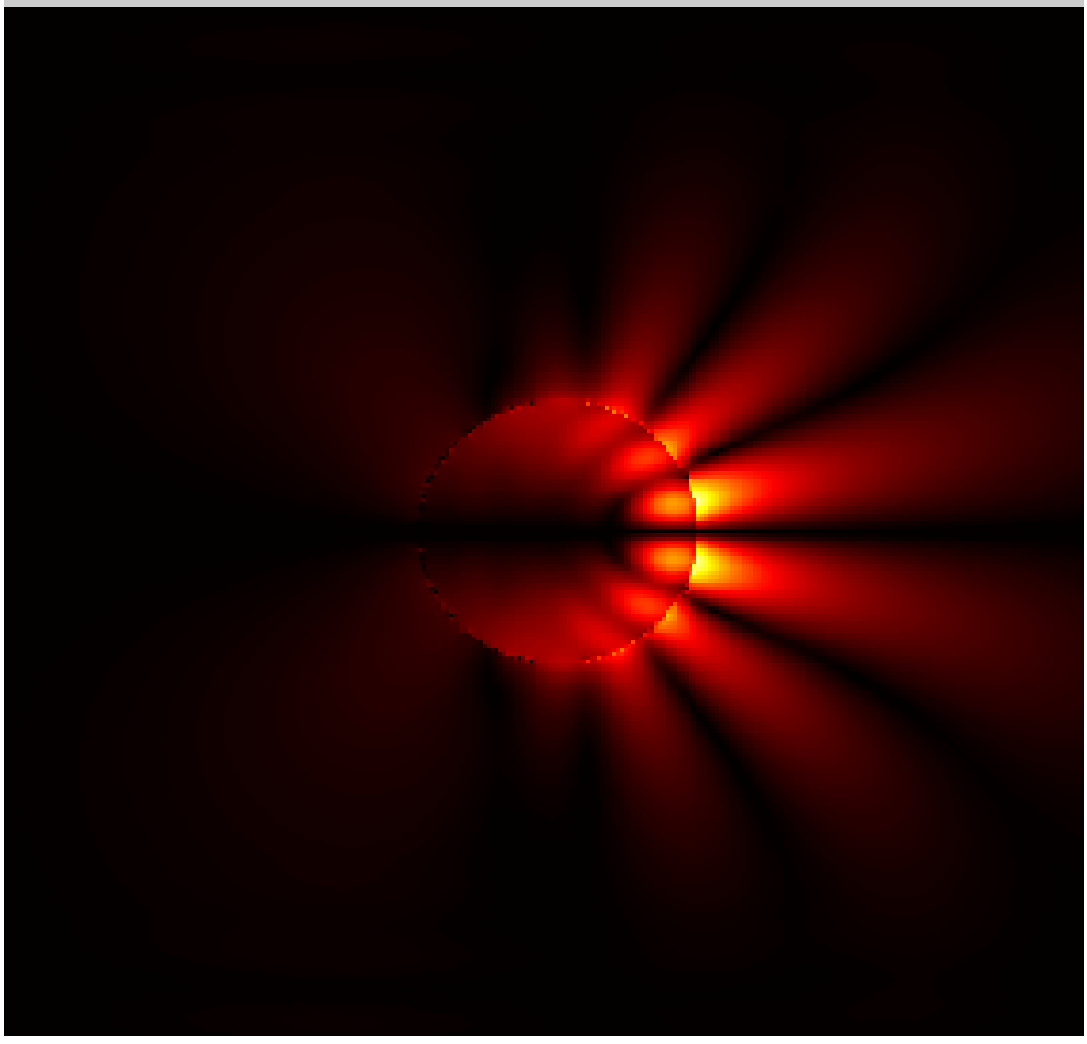
College of Optical Sciences
THE UNIVERSITY OF ARIZONA®



UAFDTD User's Manual

11/19/06

Table of Contents

Chapter 1: Introduction

This chapter includes the following information:

- What is UAFDTD?
- Main Components
- UAFDTD System Requirements
- Getting Started

What is UAFDTD?

The program UAFDTD stands for University of Arizona Finite Difference Time Domain. UAFDTD is designed to calculate electromagnetic fields in materials and space based on the Finite Difference Time Domain (FDTD) method. Kane Yee first introduced this method in 1966. The code is based on Maxwell's two curl equations in derivative form in the time domain. These equations are expressed in a linearized form by means of central finite differencing (see Appendix A). Only nearest neighbor interactions need be considered as the fields are advanced temporarily in discrete time steps over user specified spatial cells of rectangular shape. Since 1966, the FDTD method has become one of the most used numerical methods to calculate electromagnetic phenomena, because it has several advantages compared to other time-domain methods. The simple data structures make it straight forward to implement and facilitate vectorization and parallelization. The main drawback with the FDTD method is its inability to accurately model curved objects and small geometrical features. This difficulty is due to the Cartesian grid, which leads to a staircase approximation of the geometry, where small details are not resolved at all. In many applications, e.g. Radio Cross Section predictions for complex objects, the staircased approximations of the objects are not accurate enough.

There are several suggested remedies to circumvent the effects of the errors introduced by the staircase approximations. One of these remedies, local subcell modeling, is present in the UAFDTD program. Not only can UAFDTD model curved

dielectric surfaces, it also has the ability to work with a wide range of frequencies, stimuli, geometries, and materials. To this list can be added the advantage of computational efficiency. UAFDTD is unique among FDTD software in that it gives the user the ability to choose from a variety of diverse electromagnetic (EM) stimuli, with the option of two types of boundary conditions. As the FDTD method can be applied to a wide range of EM phenomena, from far field responses derived from near field, to scattered fields, antenna patterns, radar cross section, etc…; having the capability to choose source stimuli derived from several methods and boundary conditions best suited for the users needs greatly reduces the computation time. UAFDTD subcell modeling can also be used to improve accuracy without a significant increase in computational resources.

UAFDTD is especially useful for the rigorous calculation of the interaction between single-frequency optical fields and nano-scale geometrical elements. The geometric elements are composed of homogeneous, real optical materials, including dispersion. Real dielectric and metallic materials are easily described by their complex refractive index. UAFDTD is currently not well suited for transitory problems with multiple-frequency sources or for geometric elements containing perfect electrical conductors.

Main Components

Before beginning to use UAFDTD, it is necessary for the user to become familiar with the integrated workings of the program. There are three main components: 1) the **Geometry**, 2) the **Excitations**; and 3) the **Viewer**. The components are related to each other in this fashion: UAFDTD solves for EM fields around the **Geometry** using the excitation data provided by the **Excitations**. Then, the **Viewer** retrieves output data as requested by the user.

First, the **Geometry** is specified, including cell size, grid dimension and objects. The geometry starts as a three dimensional rectangular grid of cells, which is called the *calculation volume*. There is no theoretical limit to the size of this grid, but computer memory determines the maximum grid size that can be used in a simulation. Building objects inside the calculation volume using the cell edges creates the geometry. When

objects are built inside this grid, they look similar to any object that is built out of blocks. The UAFDTD model is basically a blocked approximation of the real-life object.

After the geometry is built, the next step is to specify the type of **Excitation** that will be used and where it will occur inside the grid. Discrete sources with loads or plane waves can be used in UAFDTD. The user must specify at which time steps result data will be saved for later observation with the **Viewer**.

UAFDTD System Requirements

UAFDTD system requirements are slightly more demanding than the system requirements of Matlab™ V7.1, which is required to run UAFDTD. The maximum grid size in the calculation volume is determined by the amount of system RAM. A significant amount of disk space is required to save the working files from the UAFDTD calculation. For example, a typical calculation of 200x200x200 voxels can require 2GB. Currently, UAFDTD only runs in the Windows XP operating system. Of course, the speed of the calculation depends on the speed of the processor used.

Getting Started

To get started, the user must open a Matlab™ command window, select the directory where the OptiScan file is installed, and type **'startup'**

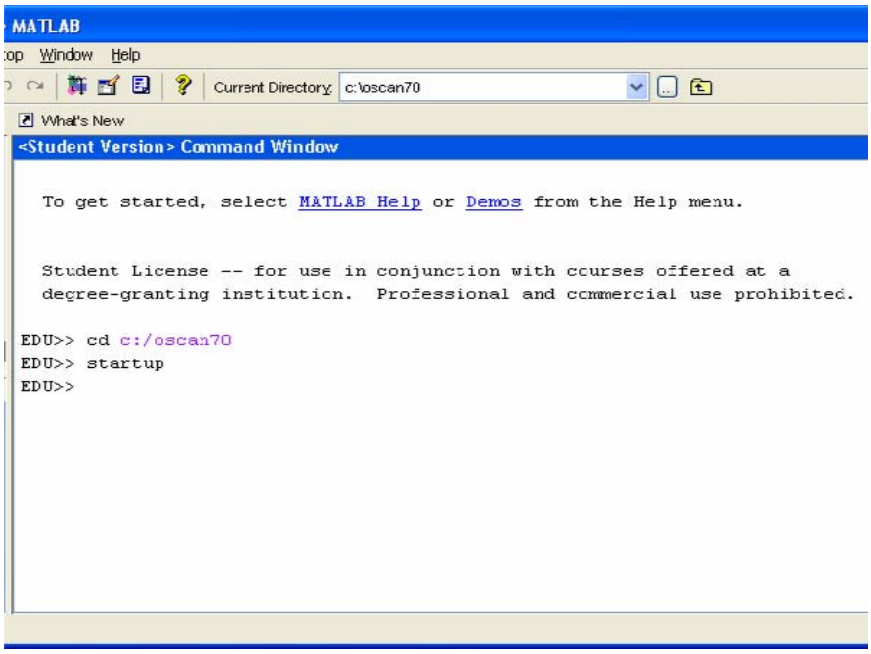


Figure 1.1: **Matlab™ Command Window**

The following window will pop up:

Figure 1.2: **OptiScan** start up window

The user can chose to open an existing project file by selecting **Open Existing OptiScan Project**, or the user can create a new project by selecting **Open New OptiScan Project**. The user can also open a sample project by clicking **Open Sample Project**, but there are no UAFDTD-specific projects in this introductory version. (For more information on the **Sample Projects** please see the **OptiScan Users Manual**)

Clicking on **Open New OptiScan Project** will open the following window:

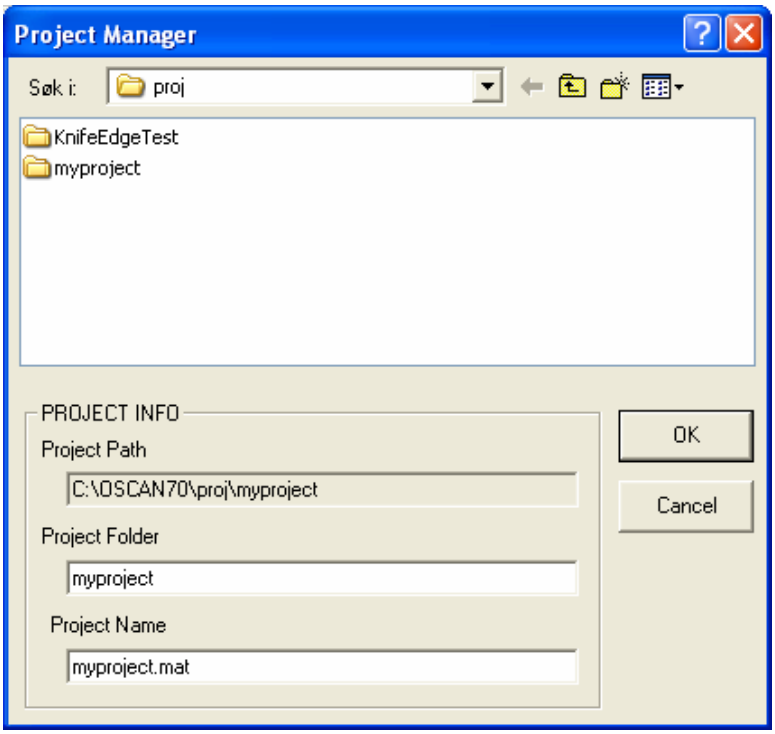


Figure 1.3: Create **OptiScan** Project

Please specify the name and location of the new project you wish to create. In this example the new project name is **myproject.mat** and will be saved in the folder **myproject**, located in **c:\OSCAN70\proj\myproject.** Click **OK** to continue creating the new project file. The following window will pop up:

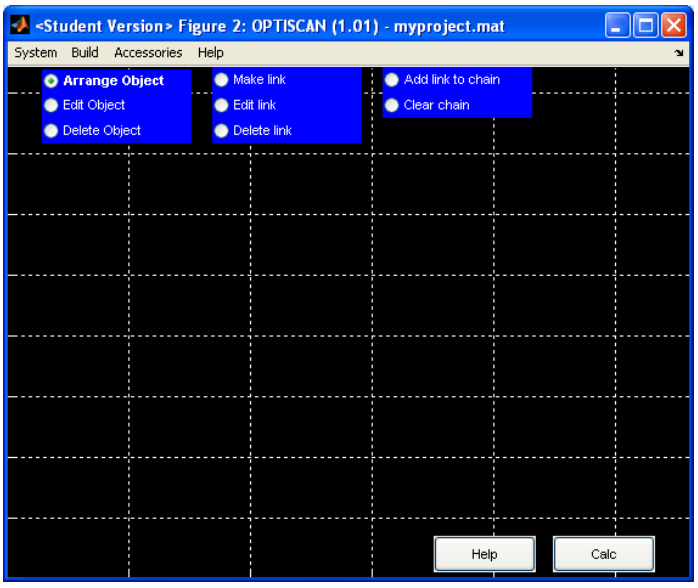


Figure 1.4: **OptiScan** project window

At this point, it is a good idea to save the initial project by selecting **Save Project** under the **System** menu bar item.

To start a FDTD calculation, click on **Accessories** on the toolbar and select **FDTD Calculator**:

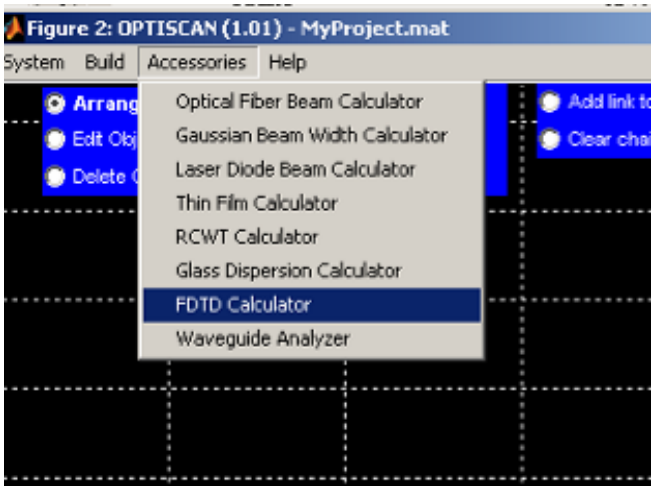
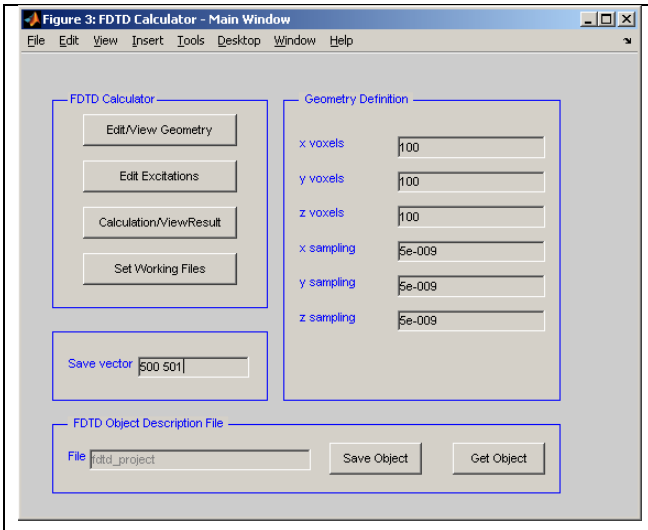


Figure 1.5: Selecting **FDTD Calculator** from the **OptiScan** project window

After selecting the calculator, the window in Figure 6 will pop up. This panel sets the size and sampling of the calculation volume in the **Geometry Definition Area**. It also provides button access to change the **Geometry**, **Excitation** and **Viewer** parameters in the **FDTD Calculator Area**. The **Save Vector** sets the time-step iteration numbers at which result data are saved for later viewing. The **FDTD Object Description File** contains a record of all the setup parameters for the calculation, including the calculation volume, geometry and excitation information. Object description files can be saved and retrieved at a later time for making new calculations or viewing results from a previous calculation. The object description files do not contain the actual result data; instead, they contain file pointer information. The directory of the object description file is the location in which result data are stored.

Chapter 2 describes how to edit the geometry, including adding box, sphere, cylinder and cone elements to the calculation volume. Chapter 3 describes how to change the excitation parameters, and Chapter 4 describes how to start the calculation and view results. Chapter 5 provides several examples of using the program.



FDTD Calculator Area:	Choose button to move to a different panel that describes the elements in the calculation volume (Geometry), type of wave used to illuminate the volume (Excitations), start the calculation (Calculate/View), or Set Working Files.
Geometry Definition Area:	Enter parameters that define the overall calculation volume.
Save Vector Area:	The time steps at which data are saved to a file. (Must be at least two sequential integers, like [2000 3000] or [500 501].)
FDTD Object Description File Area:	Select 'Get Object' to enter previously defined geometry, element list and excitation information. Select 'Save' to save the present settings.

Figure 1.6. Main Window of the FDTD Calculator Accessory.

Chapter 2: UAFDTD Geometries

This chapter includes the following information:

- Geometry Introduction
- Specifying Elements
  - Box
  - Sphere
  - Cylinder
  - Cone

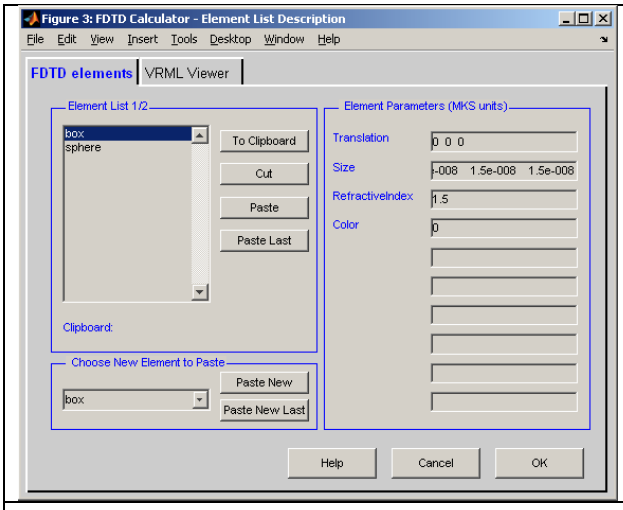
Geometry Introduction

To apply the FDTD method, the geometry of interest must be approximated as discrete material cells. Each cell edge may be defined with different dielectric properties. Since UAFDTD uses rectangular cells, the geometry is approximated using the edges, surfaces, or entire volumes of small rectangular boxes. The cell edges must be smaller than approximately one-tenth of a wavelength for accurate results (This is known as the Courant stability criterion, see Tavlove Chapter 2.7 and Appendix C). The cell dimensions must also be small enough to approximate the important geometrical features. UAFDTD provides several desirable geometry capabilities created from a library of basic objects, including cylinders, spheres, cones and boxes.

Looking at the **FDTD Calculator –Main Window** in Figure 1.6, the user can see that the UAFDTD work space is defined in three dimensions. First is the calculation volume number of unit cells along the x direction, y direction and z direction; **x voxels, y voxels and z voxels**, respectively. Following is the length of a unit cell along the x direction, y direction, and z direction; **x sampling, y sampling and z sampling**, respectively. The total calculation volume along the x direction is the (**x sampling**)*(**x voxels**), here is (5e-009)*(100) = 500nm or 0.5um. The same steps follow for the y and z directions. Please note that geometry origins always start at zero in the center of the calculation volume, meaning that the Cartesian coordinate origin (0,0,0), and all units are in MKS. Also, note that the calculation volume need not be symmetric. For example, 100nm x, 100nm y, 50nm z is perfectly valid.

Before beginning to build an object, it is very important that the user set the calculation volume to an acceptable size for the desired simulation. For example, if the calculation volume is given by 200nm x 200nm x 200nm, and the user tries to build a spherical element with a diameter of 300nm, the program will obviously not be able to comply.

Once the size of the calculation volume is sufficient for the desired simulation, the next step is to build the elements in the problem space. To begin editing the geometry, select the **Edit\View Geometry** button on the **FDTD Main Window** and the window shown in Figure 2.1 is displayed. The element list is similar to the one used for the thin-film target. Select an element in the **Element List Area** to change its parameters in the **Element Parameters Area**, including its position (*Translation* from the (0,0,0) center of the calculation volume), size, refractive index and other parameters.



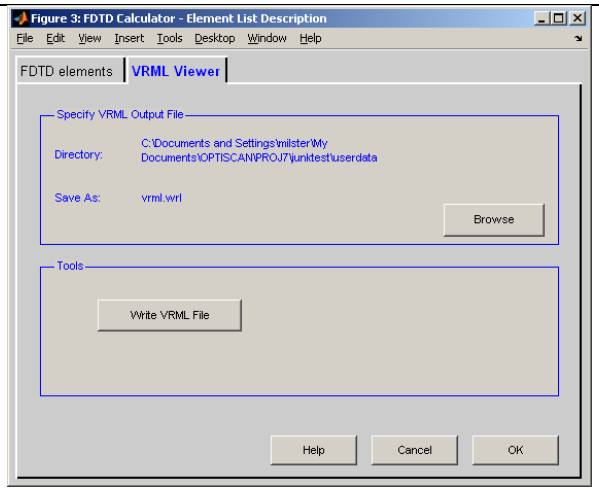
Element List Area:	Choose elements to edit that are in the calculation volume.
Element Parameters Area:	Edit area. Translation and size are [x y z] vector format.
Choose New Element Area:	Chose new elements from the drop-down menu to insert into the calculation volume.

Figure 2.1. Element List Description. (Choose **OK** to save and return to the **FDTD Main Window**)

To add new elements, select the element to add from the **Choose New Element to Paste** list. The user can choose from box, sphere, cylinder or cone shapes. Then, select ether **Paste New** or **Paste New Last**. **Paste New** puts the new element in the list just before the highlighted element. **Paste New Last** puts the element as last in the list.

Select the OK button when satisfied with the element list.

The user can choose the **VRML** Viewer tab to write a **VRML WORLD** file of the geometry constructed, as shown in Figure 2.2. The location of the VRML file is determined by the **FDTD Object Description File** on the **FDTD Main Window.** When this file is viewed with a VRML viewer, the calculation volume is shown as a light gray box.[1] With the **Cortona Client** viewer, choose the 'fit' button in the lower left-hand corner to place the calculation volume in the center of the window. The default setting is looking down into the volume from the positive z axis. Choose the 'study' button to rotate the volume. You can zoom in by using the 'fly' button. The elements should be placed in the volume correctly with respect to their size, shape and position. However, the grid sampling with the **VRML** viewer is usually much finer than with the actual grid settings described in the **FDTD Main Window** above.

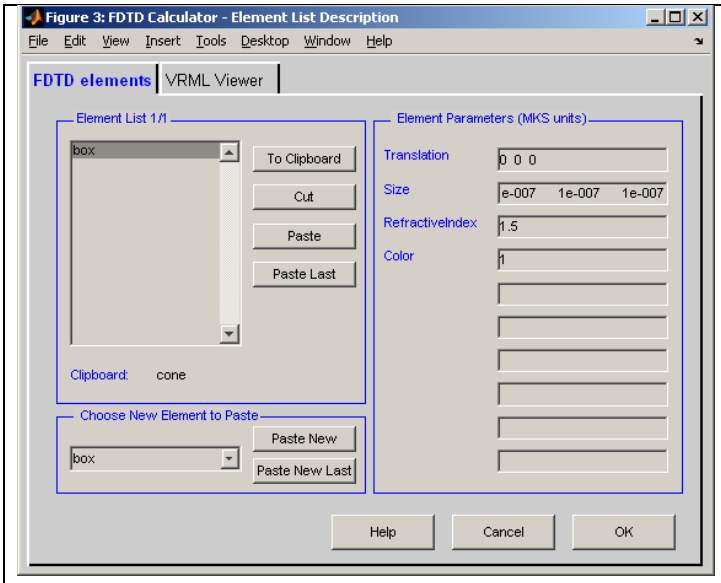


Output File Area:	Browse for selecting location of **VRML** file to be written.
Tools Area:	Select 'Write **VRML** File' to write the file. A separate viewer program can be used to see the VRML World.

Figure 2.2. **VRML** Viewer Panel. (Choose **OK** to save and return to the **FDTD Main Window**)

[1] The user needs to download a copy of a VRML viewer. There are several free viewers available that are adequate. [One example that works well is the Cortona VRML Client found on www.parallelgraphics.com.]

Specifying Geometries: Box



Translation	(x, y, z) displacement of the center of the element from the center of the calculation volume. Units are meters.
Size	x, y and z total length of the sides of the box element. Units are meters.
Refractive Index	Use the format a+b*i* to specify the complex refractive index of the homogeneous element.
Color	Color (integers 1-9) of the box in the VRML Viewer. The color does not have any effect on the calculation.

Figure 2.3. Box Element List Description. The picture below the edit widow shows how the box looks in the VRML Viewer. (Choose 'OK' to save and return to the FDTD Main Window)

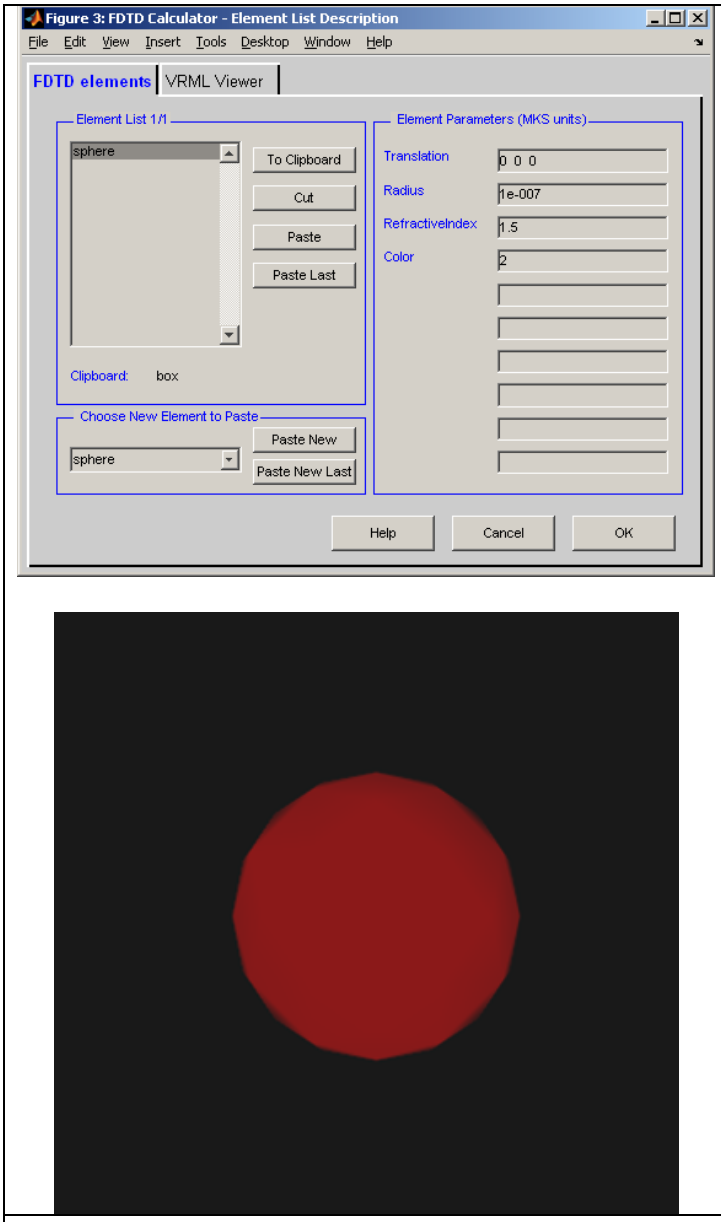
To build a box element (**Size** components need not be equal to each other, but of course can be if desired) the following information must be specified: **Translation**, **Size**, and **Refractive Index**. **Translation** is specified as a vector of x, y and z locations referenced to the center of the calculation volume. In this example, we will build a box in the center of our geometry (0,0,0). Previously, we created a calculation volume with x

sampling = y sampling = z sampling = 5e-009, and x voxels = y voxels = z voxels = 100. In our 500nm space we wish to create a rectangular box of dimension (150x150x150)*nm. We therefore enter the total length of the desired vectors starting from the center of our geometry: **Size** = 1.5e-008 1.5e-008 1.5e-008. **Note: there must be a space between each vector location (**Translation** input) and each vector size (**Size** input). The last required field for creating the box element is the **Refractive Index**. Here we are simulating a dielectric with a refractive index of 1.5. This is entered as 1.5 +0*j*, or simply as 1.5. To specify the refractive index for a metal, use the standard form a + b*j*.

After building the element, click **OK** to return to the **FDTD Main Window**.

Specifying Geometries: Sphere

The built-in sphere is a very straightforward element to use. In the **Edit Geometry** command window, the information in Figure 2.4 describing the sphere can be found.



Translation	(*x*, *y*, z) displacement of the center of the element from the center of the calculation volume. Units are meters.
Radius	Radius of the sphere. Units are meters.
Refractive Index	Use the format a+b*i* to specify the complex refractive index of the homogeneous element.
Color	Color (integers 1-9) of the box in the VRML Viewer. The color does not have any effect on the calculation.

Figure 2.4. Sphere Element List Description. The picture below the edit widow shows how the sphere looks in the VRML Viewer. (Choose 'OK' to save and return to the FDTD Main Window)

To build the sphere element, the following information must be specified: **Translation**, **Radius**, and **Refractive Index**, and **Color**. **Translation** is specified as a vector of x, y and z locations referenced to the center of the calculation volume. **Radius** is the radius of the sphere, defined by *r* in Figure 2.5 below.

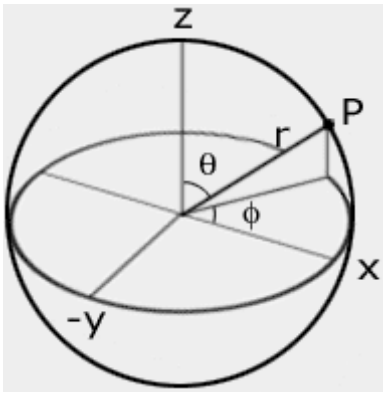


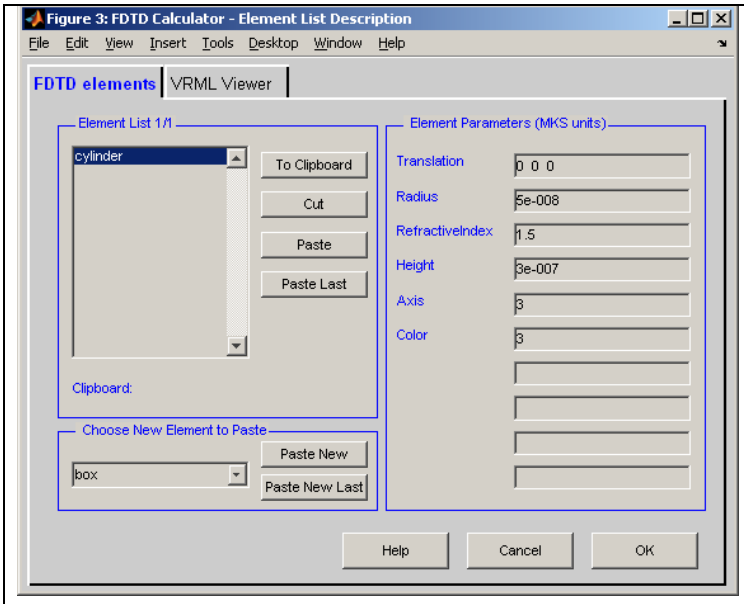
Figure 2.5: Sphere element diagram

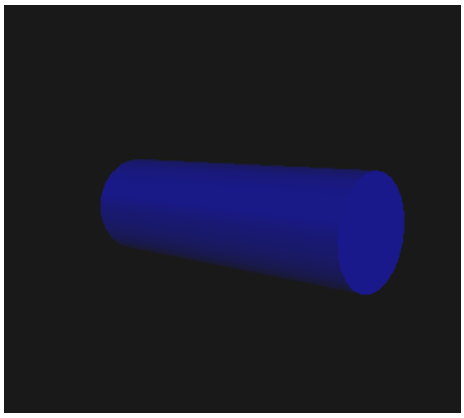
The last required field for creating the sphere element is the **Refractive Index**. Here we are simulating a dielectric with a refractive index of 1.5. This could also be entered as 1.5 +0j. To specify the refractive index for a metal please use the standard form a+b*j*. **Note: there must be a space between each vector location (**Translation** input).

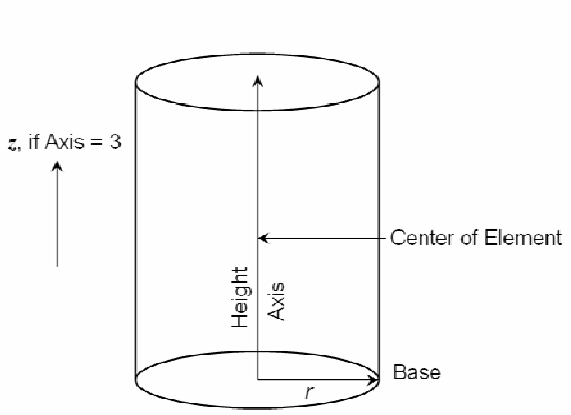
After building the element, click **OK** to return to the **FDTD Main Window**.

Specifying Geometries: Cylinder

The coordinates describing the cylindrical geometrical object are defined as shown in Figure 2.6 below.







Translation	(*x*, *y*, z) displacement of the center of the element along its axis from the center of the calculation volume. Units are meters.
Radius	Radius of the cylinder base. Units are meters.
Refractive Index	Use the format a+b*i* to specify the complex refractive index of the homogeneous element.
Height	Height of the cylinder along its **Axis**.
Axis	Specifies the orientation of the cylinder in the calculation volume. (1 = *x*, 2 = *y*, 3 = *z*). Must be an integer.
Color	Color (integers 1-9) of the box in the VRML Viewer. The color does not have any effect on the calculation.

Figure 2.6. Cylinder Element List Description. The picture below the edit widow shows how the cylinder looks in the VRML Viewer. (Choose 'OK' to save and return to the FDTD Main Window)

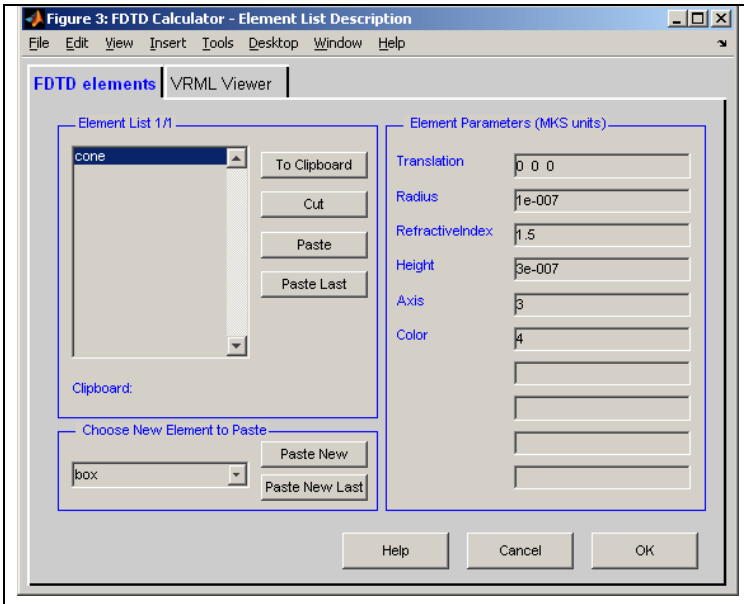
To build the cylinder element, the following information must be specified: **Translation**, **Radius**, **Refractive Index**, **Height** and **Color**. The center of the element will be displaced by the amount of **Translation** referenced to the center of the calculation volume. **Radius** is the radius of the cylinder base, and **Height** is the length of the cylinder along its axis. **Axis** defines the direction of the cylinder's main axis. **Axis** may have the values 1, 2 or 3, which correspond to the x, y and z axes, respectively.

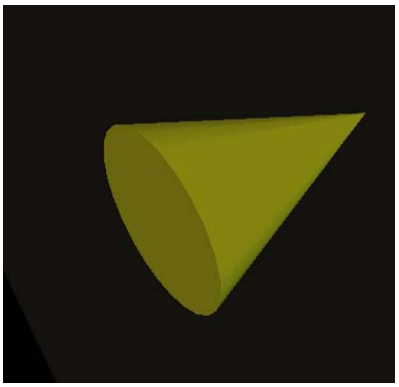
The last required field for creating the cylinder element is the **Refractive Index**. Here, we are simulating a dielectric with a refractive index of 1.5. This could also be entered as 1.5 +0*j*. To specify the refractive index for a metal please use the standard form a + b*j*. **Note: there must be a space between each vector location (**Translation** input).

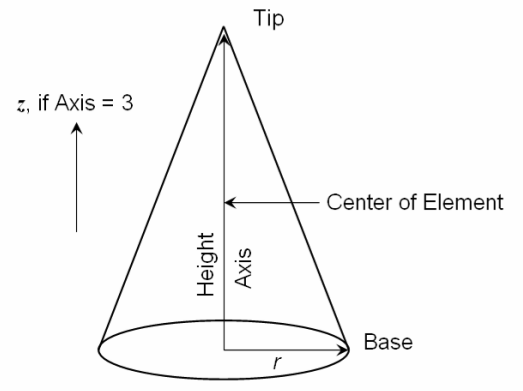
When done building the element, click **Ok** to return to the **FDTD Main Window.**

Specifying Geometries: Cone

The built-in cone element is defined as shown in Figure 2.7 below







Translation	(*x*, *y*, z) displacement of the center of the element along its axis from the center of the calculation volume. Units are meters.
Radius	Radius of the cone base. Units are meters.
Refractive Index	Use the format a+b*i* to specify the complex refractive index of the homogeneous element.
Height	Height of the cylinder along its **Axis**. Units are meters.
Axis	Specifies the orientation of the cylinder in the calculation volume. (±1 = ±*x*, ±2 = ±*y*, ±3 = ± *z*). Must be an integer.
Color	Color (integers 1-9) of the box in the VRML Viewer. The color does not have any effect on the calculation.

Figure 2.7. Cone Element List Description. The picture below the edit widow shows how the cone looks in the VRML Viewer. (Choose 'OK' to save and return to the FDTD Main Window)

To build the cone element, the following information must be specified: **Translation**, **Radius**, **Refractive Index**, **Height** and **Color**. The center of the element will be displaced by the amount of **Translation** referenced to the center of the calculation volume. **Radius** is the radius of the cylinder base, and **Height** is the length of the cone along its axis. **Axis** defines the direction of the cone's main axis. **Axis** may have the values ±1, ±2 or ±3, which correspond to the ±x, ±y and ±z axes, respectively. A minus value for **Axis** sets the tip of the cone in the negative axis direction. For example, the cone diagram shown in Figure 2.7 is shown with **Axis** = 3. If **Axis** = -3, the tip of the cone would be reversed and point along the *–z* axis.

The last required field for creating the cone element is the **Refractive Index**. Here, we are simulating a dielectric with a refractive index of 1.5. This could also be entered as 1.5 +0*j*. To specify the refractive index for a metal please use the standard form a + b*j*. **Note: there must be a space between each vector location (**Translation** input).

When done building the element click **Ok** to return to the **FDTD Main Window.**

Chapter 3: UAFDTD Electromagnetic Source Options

This Chapter includes the following information:

- Introduction to Boundary Conditions in UAFDTD
- Scattered Field Plane Wave UPML
- Plane Wave with PBC
- Point Source Array UPML
- Point Source Array with PBC
- Multi-Layer Scattered Field Plane Wave UPML

Introduction to Boundary Conditions in UAFDTD

A fundamental problem with the FDTD method is having appropriate boundary conditions for each simulation. One advantage of UAFDTD is that two boundary conditions (UPML and PBC) are offered in combination with several source options.

For many applications, the geometric elements are located in free space, and we would like to calculate the scattered or radiated fields to propagate into boundless space, satisfying a radiation condition. Unfortunately, any FDTD computational space is by necessity bounded, and when the scattered or radiated fields arrive at the boundary they are reflected back into the computational space, unless we take preventive measures with special boundary conditions.

To combat this reflection issue, UAFDTD has two boundary conditions: Uniaxial Perfectly Matched Layer (UPML) and Periodic Boundary Conditions (PBC). UPML is an artificial absorbing material (for details see Taflove Chapter 7.5). UAFDTD can terminate dielectric, lossy and Lorentz-type materials, and material objects can be directly in contact with UPML regions. UPML is an ideal boundary condition when dealing with scattered field phenomena. (Materials for which the real part of the refractive index is smaller than the imaginary part are currently modeled by using the Lorentz dispersion model (see Taflove Chapter 9.3.3).)

The second option, PBC, is designed for when dealing with periodic structures, such as waveguides, photonic crystals, and data marks, to name a few. When using PBC, the simulation time for a large periodic structure is drastically reduced. Instead of

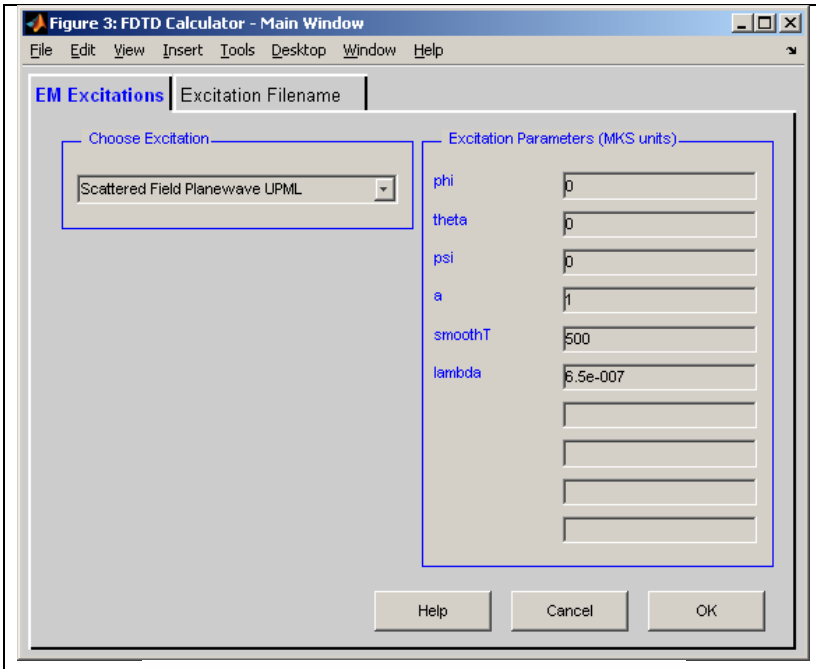
making a large geometry of the periodic structure, only one period of the pattern need be simulated to obtain results for the entire structure.

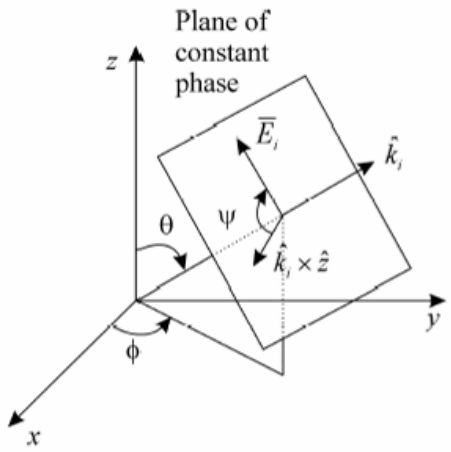
In addition, some problems require the field distribution around small asperities in otherwise infinite thin layers. For example, the problem of calculating the fields associated with a small hole in metallic thin films is interesting in near-field optics. The difficult problem of describing the response of this system to non-normal angles of incidence is solved in UAFDTD with the choice of Multi-Layer Scattered Field Plane Wave UPML excitation. (For more detail about this technique, see Olkkonen *et al.*, Proc. SPIE **5380**(1), pp. 360-7, 2004.)

Scattered Field Planewave UPML

The **Scattered Field (SF) Plane Wave UPML** excitation can be used to simulate scattering of a plane wave from an isolated geometric element in a uniform non-lossy background. This method employs the scattered-field technique (see Kunz), which specifies the incident field analytically. Maxwell's equations are reformulated in such a manner that only the scattered field components are solved by the FDTD method. The total field is obtained as a superposition of the incident and the scattered fields.

To specify the **Scattered Field Plane wave UPML** Excitation click **Edit Excitation** on the **FDTD Main Window**. The **EM Excitation** window will appear, as shown in Figure 3.1 below. To choose the **Scattered Field Plane Wave UPML** excitation, select the drop tab in **Choose Excitation** and select the **Scattered Field Plane Wave UPML** option.





Phi	Plane Wave rotation with respect to the positive *x* axis. Units are degrees.
Theta	Plane Wave angle with respect to the positive *z* axis. Units are degrees.
Psi	Rotation of the electric vector around the direction of propagation. Only values of 0 degrees (TE polarization) and 90 degrees (TM polarization) are allowed. Units are degrees.
a	Defines amplitude and is measured in [V/m]

	SmoothT	Value associated with a function designed to decrease calculation time and need not be changed.
	Lambda	Wavelength (with respect to vacuum) of the source. Units are meters.

Figure 3.1. **Scattered Field Plane Wave UPML** Description. (Choose **OK** to save and return to the **FDTD Main Window**).

The coordinates used to define the propagation direction and polarization of the incident plane wave in three dimensions are the standard spherical coordinates, phi and theta (see Figure 14). The incident unit wavevector ($\hat{k}$) is oriented with an angle theta (θ) relative to the +z-axis of the space lattice, where $0^0 < \theta < 180^0$; and with an angle phi (φ) relative to the +x-axis of the lattice, where $0^0 < \varphi < 360^0$. The **TE** and **TM** modes are represented by psi (ψ) and can have a value of 0 or 90 degrees, where 0 degrees represents **y** polarized and 90 degrees represents **x** polarized. The element **a** defines amplitude and is measured in [V/m]. **SmoothT**[2] is a value associated with a function designed to decrease calculation time and need not be changed. **lambda** is the wavelength, which the user can specify.

Once all desired information is specified, click **OK** to return to the **FDTD Main Window** and start the calculation.

[2] **SmoothT** is a smoothing time step of the excitation designed to avoid introducing high frequency spurious oscillations to the FDTD mesh. In all cases, the smoothing function is of the form:

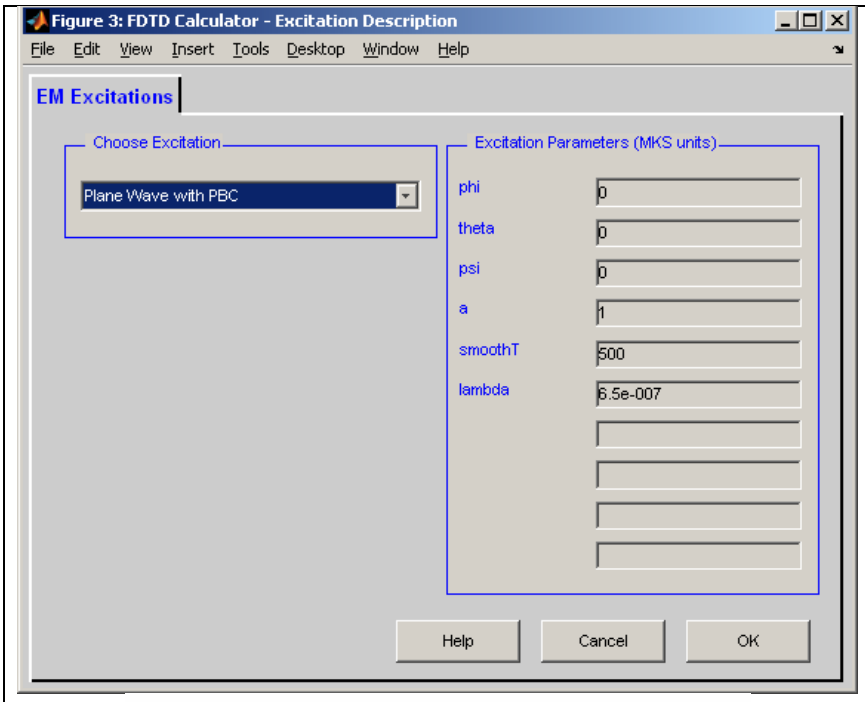
$$f(n) = \begin{cases} \sin^2\left(\frac{n}{\text{SmoothT}}\frac{\pi}{2}\right) & , n \le \text{SmoothT} \\ 1.0 & , \text{otherwise,} \end{cases}$$

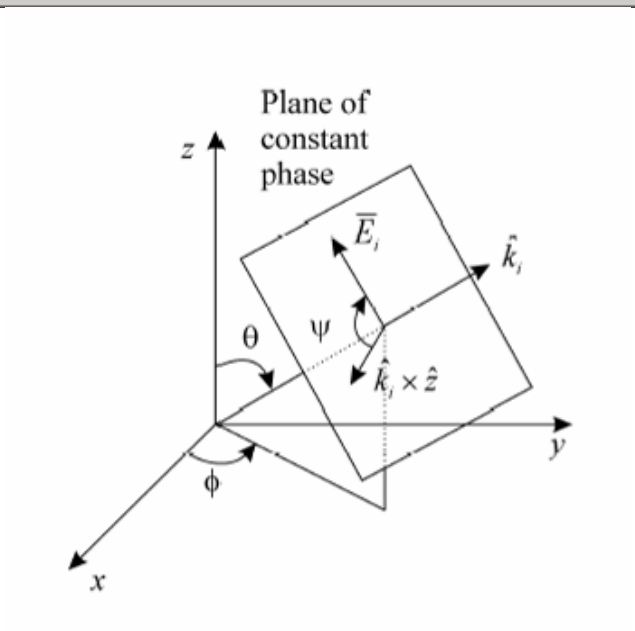
Where *n* refers to the time step index, which starts from zero and increases by one after each iteration.

Plane wave with PBC

The plane wave with PBC excitation option is designed for simulating any structure that is spatially periodic in the *x* and or *y* direction with an incident plane wave stimulus. This technique is optimal for simulations involving waveguides, photonic crystals, and data marks, to name a few. When using plane wave with PBC for periodic structures the simulation time is drastically reduced (compared with using UPML boundary condition for periodic structures). Instead of making a large geometry of the periodic structure, only one period of the pattern need be simulated to obtain results for the entire structure

To choose the **Planewave with PBC** Excitation click **Edit Excitation** on the **FDTD Main Window**. The **EM Excitation** window will appear, as shown in Figure 3.2 below. To choose the **Plane Wave PBC** excitation, select the drop tab in **Choose Excitation** and select the **Plane Wave PBC** option.





Phi	Plane Wave rotation with respect to the positive *x* axis. Units are degrees.
Theta	Plane Wave angle with respect to the positive *z* axis. Units are degrees.
Psi	Rotation of the electric vector around the direction of propagation. Only values of 0 degrees (TE polarization) and 90 degrees (TM polarization) are allowed. Units are degrees.
a	Defines amplitude and is measured in [V/m]

	SmoothT	Value associated with a function designed to decrease calculation time and need not be changed.
	Lambda	Wavelength (with respect to vacuum) of the source. Units are meters.

Figure 3.2. **Plane Wave with PBC** Description. (Choose **OK** to save and return to the **FDTD Main Window**).

The coordinates used to define the propagation direction and polarization of the incident plane wave in three dimensions are the standard spherical coordinates, phi and theta. The incident unit wavevector ($\hat{k}$) is oriented with an angle theta (θ) relative to the +z-axis of the space lattice, where $0^0 < \theta < 180^0$; and with an angle phi (φ) relative to the +x-axis of the lattice, where $0^0 < \varphi < 360^0$. The **TE** and **TM** modes are represented by psi (ψ) and can have a value of 0 or 90 degrees, where 0 degrees represents **y** polarized and 90 degrees represents **x** polarized. The element **a** defines amplitude and is measured in [V/m]. **SmoothT**[3] is a value associated with a function designed to decrease calculation time and need not be changed. **Lambda** is the wavelength, which the user can specify.

[3] **SmoothT** is a smoothing time step of the excitation designed to avoid introducing high frequency spurious oscillations to the FDTD mesh. In all cases, the smoothing function is of the form:

$$f(n) = \begin{cases} \sin^2\left(\frac{n}{\text{SmoothT}}\frac{\pi}{2}\right) & , n \le \text{SmoothT} \\ 1.0 & , \text{otherwise,} \end{cases}$$

Where *n* refers to the time step index, which starts from zero and increases by one after each iteration.

Once all desired information is specified, click **Ok** to return to the **FDTD Main Window**.

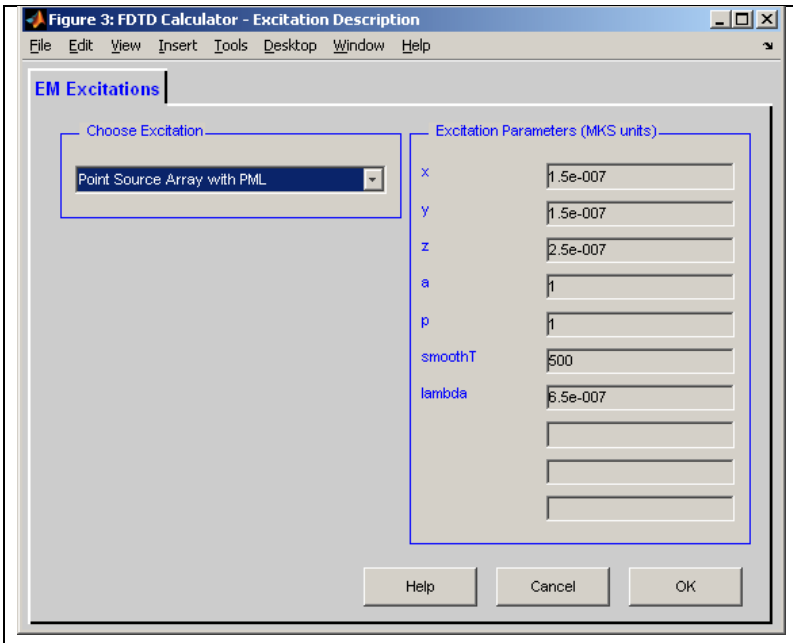
Point source Array PML

A pointsource array is a linear combination of "hard" sources. The user is free to specify the number of hard sources to be used in the point source array. The only limitation on number of sources is the computational speed and power of the machine used to do the calculation. For most simulations, including waveguide investigation, a small point source array is sufficient.

The hard source used in UAFDTD is a sinusoidal function created to satisfy a desired time function containing electric and magnetic field components in the FDTD lattice space. The source is set to run independently of any other objects in the model, and can be considered as a driving force.

The user defines the wavelength and polarization of the source. Only one wavelength can be specified to run the pointsource array. The user is also able to specify the number of sources to be used in the pointsource array, the distance between the sources in the array (in units of Yee cells), and the location of the array inside the FDTD space.

To use the Point Source Array excitation option, first click **Edit Excitations** on the **FDTD Main Window**. Figure 3.3 appears, as shown below.



X, Y,Z	Source locations in the calculation volume. Values are entered as vectors. Vectors must be the same length.
a	Defines amplitude and is measured in [V/m]
p	**P** can have values of 1, 2, or 3, which represent x, y, or z polarizations, respectively
SmoothT	Value associated with a function designed to decrease calculation time and need not be changed.
Lambda	Wavelength (with respect to vacuum) of the source. Units are meters.

Figure 3.3. **Point Source Array with PML** Description. (Choose **OK** to save and return to the **FDTD Main Window**).

The **Excitation Parameters x, y, z** are the locations of the sources along the x, y, and z axes, respectively. The elements **a** and **p** define amplitude [V/m] and polarization of the point sources, respectively. **P** can have values of 1, 2, and 3, which corresponds to x, y, and z polarization, respectively. All arrays **x, y, z, a,** and **p** can contain an arbitrary

amount elements, but they must be of equal length. The user is free to choose any source location as long as it lies within the boundaries created in the **Edit Geometry** section. **SmoothT**[4] is a value associated with a function designed to decrease calculation time and need not be changed. **Lambda** is the wavelength, which the user can specify. If the user wishes to run a calculation using PBC Planewave excitation, **a** must be set to 1.

Once all desired information is specified, click the save icon or **OK** to return to the **UAFDTD Main Window**.

Point Source Array with PBC

The Point Source Array with PBC excitation option is designed for simulating any structure that is spatially periodic in the *x* and or *y* direction with an point source stimulus. This technique is optimal for simulations involving waveguides, photonic crystals, and data marks, to name a few. When using Point Source Array with PBC for periodic structures the simulation time is drastically reduced (compared with using UPML boundary condition for periodic structures). Instead of making a large geometry of the periodic structure, only one period of the pattern need be simulated to obtain results for the entire structure.

See above Point Source Array with PML above for parameter description.

Multi-layer Scattered Field Plane Wave UPML

The Multi-layer (ML) scattered field plane wave should be used when a scatterer resides in a box-type geometry invariant in the x and y directions. The scatter can be, for example, an aperture in one of the layers of the box type geometry. This excitation enables the simulation of plane wave transmissions through an aperture in film having infinite lateral extent. Note that the real part of the refractive index of the scatterer has to

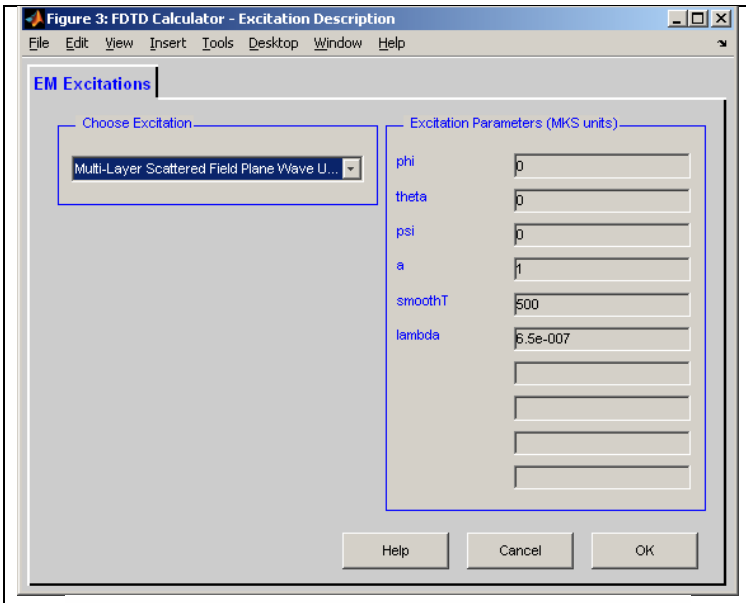
[4] **SmoothT** is a smoothing time step of the excitation designed to avoid introducing high frequency spurious oscillations to the FDTD mesh. In all cases, the smoothing function is of the form:

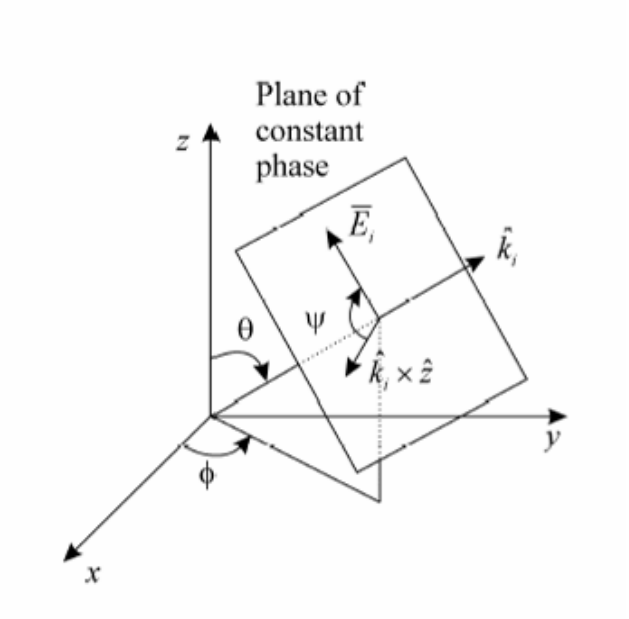
$$f(n) = \begin{cases} \sin^2\left(\frac{n}{\text{SmoothT}}\frac{\pi}{2}\right) & , n \le \text{SmoothT} \\ 1.0 & , \text{otherwise,} \end{cases}$$

Where *n* refers to the time step index, which starts from zero and increases by one after each iteration.

be larger than the imaginary part (like with air). However, the background box type geometry can be arbitrary in respect to the refractive indexes.

To use the Multi-layer Scattered Field Plane Wave UPML excitation option, first click **Edit Excitations** on the **FDTD Main Window**. The screen shown in Figure 3.4 appears.





Phi	Plane Wave rotation with respect to the positive *x* axis. Units are degrees.
Theta	Plane Wave angle with respect to the positive *z* axis. Units are degrees.
Psi	Rotation of the electric vector around the direction of propagation. Only values of 0 degrees (TE polarization) and 90 degrees (TM polarization) are allowed. Units are degrees.
a	Defines amplitude and is measured in [V/m]
SmoothT	Value associated with a function designed to decrease calculation time and need not be changed.
Lambda	Wavelength (with respect to vacuum) of the source. Units are meters.

Figure 3.4. Multi-Layer Scattered Field Plane Wave UPML Description. (Choose **OK** to save and return to the **FDTD Main Window**).

The coordinates used to define the propagation direction and polarization of the incident plane wave in three dimensions are the standard spherical coordinates, phi and theta. The incident unit wavevector ($\hat{k}$) is oriented with an angle theta (θ) relative to the +z-axis of the space lattice, where $0^0 < \theta < 180^0$; and with an angle phi (φ) relative to the +x-axis of the lattice, where $0^0 < \varphi < 360^0$. The **TE** and **TM** modes are represented by psi (ψ) and can have a value of 0 or 90 degrees, where 0 degrees represents **y** polarized and 90 degrees represents **x** polarized. The element **a** defines amplitude and is measured in [V/m]. **SmoothT**[5] is a value associated with a function designed to decrease calculation time and need not be changed. **Lambda** is the wavelength, which the user can specify.

Once all desired information is specified, click **OK** on **FDTD Main Window**.

Chapter 4: Running Simulation and Viewing Results

This Chapter includes the following information:

- Before Running Simulation: Save Vector
- Running Simulation
- Viewing Results
- Change File Names

Before Running Simulation: Save Vector

Before starting the **FDTD** calculation, set the desired time step indices which result data are saved. The **Save Vector**, which contains the time-step iteration numbers (positive integers) at which data are saved to files, is used for this purpose. The actual time increment between iterations is set automatically by UAFDTD, but simple

[5] **SmoothT** is a smoothing time step of the excitation designed to avoid introducing high frequency spurious oscillations to the FDTD mesh. In all cases, the smoothing function is of the form:

$$f(n) = \begin{cases} \sin^2\left(\frac{n}{\text{SmoothT}}\frac{\pi}{2}\right) & , n \le \text{SmoothT} \\ 1.0 & , \text{otherwise,} \end{cases}$$

Where *n* refers to the time step index, which starts from zero and increases by one after each iteration.

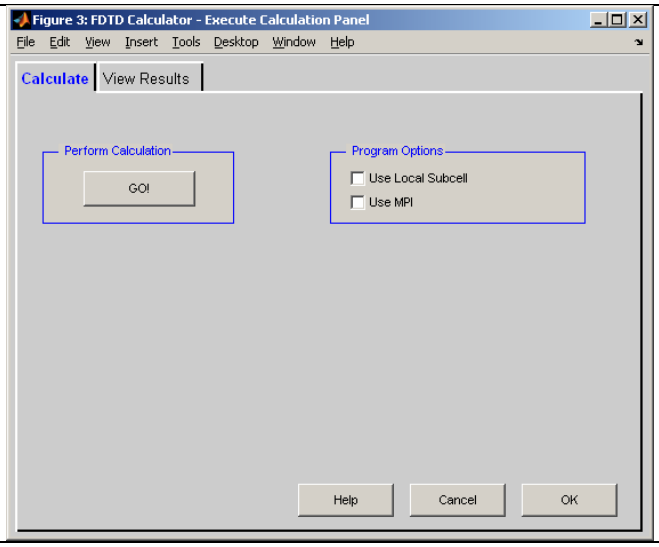
calculations require only approximately 500 iterations to converge to a steady-state solution. More complicated geometries require a larger number of time steps. The **Save Vector** information must be entered on the **FDTD Main Window.** Time steps must be listed in increasing order for the program to save data. The only data that can be viewed upon completion of the simulation are the time-step iteration numbers specified in **Save Vector**. Since the volume of data is usually significant for each time step, only a small number of time steps are saved for a typical calculation. However, there is no limit to the number of time-step data files that can be saved.[6]

Before starting the calculation, it is a good idea to save the FDTD Object Description File from the **FDTD Main Window**. This file contains all of the setup information you have entered and sets the directory for the output. First, select the 'Save Object' button on the **FDTD Main Window**. Then, select the proper directory and object file name with the browser.

Running the Simulation

After editing the **Geometry** and the **Excitations** and specifying the **Save Vector**, select the **Calculate/View Results** tab on the **FDTD Main Window** and the window shown in Figure 4.1 appears.

[6] Through MATLAB it is possible to link all saved time-step data together to create a 'movie' (see MATLAB help files).



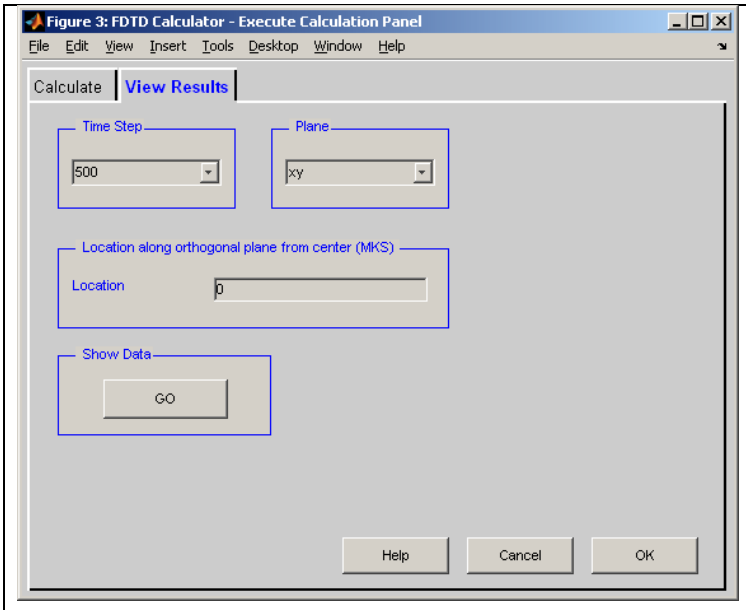
Perform Calculation Area:	Select the **Go** button to start the calculation. The program will display progress in the Matlab Command Line Interface Window.
Program Options Area:	Select Local Subcell for curved dielectric surfaces and MPI for parallel processing.

Figure 4.1: **Execute Calculation Panel**. (Choose **OK** to return to the **FDTD Main Window)**

The calculation time depends on the size of the **FDTD** simulation created. The progress of the calculation and estimated time of calculation is shown in the Matlab Command Window. The output continually updates an estimate of how much longer the calculation will take. The first estimates are not accurate. Please note that information cannot be entered or changed once this output appears.

Viewing Results

To view data select the **View Results** tab, as shown in Figure 4.2. Only time steps saved in the **Save Vector** on the **FDTD Main Window** are possible to view. After checking the **Save Vector** information**,** select the field of interest. Plot output is stored in the OptiScan project's plotdata folder.



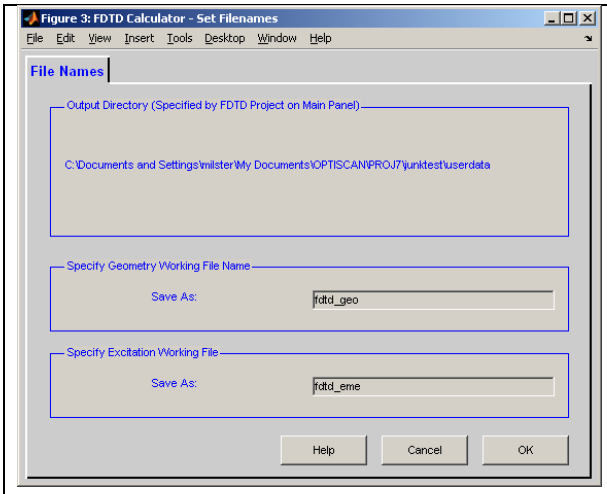
Time Step Area:	The pull-down menu selects the time step value (from the choices on the FDTD Main Window) to be observed.
Plane Area:	Select xy, xz or yz plane.
Location Area:	Enter offset (in meter units) of the distance between the center of the calculation volume and the observation plane along the axis orthogonal to the plane.
Show Area:	Select the 'Go' button to start the calculation for viewing results.

Figure 4.2. **View Results** Panel. (Choose 'OK' to save and return to the **FDTD Main Window**)

The user can select the **Save Vector**, plane and location from the center of the matrix along the orthogonal axis. Select the **GO** button to produce two figures. A series of progress indicator bars shows progression of the loading process. One figure contains the E field data, and the other figure contains the H field data of the plane selected in the GUI window. The user can select to view the x, y or z components. This interface window is very similar to the look object in an OptiScan calculation, and the user can view magnitude, real, imaginary, phase or irradiance of the components by selecting these options from the Views menu bar item. Also, the total irradiance can be viewed.

Change File Names

To change or save file names, click **Set Working Files** on the **FDTD Main Window** the window shown in Figure 2.30 will pop up. Usually, it is not necessary to change these files, as they are only temporary files used during the calculation.



Output Directory Area:	Set by saving FDTD Object File in FDTD Main Window.
Specify Geometry Working Filename Area:	Enter new name for working file, if required. (Usually, this is not required.)
Specify Excitation Working Filename Area:	Enter new name for working file, if required. (Usually, this is not required.)

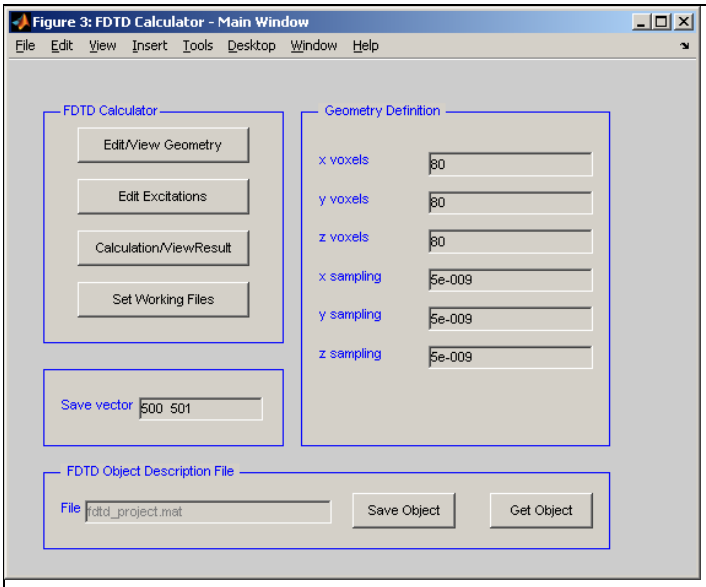
Figure 4.3 File Names Description. (Choose **OK** to save and return to the **FDTD Main Window**)

Chapter 5 Examples:

This Chapter includes the following information:

Example 1a: Cone on flat dielectric surface

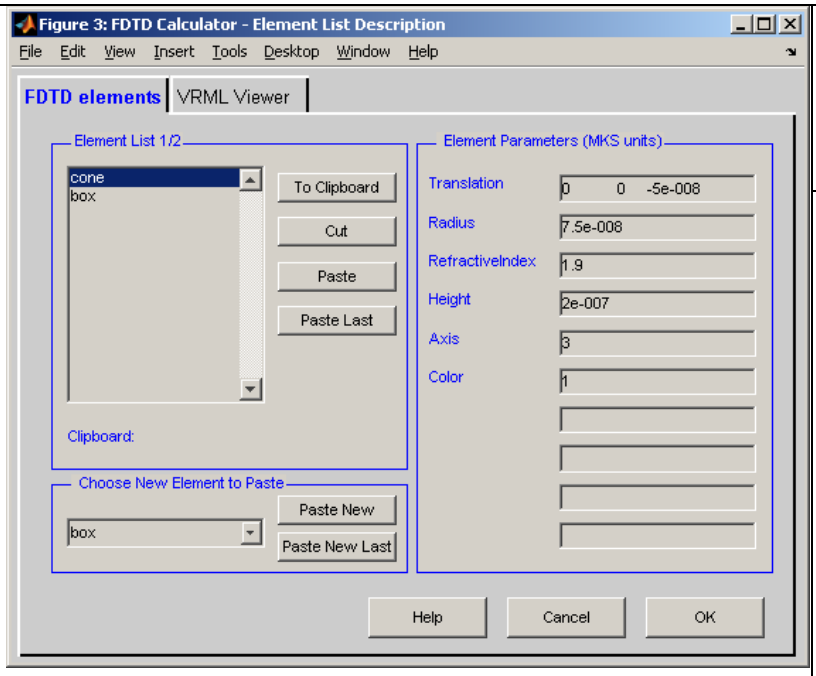
Here we have an example simulation of a cone attached to a flat dielectric surface. The calculation volume is 80 voxels on a side with 5nm sampling, for a total calculation volume of 400nm. The data are saved after 500 and 501 iterations. The geometry consists of a cone and a box. The cone base radius is 75nm, and the height is 200nm. It is shifted slightly down in the calculation volume by 50nm. The box has a large extent in the x and y directions in order to simulate an infinite transverse plane. Both colors are set to 1 (green), in order to indicate that they are the same index of refraction. The excitation used is Multi-Layer Scattered Field Plane Wave UPML in order to simulate the infinite plane properly.



FDTD Calculator Area:	
Geometry Definition Area:	Calculation Volume is set at **x voxels** = **y voxels** = **z voxels** = 80, with **x sampling** = **y sampling** = **z sampling** = 5nm
Save Vector Area:	The number of time steps at which data are saved to a file. (here is set to 500 and 501)
FDTD Object Description File Area:	Select 'Get Object' to enter previously defined geometry, element list and excitation information. Select 'Save' to save the present settings.

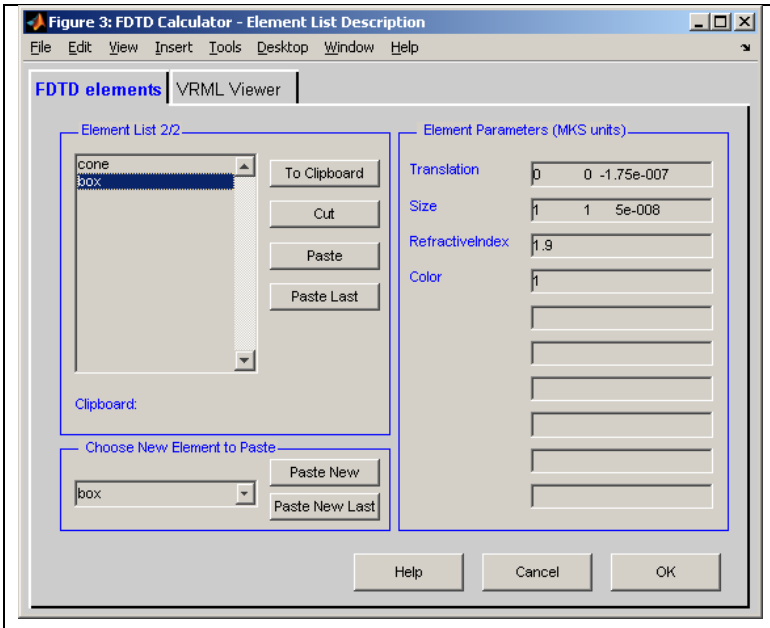
Figure 5.1. Main Window of the FDTD Calculator Accessory.

After specifying the UAFDTD work space the next step is to click the **Edit/View Geometry** to create the flat dielectric surface and the dielectric cone.



Element List Area:	Cone and box are the elements selected and pasted.
Element Parameters Area:	For the cone: **Translation** is set to (0 0 5e-008) **Radius** is set to 7.5e-008. **Refractive Index** is 1.9. **Height** is set to 2e-007. **Axis** is set to 3. **Color** is set to 1.
Choose New Element Area:	New Element chosen is **box**

Figure 5.2. Element List Description. (Choose **OK** to save and return to the **FDTD Main Window)**



Element List Area:	Cone and box are the elements selected and pasted.
Element Parameters Area:	For the cone: **Translation** is set to (0 0 -1.75e-007) **Size** is set to (1 1 5e-008). **Refractive Index** is 1.9. **Color** is set to 1.
Choose New Element Area:	All elements are now chosen and specified

Figure 5.3. Element List Description. (Choose **OK** to save and return to the **FDTD Main Window**)

Once all geometry elements are specified it is recommended to view the newly created geometry in the VRML viewer to verify that all elements were correctly specified.

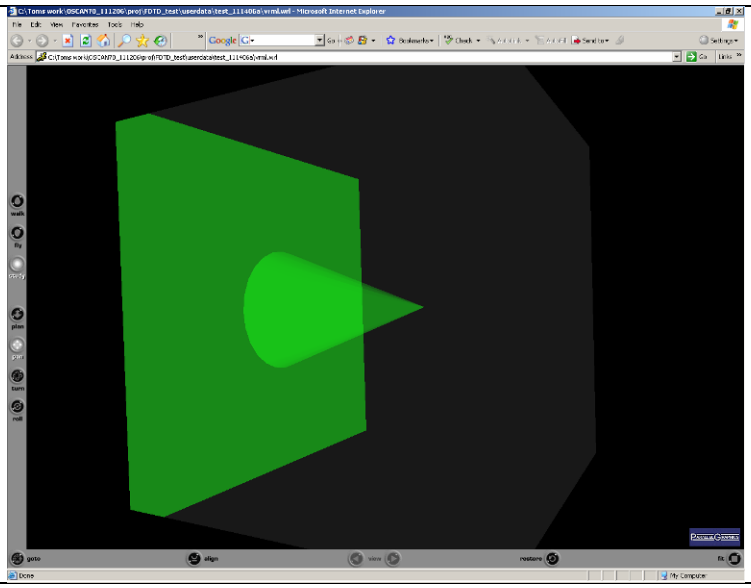
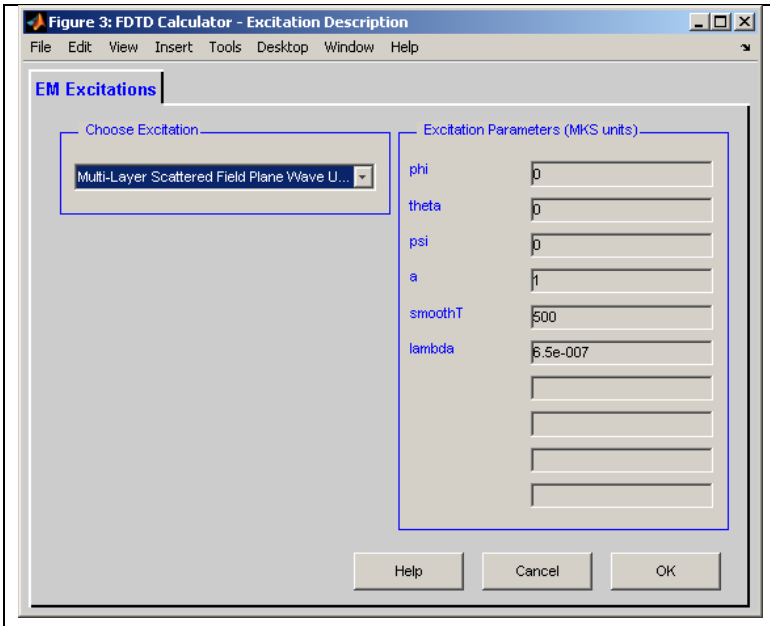


Figure 5.4. Dielectric cone on a flat substrate. Both materials are the same index of refraction (n = 1.9). The cone is located with its apex pointing in the positive *z* axis. The excitation plane wave originate from the –*z* direction and propagates to the +z direction.

Once all geometry elements are satisfactory, the next step in creating a simulation is to choose the **EM Excitation** (see Chapter 3 for details)



Choose Excitation Area:	**Multi-Layer Scattered Field Plane Wave UPML** is the selected excitation
Excitation Parameters Area:	Parameters are set as follows: **phi** = **theta** = **psi** = 0, **a** = 1, **smoothT**= 500, **lambda** = 650nm

Figure 5.5. Excitation Description. (Choose **OK** to save and return to **FDTD Main**

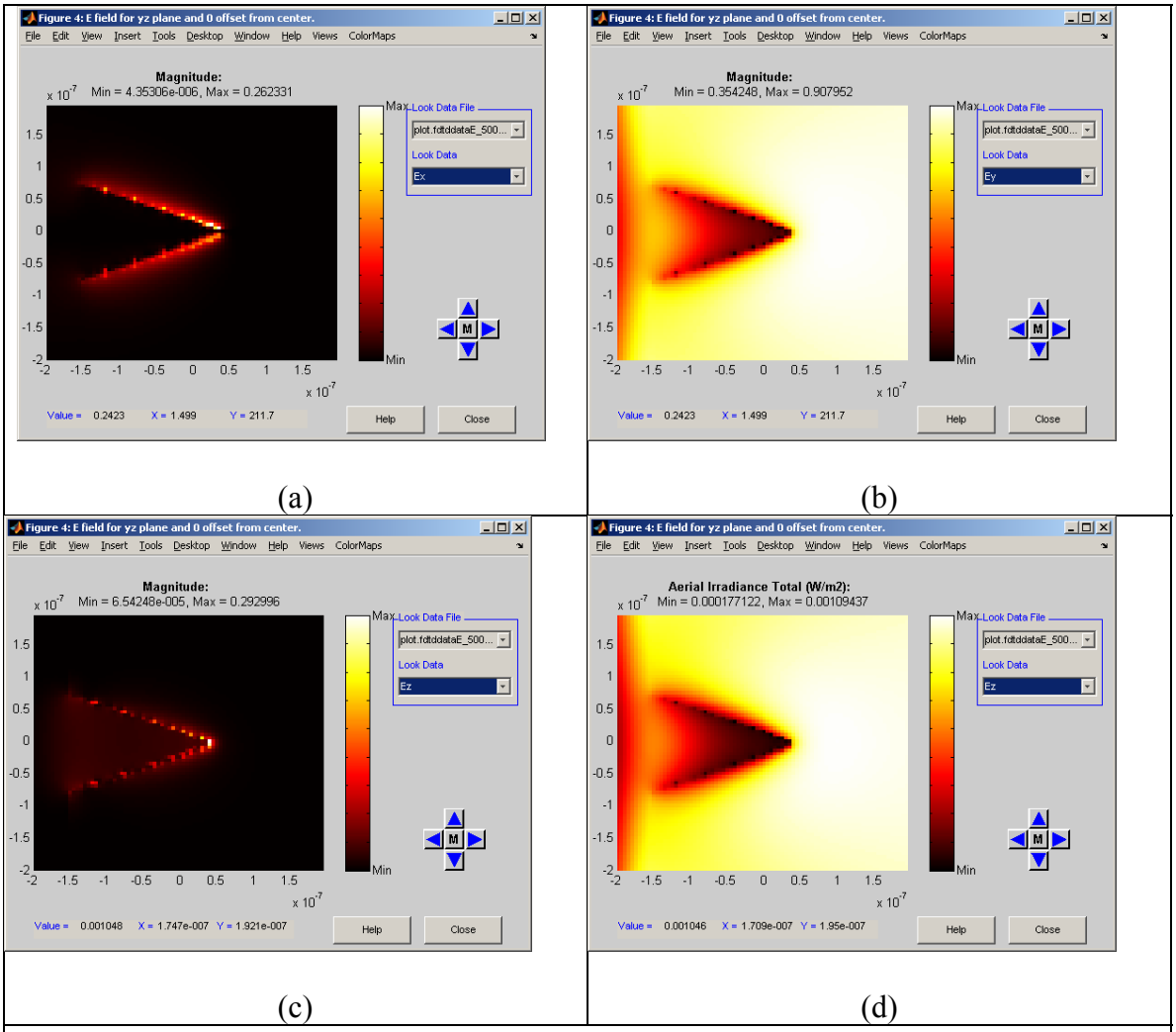
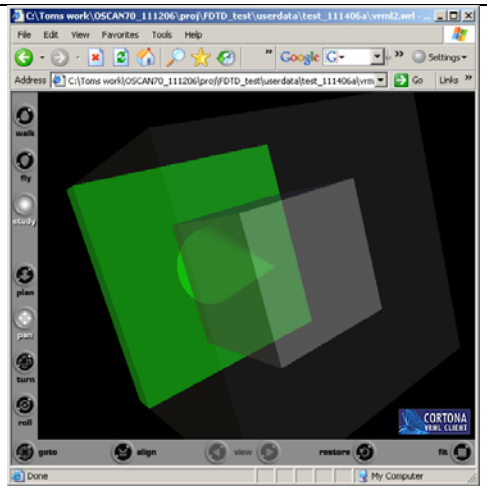
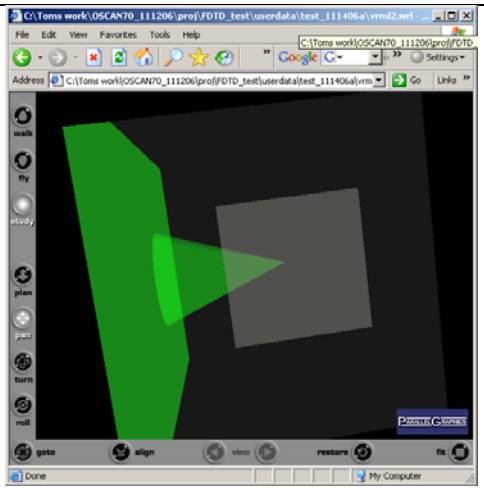


Figure 5.6. Results for Example 1a (yz plane at Location = 0). (a) Ex field magnitude; (b) Ey field magnitude; (c) Ez field magnitude; and (d) Irradiance (sum of square magnitudes for x, y and z fields).

Example 1b: Truncated cone on flat dielectric surface



(a)



(b)

Figure 5.7. Truncated dielectric cone on a flat substrate. The cone of Example 1 is truncated by placing a box (n = 1) over the tip of the cone. The default calculation volume refractive index is n = 1.

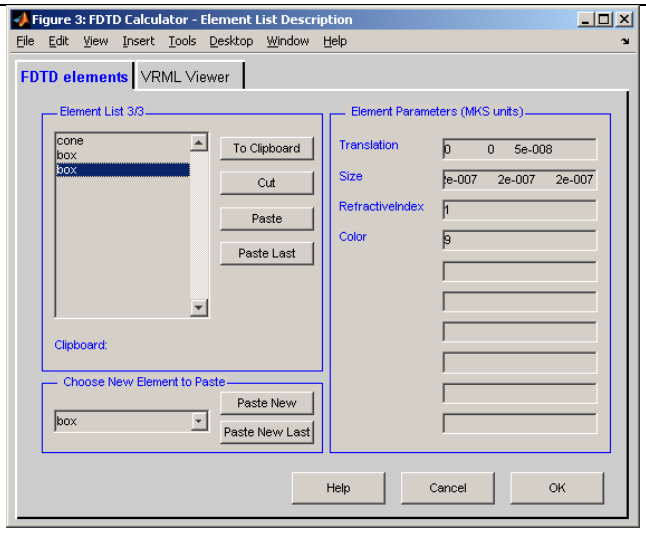


Figure 5.8. A box is added last in the Element List to overwrite the tip of the cone.

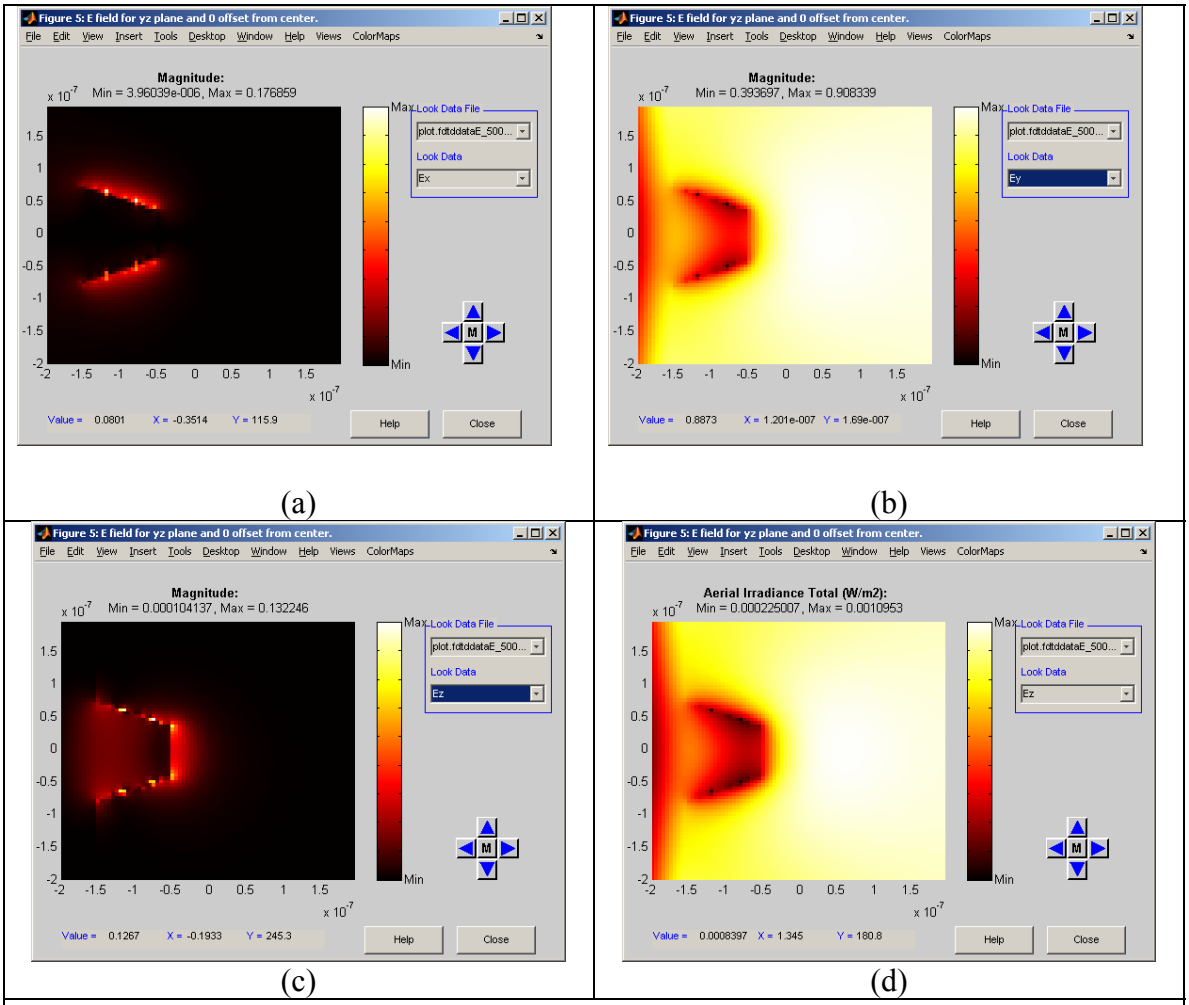


Figure 5.9. Results for Example 1b (yz plane at Location = 0). (a) Ex field magnitude; (b) Ey field magnitude; (c) Ez field magnitude; and (d) Irradiance (sum of square magnitudes for x, y and z fields).

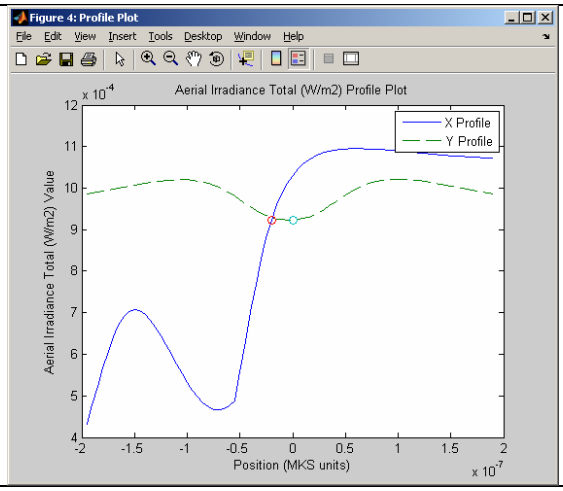


Figure 5.10. Profile of Example 2 Total Irradiance (yz plane at Location = 0). This profile is acquired by simply placing the cursor over the image and clicking with the mouse button.

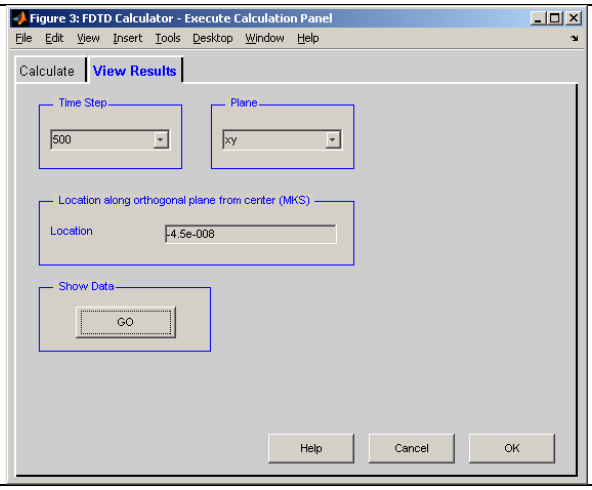


Figure 5.11. Setup for xy plane observation of Example 1b just above the tip of the truncated cone.

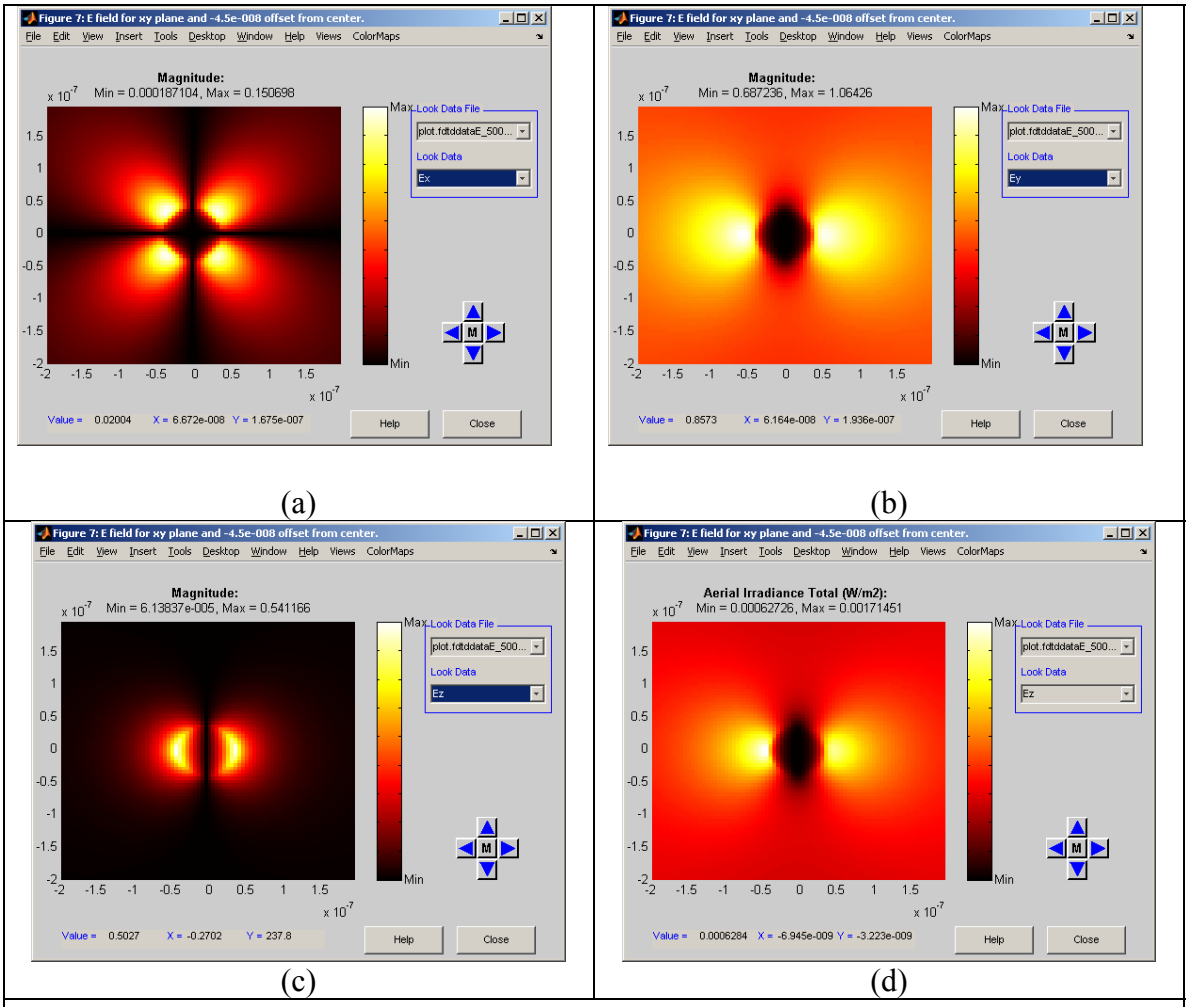


Figure 5.12. Results for Example 1b (xy plane at Location = -45nm, just abov the tip of the truncated cone). (a) Ex field magnitude; (b) Ey field magnitude; (c) Ez field magnitude; and (d) Irradiance (sum of square magnitudes for x, y and z fields).

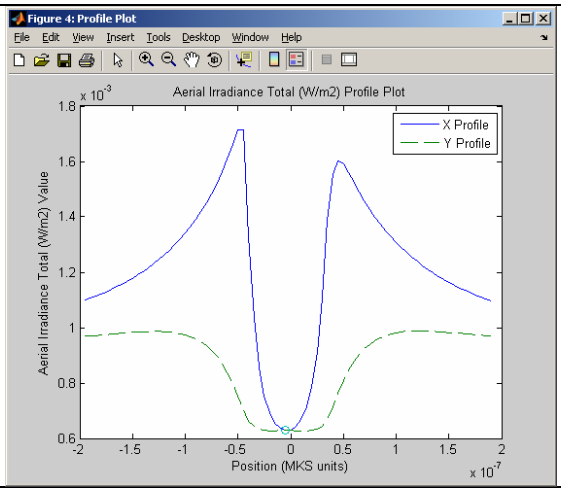


Figure 5.13. Profile of Example 1b Total Irradiance (xy plane at Location = -45nm, just above the tip of the truncated cone). This profile is acquired by simply placing the cursor over the image and clicking with the mouse button.

Example 2: Dielectric Sphere and Box

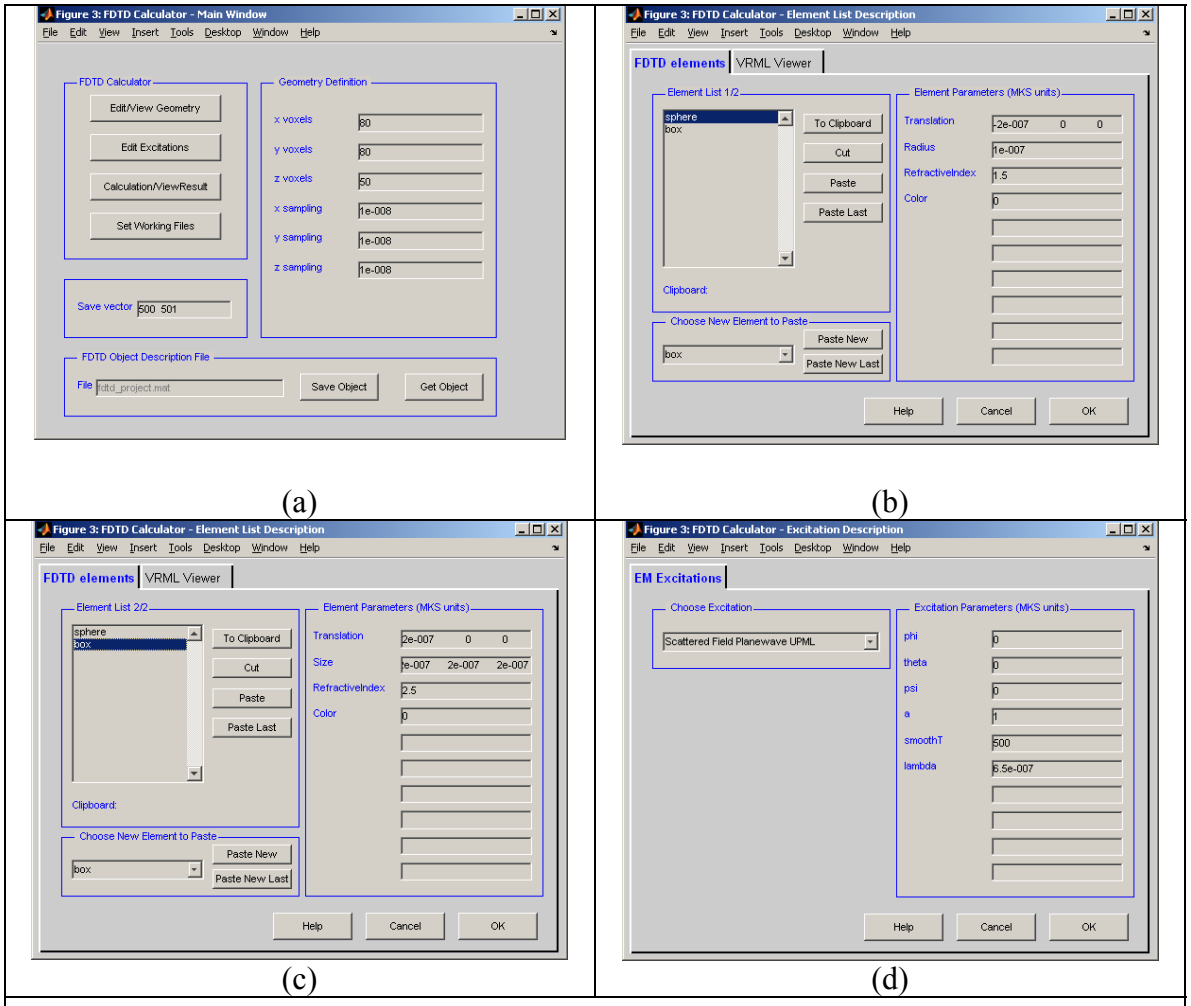


Figure 5.14. Panel description of Example 2: Dielectric Box and Sphere. (a) The calculation volume is 80 voxels on a side with 5nm sampling, for a total width of 400nm. The data are saved after 500 and 501 iterations; (b-c) The geometry consists of a box and a sphere. The sphere radius is 100nm with $n = 1.5$. It is shifted -200nm in the x direction. The box is a cube with 200nm sides and $n = 2.5$. It is shifted 200nm in the x direction. Both colors are set to 0, which will produce different colors in the VRML viewer file; and (d) The excitation is "Scattered Field PlaneWave UPML".

Once all geometry elements are specified it is recommended to view the newly created geometry in the VRML viewer to verify that all elements were correctly specified.

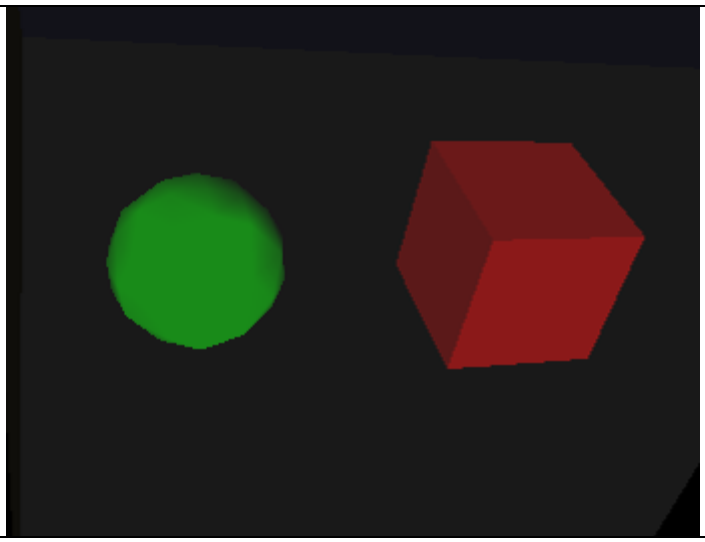


Figure 5.15. Dielectric sphere and box. The excitation plane wave is from the –*z* direction and is *y* polarized.

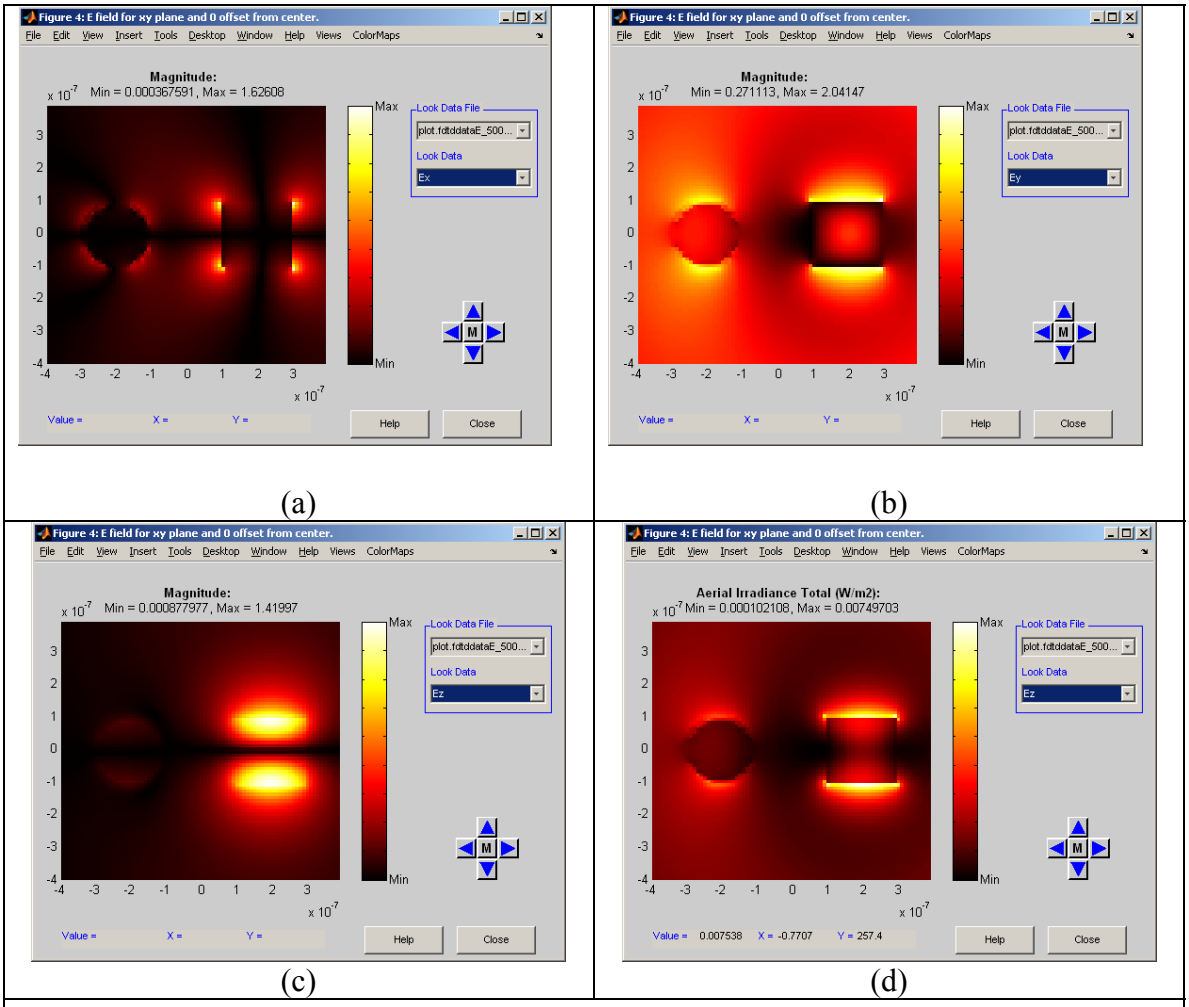
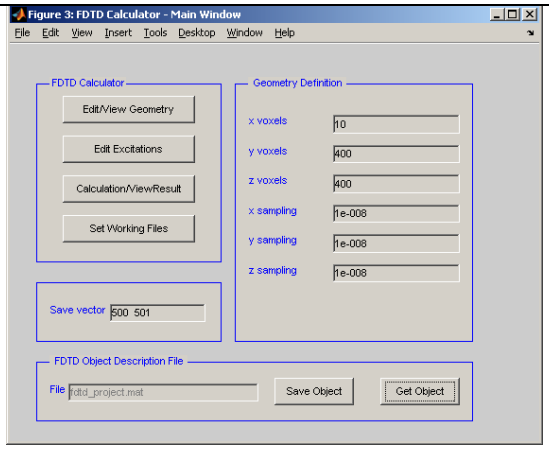
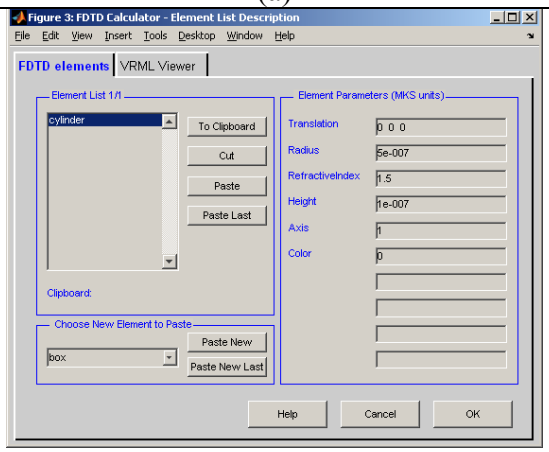


Figure 5.16. Results for Example 2 (xy plane at Location = 0). (a) Ex field magnitude; (b) Ey field magnitude; (c) Ez field magnitude; and (d) Irradiance (sum of square magnitudes for x, y and z fields).

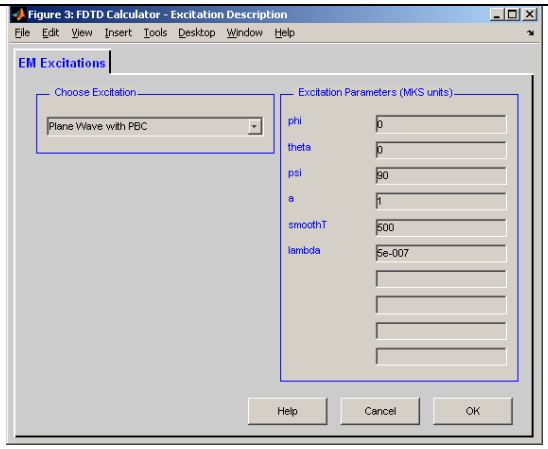
Example 3: NanoJet Cylinder with X- polarization (Periodic Boundary Conditions)



(a)



(b)



(d)

Figure 5.17. Panel description of Example 3: NanoJet Cylinder with X- polarization (PBC). (a) The calculation volume is 80 voxels on a side with 5nm sampling, for a total width of 400nm. The data are saved after 500 and 501 iterations; (b-c) The geometry consists of a box and a sphere. The sphere radius is 100nm with $n = 1.5$. It is shifted -200nm in the x direction. The box is a cube with 200nm sides and $n = 2.5$. It is shifted 200nm in the x direction. Both colors are set to 0, which will produce different colors in the VRML viewer file; and (d) The excitation is **Plane Wave with PBC**

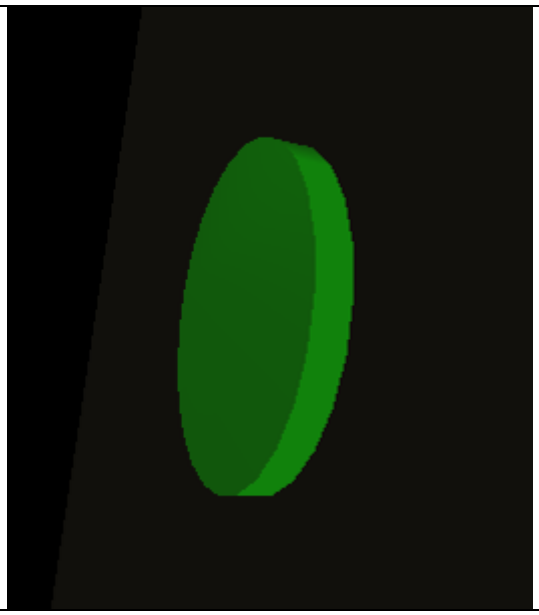


Figure 5.18. A Thin cylinder is used with periodic conditions in two directions in order to effectively extend the cylinder to infinity in the *x* direction. The excitation plane wave is from the *–z* direction and is *y* polarized.

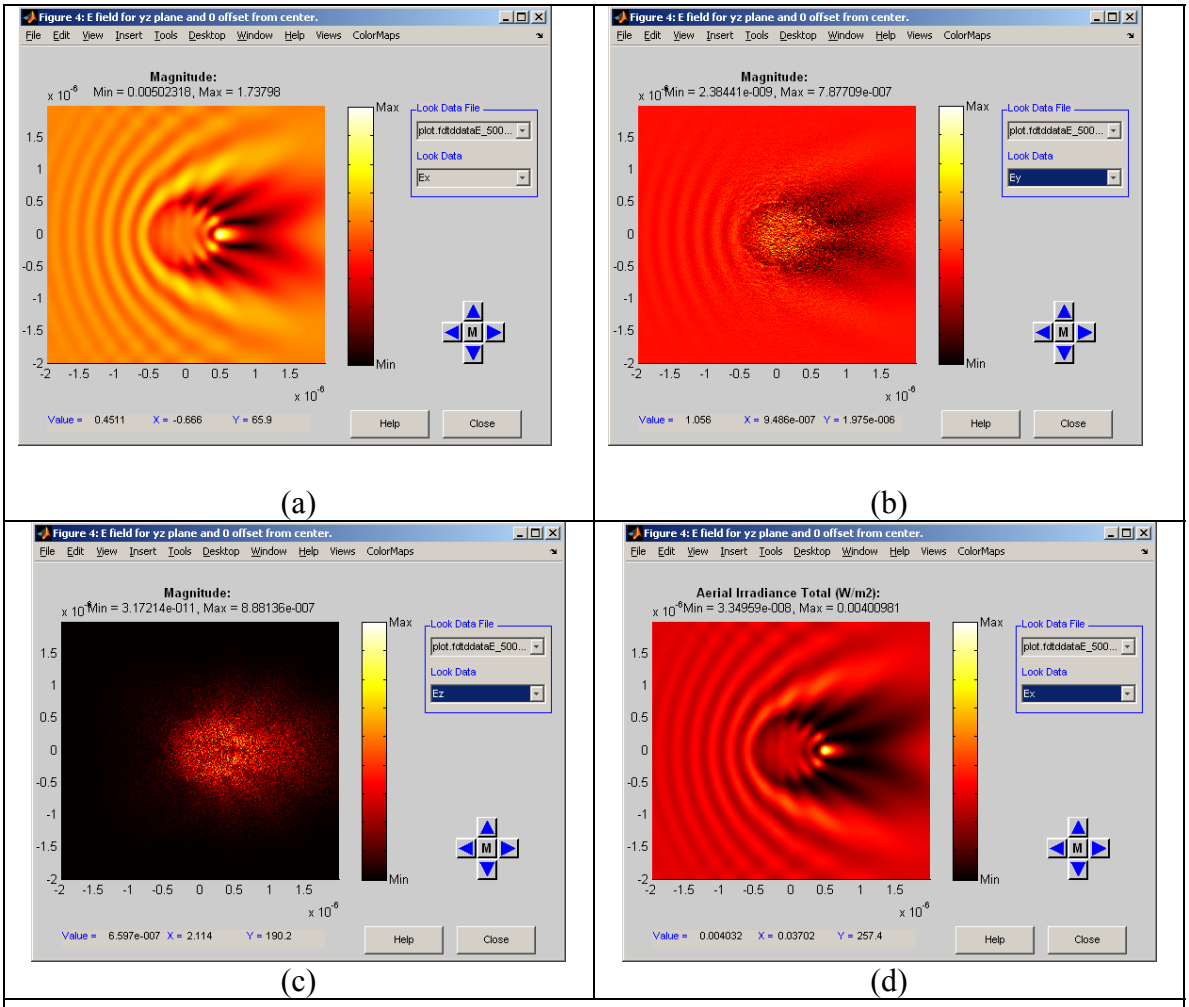
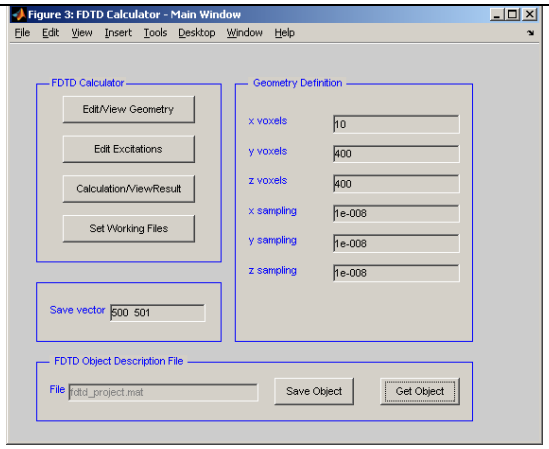
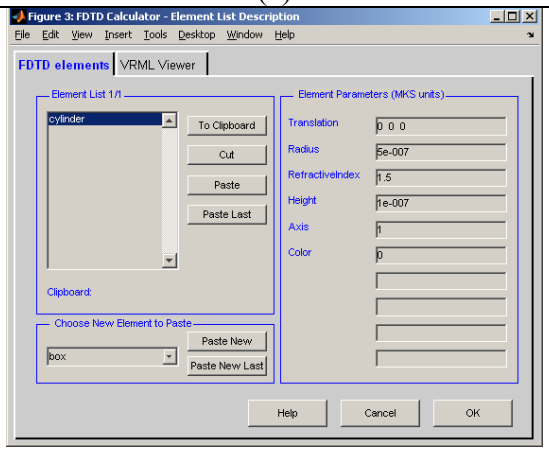


Figure 5.19. Results for Example 3 (yz plane at Location = 0). (a) Ex field magnitude; (b) Ey field magnitude (Almost zero); (c) Ez field magnitude (Almost zero); and (d) Irradiance (sum of square magnitudes for x, y and z fields).

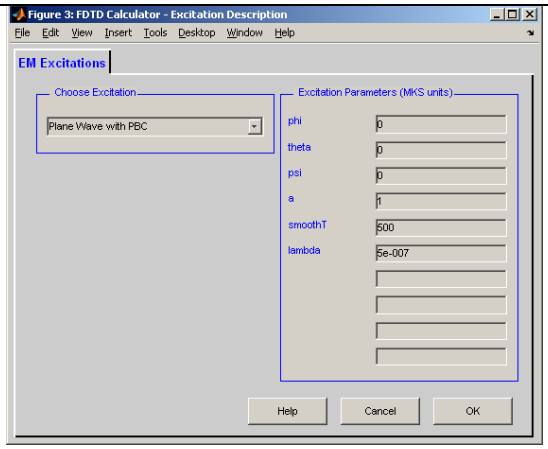
Example 4: NanoJet Cylinder with Y- polarization (Periodic Boundary Conditions)



(a)

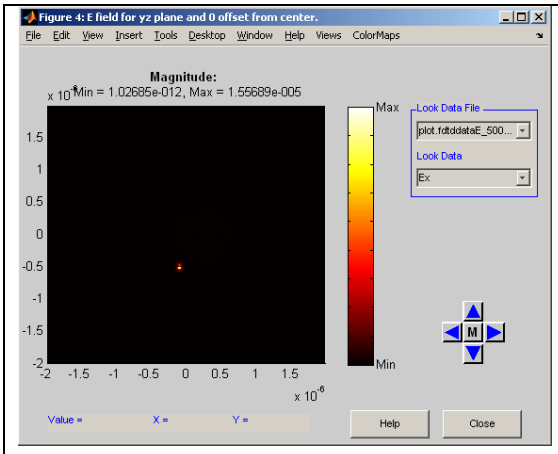


(b)

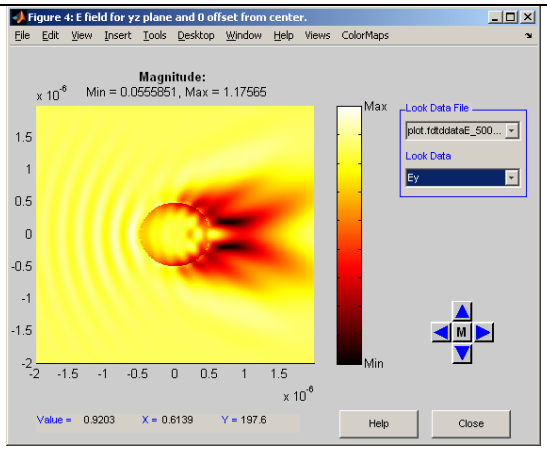


(d)

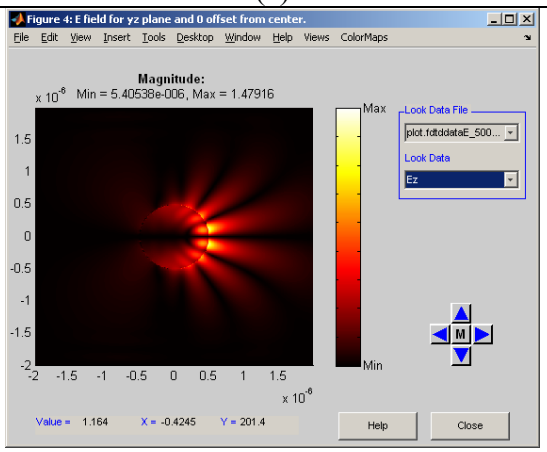
Figure 5.20. Panel description of Example 4: NanoJet Cylinder with Y- polarization (PBC). (a) The calculation volume is 10 x voxels, 400 y voxels, and 400 z voxels, with 10nm sampling. The data are saved after 500 and 501 iterations; (b-c) The geometry consists of a cylinder. The cylinder radius is 50nm with $n = 1.5$ (d) The excitation is **Plane Wave with PBC**.



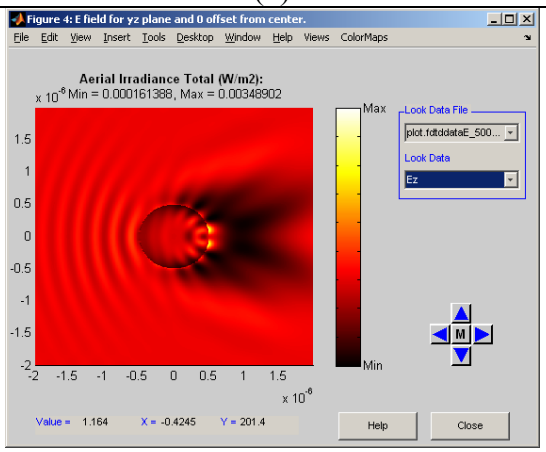
(a)



(b)



(c)



(d)

Figure 5.21. Results for Example 4 (yz plane at Location = 0). (a) Ex field magnitude(Almost zero); (b) Ey field magnitude; (c) Ez field magnitude (Almost zero); and (d) Irradiance (sum of square magnitudes for x, y and z fields).

Example 5: Au Rectangle with Y- polarization

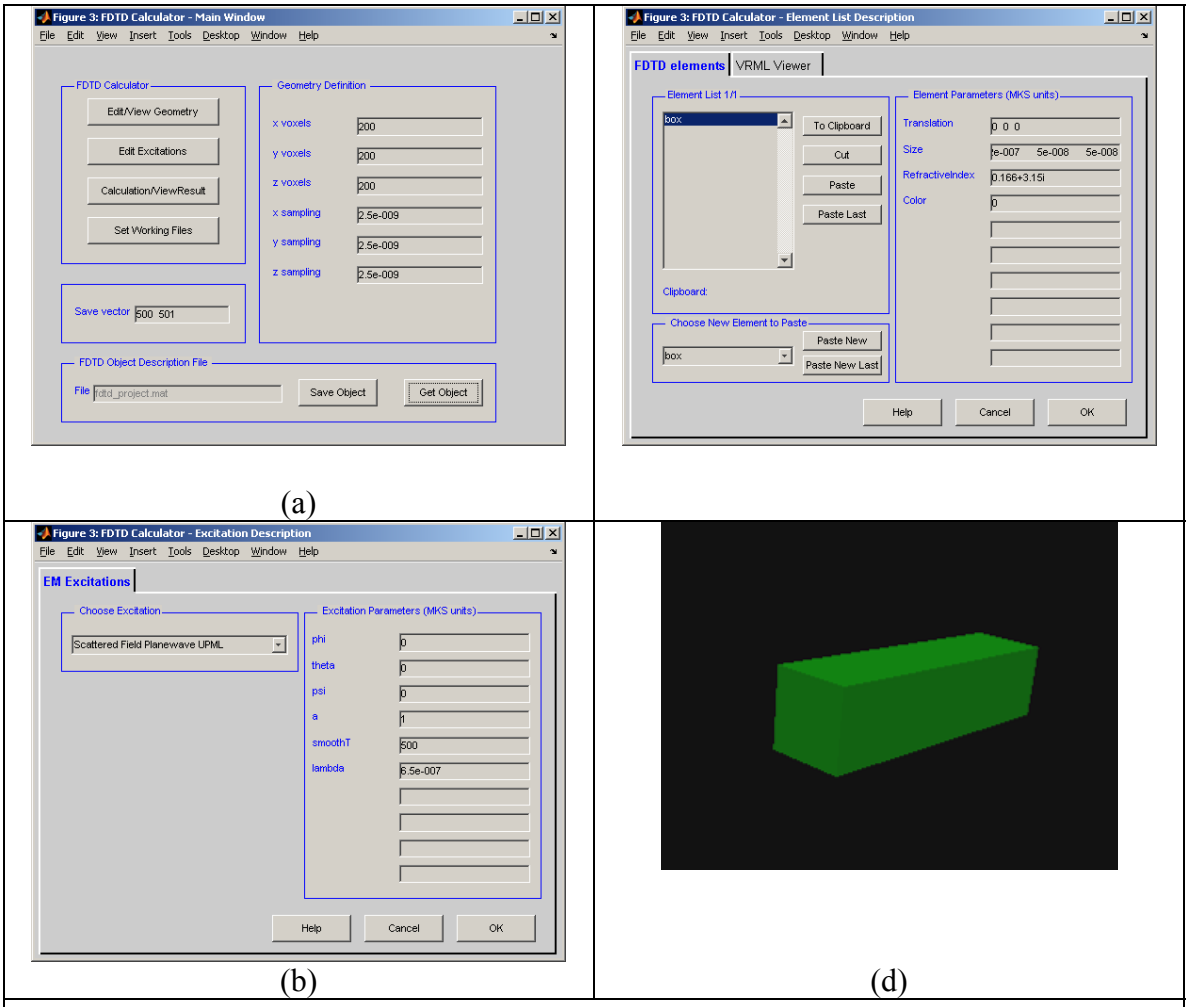


Figure 5.22. Panel description of Example 5: Au rectangle with Y polarization. (a) The calculation volume is 200 voxels on a side with 2.5nm sampling, for a total width of 500nm. The data are saved after 500 and 501 iterations; (b) The geometry consists of a box with a complex index of refraction. The box is a rectangle with 50nm sides in *y* and *z* and a 200nm length along *x*; (c) The Y polarization is set with psi = 0. The excitation is **Scattered Field PlaneWave UPML**; and (d) VRML view of the element in the calculation volume.

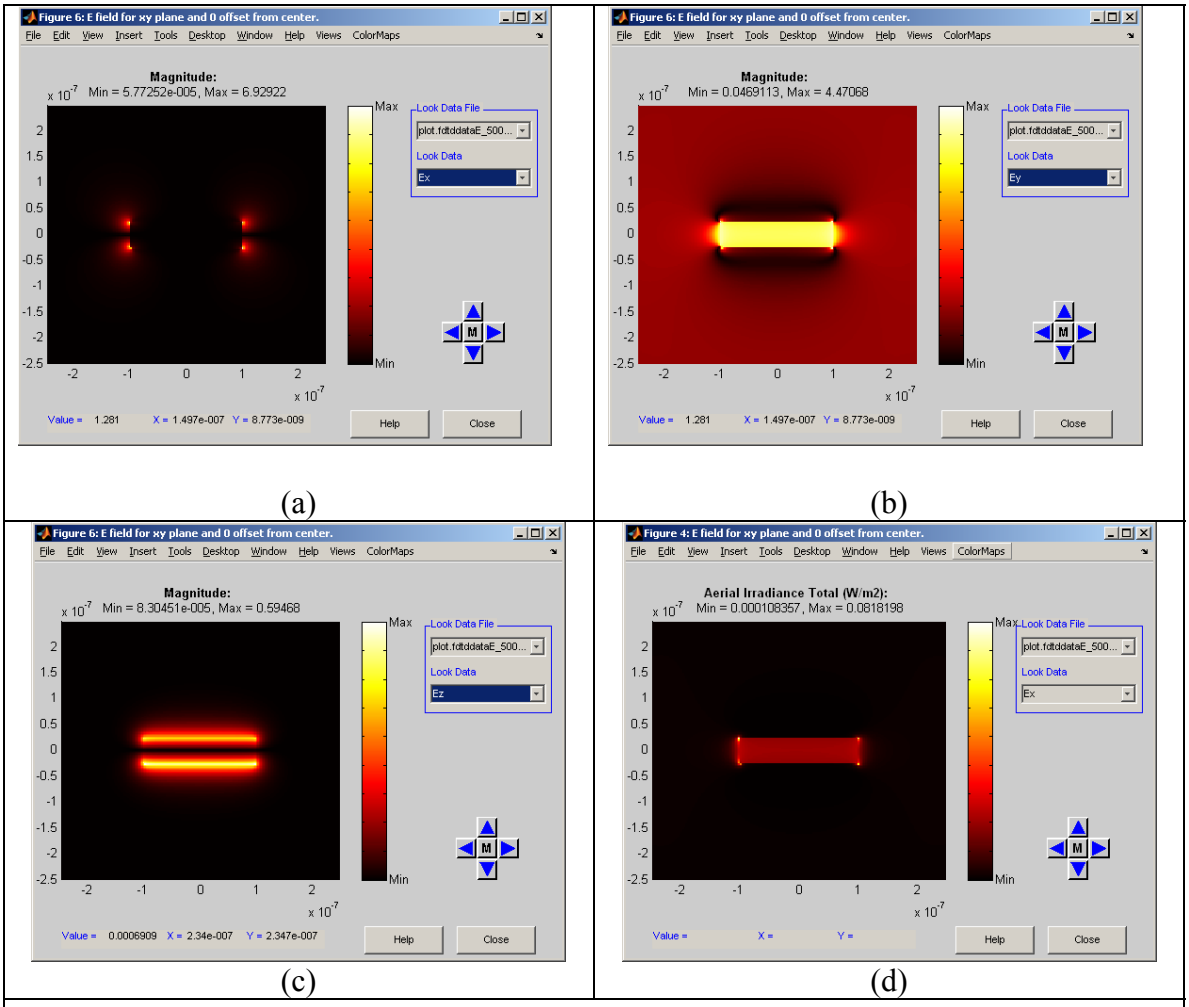


Figure 5.23. Results for Example 5 (xy plane at Location = 0). (a) Ex field magnitude; (b) Ey field magnitude; (c) Ez field magnitude; and (d) Irradiance (sum of square magnitudes for x, y and z fields).

Example 6: Au Rectangle with X - polarization

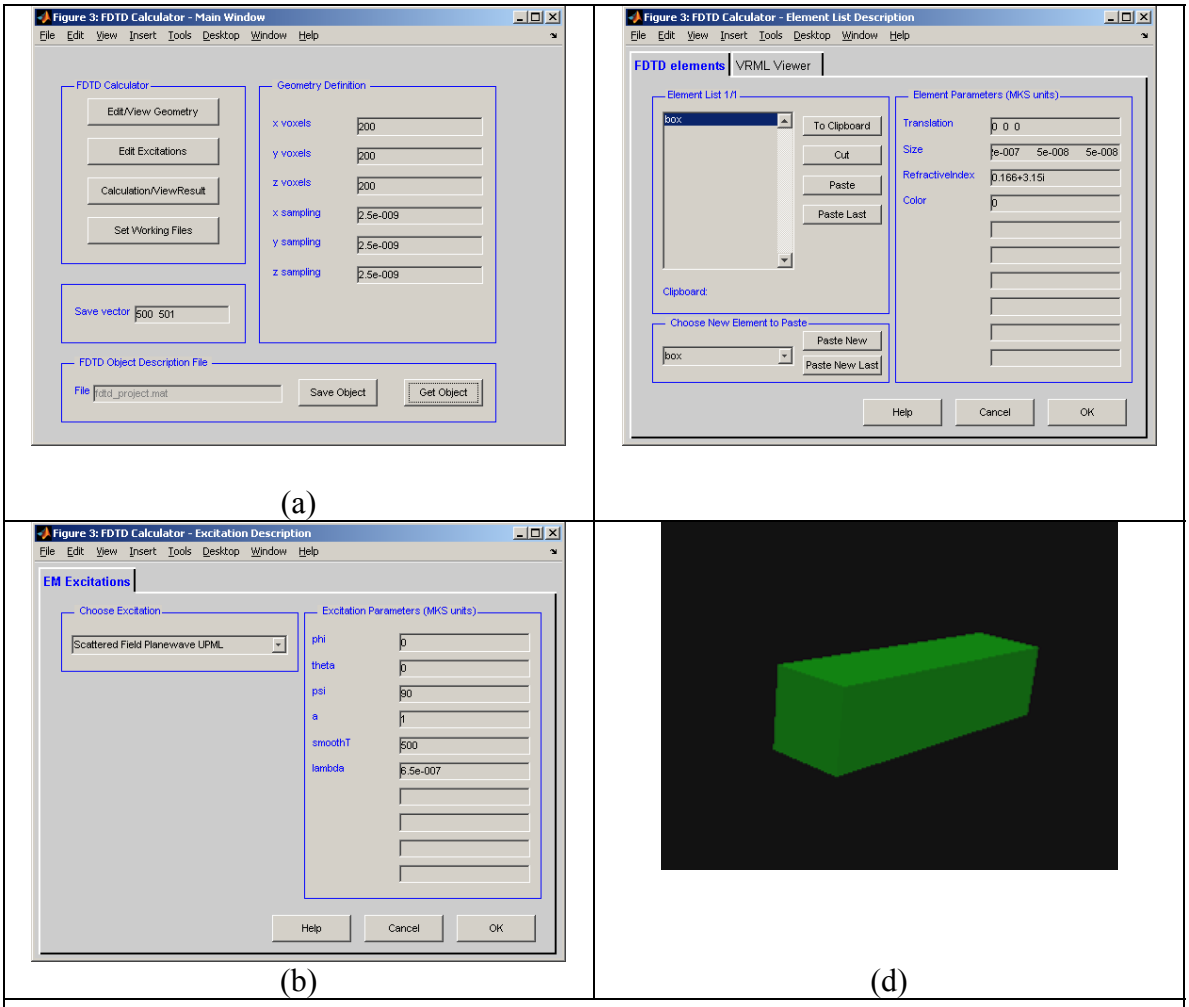


Figure 5.24. Panel description of Example 6: Au rectangle with X polarization. (a) The calculation volume is 200 voxels on a side with 2.5nm sampling, for a total width of 500nm. The data are saved after 500 and 501 iterations; (b) The geometry consists of a box with a complex index of refraction. The box is a cube with 50nm sides in *y* and *z* and a 200nm length along *x*; (c) The X polarization is set with psi = 90. The excitation is Scattered Field PlaneWave UPML; and (d) VRML view of the element in the calculation volume.

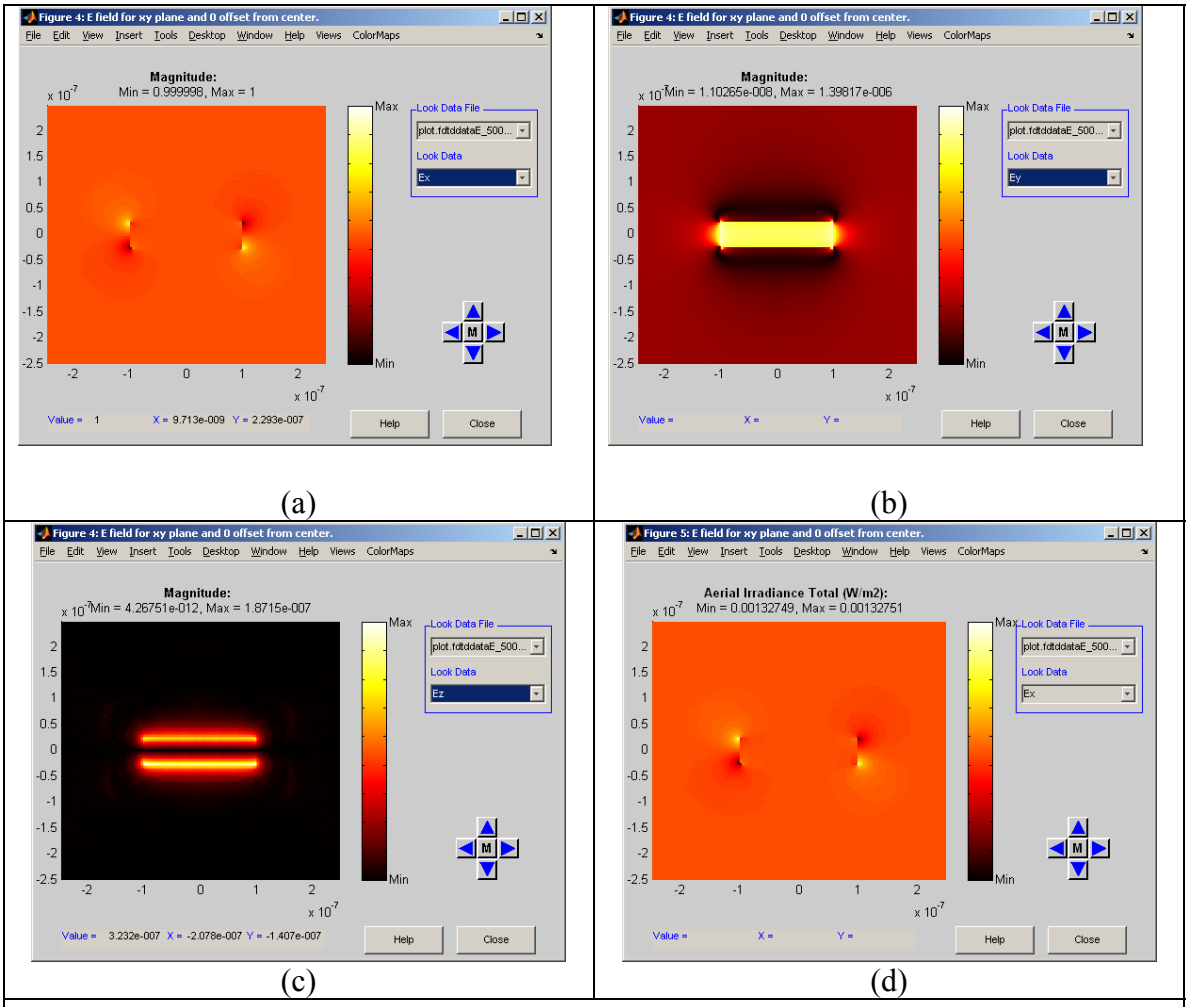


Figure 5.25. Results for Example 6 (xy plane at Location = 0). (a) Ex field magnitude; (b) Ey field magnitude (Almost zero); (c) Ez field magnitude (Almost zero); and (d) Irradiance (sum of square magnitudes for x, y and z fields).

Example 7: Hole in an Al Screen

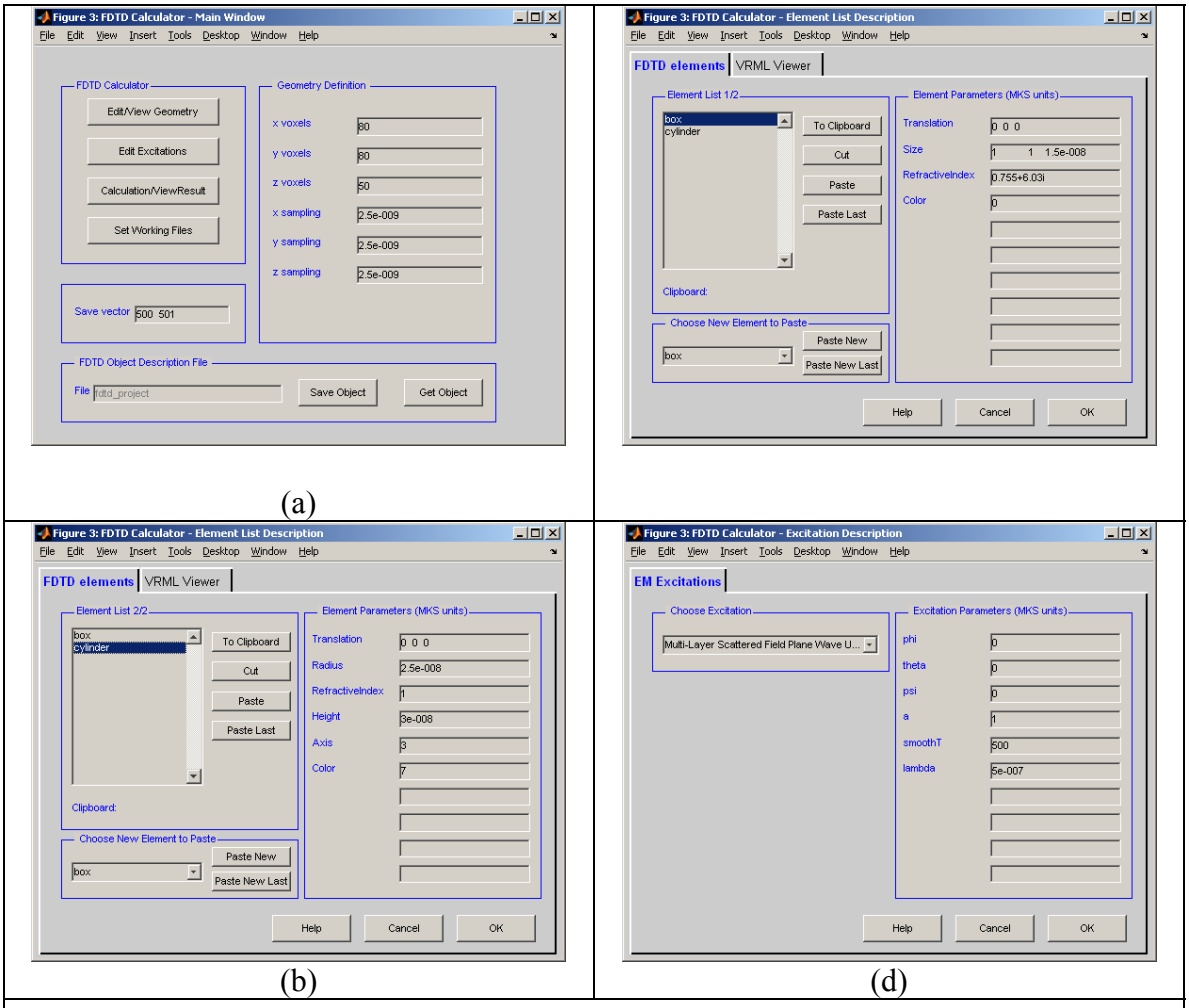


Figure 5.26. Panel description of Example 7: Hole in Al Screen. (a) The calculation volume is 80 x voxels, 80 y voxels, and 50 z voxels with a sampling of 2.5nm sampling, for a total width of 200nm. The data are saved after 500 and 501 iterations; (b-c) The geometry consists of an Al box and a cylinder. The box is very wide, which simulates the infinite layer, and it is 15nm thick. The cylinder radius is 25nm with $n = 1$, which is effectively making a hole in the Al; and (d) The excitation is **Multi-Layer Scattered Field Plane Wave UPML.**

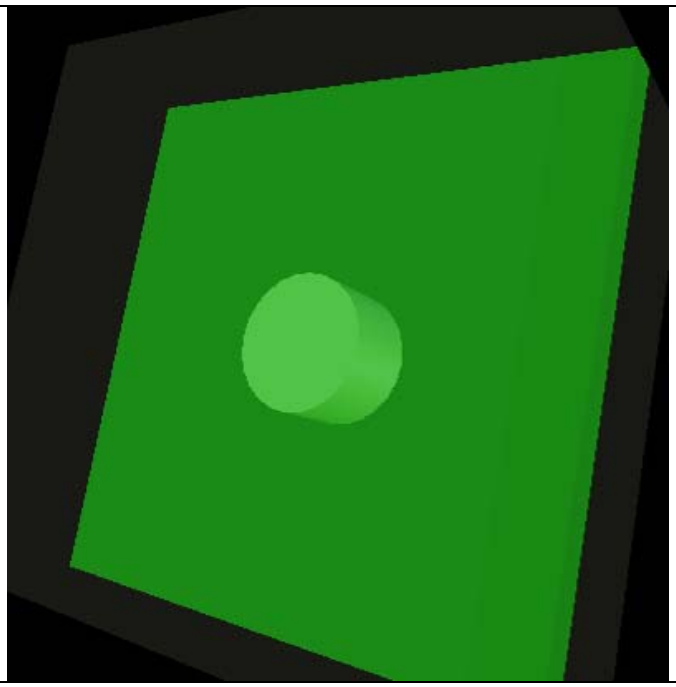


Figure 5.27. An air cylinder is used with an infinite Al screen The excitation plane wave is from the $-z$ direction and is y polarized.

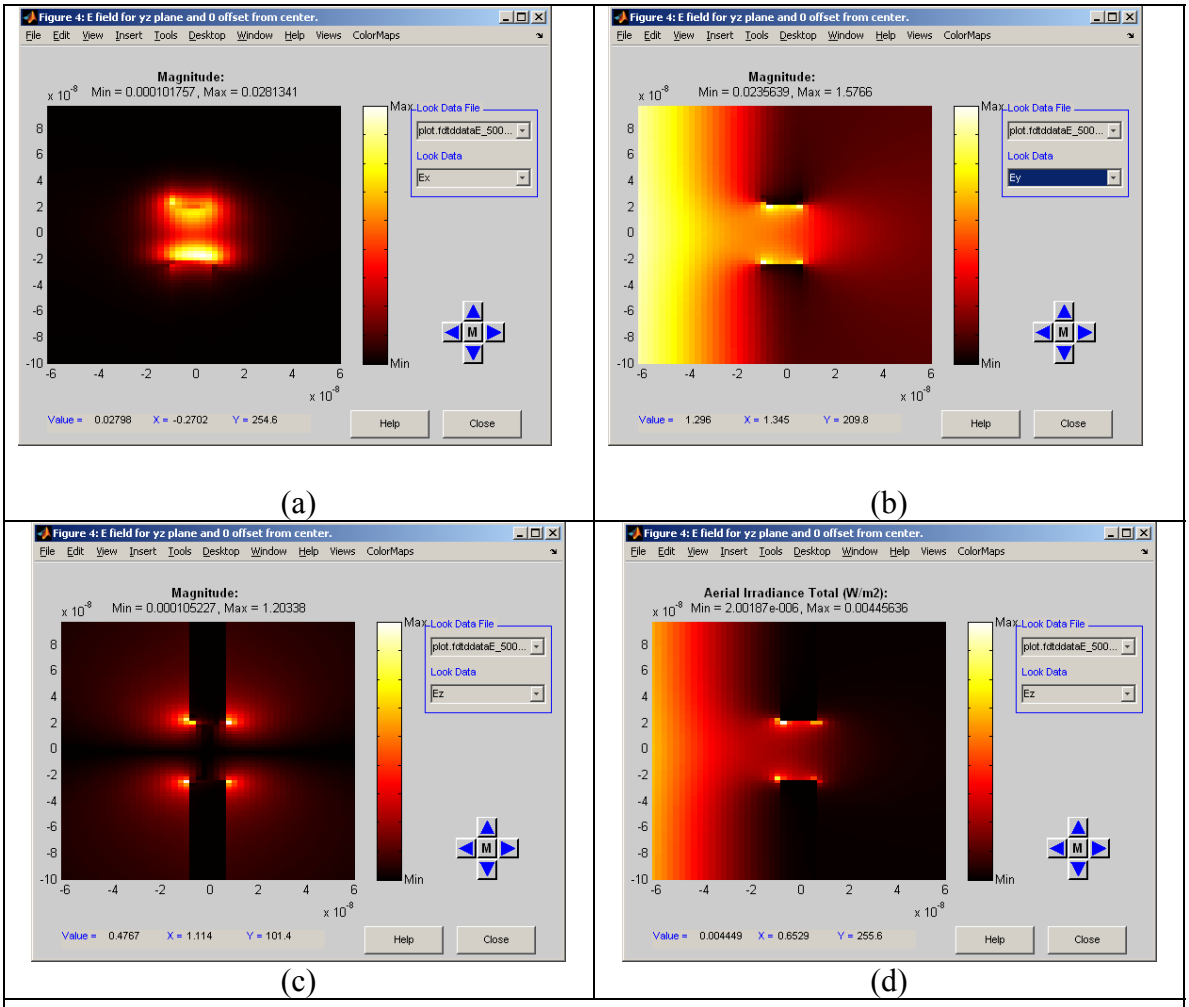


Figure 5.28. Results for Example 7 (yz plane at Location = 0). (a) Ex field magnitude; (b) Ey field magnitude; (c) Ez field magnitude; and (d) Irradiance (sum of square magnitudes for x, y and z fields).

Appendix A: Finite Difference Time Domain

As mentioned in the Introduction, Finite Difference Time Domain is based Kane Yee's algorithm for Maxwell's four curl equations. These equations in a linear medium are:

$$\nabla \times \mathbf{E} = \frac{-\partial \mathbf{B}}{\partial t} \tag{A1.1}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J} \tag{A1.2}$$

$$\nabla \cdot \mathbf{D} = \rho \tag{A1.3}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{A1.4}$$

where

$$\mathbf{D} = \varepsilon \mathbf{E} \tag{A1.5}$$

$$\mathbf{B} = \mu \mathbf{H} \tag{A1.6}$$

These four equations satisfy the information required for linear isotropic materials to completely specify the field behavior over time: contingent on the initial field distribution being specified and satisfying Maxwell's equations.

Rewriting these equations in a form more appropriate for computation, we have:

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu}\left(\nabla \times \mathbf{E}\right) - \frac{\sigma^*}{\mu}\mathbf{H} \tag{A1.7}$$

$$\frac{\partial \mathbf{E}}{\partial t} = -\frac{\mathbf{J}}{\varepsilon} + \frac{1}{\varepsilon}\left(\nabla \times \mathbf{H}\right) \tag{A1.8}$$

where J=σ**E** to allow for lossy dielectric material and have included the possibility for magnetic loss by adding a magnetic conductivity term σ^* .(Kunz*)

The two curl equations can then be discretized to obtain a total field FDTD technique:

$$\mathbf{E} = \mathbf{E}^{total} = \mathbf{E}^{incident} + \mathbf{E}^{scattereed}$$
$$\mathbf{H} = \mathbf{H}^{total} = \mathbf{H}^{incident} + \mathbf{H}^{scattered} \qquad \textbf{(}\text{A1.9}\textbf{)}$$

The separate field approach is essential to the FDTD theory. Without discretizing the curl equations, viewing the scattered field alone would be near impossible: especially in the situations where the scattered fields are of much lower amplitude than the total fields. Another advantage of the separate field approach, versus other methods, is that it is always possible to combine the incident and scattered fields to obtain the total field and all the insight the total field behavior provides.

The incident field is defined as the field that would exist in the absence of the scatterer; the incident field always propagates in free space, even when passing through the interaction object of scatterer material. In UAFDTD, and in general the FDTD method, the incident field is specified by the user, and therefore analytical. In contrast, the scattered fields are found computationally. Only scattered fields need to be absorbed at the outer boundaries of the problem space. The scattered wave responds to the incident wave on and within the interaction object so as to satisfy Maxwell's equations, which become boundary conditions on or within the interaction objects: In the limit of a perfect conductor $\mathbf{E}^{scattered} = -\mathbf{E}^{incident}$ in the scatterer. For all the other objects the scattered fields depend on the defining parameters of the material (See Specifying Geometries). In FDTD space all media is described by Maxwell's Equations therefore all scattered fields are subject to these equations while inside the media and subject to the free space Maxwell's Equations while outside the media (Kunz 13).

Using separate field formalism, we take a closer look at Maxwell's equations to view the incident and scattered fields:

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \tag{A1.10}$$

$$\nabla \times \mathbf{H} = \varepsilon \frac{\partial \mathbf{E}}{\partial t} \tag{A1.11}$$

let

$$\mathbf{E} = \mathbf{E}^{total} = \mathbf{E}^{incident} + \mathbf{E}^{scattereed}$$
$$\mathbf{H} = \mathbf{H}^{total} = \mathbf{H}^{incident} + \mathbf{H}^{scattered}$$

using the above substitution and subtracting the incident fields, we obtain the equations governing the scattered fields in media.

$$\nabla \times \mathbf{E}^{scattered} = -\mu \frac{\partial \mathbf{H}^{scattered}}{\partial t} - \sigma^* \mathbf{H}^{scattered} - \left[\left(\mu - \mu_0 \right) \frac{\partial \mathbf{H}^{incident}}{\partial t} + \sigma^* \mathbf{H}^{incident} \right] \tag{A1.12}$$

$$\nabla \times \mathbf{H}^{scattered} = \varepsilon \frac{\partial \mathbf{E}^{scattered}}{\partial t} + \sigma \mathbf{E}^{scattered} + \left[\left(\varepsilon - \varepsilon_0 \right) \frac{\partial \mathbf{E}^{incident}}{\partial t} + \sigma \mathbf{E}^{incident} \right] \tag{A1.13}$$

with $\mu, \varepsilon, \sigma^*$, and σ inside the scatterer, and with $\sigma^* = \sigma = 0, \mu = \mu_0,$ and $\varepsilon = \varepsilon_0$ outside the scatterer, are determined computationally. This is the only set of equations needed for separate field formalism. The incident fields are given analytically, as stated above, and must therefore be Maxwellian in form in order for them to be viable in the program.

As expected, the equations governing the total field outside the scatterer in free space are:

$$\nabla \times \mathbf{E}^{total} = -\mu_0 \frac{\partial \mathbf{H}^{total}}{\partial t} \tag{A1.14}$$

$$\nabla \times \mathbf{H}^{total} = \varepsilon_0 \frac{\partial \mathbf{E}^{total}}{\partial t} \tag{A1.15}$$

The separate field equations above can now be rearranged so that the time derivative of the field is expressed as a function of the remaining terms. This technique is most straightforward when beginning with the simple situation of a perfect conductor with free space outside the scatterer, where $\sigma^* = \sigma = 0, \mu = \mu_0,$ and $\varepsilon = \varepsilon_0$. First we begin with rewriting equations (A1.12) and (A1.13)

$$\frac{\partial \mathbf{H}^{scattered}}{\partial t} = -\frac{\sigma^*}{\mu}\mathbf{H}^{scattered} - \frac{\sigma^*}{\mu}\mathbf{H}^{incident} - \frac{(\mu - \mu_0)}{\mu}\frac{\partial \mathbf{H}^{incident}}{\partial t} - \frac{1}{\mu}\left(\nabla \times \mathbf{E}^{scattered}\right) \quad \text{(A1.16)}$$

$$\frac{\partial \mathbf{E}^{scattered}}{\partial t} = -\frac{\sigma}{\varepsilon}\mathbf{E}^{scattered} - \frac{\sigma}{\varepsilon}\mathbf{E}^{incident} - \frac{(\varepsilon - \varepsilon_0)}{\varepsilon}\frac{\partial \mathbf{E}^{incident}}{\partial t} + \frac{1}{\varepsilon}\left(\nabla \times \mathbf{H}^{scattered}\right) \quad \text{(A1.17)}$$

In the case of the perfect conductor, equation (A1.17) governing the scattered field may be written as

$$\frac{\varepsilon}{\sigma}\frac{\partial \mathbf{E}^{scattered}}{\partial t} = -\mathbf{E}^{scattered} - \mathbf{E}^{incident} - \frac{(\varepsilon - \varepsilon_0)}{\sigma}\frac{\partial \mathbf{E}^{incident}}{\partial t} + \frac{1}{\sigma}\left(\nabla \times \mathbf{H}^{scattered}\right) \quad \text{(A1.18)}$$

for the free space conditions $\sigma^* = \sigma = 0, \mu = \mu_0,$ and $\varepsilon = \varepsilon_0$, we have the following equations

$$\frac{\partial \mathbf{H}^{scattered}}{\partial t} = -\frac{1}{\mu_0}\left(\nabla \times \mathbf{E}^{scattered}\right) \quad \text{(A1.19)}$$

$$\frac{\partial \mathbf{E}^{scattered}}{\partial t} = \frac{1}{\varepsilon_0}\left(\nabla \times H^{scattered}\right) \quad \text{(A1.20)}$$

For a perfect conductor $\sigma = \infty$, and for this situation equation (A1.18) reduces to

$$\mathbf{E}^{scattered} = -\mathbf{E}^{incident} \tag{A1.21}$$

We apply this equation at the boundary between the perfect conductor and free space: all other interior portions of the perfect conductor are completely isolated from the rest of the problem space. Having equations (A1.19), (A1.20) for the scattered field and the relation equation (A1.21), only an incident field is needed to apply the FDTD technique.

The next task is to difference the free space scattered field equations. In other words, differencing replaces derivatives with differences:

$$\frac{\partial f}{\partial t} \equiv \lim_{\Delta t \to 0} \frac{f(x,t_2) - f(x,t_1)}{\Delta t} \approx \frac{f(x,t_2) - f(x,t_1)}{\Delta t} \tag{A1.22}$$

$$\frac{\partial f}{\partial x} \equiv \lim_{\Delta t \to 0} \frac{f(x_2,t) - f(x_1,t)}{\Delta x} \approx \frac{f(x_2,t) - f(x_1,t)}{\Delta x} \tag{A1.23}$$

Note: To ensure stability Δx and Δt must be bounded in order to ensure numerical stability. where Δt is given by the Courant stability criterion, $\Delta t \le \frac{\Delta x}{c\sqrt{3}}$.

Breaking down A1.22 and A1.23 into their scalar parts, we have

$$\frac{\partial E_x^{scat}}{\partial t} = \frac{1}{\varepsilon_0}\left(\frac{\partial H_z^{scat}}{\partial y} - \frac{\partial H_y^{scat}}{\partial z}\right) \tag{A1.24}$$

$$\frac{\partial E_y^{scat}}{\partial t} = \frac{1}{\varepsilon_0}\left(\frac{\partial H_x^{scat}}{\partial z} - \frac{\partial H_z^{scat}}{\partial x}\right) \tag{A1.25}$$

$$\frac{\partial E_z^{scat}}{\partial t} = \frac{1}{\varepsilon_0}\left(\frac{\partial H_y^{scat}}{\partial x} - \frac{\partial H_x^{scat}}{\partial y}\right) \tag{A1.26}$$

$$\frac{\partial H_x^{scat}}{\partial t} = \frac{1}{\mu_0}\left(\frac{\partial E_y^{scat}}{\partial z} - \frac{\partial E_z^{scat}}{\partial y}\right) \tag{A1.27}$$

$$\frac{\partial H_y^{scat}}{\partial t} = \frac{1}{\mu_0}\left(\frac{\partial E_z^{scat}}{\partial x} - \frac{\partial E_x^{scat}}{\partial z}\right) \tag{A1.28}$$

$$\frac{\partial H_z^{scat}}{\partial t} = \frac{1}{\mu_0}\left(\frac{\partial E_x^{scat}}{\partial y} - \frac{\partial E_y^{scat}}{\partial x}\right) \tag{A1.29}$$

For simplicity, we will only treat the pair E_x^{scat} and H_y^{scat}, the other components follow naturally. Replacing derivatives with differences we find:

$$\frac{E_x^{scat,n} - E_x^{scat,n-1}}{\Delta t} = \frac{1}{\varepsilon_0}\left[\frac{\Delta H_z^{scat,n-\frac{1}{2}}}{\Delta y} - \frac{\Delta H_y^{scat,n-\frac{1}{2}}}{\Delta z}\right] \tag{A1.30}$$

$$\frac{H_y^{scat,n+\frac{1}{2}} - H_y^{scat,n-\frac{1}{2}}}{\Delta t} = \frac{1}{\mu_0}\left[\frac{\Delta E_z^{scat,n}}{\Delta x} - \frac{\Delta E_y^{scat,n}}{\Delta z}\right] \tag{A1.31}$$

This completes the perfect conductor separate field formulation. For more information FDTD please refer to *The Finite Difference Time Domain Method for Electromagnetics* by Kunz and Leubbers; and *Computational Electrodynamics: The finite difference time domain method* by Taflove and Hagness.

Appendix B: Yee Cell and Courant Stability Criterion

As mentioned in the Introduction, Kane Yee developed a set of finite difference equations for the time dependent Maxwell's curl equations in 1966. Yee's Finite Difference Time Domain Method solves for both electric and magnetic fields in time and space using the coupled Maxwell's curl equations rather than solving for the electric or magnetic field alone with a wave equation (see Apendix A).

As illustrated in Figure B1, the Yee algorithm centers its $\vec{E}$ and $\vec{H}$ components in three-dimensional space so that every $\vec{E}$ component is surrounded by four circulating $\vec{H}$ components, and every $\vec{H}$ component is surrounded by four circulating $\vec{E}$ components.

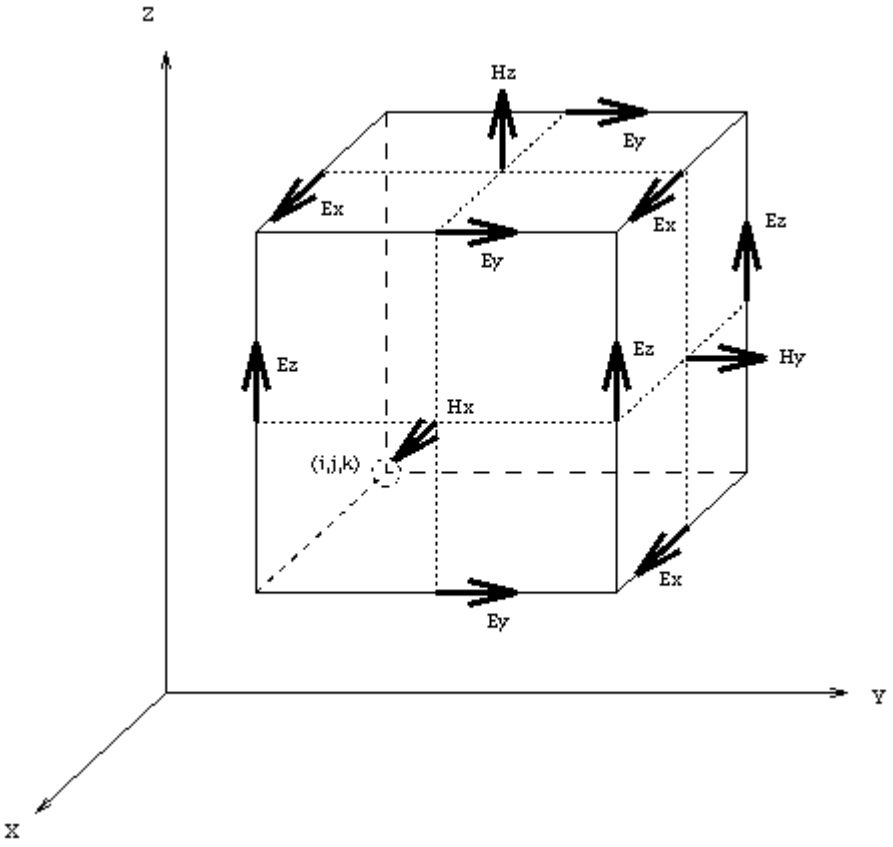


Figure B1: Position of electric and magnetic field components in Yee cell.

In the cubic unit cell of the Yee space lattice, the continuity of tangential $\vec{E}$ and $\overrightarrow{H}$ fields are naturally maintained across an interface of dissimilar materials if the interface is parallel to one of the lattice coordinate axes (which is always the case with UAFDTD). The location of the $\vec{E}$ and $\overrightarrow{H}$ components in the Yee cell and the central-difference operations on these components implicitly enforce the two Gauss' Law relations(see Taflove Chapter 2). Thus the Yee (UAFDTD) mesh is divergence-free with respect to its **E** and **H** fields in the absence of free electric and magnetic charge.

As illustrated in Figure B2, the Yee algorithm also centers its $\vec{E}$ and $\overrightarrow{H}$ components in time in what is termed as a leapfrog arrangement. All the $\vec{E}$ computations in the simulation space are calculated and stored in memory for a particular time point using previously stored $\overrightarrow{H}$ data. Then all of the $\overrightarrow{H}$ computations in the space are calculated and stored in memory using the $\vec{E}$ data just computed. The cycle begins again with the recomputation of the $\vec{E}$ components based on the newly obtained $\overrightarrow{H}$ components. This process continues until time-stepping is concluded.

The advantages to using this Leapfrog time-stepping technique is that it is fully explicit, thereby avoiding problems involved with simultaneous equations and matrix inversion. Because the finite-difference expressions for the time derivatives are central difference in nature and second-order accurate (see Appendix A), the time-stepping algorithm is nondissipative. That is, numerical wave modes propagating in the mesh do not spuriously decay due to a nonphysical artifact of the time-stepping algorithm (Taflove Chapter 2.5).

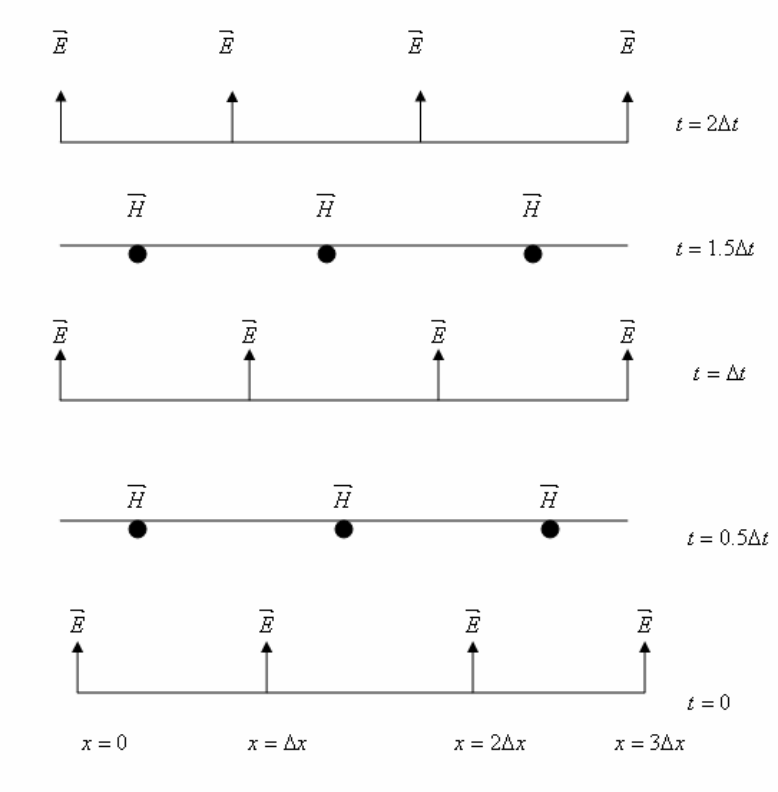


Figure B2: Space-time chart of the Yee algorithm for a one-dimensional wave propagation example showing the use of central differences for the space derivatives and leapfrog for the time derivatives. Initial conditions for both electric and magnetic fields are zero everywhere in the grid.

Appendix C: Cell size and Courant Stability Condition

The choice of cell size is critical in applying the FDTD technique. The cell size must be small enough to permit accurate results at the highest frequency of interest, and yet be large enough to keep resource requirements manageable. Cell size is directly affected by the materials present. The greater the permittivity or conductivity, the shorter the wavelength at a given frequency and the smaller the cell size required. Please note that UAFDTD is a volumetric computational software, so that if some portion of the computational work space is filled with frequency dependent material the wave length in the material must be used to determine the maximum cell size. For problems containing electrically dense materials this results in using cell sizes that are much smaller than if only free space were being considered.

The fundamental constraint when selecting the cell size is that it must be much less than the smallest wavelength. As a rule of thumb there should be no more than 10 cells per wavelength, meaning that the side of each cell (in UAFDTD this is listed as x sampling, y sampling and z sampling) should be no more than $1/10\lambda$. The cell size has an upper limit defined by the Nyquist sampling limit of $\lambda=2\Delta x$

To understand why the cell size must be much smaller than one wavelength, consider that at any particular time step the FDTD grid is a discrete spatial sample of the field distribution. From the Nyquist sampling theorem, there must be at least two samples per wavelength, as shown in Figure B2 in Appendix B, in order for the spatial information to be adequately sampled. Because sampling is often not exact, and the smallest wavelength is usually not precisely determined, more than two samples per wavelength are required. Another related consideration is grid dispersion error. Due to the approximations inherent in FDTD, waves of different frequencies will propagate at slightly different speeds through the work space. This difference in propagation speed also depends on the direction of propagation relative to the grid. For accurate and stable results, the grid dispersion error must be reduced to an acceptable level, which can be readily accomplished by reducing the cell size.

Another cell size consideration is that the important features of the problem geometry must be accurately modeled (see also Appendix D). Normally this will be met

automatically by making the cells smaller than 1/10λ or so, unless some special geometry characteristic smaller than this is a factor in determining the cell size.

Appendix D: Subcell Model

The most commonly used method for the time-domain Maxwell equations is the Finite-Difference Time-Domain method (FDTD). This is an explicit, second-order accurate method, which is used on a staggered Cartesian grid (see Appendix A). The main drawback with the FDTD method is its inability to accurately model curved objects and small geometrical features. This is due to the Cartesian grid, which leads to a staircase approximation of the geometry and small details are not resolved at all.

An approach to circumvent this drawback, but still take advantage of the benefits of the FDTD method and avoid staircasing errors but still retain the efficiency of the FDTD method is to use local subcells. The ability to model features that are small relative to the cell size is often important in electromagnetic simulations. The subcell method gives considerable modeling flexibility compared to earlier methods and is proven stable. The results show excellent consistency and very good accuracy on different electromagnetic simulations with curved structures.

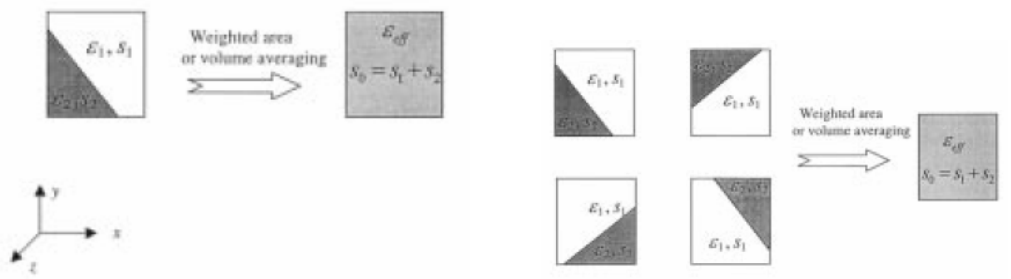
Recent FDTD studies have shown that stepped approximations of curved surfaces can shift center frequencies of resonant responses and can possibly introduce spurious null. The contour path (CP) method can be applied where the local grid cells are deformed due to the curved surfaces. Among several types of FDTD conformable surface models, *locally distorted grid models* is the most suitable type. These preserve the basic Cartesian grid arrangement of field components at all space cells except those immediately adjacent to the structure surface. Space cells adjacent to the structure surface are deformed to conform to the surface locus. Space cells not near to the media interface follows traditional FDTD algorithm which is a direct solution of Maxwell's time dependent curl equations (see Appendix A). Space cells adjacent to the media interface are calculated by the CP algorithm, which is based on Ampere's and Faraday's laws, shown below:

$$\oint_c \vec{H}\cdot\vec{dl} = \iint_s (\vec{J_c}+\vec{J_s})\cdot\vec{dS} + \frac{\partial}{\partial t}\iint_s (\vec{J_c}+\vec{J_s})\cdot\vec{dS}$$

$$\oint_c \vec{E}\cdot\vec{dl} = -\frac{\partial}{\partial t}\iint_s \vec{B}\cdot\vec{dS}$$

where $\vec{E},\vec{D},\vec{H},\vec{B},\vec{J}_s,\vec{J}_c$ denote the electric field, electric flux density, magnetic field, magnetic flux field, conduction current and source current, respectively and the C contours enclose the S surfaces.

Typically, previous conformal dielectric FDTD algorithms employ the weighted area or volume average to deal with the cells filled with different materials, as shown in figure 1(a). However this approach yield the same effective value for dielectric distributions, even when the geometry of the fillings are different, as depicted in figure 1(b).



(a) (b)

Figure1. (a) Conformal dielectric technique. (b) Different material distributions lead to the same effective dielectric constant.

The Yu-Mittra technique utilize a linear average concept and effective dielectric constants, $\varepsilon_x^{eff}(i, j, k)$ and $\varepsilon_y^{eff}(i, j, k)$ as defined in below,

$$\varepsilon_x^{\,eff} = \left(\Delta x_2(i,j,k)\cdot\varepsilon_2 + \left(\Delta x - \Delta x_2(i,j,k)\right)\cdot\varepsilon_1\right)/\Delta x(i,j,k)$$
$$\varepsilon_y^{\,eff} = \left(\Delta y_2(i,j,k)\cdot\varepsilon_2 + \left(\Delta y - \Delta y_2(i,j,k)\right)\cdot\varepsilon_1\right)/\Delta y(i,j,k)$$

The update equations for the electric and magnetic fields are thus written as,

$$E_x^{\,n+1}(i,j+1,k) = E_x^{\,n}(i,j+1,k) + \frac{\Delta t}{\varepsilon_1(i,j+1,k)\cdot\Delta y}\cdot\left\{H_z^{\,n+1/2}(i,j+1,k) - H_z^{\,n-1/2}(i,j,k)\right\}$$
$$-\frac{\Delta t}{\varepsilon_1(i,j+1,k)\cdot\Delta z}\cdot\left\{H_y^{\,n+1/2}(i,j+1,k) - H_y^{\,n-1/2}(i,j,k)\right\}$$

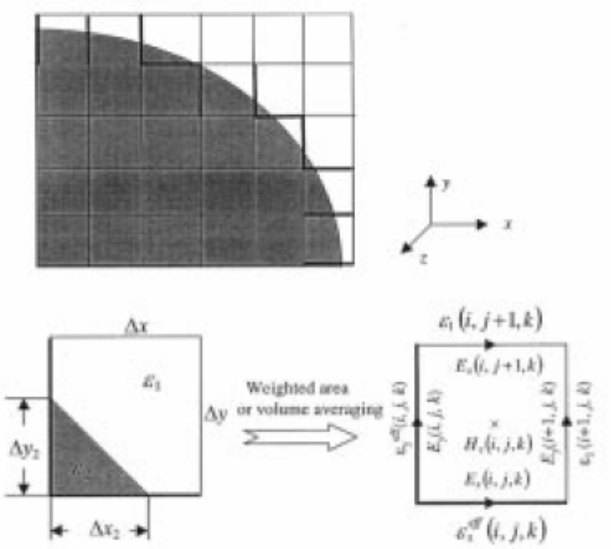


Figure 2. Linear weighed dielectric constant averaging procedure.

While the program is generating geometrical matrix, each cell is checked if it is located at the edges. If it is, the UAFDTD calculates parameters such as positions, type of filling, refractive indices and truncated sizes of cell. Therefore each object generation function has to be modified to carry out this task.