform grid computational domain decomposition with $N_p = 8$, $N_{px} = 2$, $N_{py} = 2$, $N_{pz} = 2$. Processor rank changes first along the z-axis, then along the y- and x-axis.

This appendix describes conventions used for the computational domain decomposition in simulations with multiple processors. The number of processors and the desired decomposition of the computational domain are specified through the command line arguments:

```
mpiexec.exe -np Np Sim3D_Max.exe parameters.input Npx Npy Npz
```

Each processor is assigned an integer number (its rank) in the range from 0 to $N_p - 1$, where N_p is the total number of CPUs used in the computation. The N_{px}, N_{py}, and N_{pz} specify number of processors per coordinate direction, see Figure 35, so the product $N_{px}N_{py}N_{pz}$ must be equal to N_p. If the total number of grid points along each coordinate axis is n_x,n_y and n_z, the number of points per processor in the parallel computation will be n_x/N_{px},n_y/N_{py} and n_z/N_{pz}. The ratios of n_x to N_{px}, ... must be integer numbers. If the number of grid points along any of the axis is not integer divisible by the number of the processors specified for that axis, the number of grid points is increased to the closest integer, such that n_x/N_{px}, ... is an integer number. The new grid points are added to the corresponding axis, and the computational domain size is updated, as follows: for the `Uniform Grid` the $xmax$, $ymax$, or $zmax$ is increased ; for the `Non-Uniform Grid1` the extent of the $w1$ region for x- and y-axis is increased, or $h3$ region for the z-axis is increased; for the `Non-Uniform Grid2` the number of cells is increased in the last grid-specifying region for the x-, y- or z-axis.

The number of processors used in the computation has no effect on the number and structure of the input or output files, with the following exceptions:

1. The output files defined in the `Monitor` entries (section 4.24) of the parameter file are created by each processor separately. The rank of the processor that creates the file is appended to the filename. Each processor writes into the monitor file the data for the set of monitor spatial points that cross its computational sub-domain.

2. The output files that contain user-defined source data interpolated into the FDTD grid (section 4.18) are also written separately by each processor, with rank of the processor appended to the filename.

Figure 39 shows an example of a 2D computational domain with a monitor and user-defined source. The domain is partitioned among six processors. The resulting 3 output monitor files, and 2 output interpolated source data files will have different number of spatial points.

Appendix B

Parallel performance and load balancing

The parallelization of the code is based on the Single Program Multiple Data parallel programming model with partitioning of the spatial grid between a number of processors that simultaneously advance the solution in time. The explicit time update and short finite-difference stencil of the FDTD method lead to a high degree of memory access locality, and enable good speedup, as shown in Figure 36 using two types of tests. The tests on

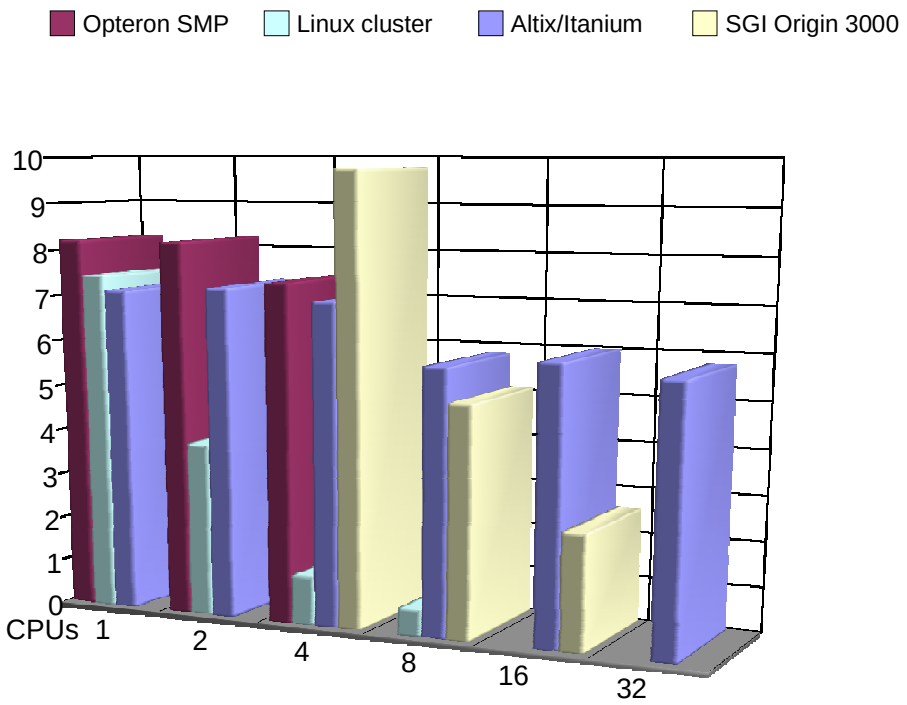


Figure 36: Measured relative runtime vs number of processors for a Linux cluster of AMD Athlon(tm) MP 2400+ workstations connected by a 1Gb/s network, 4 processor Microway Opteron846 system, SGI Altix/Itanium^2 and SGI Origin 3400 systems. Benchmark tests were identical only for Altix and Opteron systems, and these can be cross-compared.

Altix and Opteron systems, done with 5.6×10^6 grid points per processor, show that the run time stays almost constant when the problem size increases proportionally to the number of processors. The tests done on Linux cluster, with fixed 1GB total problem size, show linear decrease of the run time with number of CPUs. Similar scaling was measured on the SGI Origin system, with fixed total problem size 4×10^6 grid points distributed to 4,8 and 16 processors.

A number of factors can contribute to reducing the load balance and hence parallel computation efficiency. When material distribution in the computational domain and choice of the processor grid are such that one of the processors ends up with most of its sub-domain occupied by the material that requires more CPU-intensive update of the constitutive equations than materials present on other processors (e.g. Lorentz or MPM material models vs dielectric, Figure 37), then the overall run-time may be dominated by the processor with the highest load, reducing parallel efficiency.

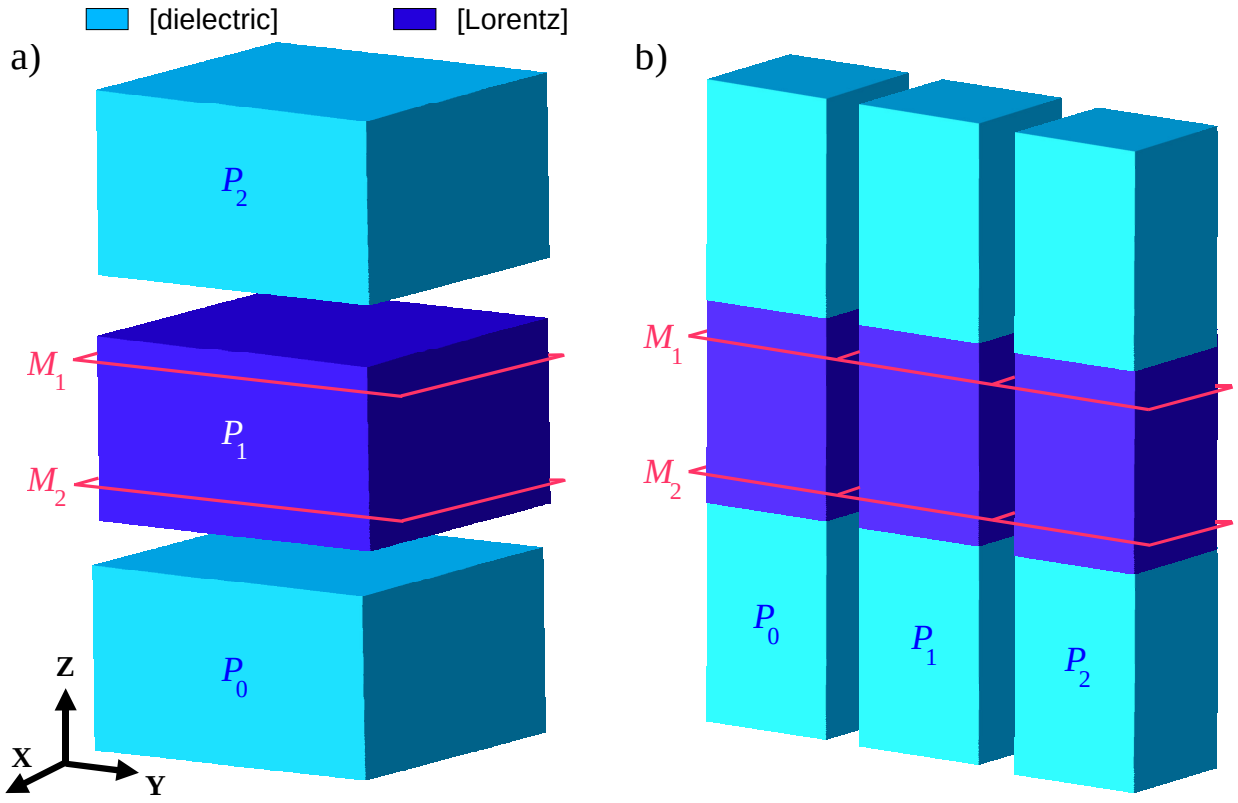


Figure 37: a) Example of computational domain decomposition along the z-axis, and resulting uneven load assignment: processor P_1 updates in its sub-domain computationally more expensive Lorentz material model and also has to process monitors M_1 and M_2 in the XY-plane. b) Load balance in this example can be achieved by partitioning the computational domain along the y-axis.

Similar problems occur when computationally expensive monitors (e.g. Fourier Transform or volume energy monitors) have locations such that, with some choice of the processor grid, all monitors end up in one or two CPUs, instead of being distributed evenly among all processors. When possible, proper processor grid should be selected to produce a better apriori load

balancing. The actual measured load balance may still vary due to the size of PML layers, which are more compute intensive than other equation updates, and per processor problem size that together with processor cache utilization can have an impact on parallel computation efficiency.

Appendix C

Staggered positioning of the field components in user-defined sources

Since the **E** and **H** fields in the FDTD method are not collocated in space and time, the user-defined sources in 2D computations (section 4.18) must properly take into account the staggering of the fields. The field locations for the 2D TM_x (Ex_Hy_Hz) and TE_x (Hx_Ey_Ez) modes are shown in Figure 38. When complex field amplitude distribution is specified in the user file

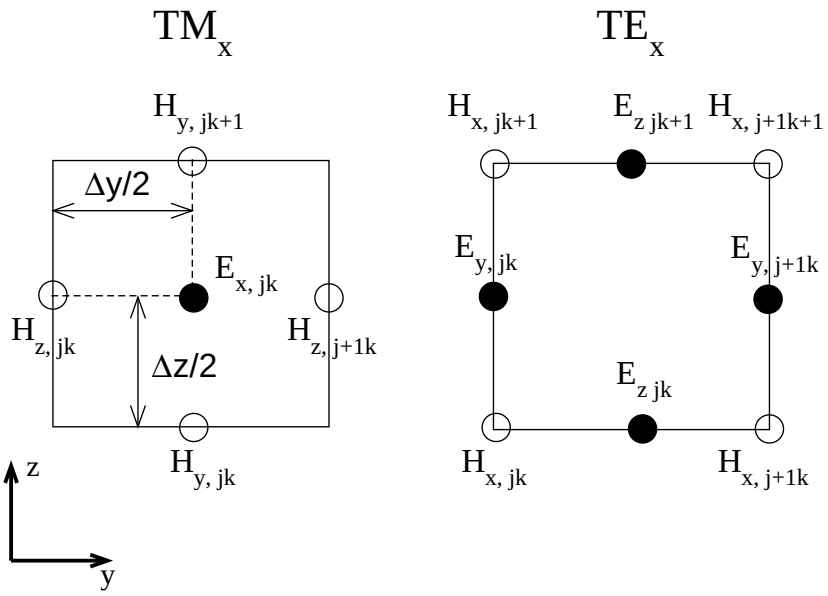


Figure 38: **E** and **H** field staggering in 2D Ex_Hy_Hz and Hx_Ey_Ez modes.

for a source defined along the y-axis (i.e. `zmin=zmax` in the UserSource: definition), the E_x, H_y fields for the Ex_Hy_Hz mode, and H_x, E_y fields for the Hx_Ey_Ez mode must be defined at positions separated by a half-cell size in the z-direction. Specifically, for the Ex_Hy_Hz mode, when the complex amplitude of the E_x field is specified along a line parallel to the y-axis (index j), with some constant `zmin=zmax` (index k) of the source, the corresponding

H_y complex fields must be defined for the same position along the y-axis, but with z-positions shifted by $-\Delta z/2$. For the Hx_Ey_Ez mode, when the complex amplitude of the E_y field is specified along a line parallel to the y-axis (index j), with some constant `zmin=zmax` (index k) of the source, the corresponding H_x complex fields must also be defined for the same position along the y-axis, but with z-positions shifted by $-\Delta z/2$.

For example, if a user-defined Ex_Hy_Hz source has electric and magnetic field dependence on the space coordinates in the form $E_x(y,z) = \bar{E}_x(y)e^{i\beta z}$ and $H_y(y,z) = \bar{H}_y(y)e^{i\beta z}$, then the input source file may contain complex amplitudes $\bar{E}_x(y)$ and $\bar{H}_y(y)e^{-i\beta\Delta z/2}$.

Similar definitions apply to a source defined along the z-axis (with the `ymin=ymax` in the UserSource: definition) for E_x, H_z fields for the Ex_Hy_Hz mode, and H_x, E_z fields for the Hx_Ey_Ez mode. In this case, the corresponding shifts of the field positions are along the y-axis.

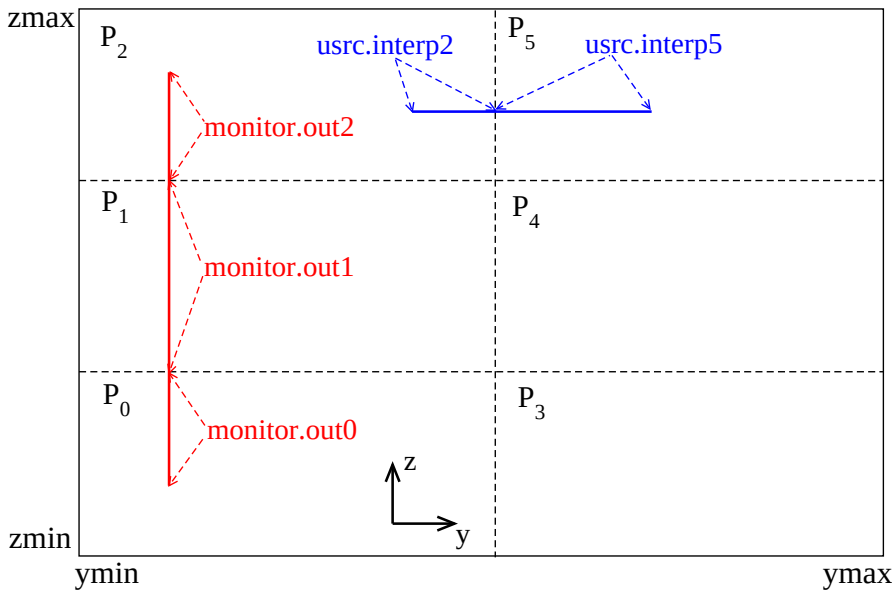


Figure 39: Example of monitor and user-defined source output for the 2D computational domain decomposition with $N_p = 6$ processors.

Appendix D

Monitor file formats

The content of various monitor files is described below by specifying the first line, containing monitor mode, type, number of points in space and time (or frequency) domain, followed by specification of the fields (columns) of the subsequent lines. Time is output in units of ns, frequency - in units of THz, and spatial coordinates - in units of μm. **E,H,S** and integrals are in MKS units.

```
TIME-HISTORY MONITORS
  line monitor of fields in 2D:
     [mode] [time-history] Ny Nz Ntime
     time coordy coordz F1 F2 F3 Sy Sz

     where F1=Ex F2=Hy F3=Hz for mode=Ex_Hy_Hz
           F1=Hx F2=Ey F3=Ez for mode=Hx_Ey_Ez

  line or plane monitor of fields in 3D:
     [3D] [time-history] Nx Ny Nz Ntime
     time coordx coordy coordz Ex Ey Ez Hx Hy Hz Sx Sy Sz

  plane monitor of energy flux integral in 3D:
     [3D] [integral-Sn-time-history] Ntime
     time integral_over_area_of_Sn

  volume monitor of energy in 3D:
     [3D] [energy-time-history] Ntime
     time integral_over_volume_(E*D+H*B)/2

FOURIER-TRANSFORM MONITORS
  "r" and "i" after field component names stand for "real" and
  "imaginary" parts of complex-valued data

  line monitor of fields in 2D:
```

```
   [mode] [fourier-transform] Ny Nz Nfreq
   frequency coordy coordz F1r F1i F2r F2i F3r F3i Sy Sz

   where F1=Ex F2=Hy F3=Hz for mode=Ex_Hy_Hz or
         F1=Hx F2=Ey F3=Ez for mode=Hx_Ey_Ez

line or plane monitor of fields in 3D:
   [3D] [fourier-transform] Nx Ny Nz Nfreq
   frequency  coordx  coordy  coordz Exr Exi Eyr Eyi Ezr Ezi ->
                                  -> Hxr Hxi Hyr Hyi Hzr Hzi ->
                                  -> Sx Sy Sz

plane monitor of energy flux integral in 3D:
   [3D] [integral-Sn-fourier-transform] Nfreq
   frequency integral_over_area_Sn
```

Appendix E

The "Publications" subdirectory of the distribution CD provides article reprints illustrating application of Sim3D_Max^{TM} to modeling optical disk storage media elements:

M. Mansuripur, A.R. Zakharian, J.V. Moloney, *Interaction of light with sub-wavelength structures,* Optics and Photonic News, March (2003) pp.56-61.

M. Mansuripur, A.R. Zakharian, J.V. Moloney, *Transmission of Light Through Small Elliptical Apertures (Part - I) ,* Optics and Photonic News, March **15** (2004) pp.38-43.

M. Mansuripur, A.R. Zakharian, J.V. Moloney, *Transmission of Light Through Small Elliptical Apertures (Part - II) ,* Optics and Photonic News, April **15** (2004) pp.44-48.

A.R. Zakharian, J.V. Moloney, M. Mansuripur, *Computer simulations of the near-field effects in high-density optical disk data storage,* IEEE Computing in Science and Engineering, Nov/Dec (2003) pp.15-21.