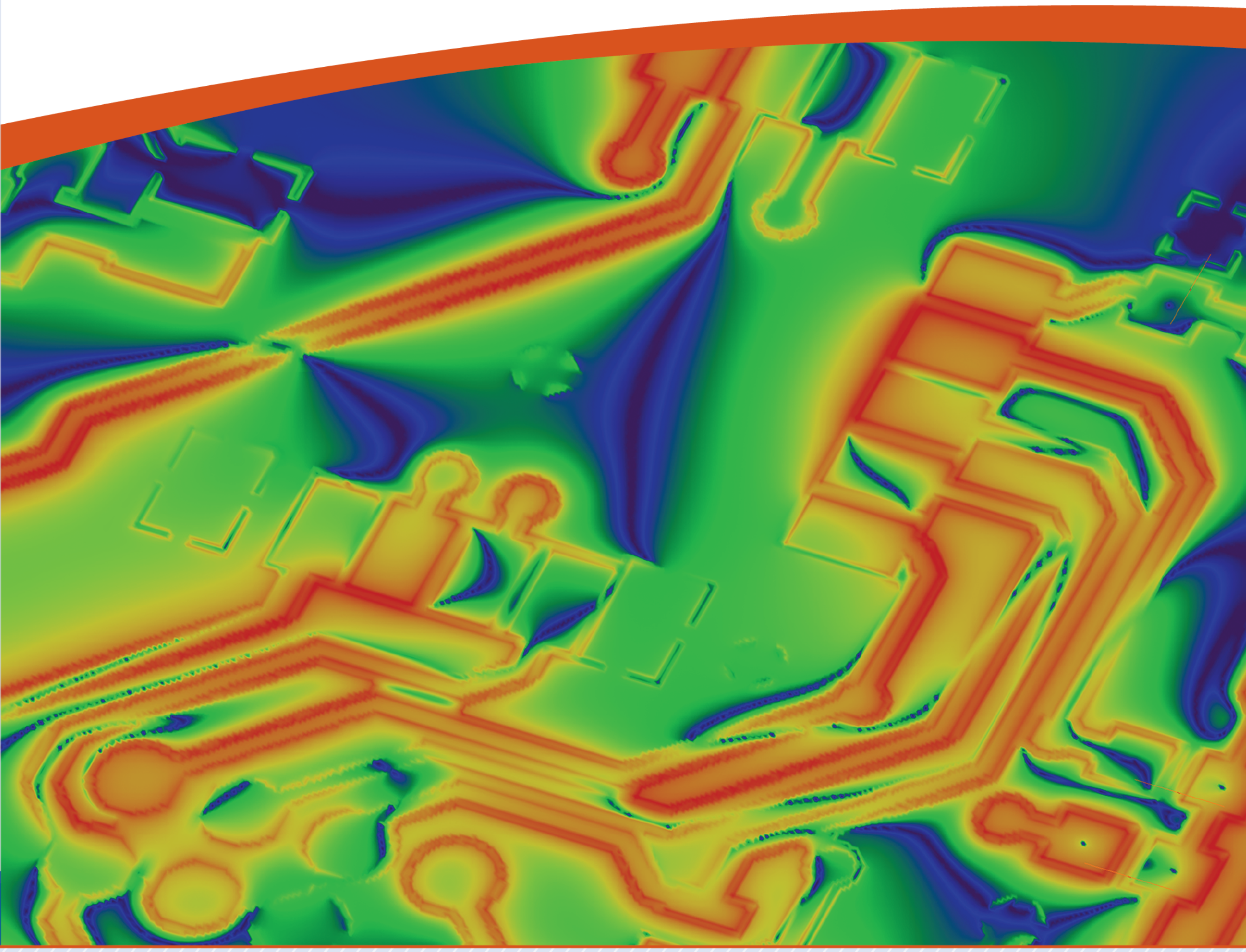
XFdtd

Full-wave, 3D, Electromagnetic Analysis Software

REFERENCE MANUAL



REMCOM®

Electromagnetic Simulation Solutions

XFdtd® Reference Manual

Release 7.3.0

Remcom Inc.
315 S. Allen St., Suite 416
State College, PA 16801

+1.888.7.REMCOM
+1.814.861.1299
+1.814.861.1308 *fax*
www.remcom.com

XFdtd is versioned with four numbers as *major.minor.feature.bugfix*. A change in any number except the *bugfix* number indicates that new features have been added to the software.

This document has been updated for, prepared and delivered with XFdtd 7.3.0.4, November 2012.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to XFdtd® Release 7

XFdtd 7, also known as XF7, is a three-dimensional full-wave electromagnetic solver based on the Finite-Difference Time-Domain method. Since 1994, Remcom has provided powerful yet affordable electromagnetic simulation software to academia, industry and government users. XFdtd 7 is an exceptional addition to our software product line with many new features. While XFdtd has its roots in the simulation of cellular telephones, current uses for the software have reached markets as diverse as chemistry, optics, ground-penetrating radar, and biomedical devices, in addition to wireless, microwave circuit and radar scattering applications.

1.1 How to Read This Manual

This reference manual is intended to give detailed explanations of the software features and menus. For typical usage examples, refer to the XFdtd Users Guide. The chapters of this manual are intended to introduce the important concepts that govern the use of the software. While there is no single "correct" way to create and execute a calculation from start to finish, the organization of this manual is intended to guide a new user through various steps of the simulation process.

While every effort has been made to make the software intuitive and easy to use, XFdtd does require some general knowledge of the concepts of the simulation technique on which it is based. Chapter 2 contains a brief summary of the Finite-Difference Time-Domain (FDTD) method for electromagnetic simulation.

Chapters 3 through 9 outline the step-by-step process of preparing a project for simulation. Chapters 3 and 4 provide a general overview of XFdtd and the graphical interface in order to familiarize the user with the software. Chapter 5 details the process of creating new geometry, applying materials, importing geometry from external files and the various tools that facilitate this process. Chapter **??** outlines how to add discrete components, external excitations, waveforms and voltage points, and also provides some of the characteristics and governing equations associated with these features. Chapter 8 details the process of defining an effective grid and mesh that will provide the desired level of detail without allocating excessive memory usage and calculation time. XFdtd provides several tools for customizing the grid based on specific features and regions in the geometry that are explained in this chapter. Lastly, Chapter 9 provides the guidelines for properly considering how the calculation engine treats the outer bounds of the project.

Chapters 10, 11 and 12 detail the procedure for requesting results, running the calculation engine and viewing the requested results. XFdtd 7 has greatly improved from previous major versions with the introduction of sensors, which function to collect results. Chapter 10 provides a full-description of sensors, their improvement over the former methods for data collection in earlier versions and the data that each type collects. Users interested in the biological features of XFdtd such as the human body meshes, Specific Absorption Rate (SAR) and temperature rise calculations should reference this chapter for information about biological sensors.

During the calculation process (described in Chapter 11), results are derived by means of RESULTS objects that are generated from their corresponding sensors. Chapter 12 documents the various ways of viewing these output results.

Finally, Chapter 13 details several "extras" that have been incorporated for customizing and organizing projects. These include three powerful tools that have been added since previous major versions that greatly facilitate and shorten the simulation process for current projects as well as future projects.

Several appendices are available to provide users with additional information about the features of XFdtd. Appendix A details and gives examples of the various GEOMETRY TOOLS including the 2-D shape tools, constraint tools, snapping tools, 3-D modeling operations, advanced 3-D modeling operations, modification operations and 3-D patterns. Appendix B provides important supplementary information about choosing an appropriate cell size, as well as a discussion of fixed points and grid regions, two powerful tools for customizing the grid. Finally, Appendix C offers several resources and reference points for previous users of XFdtd.

1.2 Important Changes for Users of Previous Versions of XFdtd

XFdtd 7 is a significant improvement over previous major versions of XFdtd. It was designed to increase the productivity of the user with a more intuitive interface, cross-platform functionality, and added speed and flexibility. With advanced meshing, parameterization, scripting and a number of other ground-breaking new features, XFdtd is now a more accessible and productive tool than other comparable products in the market thus far.

Below is a summary of the highlights of this latest release of XFdtd:

- **A Cross-Platform User Interface.** You now have the flexibility to use XFdtd on the platform of your choice. The software will run natively and in its entirety on Linux, MacOS and Windows.
- **Scripts and Macros.** These new additions to XFdtd allow you to customize and perform more complicated tasks than previous major versions.
- **Improved Parameterization.** Adjust and fine-tune your designs to a greater degree with extensive parameterization features.
- **Modeler Enhancements.** XFdtd now offers more powerful CAD modeling.
- **Improved Meshing Accuracy and Speed.** XFdtd creates a more accurate and memory-efficient representation of your model.
- **XACT: Accurate Cell Technology.** Improves simulation speed and accuracy by representing geometric features at sub-cell resolution in the mesh (see Chapter 8.3.4).

- **Imported CAD File Editing.** Modify CAD files on your own and eliminate down time waiting for others to do it for you.
- **ODB++ Printed Circuit Board (PCB) Import.** XFdtd imports PCB geometry from circuit layout design tools, using the PCB industry's standard file format for data exchange.
- **User Interface Refinements.** The more sophisticated and powerful GUI simplifies the learning curve for new users and increases efficiencies for all users.
- **Comprehensive Output Control.** XFdtd offers greatly expanded data-saving and displaying features interfaced through a more powerful browser.
- **Copy and Paste.** XFdtd now enables users to copy and paste, as well as drag-and-drop, items in the project tree (both within an instance and across multiple instances).

▶ Refer to Appendix C to see additional resources for users of previous major versions of XFdtd.

1.3 A Guide to the Graphics

In an attempt to make this manual as easy to read as possible, we've used several key icons to help organize the material.

→	At the beginning of each chapter, there is a summarized list of topics you will find within. This symbol marks each topic.
▶	Wherever there is a reference to another section with additional information on the topic being discussed, this symbol will lead you there.
✓	This symbol marks a "smart tip," which offers a helpful way to think about a topic or complete a task.
⚙	There are times when the user must be careful to remember something or especially cautious. This symbol will warn you to pay attention!
•	Lists are marked by the classic bullet point. Sub-topics within the list are marked with a dash (-).

Given XFdtd's support of multiple platforms, keep in mind that the interface you see on your computer screen may appear slightly different from what is shown throughout this manual. Minor interface changes, such as button locations and menu layouts, have been individually configured for maximum performance within each platform. In some cases, figures have also been modified slightly from their original version to fit this manual. All screen shots within this manual were taken on a computer running Windows XP.

Chapter 2

The Finite-Difference Time-Domain Method

In this chapter, you will learn...

→ the benefits of using FDTD to perform your electromagnetic simulation

→ an (abridged) explanation of the theory behind FDTD

→ several main factors to consider as you prepare to set up an electromagnetic calculation in XFdtd

This chapter provides an introduction to the concepts of the Finite-Difference Time-Domain (FDTD) method. The approach has existed since the 1960's but has gained popularity in recent years as computer performance has increased.

- For more detailed information on FDTD, refer to the text *The Finite Difference Time Domain Method for Electromagnetics* by Kunz and Luebbers[1], and *Computational Electrodynamics: The Finite-Difference Time-Domain Method, Third Edition* by Taflove and Hagness[2].

2.1 Why Use FDTD?

While many electromagnetic simulation techniques are applied in the frequency-domain, FDTD solves Maxwell's equations in the time domain. This means that the calculation of the electromagnetic field values progresses at discrete steps in time. One benefit of the time domain approach is that it gives broadband output from a single execution of the program; however, the main reason for using the FDTD approach is the excellent scaling performance of the method as the problem size grows. As the number of unknowns increases, the FDTD approach quickly outpaces other methods in efficiency.

FDTD has also been identified as the preferred method for performing electromagnetic simulations for biological effects from wireless devices. Researchers have shown the FDTD method to be the most efficient approach in providing accurate results of the field penetration into biological tissues.

- See IEEE publication *C95.3 Recommended Practice for Measurements and Computations with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 100 kHz to 300 GHz* [3], and the Toflove and Hagness text [2].

2.2 A Brief Summary - FDTD Basics

In the FDTD approach, both space and time are divided into discrete segments. Space is segmented into box-shaped cells, which are small compared to the wavelength. The electric fields are centered on the edges of the box and the magnetic fields are centered on the faces as shown in Figure 2.1. This orientation of the fields is known as the Yee cell, and is the basis for FDTD.

- For a description of the Yee cell, see IEEE publication *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media* [4].

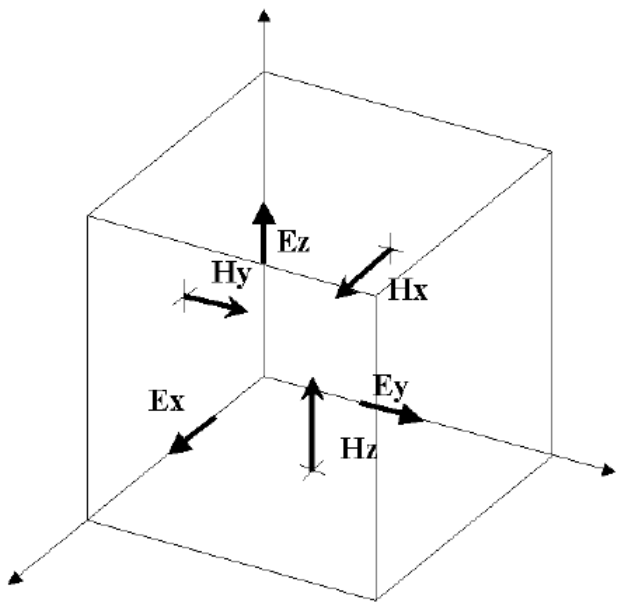


Figure 2.1: The Yee cell with labeled field components

Time is quantized into small steps where each step represents the time required for the field to travel from one cell to the next. We refer to this as the "timestep". Given the offset in space of the magnetic fields from the electric fields, the values of the field with respect to time are also offset. The electric and magnetic fields are updated using a leapfrog scheme where the electric fields and then the magnetic are computed at each step in time.

- See Section 2.7 for a description of how the timestep is calculated.

When many FDTD cells are combined together to form a three-dimensional volume, the result is an FDTD grid. Each FDTD cell will overlap edges and faces with its neighbors, so by convention each cell will have three electric fields that begin at a common node associated with it. The electric fields at the other nine edges of the FDTD cell will belong to other, adjacent cells. Each cell will also have three magnetic fields originating on the faces of the cell adjacent to the common node of the electric fields, as shown in Figure 2.1.

Within the mesh, materials such as conductors or dielectrics can be added by changing the equations for computing the fields at given locations. For example, to add a perfectly conducting wire segment to a cell edge, the equation for computing the electric field can be replaced by simply setting the field to zero since the electric field in a perfect conductor is always zero. By joining numerous end-to-end cell edges defined as perfectly conducting material, a wire can be formed. Introducing other materials or other configurations is handled in a similar manner and each may be applied to either the electric or magnetic fields depending on the characteristics of the material. By associating many cell edges with materials, a geometrical structure can be formed within the FDTD grid such as the dielectric sphere shown in Figure 2.2. Each small box shown in the figure represents one FDTD cell.

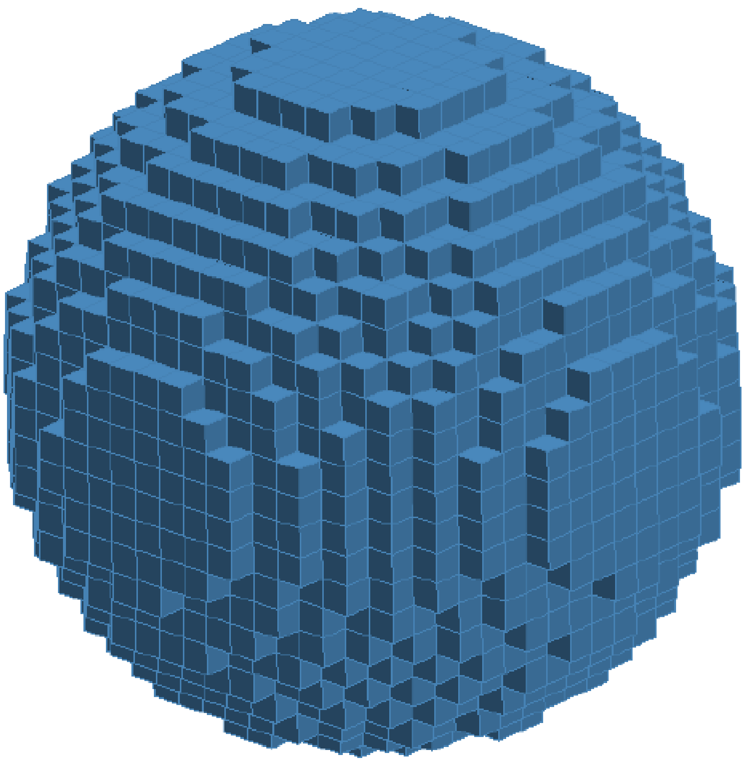


Figure 2.2: A dielectric sphere as meshed in an FDTD grid. The individual cell edges (electric field locations) are displayed as the overlapping grid lines.

The cell size, the dimensions of the box, is the most important constraint in any FDTD simulation since it determines not only the step size in time, but also the upper frequency limit for the calculation. A general rule of thumb sets the minimum resolution, and thus the upper frequency limit, at ten cells per wavelength. In practice the cell size will often be set by dimensions and features of the structure to be simulated such as the thickness of a substrate or the length of a wire.

An excitation may be applied to an FDTD simulation by applying a sampled waveform to the field update equation at one or several locations. At each step in time, the value of the waveform over that time period is added into the field value. The surrounding fields will propagate the introduced waveform throughout the FDTD grid appropriately, depending on the characteristics of each cell. A calculation must continue until a state of convergence has been reached. This typically means that all field values have decayed to essentially zero (at least 60 dB down from the peak) or a steady-state condition has been reached.

2.3 Materials

FDTD is capable of simulating a wide variety of electric and magnetic materials. The most basic material is free space. All FDTD cells are initialized as free space and the fields at all cell edges are updated using the free space equations unless another material is added to replace the free space.

Perfectly conducting electric and magnetic materials are simulated by setting the electric or magnetic field to zero for any cell edges located within these materials. Because of the simplicity of the calculation for these materials, it is better to use a perfect conductor rather than a real conductor whenever feasible. Conductors such as copper can be simulated in FDTD, but since the equations for computing the fields in copper material are more complicated than those for a perfect conductor, the calculation will take longer. For cases where only a small percentage of the FDTD cells are defined as a conductor, the difference in execution time will hardly be noticeable.

Frequency-independent dielectric and magnetic materials, considered normal materials by XFdtd, are defined by their constitutive parameters of relative permittivity and conductivity for the electrical material or relative permeability and magnetic conductivity for the magnetic material. In most cases, even when performing a broadband calculation, these materials are appropriate since the parameters do not vary significantly over the frequency range.

In some cases a frequency-independent material is not appropriate and instead a frequency-dependent, or dispersive, material should be substituted. Some common examples of frequency-dependent materials are high water content materials such as human tissues and metals when excited at optical frequencies. Included in XFdtd is the capability to simulate electric and magnetic Debye and Drude materials such as plasmas, Lorentz materials, and anisotropic magnetic ferrites, as well as frequency-independent anisotropic dielectrics, and nonlinear diagonally anisotropic dielectrics.

2.4 Near-Zone Versus Far-Zone

For any given calculation, the geometry of the structure being simulated is defined by setting the cell edges at specific locations to certain materials. The entire FDTD geometry space, commonly called the grid (without applied materials) or the mesh (with applied materials), is composed of a three-dimensional block of these cells.

This three-dimensional volume is considered to be the near-zone region in XFdtd in terms of data storage. The field value at any edge in the FDTD grid may be observed as a function of time by saving a near-zone point over a range of timesteps. Other types of data such as steady-state field magnitudes, specific absorption rates, S-Parameters, or impedance also may be stored as near-zone (within the grid) values.

It is possible to make an FDTD grid that is large enough to allow sampling of points in the far-field of a geometry. In general, this will be extremely costly in terms of computer memory and calculation time since the number of unknowns (cells) will most likely be large. Note that each FDTD cell has a maximum size of one-tenth of a wavelength, so moving several wavelengths away from a structure will require many cells. In most cases, this is not an appropriate method of monitoring far-field results.

A more practical method for computing far-zone radiation gain or radar scattering patterns is to use a mathematical transformation to convert the near-zone values in the FDTD grid into far-field values at locations away from the grid. In XFdtd, this transformation is based on near-zone fields stored on the six faces of an imaginary box enclosing the geometry. The faces of the box are located five FDTD cells from each outer edge of the FDTD grid. For the transformation to be valid, all parts of the XFdtd geometry must be contained within the box.

The coordinate system used in XFdtd is defined with the azimuthal (phi) angle referenced from the X-axis and the elevation (theta) angle referenced from the Z-axis, as shown in Figure 2.3. This coordinate system is used for locating far-zone positions and for defining the incident plane wave direction in XFdtd.

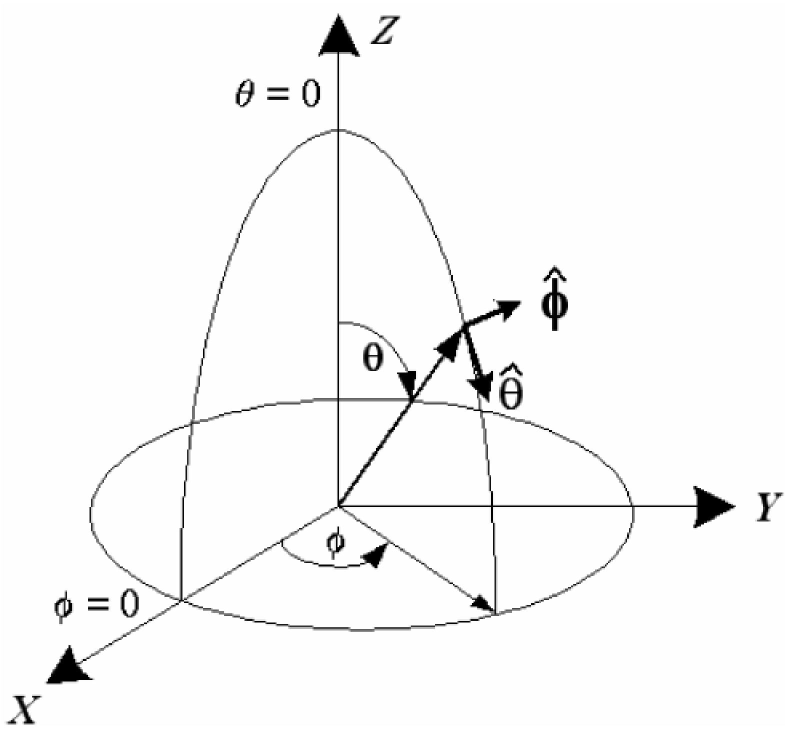


Figure 2.3: The coordinate system used in XFdtd for far-zone and incident plane wave directions

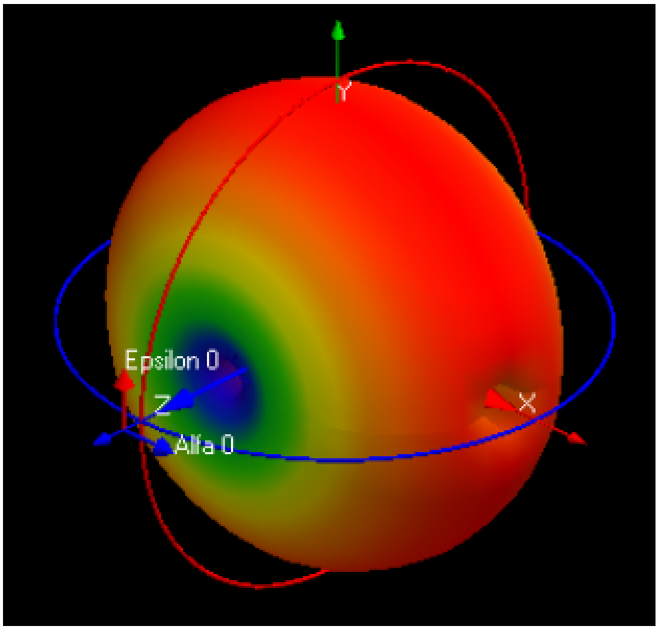


Figure 2.4: The coordinate system for Alpha, Epsilon far-zone patterns

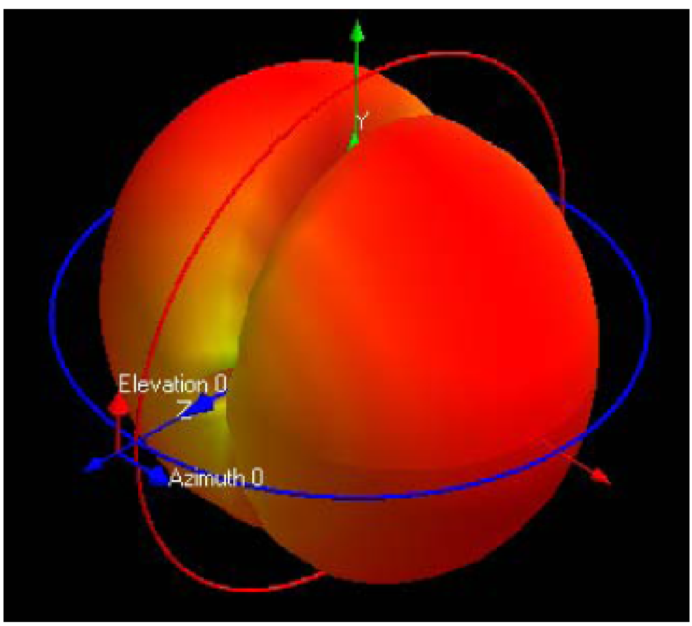


Figure 2.5: The coordinate system for Azimuth, Elevation far-zone patterns

2.5 Broadband vs. Single-Frequency Calculations

As mentioned in the introduction, XFdtd uses a time-domain solver, which enables results for a single-frequency calculation or a multiple-frequency (broadband) calculation for which there is sufficient input excitation. In other words, one computation provides results for the frequency range of the excitation pulse. For example, a properly-defined Gaussian pulse can provide excitation from DC to the maximum frequency supported by the mesh, which is limited only by computer resources.

Many results are automatically available for all excitation frequencies. Certain data, such as SAR, may require significant computation time and memory for each frequency, so the user is given the ability to specify the individual frequencies for which they are interested in this data.

2.6 Outer Boundaries

A three-dimensional grid of cells forms the XFdtd geometry and the fields updated at every cell location are dependent on the neighboring fields. However, due to memory limitations the grid must end at some point and because of this, the fields on the outer edges of the grid cannot be updated correctly. To correct this situation, outer radiation boundary conditions are applied at the edges of the XFdtd grid.

The outer boundary is a method for absorbing fields propagating from the XFdtd grid toward the boundary. By absorbing these fields, the grid appears to extend forever. The performance of the outer boundaries is an important factor in the accuracy of the XFdtd calculation, and care should be taken to correctly use them.

In some cases a reflecting boundary rather than an absorbing one is preferred. A perfectly conducting boundary (either electric or magnetic) may be used to image the fields in an XFdtd calculation.

2.7 Computer Resources

The XFdtd software estimates computer memory resources needed for simulation. The information in this section is presented to explain the basis for this estimate.

FDTD is a computationally intensive method and most reasonable calculations will need a fast computer and several hundred megabytes of computer memory. For most applications, it is fairly simple to estimate the amount of computer memory required for a calculation. The most important factor for the memory usage, and in large part the run time, is the number of FDTD cells used to represent the structure under test. Each FDTD cell has six field values associated with it: three electric fields and three magnetic fields. Additionally each cell has six flags associated with it to indicate the material type present at each of the six field locations. The field values are real numbers, each four bytes in length, while the flags are each one byte. This gives a memory usage per FDTD cell of 24 bytes for fields and 6 bytes for flags, for a total of 30 bytes.

✓ To estimate the total memory required, in bytes, simply multiply the number of FDTD cells by the 30 bytes per cell value. There is some overhead in the calculation, but it is generally quite small. Three notable exceptions are: broadband far-zone directions that allocate six one-dimensional real value arrays per direction; the use of DFT frequencies, i.e. collecting steady-state data when using a broadband pulse for excitation; and the use of the PML outer boundary.

Estimating the execution time of an XFdtd calculation is more complicated since computer processor performance varies.

- ✓ One method of estimating is to compute the total number of operations to be performed. There are about 80 operations per cell, per timestep during the XFdtd calculations. The total number of operations is found by multiplying the number of cells, the number of timesteps, and the factor of 80 operations per cell, per timestep.

If the value of the floating-point performance of the processor is known, it is possible to compute a value for execution time. In general, however, a better estimation method is to determine the execution time of a simple problem on a given computer, and then scale the time by the ratio of the number of operations between the desired calculation and the simple one.

The timestep size and number of required timesteps are problem-dependent. The size of the timestep is determined by the size of the cells in the problem space. The maximum timestep allowed is:

$$\triangle t = \frac{1}{c}\left(\frac{1}{\triangle x^2} + \frac{1}{\triangle y^2} + \frac{1}{\triangle z^2}\right)^{-1/2}$$

where

c is the speed of light

$\triangle x$, $\triangle y$, and $\triangle z$ are the lengths of the cell sides, in meters

The timestep used for computation must be no longer than the smallest timestep limit for each of the cells in the problem space.

Chapter 3

An Overview of XFdtd

In this chapter, you will learn...

→ how to categorize the major components of the XFdtd interface

The XFdtd interface has been redesigned from earlier versions to give users a great deal of freedom and functionality while creating their projects. Consequently, there is no single "correct" way to set up and run a calculation. This chapter provides a general overview of the process of creating a project.

Figure 3.1 shows the XFdtd Graphical User Interface (GUI) with several key menus expanded.

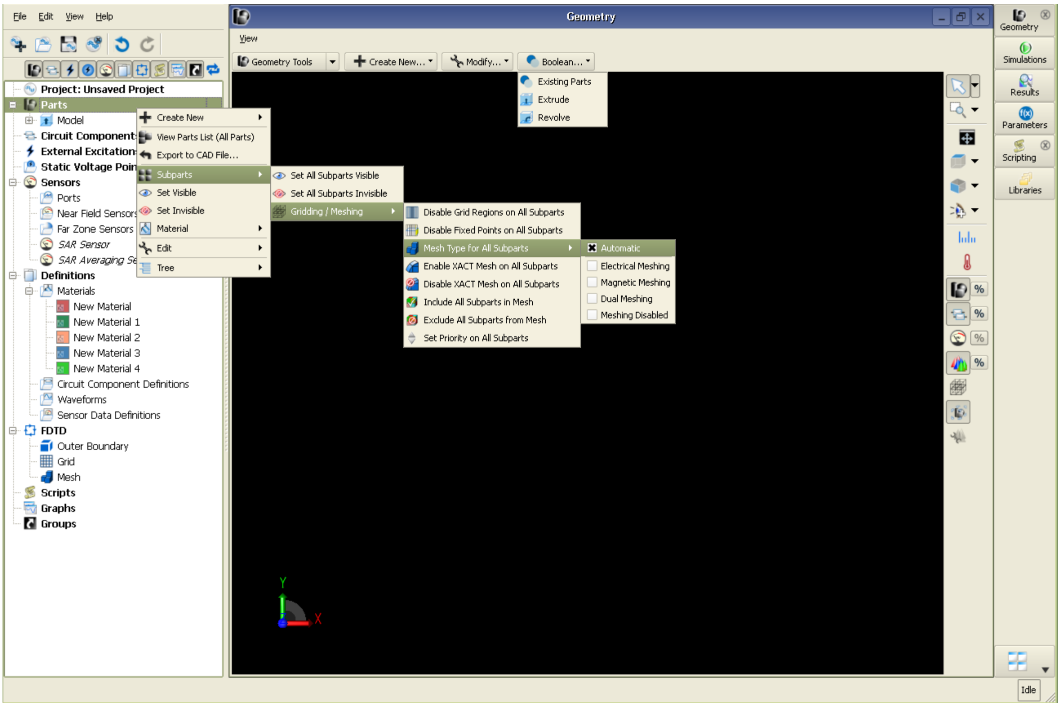


Figure 3.1: The XFdtd GUI with several expanded menus

Typically, an XFdtd project will begin with the creation of the physical geometry. Geometric modeling takes place within the GEOMETRY TOOLS dialog of the GEOMETRY workspace window in the XFdtd GUI. Here, objects can be created from scratch or imported from external files. The SPECIFY ORIENTATION tab within this dialog contains flexible tools used to position physical parts in the geometry.

Figure 3.2 shows the location of the GEOMETRY TOOLS dialog.

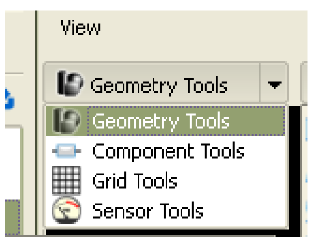


Figure 3.2: Zooming in to the Geometry Tools option

The PROJECT TREE has a DEFINITION branch containing four nodes: 1) Materials, 2) Circuit Component Definitions, 3) Waveforms and 4) Sensor Data Definitions. Add all new materials under the Materials node, and so forth for the other defintion types. Once a definition has been created, for example a material definition, it can be assigned to geometry objects by dragging the definition and dropping it on the object. Many objects can be assigned to a given definition. Select a group of objects before the drag-and-drop, or perform more than one drag-and-drop. All objects referencing a given definition always see the definition's current value, so a change to a single definition immediately is reflected in every object referencing that definition.

Material definitions are created as DEFINITION objects in the MATERIAL EDITOR and stored in the PROJECT TREE so that they can be easily applied to geometry objects by drag-and-drop.

Figure 3.3 shows the location of XFdtd project DEFINITIONS.

Figure 3.3: Zooming in to the Definitions branch

After geometry objects are created and given valid material definitions, discrete circuit components may be added within the COMPONENT TOOLS dialog of the GEOMETRY workspace window. Circuit component definitions are created within the CIRCUIT COMPONENT DEFINITION EDITOR and stored as DEFINITION objects in the PROJECT TREE. A definition can be applied easily to the appropriate components by drag-and-drop.

In addition to discrete sources, external excitation sources can be added using the PLANE WAVE EDITOR and GAUSSIAN BEAM EDITOR. Waveform definitions are created in the WAVEFORM EDITOR as DEFINITION objects within the PROJECT TREE and are applied by drag-and-drop to objects that require waveform definitions.

Figure 3.4 shows the locations of XFdtd project CIRCUIT COMPONENTS and EXTERNAL EXCITATIONS.

Figure 3.4: Zooming in to the Circuit Components and External Excitations branches

Next, the grid (which provides the “blueprint” for the meshing operation) is defined within the GRID TOOLS dialog. This dialog is accessed within the GEOMETRY workspace window or by double-clicking on the GRID icon in the PROJECT TREE. General characteristics such as limits and cell, bounding box and padding sizes are defined in this dialog. Additionally, customized regions may be defined by adding fixed points and grid regions so that more important regions of the project can be meshed more finely, while less-important regions can be meshed with larger cells. Object-specific grid definitions may be applied within the GRIDDING PROPERTIES EDITOR, accessed by right-clicking a geometry object in the PROJECT TREE. This editor provides options for adding fixed points and grid regions to specific objects based on its characteristics, which in some cases may be more convenient than defining them within the general grid definitions provided in the GRID TOOLS dialog.

Figure 3.5 shows the location of XFdtd project GRID TOOLS and GRIDDING PROPERTIES EDITOR.

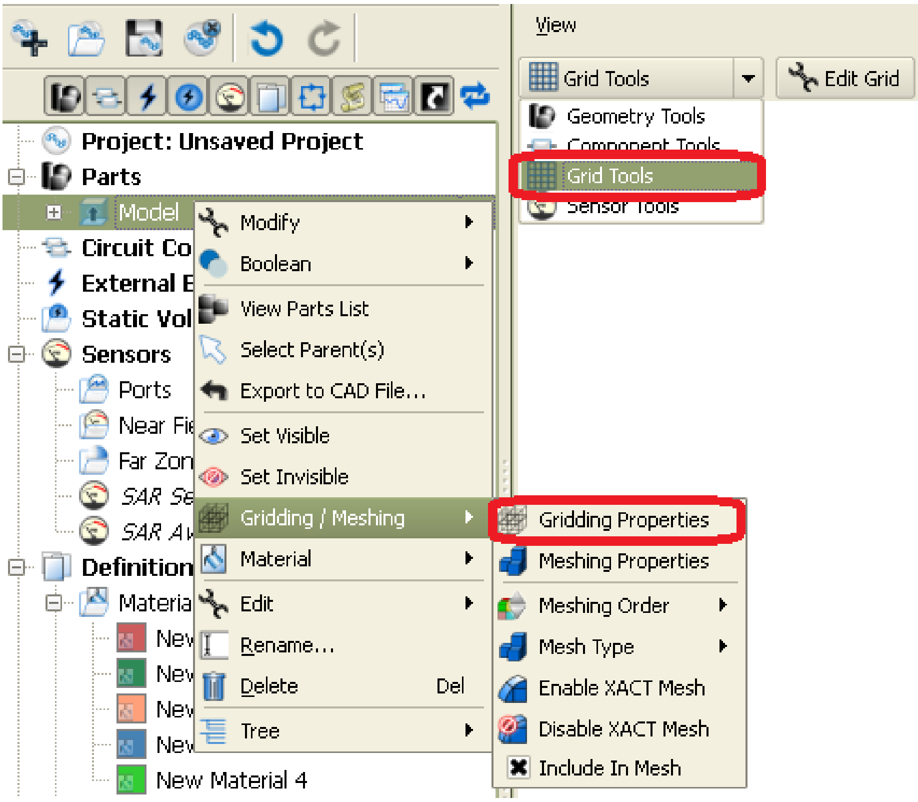


Figure 3.5: Zooming in to the Grid Tools and Gridding Properties menu options

Once the grid is defined, the location of every cell edge is known, and the project can be meshed. During the meshing operation, materials are applied to the appropriate cell edges. Meshing occurs automatically in XFdtd as soon as the project is viewed within MESH VIEW. Subsequent meshes are generated by manually pressing the REMESH NOW button or by selecting the AUTOMATIC REMESHING feature, which will generate a mesh after every change to the geometry is made. AUTOMATIC REMESHING is the default. Both of these options are found by right-clicking on the MESH icon in the PROJECT TREE. Meshing considerations such as MESHING PRIORITY and TYPE (magnetic and/or electrical) may be specified in the MESHING PROPERTIES EDITOR or by right-clicking on a specific PARTS object in the PROJECT TREE.

Figure 3.6 shows the location of the meshing options dialog.

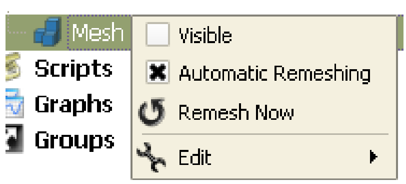


Figure 3.6: Zooming in to the meshing options dialog

OUTER BOUNDARY conditions, which regulate how the calculation engine treats the boundaries of the project, are specified within the OUTER BOUNDARY EDITOR. This editor is accessed by double-clicking on the corresponding branch in the PROJECT TREE.

Figure 3.7 shows the branch where you can access the OUTER BOUNDARY EDITOR.

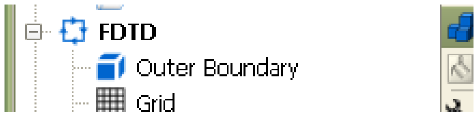


Figure 3.7: Zooming in to the Outer Boundary branch

Once the various components of the project are defined, SENSOR objects can be added. Sensors are simply objects that request data. There are several different types of sensors based on what type of data they retrieve. They can be added from the PROJECT TREE or within the SENSOR TOOLS dialog in the GEOMETRY workspace window.

Figure 3.8 shows the two locations from which you can add SENSOR objects.

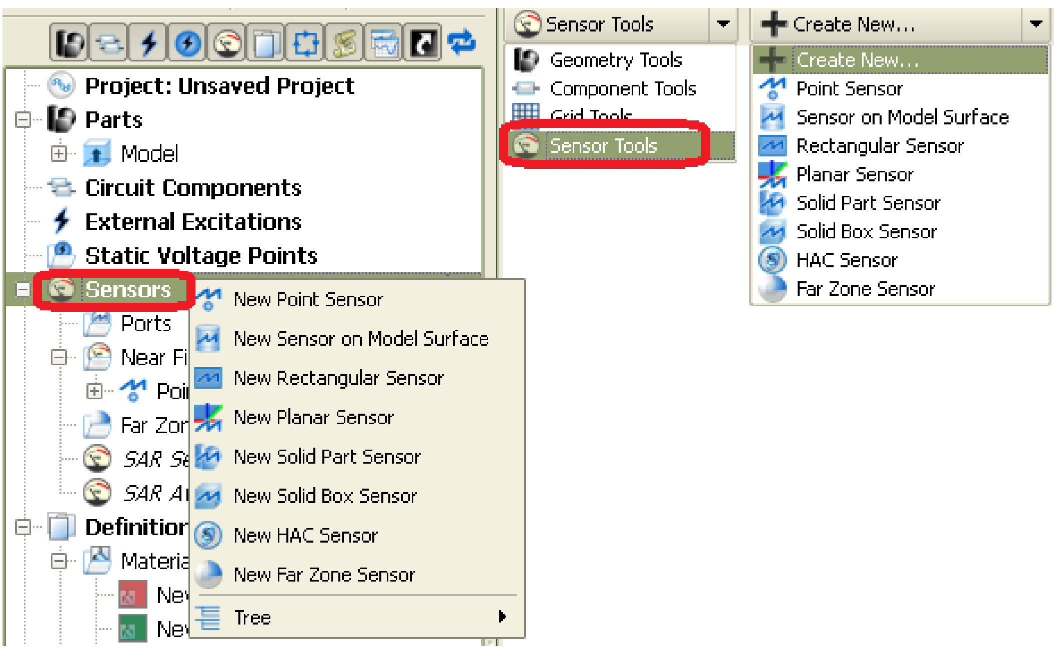


Figure 3.8: Zooming in to the Sensor menu options

Calculation criteria are specified within the SIMULATIONS workspace window once sensors have been added to retrieve all desired results. Here, specifications such as source type, parameter sweep definitions, S-Parameter feeds, frequencies of interest, total/scattered field interfaces and termination criteria are defined for each simulation that will be run by the calculation engine, XFsolver.

After running the calculation, view the results from the RESULTS workspace window.

Figure 3.9 shows the SIMULATIONS and RESULTS workspace windows.

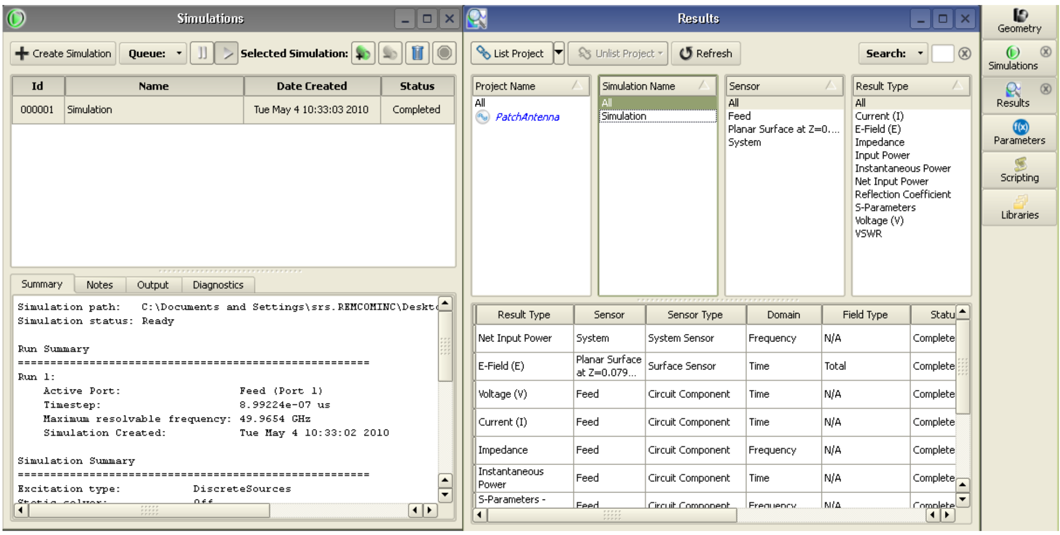


Figure 3.9: Zooming in to the Simulations and Results workspace windows

Chapter 4

The XFdtd Interface

In this chapter, you will learn...

→ the features of the APPLICATION MENU BAR

→ the branches of the PROJECT TREE

→ the major functions of the workspace windows

→ what to do when the ⚠ icon appears

Older versions of XFdtd incorporated the use of tabbed windows to guide the user through the various stages of the simulation process. While this interface was a convenient way to set-up projects in previous versions, it is not sufficient to present additional features that have now been incorporated into XFdtd 7. As a result, the XFdtd interface has been redesigned to accommodate these new features while remaining as simple and intuitive for the user as possible. Rather than relying solely on tabbed sections to navigate through the simulation process, XFdtd utilizes a workspace design as well as the powerful PROJECT TREE to control the various aspects of the XFdtd simulation.

- Users familiar with older versions of XFdtd should refer to Appendix C for a discussion of these changes.

The XFdtd workspace is comprised of several parts that are intended to make it as easy as possible to create projects and run calculations. Figure 4.1 highlights the various parts of the new interface.

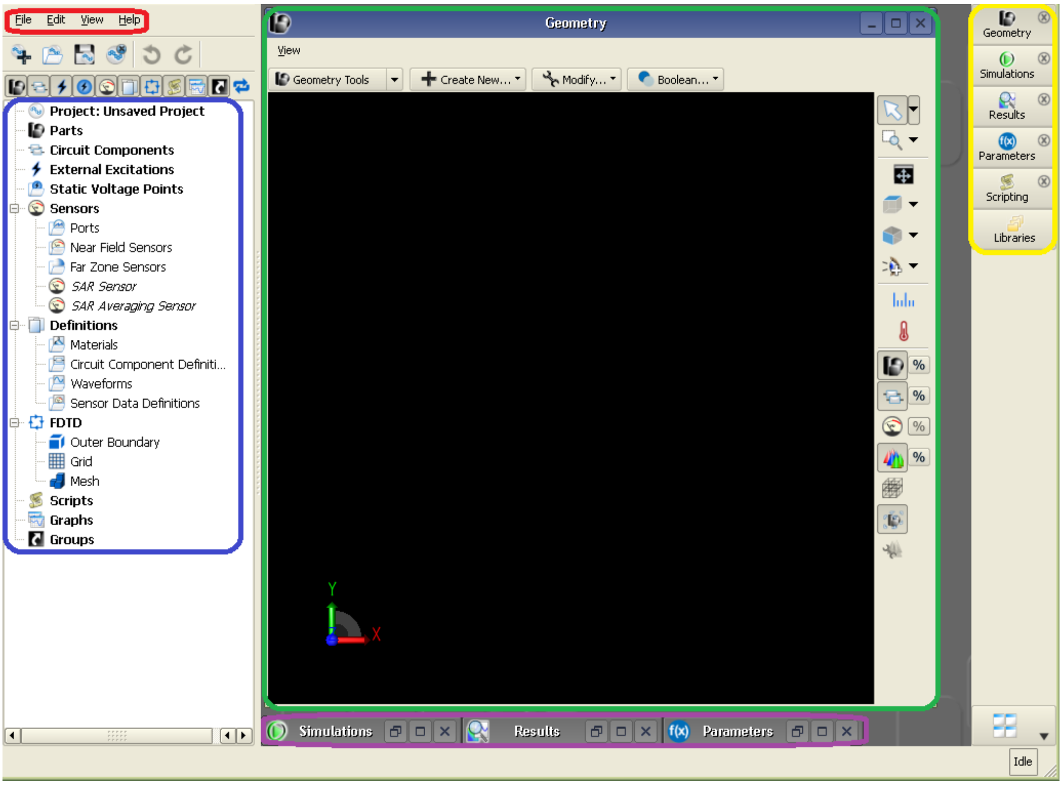


Figure 4.1: The XFdtd 7 workspace

The APPLICATION MENU BAR is seen outlined in red in the upper left of the figure.

The PROJECT TREE, outlined in blue, is used to display and organize the contents of the project within categorized branches. The row of buttons above it toggles the various branches of the tree on and off.

The series of tabs along the right of the screen (outlined in yellow) store the various workspace windows so that they can be easily accessed within the project workspace. Minimized workspace windows are stored within the actual workspace, as outlined in purple in Figure 4.1. Open workspace windows are displayed in this area as well, as shown in green.

4.1 Application Menu Bar and Toolbar

4.1.1 File Drop-Down Menu

New Project Opens a new XFdtd project. By default, the project is blank. To set a template as the default new project, click the Managing Project Templates link.

New Project from Template Opens an XFdtd project from a saved template. (Templates are described in detail below.)

Open Existing Project Opens an existing project.

New Window Submenu containing items to open a New Project, a New Project from Template or an Existing Project in a new application window.

Save Project Saves a new project to a specified directory.

Save Project As Saves a project to a new or existing directory.

Close Project Closes the current project.

Import Imports an existing CAD file, voxel object file, XFdtd Version 6 file, XFdtd mesh file, or any other supported file format into the current XFdtd project.

Export Parts Exports parts from XFdtd to a file in a specified directory.

Managing Project Templates Select this option to create a new template or to assign a default template to be loaded whenever a new project is opened.

To create a new template, simply select an existing saved project file. This file will be saved as a template. Press the CREATE NEW button and assign the template a name. The default template is a blank project, but this can be changed to any other template that has been saved in this dialog.

Templates are useful for beginning new projects that contain identical parts to existing projects. For example, a new project may reuse the same geometry and discrete components that have already been created in an existing project file. In this case, the existing project file can be used to create a template so that this common geometry can be used over and over without having to rebuild the project from scratch.

Recent Projects Lists the five most recently loaded XFdtd projects.

Quit Closes the current XFdtd session.

4.1.2 Edit Drop-Down Menu

Undo/Redo The UNDO button will undo any action or series of actions. Similarly, the REDO button will repeat any actions that were mistakenly erased.

Project Properties The PROJECT PROPERTIES EDITOR contains settings that apply to the Project as a whole. It has several tabs of settings, as described below.

The FREQUENCY RANGE OF INTEREST tab allows control of the frequency range of interest for the project. This setting that allows XFdtd to automatically provide many simulation input parameters, more accurate project validation and tailored output. It is simply the frequency range, defined as the minimum and maximum frequencies, over which the user expects to retrieve valid or interesting simulation results. A specific range of interest via parameterized input can be specified, or the behavior of previous versions of XFdtd retained by allowing the Frequency Range of Interest to be specified by the grid settings based on 10 cells per wavelength for the maximum cell edge in the grid (this is currently the default behavior). The Frequency Range of Interest is currently used in the specification of the Automatic waveform, the Automatic Good Conductor determination on materials, validation of the Grid, and formatting of frequency-based output to the Run's Frequency Range of Interest.

The TIME DOMAIN DATA SAMPLING INTERVAL defines time between samples when collecting time domain data. Sensors can individually define the sampling or defer the choice to this project-level setting.

Figure 4.2 shows the FREQUENCY RANGE OF INTEREST tab.

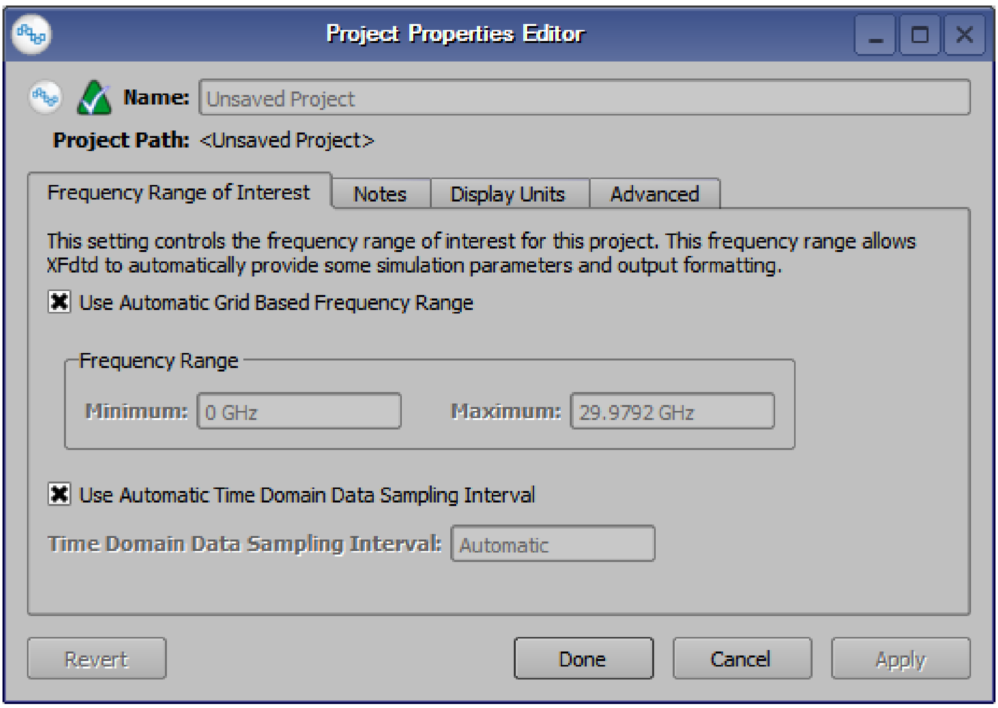


Figure 4.2: The Frequency Range of Interest tab in the Project Properties Editor

The DISPLAY UNITS tab defines the default units used in XFdtd and controls the units displayed in the geometry window. This setting will also control the default unit setting assigned to values that are entered by the user without units.

Modifying the display unit only changes how numbers are displayed and entered. It does not affect their scale.

A UNIT SET definition (XFDTD DEFAULT, SI METRIC, GAUSSIAN METRIC (GCS) or US CUSTOMARY) can be applied to all of the parameters listed under this tab, or a CUSTOM definition can be applied to any individual parameter.

Figure 4.3 shows the DISPLAY UNITS tab.

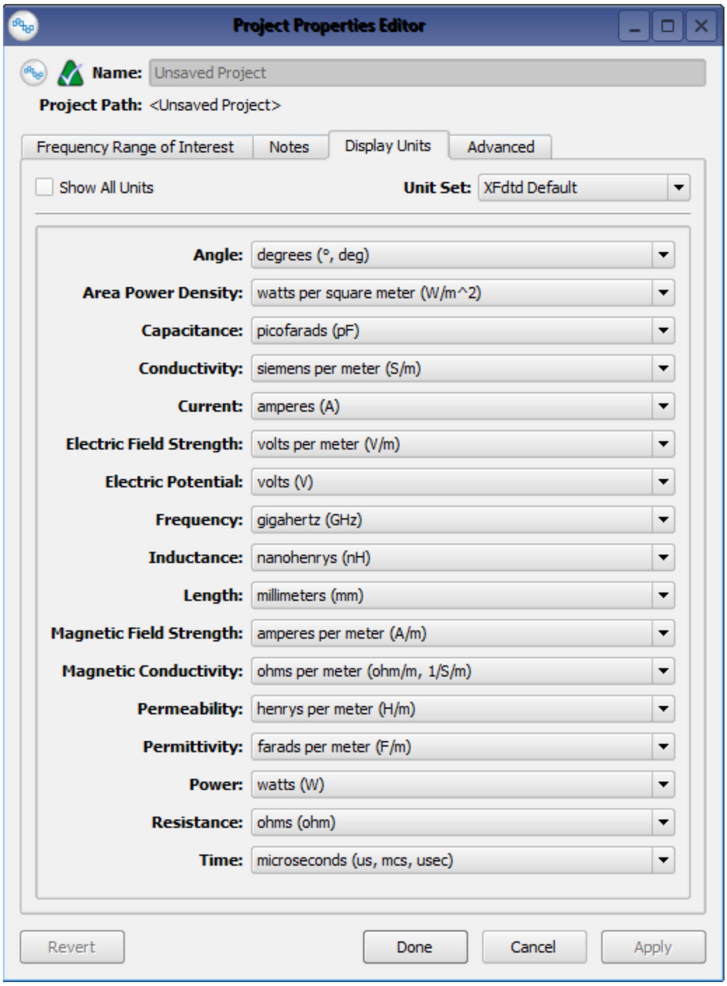


Figure 4.3: The Display Units tab in the Project Properties Editor

Under the ADVANCED tab, the user has the option of modifying the project timestep. The timestep that is used for the simulation is computed from the grid definition and the materials that are used in the space. In some special circumstances, the user may wish to make that timestep a little bit smaller to improve results or compensate for some other special conditions. The CUSTOM TIMESTEP MULTIPLIER is a factor by which the computed timestep is multiplied to obtain the final timestep that is actually used.

- See Appendix Section B.1 for more on how the timestep is calculated from the grid definition.

The XACT ACCURACY editors are used to further control the timestep when using an XACT mesh on any part in the simulation. See Chapter 8.3.4 for more information on setting this property.

Under this tab, the user can also assign New Part Modeling Units, which defines the units assigned to a new part when it is created (in, m, cm, etc.). Changing the value in the New Part Modeling Units can be helpful when a user needs to create an part that uses a different unit of measurement than other parts in the geometry. The units assigned to a new part are used by the 3-D modeling engine, which needs to know the range of units to correctly construct solid models.

After updating the value of the New Part Modeling Units drop-down list, all new parts created use that value. The value in the field can be updated as often as necessary.

This unit defines how the part is modeled internally, and cannot be changed once the part is created.

Note that this unit is not the same as what is defined in Display Units.

Figure 4.4 shows the Advanced tab.

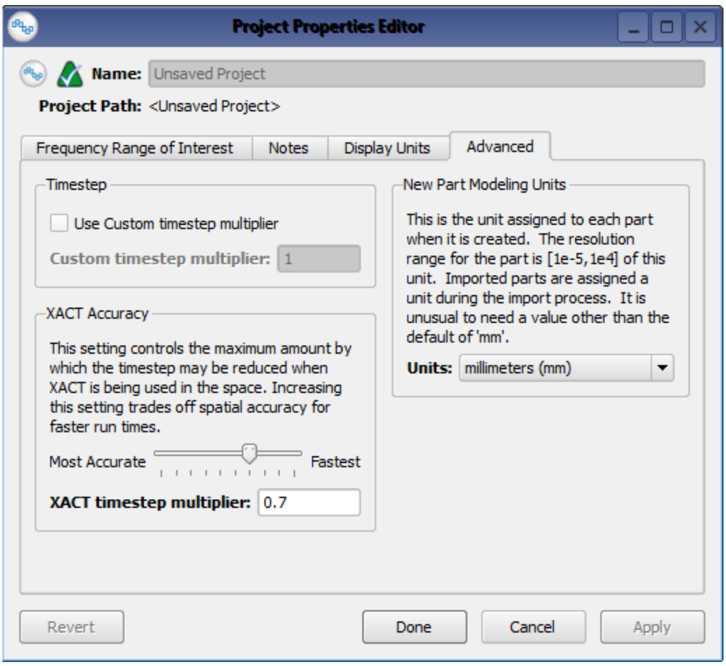


Figure 4.4: The Advanced tab in the Project Properties Editor

Application Preferences The five main tabs under Application Preferences are General, Interface, Modeling, Graphs and Compatibility.

The General Tab (Figure 4.5)

- The options in the Startup section control what happens at the start of XFdtd.
  - Show tip of the day, when enabled, will display a pop-up with a helpful tip about how to use XFdtd.
  - Show License Window, when enabled, will cause XFdtd to show the licensing window when the application starts where the user can specify where the license file/server is.

- Under Results, choose whether to load or clear results from previously-run simulations. Note that this does not delete simulation results saved to disk. Users can also set the minimum plot size for Fourier transformations.
- Under Recent Projects, select the number of projects that should appear in the Recent Projects submenu in the File menu.
- Under Templates, enter the folder location where XFdtd project templates are stored.
- Under Macros Menu, enter one or more folders (separated by ';' on Windows and ':' on Linux and Mac) where XFdtd should look for .xmacro files to populate the Macros Menu (see Section 4.1.4 Macros Menu.)
- Under Screen Capture, enter the folder where automatically-named screen shots should be saved. The global shortcut key combination *Ctrl+Alt+F+11* is defined to save a screen shot of the Geometry View or a Graph Window, whichever is "on top" in the application workspace, to an automatically-named file in this folder. If neither the Geometry View nor a Graph Window is on top, no action will be taken.
- Rendering Options controls several display options in the XFdtd GUI:
  - The TRANSPARENCY ALGORITHM determines the way XFdtd renders the %PARTS OPACITY of an object.
    - Z-SORT ONLY NICE is not as efficient as Z-SORT ONLY FAST, but removes most artifacts (small areas of inaccurate rendering.)
    - Z-SORT ONLY FAST is the most efficient rendering option, but may contain artifacts (small areas of inaccurate rendering.)
    - DEPTH PEELING is supported only on certain graphics cards. It is more accurate but runs slower than Z-SORT ONLY.
    - PAINTERS ALGORITHM is the most accurate rendering option but is the slowest.

    The TRANSPARENCY ALGORITHM setting only affects the way an object is displayed in the XFdtd interface and will not change calculation results.
  - TEXT COLOR defines the color of text automatically displayed in geometry view, such as coordinates, lengths, etc.
- The UNDO/REDO HISTORY section controls the how many items can remain on the undo stack. It gives the user the ability to enhance the application performance by limiting the memory used by the GUI.

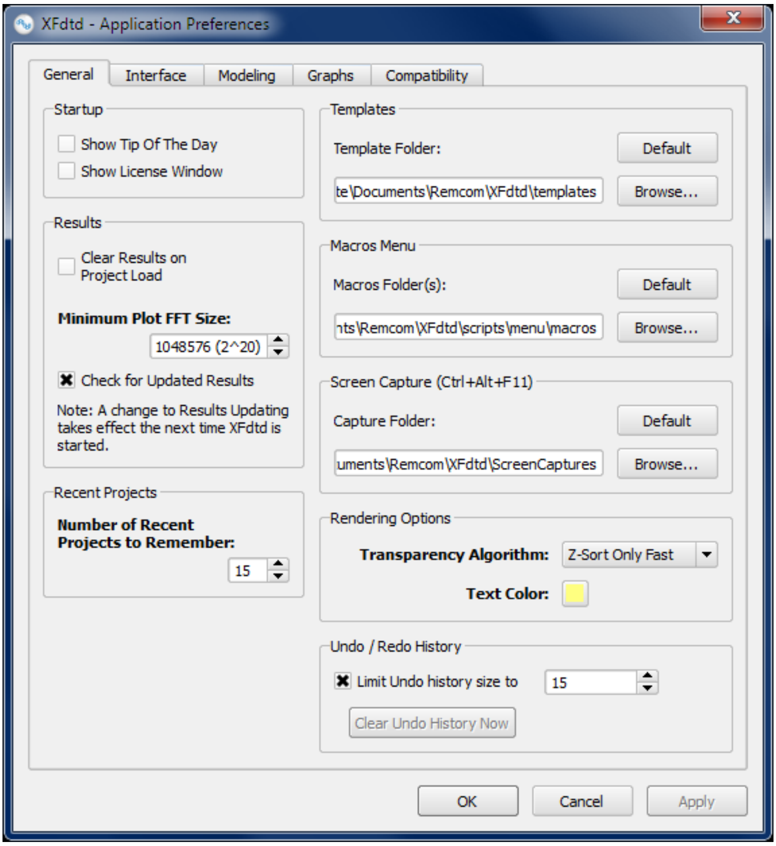


Figure 4.5: The General tab under Application Preferences

The INTERFACE Tab (Figure 4.6)

- Object Editing: Controls what happens when a new object is added to the PROJECT TREE (e.g. right-clicking on the tree to add a new material.)
  - NONE will simply add the new object to the tree.
  - EDIT NAME will add the object to the tree and provide a blinking cursor to add the object's name in the tree.
  - EDIT PROPERTIES adds the object and prompts the appropriate dialog window to immediately pop-up in order to edit the new object's properties (e.g. creating a new material will cause the MATERIAL EDITOR to pop-up).
- Layout: Allows the user to set his or her preferences for saving or restoring the GUI layout.
- Information: Allows the user to set his or her preference for decimal precision in tooltips.
- Gridding: Enables or disables the use of the simplified grid.
- Tool Bar Icon Size: Adjusts the size of the icons in the toolbar (e.g. New Project, Loading Existing Projects, etc.).

- Project Tree: The ICON/TEXT SIZE scroll adjusts the size of the items in the PROJECT TREE.
- Workspace:
  - SHOW ALL TABS shows all workspace windows in tabbed workspace regardless of whether they are active or not.
  - SHOW ONLY ACTIVE TABS stores only the active tabs that are stored in the project workspace.
  - DON'T SHOW TABS removes the tabbed workspace. Windows can still be accessed in the VIEW menu in the Application Toolbar.
- Appearance: Allows the user to change the appearance of the GUI and buttons within.

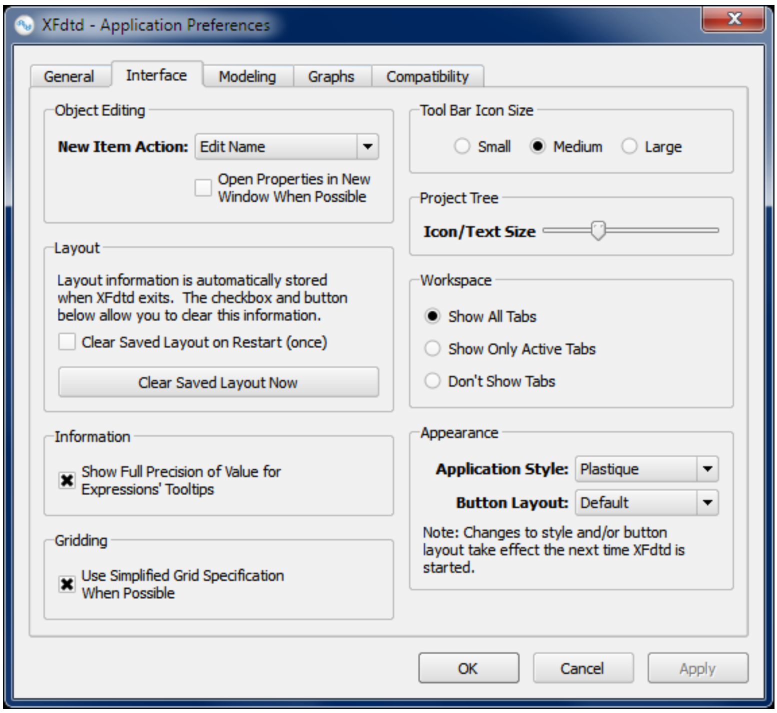


Figure 4.6: The Interface tab under Application Preferences

The MODELING Tab (Figure 4.7) contains two tabs (Part and Material) that allow the modification of the defaults appearances for new Parts and Materials. A new color assigned will be assigned to each new Material if the “Automatically Assign Unique Colors When Possible” checkbox is selected. If not selected, XFdtd will assign color defined in the FACE COLOR field to each new material.

Additionally, options for various elements of the view are customizable in the “View Options” grouping.

- Components: The color used to render circuit components.
- Background: The background color of the View.
- Construction Grid: The color for the lines of the construction grid of the Sketcher and Orientation tools.

- Mesh Slice: The color for mesh edges that are Free Space.
- Invert Mouse Wheel: Switches the zoom direction when rolling the mouse wheel.
- Separate View Tools: Toggles between showing all View tools in a drop-down list or as individual items.
- Smooth View Transitions: If marked, XFdtd performs smooth rotations between different view orientations. If unmarked, XFdtd snaps to the selected view orientation.
- Automatically Assign Unique Colors When Possible: If marked, XFdtd automatically assigns a unique color to each material a user adds to a project. If unmarked, XFdtd always assigns the color defined in the FACE COLOR field to each new material a user adds to a project.

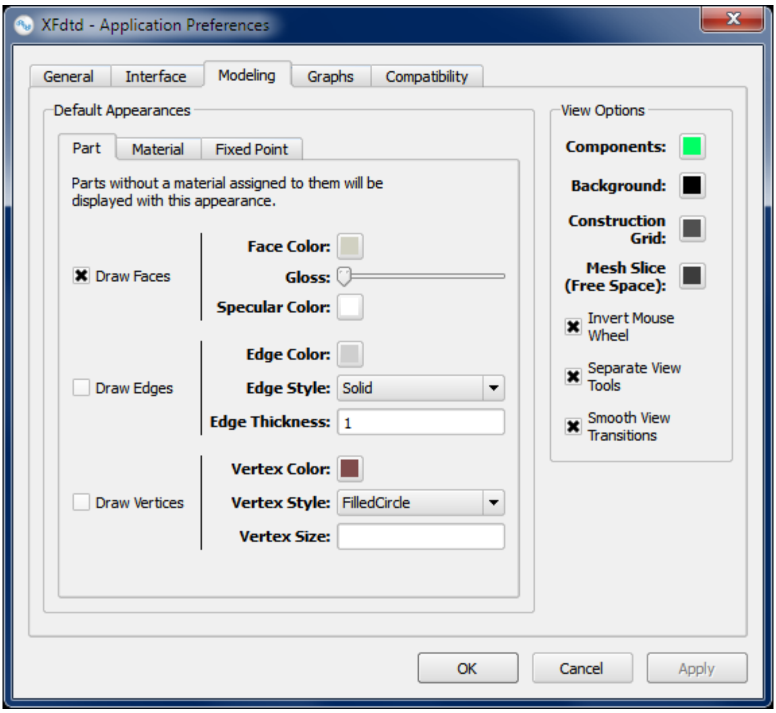


Figure 4.7: The Modeling tab under Application Preferences

The GRAPHS Tab (Figure 4.8) contains checkboxes that enable the user to determine what result fields are used to construct the default name for a plot when it is graphed or loaded into the Field Viewer. In a Graph, the default name will appear as the name of the plot in the Plot Properties tab and in the Legend. This name can be edited in the Plot Properties tab (see Section 12.3.2 Customizing Plot.) In the Field Viewer, the name is used in the combobox selector for what data is being displayed (see Section 12.4.3.4 The Field Editing Toolbar.)

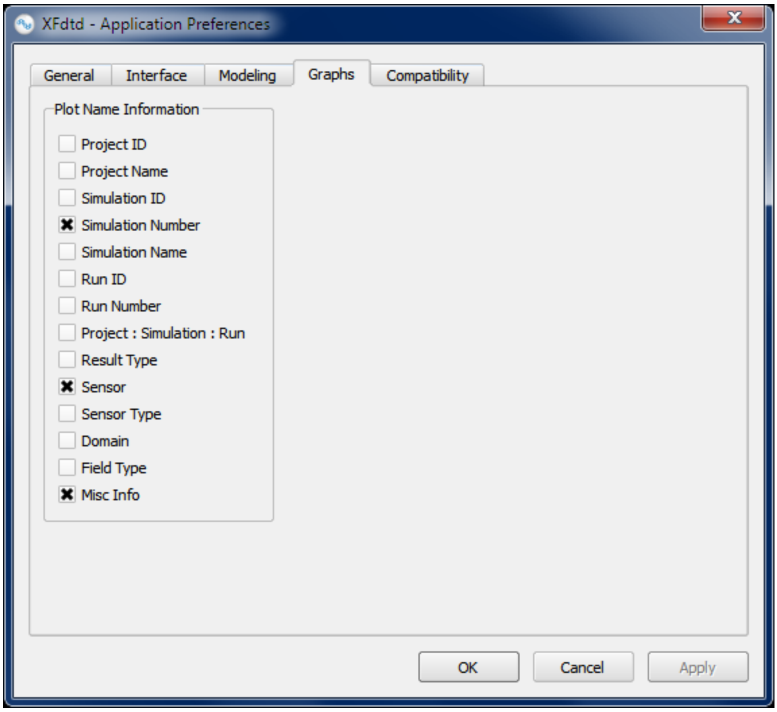


Figure 4.8: The Graphs tab under Application Preferences

When a behavior of XFdtd is changed, the settings on the COMPATIBILITY Tab (Figure 4.9) can be used to revert the behavior back to a previous version if desired. Generally, reverting to a previous behavior should not be needed. Items on this tab should be self-explanatory, and any changes that warrant an entry on this tab will also be noted in the What's New window and in the change log.

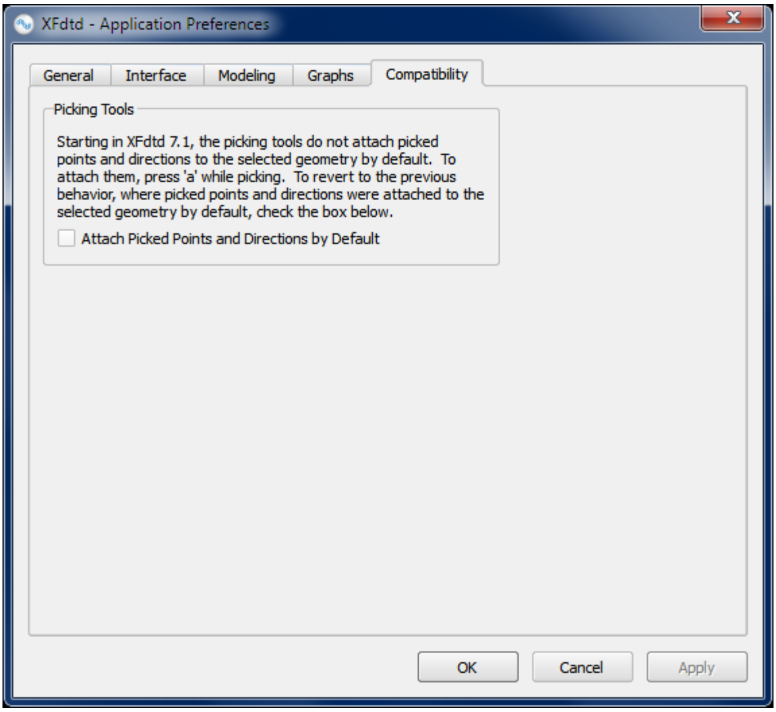


Figure 4.9: The Compatibility tab under Application Preferences

4.1.3 View Drop-Down Menu

The View menu item allows the visibility of toolbars and windows to be toggled. Note that when the button bar on the right of the application is hidden in the Application Preferences, these windows can be be shown using the View menu.

4.1.4 Macros Drop-Down Menu

In addition to being able to bundle function and macro scripts with a Project (see section 13.2 "Scripting Workspace Window"), macros that are used frequently and are project-independent can be placed in a special folder in the file system to be automatically detected and executed from the Macros menu. The Macros menu is populated by looking at each of the folders specified in the Application Preferences under General→Macros Menu for `.xmacro` files and subfolders. The default locations are
`{My }Documents\Remcom\XFdtd\scripts\menu`
on Windows and
`$HOME/Remcom/XFdtd/scripts/menu`
on Linux.

To add, remove or update a macro in the Macros menu, simply add, remove or edit the macro file in the folders listed in the Macros Menu preference. A menu item is added for each `.xmacro` file that when selected will execute the macro, and a submenu item is added for each subfolder that contains at least one `.xmacro` file. This allows convenient grouping and organization for large sets of macros.

By default, the menu item will not have an icon and will use the `.xmacro` file's name. However, this can be changed by adding a header to the macro file. The header format consists of zero or more comment lines with keyword:value pairs as documented below. All Remcom-provided macro files have this header and can be referenced for more examples.

```
//  Name: {menu name}
//  Icon: {name of .ico file}
//  Created: {date}
//  Version: {string}
//  Author:  {name}
//  Modified: {date and author}
//  Last Modified: {date and author}
//  Description:
//  {first line of description}
//  {next line of description}
//  ....
//  {last description line}
```

Each of `Name:`, `Icon:`, `Created:`, `Version:`, `Author:`, `Modified:`, `Last Modified:` and `Description:` are optional. The `Modified:` keyword can be specified more than once. If other keywords are specified multiple times, its value takes on the last one. `Authors:` is also recognized as a synonym for `Author:`. The specified icon file is usually a file in the same folder as the macro, but can also be specified with a relative or absolute filepath.

Submenus can also be given names, icons and descriptions by placing the information in a file named `menu.desc` in the submenu folder using the format described above.

Each menu ends with the item "Macro Information", which is a special menu item that will show a dialog with information about each of the macros and submenus of that particular menu. This is especially useful if each of the macros in the menu has a good description, since this menu item then serves as the "help text" for the macros in the menu.

Example:

```
//  Name: Markup S-Params Graph
//  Icon: Marker.ico
//  Created: Oct 1, 2010
//  Author:  J. Doe
//  Modified: Jan 22, 2011 by G.Buck
//  Modified: Nov 3, 2011 by T.Who
//  Description:
//  Set the bounds of an existing S-Parameters graph to specific,
//  values, add markers at a set of specified frequencies, and add
//  markers at specific dB down levels.  If more than one
//  S-Parameters graph exists, the user will prompted to select the
//  one to modify.  The specific value for the bounds and markers
//  are customizable in the first section of the macro file.
```

Any messages produced by the macro (output text or error messages) will appear in the lower pane of the Scripting Window.

Limitations:

- Only macro scripts (.xmacro), not function scripts (.xfunc), are currently supported.
- Macro Menu macros are currently not directly editable (in-place) in the application. However, they can be imported, edited and then exported.
- Macros should be fully debugged before using them in the Macros menu as it currently difficult to debug them because of the previous limitation. It is suggested that the macro be developed and tested as a Macro in a Project and then exported once it is stable.

This capability will be enhanced and expanded in the future as we receive customer feedback on its use.

4.1.5 Help Drop-Down Menu

Documentation Contains links to the XFdtd Reference Manual, User's Guide and online Scripting Documentation.

Support Opens a web browser page to the Remcom Support Portal.

Contact Remcom Opens a web browser page to the Contact Remcom page.

Command-line Help Displays command-line options that may be useful in some circumstances. For example, if XFdtd is taking a long time to start up because a library on a network drive is linked-to, the `--load-libraries=false` option can be used to prevent the library from be loaded.

Licensing Allows the display and configuration of XFdtd licensing information.

Tip of the Day Opens Tip of the Day, which offers many tips and trick for using XFdtd effectively. This is the same window that is opened on startup if the "Show Tip of the Day" checkbox is selected on the Application Preferences General tab.

About XFdtd Shows XFdtd version information as well as Third Party licenses. When contacted, Remcom support will usually ask for the version of XFdtd for which support is being requested. It is convenient to copy and paste the version information from this window or even to provide a screenshot of this window for that purpose. Additionally, the version number is always displayed in the titlebar of the application.

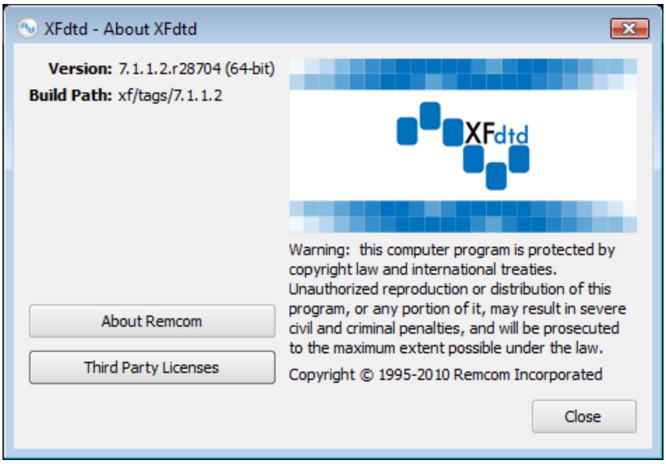


Figure 4.10: The About XFdtd Window

4.2 The Project Tree

The PROJECT TREE provides a tree-structured representation of the active project. It is organized into the following branches:

- PARTS
- CIRCUIT COMPONENTS
- EXTERNAL EXCITATIONS
- STATIC VOLTAGE
- SENSORS
- DEFINITIONS
- FDTD
- SCRIPTS
- GRAPHS
- GROUPS

This section describes in detail the branches and display of the Project Tree, as shown in Figure 4.11.

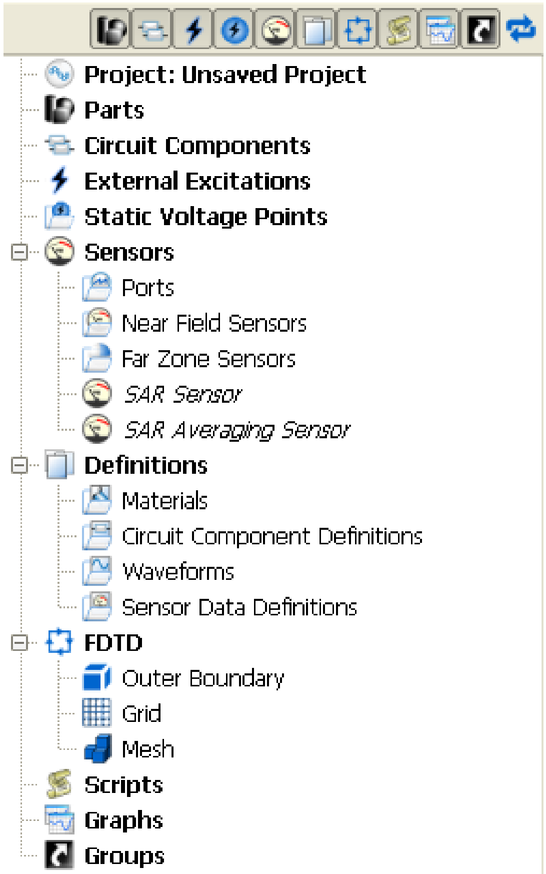


Figure 4.11: The Project Tree

4.2.1 The Branches of the Project Tree

Parts

The PARTS branch organizes the physical geometry of a project. It also lists material definitions and modeling operations applied to parts. To set up an ASSEMBLY, right-click on the Parts branch and select CREATE NEW: ASSEMBLY, as shown in Figure 4.12. Assemblies can also be created from existing parts by selecting the desired parts, right-clicking and choosing GROUP INTO ASSEMBLY. Any parts object or assembly can be exported to a CAD file using this right-click menu as well.

- See Section 5.3 for information about adding geometry parts to the project.

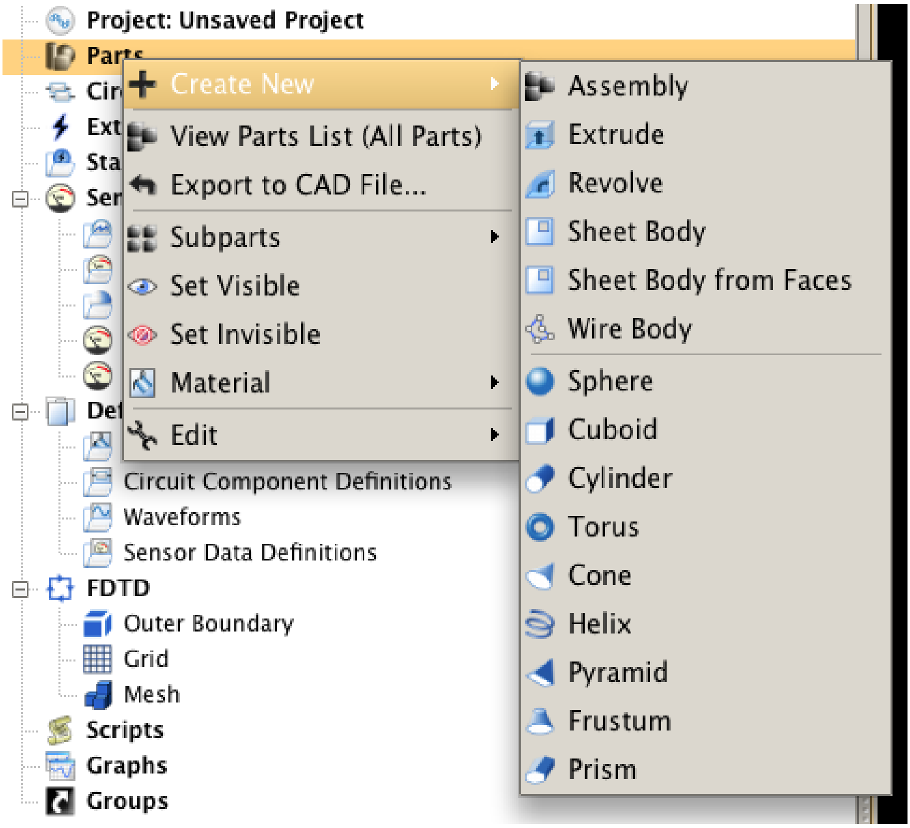


Figure 4.12: Adding a new Part or Assembly from the Project Tree

The PARTS LIST is also helpful for defining and organizing characteristics of geometric parts. To access this menu, right-click on any part or assembly in the PROJECT TREE and select VIEW PARTS LIST. A window will appear displaying each part for the current part or assembly selected. Using the right-click menu, it is possible to change the meshing priority, enable grid regions or fixed points, include the part in the mesh, change the visibility, and assign materials. The search bar also enables searching by selected fields.

✓ Mousing over a gray checkmark in the list will bring up a tooltip explaining why it is not fully solid.

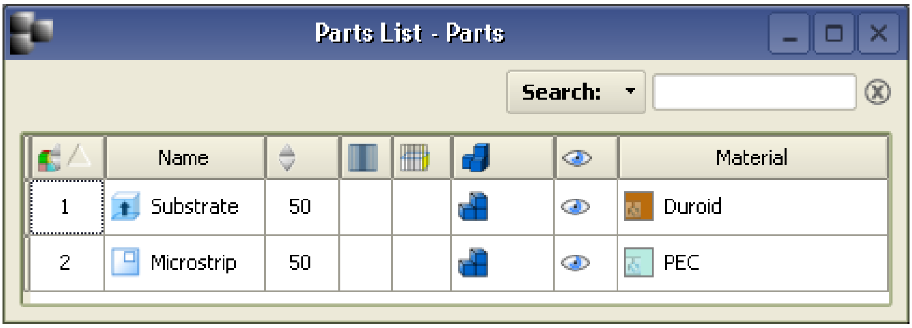


Figure 4.13: Viewing the Parts List

Circuit Components

The CIRCUIT COMPONENTS branch organizes discrete circuit components in a project. It also lists any associated CIRCUIT COMPONENT DEFINITION objects applied to a component. Figure 4.14 shows how to add a component by right-clicking in the Project Tree.

- See Section 7.1 for information about adding new components and associated tools.

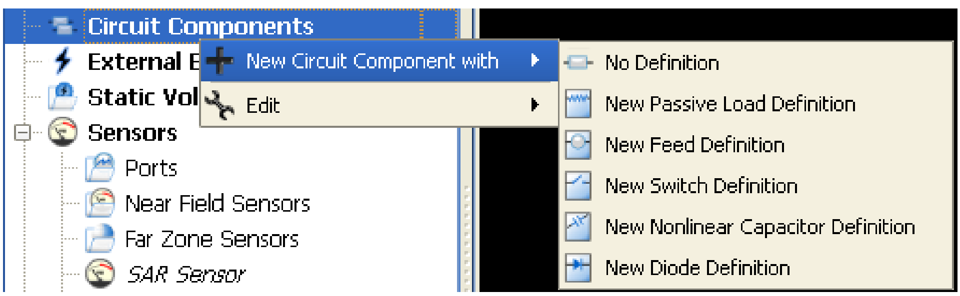


Figure 4.14: Defining a new Circuit Component from the Project Tree

External Excitations

The EXTERNAL EXCITATIONS branch organizes the external excitations (PLANE WAVES and GAUSSIAN BEAMS) applied to a project. An external excitation is not valid until it is associated with a WAVEFORM DEFINITION object. Once the association is made, the definition object is shown in the tree along below the external excitation. Figure 4.15 shows how to add a plane wave or Gaussian beam excitation by right-clicking in the Project Tree.

- See Sections 7.5 and 7.6 for information about defining external excitations using the PLANE WAVE EDITOR and GAUSSIAN BEAM EDITOR.

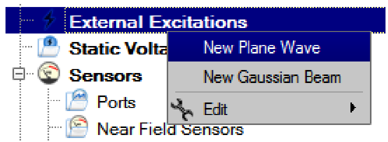


Figure 4.15: Creating a new External Excitation from the Project Tree

Static Voltage Points

The STATIC VOLTAGE POINTS branch organizes the static voltage points in a project. Figure 4.16 shows how to add a static voltage point by right-clicking in the project tree.

- See Section 7.8 for information about static voltage points and the static solver.

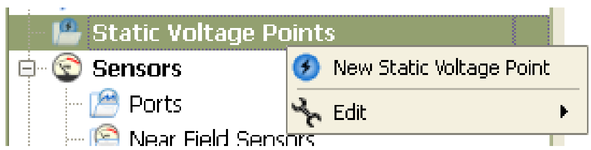


Figure 4.16: Defining a new Static Voltage Point from the Project Tree

Sensors

The SENSORS branch organizes the sensors defined in a project. Sensors are responsible for saving all data collected during a calculation. They are added by right-clicking on the appropriate branch of the Project Tree and choosing the desired sensor, as seen in Figure 4.17.

- See Section 10 for more information about sensors and related definitions.

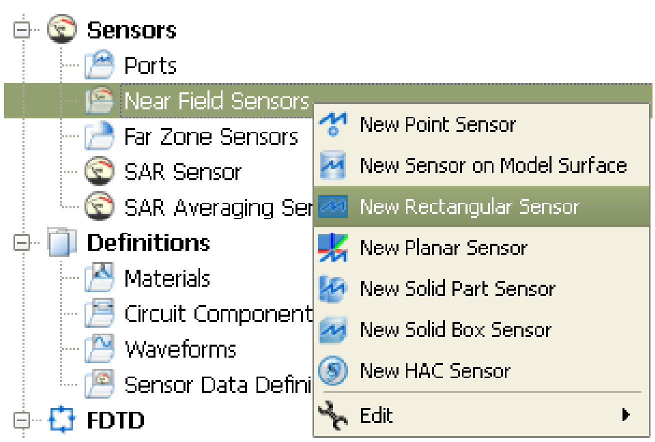


Figure 4.17: Adding a Near-Zone Sensor from the Project Tree

Definitions

The DEFINITIONS branch stores definitions that can be applied to or shared with other objects within the project. To add a new definition object, right-click on the branch and select the desired object, as seen in Figure 4.18. This definition is applied to other objects in the Project Tree by clicking and dragging the definition object onto the desired object. Several editors are accessible from this branch, depending on what type of definition object is selected.

- See Section 6.1 for more about the MATERIAL EDITOR.
- See Section 7.2 for more about the CIRCUIT COMPONENT DEFINITION EDITOR.
- See Section 7.7 for more about the WAVEFORM EDITOR.
- See Section 10 for more about several of the sensor editors.

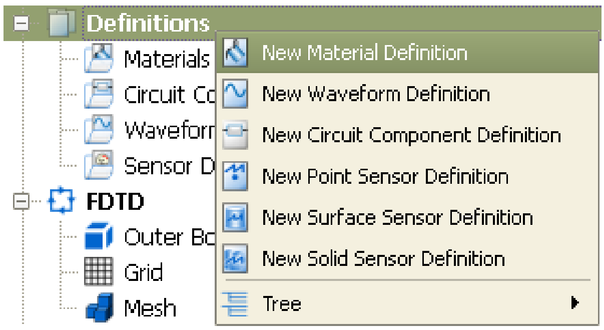


Figure 4.18: Accessing a Definition Editor from the Project Tree

FDTD

The FDTD branch stores definitions associated with the outer boundaries of the project as well as the grid and mesh, as seen in Figure 4.19.

- Double-clicking on the OUTER BOUNDARY icon will bring up the OUTER BOUNDARY EDITOR.
- Double-clicking on the GRID icon will bring up the GRID TOOLS dialog, used to specify the characteristics of the grid.
- Double-clicking on the MESH icon will toggle the MESH VIEW on or off.

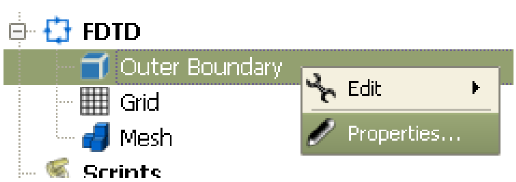


Figure 4.19: Viewing the FDTD branch

Scripts

The SCRIPTS branch stores user-defined scripts. Right-click on this branch to add a new script or to import an existing macro or function script to the project. After adding the script, right-click on the script object, as seen in Figure 4.20, to execute or edit the script in the SCRIPTING workspace window.

- For more information on scripting, see Section 13.2.

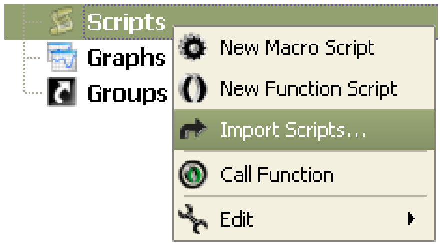


Figure 4.20: Accessing Scripting tools in the Project Tree

Graphs

The GRAPHS branch organizes the graphical output associated with data collected during a calculation.

- See Section 12.3 for more about viewing calculation results.

Groups

The GROUPS branch allows the user to create fully customizable shortcut groups that may include any grouping of objects (PARTS objects, SENSOR objects, DEFINITION objects, etc.). The Groups branch is a tool for users to conveniently organize their projects. It also contains automatically generated groups created by XFdtd to store deletions, additions, or other modifications to CAD files. To add a shortcut group, right-click on the branch as seen in Figure 4.21.

- For more about how to import CAD files, see Section 5.4.

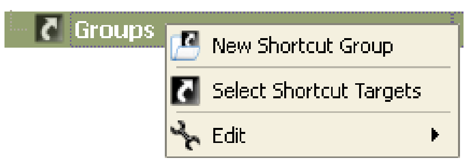


Figure 4.21: Adding a Group within the Project Tree

4.2.2 Customizing the Display of the Project Tree

The row of buttons above the PROJECT TREE toggle the branches of the tree on and off and simplify the display. This row is shown in Figure 4.22. Active branches are displayed as fully colored buttons (i.e.), and hidden branches are dulled (i.e.).

Figure 4.22: Toggling the branches of the Project Tree

Expanding and Collapsing the Project Tree

The PROJECT TREE is easier to navigate using the expand (+) and collapse (-) buttons, which toggle the visibility of each branch. Figure 4.23 demonstrates an expanded and collapsed branch in red.

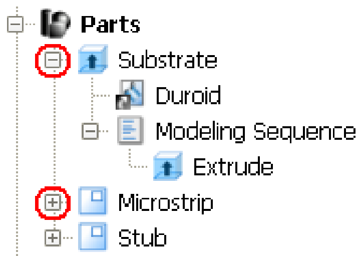


Figure 4.23: Expanding and collapsing individual branches in the Project Tree

Additionally, it is possible to expand or collapse all branches by right-clicking on any branch in the tree and selecting the TREE icon, as seen in Figure 4.24.

Figure 4.24: Expanding and collapsing all branches in the Project Tree

4.3 Workspace Windows

The XFdtd workspace windows are a series of windows that each has its own designated function. Each window is described below. This section also explains how to display active windows and how to control the display of workspace window buttons in the tabbed workspace.

Figure 4.25 shows the location of the XFdtd workspace window buttons.

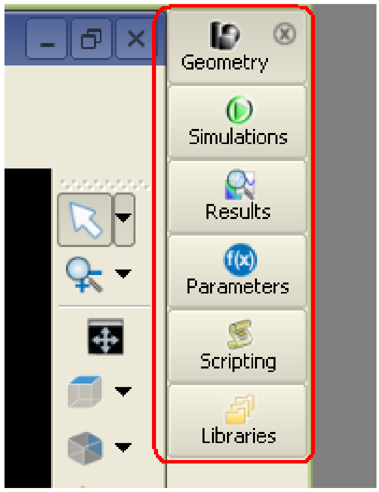


Figure 4.25: Workspace windows toolbar

4.3.1 Summary of Workspace Windows

Geometry

The GEOMETRY workspace window comprises the main project viewing area. The window contains four main tools used to add and edit the fundamental elements of a project:

- GEOMETRY TOOLS
- COMPONENT TOOLS
- GRID TOOLS
- SENSOR TOOLS

This window also contains buttons to manipulate the project view.

▶ The capabilities of this window are detailed in Chapter 5.

Simulations

The SIMULATIONS workspace window provides the main interface to define simulations to send to the calculation engine. Each time the project is modified and saved, the user must define a new simulation to register the change. This workspace window stores definitions such as source types, parameter sweeps, S-Parameters, frequencies of interest, scattered/total field interfaces and termination criteria that are specific to a calculation.

▶ Refer to Section 11.1 for a detailed discussion of setting up and running simulations within XFdtd.

Results

The RESULTS workspace window stores all of the results available for a particular project. It is also possible to load the results of a past project into this window without having to load the entire project itself. This makes it convenient to compare the results of several projects in a single interface.

- See Section 12.1 for more about this workspace window and reviewing results.

Scripting

The SCRIPTING workspace window allows users to create, edit, manage and execute user-defined scripts.

- Section 13.2 details the capabilities of this window.

Libraries

The LIBRARIES workspace window allows users to create and save collections of objects grouped by category. Because a library is saved to its own filename (and not to a specific project), it is easy to import common objects and definitions to multiple projects.

- Section 13.3 details the capabilities of this window.

Parameters

The PARAMETERS workspace window allows users to create, edit and delete parameters that are referenced universally throughout the project. A parameter can be a simple numeric value or a mathematical expression. They can be used to run "Parameter Sweeps," which are specified in the SIMULATIONS workspace window.

- Section 13.1 details the capabilities of this window.

4.3.2 Organizing Project Workspace

There are three ways to display and organize active windows in the project workspace: MAXIMIZED, CASCADE and TILE.

In MAXIMIZED mode, windows are maximized to take up the entire workspace. Selecting tabs in the tabbed workspace will bring the corresponding window to the front.

In CASCADE mode, the windows appear stacked on top of each other, each with its own title bar showing. Users can switch windows by clicking on the title bar, or by using the tabs at the top of the workspace. To activate this mode, select the Cascade mode icon in the lower right-hand corner, as seen in Figure 4.26.

Figure 4.26: Project workspace view modes

In TILE Mode, the workspace automatically sizes any non-minimized windows to fill up the space without overlapping. This is useful in order to quickly view all windows as large as possible. Any minimized windows will remain minimized. To assume this mode, select the Tile mode icon in the lower right-hand corner, as seen in Figure 4.26.

4.3.3 Tabbed Workspace Window Display

All of the workspace windows are easily accessible from the tabbed workspace to the right of the screen. Unless the application preferences are defined to show all windows, only active workspace windows are located in this workspace. Any additional workspace windows can be added to the active workspace by right-clicking and selecting the desired window, or by selecting VIEW in the APPLICATION MENU BAR. If a window is already open and present in the workspace, right-clicking in the tabbed workspace or using the VIEW menu to select a window will instead close the window. Similarly, the windows can be closed by clicking the button in the upper-right corner of the appropriate tab.

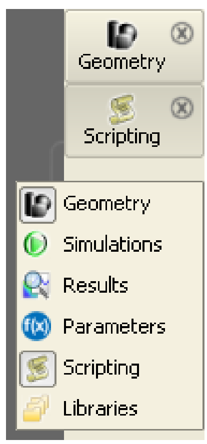


Figure 4.27: The tabbed workspace, displaying only Geometry and Scripting workspace tabs

4.4 Troubleshooting Invalid Operations within the XFdtd Interface

The XFdtd interface will display an error icon, , to indicate when any invalid definition is created within a project. This icon will appear in the dialog or editor window as the invalid definition is created, as well as in the PROJECT TREE beside the invalid object or definition. Additionally, an icon will appear next to the project's name at the top-level of the Project Tree to indicate globally that the project contains an invalid definition. A few of the most common invalid operations are discussed below, but in most cases the source of an error are easily pinpointed by holding the mouse over the ⚠ icon. A brief message will appear, summarizing the source of the error and the steps necessary to validate the project.

The ⚠ icon often appears when objects are created without applying the appropriate definitions. For example, the icon will appear if a PARTS object is created without applying a material definition, or if an EXTERNAL EXCITATIONS object is added without an applied waveform definition. In these cases, the error will be resolved by dragging and dropping the appropriate DEFINITIONS object (if one exists) onto the invalid object in the tree.

Other common errors occur due to invalid or conflicting grid definitions. If grid definitions are assigned that cause the project to exceed the maximum number of cells or memory size (defined in the LIMITS tab of the GRID TOOLS dialog), then the ⚠ icon will appear. In this case, the limit may need to be increased or removed or else the number of cells in the project will need to be reduced by increasing their minimum cell size. XFdtd will also generate an error if the MAXIMUM CELL STEP FACTOR defined in the LIMITS tab is too low to create valid transition regions or if a MINIMUM cell size definition is specified anywhere in the interface that falls below the universal MINIMUM CELL SIZE definition in this tab.

- For more about grid definitions and associated invalid operations, see Chapter 8.

Chapter 5

Creating Geometry

In this chapter, you will learn...

→ how to create, edit and import the geometry of your XFdtd project

→ how to align your geometry and view it from any angle

This chapter will focus on geometric modeling within the XFdtd interface. It begins by describing the basic functions available within the GEOMETRY workspace window, where the project geometry is created. The VIEW TOOLS buttons are detailed in the following section.

Under GEOMETRY TOOLS, the reader is introduced to the 2-D and 3-D modeling tools, modification and boolean operations, and patterned arrays that are available within this dialog. In addition to creating geometry from scratch within Geometry Tools, external files, such as CAD, voxel and mesh files can also be imported and modified.

After the geometry is created or imported into a project, it is often necessary to adjust an object's orientation. XFdtd has the capability of orienting not only geometric parts but also other physical parts, such as components and sensors. The final section of the chapter details the SPECIFY ORIENTATION tab that is available during any editing session involving a physical part.

5.1 Geometry Workspace Window

The GEOMETRY workspace window provides the graphical interface of the XFdtd project. Within the window, there is a series of VIEW TOOLS along the right of the viewing area that can be used to manipulate the view of the simulation space. Along the top of the viewing area, the window contains GEOMETRY TOOLS used to create and edit various aspects of the project geometry.

The first drop-down menu at the top of the GEOMETRY workspace window contains four tools:

- GEOMETRY TOOLS
- COMPONENT TOOLS

- Grid Tools
- Sensor Tools

Later sections in this chapter will describe the operations available in XFdtd within Geometry Tools. Later chapters provide a more detailed discussion of the remaining tools.

- For more about creating discrete components with Component Tools, see Section 7.1.
- For more about defining grid and mesh characteristics with Grid Tools, see Section 8.1.
- For more about collecting data in XFdtd with Sensor Tools, see Section 10.1

Figure 5.1 shows the Geometry workspace window.

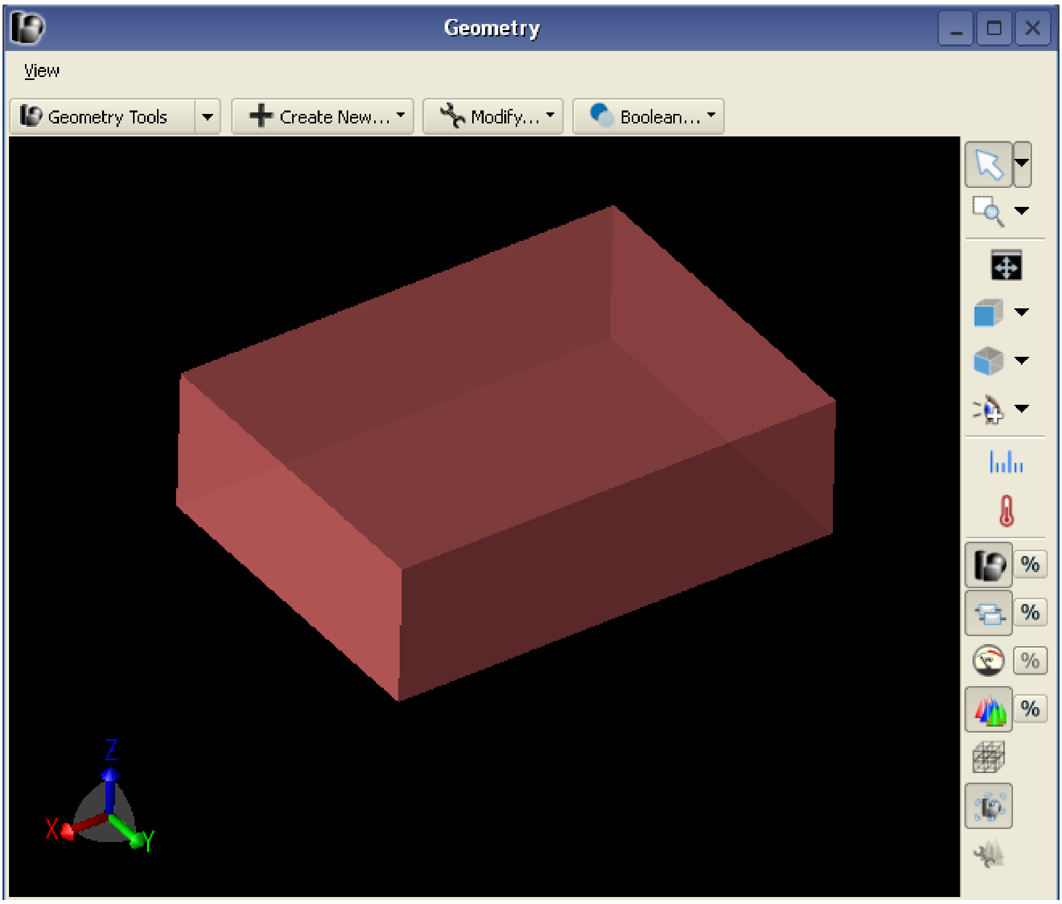


Figure 5.1: The Geometry workspace window

5.2 View Tools

The VIEW TOOLS are used to alter the perspective of the viewing window by manual rotation, translation and zoom, as well as automatic orientations to achieve the desired perspective. The View Tools are visible on the right-hand side of the GEOMETRY workspace window. They can also be found in the top left-hand corner under the VIEW drop-down menu, as seen in Figure 5.2.

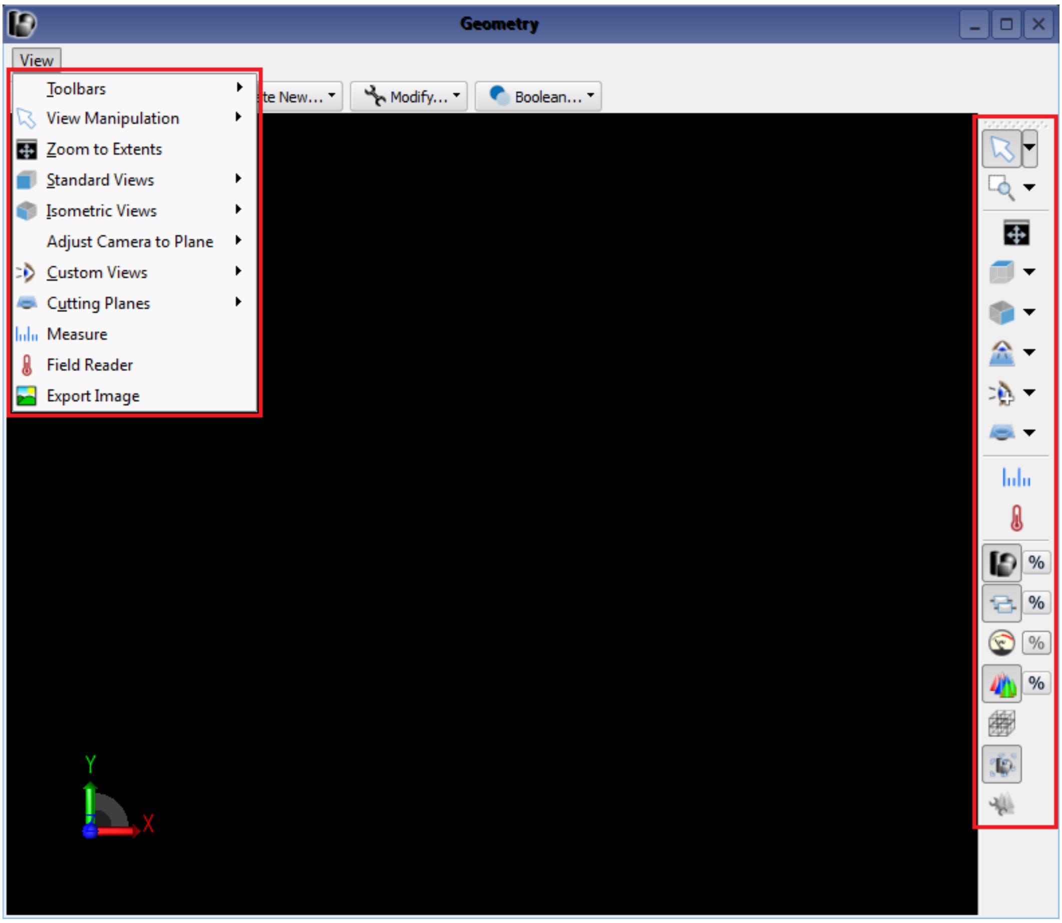


Figure 5.2: The View Tools

The VIEW TOOLS may be hidden by right-clicking on the toolbar and unselecting the toolbar check-box. The toolbar can be unhidden using the VIEW drop-down menu. Each tool is detailed below.

Select

The SELECT tool is the default tool in the GEOMETRY workspace window. It is used to select objects as well as manipulate the view of the simulation space.

- Rotation about a fixed point:

  - Left-click and drag.
  - Click the mouse wheel and drag.
- Translation (panning):
  - Right-click and drag.
  - Hold SHIFT, left- or right-click and drag.
- Zooming:
  - Roll the mouse wheel backwards to zoom in or forwards to zoom out.
  - Hold CTRL, left-click and drag the mouse up to zoom in or down to zoom out.
  - ▶ You can reverse the zoom direction by setting a property under the MODELING tab of APPLICATION PREFERENCES, as described in Section 4.1.2.

Orbit

The ORBIT tool performs rotation of the simulation space through left-clicking and dragging.

Pan

The PAN TOOL tool performs translation of the simulation space through left-clicking and dragging.

Zoom

Zoom-in or zoom-out of simulation space by left-clicking and dragging the mouse up or down.

Zoom to Window

Zoom into a rectangular shaped area of the geometry as specified by the user. To use, select the tool, then left-click and drag the mouse to designate the rectangular zoom area.

Zoom to Extents

Select this tool to automatically zoom and bring the entire geometry into view.

Standard View, Isometric View, and Custom View

The STANDARD VIEW, ISOMETRIC VIEW, NORMAL VIEW and CUSTOM VIEW buttons function to automatically change the perspective of the objects in the GEOMETRY workspace window. Each button has a drop-down list with several available perspectives; the list is shown by clicking the down arrow immediately to the right of the button. The first of these buttons changes the view to the following orientations:

View from +Y (Front)	View from -Y (Back)
View from +Z (Top)	View from -Z (Bottom)
View from +X (Right)	View from -X (Left)

The second button changes the view to the following orientations:

View from +X / +Y / +Z	View from -X / +Y / +Z
View from +X / +Y / -Z	View from -X / +Y / -Z
View from +X / -Y / +Z	View from -X / -Y / +Z
View from +X / -Y / -Z	View from -X / -Y / -Z

The third button is used to adjust the camera position so it is perpendicular to the cutting/sketching plane, viewing either from the top or the bottom of the plane.

Top of Plane   Bottom of Plane

Pressing CTRL and clicking on any of these button iterates to the next perspective, while pressing SHIFT and clicking on the button rotates the view around the axis normal to the screen.

If these buttons do not achieve the desired perspective, use the Select, Orbit or Pan tools to customize the orientation, and save the desired view by clicking the Add View button. Multiple custom orientations can be saved in this way and restored by selecting them from the button's drop-down list.

For all of the buttons described above, the last-used view is remembered so that when that button is clicked again, that view is the one that is shown.

Cutting Plane

Use this tool to toggle the cutting plane on and off. The cutting plane will appear in the geometry view as a translucent blue rectangle sized which is proportional to the bounds of the geometry. The cutting plane menu provides options for reorienting the cutting plane, saving the current cutting plane's orientation and for recalling previously saved orientations.

When the cutting plane is active, it visually clips all geometry in the positive Z direction. Solid body, sheet body and wirebody geometry that is clipped by the plane form edges and vertices that can be used for snapping. The sketcher provides a snapping tool to toggle this behavior. Point, direction and plane picking tools also snap to these locations.

Measure Tool

This tool measures the 3-D distance between any two points by left-clicking on a starting point and dragging to an ending point. A box in the lower-right corner of the GUI displays the coordinates of the cursor position in 3-D space. A box in the lower-left corner of the GUI displays axis-aligned distances.

Figure 5.3 shows the Measure Tool calculating the distance between corners of two rectangles and the location of the cursor in world coordinates.

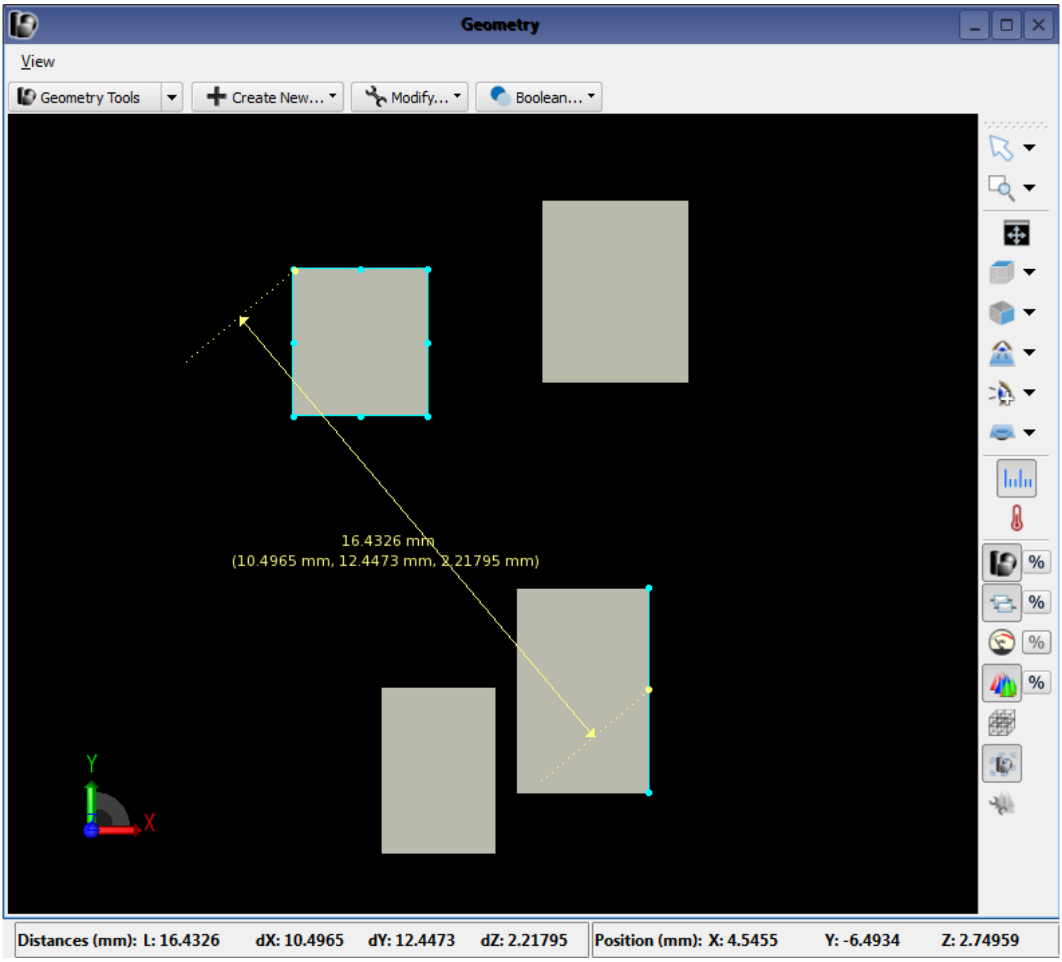


Figure 5.3: Using the Measure Tool

Field Reader Tool

This tool measures field values at the location where the mouse hovers over the geometry.

- See Section 12.4.3.2 for more on the field reader tool.

Export Image

This action takes a screen shot of the geometry as it is currently shown in the GEOMETRY workspace window and saves it to a specified location.

The global shortcut key combination *Ctrl+Alt+F12* can also be used to initiate this action. A screen shot of the Geometry View or a Graph Window, whichever is "on top" in the application workspace, will be saved to a file. If neither the Geometry View nor a Graph Window is on top, no action will be taken.

- This action is only available from the VIEW menu.
- The global shortcut key combination *Ctrl+Alt+F11* is also defined to save a screen shot as described above to an automatically-named file in the folder defined in the Application Preferences under "Screen Capture."

Copy Image to Clipboard

This action takes a screen shot of the geometry as it is currently shown in the GEOMETRY workspace window and saves to to the clipboard for pasting into other applications.

The global shortcut key combination *Ctrl+Alt+F10* can also be used to initiate this action. A screen shot of the Geometry View or Graph Window, whichever is "on top" in the application workspace, will be copied to the clipboard. If neither the Geometry View nor a Graph Window is on top, no action will be taken.

- This action is only available from the VIEW menu.

% Opacity and Visibility Tools

The VISIBILITY buttons control the view of parts of the project. Clicking any of these buttons will hide its corresponding objects. They include:

- PARTS VISIBILITY - Toggles the geometric parts on and off.
- COMPONENTS VISIBILITY - Toggles the circuit components on and off.
- SENSORS VISIBILITY - Toggles the sensors on and off.
- OUTPUT VISIBILITY - Toggles the output result fields on and off.

Clicking the %OPACITY button to the right of any of these will bring up a slider to customize the translucency of its objects. The sliders change the alpha of the objects, making them more or less translucent as the slider is dragged right or left, respectively. When the project is in Mesh View mode, these buttons are convenient for turning off the view of the solid geometry so that the view of the cell edges is not obstructed.

- There are several ways XFdtd can render this translucency. See the notes on TRANSPARENCY ALGORITHM in Section 4.1.2 for more on how to adjust these settings.

Mesh View

This button toggles the MESH VIEW on and off. Alternatively, double-clicking on the FDTD: MESH branch of the PROJECT TREE will enable Mesh View.

When in Mesh View, there are two main viewing modes, MESH CUTPLANES and 3-D MESH, that are controlled by radio buttons along the bottom of the GEOMETRY workspace window. A valid mesh must be generated to use these viewing options.

- See Section 8.3 for more on generating a mesh with the MESHING PROPERTIES EDITOR.

The first mode, MESH CUTPLANES, creates cutplanes of the mesh in any or all of the three primary planes. Toggle any of these cutplanes on or off by checking or unchecking their respective boxes. The sliders associated with each of these planes are enabled when its respective plane is turned on. The slider moves the cutplane throughout the slices in the mesh. Additionally, each checked plane will activate the following icons, which aid in manipulating the cutplanes view:

- Displays all of the geometry/mesh above the cutplane.

- Displays all of the geometry/mesh below the cutplane.

- Toggles on and off the view of the mesh that occurs normal to the cutplane.

- Toggles the view of the mesh cutplane on and off.

E - Toggles the view of the electric components on and off.

H - Toggles the view of the magnetic components on and off.

There are also two checkboxes available:

VIEW MESH INFORMATION - Displays a dialog box with information about the mesh at the location of the cursor. XFdtd highlights the exact gridline on which the cursor rests (Figure 5.4).

- Mesh Edge
  * Index
  * Length
  * Type
- Grid Node Position (mm)
  * X
  * Y
  * Z
- Mouse Position (mm)
  * X
  * Y
  * Z
- Material

SYNCHRONIZE SLIDERS - Moves the cutplane simultaneously with a slider adjustment while the mouse button is still pressed.

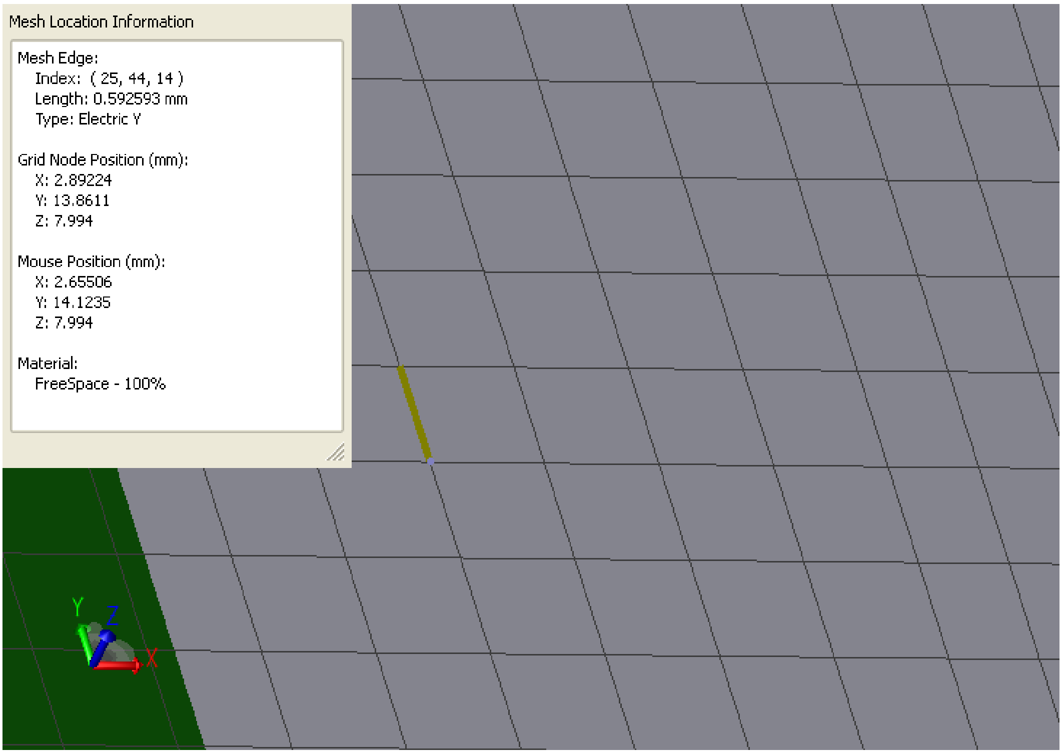


Figure 5.4: Viewing XFdtd mesh information

Figure 5.5 shows this first mode that is displayed when the MESH VIEW icon is pressed. Note that this is only a preview of the mesh when it is shown while editing the grid within the GRID TOOLS dialog. Any other time, it is a representation of the most recently generated mesh.

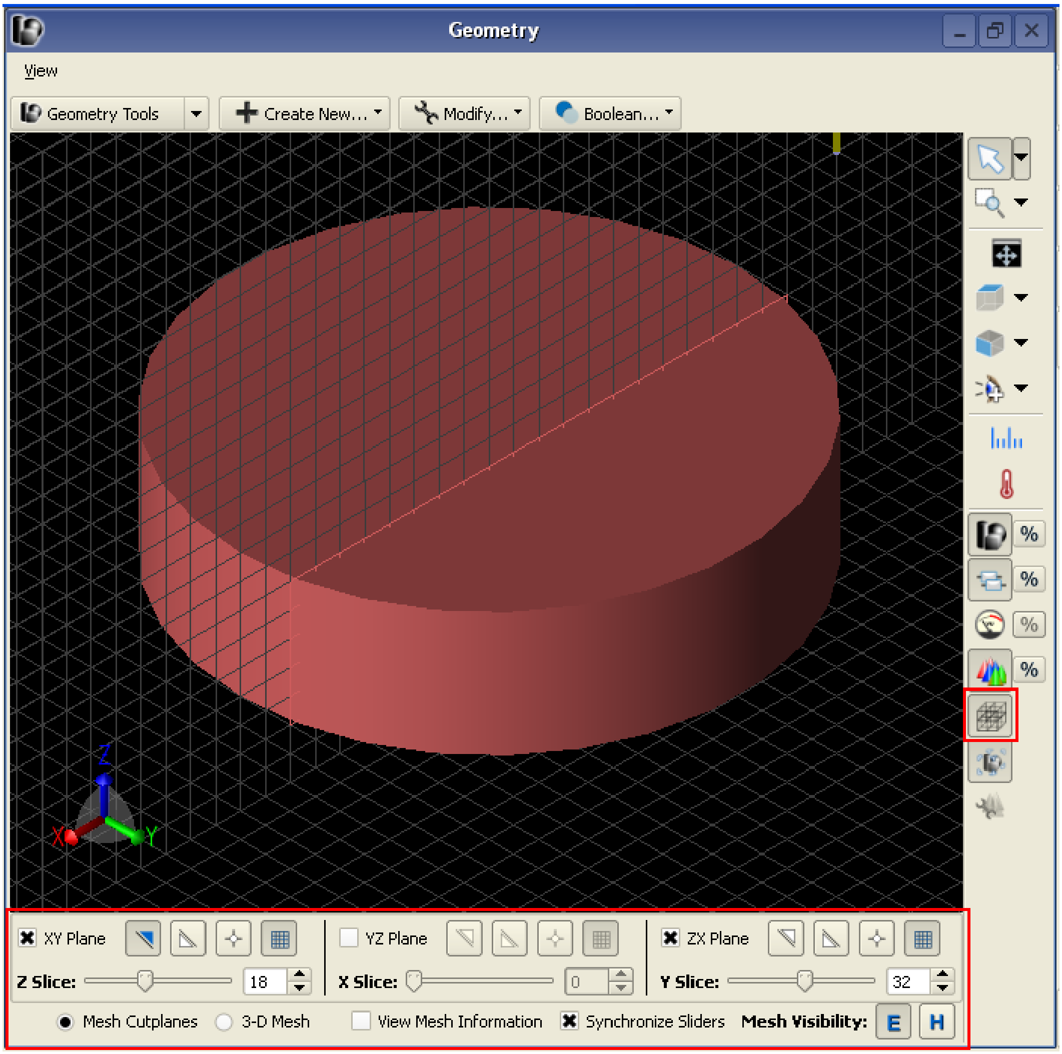


Figure 5.5: Viewing the XFdtd mesh

The second mode, 3-D MESH, provides several different options to view the mesh. Select any of the radio buttons within this option to create a customized three-dimensional view of the mesh. The MESH VISIBILITY icons are available in this mode as well so that E ELECTRICAL and H MAGNETIC components of the mesh may be toggled on and off.

Figure 5.6 shows an object in 3-D MESH mode with ALL EDGES displayed.

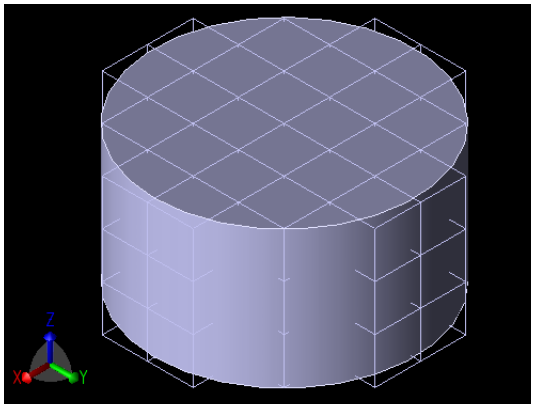


Figure 5.6: Viewing a 3-D mesh

Meshing Options

The AUTOMATIC REMESHING feature is accessed by right-clicking on the MESH icon in the PROJECT TREE. When this feature is enabled, remeshing is performed any time a change is made to the geometry. If this feature is not enabled, remeshing must be performed manually. Automatic remeshing may not be desirable when large meshes are imported because of their memory and performance demands.

Figure 5.7 shows the Automatic Remeshing dialog.

Figure 5.7: The Automatic Remeshing dialog

Toggle Bounding Box Visibility

This button toggles the visibility of the bounding box for the geometry when the geometry is selected.

Toggle Output Viewing Controls

This button toggles the visibility of the output viewing controls for sensor results.

5.3 Geometry Tools

XFdtd 7 differs from earlier versions in its creation of geometric objects. It constructs geometric objects with a series of repeatable actions or *Features*, collectively known as a *Modeling Sequence*. This approach is known as “Feature Based Modeling”, and provides more flexibility in customizing an object. Any unwanted step can be easily undone by suppressing that feature rather than having to rebuild the entire object without the unwanted step. Every feature in the modeling sequence is represented as a separate object in the tree to facilitate even simpler additions, deletions and modifications to the modeling sequence.

This section describes the GEOMETRY TOOLS interface, where geometric modeling in XFdtd is performed. This interface enables the user to create new geometry, modify existing geometry, perform boolean operations and create patterns. To begin using Geometry Tools, open the GEOMETRY workspace window and select Geometry Tools from the drop-down menu.

- A more comprehensive discussion of each Geometry Tool is available in Appendix A.

5.3.1 Creating New Geometry

After selecting GEOMETRY TOOLS, click CREATE NEW to prompt a drop-down menu to appear. This menu includes the following modeling operations:

ASSEMBLY	EXTRUDE	REVOLVE
SHEET BODY	SHEET BODY FROM FACES	WIRE BODY
SPHERE	CYLINDER	CONE
CUBOID	TORUS	PYRAMID
PRISM	FRUSTUM	HELIX

Figure 5.8 shows the CREATE button that displays when GEOMETRY TOOLS is selected.

- For more on defining extrusions or revolutions, refer to Appendix Section A.4.
- An image of each boolean operation is available in Appendix Section A.7.

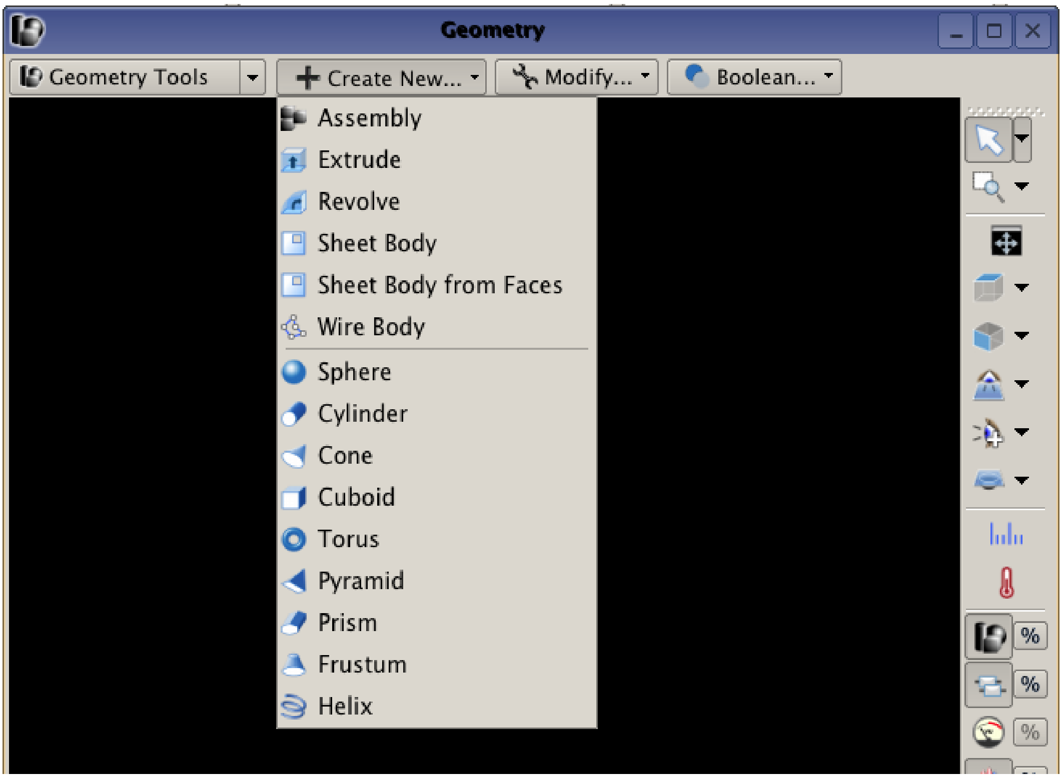


Figure 5.8: The Create menu of the Geometry workspace window

Additionally, these tools are accessible from the Project Tree by right-clicking on the PARTS branch, as seen in Figure 5.9.

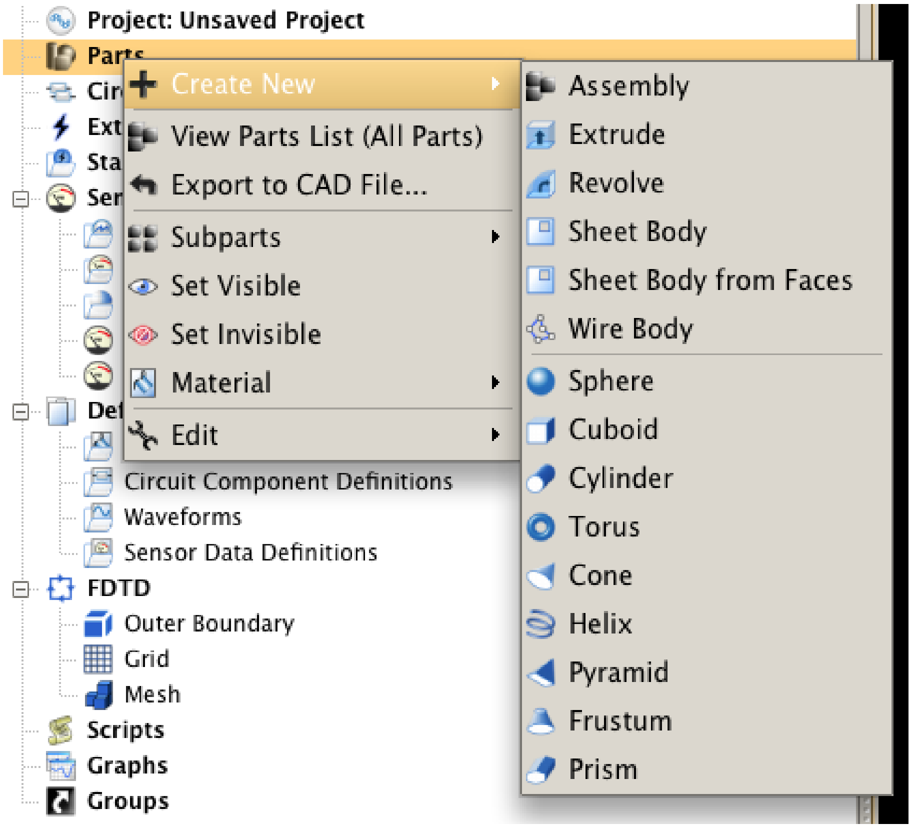


Figure 5.9: Accessing the Create menu from the Project Tree

Selecting any of these operations will prompt a similar series of cross-sectional editing tools.

5.3.1.1 Specify Orientation Tab

The SPECIFY ORIENTATION tab provides tools for orienting geometric parts in the simulation space.

- For a detailed discussion of this tab, refer to Section 5.5.

5.3.1.2 Edit Cross Section Tab

The EDIT CROSS SECTION tab, seen in Figure 5.10, contains four toggle buttons: SHAPES, TOOLS, CONSTRAINTS and SNAPPING. Each toggles its own toolbar with corresponding operation buttons below. These tools are also available in menus next to the VIEW drop-down menu in the upper-left corner.

Additionally, a menu under the CONSTRUCTION GRID button is available to edit the spacing of the visible grid lines in the 2-D sketcher. (This has no impact on the FDTD grid definition.)

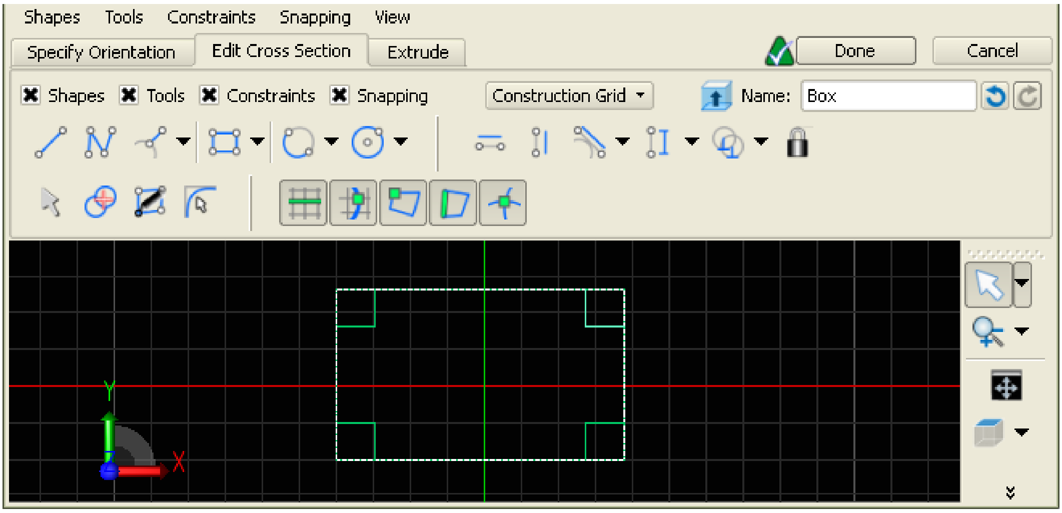


Figure 5.10: The Edit Cross Section tab of the Geometry workspace window

The current object is named in the box labeled NAME in the upper right. If a name is not defined in this box, the object is assigned a default name when it is added to the project. To the right of the NAME dialog box are two buttons: UNDO and REDO. Clicking the Undo button will undo actions carried out in the EDIT CROSS SECTION tab. Similarly, the Redo button will repeat any actions erased during an undo operation.

Shapes. The SHAPES sketching tools are used to create simple 2-D geometries for wire bodies and sheet bodies. They also serve as a starting point to define 2-D cross sections for 3-D bodies such as extrusions, revolutions, and more complicated solid modeling operations.

- ✓ Pressing BACKSPACE will back up one step when using a multi-step creation tool.
- ✓ Pressing ESC once will cancel the shape, and pressing it a second time will deactivate the edge creation tool and activate the default SELECT tool.
- ✓ Pressing TAB will bring up a dialog to specify the position.

The following is a list of the 2-D shapes available in XFdtd.

STRAIGHT EDGE	POLYLINE EDGE
PERPENDICULAR EDGE	TANGENT LINE
RECTANGLE	POLYGON
N-SIDED POLYGON	3-POINT ARC
ARC CENTER, 2 POINTS	2-POINT ARC
CIRCLE CENTER, RADIUS	3-POINT CIRCLE
2-POINT CIRCLE	ELLIPSE

▶ For a detailed description of each shape tool, see Appendix Section A.2.1.

Tools. The TOOLS buttons provide useful functionality to users while sketching in the 2-D sketcher.

- SELECT/MANIPULATE
- TRIM EDGES
- INSERT VERTEX
- FILLET VERTEX

▶ For a detailed description of each 2-D sketcher tool, see Appendix Section A.2.2.

Constraints. The CONSTRAINT tools place restrictions on geometric parts that must be satisfied in order to consider the model valid. They ensure that the user's intent is sustained throughout a calculation when parameters may change. Applying a constraint to an object will often affect other characteristics of the object. For instance, applying a horizontal constraint to one side of an irregular quadrilateral will most likely change the length of one or more sides and the angles that form with those connecting sides. Thus, it is important to lock any points that are intended to stay static by:

- selecting the LOCK CONSTRAINT tool and clicking on the the appropriate vertex or side.
- selecting the SELECT/MANIPULATE tool, right-clicking on the appropriate vertex or side and selecting LOCK POSITION, as seen in Figure 5.11.

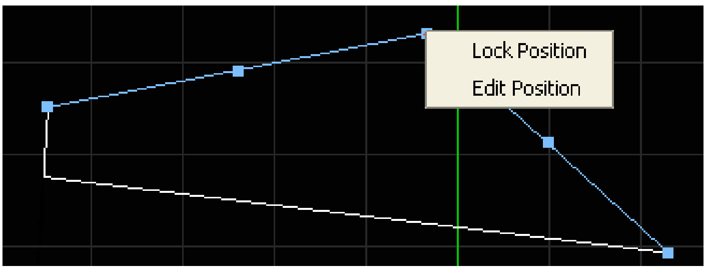


Figure 5.11: Locking or editing a vertex's position with the Select/Manipulate tool

▶ See Appendix Section A.2.2 for more about the SELECT/MANIPULATE tool's functionality.

Each type of CONSTRAINT tool has its own symbol or letter that is visible when the mouse is held over the constrained segment.

- HORIZONTAL
- VERTICAL
- COLLINEAR
- PARALLEL
- PERPENDICULAR
- TANGENT
- CONCENTRIC
- ANGLE
- DISTANCE
- EQUAL LENGTH
- EQUAL DISTANCE
- RADIUS
- EQUAL RADIUS

▶ For a detailed description of each constraint, see Appendix Section A.2.3.

Snapping. Snapping tools are available to facilitate the exact placement of vertices on the sketching plane. When snapping is enabled, the mouse will be "snapped" to the closest of one or more snapping landmarks if one comes within range. For example, if SNAP TO GRID LINES is selected, the mouse is snapped to points on the closest grid line as it is moved around in the sketching plane. This makes it much easier to place a vertex without having to zoom in to a discrete position.

Several snapping options can be selected at a time, in which case, the vertex will be snapped to the closest landmark that is within range of the mouse. Blue dots and blue lines represent the snapped location of the mouse when snapping is enabled. In the case that the mouse is not within sufficient range of a selected landmark listed below, a vertex will be placed at its exact location on the sketching plane as if snapping were not turned on.

- Snap to Grid Lines
- Snap to Grid/Edge Intersections
- Snap to Vertices
- Snap to Edges
- Snap to Edge/Edge Intersections
- Snap to Part Cross Sections

▸ For a detailed description and image of each, see Appendix Section A.2.4.

Customizing the Construction Grid. The Construction Grid drop-down controls the spacing and appearance of the visible grid lines in the 2-D sketcher. Changes made with this tool do not affect the properties of the grid (fixed points, grid regions, target cell sizes, etc.).

- Automatically adjust line spacing adjusts the line spacing with the current zoom level. The lines move closer together as you zoom in and further apart as you zoom out.
- Line spacing is available when automatic isn't checked. This is the spacing between adjacent lines of the construction grid.
- Highlight interval emboldens every Nth line of the grid.
- Mouse spacing controls the minimum resolvable distance by the mouse. As you move the mouse, you will be unable to move between two points closer than this specified distance.

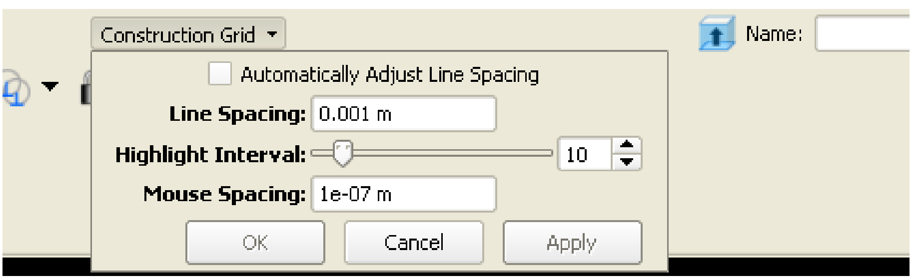


Figure 5.12: Editing the Construction Grid

5.3.1.3 3-D Operation Tabs

If subsequent tabs are available to the right of the Edit Cross Section tab, then continue on to complete a 3-D operation. (These tabs are not available for 2-D objects.) Figure 5.13 shows the Advanced drop-down menu inside of the Extrude tab, available when an Extrude operation is selected. This menu contains operations that can be applied to the 3-D object.

▸ See Appendix Section A.5 for more on each of these operations.

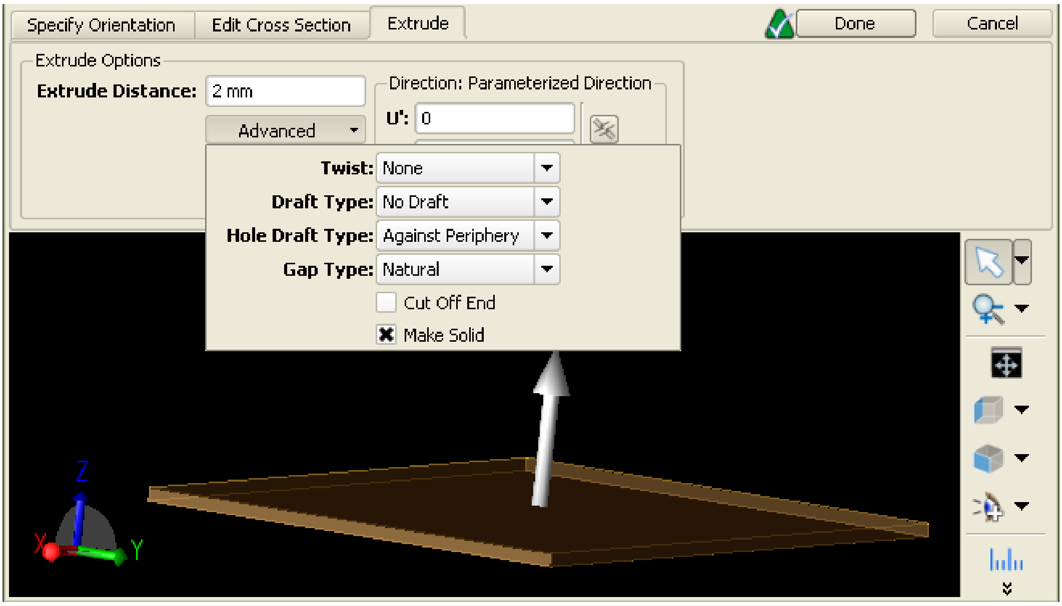


Figure 5.13: The Extrude tab of the Geometry workspace window

5.3.2 Modifying Existing Geometry

The MODIFY button in the GEOMETRY workspace window may be selected to modify the geometry of existing objects in the project.

- SPECIFY ORIENTATION
- BLEND EDGES
- OFFSET SHEET EDGES
- LOFT FACES
- OFFSET FACES
- STRETCH
- CHAMFER EDGES
- SHELL
- THICKEN SHEET
- REMOVE FACES
- TWIST

▶ See Appendix Section A.6 for images of each of these operations.

The TRANSFORM submenu allows an existing object to be transformed by any of the following operations:

- SCALE
- REFLECT
- TRANSLATE
- SHEAR
- ROTATE

▶ Refer to Appendix Section A.9 for a description of each transformation.

The CREATE PATTERN submenu presents options for replicating a single selected object multiple times in one of the organized arrangements listed below.

- LINEAR/RECTANGULAR
- CIRCULAR/ELLIPTICAL

▶ Refer to Appendix Section A.8 for the definitions and images associated with these patterns.

5.3.3 Boolean Operations

The following boolean operations are available in XFdtd:

EXISTING PARTS   EXTRUDE   REVOLVE

The EXISTING PARTS tool provides several boolean operations to subtract, intersect, unite or chop multiple objects. For these operations, users must select at least one object to be the BLANK and at least one object to be the TOOL which acts on the blank. If necessary, users may select multiple objects to be either the BLANK or the TOOL. Users may also save their original tools and blanks by selecting the appropriate checkbox (KEEP ORIGINAL TOOLS or KEEP ORIGINAL BLANKS) when choosing a boolean operation.

✓ The Existing Parts tool is also available through the PROJECT TREE by selecting the desired parts, right-clicking and choosing EXISTING PARTS.

It is also possible to EXTRUDE or REVOLVE holes through any part(s) with its respective tool in this menu. One or more objects are selected in the PICK BLANK tab and the cross section of the hole is sketched and oriented in the EDIT PROFILE and FEATURE ORIENTATION tabs, as described in Sections 5.3.1.2 and 5.5, respectively. Then the shape of the removed section is specified in the EXTRUDE BOOLEAN tab or REVOLVE tab depending on which operation is selected. The PREVIEW tab shows a preview of the object before the changes are formally applied to the project.

- For more on defining extrusions or revolutions, refer to Appendix Section A.4.
- An image of each boolean operation is available in Appendix Section A.7.

5.4 Importing CAD, Voxel, Mesh and PCB Files

In order to enhance the geometric modeling process, XFdtd has the ability to load CAD model, voxel data and mesh objects.

5.4.1 CAD files

CAD Import Options

The CAD Importer is used to import CAD files from many popular modeling packages into XFdtd for use in simulations. To import a CAD file, simply select
FILE > IMPORT > CAD FILE(S) and load the desired file.

After the user selects the CAD file to load, a dialog box with several important options will appear, as seen in Figure 5.14.

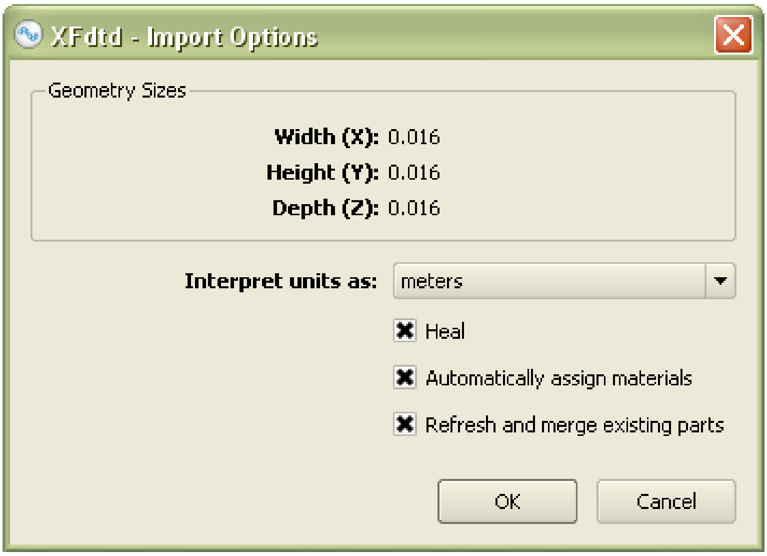


Figure 5.14: CAD import options

- The INTERPRET UNITS AS drop-down list assigns the units to the CAD file after it is imported.
- The HEAL check-box will check imported files for errors and correct them as needed. In particular, objects imported from IGES files and STEP files may have errors. For more complex objects, this can be a time consuming process and pop-up window will display the progress of the operation.
  - ✓ You can also heal a CAD object after it is imported by right-clicking on the object in the PROJECT TREE and selecting HEAL.
- The AUTOMATICALLY ASSIGN MATERIALS check-box appears after the external CAD files have been read. When this option is selected, color information will be extracted from the imported parts if it is available. If a material exists in the project that has the color of the imported part, that material is assigned to the part. If no material is found, and a color is available, a new material is created and assigned.
- The REFRESH AND MERGE EXISTING PARTS check-box should be checked when the user has already loaded a CAD file into an XFdtd project and desires to update it with a newer (external) version of the source file. When this option is selected:
  - Parts that are *used* within your XFdtd project will be updated with any geometrical changes present in the newly imported CAD file.
  - Parts that are *new* to the imported CAD file are added to the project.
  - Parts that have been *deleted* in the imported CAD file, but are still present in the XFdtd project, will remain in the project without change.
  - Parts that have been *deleted* in the XFdtd project, but are still present in the CAD file, will be added to the XFdtd project with MESHING DISABLED and its VISIBLE property unchecked.
  - Parts that *were found* in the project prior to the CAD import remain, but the new CAD database places the parts in a new parent assembly hierarchy.

– Parts that *were* in a former parent assembly are identified, as is the assembly to which they used to belong.

✓ In each of the above six cases, all changes from the original files will be documented in a shortcut group in the GROUPS branch of the PROJECT TREE. The following shortcut groups are available:

- **New to CAD Database**
- **No longer in CAD database**
- **Present in CAD database, but previously deleted**
- **Parts that moved or changed size**
- **Parts in a new parent assembly**
- **Parts in a former parent assembly**

Modifying a CAD File

Once imported, an assembly containing all of the parts of the CAD file is added to PARTS branch of the PROJECT TREE. Since every part of the CAD file is treated as its own separate object, all available modeling operations can be applied to any individual object imported from the file. Selecting an operation in the MODIFY drop-down box within GEOMETRY TOOLS will allow the user to select any part to modify.

Figure 5.15 shows an imported CAD object before a modification operation is applied. Figure 5.16 shows the CAD object after an OFFSET FACES operation is applied to one of its parts, and the resulting MODELING SEQUENCE object that is added to the tree.

▶ Also see Section 8.4 for important considerations when meshing imported CAD objects.

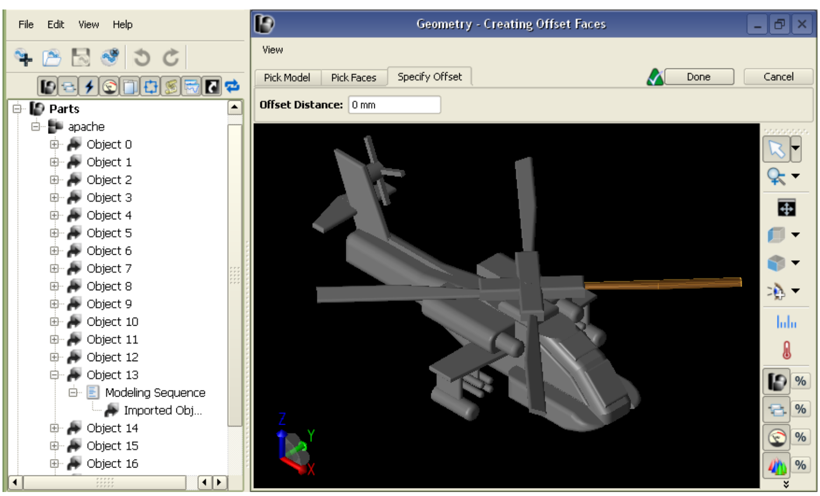


Figure 5.15: CAD file before modification

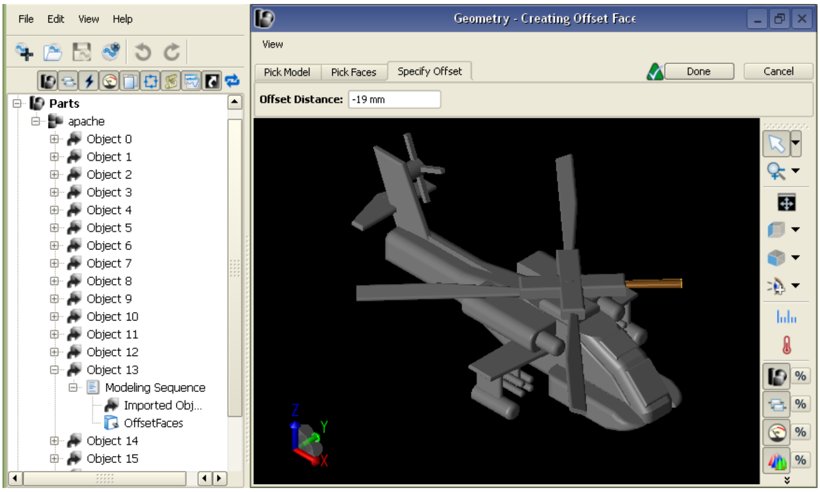


Figure 5.16: CAD file after modification

5.4.2 Voxel Data

In XFdtd, voxel objects are geometric parts defined by volumetric elements ("voxels") on a regular, rectilinear grid in three dimensional space, with each voxel representing a specific material. Due to their complexity, it is common to use voxel representations of biological subjects such as the human body. XFdtd supports the import of several types of voxel data:

- VariPose® Models or those defined with an MMF file
- ICRP Adult Reference Computational Phantoms[5]
- NICT Models[6]
- Virtual Population Models[7]

All voxel data is imported through the FILE > IMPORT > VOXEL OBJECT menu.

The voxel data import workflow is essentially the same for all voxel data types. After selecting a file representing the voxel data, a dialog is displayed that:

- Summarizes the count and resolution information of the data
- Provides the option to create default materials for tissues identified in the data but which are unknown
- Allows setting options for how to visualize the data after import
- Allows setting any other model-specific settings

When the dialog is accepted, the voxel data is imported and analyzed. Tissue assignments are either provided as part of the voxel specification or are implicit for the type of data, depending upon the source. For each tissue found in the imported data, XFdtd either uses an existing material definition if one exists in the project with the name of the tissue, or creates one for the tissue and adds it to the project. The electromagnetic properties and density for the latter are defined using the Tissue Properties Database published by IT'IS[8] (except where noted below), and a note is added to the Material definition indicating from which tissue those properties were obtained. If custom tissue definitions are desired when importing a voxel model, ensure that material definitions for the custom tissues are defined in the project before importing the voxel model (XFdtd's Libraries capability is useful for this workflow). When the process is complete, an import report is displayed showing all actions taken.

Material assignments for existing voxel objects can be displayed and edited later by choosing MATERIAL > VOXEL MATERIAL ASSIGNMENTS from the voxel object's right-click menu in the PROJECT TREE. The visualization options shown in the import dialog can also be changed later by choosing VOLUME VISUALIZATION OPTIONS from the voxel object's right-click menu in the PROJECT TREE.

Information specific to the import of each type of voxel data is outlined in the following subsections.

VariPose® (*.mmf)

VariPose® models or any other voxel data for which an MMF (`*.mmf`) metadata file is provided can be imported using the MMF(*.MMF VARIPOSE FORMAT) menu item. The MMF file provides voxel counts, resolution, material names, colors for the data, which is in a separate file referenced by the MMF file. Note that the density attribute of the MMF file is ignored for this import; the density provided in the tissue database noted above is used instead. When importing VariPose-formatted data, the MMF file is selected by the user, not the file containing the voxel data itself.

The voxel counts and resolutions displayed in the import dialog (see Figure 5.17) are read-only since they are specified in the MMF file. Figures 5.18 and 5.19 show the import report and the import result for the 5 mm VariPose man. Note that an object has been added to the PARTS branch that for the voxel object and materials created for that object added to the DEFINITIONS: MATERIALS branch.

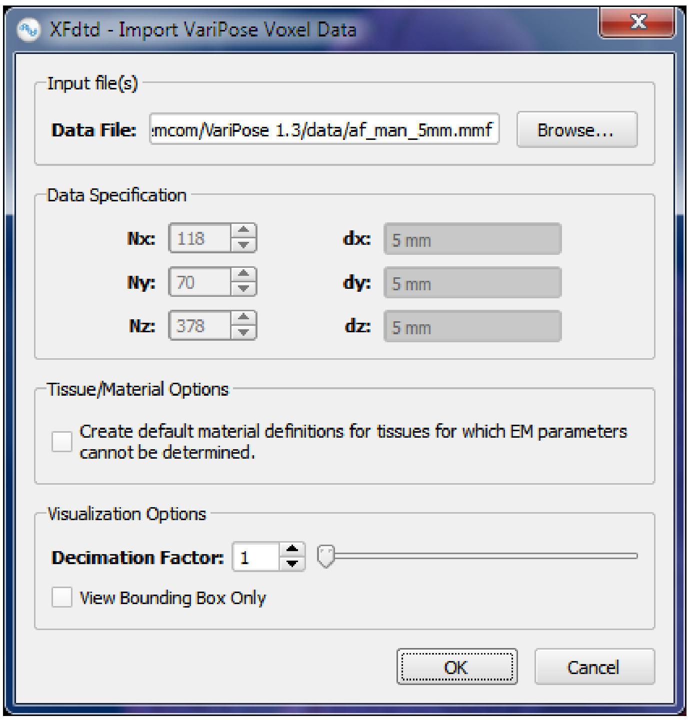


Figure 5.17: The import dialog for MMF voxel data

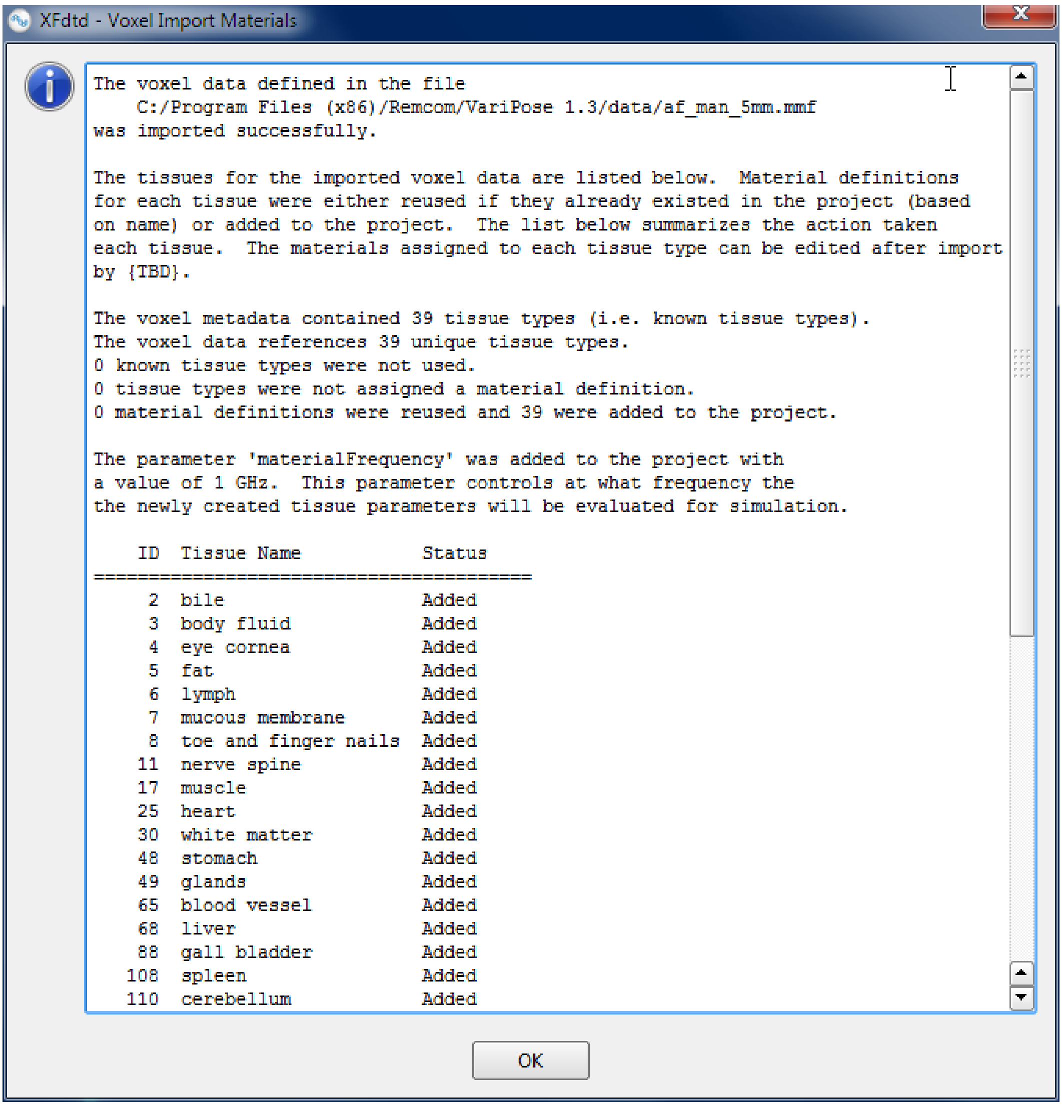


Figure 5.18: The import report for the 5 mm VariPose man

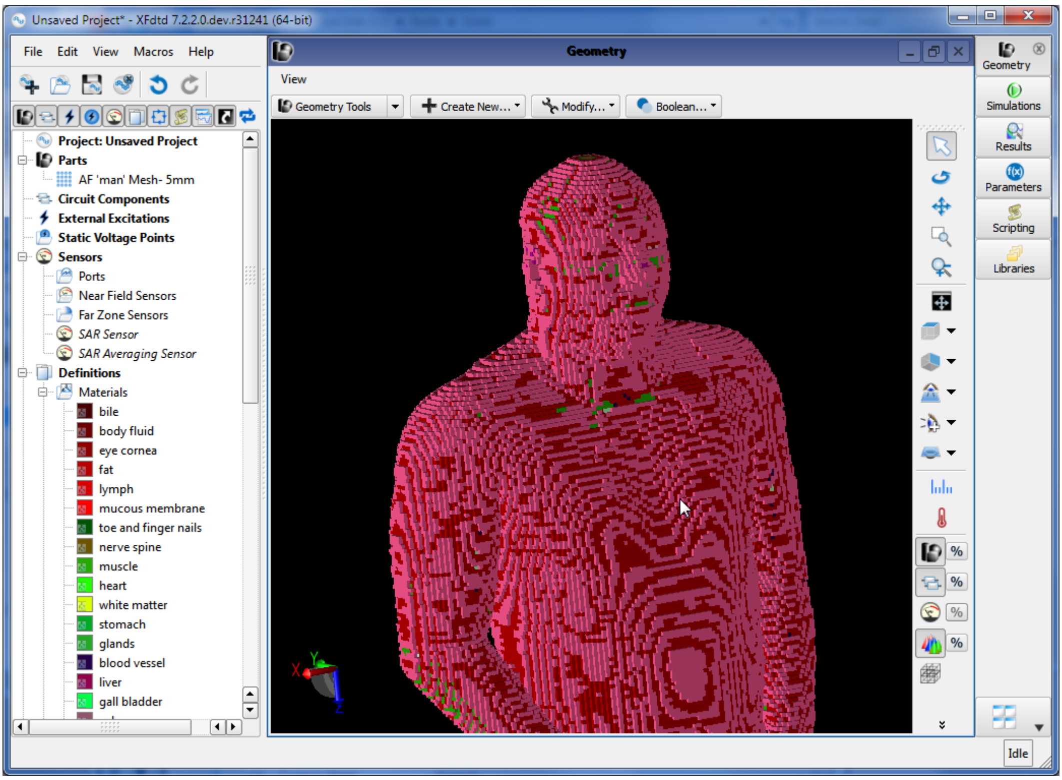


Figure 5.19: The result of importing the 5 mm VariPose man

ICRP Adult Reference Phantoms

The Adult Reference Computational Phantoms defined in International Commission on Radiological Protection (ICRP) Publication 110[5] can be imported directly into XFdtd using the ICRP ADULT REFERENCE menu item. The ICRP has defined two phantoms, one male and one female. XFdtd requires the data file (`AM.dat` or `AF.dat`, for male and female, respectively) and the associated "organs" file (`AM_organs.dat` or `AF_organs.dat`) that contains the organ (or tissue) definitions to import each of these phantoms. The data file is selected by the user while the "organs" file is initially inferred (but can also be selected independently in the import dialog).

The voxel counts and resolutions are implicit for these phantoms, so the import dialog provides a drop-down (see Figure 5.20) to select either the male or female reference. For either of these, the counts and resolutions are read-only. However, there may be occasions when a user may have a subset of one of the phantoms– for this case, there is also a "Custom" setting which allows editing of the counts and resolutions. Additionally, these phantoms contain an extra layer of skin at the top of the head and bottom of the feet that was added artificially. While XFdtd does not import these layers by default, the dialog contains an option to include them. The "organs" file provides tissue densities, and though the electromagnetic properties for the tissues is obtained from the tissue database noted above, the tissue densities from this file are used.

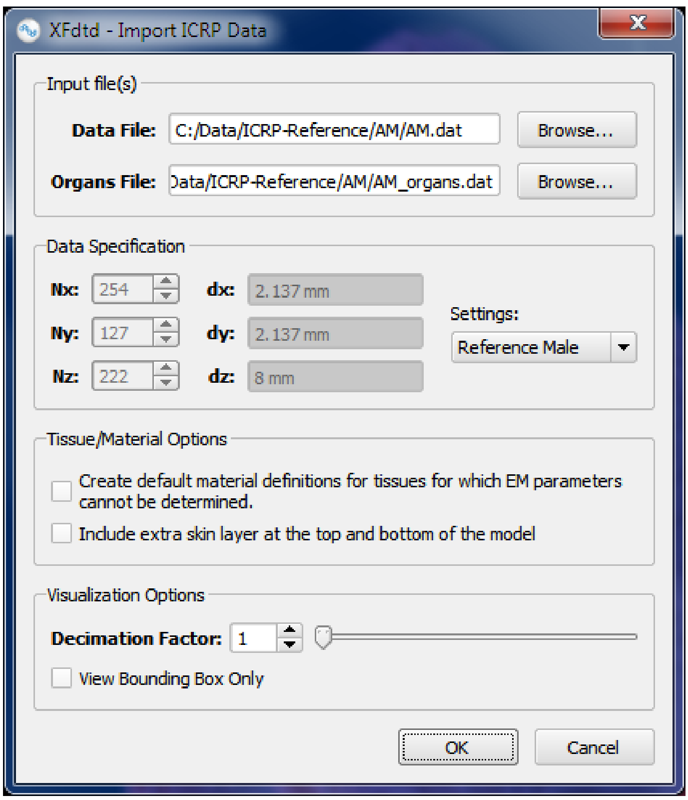


Figure 5.20: The import dialog for the ICRP Reference Phantoms

NICT Models

Voxel models prepared by the Electromagnetic Compatibility (EMC) Group at the National Institute of Information and Communications Technology (NICT)[6] can be imported directly into XFdtd using the NICT Models menu item. NICT has prepared primarily an adult male model (name Taro, `Taro.raw`), an adult female (named Hanako, `Hanako.raw` and a female 26 weeks pregnant (`NICT_Pregnant_26w.raw`). These models consist of raw data; the voxel counts and resolutions and tissue definitions are implicit, so the user need import only the raw data.

Similar to the ICRP Phantoms, the import dialog for NICT data (see Figure 5.21) provides a dropdown menu to choose which model is being imported in order to set the voxel counts and resolutions. If a new model or subset of an existing model is being imported, a Custom setting is provided so that the counts and resolutions can be entered manually.

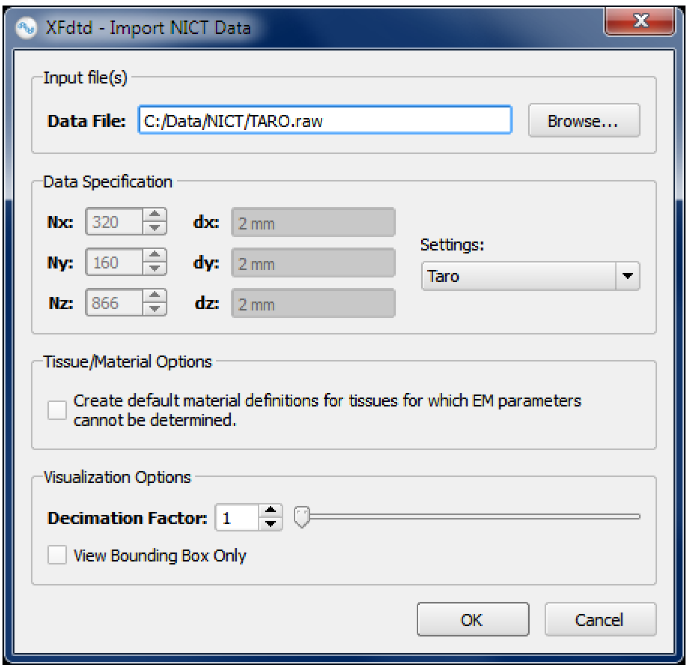


Figure 5.21: The import dialog for NICT Models

Virtual Population Models

Models from the Virtual Family and the Virtual Classroom (collectively known as the Virtual Population) created by the Foundation for Research on Information Technologies (IT'IS Foundation), as well as models re-voxelized using the Virtual Population Tool can be imported directly into XFdtd using the Virtual Population menu item. Each model is delivered as a zipfile that contains a `*.raw` and a `*.txt` file for the model, which are both required by XFdtd. The `*.raw` file is specified as the import file to XFdtd (XFdtd does not import the accompanying `*.sat` file), so the zipfiles must be unpacked before importing them into XFdtd. The `*.raw` file contains the actual voxel data, while the `*.txt` file contains tissue definitions and voxel counts and resolutions; therefore the counts and resolutions are read-only in the import dialog (see Figure 5.22).

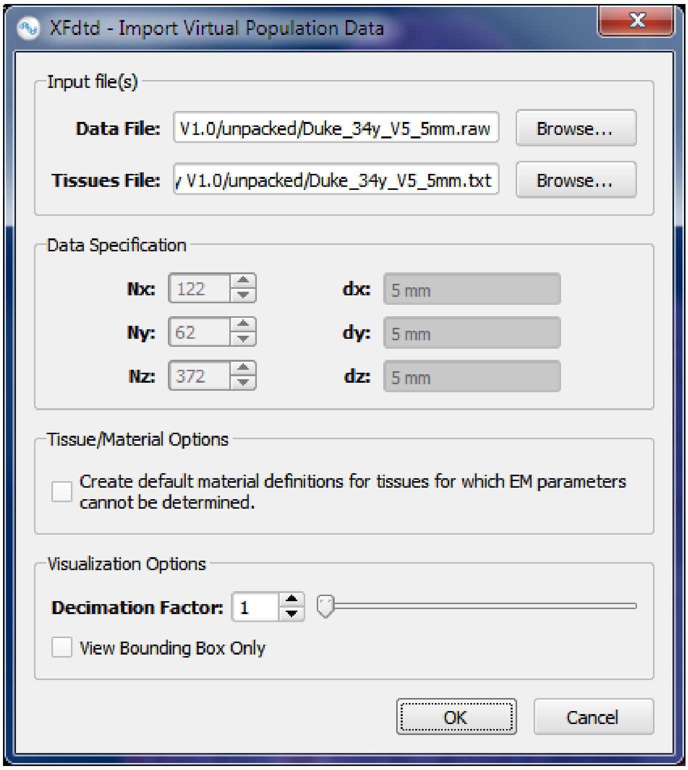


Figure 5.22: The import dialog for Virtual Population Models

Viewing/Editing Voxel Object Material Assignments

Material assignments for existing voxel objects can be displayed and edited by choosing MATERIAL > VOXEL MATERIAL ASSIGNMENTS from the voxel object's right-click menu in the PROJECT TREE. This will open a dialog that displays for each voxel data ID, what tissue name was assigned to that ID in the source data and what material in the XFdtd Project that is currently assigned to that ID. These assignments can changed by selecting one or more rows, right-clicking and then selecting one of the menu options (as shown in Figure 5.23):

Assign Existing Material...– Assign a material that already exists in the Project to the selected IDs.

Automatically Generate and Assign New Tissue Material– This will perform a similar algorithm that was used during the import process to automatically determine appropriate tissue parameters (based on tissue name and source data type), then add a new material to the Project with those parameters. This would be used primarily to assign materials to voxel objects in existing Projects that do not have appropriate materials already assigned to them or to replace accidentally deleted materials.

Remove Material Assignment– Remove the assignment of any material to the selected IDs.

A report containing information on what materials were created and from what tissues the electromagnetic properites were taken will be shown when the dialog is dismissed.

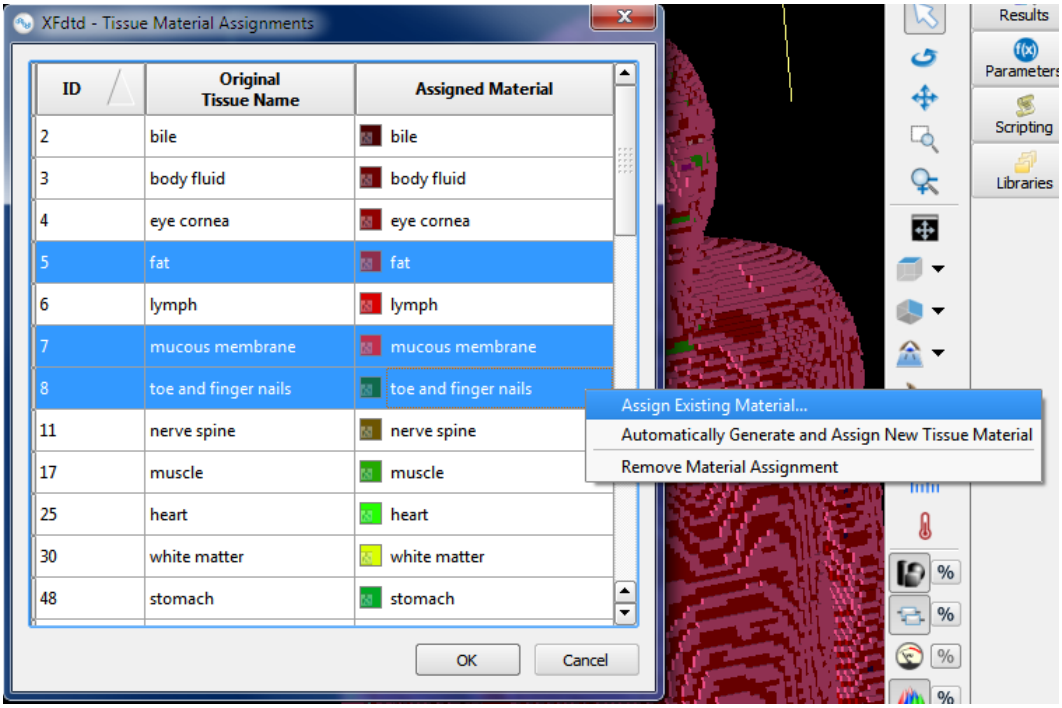


Figure 5.23: Assigning tissue materials to voxel objects

Meshing a Voxel object

To set the meshing parameters for the voxel object, right-click on the object in the PROJECT TREE and select GRIDDING / MESHING > MESHING PROPERTIES to open the MESHING PARAMETERS EDITOR.

- See Section 8.3.6 for more on the settings of the MESHING PARAMETERS EDITOR.

5.4.3 Mesh Objects

A mesh object is essentially a collection of edges with applied materials. It is a subsection of the XFdtd grid from a previous version of XFdtd. Mesh objects, like voxel objects, are linked to an external mesh object data file, and are loaded through the FILE > IMPORT > XFDTD VERSION 6 FILE option. Figure 5.24 displays an imported *.mesh file.

- The import of Version 6 mesh objects with adaptive grid regions is not supported.
- See Appendix Section C.2 for a discussion on importing other Version 6 files.

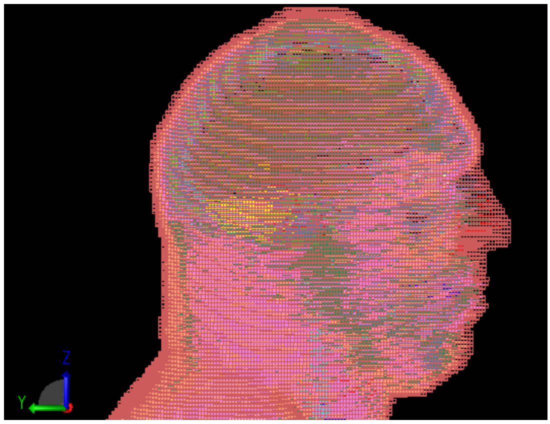


Figure 5.24: The head of an imported human body mesh object

Figure 5.25 shows the PROJECT TREE after the `*.mesh` file seen above was imported. Note that an object has been added to the PARTS branch that contains the mesh object and a list of all the materials contained in the object have been added to the DEFINITIONS: MATERIALS branch.

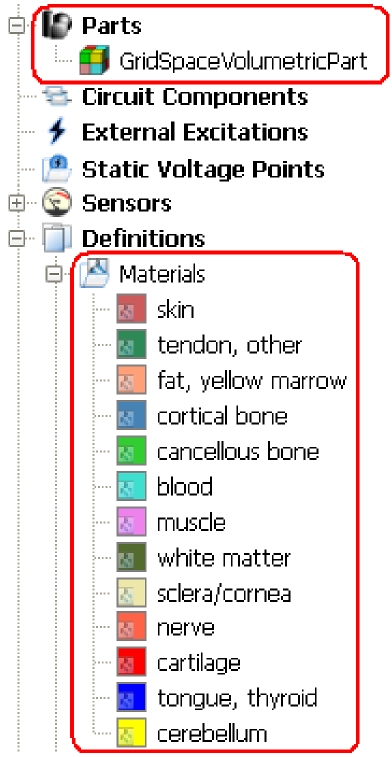


Figure 5.25: The Project Tree with imported mesh object

Meshing a Mesh object

Like a voxel object, to set the meshing parameters for a mesh object, right-click on the object in the PROJECT TREE and select GRIDDING / MESHING > MESHING PROPERTIES to open the MESHING PARAMETERS EDITOR.

- See Section 8.3.6 for more on configuring the settings of the MESHING PARAMETERS EDITOR.

5.4.4 PCB

The FDTD method can be used to simulate how energy propagates through and around Printed Circuit Boards (PCBs). XFdtd's user interface was not designed to create complex PCB designs thus geometry must be imported from another application. The ODB++ format was chosen because it is a standardized format which facilitates data exchange between many applications in the PCB industry.

XFdtd reads the ODB++ database and constructs a three dimensional representation of the data. The import window provides several options to customize the construction of the solid model. To initiate the ODB++ import, first select FILE > IMPORT > PCB and then choose the desired directory. XFdtd will read the data and open the Import PCB window seen in Figure 5.26.

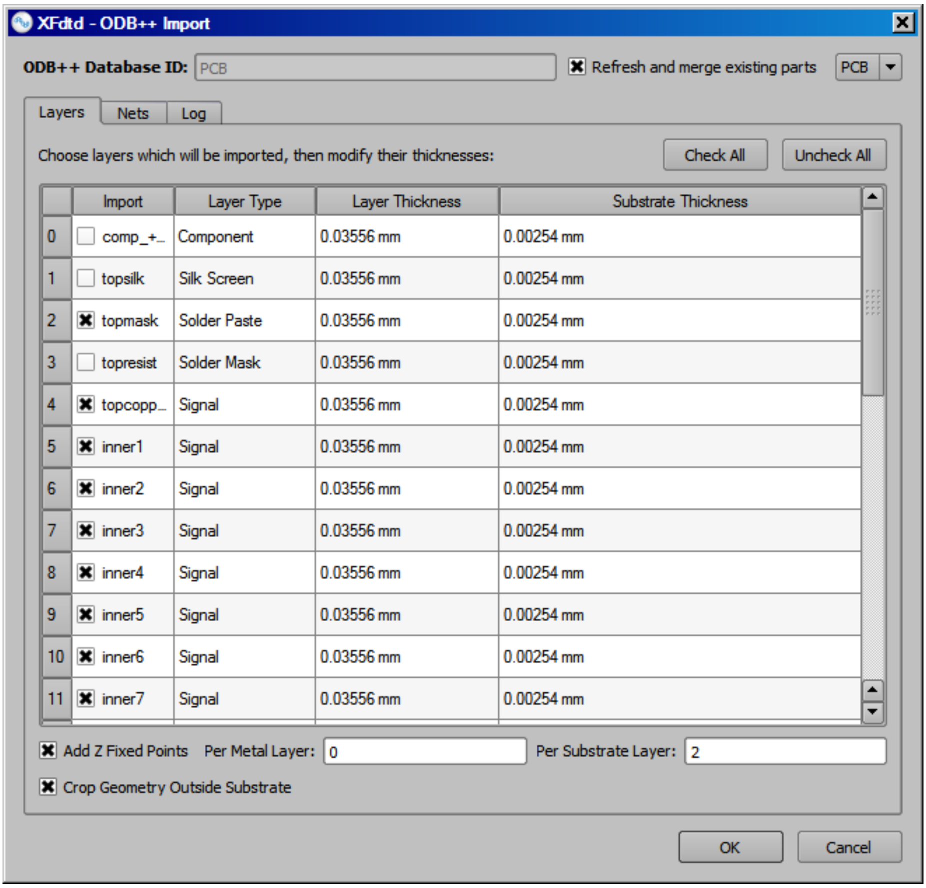


Figure 5.26: ODB++ Layers Tab

At the top of the import window resides a field to specify the ODB++ Database ID. This ID allows solid geometry to be identified as an ODB++ database which can refreshed and merged similar to CAD geometry. When importing a ODB++ database, parts in the project are examined to determine if an existing database is available for merge. The REFRESH AND MERGE EXISTING PARTS and associated database are only available if an existing database has been detected.

The Layers tab allows the user to specify which layers will be imported and the thickness of each layer. Layers of the types signal, power ground, drill and mixed will be checked by default, but each layer can be changed by clicking the checkbox next to the layer name. If a database is being merged the settings of the previous import will be used, otherwise the layer thickness and dielectric thicknesses are populated given they were provided in the database or zero if not specified in the file. Each thickness can be modified by clicking on the value in the table and entering a different value. The ADD Z FIXED POINTS option specifies whether or not to add manual fixed points to the grid per layer in order to resolve the geometry adequately in the mesh. The CROP GEOMETRY OUTSIDE SUBSTRATE option excludes all ODB++ features completely outside the profile of the substrate from being imported.

The Nets tab, as seen in Figure 5.27, will be available if the ODB++ database contained EDA data. If there is no EDA data or no nets are chosen, the geometry for each layer is separated into pads, traces, surfaces and the profile. If nets are chosen, the geometry for each net is separated in its own part, and then the remaining geometry is separated into layers. The BOOLEAN NETS option attempts to reduce the complexity of the solid geometry by booleaning subnets together.

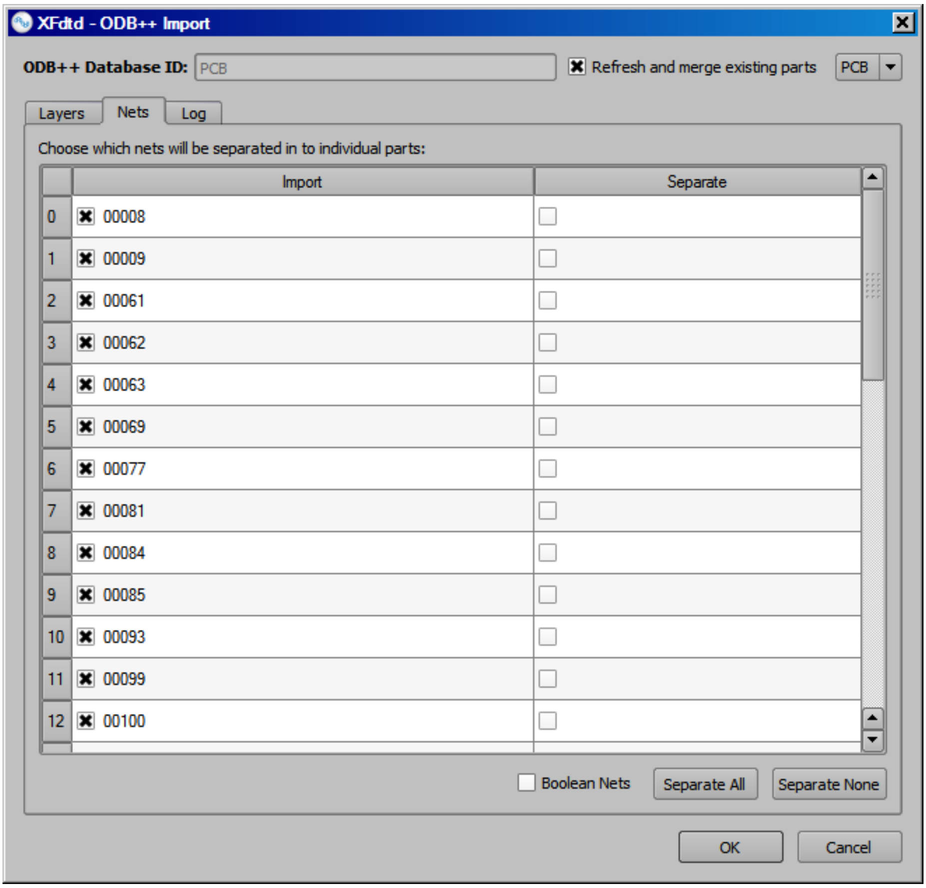


Figure 5.27: ODB++ Nets Tab

The Log tab in Figure 5.28, provides the feedback generated when the data was read from the ODB++ database. The contents of this log will indicate any errors encountered. The messages provided in the log do not include the status of converting the data to the solid models. After the user chooses OK, the data is converted into parts. In the event of an error during this conversion, a window will appear after conversion has finished indicating the problems encountered.

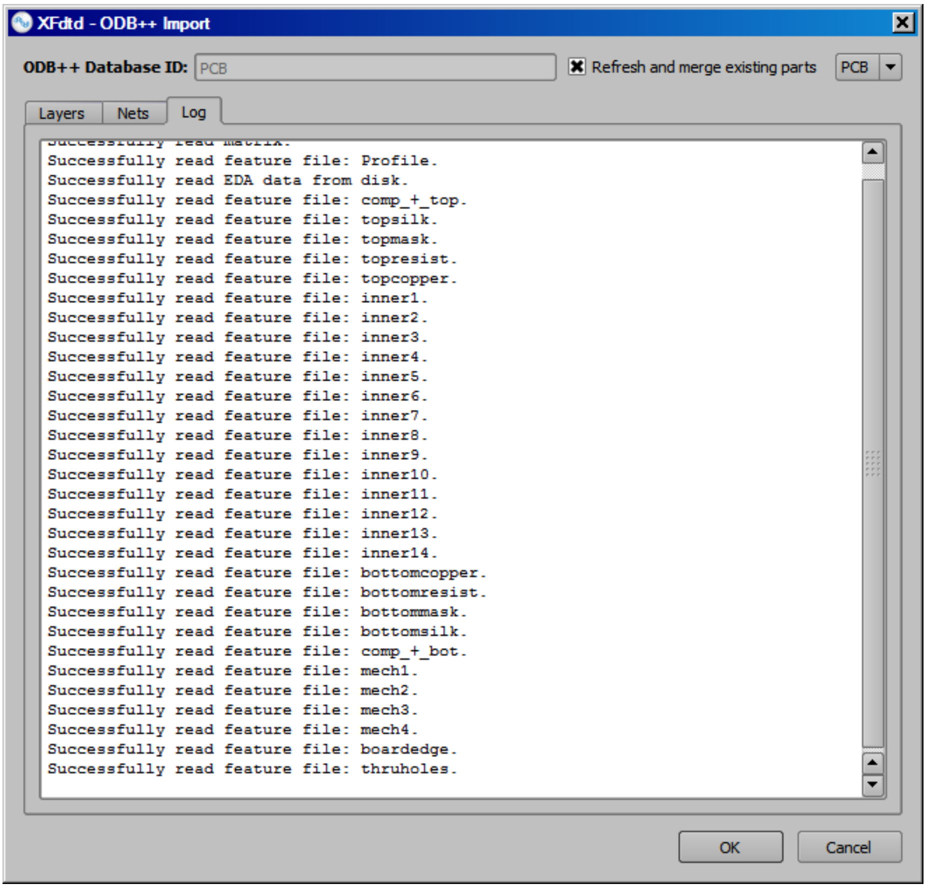


Figure 5.28: ODB++ Log Tab

5.5 Orienting Objects in the Simulation Space

The SPECIFY ORIENTATION tab is available for positioning any physical object in the simulation space. This tab provides tools for translating, twisting and rotating an object using three different coordinate systems. There are two primary modes that are available within this tab: BASIC MODE, the default mode which is sufficient under most circumstances, and ADVANCED MODE, which has more powerful functionality. The PICK and ALIGN drop-down menus available in each mode provide additional options for orienting objects.

At the end of this section, two examples are provided of how 2-D sketches are effectively placed in the simulation space so that 3-D modeling operations are carried out correctly.

5.5.1 Overview of Global, Reference and Local Coordinate Systems

Within the XFdtd interface, there are three primary coordinate systems: Global, Reference and Local. They are distinguished in the GUI in the following reference forms:

- Global coordinates: X, Y and Z

- Reference coordinates: U, V and W
- Local coordinates: U', V' and W'

The Global Coordinate System does not change throughout the course of a project, so its origin and three primary axes always remain the same. The 3-D orientation object marked with X, Y and Z in the lower left corner of the GEOMETRY workspace window represents the global coordinate system. This set of orientation vectors is referred to as the GLOBAL TRIAD.

The Reference Coordinate System refers to the orientation and location of objects within their native assembly. If an assembly has not been translated, rotated or shifted, the Reference Coordinate System will be the same as the Global Coordinate System; otherwise, it will be different. Additionally, if an assembly exists within another assembly, its Reference Coordinate System will only consider the location and orientation of its own assembly.

The Local Coordinate System is initiated when an editing session begins within the SPECIFY ORIENTATION tab. It is initially the same as the Reference Coordinate System, but it changes as translations, rotations and other adjustments are made to the orientation of the working coordinate system. Figure 5.29 shows the set of orientation vectors displayed in this tab, known as the ORIENTATION TRIAD. This coordinate system is reset each time the editing session is reopened, unlike the Reference Coordinate System, which is preserved across editing sessions.

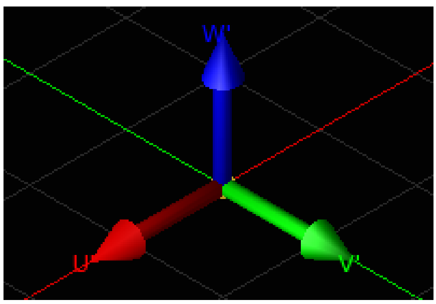


Figure 5.29: The Orientation Triad governs the location and orientation of the Local Coordinate System.

The ORIENTATION TRIAD is used to correctly orient objects within the simulation space. The center of the Orientation Triad is referred to as the ORIGIN. The Origin can be defined manually by typing in its coordinates or by clicking on the intended origin in the simulation space with the PICK: ORIGIN tool. It may also be manually shifted by clicking and dragging the directional axis vector of interest.

- Read ahead to Section 5.5.3 for more about the PICK: ORIGIN tool.

5.5.2 Basic Mode

Figure 5.30 shows the SPECIFY ORIENTATION tab in Basic Mode. The CONTEXT controls whether the Orientation Triad's position and orientation is cited in the Global or Reference Coordinate System. Notice that in Figure 5.30, the coordinates are all defined in terms of X, Y and Z, denoting the Global Coordinate System.

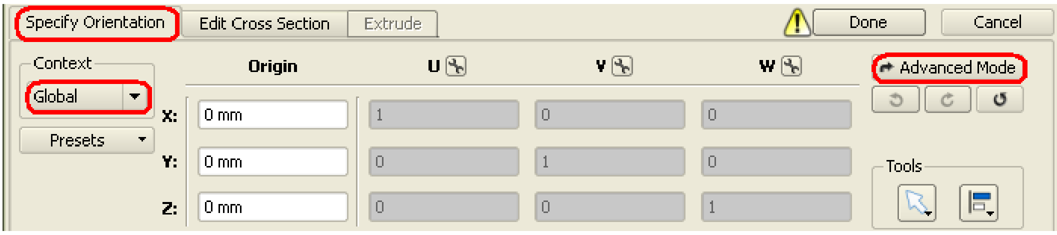


Figure 5.30: The Specify Orientation tab in Basic Mode

The origin and orientation of the Orientation Triad can be adjusted in several ways within Basic Mode. All of these methods are also available in Advanced Mode.

The ORIGIN coordinates specify the location of the origin of the Orientation Triad. The origin can be updated manually or can be adjusted by clicking and dragging on one of the colored primary-axis vectors of the Triad. The location display will update automatically.

The orientation of the axes is defined by clicking on the icon found next to any of the axis definitions. This will change the column from gray (read-only) to white so the values in the column can be edited. The orientation may also be adjusted by right-clicking on the axis and choosing the appropriate tool in the context menu, displayed in Figure 5.31.

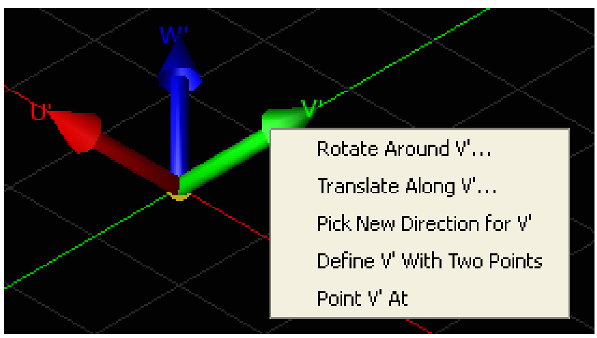


Figure 5.31: Customizing the orientation of the simulation space

- ROTATE AROUND [AXIS NAME] - rotates the coordinate system about the chosen axis.
- TRANSLATE ALONG [AXIS NAME] - prompts the user to type in a translation distance along the selected axis (analogous to clicking and dragging on the axis vector.)
- PICK NEW DIRECTION FOR [AXIS NAME] - redefines the direction of the selected axis in a new direction specified by the user. This tool is also useful to align the selected axis with the surface of other objects in the simulation space.
  - ✓ Clicking a point in the simulation space will assign the direction vector directly into the space. Thus, adjusting the view with the VIEW buttons may facilitate this assignment.
- DEFINE [AXIS NAME] WITH TWO POINTS - aligns the axis to the direction of the vector between two user-selected points.

- POINT [AXIS NAME] AT - redirects the selected axis by directing the vector from its origin to a user-selected point.

Clicking on ADVANCED MODE in the Basic Mode window will bring up the Advanced Mode window.

See Section 5.5.4 for a description of the ADVANCED MODE interface.

5.5.3 Alignment Tools

Direction Picking Tools

The DIRECTION PICKING TOOLS menu provides six tools for defining the location and direction of the Orientation Triad. In the descriptions of tools used to align one object to another, "reference object" will refer to the pre-existing, stationary object used as a reference for the alignment, and "object to align" will refer to the object that will be aligned to the reference object using the respective tool.

Simple Plane. Select a plane on a reference object to orient the plane of the object to align. The normal vector will be directed out of (orthogonal to) the selected plane, as shown in Figure 5.32. XFdtd will adjust the object to align so that its face is aligned with the selected plane of the reference object. Press the SPACE key until the object is correctly oriented.

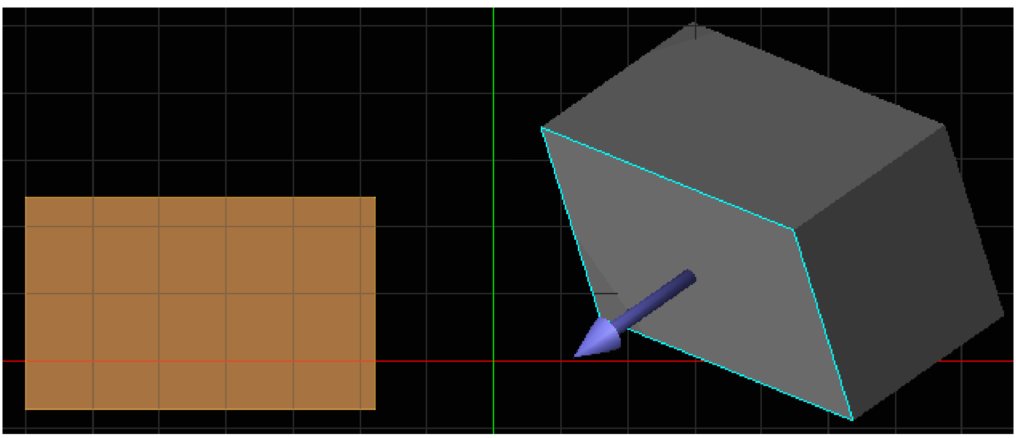


Figure 5.32: Aligning a new object with an existing reference plane

Origin. Select a point anywhere in the simulation space to position the origin of the Orientation Triad. (The normal vectors will not change orientation.)

Normal. This tool allows the user to adjust the direction of the normal vector of the Orientation Triad while maintaining the same origin point. The normal direction can be defined by clicking on a point in the simulation space, in which case the W'-normal vector will be positioned directly into the simulation space. The normal vector can also be aligned with the face of a reference object by holding the mouse over one of its faces, and the normal vector of the object to align will be aligned with this face of interest.

✓ It is convenient to use the VIEW buttons to position the simulation space so that this placement achieves the desired orientation.

✓ Press SPACE to rotate the direction of the orientation vector before placing it.

Direction for Twist. Twisting about an axis is performed by defining a normal vector. The object will be twisted about the origin as the twist slider is moved left and right. Figure 5.33 shows the placement of the normal (purple arrow) and the radius of twist about the origin.

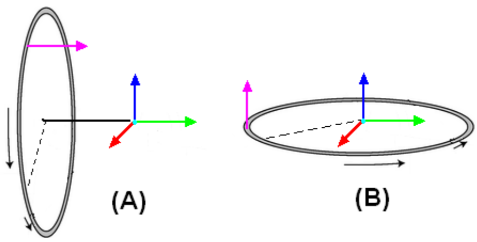


Figure 5.33: Twisting an object about the origin

Axis for Rotation. Unlike the DIRECTION FOR TWIST, the AXIS FOR ROTATION moves the object and Orientation Triad about a user-defined axis. The object will not be twisted as it is rotated (i.e. a face originally faced in the Z-direction will always face in that direction independent of its displacement.) Figure 5.34 shows the rotation of the Orientation Triad and object about the user-defined axis (shown as a purple arrow.)

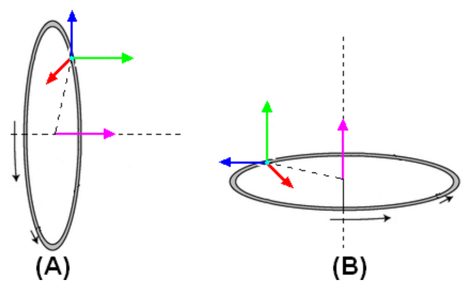


Figure 5.34: Rotating an object about a user-defined axis

Axis for Rotation and Twist. This tool is analogous to the AXIS FOR ROTATION tool except when the object is rotated around the picked axis, it will also be twisted.

✓ The DIRECTION PICKING TOOLS hotkeys are a convenient option for placing your tool in the exact position desired. The hotkeys menu for each respective tool will appear after selecting the tool and holding the mouse over the geometry. Table 5.1 describes the available options.

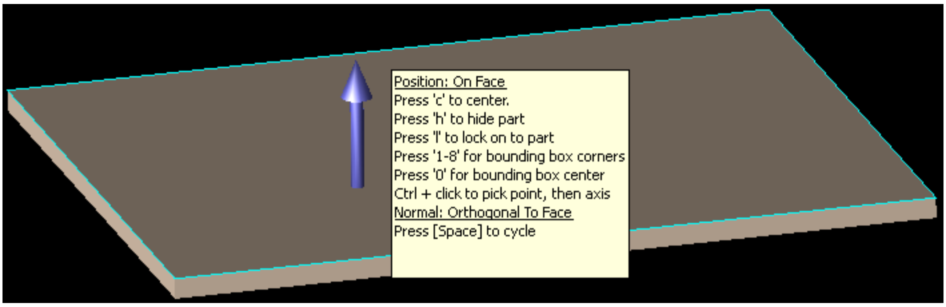


Figure 5.35: The hotkeys menu appears when a direction picking tool hovers over an object.

Align menu

The ALIGN menu provides several tools for aligning features of a new object with those on a reference object.

Align Directions. This tool aligns the face, edge or vertex of the object to align with the corresponding face, edge, or vertex of the reference object. Pick the face, edge or vertex on the object to align, then pick the face, edge or vertex of the reference object. The selected faces, edges or verticies will be used to orient the coordinate system such that the objects are pointing in the same direction. Figure 5.36 shows an object to align (left) being aligned with a face on the reference object (right.) The bottom image shows the two objects after they have been aligned.

✓ Press SPACE to reverse the direction of the orientation vector before placing it.

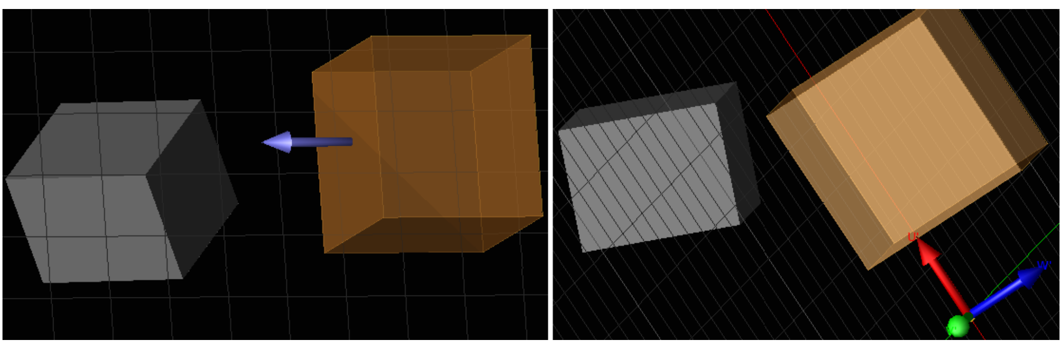


Figure 5.36: Using Match Directions to align two objects

Match Points. This tool matches a user-selected point on the object to align with a user-selected point on the reference object. Select a point on the object to align and select a second point on the reference object. Figure 5.37 shows the original objects in Figure 5.36 after two of their corners were matched.

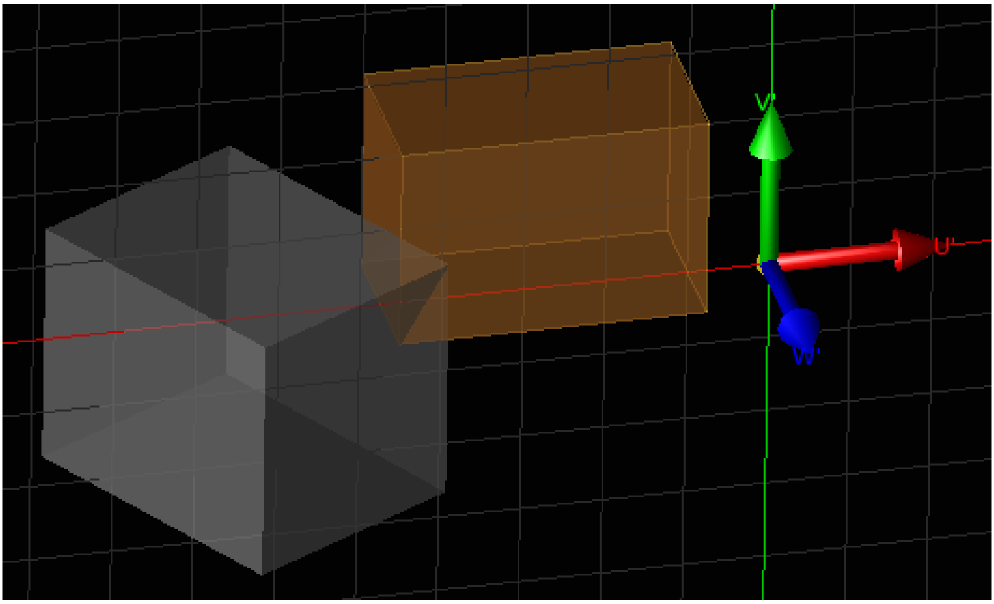


Figure 5.37: Aligning points on objects

Match Points & Directions. This tool functions much like the MATCH POINTS tool above except that it aligns the normals of two objects rather than discrete points. Select the object to align then select the face of the reference object. The two positions will be used to translate the first part such that the two selected points are coincident. Since aligning the faces of two objects do not give regard to the placement of the moved object, it is often useful to slide the moved object along an axis on the Orientation Triad until the desired placement is achieved.

✓ Press SPACE to reverse the direction of the orientation vector before placing it.

5.5.4 Advanced Mode

The options available in Advanced Mode allow the user to manipulate virtually all aspects of the orientation of the coordinate system. The three main areas for editing in this mode are:

- ANCHOR: The anchor point is defined by a location within the reference coordinate system. This location, along with the offsets defined by TRANSLATIONS, determines where the origin (0,0,0) of the local coordinate system is located.
- AXIS: Defines the primary direction of the local coordinate system. Users can pick which direction the axis defines, U', V', or W'. Whatever value is entered or picked will exactly define the U' V' or W' direction.
- TWIST: Defines a secondary direction that orients the plane defined by the remaining axes. If the Axis is used to define U', Twist orients either V' or W', eliminating the remaining degree of freedom.

There are also fields for defining TRANSLATIONS and ROTATIONS. These fields are described below.

Finally, the PICKING TOOL and DETATCH TOOL buttons enable the user to use the mouse to either select or detatch an anchor point, axis direction or twist direction, respectively. The PICKING TOOL is always available. The DETATCH TOOL is only available when an anchor point, axis direction or twist direction is defined using reference geometry.

When using a Picking Tool, mousing over the existing geometry displays a selection arrow (for directions) or sphere (for positions.) Depending on the position or direction and the location of the indicator relative to the geometric object, extra Picking Tool functions are available. The following table displays these extra functions.

Table 5.1: Direction and Location Picking Functions

Function	**Description**
Hold 'c' to center	Centers the anchor point on the face.
Press 'h' to hide part	Temporarily hides any part (or parts) that the user points at. The part reappears when the current picking operation completes. This is useful if the user wants to hide a part that obscures another part.
Press 'l' to lock	On mouseover, shows the selected part by temporarily removing any objects that may block the user's view. The parts will reappear after the cursor moves away.
Press [Space] to cycle	Cycles the indicator through the normal, parallel, and isometric directions that can be determined using the indicated geometry and its connected entities. Indicator movement depends on how many edges/faces meet at the particular entity the directional indicator is over.
Press '1-8' for bounding box corners	Moves the anchor to each corner of the bounding box surrounding the selected part. The bounding box is the smallest axis aligned box that can enclose the object the user is pointing at.
Press '0' for bounding box center	Moves the anchor to the center of the bounding box surrounding the selected part.
CTRL Key	When picking a location and a direction, users can hold down the CTRL key to pick the location, then pick the direction. This is most useful when the normal of the face you're trying to select isn't at the location you want.

The Picking and Detach tools for the Anchor, Axis and Twist areas operate independently based on the radio button selected in the Specify Orientation Tab (Anchor, Axis, Twist). For example, if the Axis radio button is selected, the PICKING TOOL will enable the user to place an axis on an object and the DETACH TOOL (if active) will enable the user to detach the existing axis from an object. The user cannot select or detach any existing anchor points or twist directions when the Axis radio button is selected, only axes.

Translations can also be applied manually in any direction within the TRANSLATION definition. A translation will not affect the location of the anchor point, only the location of the Orientation Triad and attached object. It is important to be aware of which of the three coordinate systems is selected in the CONTEXT definition when applying a manual translation, since it may affect the proper placement of the geometry.

Similarly, rotations may also be manually defined within the ROTATION definition. The rotation, across the angle specified within ANGLE, will be applied to the Orientation Triad (and attached object) at the anchor point, in the direction specified by the U', V' and W' definitions. For example,the Anchor will be rotated π radians about the W' axis by defining

$$\begin{aligned} U' &= 0 \\ V' &= 0 \\ W' &= 1 \\ Angle &= \text{"PI rad"} \end{aligned}$$

If the definitions were changed to

$$\begin{aligned} U' &= 1 \\ V' &= 0 \\ W' &= 1 \\ Angle &= \text{"PI rad"} \end{aligned}$$

then U' and W' will be automatically adjusted to U' = "0.707107" and W' = "0.707107", since the resulting rotation will rotate the orientation 0.707107 radians in the U' and W' directions.

The net movement of the Orientation Triad in the coordinate system will be

$$U + U' + U'_{rotation}$$

$$V + V' + V'_{rotation}$$

$$W + W' + W'_{rotation}$$

The ANCHOR, AXIS and TWIST buttons are detailed below.

Orienting the Anchor

Figure 5.38 shows the ANCHOR editing dialog within Advanced Mode.

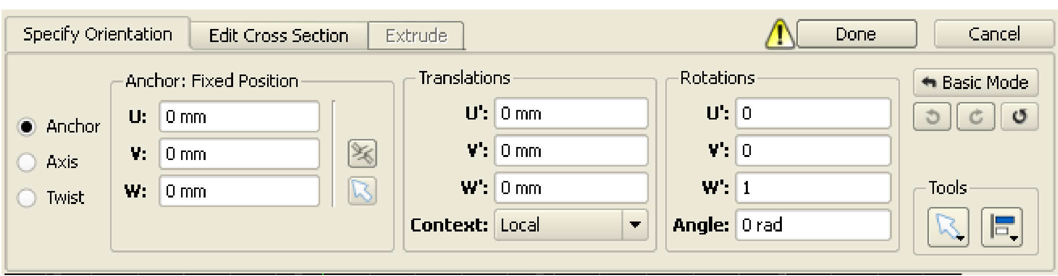


Figure 5.38: Modifying the location and orientation of the anchor

In Advanced mode, the anchor point is represented by a blue dot in the simulation space. It remains in place regardless of rotations and translations that are applied to the Orientation Triad. When this dialog is first opened, U, V and W within the ANCHOR:FIXED POSITION section represent the location of the anchor point in the Reference Coordinate System. Adjusting the location of the anchor point will adjust the location of the Orientation Triad as well (although moving the Orientation Triad will NOT affect the position of the anchor point.)

Advanced Anchor Point Capabilities

Right-clicking an anchor point displays its context menu. The menu contains two items: DETACH ANCHOR POINT and ADVANCED.

- DETACH ANCHOR POINT: Selecting this item detaches the anchor point from the specified object.
- ADVANCED: Selecting this item displays Advanced dialog window. This window enables the user to parameterize the location of the anchor point. It contains two sections: **Parameters** and **Offsets**.

  - **Parameters**: The Parameters section contains sliders that enable the user to make minute adjustments to the parametric position of the anchor. If the anchor is attached to a surface of a part, sliders U and V appear. If the anchor is attached to a part's edge, slider U appears. If the anchor is attached to a vertex or the center of an edge, no sliders appear. U defines the parametric position of the anchor along an edge. U,V defines the parametric position along a face or surface. Parameter values always vary from 0 to 1 with 0.5 corresponding to the middle of the edge, surface or face.

  - **Offsets**: The Offsets section enables users to offset the anchor point for X, Y and Z of the Global Coordinate System. For more information on the Global Coordinate System, see Section 5.5.1.

Parameters defined in the PARAMETERS workspace window can be used as values in the Parameters section and the Offset section.

In the Parameters section, U and V are not the same as reference coordinates U and V.

Orienting the Axis

Figure 5.39 shows the AXIS editing dialog within Advanced Mode.

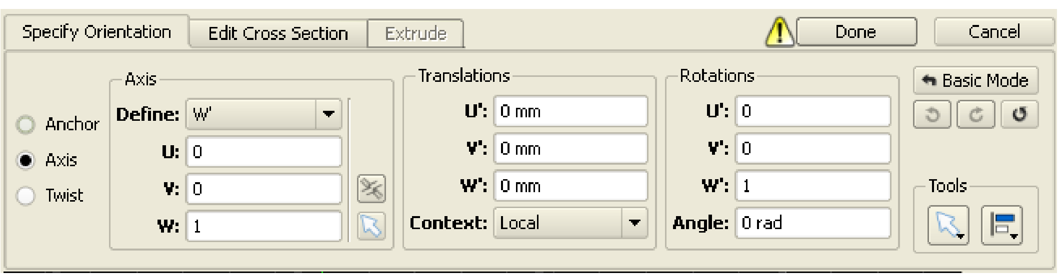


Figure 5.39: Modifying the orientation of the axis

The AXIS dialog controls the orientation of the axes of the anchor. The default orientation of the Orientation Triad vectors are:

U': U = 0

V': V = 0

W': W = 1

These can be redefined to assume any orientation. Changing the orientation of one axis will automatically adjust the orientation of the other two axes since they must always exist at 90 degrees from one another.

The DEFINE drop-down list allows you to specify which axis is associated with the direction defined in the U, V and W orientation boxes.

Twisting the Axis

The twist axis defines a second direction for orthogonalizing the triad. Figure 5.40 shows TWIST editing dialog within Advanced Mode.

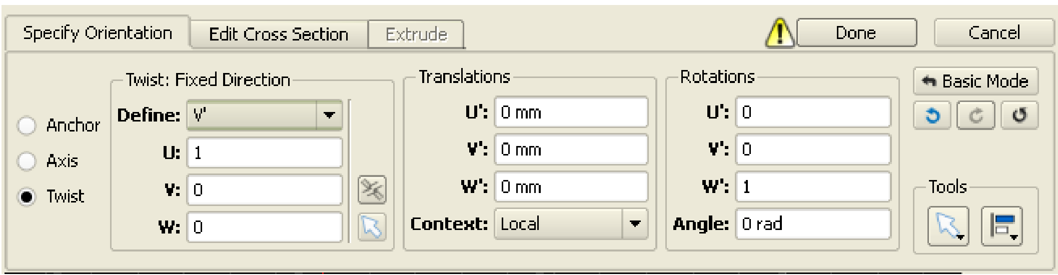


Figure 5.40: Twist dialog

5.5.5 Orienting 2-D Sketches on the Sketching Plane

The correct placement of the Orientation Triad in relation to its corresponding parts is critical to some 3-D modeling operations, especially during revolution and sweeping operations. It is important to understand the functionality of the Orientation Triad in relation to the operation that is performed. The following two examples show how the Orientation Triad should be placed about a 2-D cross section.

Orienting sketches for revolutions

Figures 5.41 through 5.44 demonstrate the placement of the Orientation Triad during a revolution operation. Figure 5.41 shows a simple 2-D cross section sketched in the EDIT CROSS SECTION tab. Notice that the sketch does not intersect any of the primary axes.

It is important that the 2-D sketch does not touch any axis of revolution, or else the revolution cannot be performed.

Figure 5.42 shows the revolution of the circle about the V'-axis and Figure 5.43 shows the revolution of the circle in the U'- and V'-directions. Figure 5.44 shows how the revolution operation is used to create hollow or solid sphere from an open or closed semi-circle cross section, respectively.

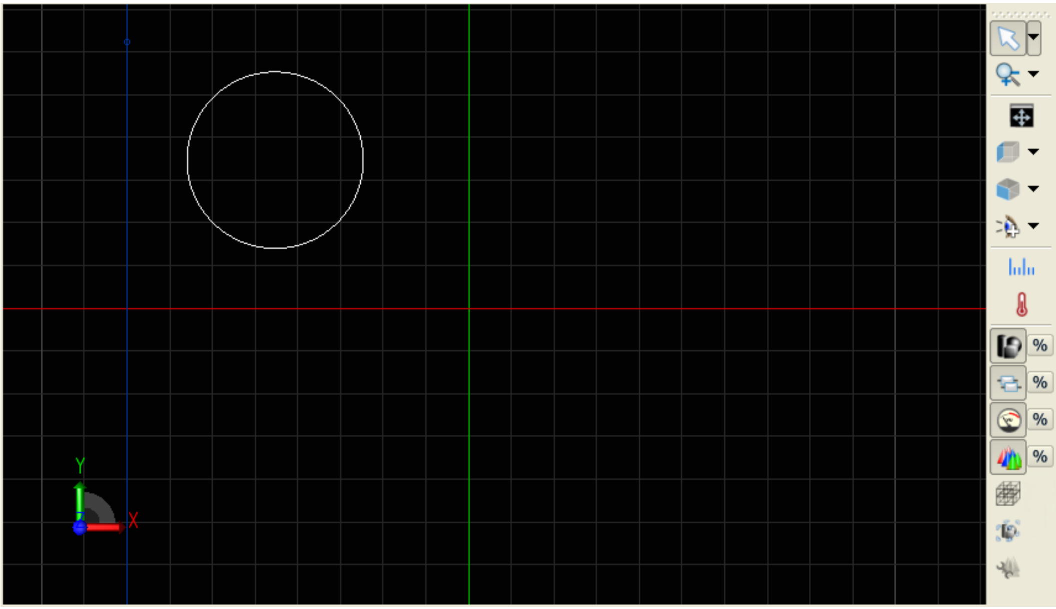


Figure 5.41: Orienting a 2-D cross section about an axis of revolution

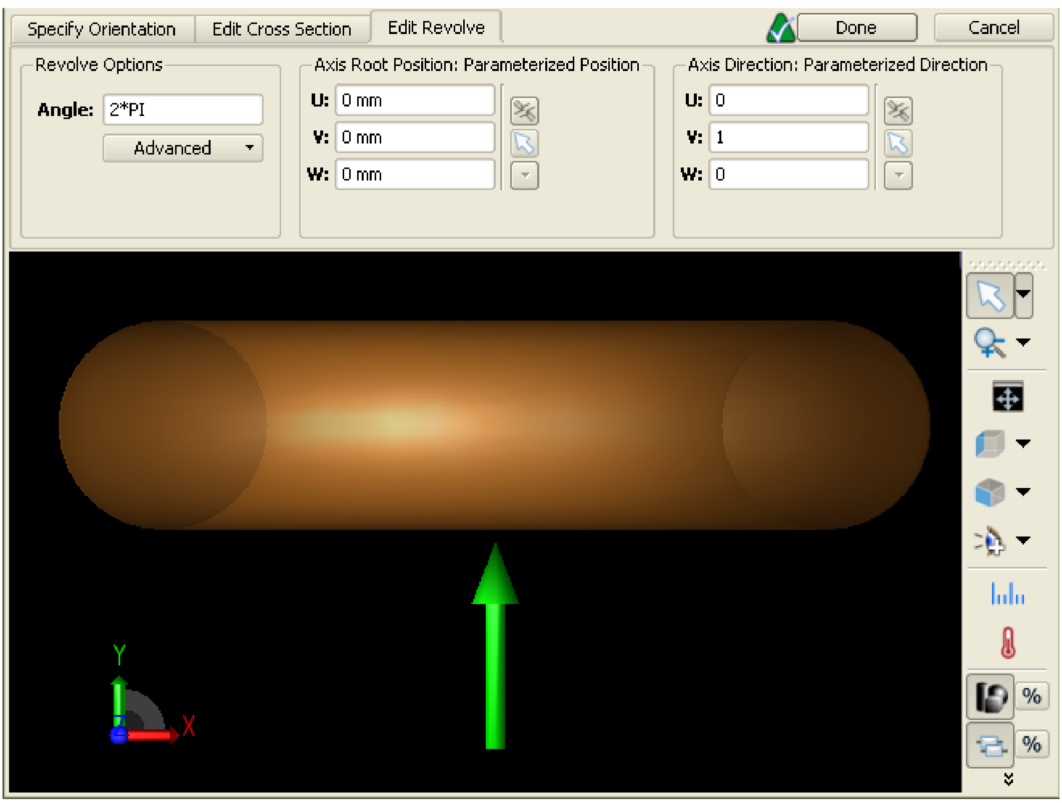


Figure 5.42: Cross section revolved about the V'-axis

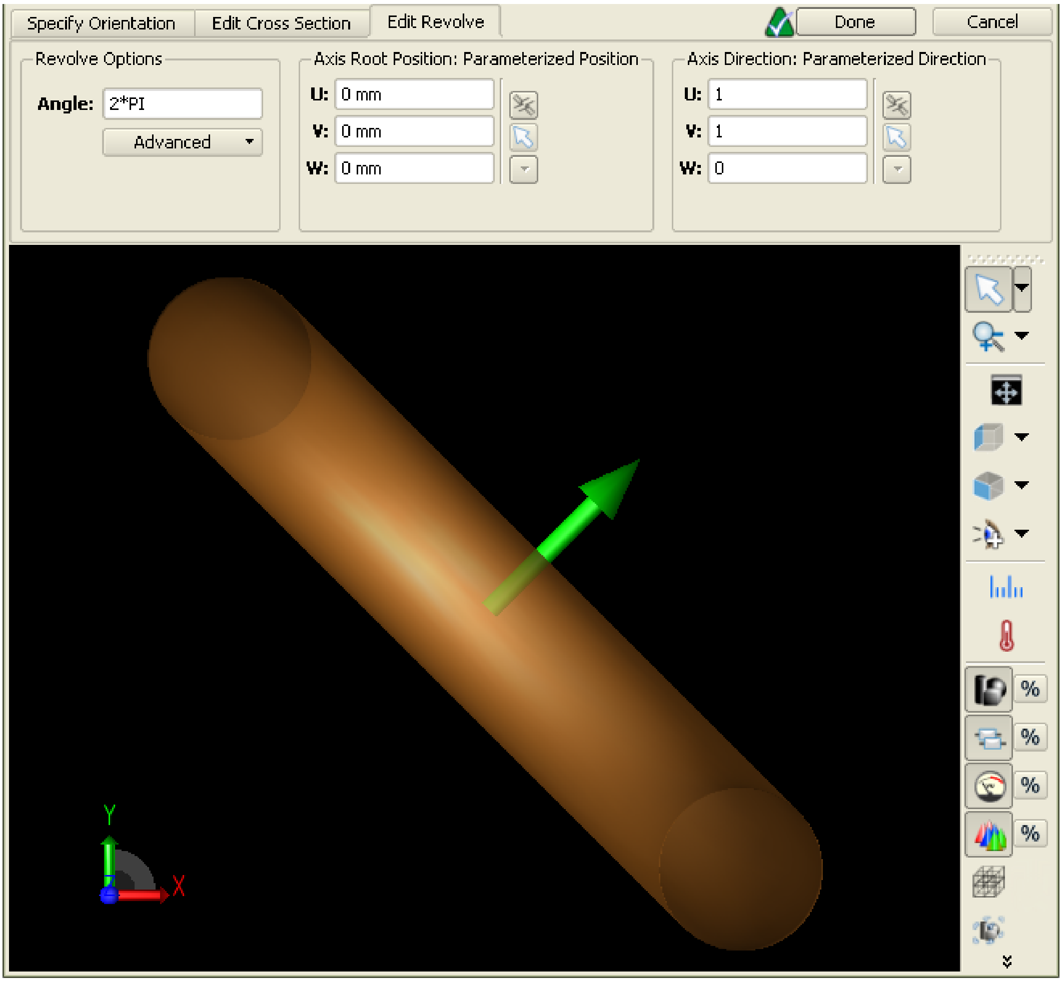


Figure 5.43: Cross section revolved about the U'- and V'-axes

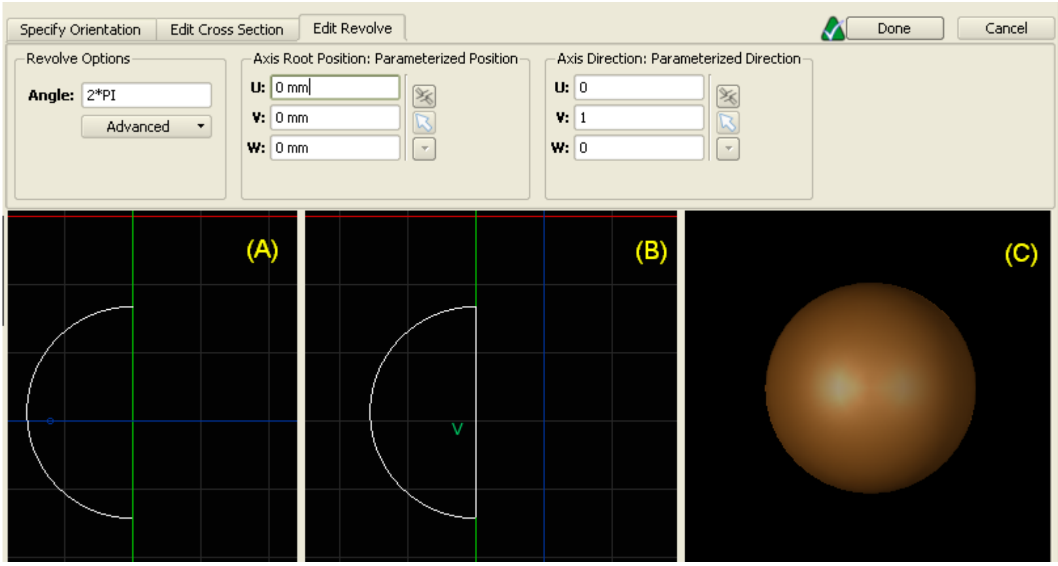


Figure 5.44: Using (A.) open (hollow) and (B.) closed (solid) cross sections to (C.) create a sphere with the Revolution operation

5.5.6 Using Locators to Orient Parts

A Locator is a triad that can be placed on a part or assembly to aid in its orientation. Locators are created by right-clicking on a part and selecting **Create New... ▶ Locator**. This creates a Locator associated with the selected part. By double-clicking on the Locator in the LOCATORS branch of the tree, the EDIT LOCATOR tab will be activated which is used to modify the location and orientation of the Locator object.

Parts can be reoriented by matching their Locators. To do this, select two locators in the tree, then right-click to show the context menu. Choose the **Match Locators** menu item. The Match Locators dialog is shown, defining which part will be moved and which will remain fixed in space. The SWAP ORDER button can be used to reverse the default. The part that will be moved by default is determined by the selection order.

Chapter 6

Creating Materials

In this chapter, you will learn...

→ how to add materials to your XFdtd project

→ how to correctly define materials and apply them to geometric objects

Once objects are created and situated correctly in the XFdtd simulation space, material definitions must be added or else the project will not be considered valid. The tabbed windows within the MATERIAL EDITOR interface are used to define electric and/or magnetic materials based on their constitutive parameters. After adding materials to the project, simply drag and drop the material in the PROJECT TREE onto the desired geometry to apply it to that object.

6.1 Adding a New Material

To add a new material, right-click on DEFINITIONS:MATERIALS branch of the PROJECT TREE and select NEW MATERIAL DEFINITION, as seen in Figure 6.1. XFdtd adds a MATERIAL object to this branch and assigns it a color. The color is assigned from an existing color sequence, in conjunction with examining the colors already assigned to existing materials to avoid colors that are already in use.

Depending on the project preferences, the MATERIAL EDITOR window will appear automatically. If not, simply double-click on this object to bring up the editor. Similarly, double-click on any existing Material icon to edit an existing material within the Material Editor.

- See Section 4.1.2 for information about project preference definitions.

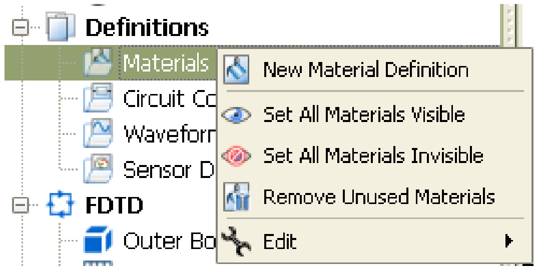


Figure 6.1: Adding a New Material Definition to the project

Once the MATERIAL EDITOR window is open, enter the name of the material in the NAME dialog box. Define the TYPE of material as PHYSICAL or FREE SPACE. FREE SPACE is the most basic material definition. Every other type of material is included within the PHYSICAL definition, in which case the ELECTRIC and MAGNETIC types should be assigned in their respective drop-down lists.

There are five electrical and magnetic material types available in XFdtd:

- FREE SPACE
- PERFECT CONDUCTOR
- ISOTROPIC
- DIAGONALLY ANISOTROPIC
- ANISOTROPIC (ELECTRIC only)

Although frequency-independent materials require the least memory during calculations, there are some cases in which frequency-independent materials are not appropriate. Frequency-dependent (or dispersive materials) should be used in these instances. Some common examples of frequency-dependent materials are high water content materials such as human tissues and metals excited at optical frequencies. XFdtd has the capability of simulating electric and magnetic Debye and Drude materials such as plasmas, Lorentz materials and anisotropic magnetic ferrites, as well as frequency-independent anisotropic dielectrics and nonlinear diagonally anisotropic dielectrics. These additional sub-types are specified within the ISOTROPIC, DIAGONALLY ANISOTROPIC and ANISOTROPIC definitions.

The following sections will detail the various types of materials.

6.1.1 Free Space

FREE SPACE is the most basic material. By default, all FDTD cells are initialized as free space, and the fields at all cell edges are updated using the Free Space equations. This material sets relative permittivities and permeabilities to one and conductivities to zero.

Figure 6.2 shows the MATERIAL EDITOR when the FREE SPACE material is defined. Notice that no ELECTRIC or MAGNETIC tab is available, since both are defined as FREE SPACE material.

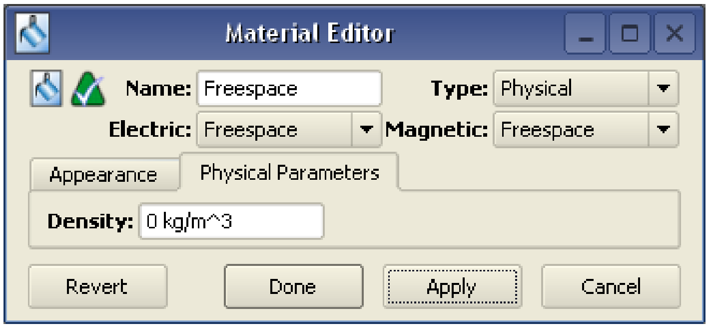


Figure 6.2: Defining a Free Space material

6.1.2 Perfect Conductors

A PERFECT CONDUCTOR has infinite conductivity and all fields found within it are zero. It has the same settings as the FREE SPACE material, as seen in Figure 6.2 above. It should typically be used as an approximation when a good conductor is needed in an electromagnetic calculation and losses aren't important. Attempting to include the effects of a good conductor (rather than perfect conductor) may be difficult since the wavelength inside the good conductor will become very small, requiring extremely small FDTD cells to provide adequate sampling of the field values inside the material. This can, however, be overcome by checking the SURFACE CONDUCTIVITY box in the EDIT MATERIAL dialog.

▸ You can read more about the SURFACE CONDUCTIVITY box in Section 6.1.3.4.

6.1.3 Electric Materials

6.1.3.1 Isotropic Materials

XFdtd includes several ISOTROPIC materials:

- NONDISPERSIVE
- DEBYE/DRUDE
- LORENTZ
- SAMPLED
- NONLINEAR

Figure 6.3 shows the MATERIAL EDITOR when an ISOTROPIC material is defined. Note that only the ELECTRIC tab is available since MAGNETIC is defined as FREE SPACE.

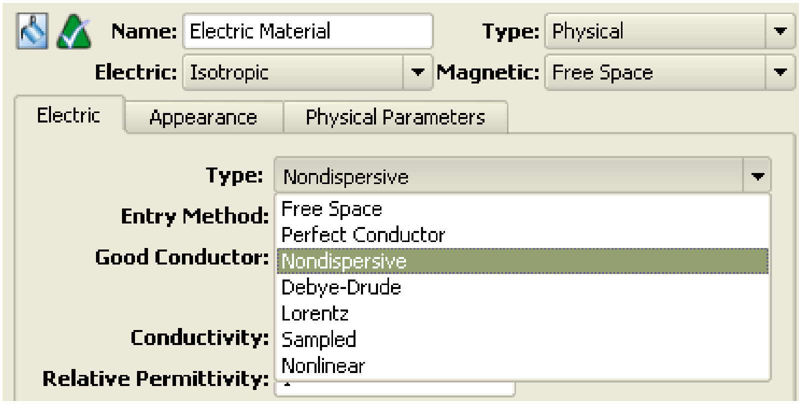


Figure 6.3: Defining an Electric material

Nondispersive. Nondispersive material properties do not vary with frequency. The continuous-time expressions of Maxwell's equations for linear, isotropic, nondispersive materials discretized in XFdtd are:

$$\epsilon\frac{\partial E(r,t)}{\partial t} = \bigtriangledown \times H(r,t) - \sigma E(r,t) \text{ and } \mu\frac{\partial H(r,t)}{\partial t} = - \bigtriangledown \times E(r,t) - \sigma^* H(r,t)$$

where

ϵ represents the electric permittivity

σ represents the electric conductivity

μ represents the magnetic permeability

σ^* represents the magnetic conductivity

Nondispersive isotropic materials also display a GOOD CONDUCTOR control. This control effects whether the material is considered a Good Conductor (i.e. high-conductivity) material for the purposes of XACT meshing (see Chapter 8.3.4). The default Automatic setting defines any material with a conductivity of at least 100 S/m as a Good Conductor; however, the On and Off settings can be used to override that behavior.

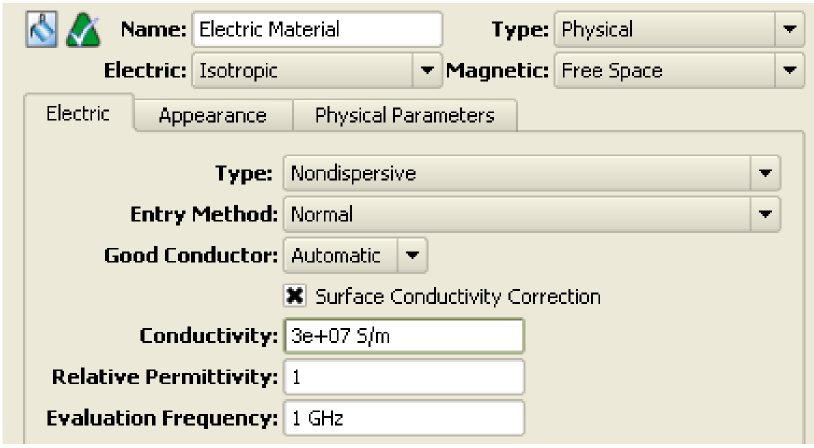


Figure 6.4: Defining a Nondispersive material

Debye/Drude. For a Debye/Drude material, the electrical Conductivity (σ) in $\frac{S}{m}$, Infinite Frequency Relative Permittivity (ϵ_∞), Number of Poles, Static Relative Permittivity (ϵ_r) and Relaxation Time (τ) in seconds must be specified. For a Debye material, σ must equal zero. A non-zero conductivity value results in a Drude material.

- This is discussed in detail in Chapter 8 of the Kunz and Leubbers text [1].

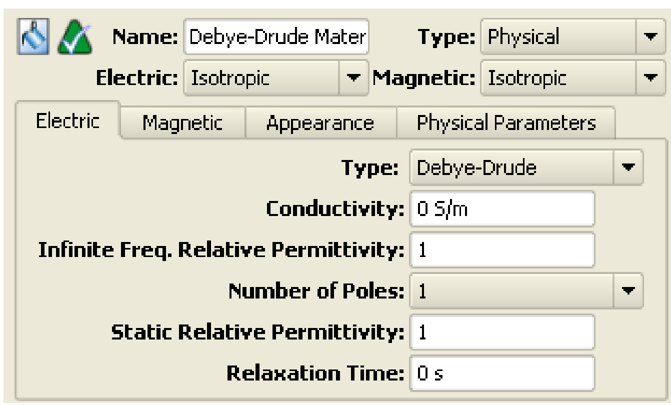


Figure 6.5: Defining a Debye/Drude material

These parameters cannot be set arbitrarily or instability can occur. One constraint is that the XFdtd timestep must be small enough to accurately calculate the transient behavior of the material. If the XFdtd timestep is three percent of the relaxation time or smaller, the time variation of the material parameters should be sufficiently resolved. Typically, the XFdtd timestep is a very small fraction of the relaxation time. In order to be clear about the signs in the following discussion, note that we are using the engineering time variation of:

$$-j\omega t$$

and we are defining the complex permittivity as:

$$\epsilon = \epsilon^{'} - j\epsilon^{''}$$

For the XFdtd calculation to be stable, the imaginary (loss) part ($\epsilon^{''}$) of the complex permittivity, including the effect of the conductivity term, must be positive for all frequencies from zero frequency to infinite frequency. This condition results in a passive material. If $\epsilon^{''}$ is negative, then the material has gain and the XFdtd calculation will become unstable as the field amplitudes grow.

- See equation 8.29 of the Kunz and Leubbers text [1].

For a DEBYE material ($\sigma = 0$), stability is assured by setting ϵ_r to a larger value than ϵ_∞. In order to have realistic behavior at high frequencies, ϵ_∞ should be no less than one and should not be much larger than one. Thus the condition for strictly DEBYE material to be stable is:

$$\begin{aligned} \epsilon_\infty &\geq 1 \\ \epsilon_r &> \epsilon_\infty \\ \sigma &= 0 \end{aligned}$$

If σ is not zero, then the material has DRUDE behavior. There are different conditions that can be satisfied for the imaginary part of the complex permittivity to be positive so that XFdtd produces stable results. If the static permittivity is greater than the infinite frequency permittivity then σ can have any positive value. This results in the simplest set of conditions for a stable DRUDE Material:

$$\begin{aligned} \epsilon_\infty &\geq 1 \\ \epsilon_r &> \epsilon_\infty \\ \sigma &\geq 0 \end{aligned}$$

These conditions are too restrictive to allow general DRUDE materials to be included in XFdtd. The more general DRUDE conditions are:

$$\begin{aligned} \epsilon_\infty &\geq 1 \\ \textbf{If } (\epsilon_r < \epsilon_\infty)\textbf{, then: } \sigma &\geq \epsilon_0 \frac{\epsilon_\infty - \epsilon_r}{\tau} \\ \textbf{otherwise: } \sigma &\geq 0 \end{aligned}$$

where ϵ_0 is the FREE SPACE PERMITTIVITY of 8.854e-12 $\frac{F}{m}$.

- More general conditions for DRUDE materials can be determined from the discussion in Chapter 8, Section 3 the Kunz and Leubbers text [1].

Lorentz. Stability in LORENTZ materials should be obtained as long as CONDUCTIVITY ≥ 0 and the XFdtd timestep is 3% of the relaxation time or less. The limits on the material parameters are:

$$\begin{aligned} \sigma &\geq 0\frac{S}{m} \\ \epsilon_\infty &\geq 1 \\ \epsilon_r &\geq \epsilon_\infty \\ ResonantFrequency\ (\Omega) &> 0 \\ DampingCoefficient &> 0 \end{aligned}$$

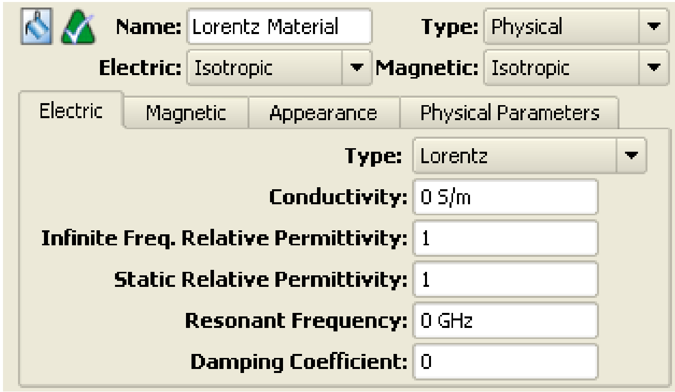


Figure 6.6: Defining a Lorentz material

Sampled. This material allows the user to enter relative permittivities and conductivities sampled at various frequencies.

The conductivity and permittivity of the material are interpolated using EVALUATION FREQUENCY from the data provided below. The EVALUATION FREQUENCY frequency should be roughly in the middle of the frequency range of interest. If the conductivity and/or permittivity vary significantly over your range of interest, your results, such as S-Parameters, may deviate from what you would obtain by using a sinusoidal waveform at specific frequencies.

Note that this is not a dispersive material and will not automatically be converted to one.

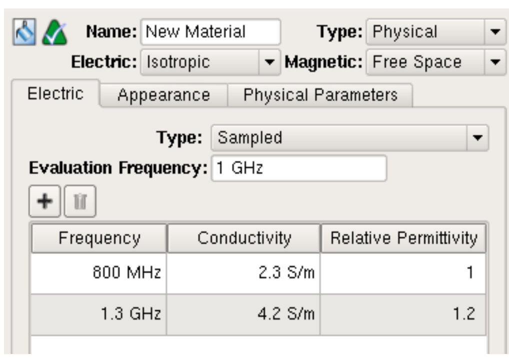


Figure 6.7: Defining a Sampled material

Nonlinear. The relative permittivity of a nonlinear isotropic dielectric material satisfies:

$$\epsilon_r = \epsilon_{r2} + \frac{\epsilon_{r1} - \epsilon_{r2}}{1 + a_1\left(\frac{|E| - E_s}{E_0}\right)^2 + a_2\left(\frac{|E| - E_s}{E_0}\right)^4 + a_3\left(\frac{|E| - E_s}{E_0}\right)^6}$$

Where

ϵ_r is relative permittivity

E is instantaneous cell edge E-field

ϵ_{r1} is static (low $|E|$) relative permittivity

ϵ_{r2} is infinite $|E|$ relative permittivity

E_s is the E magnitude above which the material becomes non-linear

E_0 is a scaling term

a_1, a_2 and a_3 are coefficients

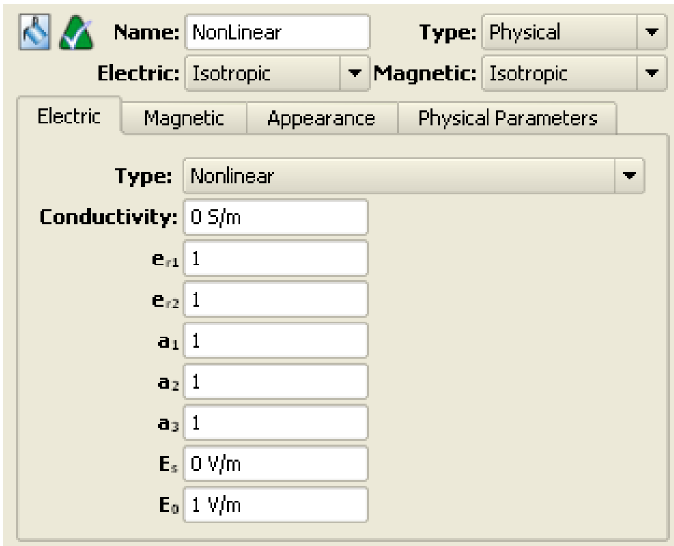


Figure 6.8: Defining a Nonlinear material

6.1.3.2 Diagonally Anisotropic

The definitions for a DIAGONALLY ANISOTROPIC are equivalent to those corresponding definitions detailed for ISOTROPIC materials, except the definitions in each of the principle directions are independently specified.

6.1.3.3 Anisotropic

Frequency-independent ANISOTROPIC materials are defined in XFdtd by the relative permittivity, $\bar{\bar{\epsilon}}$, and CONDUCTIVITY, $\bar{\bar{\sigma}}$, tensors.

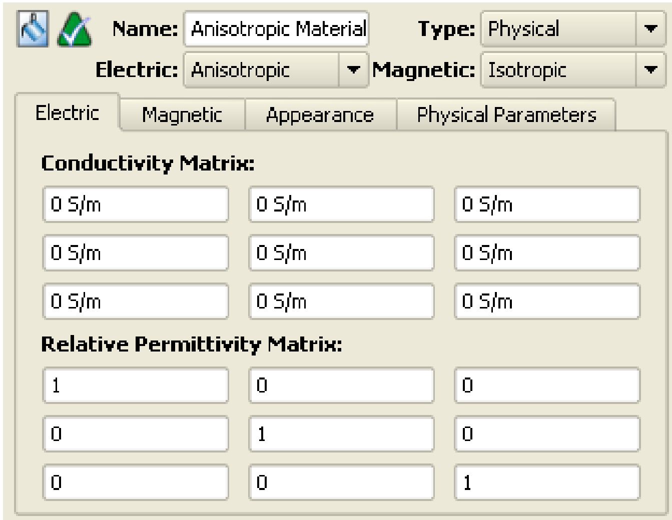


Figure 6.9: Defining an Anisotropic material

The parameters below CONDUCTIVITY represent the terms of $\bar{\bar{\sigma}}$ and the parameters below Permittivity (Infinite Frequency) represent the terms of $\bar{\bar{\epsilon}}$ as follows:

$$\begin{pmatrix} \sigma_x x & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix}$$

$$\begin{pmatrix} \epsilon_{yxx} & \epsilon_{yxy} & \epsilon_{yxz} \\ \epsilon_{yyx} & \epsilon_{yyy} & \epsilon_{yyz} \\ \epsilon_{yzx} & \epsilon_{yzy} & \epsilon_{yzz} \end{pmatrix}$$

The conductivity and permittivity for frequency-independent anisotropic dielectric materials are represented by $\bar{\bar{\sigma}}$ and $\bar{\bar{\epsilon}}$, unlike the equations for linear, non-dispersive, frequency-dependent, isotropic materials. These are used in the time-domain FDTD update equations in place of ϵ and σ :

$$\bar{\bar{\epsilon}}\frac{\partial E(r,t)}{\partial t} = \bigtriangledown \times H(r,t) - \bar{\bar{\sigma}}E(r,t)$$

6.1.3.4 Note on Complex Permittivity, Loss Tangent and Surface Conductivity Correction

Complex permittivity. The value of complex permittivity may need to be calculated for XFdtd for some materials. The real part of the complex permittivity may be used for the relative permittivity. The conductivity can be calculated from the imaginary part of the complex permittivity by multiplying by a desired output frequency value (in radian frequency), as shown by:

$$\sigma = \omega_o \epsilon^{"} \epsilon_o$$

Loss tangent. The loss tangent can be entered directly into XFdtd when it is known, typically for good dielectrics. The engine can then calculate the conductivity as a function of frequency using:

$$\sigma_{eff} = \omega_o \epsilon^{'} \tan \delta$$

Surface conductivity correction. In the case of a frequency-dependent conductor, the SURFACE CONDUCTIVITY CORRECTION box can be checked to correct the conductivity of a material for a single frequency of a sinusoidal excitation. This is necessary in cases where the penetration and loss in good conductors needs to be included in the calculation. Enabling this option allows for the reduction of wavelength in these materials without reducing the cell size to maintain the 10 $\frac{cell}{wavelength}$ limit.

6.1.4 Magnetic Materials

XFdtd also includes several types of magnetic materials. Many of these materials are simply the magnetic counterpart to the dielectrics described in the Electric Materials section (Section 6.1.3.) All restrictions noted in the Electric Materials section apply to their magnetic counterparts.

Figure 6.10 shows the MATERIAL EDITOR when a Magnetic ISOTROPIC material is defined. Note that only the MAGNETIC tab is available since ELECTRIC is defined as FREE SPACE. If ELECTRIC was defined as another type, a ELECTRIC tab would be available as well.

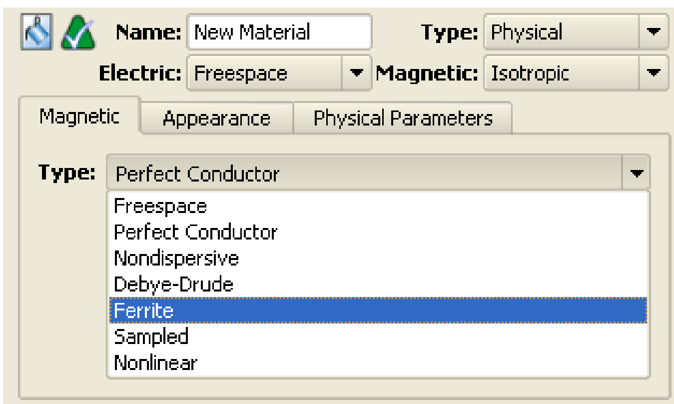


Figure 6.10: Defining a Magnetic material

6.1.4.1 Isotropic

Nondispersive.

- See ***Nondispersive*** under the electric materials section above (6.1.3.1.)

Debye/Drude.

- See ***Debye/Drude*** under the electric materials section above (6.1.3.1.)

Magnetized Ferrites. The first parameter related to magnetized ferrites is the APPLIED FIELD, (H_o.) Enter its value in units of $\frac{A}{m}$. This number will be used to calculate the Larmor precession frequency (ω_o),

$$\omega_o = \gamma H_o$$

where γ is the gyromagnetic ratio $(2.21x10^5 \frac{m}{C}.)$

Next, enter the INTERNAL MAGNETIZATION (M_o) in units of T. This number is used to calculate the saturation frequency (ω_m),

$$\omega_m = \gamma \frac{M_o}{\mu_o} .$$

Next, use the DAMPING COEFFICIENT to account for damping in the ferrite or of any absorption of power due to the ferrite. Finally, enter the direction of the biasing field using the spherical direction fields THETA and PHI.

- There are several informative references that discuss the form of the permeability tensor used for the ferrites [9, 10, 11, 12]. (The first two references do not discuss the damping coefficient.)
- See the Kung text for parameters for some commercially available ferrites [12].

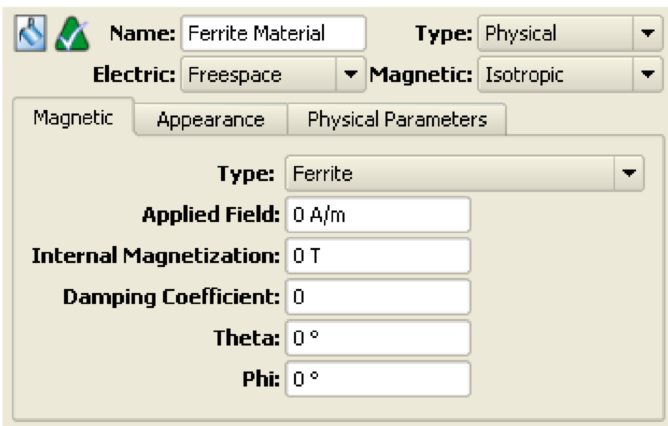


Figure 6.11: Defining an Magnetized Ferrite material

Sampled.

- See ***Sampled*** under the electric materials section above (6.1.3.1.)

Nonlinear.

- See ***Nonlinear*** under the electric materials section above (6.1.3.1.)

6.1.4.2 Diagonally Anisotropic

- See ***Diagonally Anisotropic*** under the electric materials section above (6.1.3.2.)

6.1.5 Thin Wire Material

XFdtd includes THIN WIRE material for use in in special situations where a wire with a radius much smaller than the cell size is required. In most cases, standard PEC material will serve well as a wire. However, in cases where the wire radius is important to the calculation and is less than one fourth the length of the average cell edge, the thin wire material may be used to accurately simulate the correct wire diameter.

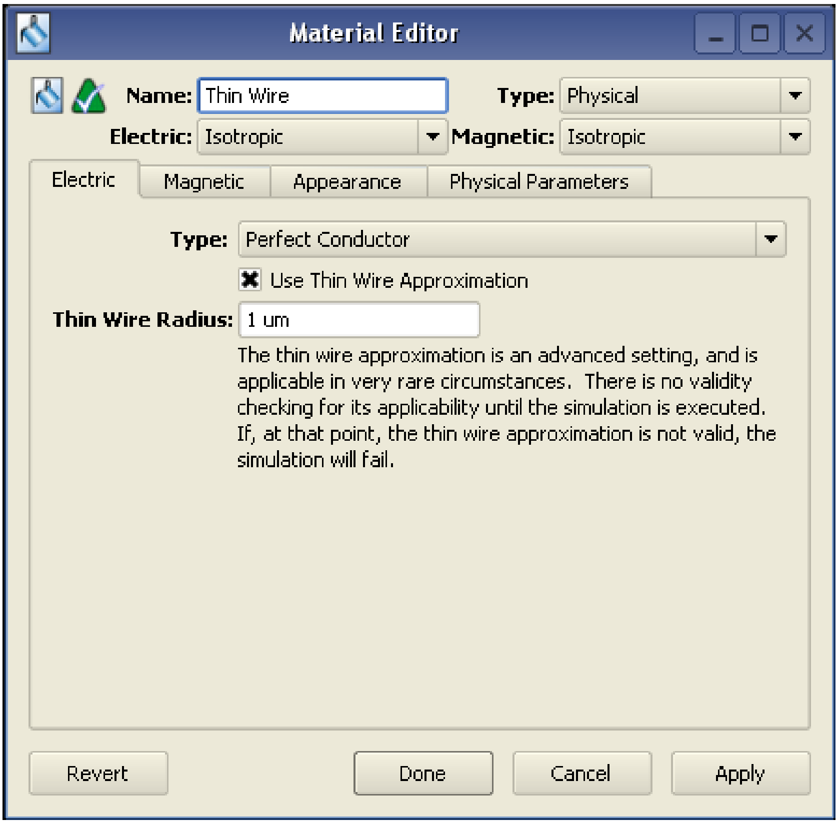


Figure 6.12: Thin Wire dialog window

Thin wire materials are subject to several restrictions due to the special modifications they require on the field update equations. Thin wires may not meet at 90 degree angles, and they may not be located on adjacent cell edges. Attaching thin wire materials end-to-end is the desired method of using this material type. Also, thin wires may not be staircased as this will lead to instabilities in the calculation.

If a feed or port will be connected to a thin wire, the cell edge at the feed should also be thin wire material, not free space. For example, if a dipole antenna is simulated in thin wire material, the feed "gap" should also be thin wire. Failing to use thin wire material in the feed gap will lead to less accurate results. If you do add a feed or port to thin wire materials, you may not specify inductance or capacitance values to that port because these will interfere with the thin wire calculation. However, resistor values (series or parallel) may be used.

Using thin wires is applicable only in very rare circumstances. XFdtd does not provide validity checking for its applicability until the simulation is run. If, at that point, the thin-wire approximation is invalid, the simulation will fail.

Thin wire material is only available in certain electrical and magnetic material combinations. In all cases, the **Type** selector (appears on the Electrical tab) of the electrical material must be set to Perfect Conductor. The following table contains appropriate Thin Wire combinations.

Table 6.1: Thin Wire Materials

Electrical Type	**Magnetic Type**
Isotropic	Free Space
Diagonally Anisotropic	Free Space
Isotropic	Perfect Conductor
Diagonally Anisotropic	Perfect Conductor
Isotropic	Isotropic
Diagonally Anisotropic	Isotropic
Isotropic	Diagonally Anisotropic
Diagonally Anisotropic	Diagonally Anisotropic

6.2 Appearance

Use the APPEARANCE tab to assign the aesthetic properties of each defined material. Colors and other properties can be assigned to the faces, edges and vertices of objects that contain the material so that they can be easily distinguished from other materials in the project.

6.3 Physical Parameters

The PHYSICAL PARAMETERS tab governs the definitions most commonly associated with biological tissue. These definitions are thus necessary only when performing biological calculations. These values are computed automatically for tissues in the Remcom high fidelity meshes.

6.4 Material Visibility

XFdtd offers the user the ability to toggle the visibility of materials. It is important to know that a material set as invisible will hide all parts defined by that material in both the geometry space and the mesh.

To toggle the visibility of a material, right-click on the material in the PROJECT TREE and select SET VISIBLE, SET INVISIBLE or SET ONLY THIS MATERIAL VISIBLE, as seen in Figure 6.13.

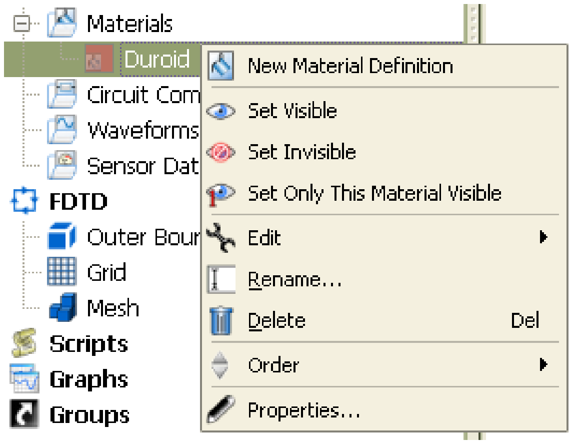


Figure 6.13: Setting the visibility of a material

✓ All parts that are made invisible by an invisible material are easily identified in the PARTS LIST.

▶ See Section 4.2.1 for more on the Parts List.

Chapter 7

Circuit Components, Waveguides, External Excitations and Voltage Points

In this chapter, you will learn about...

→ adding passive and active circuit elements to your XFdtd project,

→ adding excitations, such as plane waves and Gaussian beams,

→ adding waveguide interfaces (ports and terminations),

→ adding static voltage points.

This chapter describes how to create and define valid components, waveguide boundaries, and excitations in an XFdtd project.

CIRCUIT COMPONENTS are discrete components which are added to the geometry. Examples include resistors, capacitors, inductors, voltage sources, current sources, switches and diodes. Definitions for the components are created in the CIRCUIT COMPONENT DEFINITION EDITOR.

Waveguide boundaries are matched terminations and ports for guided wave structures, including rectangular, circular, microstrip, stripline, coplanar, and many others. The boundaries are rectangular regions, usually placed at or near an end of a guided wave structure. They can be placed anywhere within the problem space but must be oriented perpendicular to a principle axis.

There are four primary input excitation forms in XFdtd. The first type of excitation can be applied at any discrete location with a voltage or current source. The second is a modal excitation through a waveguide port. The third and fourth excitations are applied externally in the form of incident PLANE WAVES for scattering calculations, or GAUSSIAN BEAMS for optical frequency calculations. The later two excitations are configured using the PLANE WAVE EDITOR and GAUSSIAN BEAM EDITOR.

The WAVEFORM EDITOR governs the time variation of all the types of excitations. A WAVEFORM may be defined as various types of pulses for broadband calculations, a sinusoidal source, or a user-defined waveform. A waveform definition must be applied to a defined excitation for it to be valid.

STATIC VOLTAGE POINTS are inputs to the Laplace static solver which initializes E fields of PEC edges before the FDTD computation starts. Static voltage points are created and parameters are set by right clicking in the STATIC VOLTAGE POINTS branch of the Project Tree.

7.1 Component Tools

Located in the first drop-down menu of the GEOMETRY workspace window, the COMPONENT TOOLS dialog allows users to add discrete components to the XFdtd project. These components include voltage sources, current sources, feeds, lumped resistors (R), capacitors (C), inductors (L), diodes, non-linear capacitors and switches. Discrete sources (such as voltage and current sources) are locations at which the electric field is modified by the addition of some type of input waveform. This dialog is also used to set the properties of each component, such as spatial orientation, polarity and alignment.

Definitions such as resistance, inductance, capacitance, etc. are specified in the CIRCUIT COMPONENT DEFINITION EDITOR so that they can be reused for multiple components if necessary. To access this editor, navigate to the DEFINITIONS:CIRCUIT COMPONENT DEFINITIONS branch of the PROJECT TREE and double-click on the component's CIRCUIT COMPONENT DEFINITION object.

Figure 7.1 shows the various components available within the COMPONENT TOOLS dialog.

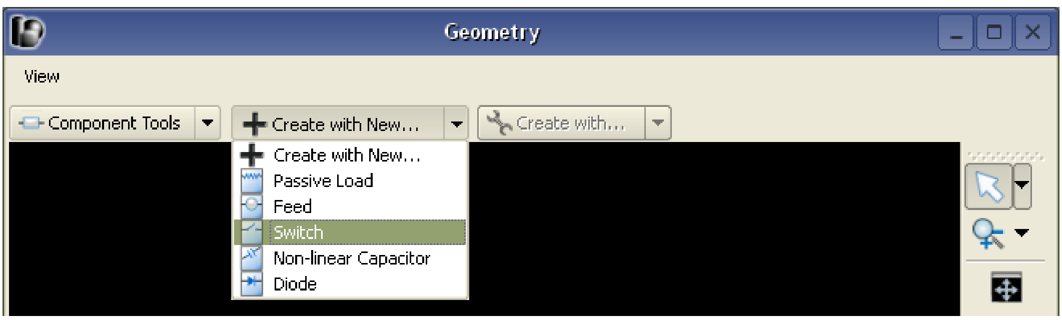


Figure 7.1: Adding a new component with Component Tools

7.1.1 Adding a New Component

Once COMPONENT TOOLS is visible in the first-drop down button of the GEOMETRY workspace window, clicking CREATE WITH NEW will prompt a drop-down menu to appear. This menu includes:

PASSIVELOAD FEED SWITCH
NON-LINEAR CAPACITOR DIODE

Selecting any of these components will prompt a similar series of tools to place the desired component into the simulation space. The following sections will detail the definitions associated with these tools.

- For more information about the discrete sources, refer to Section 7.2.

CIRCUIT COMPONENTS are also added by right-clicking in the PROJECT TREE as in Figure 7.2.

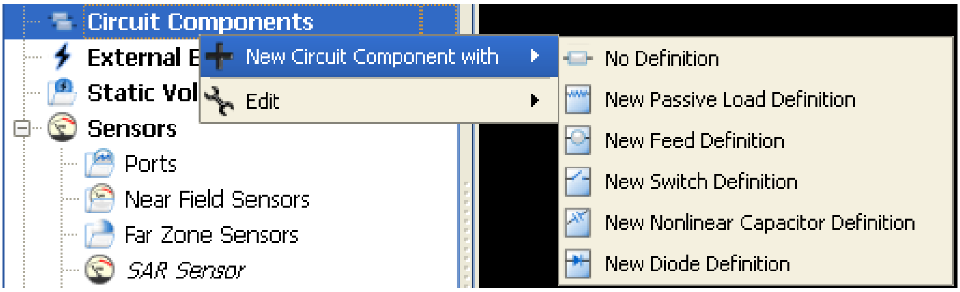


Figure 7.2: Accessing the Component Tools dialog from the Project Tree

Connections tab

Discrete sources are placed along a cell edge on which the electric field is modified by the addition of some type of input waveform. The X, Y and Z values defined in the CONNECTIONS tab specify the location of the desired cell edge in the mesh. Users may enter this data manually in the respective fields or use the DETATCH TOOL, PICKER and MODIFY tools to define the appropriate data points.

- DETATCH TOOL: The Detach tool enables the user to detatch an endpoint of a discrete source from the geometric object to which it is attached. This tool is only available if the user placed one or both endpoint(s) with the Picker tool. When detatched from the object, the endpoint will not move when the object moves or is modifed (changes in extrude length, for example.)
- PICKER TOOL: The Picker tool provides an alternate way for the user to place a discrete source. By associating an endpoint with a specific location on the geometry of a part, the discrete component will move (expand or contract) if the part moves. To use the Picker Tool, click the icon by Endpoint 1 and place the first endpoint on the part. Then, click the icon by Endpoint 2 and place the second endpoint on the part.
- MODIFY TOOL: The Modify tool enables the user to exert pinpoint control over the location of the discrete source. It contains two sections: **Parameters** and **Offsets**.

  - Parameters: The Parameters section contains two sliders, U and V that enable the user to make minute adjustments to the location of the endpoint of a discrete source in the x- and y-planes.

  - Offsets: The Offsets section enables users to offset the endpoint of a discrete source for X, Y and Z.

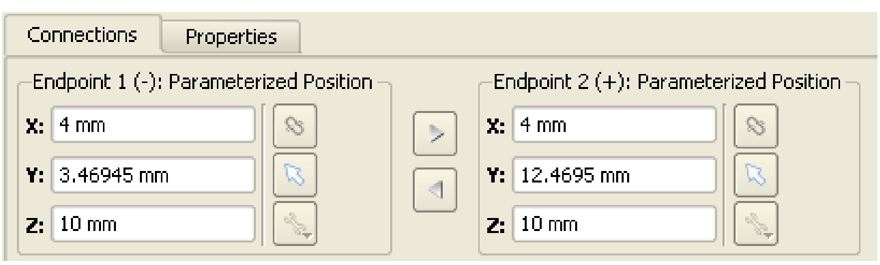


Figure 7.3: Editing component connections

Properties tab

In the PROPERTIES tab, the name of the component, direction, and polarity is defined. The component(s) can be aligned with either the X-, Y- or Z-axes, by selecting X, Y or Z, respectively. Otherwise, the default AUTO selection will automatically align the component based on ENDPOINT 1 and ENDPOINT 2 defined in the CONNECTIONS tab.

The user can include the component in the mesh or assign the component as a port by checking INCLUDED IN MESH or THIS COMPONENT IS A PORT, respectively. When the latter is selected, XFdtd will automatically add a PORT sensor at this location.

- Keep in mind that a port that contains only passive components cannot be the active port.
- If lumped reactive elements are used in an active port specification, the resulting S-Parameters, though valid for the complex system impedances defined, will not correspond to the (usually 50 ohm-based) S-Parameters that would be measured for the system on a network analyzer.
- See Section 10.5 for more about port sensors and the data that they collect.

Clicking the ADVANCED button displays several additional controls that affect the generation of the computational grid in the vicinity of the component. The EVENLY SPACED IN ORTHOGONAL DIRECTIONS check box enforces locally uniform grid cell sizes at the location of the component in each of the two directions perpendicular to the component in the mesh. For instance, if the component is Z-directed, enabling even spacing will cause the X grid cell sizes to be equal and the Y grid cell sizes to be equal on all cells adjacent to the component in the mesh. As this typically reduces the size of grid cells in the vicinity of the component, it is unchecked by default and should be left off except in special cases. The ENABLE FIXED POINT ON ENDPOINT 1 and ENABLE FIXED POINT ON ENDPOINT 2 checkboxes allow you to instruct the Grid to pass X-, Y-, and Z-directed grid lines through the location of the given endpoint.

Figure 7.4 displays the PROPERTIES tab for editing a feed.

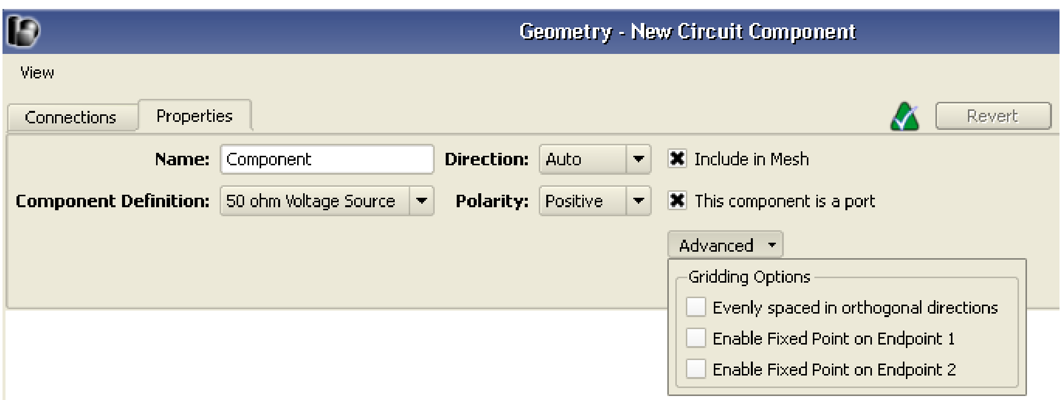


Figure 7.4: Editing component properties

- If a component is added before it is defined in the CIRCUIT COMPONENT DEFINITION EDITOR, a default definition will be created so that the component is valid. To edit its properties, double-click on this default definition under the DEFINITIONS branch. Likewise, if the component requires a waveform definition, a default definition will be added to this branch.

7.1.2 Adding a Component Using an Existing Definition

The CREATE WITH drop-down menu functions similarly to the CREATE WITH NEW drop-down menu described above except that a pre-existing component definition is applied to the component that is to be added. For this reason, the CREATE WITH menu is not active until a component definition has already been created within the CREATE WITH NEW dialog- it is a list all of the pre-existing component definitions that have *already* been added to the project. This menu makes it easy to add identical components to a project.

7.1.3 Component Visibility

XFdtd offers the user the ability to toggle the visibility of components. Unlike Materials, when a component is set as invisible it is only hidden in the geometry space. It is not hidden in the mesh. To hide a component from the mesh, un-mark the INCLUDE IN MESH checkbox.

To toggle the visibility of a component, right-click on the component in the PROJECT TREE and select either SET VISIBLE, or SET INVISIBLE, as seen in Figure 7.5.

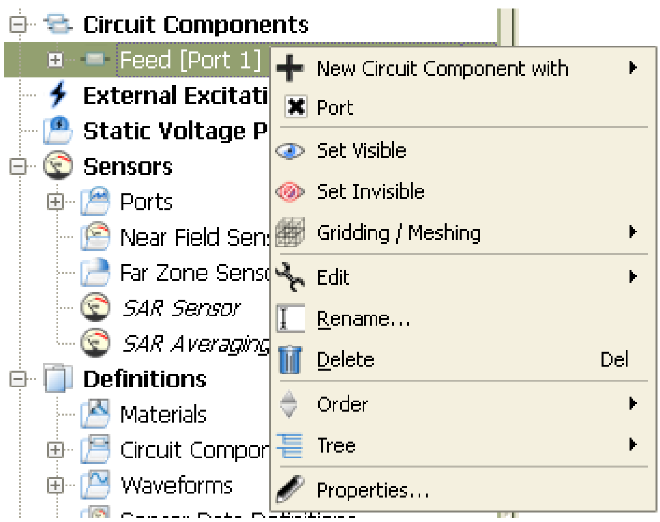


Figure 7.5: Setting the visibility of a component

7.2 Circuit Component Definition Editor

The CIRCUIT COMPONENT DEFINITION EDITOR is used to define the parameters associated with discrete components. Components such as voltage sources, current sources, feeds, lumped resistors (R), capacitors (C), inductors (L), diodes, nonlinear capacitors and switches are defined in this window.

It is important to recognize the purpose of the COMPONENT TOOLS interface versus that of the CIRCUIT COMPONENT DEFINITION EDITOR. The former places the physical component into the project (and creates an object in the PROJECT TREE that represents the actual component), while the latter creates a

CIRCUIT COMPONENT DEFINITION object for the parameters of that component (or components) that can be used over and over again by dropping it onto multiple components.

The CIRCUIT COMPONENT DEFINITION EDITOR is accessed by double-clicking on any object found in DEFINITIONS:CIRCUIT COMPONENT DEFINITIONS branch. Figure 7.6 shows the sample setup for a feed. Note that the TYPE of component determines the diagram in the editor.

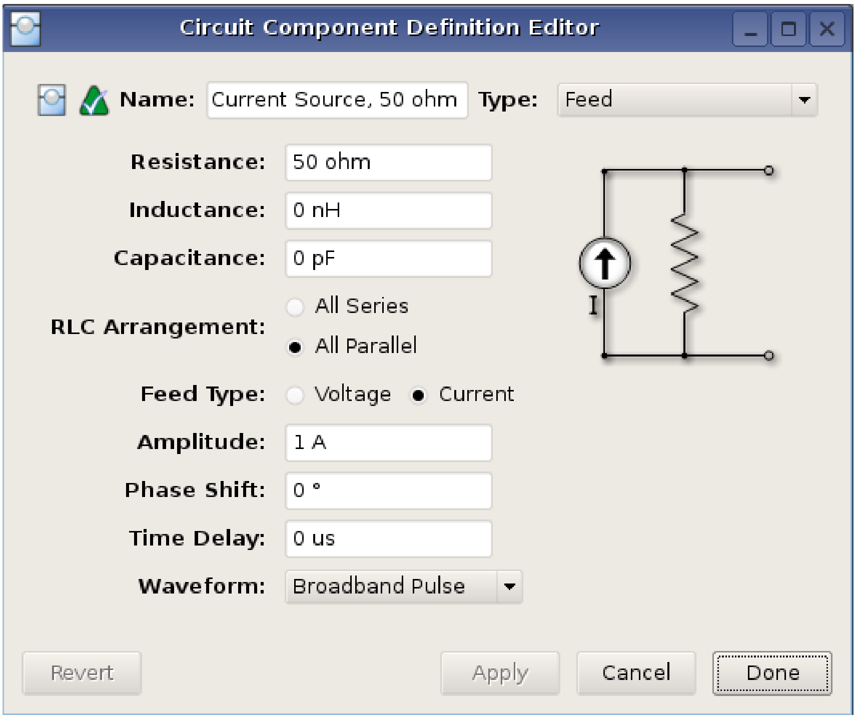


Figure 7.6: The Circuit Component Definition Editor

7.2.1 Passive Load

When a component with no source is desired, the user may create it by selecting PASSIVE LOAD. Passive components include lumped resistors (R), inductors (L) and capacitors (C). Since passive lumped loads do not radiate energy, they may be added to the calculation when either PLANEWAVE or GAUSSIANBEAM excitations are selected.

▶ See Sections 7.5 and 7.6 for more about these excitations.

The RLC elements lumped at component locations can be combined in several configurations:

- The All Series configuration places a resistor, an inductor, and a capacitor in series. Each of these RLC elements are optional, and can be removed (shorted) by setting the appropriate element parameter to 0. For instance, the series capacitor can be removed by entering a capacitance of 0 F, resulting in a resistor in series with an inductor.

- The All Parallel configuration places the resistor, inductor and capacitor in parallel with each other. Each of these RLC elements are optional and can be removed by setting the appropriate element parameter to 0. For instance, the parallel inductor can be removed by entering an inductance of 0 H, resulting in a resistor in parallel with a capacitor.
- The RL || C configuration places the series combination of resistor and inductor in parallel with a capacitor. This arrangement may be useful for modelling a non-ideal inductor that includes the effects of wire resistance and self-capacitance. In this configuration, none of the RLC elements are optional, and all element parameters must be specified with values greater than zero.

The selected configuration is shown schematically in this window. Regardless of the configuration, the resulting lumped impedance acts upon a single FDTD mesh edge.

There are two ways the RLC values can be specified. Both methods result in a single set of RLC values being used for simulation. The first, a NORMAL RLC specification, is appropriate when the equivalent circuit can be represented as constant RLC values over all frequencies. Figure 7.7 shows a PASSIVE LOAD with a NORMAL RLC Specification.

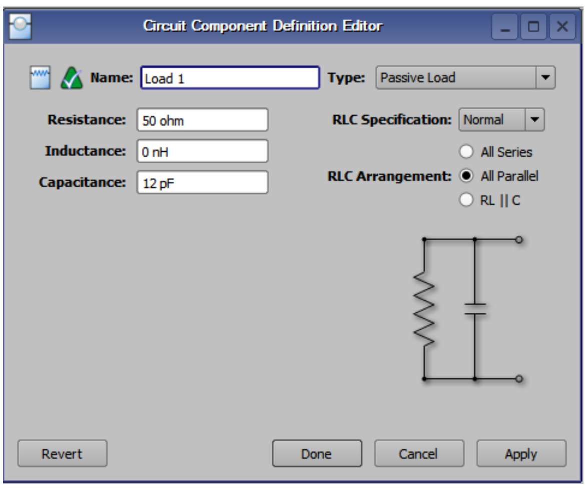


Figure 7.7: Editing a normal load within the Circuit Component Definition Editor

The second specification method, SAMPLED, is intended to represent a component whose equivalent RLC values are frequency dependent. The single set of RLC values to be used for simulation is determined by a linear interpolation between the samples using an evaluation frequency. This frequency should be near the middle of the frequency range of interest to match the impedance response as closely as possible for the entire band. If the characteristic impedance of the circuit using interpolated RLC values varies significantly over the frequency range of interest, the use of multiple sinusoidal runs instead of a single broadband run is recommended to ensure validity of results. Figure 7.8 shows a PASSIVE LOAD with a SAMPLED RLC specification.

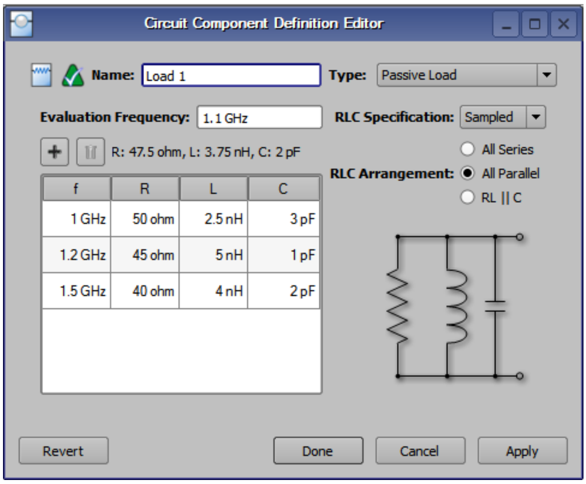


Figure 7.8: Editing a sampled load within the Circuit Component Definition Editor

Note on displacement current and lumped elements

There are physical limitations to the lumped element approximation in FDTD. Each FDTD mesh cell includes a volume of free space. That free space volume has capacitance, and a displacement current flows through it. If a lumped RLC element is specified that results in a high impedance, the free space displacement current may be significant relative to the current through the lumped elements in XFdtd. In this situation, if the displacement current is neglected, the result is non-physical and Maxwell's equations cannot be satisfied. So for lumped element calculations, the displacement current is included even though the corresponding mesh cell capacitance is not indicated on the port circuit schematic.

- ✓ To determine the relative significance of the displacement current, one can calculate the capacitance of the FDTD mesh cell. For a Z-directed component, the capacitance of the cell is given by $\frac{\epsilon\Delta_x\Delta_y}{\Delta_z}$, where Δ_x, Δ_y, and Δ_z are the mesh cell dimensions and ϵ the permittivity of the material at that cell location, usually that of free space, 8.854e-12 $\frac{F}{m}$.
- ▶ This is discussed on page 192 of the Kunz and Luebbers FDTD book [1].

As long as the impedance of this mesh cell capacitor is large compared with the impedance of the RLC lumped element circuit, its inclusion in the XFdtd calculation has a negligible effect. If the RLC lumped element circuit has an impedance comparable or larger than the mesh cell capacitance, then the inclusion of this capacitance keeps the result stable and physically correct.

- ✓ For calculating impedances, a frequency corresponding to the sine wave frequency or the highest frequency of interest within the spectrum of the excitation waveform should be used.
- ✓ Similarly, a lumped inductor should not be less than the inductance of the FDTD cell, given by $\frac{\mu\Delta}{4}$, where μ is the permeability of that point in space and Δ is the mesh cell length in the direction of the inductor.

7.2.2 Feed

An active source, or FEED, usually refers to an active component together with any passive components at the same cell edge. An active component is a cell edge on which the electric field is modified by the addition of some type of input waveform. Voltage and current sources are active components.

Series and Parallel configurations of RLC components are available in conjunction with a voltage or current source. The choice of Series or Parallel for RLC Arrangement affects only the configuration of the RLC elements, and not their relation to the source. As such, the (series or parallel) RLC load is always in series with a voltage source or in parallel with a current source.

The amplitude of the source's input waveform may be specified here, as well as the polarity. The phase of the source may be specified if the input waveform is a sinusoid. Otherwise this option will be unavailable.

The amplitudes specified for all voltage and current sources are peak values, not RMS.

A time delay may also be specified for voltage and current sources when a AUTOMATIC, GAUSSIAN, GAUSSIAN DERIVATIVE or MODULATED GAUSSIAN waveform is used. The time delay is specified in timesteps and is applied to the beginning of the source waveform. For example, if a time delay of 200 timesteps is specified for a given source port, the input waveform at that port will begin 200 timesteps later than the defined waveform. This functionality allows for any number of Gaussian excitations to be applied at different times throughout a simulation.

The voltage across the FDTD mesh edge (electric field times edge length) and the FDTD mesh edge current include the effects of both the RLC components at that mesh edge and the voltage/current source. This is illustrated in Figure 7.9 for a voltage source, V_s , in series with a source resistance, R_s. The voltage across the FDTD mesh edge is determined by the voltage source in combination with the source resistance, so that the mesh edge voltage differs from the source voltage by the voltage drop across the source resistance.

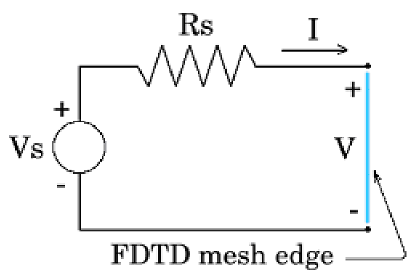


Figure 7.9: Feed schematic, including FDTD mesh edge voltage, V, and current, I

7.2.2.1 Multiple Voltage and/or Current Sources

For calculations with multiple voltage and/or current sources, such as antenna arrays or multi-port S-Parameter calculations, multiple feeds may be specified. They are specified in the SIMULATIONS workspace window before the calculation is run.

- For more about setting up an S-Parameter simulation, see Section 11.1.1.

For antenna calculations, all feeds are normally excited; however, for S-Parameter calculations only one feed can be excited for a particular XFdtd calculation. For broadband feeds, each source function must have the same pulse width but may have different amplitudes. Alternatively, they may all use the same user-supplied file of voltage versus time. The polarity can be adjusted by clicking the desired button. This may be useful in controlling the sign of the phase terms in S-Parameter calculations. For each feed, independent source resistances may be specified. For sinusoidal excitations, each feed can be specified with a different magnitude and phase.

7.2.2.2 Specifying the Source Resistance

For most XFdtd calculations, active sources will consist of a voltage source with a series source resistance. This is the default configuration for feeds. The default value for the source resistance is $50\ \Omega$, since that is the most common reference. If an S-Parameter calculation is made, the S-Parameters will be in reference to the port resistance.

- ✓ S-Parameters can be calculated for any reference impedance by changing the value of the resistance at each port.

If the value of the source resistance is not determined by the desired S-Parameter reference, then for most calculations the source resistance should be chosen to match the structure being driven. This will strongly excite the structure and also dissipate resonances most efficiently. For example, for a microstrip with $50\ \Omega$ characteristic impedance, a source resistance of $50\ \Omega$ would typically be a good choice.

For antenna calculations, determining a good approximation to an actual antenna feed is not always simple. Many antennas are fed with coaxial cable. For most XFdtd calculations the coaxial cable itself need not be meshed because it is only used to feed the antenna, and the fields inside of it are not of primary interest. The simplest approach to simulating this is to locate a port in line with the center conductor of the coaxial cable where the cable is connected to the antenna. The impedance calculated by XFdtd will then be at this point in the antenna.

- ✓ The port resistance typically should be set equal to the characteristic impedance of the coaxial cable used to feed the physical antenna. This will automatically refer S-Parameters to this resistance value.

To reduce the number of timesteps needed for the transients to dissipate for broadband calculations, include a source resistance equal to the characteristic impedance of the coaxial cable being approximated. This is similar to driving an actual circuit or antenna using a matched source.

For some situations, it is desirable to match a voltage or current source to a reactive load. In this situation, the RLC capabilities of XFdtd components can be utilized.

If the coaxial cable or other feed geometry is important to the calculation, XFdtd may be used to mesh the cable itself. In this situation it is important to determine the characteristic impedance of the coaxial cable as meshed, and to match the port resistance to the characteristic impedance.

Another important advantage to including a source resistance is reducing the number of FDTD timesteps necessary for convergence of the electromagnetic calculation. This is especially important for resonant devices, such as many antenna and microstrip circuits. With a “hard” source consisting of a voltage source without series resistance, a resonant microstrip antenna may require 64,000 timesteps for the transients to dissipate. The addition of a source resistance might reduce this to 4,000 timesteps. Similar time savings may be encountered for microwave circuits.

7.2.3 Diode

A DIODE can be defined by specifying the following:

- SATURATION CURRENT (I_s)
- JUNCTION POTENTIAL (V_j)
- DEPLETION CAPACITANCE at zero bias (C_{j0})
- the sum of TRANSIT TIME (τ_d) for holes and electrons
- EMISSION COEFFICIENT (n)
- junction GRADING COEFFICIENT (m)
- the FORWARD COEFFICIENT (FC), which determines when the junction is heavily forward biased

For stability consideration, voltages applied to the diode should not exceed 15 volts.

▸ Further information regarding the FDTD diode formulation may be found in [13].

Figure 7.10 shows editing dialog for a DIODE within the CIRCUIT COMPONENT DEFINITION EDITOR.

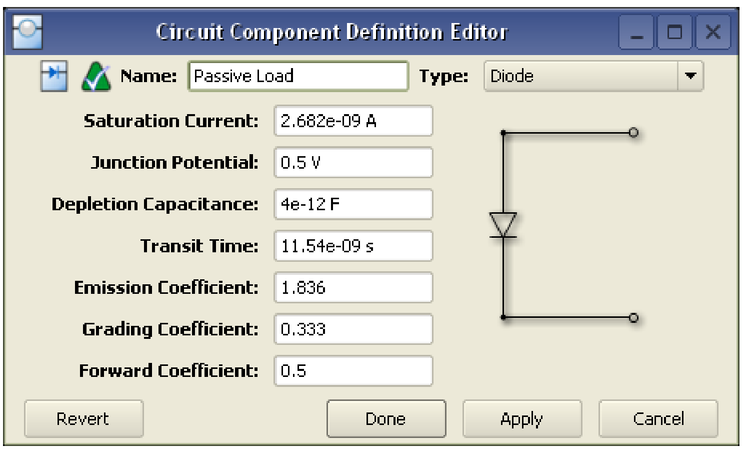


Figure 7.10: Editing a diode within the Circuit Component Definition Editor

7.2.4 Nonlinear Capacitor

The NON-LINEAR CAPACITOR contains parameters which correspond to the following equation:

$$C = C_2 + \frac{C_1 - C_2}{1 + a_1(\frac{|V| - V_s}{V_0})^2 + a_2(\frac{|V| - V_s}{V_0})^4 + a_3(\frac{|V| - V_s}{V_0})^6}$$

where

C is instantaneous cell edge capacitance

V is instantaneous cell edge voltage

C_1 is static (low $|V|$) capacitance

C_2 is infinite $|V|$ capacitance

V_s is voltage magnitude

V_0 is a scaling voltage

a_1 , a_2 and a_3 are coefficients

✓ A non-linear capacitor may be combined with a parallel resistor.

Figure 7.11 shows editing dialog for a Non-linear Capacitor within the Circuit Component Definition Editor.

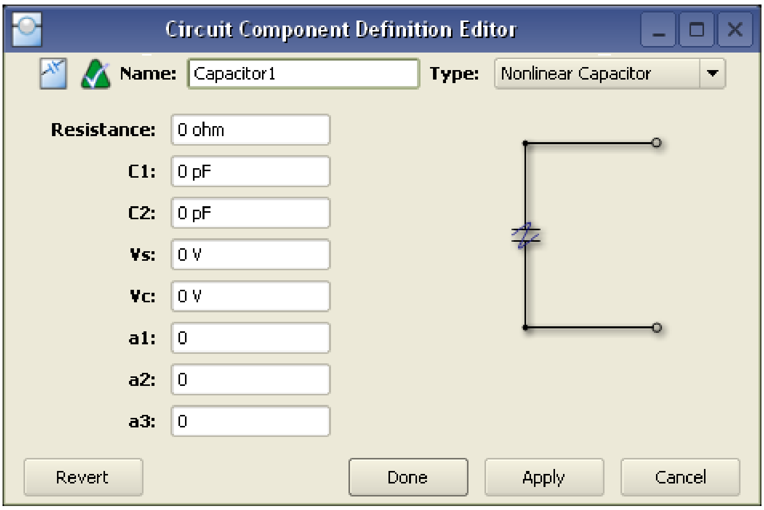


Figure 7.11: Editing a nonlinear capacitor within the Circuit Component Definition Editor

7.2.5 Switch

A special feature of XFdtd is its ability to include timed and programmable Switch components. This allows a change in the configuration of the geometry during a calculation. Any number of switches may be introduced into the geometry, with each switch being specified as a separate component. Switch states may be changed only once during a calculation.

The initial state of the switch, whether open or closed, may be specified as well as the timestep where the switching action begins. The switching action is spread over a variable number of timesteps to reduce switching transients. The switch transition should on the order of 60 timesteps.

To define the properties of the switch, select Open or Closed to define the switch's initial state in the Circuit Component Definition Editor. Press ✚ to add a transition and define the Start Time, Duration and Transition type of the switch by double-clicking on the default values provided in the chart in the editor.

Figure 7.12 shows editing dialog for a multiple transition Switch within the Circuit Component Definition Editor.

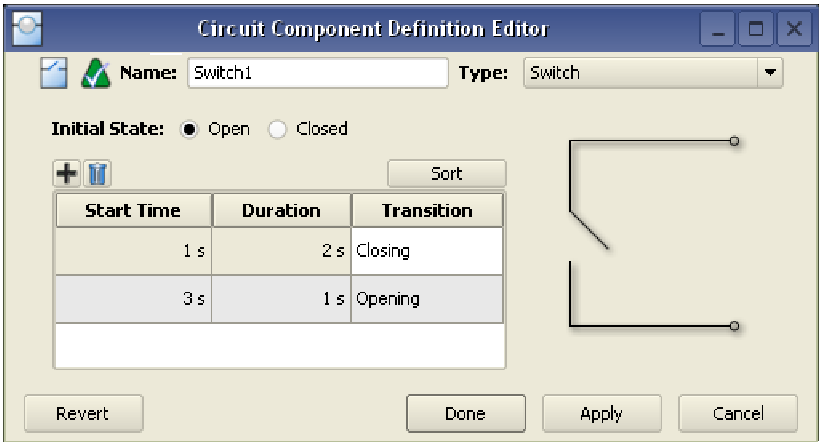


Figure 7.12: Editing a switch within the Circuit Component Definition Editor

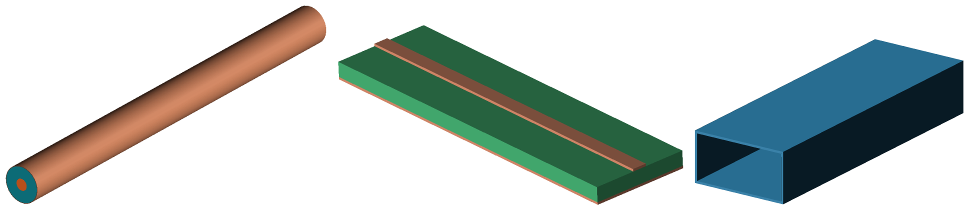
A switch may be programmed to allow multiple open/close transitions during a simulation by adding subsequent entries to this initial definition. A switch transition consists of the timestep at which the switch will activate and the duration (in timesteps) that the Gaussian switching function will be applied. Simply click the ✚ button found above the definitions chart to add one or more switch transitions and define the START TIME and DURATION that the transition will occur. The transition type will be automatically generated based on the start time of each transition. Furthermore, the SORT button will automatically sort the transition from the earliest to latest occurring state.

During the switch transition, the electric field at the switch is changed from the open switch value to zero when closed (or vice versa) following a Gaussian function. This method provides stability for the XFdtd calculation. In order to apply the Gaussian switching function, the value of the electric field at the switch location is first calculated as if the switch were not present. This value is then multiplied by the Gaussian function with the appropriate argument based on the time since the switch state was changed. For a closed switch, the multiplier is zero, for an open switch it is one, and for intermediate times during the transition the value is that for the normalized Gaussian function. For this reason, the function of the switch depends on the material that exists at the mesh location. Usually this is free space, but it might also be dielectric.

The use of timed switches is for situations where only the actual transient results directly calculated by XFdtd are of interest. The introduction of the switching action violates the assumptions of linear system theory, so that applying Fourier transformations to the transient results produced in an XFdtd calculation with one or more switches which change state will not result in valid impedances, S-Parameters, steady-state far-zone fields or other results involving Fourier transformation, even if the transient results finally decay to zero. Broadband far-zone fields will be valid, however.

7.3 Waveguide Tools

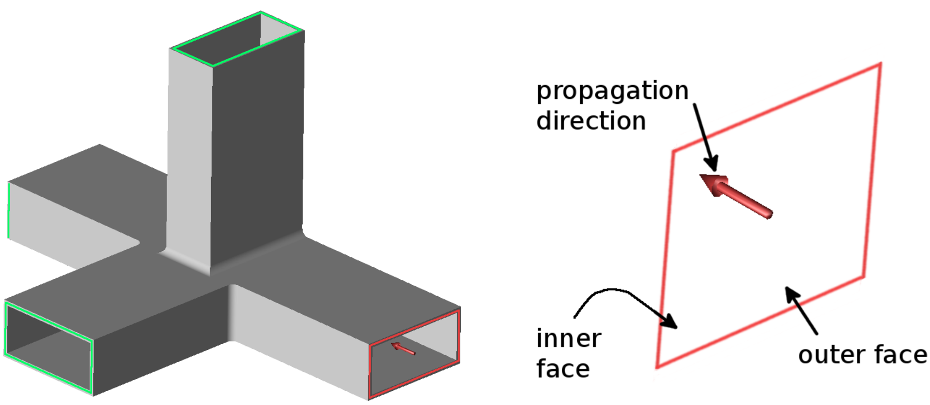
The waveguide tools provide methods for adding a matched termination and modal ports to waveguides. Here we use the broad definition of a waveguide as a structure which guides waves, including the classic conducting pipe ([14, ch. 8], [15, ch. 8] and many others), coplanar waveguide, coaxial and differential transmission line, stripline, and microstrip. Figure 7.13 shows three of many types of waveguides. These structures could be built with the GEOMETRY TOOLS (Section 5) or imported from CAD. They would appear under the Parts branch of the PROJECT TREE.



(a) Coax (b) Microstrip (c) Rectangular Waveguide

Figure 7.13: Some Guided Wave Structures

The WAVEGUIDE EDITOR described in this section defines an interface for a waveguide, usually placed near an end of a waveguide structure to act as a matched termination for the waveguide and to optionally contain one or more ports, which can excite specific modes and act as sensors for those modes. Each port is associated with a chosen mode of the waveguide, but an interface can have more than one port. Modes are found by an Eigensolver by treating the cross section of the waveguide at the interface as if it were part of an infinitely long structure of that cross section. Waveguides may be excited by signals from discrete voltage or current sources which are described in Component Tools (7.1), or at their boundaries with specified modes using the waveguide ports described in the Waveguide Port Specification Tab (7.3.4) in this section. See section (7.4) for information on which type of excitation port to use in a particular situation. Note that waveguide ports are modal, i.e. specific to a specified mode and discrete components are nodal, i.e. specific to the location regardless of what mode(s) are incident on them, and that discrete ports and waveguide ports cannot be used together in the computation of S-Parameters.



(a) Magic Tee with Interfaces (b) Waveguide Interface

Figure 7.14: Magic Tee and Waveguide Interface

As an example of how waveguide interfaces might be used, Figure 7.14a shows a magic tee with waveguide interfaces, shown in red or green, at all four terminals. As shown in Figure 7.14b the interface is a rectangular region with an inner and outer face. The inner face should face the waveguide structure. Energy incident on the inner face will be absorbed, so the interface will act as a matched termination to the waveguide. Also, if this interface contains one or more ports which are used to excite chosen modes of the waveguide, the excitation will proceed in the propagation direction, which is the direction into the waveguide, as shown by the arrow. The outer face appears as a perfect conductor to the rest of the problem space, so energy external to the waveguide will not enter the waveguide through the interface.

To access the WAVEGUIDE EDITOR, double-click on an existing WAVEGUIDE object in the WAVEGUIDES branch of the PROJECT TREE, or right-click on the branch to create a NEW WAVEGUIDE INTERFACE. The editor will open at the top of the GEOMETRY WORKSPACE WINDOW, and consists of a PROPERTIES tab, a GEOMETRY tab, a BOUNDARIES tab, and a PORT SPECIFICATION tab.

7.3.1 Waveguide Properties Tab

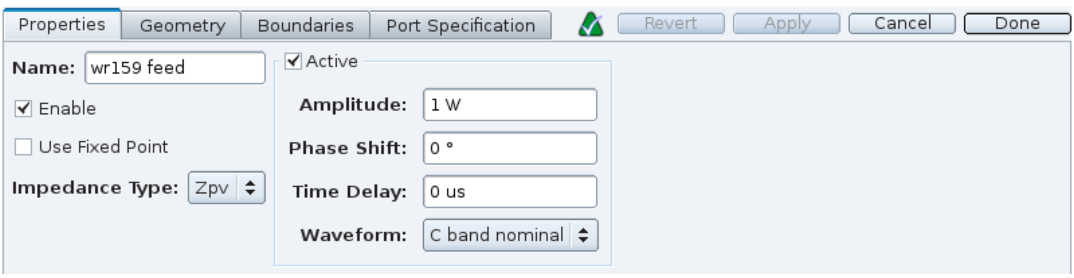


Figure 7.15: The Waveguide Tools Dialog with Properties Tab Selected

This tab allows some basic properties about the waveguide interface to be defined.

Name User-defined name.

Enable If not enabled, this interface will not be part of computations and will not act as a termination or port. Unchecking **Enable** is a convenient way to effectively remove an interface from a project without deleting it, so that it is available to be used later.

Use Fixed Point Add a fixed point at the interface in the direction of propagation (normal to interface).

Impedance Type Type of impedance computation, Zvi, Zpv, or Zpi. The impedance of modes in a waveguide can depend on how the impedance is defined. Three methods are available in XFdtd: Zvi is a voltage and current definition, Zpv a power and voltage definition, and Zpi a power and current definition. Definitions which use voltage require the specification of an impedance line. These methods, and the use of an impedance line, are explained further in the Port Specification Tab subsection (7.3.4). The default choice should be good for most basic situations. After the simulation is complete, this impedance will be available in the results browser entries for this waveguide port.

Active Allows ports (see Subsection 7.3.4) to be active. Uncheck this to use this waveguide interface as a termination only.

Amplitude Peak amplitude of the associated waveform

Phase Shift Phase shift to be applied to the associated waveform, if the waveform is sinusoidal

Time Delay Time delay to be applied to the associated waveform.

Waveform Associated waveform definition, selects the waveform to be used if a port is defined that excites the waveguide.

7.3.2 Waveguide Geometry Tab

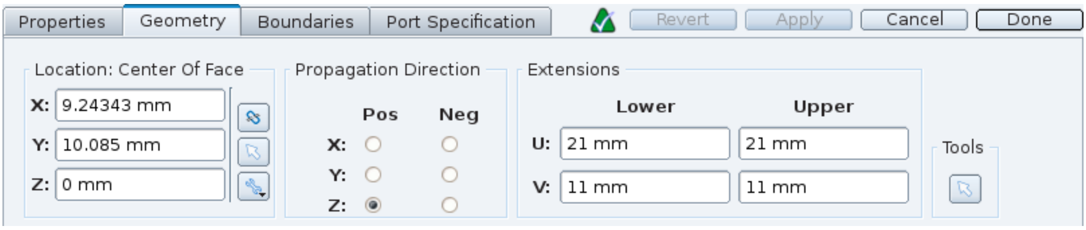


Figure 7.16: The Waveguide Geometry Tab

Use this tab to position and size the interface rectangle within the problem space. The waveguide interface is a planar rectangle transverse to the propagation directions of the waveguide, as shown in Figure 7.14b. The rectangle is defined by choosing its center, orientation (via the propagation direction), and extent. The buttons to the right of the center point are similar to some of those of the Geometry Orientation Tab (5.5) and provide for mouse selection of the center point, and attachment and offset of the interface to an object in the geometry, such as the waveguide structure. Here attachment means that the interface will change location to follow the part it is attached to, if that part is moved to a different location in the geometry. Note that the center point does not have to be the actual center of the waveguide cross section — center point and extensions are just used to locate the interface rectangle.

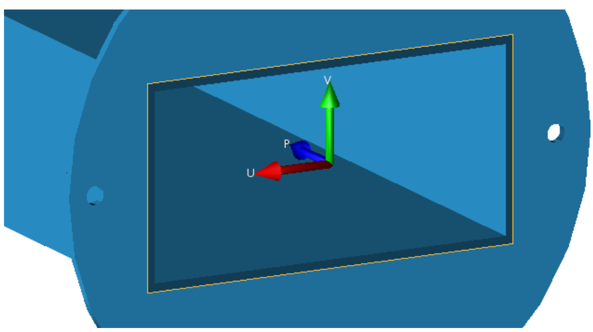
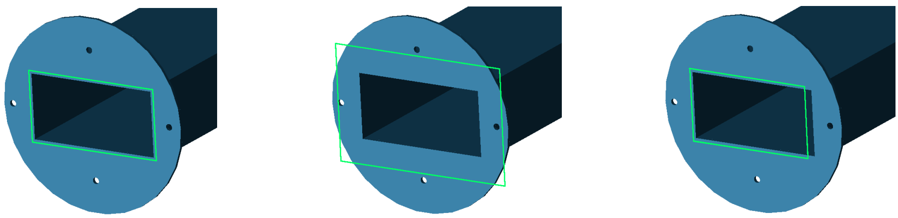


Figure 7.17: Interface Rectangle with Local (u, v) Coordinate System

The propagation direction defines the orientation of the interface rectangle, either normal to x, y, or z. It also determines which of the two faces of the interface rectangle is the inner and which is the outer, the inner being in the propagation direction. Select the direction that points into the waveguide structure. Of course, the energy inside the waveguide may propagate in either direction. If this interface has one or more ports which are used to excite the waveguide, the excitation will proceed in the specified propagation direction. Energy incident on the inner face of the interface will be absorbed. Energy incident on the outer face of the interface will encounter a perfect conductor. Setting the propagation direction also causes a local (u, v) coordinate system to be assigned to the interface, parallel to the edges of the interface and normal to the direction of propagation, P, as shown in Figure 7.17.

The rectangle's center point and extent should be chosen to position the rectangle so that it encloses most of the energy in the waveguide. For waveguides which have conducting boundaries on all transverse sides, such as rectangular, circular, and coaxial, the interface size is fairly simple: it should enclose the waveguide fields completely. Since the interface rectangle will appear as a conductor on its outer face and an absorber on its inner face, it is a good idea to not make it much larger than it needs to be to enclose the energy of the guide. Figure 7.18 shows how this might look for a rectangular waveguide.



(a) Correct (b) Correct, but overly large (c) Incorrect

Figure 7.18: Geometry of Interface for Rectangular Waveguide

For waveguides that have one or more open sides, the required size of the interface is not as obvious, though it should be large enough to enclose the largest fields so that it may properly terminate the waveguide and so that modes may be properly computed. Figure 7.19 shows the E-field typical for the quasi-TEM mode around a microstrip. The field tends to be strongest between the microstrip and the ground plane, and near the corners of the microstrip. The waveguide interface should be chosen to encompass the area of the strongest fields.

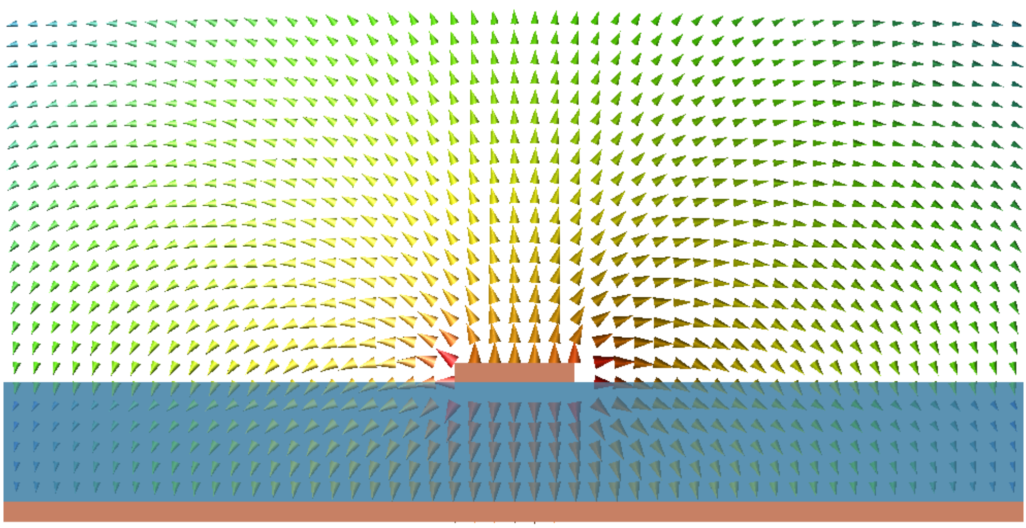


Figure 7.19: Quasi-TEM Mode E-Field around a Microstrip

The four outer edges of the interface are treated as perfect conductor, causing the tangential fields to approach zero near the outer edges in the field solution, as explained further in the description of the Boundaries Tab (7.3.3). If the interface is too small it may not be possible to compute the desired mode with good accuracy. If the interface is too large, additional nonphysical modes may be produced. For example, consider a microstrip with a conducting boundary around it. The interface and microstrip form a rectangular guide with an offset center conductor. This structure will have additional modes not present in a simple microstrip. The cutoff frequency for some of these modes will decrease as the size of the interface increases. For a small interface, the unwanted modes may have a cutoff frequency above the frequency range of interest and they will not be found, but a larger interface may have unwanted modes with cutoff frequencies in or below the frequency range of interest. Finding the best size is a balancing act:

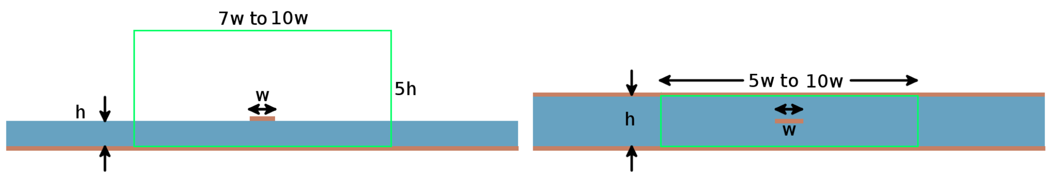
Interface too small May not have "enough space" to set up the desired mode properly,

Interface too large May have unwanted modes, may interfere with the surrounding geometry.

Figures 7.20 – 7.21 give recommended minimum sizes for the interface rectangle for some common structures based on empirical results. For other open-ended structures, keeping the interface sides 5 x (maximum structure dimension) from the structure is a fairly conservative first choice. One can verify that an interface is large enough by viewing the fields for the chosen mode and seeing that the fields do indeed drop off significantly by the time they reach the outer edges of the interface.

Figure 7.20a shows the interface for a microstrip with $w \approx h$. As w becomes small compared with h, the most intense fields tend to stay near the vertical space above and below the trace. If $w > h$, the horizontal dimension of the interface may need to be increased to around $10w$.

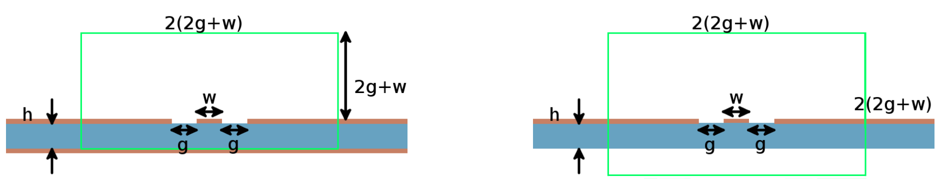
Figure 7.20b shows the interface for a stripline. The lower value of the interface width is for $w \leq h$. The height of the interface should be set so that the horizontal sides of the interface are in the metal ground plane.



(a) Microstrip (b) Stripline

Figure 7.20: Interface Size Guidelines for Microstrip and Stripline

Figures 7.21a and 7.21b show the interface for grounded and ungrounded co-planar waveguides. For the grounded case the lower horizontal edge of the interface is in the ground plane. For the ungrounded case, the interface extends the same distance below and above the plane of the conductors.



(a) Co-planar Waveguide with Ground (b) Co-planar Waveguide

Figure 7.21: Interface Size Guidelines for Co-planar Waveguide

7.3.3 Waveguide Boundaries Tab

The outer four edges of the interface are treated as a perfect conductor by the Eigensolver. The user can select whether each boundary is perfect electrical conductor (PEC) or perfect magnetic conductor (PMC).

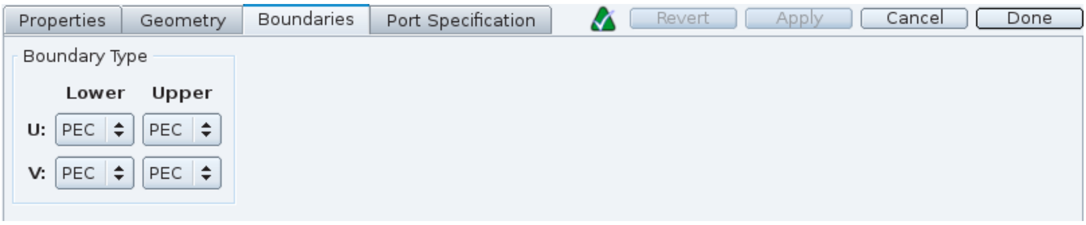


Figure 7.22: The Waveguide Boundaries Tab

Waveguides with conducting material on all transverse sides, such as classic rectangular and circular guides, will have near-zero fields at their edges so the boundary conditions from this tab will have minimal effect on the problem. However, in some open-sided waveguides the choice of boundary type may allow a more compact interface rectangle to be used.

For example, Figure 7.23 shows the E-field typical for the quasi-TEM mode around a microstrip with a possible location of the interface rectangle drawn in green. In this case the horizontal sides of the interface are u-directed and the vertical sides are v-directed. We wish the fields to be very close to zero along the edges of the interface, but due to the open-sides of the microstrip this is possible for only the lower horizontal edge, which is in the ground plane. The other edges are located away from the microstrip so that the fields there are small, but size constraints often means that we do not want the interface rectangle to be too large, so we must strike a balance between small size and performance, as discussed in the previous section (7.3.2). In the figure, the E-field along the vertical sides is strongest just above the air-dielectric boundary. At this point E is mostly tangential to vertical sides and H (not shown) is mostly normal to them. The PMC boundary type best satisfies this condition, so the vertical (v) sides should be set to PMC. The strongest E along the top horizontal boundary is directly over the microstrip and is mostly normal to the top boundary at this point (H tangential). The PEC boundary type best satisfies this for the horizontal axis, so the upper u boundary should be set to PEC. The lower horizontal boundary is in the conducting ground plane and will have near-zero fields so that boundary type is of little importance. For this case, we recommend leaving that boundary set to PEC.

Here are some recommended choices for the other structures of Figures 7.20 – 7.21. Stripline: vertical PMC, horizontal PEC (in conductor). Co-planer with and without ground: vertical PEC, horizontal PEC. For other structures, if you are not sure of the field configuration at the interface edges an Eigen solution can be performed (see the next section, 7.3.4). on a large interface and the fields viewed from that solution. In general, the larger the interface, the smaller the fields at the edge of the interface and the less impact the boundary type choice will have.

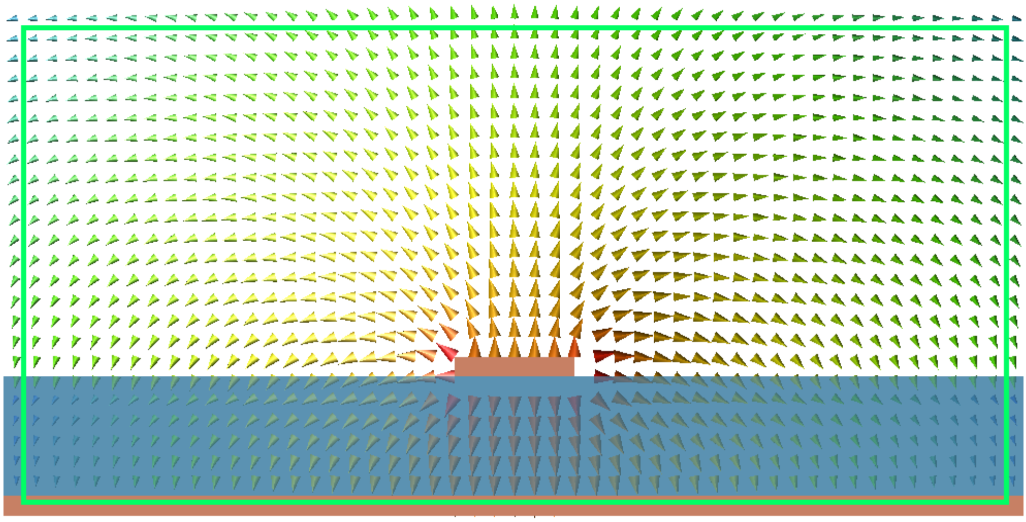


Figure 7.23: Quasi-TEM Mode E-Field around a Microstrip with Proposed Waveguide Interface

7.3.4 Waveguide Port Specification Tab

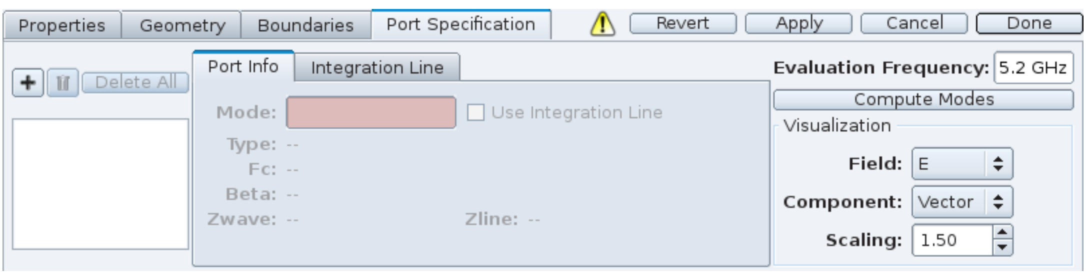
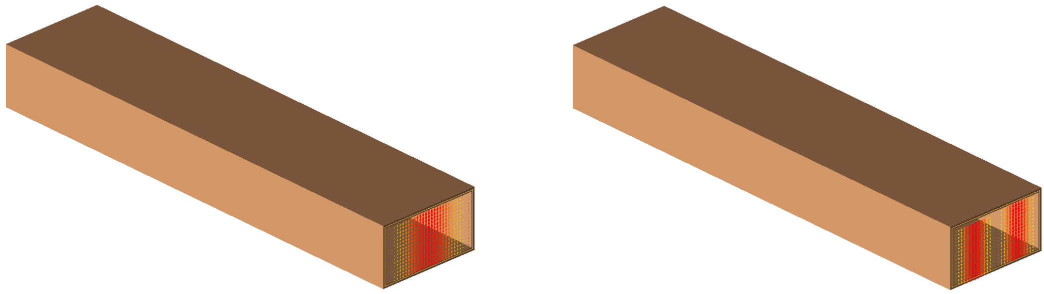


Figure 7.24: The Waveguide Port Specification Tab without a Defined Port

This tab allows the user to create ports corresponding to particular modes for the waveguide. Unlike circuit components, which may have only one port associated with them, a waveguide may have multiple ports. Each port is associated with a particular waveguide mode and will be part of the S-Parameter matrix, if S-Parameters are being computed. When a port is active, it will excite its corresponding mode in the waveguide.

If the waveguide interface has no ports, it will still act as a matched termination for energy flowing out of the waveguide structure.

When ports are present, an Eigensolver finds each requested mode by assuming that the waveguide cross section is part of an infinitely long structure of that cross section. XAct materials are not allowed within the interface rectangle, though they may appear in other parts of the waveguide. Also, the Eigensolver assumes the objects are lossless and does not account for dispersive, nonlinear, or magnetic materials. Once the modes are found, the mode distribution of the active port is applied to the interface to excite the problem.



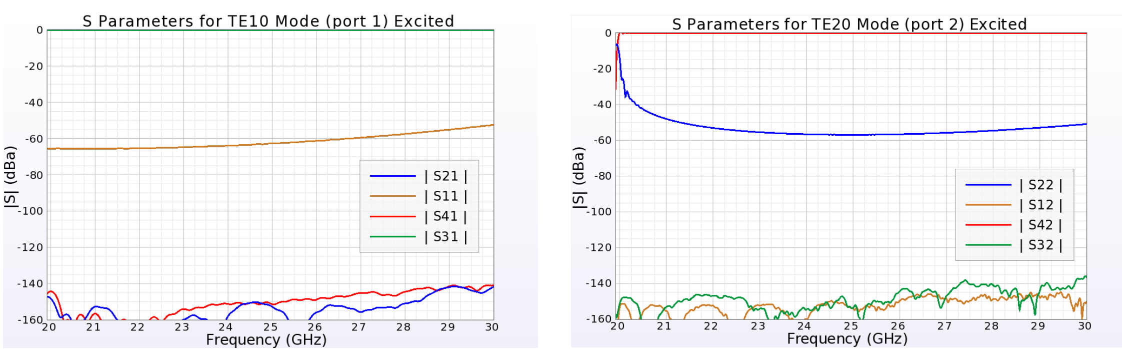
(a) TE_{10} Mode (b) TE_{20} Mode

Figure 7.25: Short Straight Rectangular Waveguide

Each port launches and acts as a sensor for a particular mode, selected by the user. Consider the simple example of a straight rectangular waveguide of Figure 7.25 with waveguide interfaces at each end. Each interface has two ports defined. One port at each interface excites/senses the TE_{10} mode, as shown in Figure 7.25a and the other the TE_{20} mode, as shown in Figure 7.25b. For convenience, call one end of the waveguide the “source” and the the other end the “load”. The port numbers are assigned as follows, where f_c is the cutoff frequency of the mode:

port	mode	f_c	location
1	TE_{10}	10 GHz	source end
2	TE_{20}	19.95 GHz	source end
3	TE_{10}	10 GHz	load end
4	TE_{20}	19.95 GHz	load end

Figure 7.26a shows S-Parameters for the case of the TE_{10} mode active at the source end (port 1). Most of the energy arrives still in the TE_{10} mode at the other end of the guide (S31) and a small amount is reflected back to the excitation port (S11). Almost no energy is detected at either of the TE_{20} ports (S21 and S41) because this uniform structure contains no discontinuities which might transfer some of the energy from one mode to other modes. Figure 7.26b shows the corresponding case for the TE_{20} mode (port 2) active — and in this case almost all of the energy stays in the TE_{20} mode.



(a) Port 1 (TE_{10}) Active (b) Port 2 (TE_{20}) Active

Figure 7.26: S-Parameters for Rectangular Waveguide

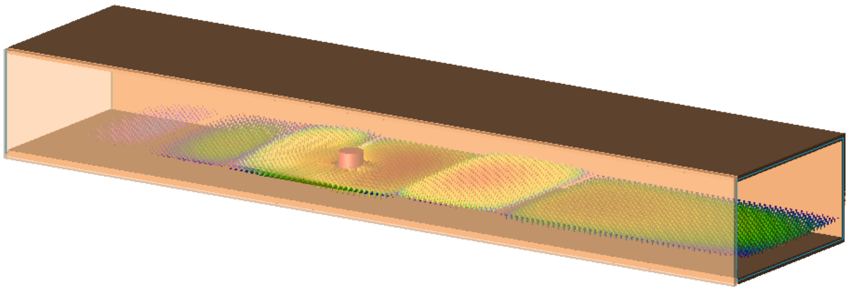
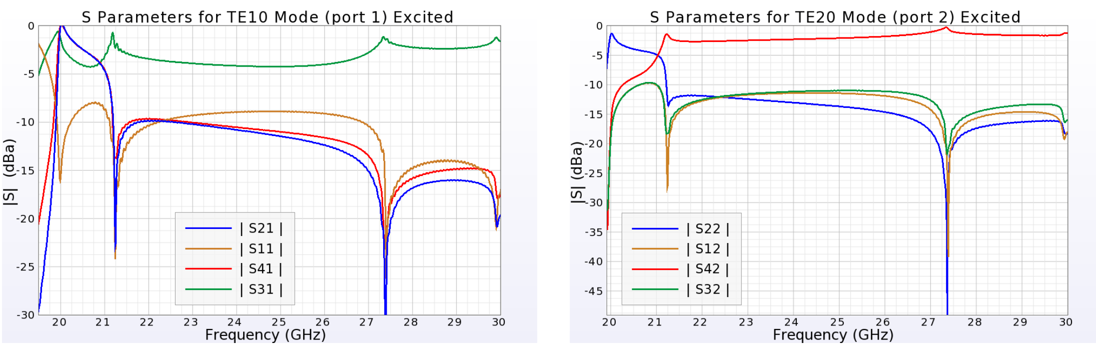


Figure 7.27: Time-domain E Field in Rectangular Waveguide with Off-center Metal Post



(a) Port 1 (TE_{10}) Active

(b) Port 2 (TE_{20}) Active

Figure 7.28: S-Parameters for Rectangular Waveguide with Post

Next the problem is modified slightly by adding a metal post off-center in the waveguide, as shown in Figure 7.27. This discontinuity will cause some of the energy from the excited mode to couple into other modes as required to satisfy boundary conditions. The simulation is re-run for the new geometry, giving the S-Parameter results of Figure 7.28. In Figure 7.28a, the excitation is by port 1 (TE_{10} mode) and in Figure 7.28b, the excitation is by port 2 (TE_{20} mode). Notice from the S-Parameter plots that for either case there is now significant energy in both modes.

7.3.4.1 Simple procedure for lowest order mode with no integration line

To add a port to a waveguide interface for the special, but common, case of a single port which uses the lowest order mode of the waveguide with no rotational symmetry and no need for an integration line the following four steps from the Port Specification tab will suffice:

1. Click ✚ to add a port to the interface. Set its name, if desired.
2. Enter the Evaluation Frequency, a frequency within the expected operating range of the waveguide.
3. Select “Compute Modes” to compute the modes for that frequency with the Eigensolver. If the Evaluation Frequency is above the lowest waveguide mode cutoff frequency, results for mode 0 should be displayed in the geometry window for the waveguide.

4. Click Apply or Done to accept this port.

7.3.4.2 More detailed explanation

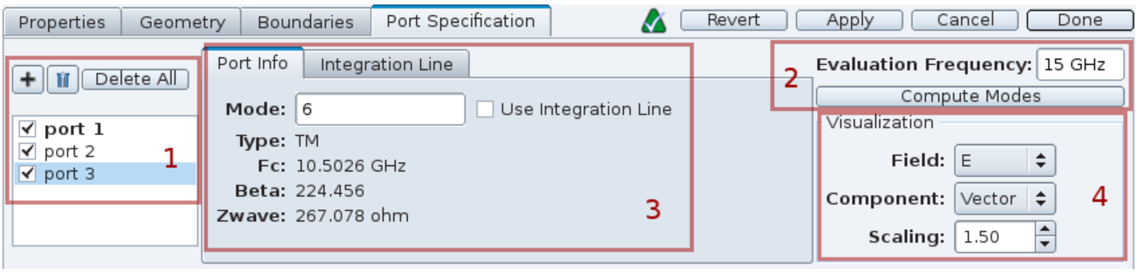


Figure 7.29: The Wavegude Port Specification Tab with Some Regions Marked

Figure 7.29 shows the Port Specification tab divided into 4 regions.

Region 1 The controls in region 1 allow ports to be added, named, deleted, and selected in various ways. A port that is checked will be part of the simulation and may be set as the active port when a simulation is created. If it is not checked, it will be ignored by the simulation. One port will be shown in bold face, “port 1” in this case, indicating that this port is the default and will be active if S-Parameters are not computed. Right-click on a port and select Make Default to make this port the default port. A port that is *selected*, such as “port 3” in this case (highlighted in blue), is the one for which modes will be displayed in region 3 once the Compute Modes button in region 2 is pressed. Click on a port to select it.

Region 2 The Evaluation Frequency is entered and mode computation performed from region 2. The modes will be evaluated for all ports in this interface at the Evaluation Frequency by the Eigensolver. If the Evaluation Frequency is below the cutoff frequency of a mode, that mode will not be displayed. The mode will be configured for the distribution at the Evaluation Frequency. For some modes, such as modes in heterogeneous waveguides, the distribution changes slightly with frequency. Best results will be obtained at the Evaluation Frequency.

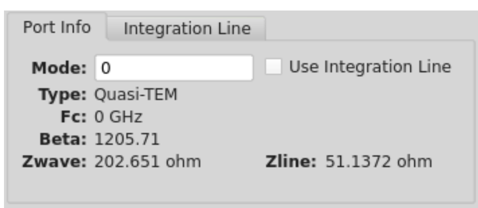


Figure 7.30: Port Information Area (Region 3) for a Transmission Line, including Zline

Region 3 This region contains mode information about the selected port, once computed, and allows specification of an integration line, explained below. The modes are ranked by increasing integers starting at 0 for the lowest order mode. The waveguide can be any arbitrary 2-D shape, not just rectangular or circular. In general, the convenience of indexing modes by (x, y) or (r, θ) coordinates, such as TE_{10} or TM_{21}, does not apply to arbitrary shapes, hence the ranking by sequential nonnegative integers. For cases such as rectangular or circular, it is straightforward to identify the indexed mode from the cutoff frequency and from the field pattern in region 4. The following information is displayed for the selected mode:

Type The "Type" can be TE for transverse electric, TM for transverse magnetic, TEM for transverse electric and magnetic, quasi-TEM for quasi transverse electric and magnetic (such as a microstrip), or EH for mixed modes with both TE and TM components.

Fc The cutoff frequency (f_c), or lowest propagating frequency, of this mode.

Beta Propagation constant (β) of this mode in the waveguide.

Zwave Wave impedance is the ratio of the transverse components of the electric and magnetic fields in the waveguide. It is computed by $\sum_{cells} \frac{|E_t(f)|}{|H_t(f)|}$ in the plane of the port, where $E_t(f)$ and $H_t(f)$ are the transverse components of E and H, and f is the evaluation frequency.

Zline Line impedance is computed when a transmission line is detected by XFdtd, which means that two or more unattached conductors are found in the port. Examples include microstrip, stripline, coax, and differential pairs. Figure 7.30 shows Region 3 for the case of a microstrip, showing Z_{line}. It is computed from $Z_{line} = \frac{P_t(f)}{I^2(f)}$, where P_t is the frequency-domain forward power and $I(f)$ is the frequency-domain forward current for the port.

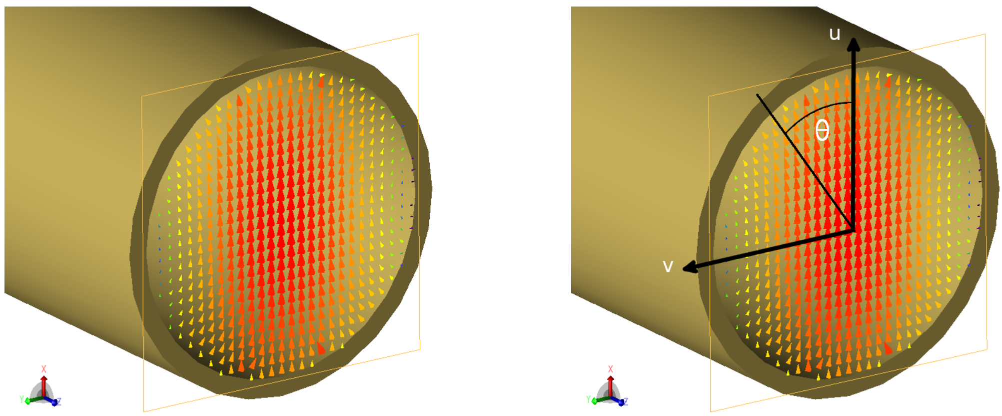
The **Integration Line** is a line along which voltage will be computed if the impedance type from the Properties tab (7.3.1) is set to "Zvi" or "Zpv". The Integration Line is also used to orient some modes when the waveguide cross section has a rotational symmetry, such as a square or circular waveguide. To use an integration line, click to check the "Use Integration Line" check box which will enable the Integration Line tab, where the line can be defined by choosing its endpoints, p_1 and p_2. These are chosen in a way that is very similar to picking the endpoints for a discrete component, as covered in subsection (7.1.1), "Adding a New Component". As with a discrete component, the direction of the line is important. The positive zero-phase direction will be set so that E points from endpoint 2 (p_2) to endpoint 1 (p_1).

Integration line for voltage computation In circuit theory, impedance may be calculated in several ways, three of which are:

$$Z_{pv}(f) = \frac{V(f)V^*(f)}{2P(f)},$$
$$Z_{vi}(f) = \frac{V(f)}{I(f)},$$
$$Z_{pi}(f) = \frac{2P(f)}{I(f)I^*(f)},$$

where $P(f)$, $V(f)$, and $I(f)$ are complex frequency-domain power, voltage, and current, and A^* indicates the complex conjugate of A. In circuit theory $Z_{pv} = Z_{vi} = Z_{pi}$. To extend these definitions to waveguides, let $P(f) \rightarrow P_t(f)$ be the power flow through the transverse cross section of the guide and $I(f)$ the total current flow. $P_t(f)$ and $I(f)$ are thus well defined. The usual way to obtain voltage, $V(f)$, is to integrate the E field along an integration line in the transverse plane of the waveguide between terminals, sides, or nodes of the guide. The Integration Line in region 3 defines this path for the computation of $V(f)$, and must be defined to compute Z_{pv} or Z_{vi}. Note that $V(f)$ is path-dependant and, in general, $Z_{pv} \neq Z_{vi} \neq Z_{pi}$ in waveguide theory. The path is usually chosen to be the straight line path that gives maximum voltage, such as between the centers of the long sides of the guide for the TE_{10} mode in a rectangular guide.

A good, and more thorough, discussion of modal impedance in waveguides can be found in [16] and [17], in addition to the many books on waveguide theory.

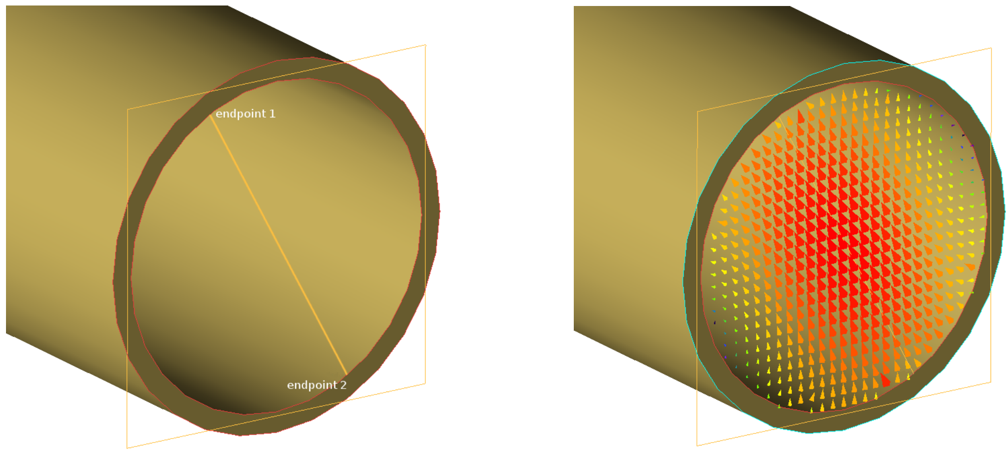


(a) Default mode orientation (b) Desired mode rotation angle θ

Figure 7.31: TE_{11} mode E field in a waveguide port of a circular waveguide

Integration line for mode orientation In certain cases where the waveguide cross section contains a rotational symmetry, such as in a square or circular waveguide, it may be possible for a mode to exist in more than one orientation. For example, Figure 7.31a shows the TE_{11} mode in a circular waveguide. Define desired orientation angle, θ, for the mode relative to the positive direction of the u axis, as shown in Figure 7.31b. θ can vary from 0 to 2π (or really, any $\theta \in \mathbb{R}$) without changing mode type or β, f_c, or Z_{wave} for the mode. This ambiguity is a type of mode degeneracy. In a circular waveguide, the Eigensolver may produce two solutions for the mode. XFdtd will attempt to orient the mode so that the strongest E field is parallel to the u-axis. If another orientation is desired, the impedance line can be used to orient the mode within the waveguide. In Figure 7.32 an integration line is used to orient the mode at an angle of $30°$ from the u-axis. The direction of the line, i.e. which end is p_1 and which is p_2, determines the positive direction of the fields, so it is important to define impedance lines consistently for like modes in each interface if S-Parameters are being computed.

Region 4 When modes are successfully computed, they are shown in the Geometry View located at the waveguide. In region 4, the field "E" or "H" may be selected as well as the x, y, or z component of the selected field or the full vector (all 3 components). "Scaling" controls the size of the vectors shown in the display.

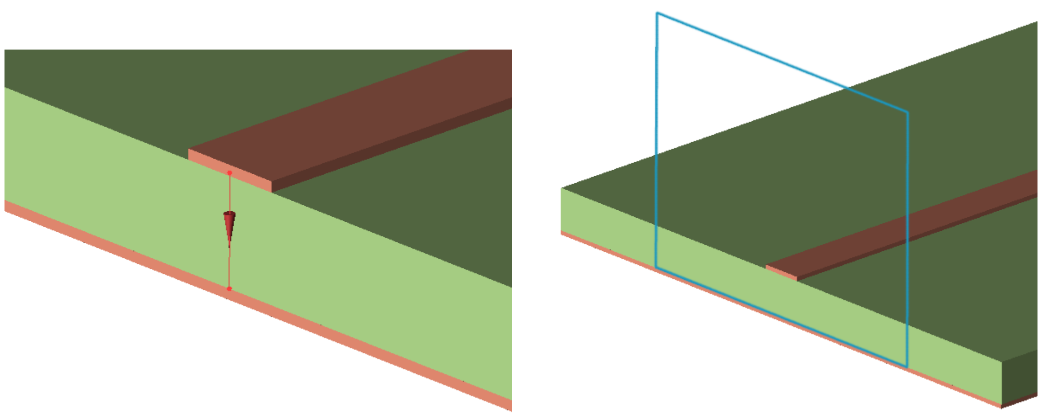


(a) Location of the integration line (b) E field of oriented TE$_{11}$ mode.

Figure 7.32: Integration line used to orient the TE$_{11}$ mode of a circular waveguide

7.4 Discrete Ports or Waveguide Ports?

Discrete ports and waveguide ports each provide a means to excite a geometry at a specific location in the problem space and to sense signals at their locations, providing port and feed related information such as S Parameters, input impedance, group delay, powers, efficiencies, etc. Discrete components, which are explained in the Components Tools section (7.1), use a voltage or current source to excite a cell edge that is usually connected between parts of the geometry, such as between a microstrip and ground plane, or two terminals of an antenna. Waveguide ports, which are explained in the Waveguide Tools section (7.3), excite a particular mode in a guided wave structure. General information about ports is provided in the Port Sensors section (10.5). Often both methods will give good results.



(a) Discrete Port (b) Waveguide Port

Figure 7.33: A Discrete Port and a Waveguide Port Feeds with a Microstrip

Each discrete port will consist of a voltage or current source, possibly combined with one or more other discrete components such as resistors, capacitors, and inductors, located at a single point in the problem space, and often connected between conductors. Figure 7.33a shows how a discrete port might be used to feed a microstrip line. In many cases one needs to excite a structure where it is neither necessary nor desirable to include a guided wave feed. Examples include exciting an antenna or filter directly at it's input, as shown in Figure 7.34 for a bowtie antenna. In this case a discrete port is clearly the most straightforward choice.

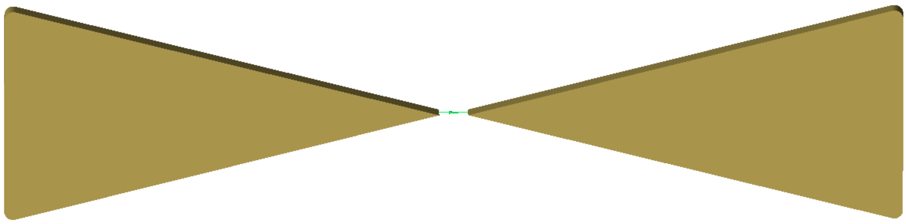


Figure 7.34: A bowtie antenna fed directly by a resistive voltage source

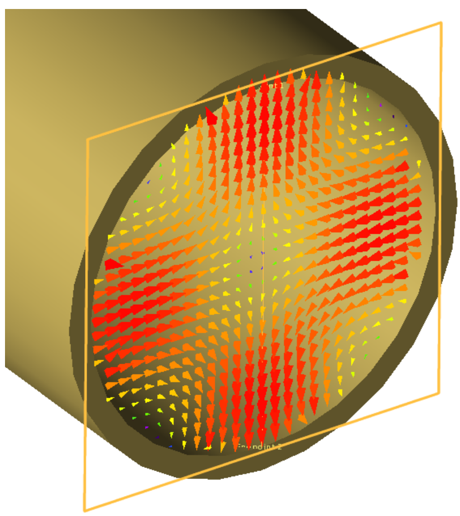


Figure 7.35: TE_{21} mode E field in a waveguide port of a circular waveguide

A waveguide port consists of a rectangular region, usually at or near an end of a guided wave structure, surrounding the cross section of the waveguide. Figure 7.33b shows the same microstrip with a waveguide port instead of a discrete feed. Waveguide ports allow the user to select a particular mode, say the fundamental quasi-TEM mode in the case of a microstrip, and excite that mode specifically (see the discussion of ports in the Waveguide Port Specification Tab subsection, 7.3.4). Fields in the rectangular port region are set to correspond to the selected mode. For example, Figure 7.35 shows a waveguide port exciting the TE_{21} mode in a circular waveguide. Waveguide ports give more control over the mode excited (and detected). One could excite a given mode in a waveguide with discrete components via voltage stubs or current loops, but a waveguide port is usually simpler. Also, a waveguide port will distribute the fields specifically for the selected mode so energy at other modes will, in general, be much less than for the stub or current loop method. A waveguide port is usually the best choice for exciting a particular mode when more than one mode may propagate.

For cases where only one mode will propagate and both methods excite that mode fairly efficiently, the best method may depend on the type of problem. Returning to the microstrip of Figure 7.33, both methods will excite the microstrip well. The discrete feed is obviously more compact. If the geometry is crowded, such as inside a mobile phone, the discrete feed may make more sense. To see when the waveguide feed may be necessary we need to look at the performance of the two feed types. The quasi-TEM mode of operation typical for a microstrip has field magnitudes in a cross section as shown in Figure 7.23. A waveguide port excites this pattern closely over the area covered by the port. A discrete feed simply applies a voltage difference between the microstrip and its ground plane. Since, usually, the quasi-TEM is the only propagating mode, the correct field distribution for the mode is achieved a very short distance along the microstrip from the feed. This small discrepancy causes a slight reflection from the discrete port that may be 30 to 40 dB below the input signal. A waveguide port at its evaluation frequency (see the

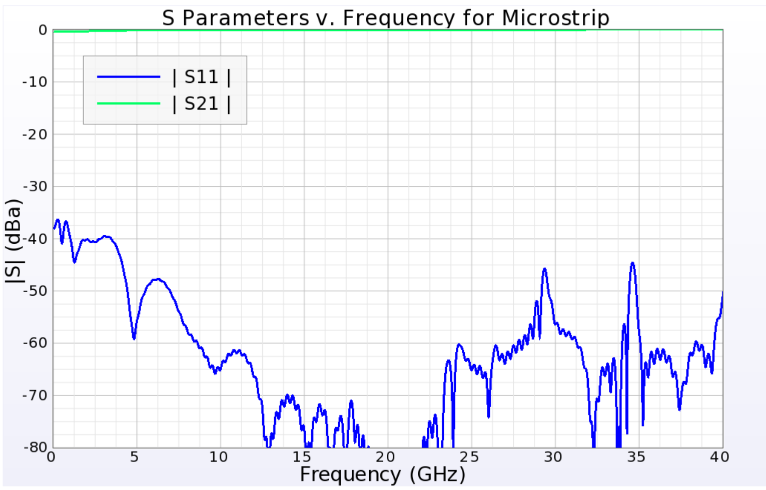


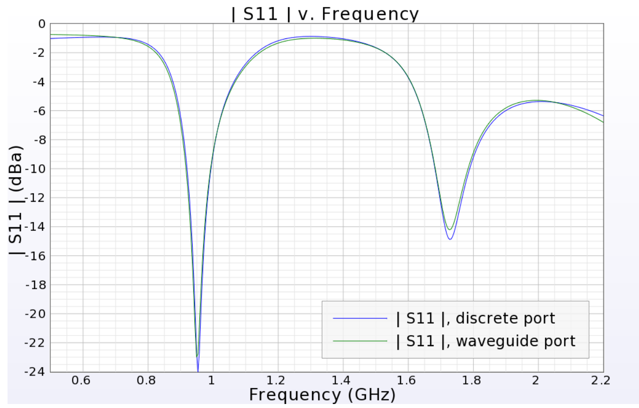
Figure 7.36: S Parameters for a Microstrip with a Waveguide Port at Each End

waveguide port discussion in subsection 7.3.4) usually has less reflection and can have a return loss "floor" of -60 to -70 dB. Figure 7.36 shows the return loss for a 50Ω microstrip with a waveguide port exciting one end and another waveguide port acting as a matched termination at the other end. The evaluation frequency for both ports is 20 GHz.

In both cases the tiny loss of forward-going energy has very little impact on the energy received at other ports from the excitation port, fields in the problem space other than those reflected to the excitation port, efficiency, power dissipated and radiated, Specific Absorption Rate (SAR), far-zone pattern, etc. It may be important, though, if very low return loss is anticipated from the excitation port, for example, measuring S11 from a coaxial connector. For many applications, such as antenna and PCB applications, both methods will give comparable results. For example, Figure 7.37 shows S11 for a three-band antenna similar to [18] next to the Specific Anthropomorphic Mannequin (SAM) head simulated once with a discrete port and once with a waveguide port. The S11 computed by each method are quite similar. Figures 7.38 and 7.39 show SAR statistics for the SAM head with 1 Watt delivered power computed by both methods and Figures 7.40 and 7.41 compare the total far-zone radiation pattern for both methods. Again, the results are very close.

Following are some general guidelines for each type of port.

(a) Geometry



(b) S11 for Discrete Port and Waveguide Port

Figure 7.37: A Triple-Band Slot Antenna Near the SAM Head

Quantity	0.82 GHz	1.9 GHz
SAR Sensor (Raw)		
Maximum Value	8.155 W/kg	15.771 W/kg
Location of Maximum	(-11.937 mm, 0.7993 mm, 70.75 mm)	(-9.945 mm, 15.258 mm, 19.25 mm)
Average Value	0.0536 W/kg	0.0905 W/kg
Total Power Dissipated	0.2834 W	0.4785 W
Total Tissue Mass	5.288 kg	5.288 kg
SAR Averaging Sensor (10g Average)		
Maximum Value	3.321 W/kg	6.144 W/kg
Location of Maximum	(-11.937 mm, 0.7993 mm, 55.75 mm)	(-9.945 mm, 12.266 mm, 25.25 mm)
Average Raw SAR Value	0.3185 W/kg	0.5428 W/kg
Total Power Dissipated	0.2583 W	0.4354 W
Total Tissue Mass	0.811 kg	0.811 kg
SAR Averaging Sensor (1g Average)		
Maximum Value	5.199 W/kg	9.7 W/kg
Location of Maximum	(-11.937 mm, 0.7993 mm, 53.75 mm)	(-9.945 mm, 15.756 mm, 24.75 mm)
Average Raw SAR Value	0.3185 W/kg	0.5428 W/kg
Total Power Dissipated	0.2583 W	0.4354 W
Total Tissue Mass	0.811 kg	0.811 kg

Figure 7.38: SAR Statistics for Triple-Band Slot Antenna Near the SAM Head with a Waveguide Port

Discrete ports recommended when:

- Using a microstrip or stripline with no need to measure return loss below -40dB,
- Geometry is crowded near the desired feed point, such as in a mobile phone or portable computer,
- Exciting a structure directly without need to model the feed line.

Waveguide ports recommended when:

- Exciting a mode other than the lowest order propagating mode,
- Exciting hollow pipe type waveguides, such as rectangular or circular (with or without ridges, partial dielectric fill, etc.),
- Launching a signal in a coaxial cable,
- Need to measure return loss below -40dB.

Quantity	0.82 GHz	1.9 GHz
SAR Sensor (Raw)		
Maximum Value	7.971 W/kg	15.245 W/kg
Location of Maximum	(-11.937 mm, 0.7993 mm, 70.75 mm)	(-9.945 mm, 15.258 mm, 19.25 mm)
Average Value	0.05235 W/kg	0.09162 W/kg
Total Power Dissipated	0.2768 W	0.4844 W
Total Tissue Mass	5.288 kg	5.288 kg
SAR Averaging Sensor (10g Average)		
Maximum Value	3.218 W/kg	6.516 W/kg
Location of Maximum	(-11.937 mm, 0.7993 mm, 56.25 mm)	(-9.945 mm, 12.266 mm, 25.75 mm)
Average Raw SAR Value	0.3097 W/kg	0.5422 W/kg
Total Power Dissipated	0.2525 W	0.4432 W
Total Tissue Mass	0.8153 kg	0.8153 kg
SAR Averaging Sensor (1g Average)		
Maximum Value	5.035 W/kg	9.959 W/kg
Location of Maximum	(-11.937 mm, 0.7993 mm, 53.75 mm)	(-9.945 mm, 12.765 mm, 25.25 mm)
Average Raw SAR Value	0.3097 W/kg	0.5422 W/kg
Total Power Dissipated	0.2525 W	0.4432 W
Total Tissue Mass	0.8153 kg	0.8153 kg

Figure 7.39: SAR Statistics for Triple-Band Slot Antenna Near the SAM Head with a Discrete Port

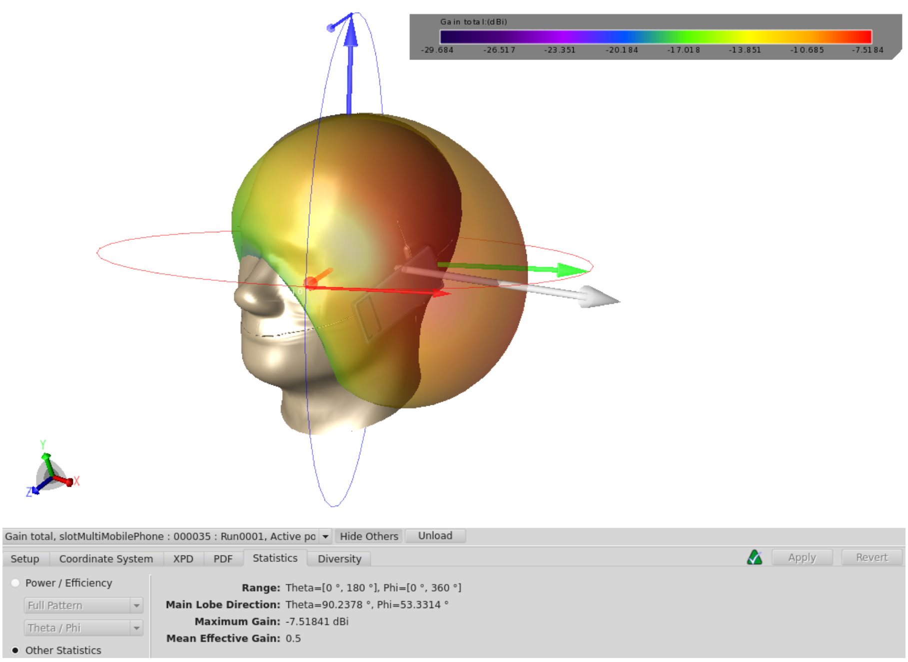


Figure 7.40: Far-zone Pattern for Triple-Band Slot Antenna Near the SAM Head with a Waveguide Port

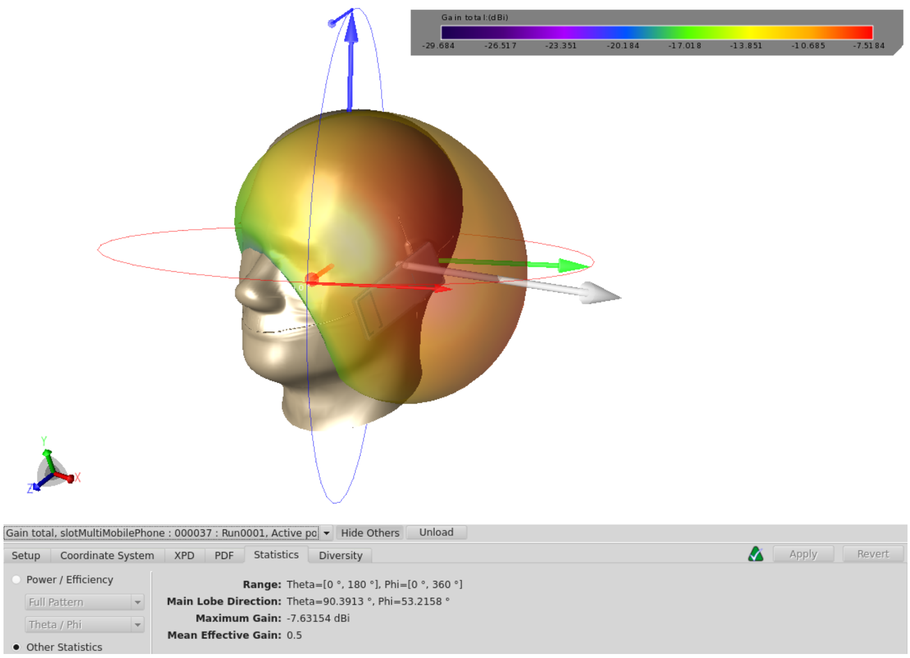


Figure 7.41: Far-zone Pattern for Triple-Band Slot Antenna Near the SAM Head with a Discrete Port

7.5 Plane Wave Editor

Of the two varieties of external excitations available in XFdtd the first is a plane wave. A plane wave source is assumed to be infinitely far away so that the constant field surfaces are planar and normal to the direction of propagation. The plane wave excitation is useful for computing radar cross section and scattering behavior.

To access the PLANE WAVE EDITOR, double-click on an existing PLANE WAVE object in the EXTERNAL EXCITATIONS branch of the PROJECT TREE, or right-click on the branch to create a NEW PLANE WAVE. The editor will open at the top of the GEOMETRY WORKSPACE WINDOW, and consists of a LOCATION tab and a PROPERTIES tab.

Figure 7.42: The Location Tab of the Plane Wave Editor

On the LOCATION tab, specify the incident direction using the INCIDENT THETA and INCIDENT PHI controls. INCIDENT THETA is the defined as the angle between the positive Z-axis and the incident direction, while INCIDENT PHI is defined as the angle between the positive X-axis and the projection of the incident direction into the XY plane. The incident wave POLARIZATION can also be specified on this tab.

Figure 7.43: The Properties Tab of the Plane Wave Editor

On the PROPERTIES tab, the excitation can be given an identifying NAME. An excitation WAVEFORM must be selected, and the AMPLITUDE of the excitation must be specified. The electric field values in the X,Y and Z directions resulting from the choice of amplitude and incident direction are displayed with the INCIDENT AMPLITUDES heading in this window.

Plane wave excitations are enabled or disabled by selecting the source type in the SIMULATIONS workspace window (see Section 11.1.1). Since passive lumped loads do not radiate energy, they may be present in a simulation which is excited by a plane wave.

All active ports will be set to passive lumped loads when a plane wave excitation is selected.

7.6 Gaussian Beam Editor

The second external excitation definition available is the GAUSSIAN BEAM source. The focused Gaussian beam is characterized by an incident electric field that has a two-dimensional, radially-symmetric Gaussian distribution in planes normal to the incident direction. It converges to maximum intensity at the focus point. All calculations with a Gaussian beam source are performed in scattered-field, though total-field values may also be saved. Unlike the plane wave and discrete sources, this source requires that the source waveform be sinusoidal.

✓ Examples where this type of source is useful include structures used at optical frequencies and situations where it is desired to illuminate only a portion of the geometry.

To access the GAUSSIAN BEAM EDITOR, double-click on an existing GAUSSIAN BEAM object in the EXTERNAL EXCITATIONS branch of the PROJECT TREE, or right-click on the branch to create a NEW GAUSSIAN BEAM. The editor will open at the top of the GEOMETRY WORKSPACE WINDOW, and consists of a LOCATION tab and a PROPERTIES tab.

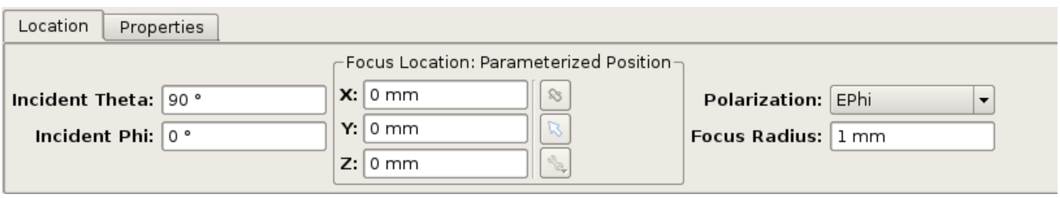


Figure 7.44: The Location Tab of the Gaussian Beam Editor

The controls on the LOCATION tab behave identically to those on the PLANE WAVE EDITOR (Section 7.5), with two important additions. The FOCUS LOCATION controls specify the location in the simulation space of maximum intensity of the excitation. The X, Y and Z coordinates can be entered manually, or chosen by graphically using the PICKER TOOL (See Section 7.1.1 for more information on using the point selection tools shown here.)

The FOCUS RADIUS editor sets the beam radius ω_0 at the focal point in the plane normal to the direction of travel at which the field strength drops to $\frac{1}{e}$ $\approx$ -8.686 dB of its maximum value. In other planes normal to the direction of travel, ω is the radius at which the field strength drops to $\frac{1}{e}$ of its maximum value in that plane and is given by:

$$\omega^2(z) = \omega_{02}(1 + \frac{\lambda z}{\pi \omega_{02\,2}})$$

where λ is the free space wavelength and, for simplicity, the direction of travel is assumed to be parallel to the Z-axis and the focus in the $z = 0$ plane.

Figure 7.45: The Properties Tab of the Gaussian Beam Editor

The controls on the PROPERTIES tab behave identically to those on the PLANE WAVE EDITOR (Section 7.5.) Note that only Sinusoidal waveforms can be used with a Gaussian Beam.

Gaussian beam excitations are enabled or disabled by selecting the source type in the SIMULATIONS workspace window (see Section 11.1.1.) Since passive lumped loads do not radiate energy, they may be present in a simulation which is excited by a Gaussian Beam.

All active ports will be set to passive lumped loads when a Gaussian beam excitation is selected.

7.7 Waveform Editor

The WAVEFORM EDITOR is used to edit waveforms that can be used in conjuction with an EXTERNAL EXCITATION or a FEED to inject energy into the space for an FDTD simulation. When used in a simulation, a waveform's field value at each timestep is applied to the field in the space as part of the field update calculation.

To create a WAVEFORM, right-click on the DEFINITIONS:WAVEFORMS branch of the PROJECT TREE and select NEW WAVEFORM DEFINITION. Double-clicking on a Waveform under the same branch will open that waveform for editing. There are seven types of waveforms available:

- AUTOMATIC
- BROADBAND
- GAUSSIAN
- GAUSSIAN DERIVATIVE
- MODULATED GAUSSIAN
- SINUSOID
- USER-DEFINED

The choice of waveform should be based on the desired output. For broadband calculations, the AUTOMATIC, BROADBAND or one of the Gaussian-type waveforms should be used since they will inject energy into the space in a wide band of frequencies. The SINUSOID may be chosen for cases where steady-state results are desired at a single frequency. The USER DEFINED waveform can be used when none of the other choices meet the requirements of the simulation.

The default when creating a new waveform is the AUTOMATIC waveform. It is recommended to use this waveform unless a specific frequency response is desired.

To produce valid results within a specific frequency range, the signal-to-noise ratio (waveform to computational noise) must not be too small. All waveform windows display a line at -60 dB as a simple indicator of when the signal-to-noise ratio is becoming too small to be generally useful. Of course, the actual useful signal-to-noise ratio floor is problem dependent. The marker also notes that appreciable amounts of energy will not be introduced into the simulation space at frequencies around or below the line.

7.7.1 Automatic

The AUTOMATIC waveform dynamically changes shape to excite the Frequency Range of Interest defined for the project. If the Frequency Range of Interest is a non-empty range including zero then the Automatic waveform will take on the shape of a Gaussian including a DC component. If the specified range is a non-empty range excluding zero the waveform will take on the shape of a Modulated Gaussian with no DC component. In each of these broadband cases the Automatic waveform will be 20 dB down at the non-zero range limits. This is done to avoid exciting frequencies outside the specified range. If the specified range is a single value the Automatic waveform will be a Sinusoid at the specified frequency. The Automatic waveform will dynamically adjust to any changes in the Frequency Range of Interest.

7.7.2 Broadband

When broadband results are desired, a BROADBAND waveform should almost always be used. This waveform provides a Gaussian pulse with the largest frequency content possible for the specific FDTD space when EXCITE ALL POSSIBLE FREQUENCIES is selected. Alternatively, the user can clamp the upper end of the frequency range by selecting EXITE UP TO A MAXIMUM FREQUENCY and choosing a frequency response magnitude and corresponding frequency. In this case, the waveform frequency response is truncated in the simulation to the maximum allowed, even if a wider frequency response is specified in this editor. Figure 7.46 shows a Broadband waveform in the WAVEFORM EDITOR.

- See Section 7.7.8 for more information on waveform frequencies.

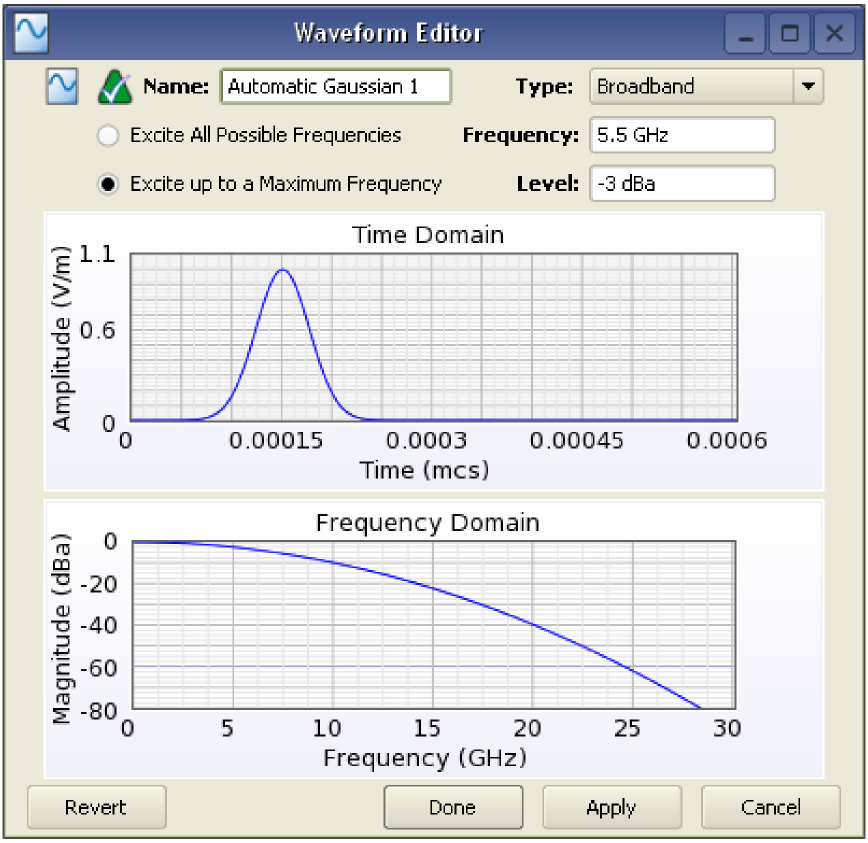


Figure 7.46: Defining a Broadband waveform in the Waveform Editor

7.7.3 Gaussian

The GAUSSIAN waveform provides broadband input and is also suitable for use when broadband results are desired. The width of the pulse is user-specified. Figure 7.47 shows a Gaussian pulse in the WAVEFORM EDITOR.

- See Section 7.7.8 for more information on specifying the width of the pulse.

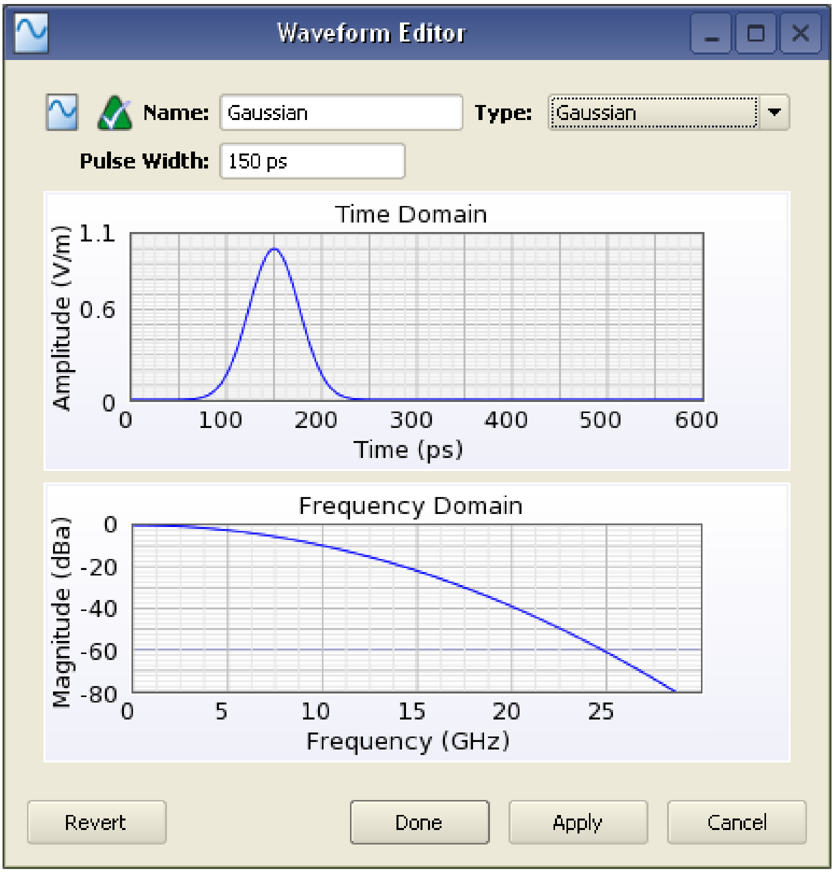


Figure 7.47: Defining a Gaussian pulse in the Waveform Editor

Since the Gaussian pulse has a non-zero average value, it should not be used for a FEED when there is a closed path (loop) of perfect conductor connected to the Feed unless the feed also contains resistance. This is because the Gaussian has a DC (zero-frequency) component which starts a steady current flowing in the loop that will never decay through loss or radiation. The symptom of this will be a source current that has an average value not equal to zero. If this occurs, the GAUSSIAN DERIVATIVE or MODULATED GAUSSIAN pulses can be used or a non-zero source resistance specified for the Feed.

7.7.4 Gaussian Derivative

The GAUSSIAN DERIVATIVE is nearly identical to the Gaussian except that it has a zero average value and thus the DC component is removed. For the same simulation, the pulse width of a Gaussian Derivative should be set somewhat greater than what would be set for a Gaussian since the Gaussian Derivative will have a wider frequency spectrum for the same pulse width. Figure 7.48 shows a Gaussian Derivative in the WAVEFORM EDITOR.

- See Section 7.7.8 for more information on pulse width.

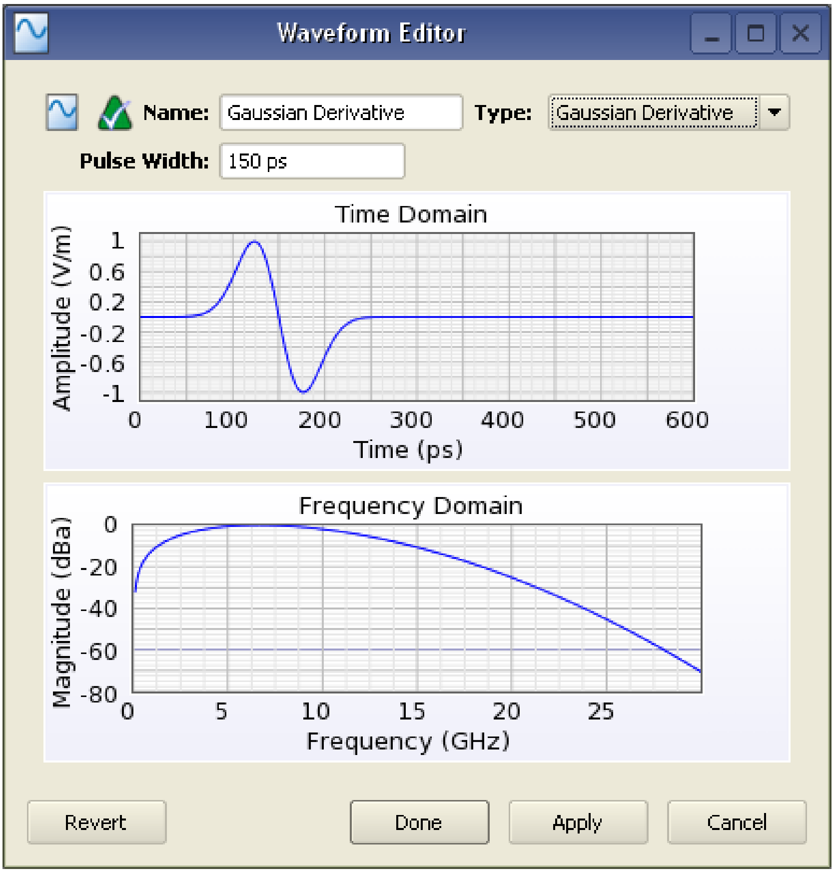


Figure 7.48: Defining a Gaussian Derivative waveform

7.7.5 Modulated Gaussian

The MODULATED GAUSSIAN should be used only when a specific frequency range is desired. This is useful in structures where low frequencies could excite non-radiating modes that could resonate and invalidate results. This waveform is a sinusoid with a Gaussian envelope with the sinusoid centered in the envelope so that the average of the pulse is zero.

Figure 7.49 shows a MODULATED GAUSSIAN in the WAVEFORM EDITOR.

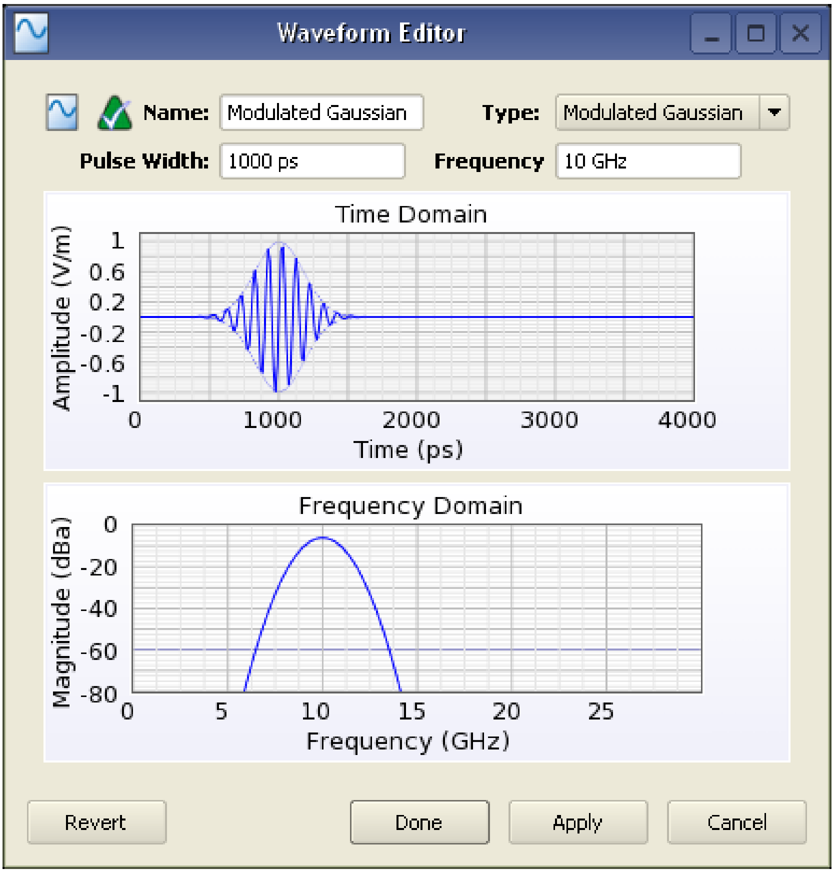


Figure 7.49: Defining a Modulated Gaussian waveform

The PULSE WIDTH for the Modulated Gaussian may be adjusted to enclose a specific frequency range. This is useful, for example, in waveguide simulations so that only frequencies in the band of single-mode operation are excited. It is also useful when band-limited devices are being simulated. For example, a broadband antenna, such as a spiral, may be designed for a specific frequency range. Exciting the antenna at frequencies outside this range may greatly increase the simulation time needed for convergence since the out-of-band energy cannot readily radiate or be otherwise dissipated by the antenna structure.

- See Section 7.7.8 for more information on pulse width.

7.7.6 Sinusoid

The SINUSOID is useful when only one frequency is of interest. When this waveform is used, a single frequency simulation is performed that provides additional result types over broadband simulations. The "ramped" part of the waveform is used and automatically configured to avoid introducing energy at any frequency other than that of the sinusoid into the simulation.

Since the Sinusoid is different than the other waveforms, it requires some special considerations. It is recommended that the length of a simulation be at least five cycles of the sinusoid when not using automatic convergence. In some cases, such as extremely low frequencies, running five cycles may not be practical, in which case even just a fraction of a cycle may be used.

Figure 7.50 shows a SINUSOID in the WAVEFORM EDITOR.

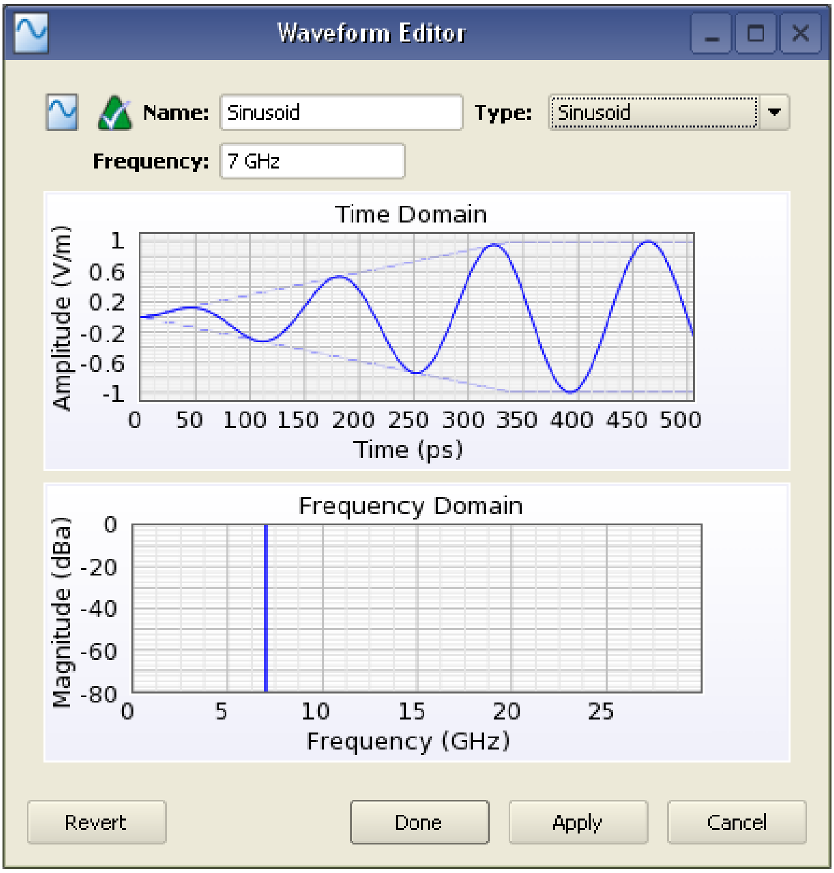


Figure 7.50: Defining a Sinusoid waveform

7.7.7 User Defined

An arbitrary waveform may be specified using the USER DEFINED waveform type. A time record of the waveform, pre-discretized in time at the timestep of the simulation, is imported from a text file (`*.src`.) The format of the file depends whether the waveform will be used for a DISCRETE VOLTAGE SOURCE or PLANE WAVE SOURCE, and is described in the following subsections.

The frequency response of the User Defined waveform is computed from the input data. Figure 7.51 shows a USER DEFINED waveform in the WAVEFORM EDITOR. An example user-defined waveform file is illustrated in Figure 7.52.

7.7.7.1 Discrete Source Format

The format for a DISCRETE VOLTAGE SOURCE-formatted user-defined waveform file is a single integer, N, in the first line of the file, which is the number of samples in the file, followed by N lines that define the sampled waveform at successive timesteps. Each line contains two numbers which are the normalized voltage of the waveform at the source followed by its time derivative. The first datapoint (on line 2 of the file) corresponds to time = timestep. The next datapoint (on line 3 of the file) corresponds to time = 2*timestep and so on. If the file contains data for fewer timesteps than a simulation needs, XFdtd will use zeroes for the unaccounted-for timesteps (see Figure 7.52.)

7.7.7.2 Plane Wave Source Format

The format for a PLANE WAVE SOURCE-formatted user-defined waveform file is a single integer, N, in the first line on the file, which is the number of samples in the file, followed by N lines that define the sampled waveform at successive timesteps. Each line contains two numbers which are the normalized electric field magnitude (in Volts/meter) followed by its time derivative. The first datapoint (on line 2 of the file) corresponds to time = timestep. The next datapoint (on line 3 of the file) corresponds to time = 2*timestep and so on. If the file contains data for fewer timesteps than a simulation needs, XFdtd will use zeroes for the unaccounted-for timesteps (Figure 7.52.)

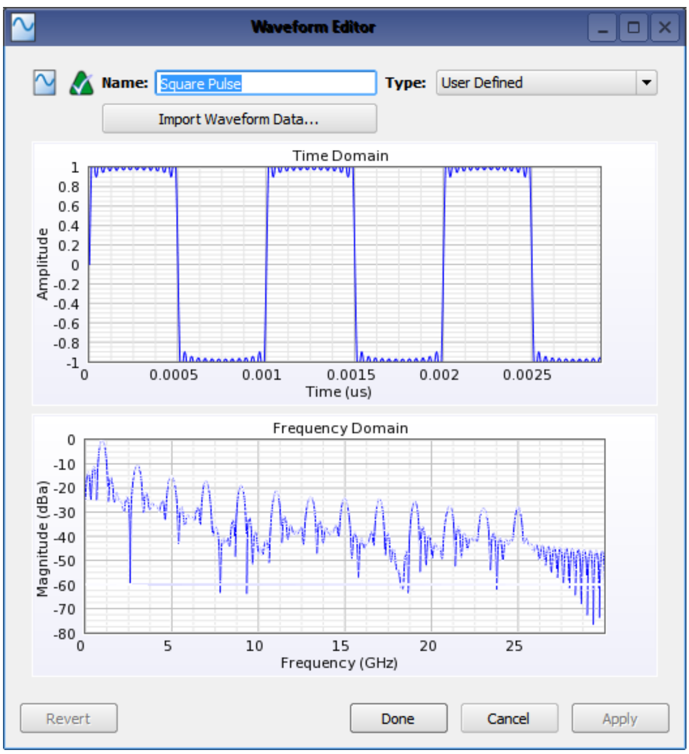


Figure 7.51: Defining a User Defined waveform

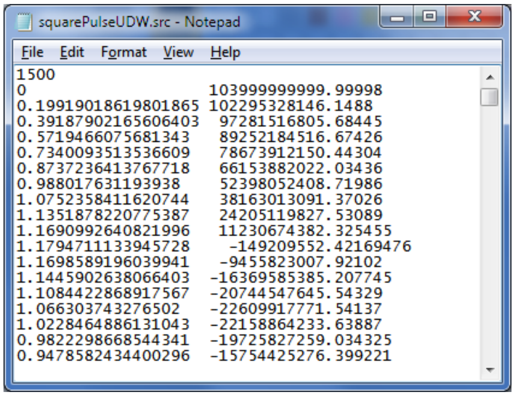


Figure 7.52: A User Defined waveform file

7.7.8 Notes on Choosing Waveform Parameters

For obtaining broadband results, the usual choice for a waveform should be the AUTOMATIC waveform. However, for certain situations, a specific pulse type may be desired. For these cases, when choosing the pulse width and/or frequency for a waveform, the constraints of the FDTD method must be kept in mind. The time rate of change of a pulse or the sine wave frequency must be low enough so that the waveform is accurately sampled. This results in the frequency being constrained by the Courant limit:

$$frequency \leq \frac{1}{10\triangle t\sqrt{3}} = \frac{c}{10\triangle x}$$

where

$\triangle t$ is the timestep

c is the speed of light

$\triangle x$ is the largest grid edge length in the space.

This prevents the waveform from introducing energy at frequencies too high for the FDTD method to produce accurate results. The pulse width should also not be so large (and thus the frequency content so low) that the calculation is not excited at frequencies for which accurate and useful results might be obtained. There must be sufficient energy in the pulse in the frequency range of interest for results to be above the numerical noise. For simulations with very small cells compared to the shortest wavelength of interest, pulse widths may be set much larger than the default size to reduce the number of timesteps needed for convergence and to increase stability.

If dielectric materials are present in the FDTD space, wavelengths are reduced inside those materials since the velocity of propagation is less than the speed of light in free space. A reasonable rule of thumb for lossless or low-loss dielectrics is that the maximum frequency for the spectrum of the pulse should be reduced by the square root of the relative permittivity (ϵ_r) times the relative permeability (μ_r), or equivalently, the PULSE WIDTH, τ, should be increased by this factor:

$$\tau_{new} = \tau_{old}\sqrt{\epsilon_r \mu_r}$$

where

τ_{new} represents the corrected pulse width

τ_{old} represents the uncorrected pulse width.

For more lossy dielectrics or conductors, the frequency and pulse width should be adjusted proportionally to the change in wavelength in the material relative to the free space. Of course, the maximum frequency for reliable results is also reduced as the frequency spectrum of the excitation pulse is reduced.

For example, suppose part of the simulation space is free space and part is a low-loss dielectric with a relative permittivity of 4.0. Further suppose the cell size is 1 cm. At 10 cells per wavelength, one would expect reasonable results up to 3 GHz with a Gaussian pulse 32 timesteps wide (for a uniform grid.) But since part of the space has a dielectric material, with the same cell size the Gaussian PULSE WIDTH should be doubled to 64 timesteps to reduce the frequency spectrum bandwidth. Correspondingly, the maximum frequency for reliable results would be decreased from 3 GHz to 1.5 GHz.

When using the BROADBAND waveform, XFdtd chooses (in the case of automatic pulse width selection) or limits the pulse width (in the case of user-specified frequency content) of the Gaussian pulse based on the restrictions above. When using a waveform with user-specified pulse widths, it checks to make sure the frequency limit, as described above, is not exceeded. If it is, then an error is given when the simulation is created.

7.8 Static Voltage Points

The STATIC VOLTAGE POINTS feature is used to create and place a voltage point on an object made of PEC material. These points are used by the Laplace static solver to initialize the starting E field within the problem space to voltage values assigned by the user. With no static voltage points defined, the initial E field at the beginning of the FDTD computation is set to 0 V/m at each cell edge.

To use this feature, right-click the Static Voltage Points branch of the PROJECT TREE and select New Static Voltage Point as seen in Figure 4.16. A static voltage point will be added to the project. Double click the new point to edit the point's properties.

Static Voltage Points are specified by graphically picking or manually entering the location of the static voltage point and by specifying the voltage. Click the **Done** button to apply your changes.

Keep in mind that only one entry per conducting object is necessary. Duplicate entries will overwrite the voltage preset value. (PEC outer boundaries and any untagged metal objects will be preset to 0_V. If the boundary is PML, the fields at a PML boundary will be initialized appropriately to prevent non-physical reflections at the interface.) Additionally, static voltage points can be used by themselves to excite an object containing PEC materials, or in conjunction with an external excitation or a discrete source.

Figure 7.53 shows the available parameters when creating a static voltage point.

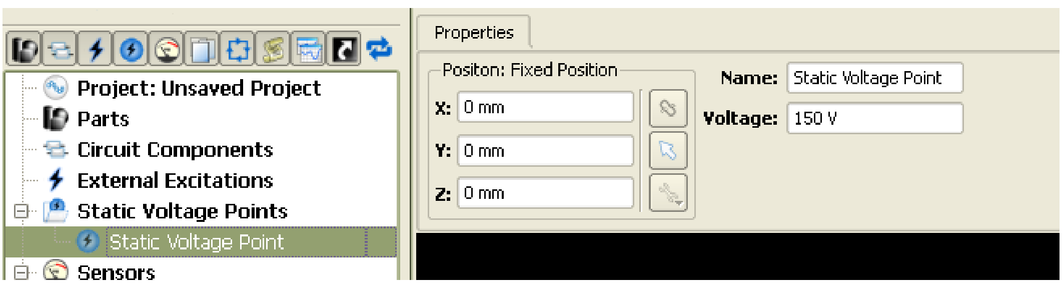


Figure 7.53: Defining a static voltage point

Chapter 8

Defining the Grid and Creating a Mesh

In this chapter, you will learn...

→ about the differences between the grid and the mesh

→ about the tools available for you to specify the properties of the XFdtd grid

→ how to set localized grid properties, right down to individual objects or single cells

→ how to tell XFdtd which objects to include in the mesh and in what order to consider them

→ how to adjust the way touching objects are meshed

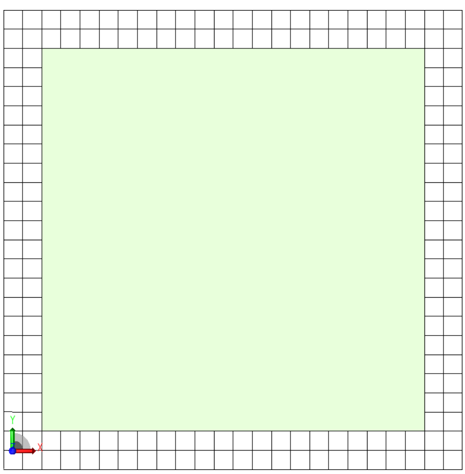


Figure 8.1: An illustration of an object surrounded by the grid

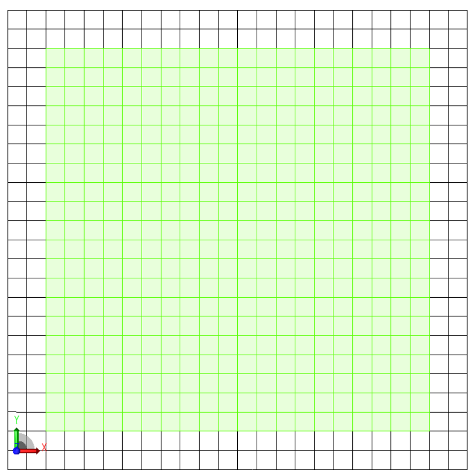


Figure 8.2: An illustration of the grid (black lines) and the electic mesh (light green lines)

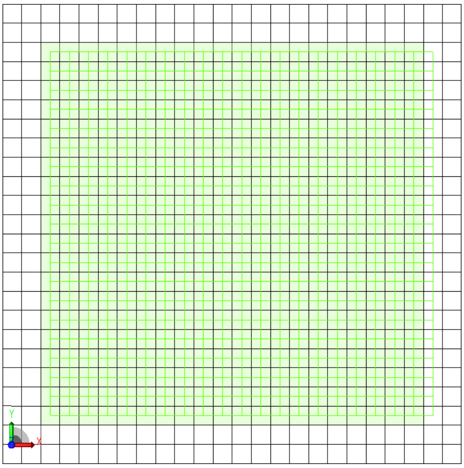


Figure 8.3: An illustration of the grid (black lines) and the magnetic mesh (light green lines). The magnetic mesh is offset from the electric mesh.

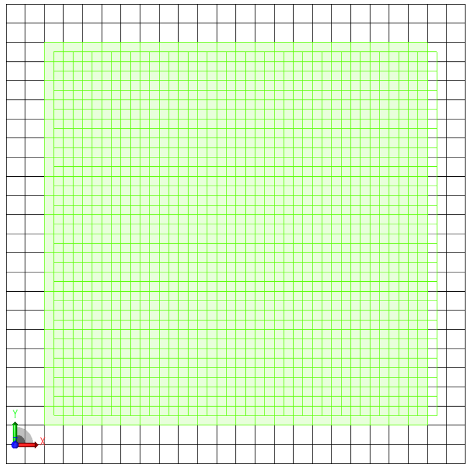


Figure 8.4: An illustration of the grid (black lines) and the electic and magnetic meshes (light green lines)

The XFdtd method requires the simulation space to be discretized in order to perform calculations. To simplify workflow, XFdtd separates the processes of defining the locations of the discrete space, gridding, and assigning materials to the defined locations. This section will further define these concepts and explain how to use them efficiently in XFdtd. To create a successful grid and mesh, users must understand the following components:

- **Part**: A part is a geometric shape either created by a user or imported into XFdtd. After creating or importing a part, a user assigns a material to it. The material assigned to each part determines its electromagnetic characteristics during the calculation.
- **Grid**: The gridding process takes the continuous space and chooses discrete axis-aligned locations. XFdtd provides tools in the GEOMETRY - EDITING GRID window to adjust the position and spacing of grid edges along the X, Y and Z axes. These tools enable XFdtd to provide an accurate representation of the solid parts while limiting the required memory and simulation time needed for the calculation.
- **Mesh**: XFdtd creates the mesh by sampling at all grid locations throughout 3-D space to determine which grid locations should be set for the mesh. It then assigns a material to each edge of the grid after inspecting where the parts and the grid overlap. An easy way to think of this process is through the simple formula ”Objects + Grid = Mesh.” Grid edges where a part exists use the material assigned to that part. All other grid edges use free space material. Circuit components also are included in the mesh at their nearest discrete location. When the lines in the grid change, the mesh changes as well.

General grid definitions are assigned to the entire grid within the GRID TOOLS dialog of the GEOMETRY workspace window. Here, definitions such as target cell size, bounds and limits are defined. Customizable grid definitions such as fixed points and grid regions can also be added.

The GRIDDING PROPERTIES EDITOR provides tools to modify the grid locations in and around individual parts thus allowing important parts to be resolved more accurately during the discretization process. This editor is useful for objects whose characteristics are not adequately considered by the general grid definitions set within the GRID TOOLS editing tabs.

Finally, the MESHING PROPERTIES EDITOR governs how each object within the boundaries of the defined grid will be meshed. The meshing order determines which object's material is assigned to mesh edges in the case of spatial conflicts. By default, every object in the project is meshed with uniform priority, but these settings can be adjusted within this editor so that higher-priority objects are meshed more carefully than lower-priority objects. In cases where parts have the same meshing priority, XFdtd uses the order of the parts in the PROJECT TREE.

- Appendix B also supplies a thorough discussion of gridding theory, concepts and recommendations within XFdtd.

The following sections explain how to enter the grid properties and specify the various grid elements using these editors.

8.1 Grid Tools

There are two ways to access the GRID TOOLS dialog. With the GEOMETRY workspace window open, either select GRID TOOLS from the drop-down menu in the window, or double-click on the FDTD: GRID branch of the PROJECT TREE. Once the dialog is open, press the EDIT GRID button to define the grid parameters for the project.

The tabs located within grid tools follow the step-by-step editing process that is used to define the grid. The first tab, SIZE, specifies general definitions that are applied to the whole grid, whereas the second and third tabs, FIXED POINTS and GRID REGIONS, allow the user to customize the grid by manually placing gridlines and changing the cell characteristics of different grid regions.

- See Appendix Section B.3 for a thorough discussion of fixed points and grid regions.

The fourth tab, LIMITS, applies constraints to the grid so that memory requirements and adjacent cell size ratios do not exceed a reasonable value. Since grid definitions largely influence the memory requirement of a particular calculation, two bars are always visible in order to monitor how much memory the intended grid definitions will use. The CURRENT bar the total amount of memory required to simulate the project, while the NEW bar shows how much memory will be required to simulate the project based on the proposed changes. Mousing over either memory estimate, displays a tooltip. The tooltip contains additional estimates (Minimum Usage, Maximum Usage and Best Estimate), and an explanation of how the estimate was derived. The Limits tab also allows specifying a smallest desired cell size. If a grid definition requires a cell smaller than the smallest desired size, the grid becomes invalid and a warning message appears.

The fifth tab, INFO, provides data about both the current and the proposed grid (the two sets of values will be the same if any proposed changes have been applied.) The grid's dimensions, upper and lower boundaries, cell counts, and minimum and maximum edge lengths are shown. A value is shown for each axis, X, Y and Z.

8.1.1 Size Tab

The SIZE tab allows the user to select whether XFdtd automatically defines the cell size and the placement of the grid at the boundaries of the project (BASIC MODE), or whether the user defines these parameters themselves (ADVANCED MODE). In Basic Mode, users define a single cell size/padding and XFdtd creates a uniform grid in X, Y and Z. In Advanced Mode, users can specify cell size/padding in each direction.

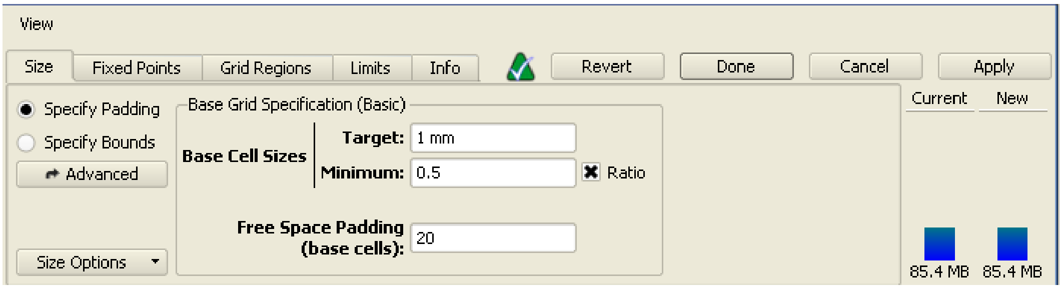


Figure 8.5: The Basic mode enables XFdtd to define cell size and placement

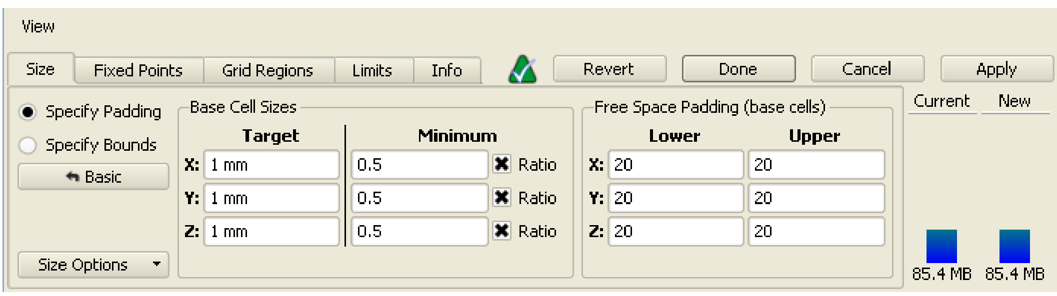


Figure 8.6: The Advanced mode enables users to define cell size and placement

Specify Padding

In Advanced Mode, there are two options for defining the outer boundaries. The first option is to specify the padding around the project's lower and upper boundaries in terms of discrete numbers of cells. The second option is to specifying a grid bounding box by its physical coordinates, independent of the geometry. Each option is detailed below.

Figure 8.7 displays the available options for specifying the padding of the outer boundaries of an XFdtd cell.

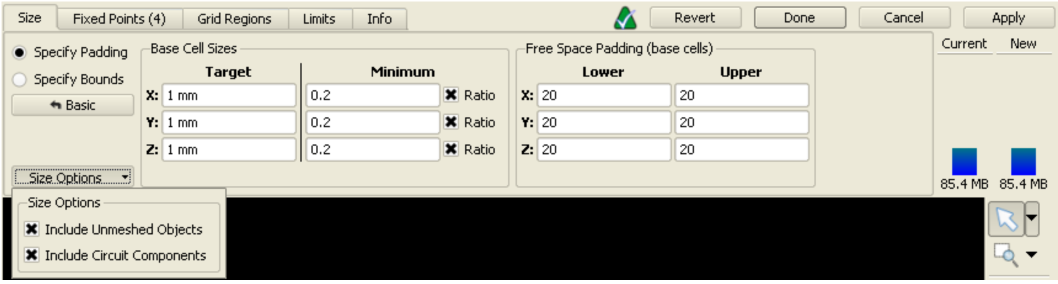


Figure 8.7: Defining a project's boundaries by cell padding within the Size tab

When SPECIFY PADDING is selected, the number of padding cells is multiplied by the main grid delta. Fixed points are placed at that distance from the edges of the geometry. The grid lies between those exterior fixed points. In the absence of any manual grid regions, manual fixed points, part grid regions or part-associated automatic fixed points, the space between the exterior fixed points is divided into cells with even spacing. The minimum number of cells, given the extent and the main grid size, is used. Each evenly-spaced cell will have an edge length no greater than the target cell size.

BASE CELL SIZES allows entry of the target cell size when generating the XFdtd grid in areas that are not part of a grid region. Note that this is the target cell size and may not reflect the actual cell size in the grid. The actual cell size may be slightly lower depending on the constraints placed on the grid, but it will never be higher, ensuring that the desired level of detail is not compromised. When there are two fixed points or grid region bounds of any type, the region between them will be divided into a number of cells. The number of cells that is required depends upon the target cell size. No cells larger than the target will be generated. The target cell size may not divide evenly into the distance between the fixed points or grid region bounds. The gridding process prefers to generate cells of a constant size. First the minimum number of cells required to span the distance is calculated and then a cell size, less than or equal to the target size, is calculated so that the distance is spanned exactly.

- See Appendix B for more details and for exceptions to the general rules.

FREE SPACE PADDING (IN CELLS) allows users to define the extent of the padding space around the geometry. The extent is specified as a number of cells, each of which is taken to be the target cell size. In simple projects this means that the number of cells specified will be added as free space around the geometry (at the main grid cell size.) In more complex projects, grid regions close to the boundary may cause transition regions containing smaller cells to intrude into the free space padding. The free space padding's extent will remain the same, whether it is made up of the specified number of cells each exactly the target cell size, or whether the padding extent contains some number of smaller-sized transition region cells.

There are also two checkboxes available in the SIZE OPTIONS drop-down list: INCLUDE UNMESHED OBJECTS and INCLUDE CIRCUIT COMPONENTS. Selecting either of these options will ensure that any unmeshed object and/or circuit component is included within the padded region. Otherwise, only meshed objects will be used to determine the size of the padded region. These options may also affect the size of the region bounded by free space padding.

Specify Bounds

Figure 8.8 displays the available options for defining the outer bounding box.

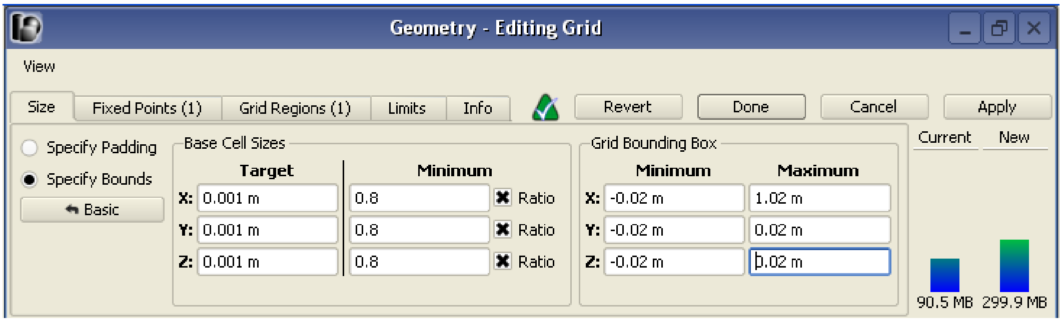


Figure 8.8: Defining a project's boundaries by a bounding box within the Size tab

There are some instances when defining a GRID BOUNDING BOX rather than a padding of free space cells may be advantageous in placing the outer limits of the grid. The grid bounding box, unlike cell padding, is independent of the geometry, so it is possible to define a grid bounding box that does not include the entire geometry. Like the SIZE tab, the GRID BOUNDING BOX has a BASIC MODE and an ADVANCED MODE (Figure 8.8.)

BASE CELL SIZES is the same as the BASE CELL SIZES definition under SPECIFY PADDING.

8.1.2 Fixed Points Tab

Fixed points are essentially gridlines placed at discrete locations in the mesh, and therefore control the placement of cell boundaries in the mesh. They are used to ensure that important parts of the geometry are considered at cell edges which cannot be controlled by uniform meshing. For example, it may be useful to place a fixed point at the edge of a PARTS object in the geometry so that its edge does not fall between cell edges.

The FIXED POINTS tab stores the definitions associated with the placement of fixed points in the geometry. To add a fixed point to the mesh, select the ADD button and define the FIXED POINT PROPERTIES by typing in the physical coordinates of interest. Note that any of the three principle directions may be checked or unchecked (in the FIXED checkbox) depending on how many points are to be defined. For example, if only the X-coordinate is defined, a fixed point will be added that intersects at that location on the X-axis, which defines a plane in YZ. Defining fixed points in all three principle directions will result in three fixed points, intersecting at the defined point.

Figure 8.9 displays the dialog associated with defining fixed points. In this instance, only X and Y are checked, so two fixed points spanning the X- and Y-planes will be added in the X (at X = -0.2 m) and Y (at Y = -0.2 m) plane, respectively.

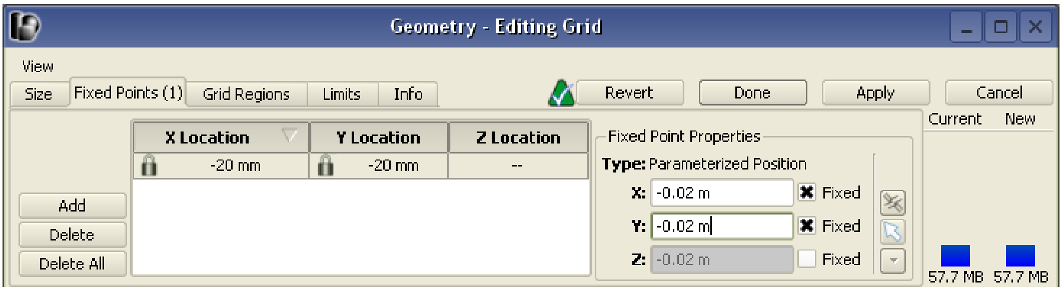


Figure 8.9: Defining Fixed Points within Grid Tools

✓ A fixed point can also be defined by checking the X, Y, and/or Z coordinate check-boxes of interest and clicking on the desired location of the plane in the simulation space with the SELECTION tool.

8.1.3 Grid Regions Tab

A GRID REGION is a bound region in the mesh that is assigned a cell size different from the cell size assigned in the SIZE tab. Grid regions may be d esirable in locations that require finer meshing and thus require smaller cells. It is also advantageous to use them in areas that are not important to the XFdtd calculation by defining them with larger cells to save memory and calculation time. Simply define the REGION BOUNDS and the CELL SIZES in each direction of interest, and a region will be placed on the grid.

Figure 8.10 displays the dialog associated with defining grid regions. In this example, the cells within the particular grid region are defined to be 0.001 m. Automatic fixed points will be merged to be spaced 60% of this distance as defined in the MINIMUM boxes.

- See Appendix B for more information about fixed point merging.

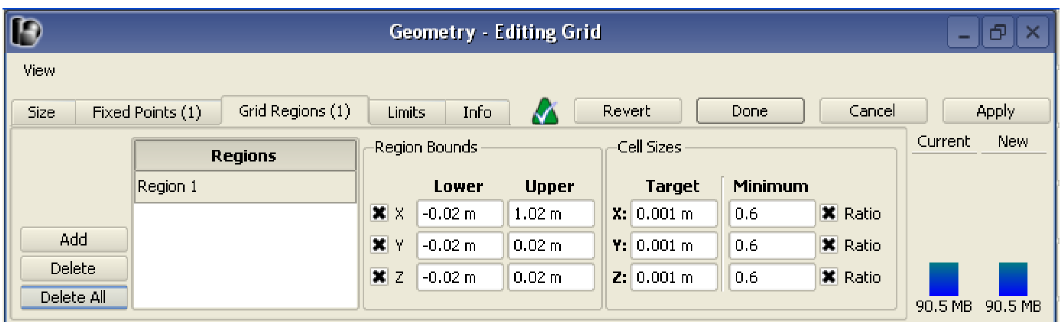


Figure 8.10: Defining Grid Regions within Grid Tools

8.1.4 Limits Tab

The LIMITS tab contains several required and several optional limits used to restrict characteristics of the grid.

The MAXIMUM CELL STEP FACTOR definition restricts the rate at which the cell size may increase between two adjacent cells. For example, if a cell is 1 mm long and the maximum cell step factor is set to 2, the next cell cannot exceed 2 mm, because that would cause the cell ratio between these two cells to exceed 2. Therefore, if a grid region has cells of 1 mm, and the default global grid target is 5 mm, additional cells are needed, forming a transition region between the grid region and the Global grid, since 1 mm:5 mm exceeds the 1:2 ratio.

- The ratio of 1:2 is used in this example because it is the largest ratio recommended in Chapter 11, Section 2 of the Taflov and Hagness text [2].

The second required limit, MINIMUM CELL SIZE, represents the global minimum cell size allowed in the project. No cell in the project can fall below this limit or the project will not be valid. A good rule of thumb for setting the MINIMUM CELL SIZE is to make it 1/2 of the target cell size. There are other MINIMUM CELL SIZE definitions within the GRID TOOLS and GRIDDING PROPERTIES EDITOR, but they must be greater or equal to this global definition to be considered a valid definition.

The two optional limits aid in restricting the size of the XFdtd project so that the calculation time and memory requirements do not exceed a defined limit. These limits can be applied to the project by restricting the number of MAXIMUM CELLS that are contained within the mesh. If this definition is applied to a project and exceeded, an error message will appear to indicate this.

Figure 8.11 displays the LIMITS tab.

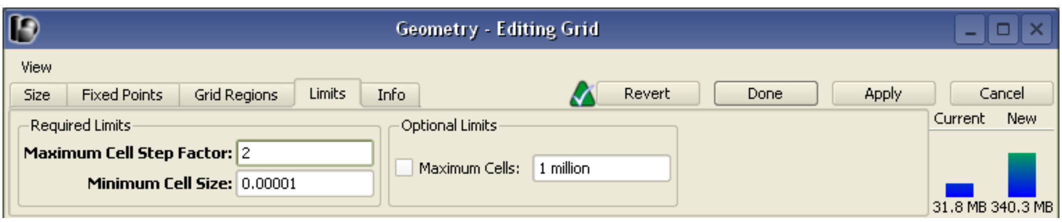


Figure 8.11: Defining Grid Regions within Grid Tools

8.2 Gridding Properties Editor

The GRIDDING PROPERTIES EDITOR window is used to define grid regions and fixed points associated with an individual object. To open the GRIDDING PROPERTIES EDITOR window, right-click on the appropriate object in the PARTS branch of the PROJECTTREE and select GRIDDING / MESHING > GRIDDING PROPERTIES as seen in Figure 8.12.

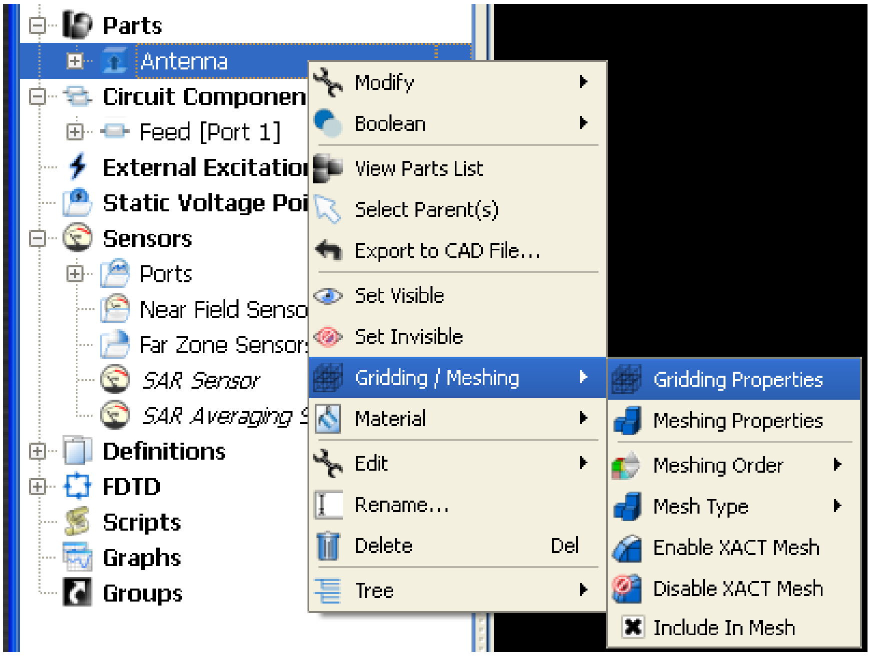


Figure 8.12: Accessing the Gridding Properties Editor

Like the SIZE tab in the GRID TOOLS dialog, the GRIDDING PROPERTIES EDITOR window contains a BASIC MODE and an ADVANCED MODE. In BASIC MODE, users enter cell size and boundary extension information, then XFdtd uses this data to automatically create cells and boundary regions for X, Y and Z.

In Advanced Mode, the USE AUTOMATIC GRID REGIONS and USE AUTOMATIC FIXED POINTS checkboxes enable either of these definitions when they are selected. Figure 8.13 shows the GRIDDING PROPERTIES EDITOR window with both grid regions and fixed points enabled. The following sections detail the definitions associated with grid regions and fixed points.

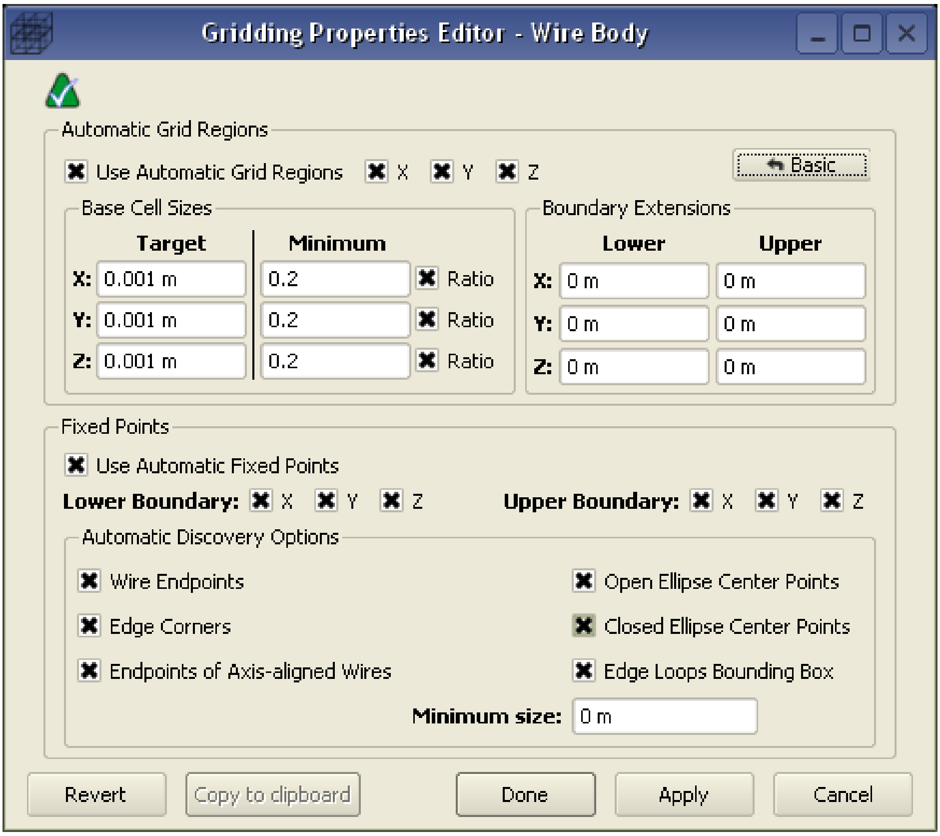


Figure 8.13: The Gridding Properties Editor

8.2.1 Automatic Grid Regions

A grid region is a region assigned a specific cell size within a specified lower and upper boundary. A custom cell size can be applied in this region to any or all of the three principal directions. They are desirable in instances in which the CELL SIZES defined to the entire grid under GRID TOOLS: GRID REGIONS is not sufficient for a particular object.

Grid regions can be added to the grid within GRID TOOLS by defining each region's physical bounds, or a grid region can be applied to a single object within this editor. In this case, the grid region will be applied everywhere within the object (plus any extended region defined in the BOUNDARY EXTENSIONS definition).

- See Section 8.1.3 for more on defining a region's physical bounds.

Cell sizes

The CELL SIZES definition refers to the maximum target cell size that may occur between the lower and upper boundaries. The actual cell size between these boundaries may be slightly lower depending on the location of these boundaries. For instance, if a CELL SIZE is defined to be 1 m in the X-direction and the upper and lower boundaries are defined at X = 1 m and X = 4.9 m, respectively, the actual cell size would be 0.975 m since the distance between the upper and lower boundaries would not permit four 1 m cells within 3.9 m. The actual cell size will never be adjusted to a value higher than the CELL SIZE definition, however.

- See Appendix Section B.2 for more information on choosing an appropriate cell size.

Boundary extensions

The upper and lower boundaries of the grid region usually occur at the edges of the object, but in the case that these bounds are not sufficient, they can be moved inward or outward in any X, Y, and/or Z direction by defining values in the LOWER BOUNDARY EXTENSIONS and UPPER BOUNDARY EXTENSIONS dialog boxes. If a positive distance is defined, the boundary will be shifted at that distance away from the object and conversely, if a negative distance is defined, the boundary will be shifted into the object.

8.2.2 Automatic Fixed Points

Fixed points are analogous to gridlines whose locations are explicitly defined to control where the edges of meshed cells fall in the geometry. Automatic fixed points are given that name because their positions are automatically determined by examining the geometry of the part with which the automatic fixed points are associated. Cell sizes are adjusted to flow as evenly as possible between the fixed points while never exceeding the target cell size at the location of each fixed point.

- See Section 8.1.4 for more on the LIMITS tab.
- For a discussion of how fixed planes vary from grid regions, see Appendix B.3.

Fixed points can be placed anywhere in the grid by specifying their location within the FIXED POINTS tab of the GRID TOOLS dialog, or they can be added within this editor to a particular object based on the object's geometry.

- See Section 8.1.2 for more on the FIXED POINTS tab.

The following settings control where fixed planes are placed in reference to the object's geometry.

Lower and upper boundaries

Places fixed point(s) at the lower or upper boundary of the object in the direction specified.

Wire endpoints

Places fixed point(s) at any discontinuity existing in the object. See Figure 8.14 for an example of a fixed point placed at an edge.

Edge corners

Places fixed point(s) anywhere that there is a discontinuity in the derivative of a object's edge (i.e. points that do not have tangents).

Figure 8.14 shows the fixed points placed at wire endpoints (yellow points) and edge corners (green points) that occur within the simple wire body (shown in white).

Notice how the cell sizes vary to accommodate the placement of the four fixed planes (in red).

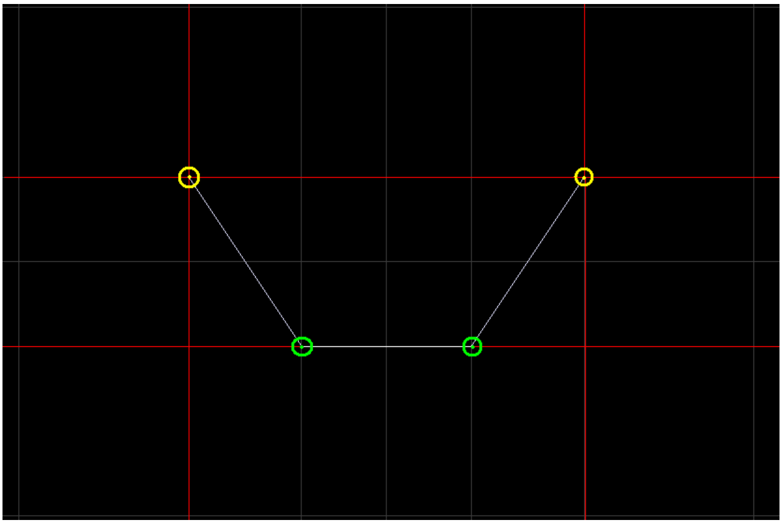


Figure 8.14: Fixed points placed at wire endpoints and edge corners

Endpoints of axis-aligned wires

Places fixed point(s) along any axis-aligned straight edges that exist in the object geometry.

Figure 8.15 shows fixed planes (in red) placed at axis-aligned straight edges.

Notice that straight edges that are not axis-aligned do not have fixed planes.

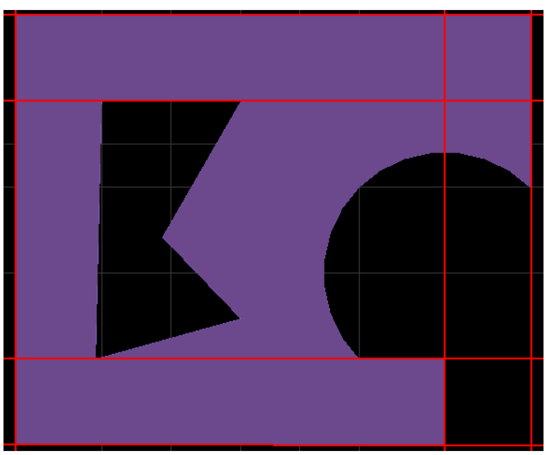


Figure 8.15: Fixed points placed at axis-aligned straight edges

Edge loops bounding box

Places fixed point(s) at the edges of *loops* or closed circuit of edges. Any loop that smaller than the MINIMUM SIZE will be ignored. Thus, to consider all loops that exist in the object's geometry, simply define the MINIMUM SIZE as 0.

Figure 8.16 shows 12 fixed points placed at the edges of loop boundaries.

Notice that the MINIMUM SIZE has been set to 0 so all loops are considered. If the minimum size was larger than one or more loops, no fixed points would be placed at its edges.

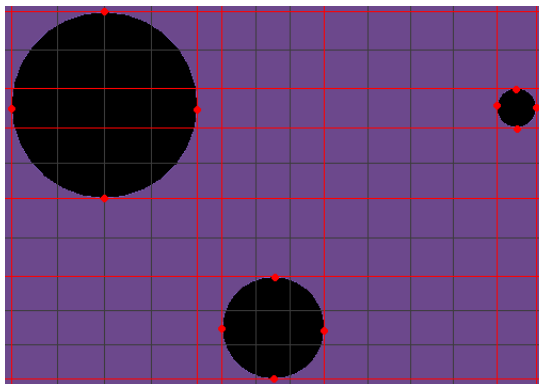


Figure 8.16: Fixed points placed at edges of loops (minimum size = 0)

Open ellipse center points

Places fixed point(s) at the center of open ellipses.

Figure 8.17 shows four fixed planes inside four filleted edges, which are treated as open ellipses.

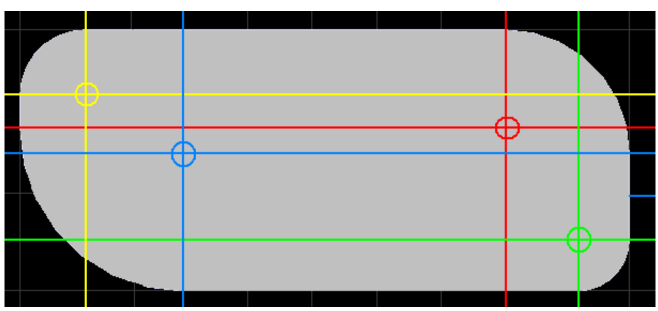


Figure 8.17: Fixed points placed at the center of open ellipses

Closed ellipse center points

Places fixed point(s) at the center of closed ellipses.

8.3 Meshing Properties Editor

The MESHING PROPERTIES EDITOR is used to adjust the properties of the mesh for individual objects. This gives the user greater control over the meshing process, when necessary, in order to create a more accurate mesh. In most cases, the default settings are sufficient. A mesh is generated automatically when calculation files are written, or when grid definitions are applied within the GRID TOOLS dialog and the project is viewed with the MESH VIEW button in the VIEW TOOLS menu.

The meshing properties editor displays different setting configurations depending on the type of part being modified. Figure 8.18 shows the MESHING PROPERTIES EDITOR for a model part, with the menu of meshing modes displayed.

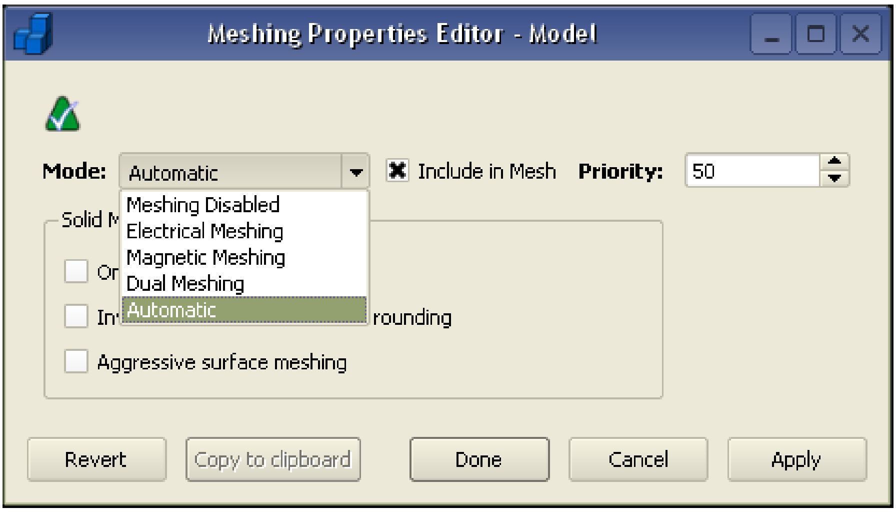


Figure 8.18: The Meshing Properties Editor for a model part

To open the MESHING PROPERTIES EDITOR window, right-click on the PARTS: *Modelname* branch of the PROJECT TREE and select GRIDDING / MESHING > MESHING PROPERTIES.

- When editing the grid, a preview of the grid is created in the MESH VIEW mode, but it is not acually meshed until the grid definitions have been applied.

When AUTOMATIC REMESHING is turned on, the mesh will be updated automatically, so that the cutplanes visible within MESH VIEW will be current. If this feature is not turned on, the ⚠ icon will indicate the mesh is out-of-date. In this case, press the Automatic Remeshing button and select REMESH NOW to generate the current version of the mesh.

- See Section 5.2 for more on the AUTOMATIC REMESHING feature.

8.3.1 Meshing Modes

There are five available meshing modes in XFdtd. Each is described below. These can be specified in the MESHING PROPERTIES EDITOR or in the PROJECT TREE by right-clicking on the object and choosing GRIDDING / MESHING > MESH TYPE, as seen in Figure 8.19.

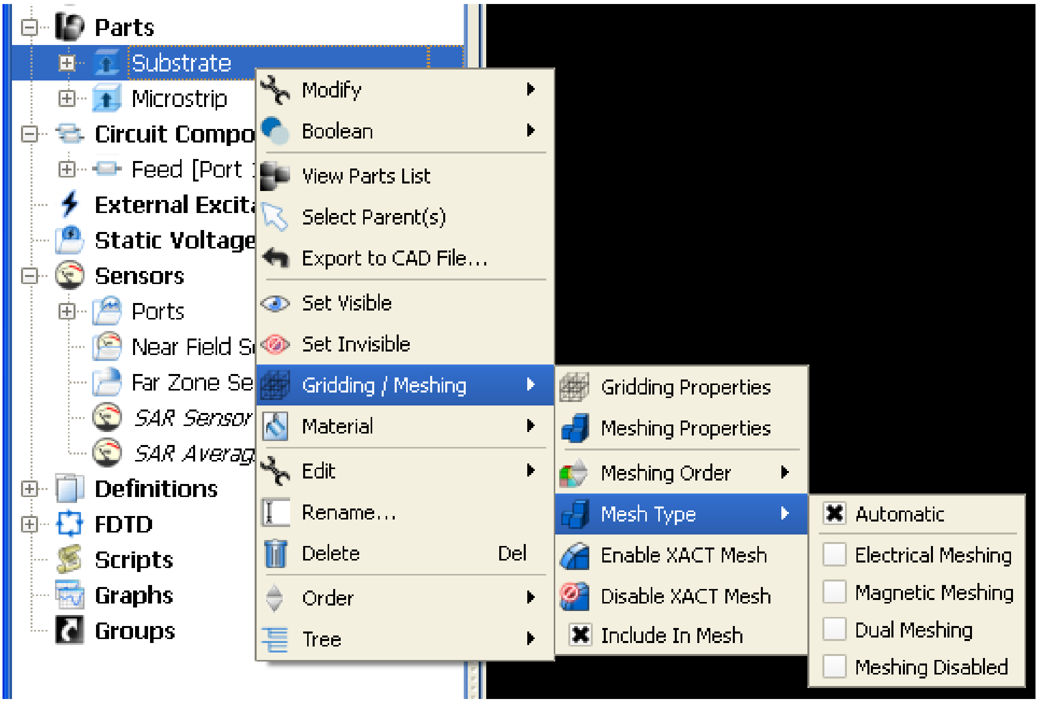


Figure 8.19: Assigning meshing modes within the Project Tree

- **Meshing Disabled.** Select MESHING DISABLED to disable meshing for objects that do not need to be considered during meshing operations. Selecting this option will reduce memory usage, and remove the object from the computation.
- **Electrical Meshing.** Select ELECTRICAL MESHING to include the object in the electric mesh only.
- **Magnetic Meshing.** Select MAGNETIC MESHING to include the object in the magnetic mesh only.
- **Dual Meshing.** Select DUAL MESHING to force the object to be included in both electrical and magnetic meshes.
- **Automatic Meshing.** The AUTOMATIC MESHING mode is selected by default. It enables XFdtd to automatically determine which parts are included/excluded from the electric and magnetic meshes. This determination is based on the electric and magnetic properties of the material assigned to the part.

  When Automatic Meshing is selected:

  - Parts with electric material, other than free space, are included in the mesh
  - Parts with magnetic material, other than free space, are included in the mesh
  - Parts made entirely of free space material are not included in the mesh.

  The following table displays the various meshing modes available in XFdtd.

  To include a part in the mesh where both the ELECTRIC and MAGNETIC material types are set to FREE SPACE, select either the Electric Mesh mode or the Dual Mesh mode.

Table 8.1: Mesh Modes

Mesh Mode	In Electric Mesh	In Magnetic Mesh
Automatic	All parts with non-free space electric material properties	All parts with non-free space magnetic material properties
Electric	Yes	No
Magnetic	No	Yes
Both	Yes	Yes
Disabled	No	No

To select a different mesh mode, right click an object in the ProjectTree and select Meshing Properties or Mesh Type. A window containing the available mesh modes appears.

Under most circumstances it is best to keep the default mode because it enables the software to automatically apply the most memory-efficient meshing option.

8.3.2 Meshing Order

The Meshing Order setting specifies the level of priority that each object has. If objects overlap in a particular region of space, the object with a higher meshing priority will overwrite the other objects. Each object is assigned a default priority of 50. This should be changed to a higher or lower value depending on whether the object should be given more or less consideration, respectively, when meshed. If two overlapping objects have the exact same mesh priority, the material used for mesh edges where the two objects overlap is determined by which object is closest to the top of the ProjectTree .

Choosing Set Priority prompts the user to enter a numerical priority definition.

Additionally, priority can be specified within the Project Tree, as displayed in Figure 8.20. Moving the selected Parts object priority Up or Down increases or decreases its mesh priority, respectively. Likewise, selecting Move to Top assigns that object the highest priority of all of the objects in the project; Move to Bottom will assign it the lowest priority.

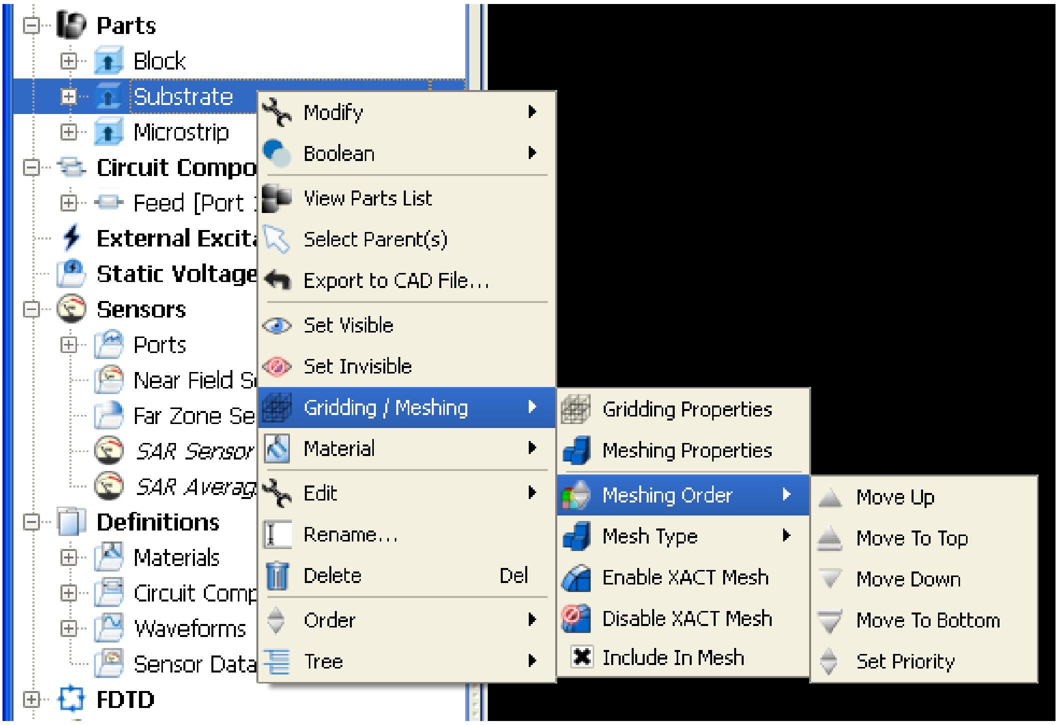


Figure 8.20: Assigning meshing priority in the Project Tree

8.3.3 Include in Mesh

The INCLUDE IN MESH setting specifies whether or not the selected part is considered when XFdtd calculates the mesh. This setting does not affect the mesh mode or mesh order of the part. The benefit is that a user can remove an object from the mesh or add it back in without having to re-set the mesh mode or order associated with the part each time they make a change. This option can be set by right clicking a part and selecting or de-selecting the GRIDDING / MESHING > INCLUDE IN MESH checkbox, or by selecting the MESHING PROPERTIES EDITOR. To add or remove multiple parts from the mesh, select them, right click, and select Include in Mesh or Exclude from Mesh.

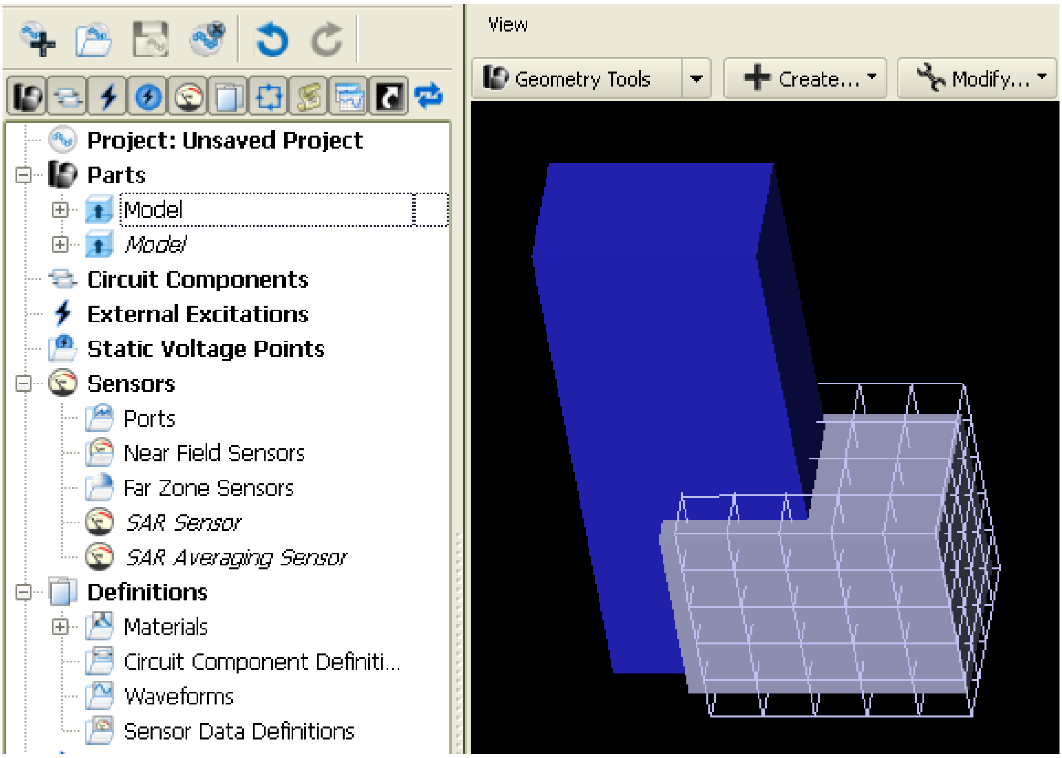


Figure 8.21: Parts included in and excluded from the mesh

8.3.4 XACT: Accurate Cell Technology

As discussed in Chapters 2.2 and 2.7, the cell size used to discretize the domain is frequently limited by the dimensions and features of the geometric structure, and this in turn will determine the runtime and memory costs of the simulation. When it is important to accurately resolve, say, the thickness of a conductor (or of a gap between conductors) or the shape of a highly curved radiating body, this typically means that a very fine grid must be used in the vicinity of the smallest geometric features. Appropriately placed Fixed Points and Grid Regions make this grid refinement straightforward, and ultimately provide more accuracy in simulation results.

However, this increase in simulation accuracy comes at a cost. Increasing the number of cells in each of the X, Y, and Z directions by a factor of N leads to a factor of N^3 increase in memory consumption and up to a N^4 increase in computation time. For this reason, it can be difficult to sufficiently refine the computational grid while keeping your simulations fast and small.

In cases such as these, utilizing XFdtd's XACT Mesh capability can be extremely helpful. XACT is a set of techniques which use geometric information to provide a subcellular discretization of the computational domain, providing a more accurate simulation for a given grid resolution. This can in turn be used to reduce the overall simulation time and memory requirements for a given level of desired accuracy. For example:

- Using XACT, it is possible to represent gaps between conductors in the mesh without necessarily having a full grid cell distance between the conductors. In turn, this may allow you to increase your cell size (and timestep) beyond what is possible with Cartesian mesh edges only, resulting in overall faster simulations.

- Using XACT, curved surfaces can be modeled more accurately than can be done with Cartesian mesh edges only, since the XACT mesh represents the boundary of the object as actually curved inside of a grid cell (see Figure 8.22). Thus, using XACT can improve the accuracy of your simulations for a given grid resolution.

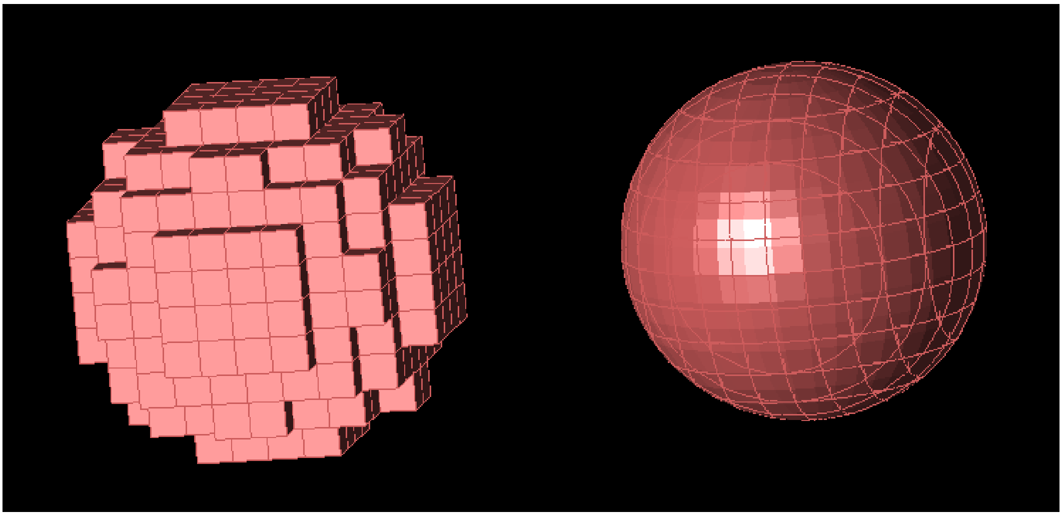


Figure 8.22: Meshed spheres with a conventional mesh (left) and XACT mesh (right)

Much more geometry information is used when computing fields near the boundary of parts with an XACT mesh than with a conventional mesh. As such, execution time and memory usage penalties are incurred when XACT is enabled. In order to avoid these penalties when unnecessary, XACT should be used with care. Here are some guidelines:

- XACT is most beneficial for parts with high conductivity or Perfect Conductor electric material properties.
- XACT is most beneficial on parts with high curvature or where the size of small geometric features is significant.
- Axis-aligned geometry that is already well-represented by a conventional Cartesian mesh will not typically benefit from using XACT.
- XACT is not intended for use on parts with cross-sectional size that is smaller than a single grid cell. In such cases, it may not be possible to generate an XACT mesh that maintains connectivity without enabling Aggressive Surface Meshing as well (see Chapter 8.3.5).

You can enable XACT on a single part from either the part's MESHING PROPERTIES EDTIOR (Chapter 8.3.5) or by choosing ENABLE XACT MESH from the GRIDDING / MESHING submenu of the part's right-click menu in the PROJECT TREE. If an assembly is selected in the PROJECT TREE, the right-click menu's SUBPARTS > GRIDDING / MESHING > ENABLE XACT MESH ON ALL SUBPARTS can be used to similar effect.

In order to mesh a part using XACT, the part's assigned material must be classified as either an electric Perfect Conductor, as a Good Conductor, or as otherwise having low conductivity. This classification is controlled by the nature of the material, and also by the GOOD CONDUCTOR choice in the MATERIAL EDITOR (See Chapter 6.1.3.1). Note that the XACT formulation for parts meshed with Good Conductor materials assumes a minimum of five skin depths thickness to the part at all locations, thus blocking all energy transfer through the part.

When XACT is enabled for any part in the simulation, XFdtd will by default reduce the simulation timestep by as much as 30% in order to maintain numeric stability in the computation. The maximum amount of timestep reduction is controllable from the XACT ACCURACY setting on the ADVANCED tab of the PROJECT PROPERTIES EDITOR. This setting allows you to trade simulation speed against the accuracy of the subcellular representation of geometry. Large values of this setting (near 1.0) will cause the simulation timestep to be reduced very little, thus giving the fastest simulation runs. Small values of this setting (near 0.1) may cause the simulation timestep to be significantly reduced (as low as 10% of its usual value) but will most accurately represent your geometry. The default value of 0.7 should be reasonable in most cases.

Please be aware of the following limitations of the XACT technique in XFdtd:

- XACT can only currently be used with simulations excited by Discrete Sources, Waveguide Interfaces and Total Field Plane Waves. Scattered Field Plane Wave, Gaussian Beam, and Static Voltage Points cannot be used with XACT in XFdtd.
- XACT can only be enabled on parts using nondispersive, isotropic electric materials.
- Some material types cannot exist on or near a part meshed with XACT. These include Dispersive, Thin Wire, and Nonlinear electric materials, and any magnetic material type other than Magnetic Freespace. Furhtermore, when an Averaged Electric Material is encountered nearby a part meshed with XACT, the Dominant Material of the average (See Chapter 8.3.6) will be used in its place.
- XACT-enabled parts cannot come within one cell of the simulation boundary in XFdtd.

In addition, care should be taken when a thin part with an XACT mesh lies less than a grid cell away from a part that is conventionally meshed. As the conventionally meshed part finds the best-fitting set of Cartesian mesh edges to describe its surfaces, it may choose some mesh edges on the opposite side of the XACT meshed part, as shown in Figure 8.23. In such a case, the best recourse may be to enable XACT for both parts, although judicious use of Fixed Points may correct this behavior in some situations.

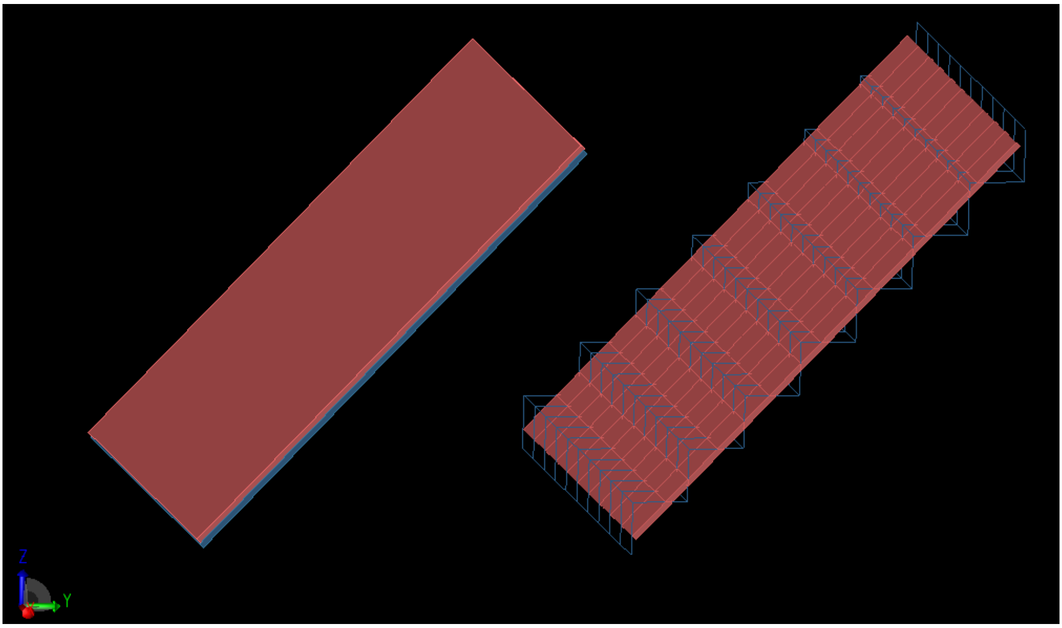


Figure 8.23: The blue plate's conventional mesh has edges on both sides of the red plate's XACT mesh

It should be emphasized that parts with XACT use a more faithfully meshed representation of the geometry than a conventional Cartesian mesh can provide. This can have unexpected consequences in cases where there are small gaps in your parts which a conventional mesh would typically short together. The XACT mesh will never do this. Thus, it is critical when using XACT that the CAD representation of your parts contains no unwanted gaps.

It is also worth noting some limitations of Sensor results near XACT-enabled parts:

- Surface sensors on XACT parts will show results from the staircase mesh of the part (the same as non-XACT parts), not the surface of the part. A typical case is to place a surface sensor on an XACT part to obtain current density (J) in order to capture current on the surface of the part. However the result can be confusing, especially for small, thin parts, since what is captured and subsequently rendered is J on the stair-case mesh surrounding the part. The actual surface current density computation will be included in a future version of XFdtd.
- Near XACT geometry, there are multiple possible places to measure E-Fields on each E-edge or H-Fields on an H-edge. For snapped sensors, E-Field and H-Fields are sampled nearest the E-grid node (this is the case for the previous note). For interpolated sensors, the E-Field nearest the center of the E-edge and the H-Field nearest the center of the H-edge are sampled.
- When you use an interpolated surface sensor, the results are rendered in a way that do not appear staircased, but the field values displayed are interpolated from field values captured at the centers of staircase mesh edges.
- For a snapped sensor, field components that are actually located some distance away from the node are co-located at an E-grid node. E.g. The Ex component reported at a grid node is actually measured a half cell away in the +X direction and the Hx component reported there is measured a half cell away in the +Y and +Z directions. This may add some confusion, because depending on

what side of the body the sensor is attached to, the displayed field components may be from the other side of the body.

8.3.5 Solid Meshing Options

Under most circumstances, the solid meshing options are not necessary. These options may be advantageous in projects that have small features that are not adequately considered by default meshing operations. Figure 8.24 shows the solid meshing options in the MESHING PROPERTIES EDITOR.

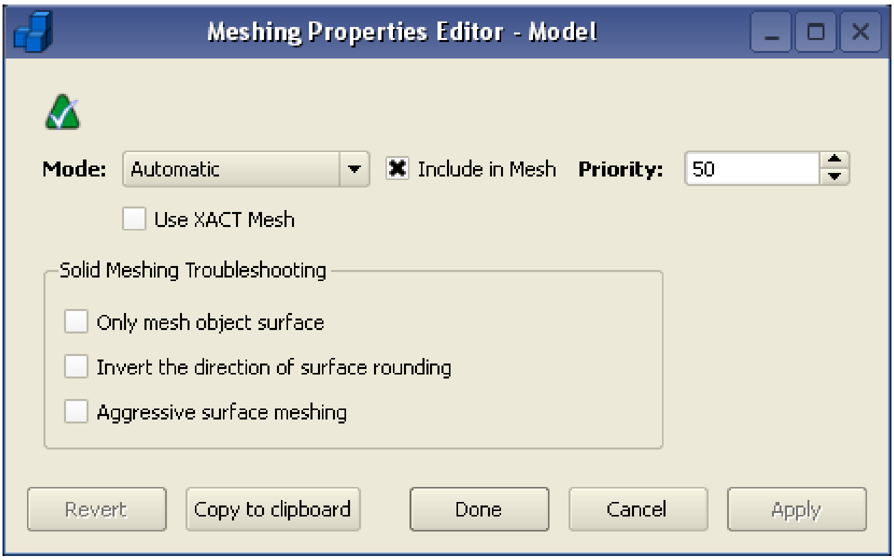


Figure 8.24: The Meshing Properties Editor for a solid model part

Only mesh object surface

The ONLY MESH OBJECT SURFACE option meshes the shell of the physical object rather than its entire volume.

Invert the direction of surface rounding

If the edge of an object falls in the center of a cell rather than at a cell edge, XFdtd's default behavior is to round the object away from the center of the object. Selecting INVERT THE DIRECTION OF SURFACE ROUNDING will invert this behavior and round the edge inward, towards the center of the object. This usually is not necessary but may be useful in cases where an object is very close to another. In this instance, rounding a surface outward can cause the objects' edges to touch and result in a short circuit that may cause inconsistent results.

Aggressive surface meshing

AGGRESSIVE SURFACE MESHING should only be enabled for an object that is very small compared to the projects' cell size. This option will create a more detailed mesh around smaller features to ensure that they are surrounded with cell edges. This would be necessary for small feature whose electrical conductivity may be compromised without detailed meshing.

8.3.6 Volume Meshing Options

Several additional settings are available when assigning mesh properties for volumetric parts, such as voxels. Figure 8.25 shows the volume meshing options in the MESHING PROPERTIES EDITOR.

Figure 8.25: The Meshing Properties Editor for a volumetric part

Volume meshing method

There are two VOLUME MESHING METHOD algorithms available for converting the volumetric data to cell edge data. Both methods use an intersection method to determine how much of a given volume is contained inside of a cell edge. Therefore, it is applied to every volume that intersects the cell edge. The data is combined to measure the percentage that each material fills in each cell edge. These percentage fill values are in increments of a half of a percentage point. This information is then used based on which algorithm is selected.

- The first method, refered to as the DOMINANT MATERIAL meshing algorithm, evaluates the list of percentage fill values, and then chooses the material that has the largest percent fill to apply as the final material for that cell edge.
- The second method, refered to as the AVERAGED MATERIAL meshing algorithm, uses the percent fill values for each cell edge to generate an "averaged material." These averaged materials are maintained in a separate list from the original materials of the geometry.

Degradation level

The DEGRADATION LEVEL is available in the AVERAGED volume meshing method. It dictates how non-averaged materials intersecting the volume of the cell edge are to be grouped and considered for averaging, by defining the size of a set of bins that represent a percentage of the total cell edge volume. For example, a degradation level of 4 subdivides the total volume (100%) by 4, meaning each bin represents 25% of the volume. Therefore, for consideration in the averaged material, a non-averaged material must fill at least half of the bin, or 12.5% of the total volume. By defining a course resolution for material consideration we are able to condense the number of averaged materials, keeping memory and calculation speed reasonable while providing a user-defined level of accuracy.

Volume calculation method

The VOLUME CALCULATION METHOD determines the accuracy of the rendered volume. It directly relates to the DEGRADATION LEVEL, as it determines the amount of material volume that fills a cell. APPROXIMATE is sufficient for most calculations, and runs most efficently. EXACT is a better choice when it's important to resolve structures that are changing rapidly within the size of a cell. By definition, it is more accurate but also requires more memory.

8.4 Touching Objects

The FDTD mesh is a discretization of a continuous 3-D space into cells of user-specified size. The process of transferring these geometric representations into this discretized space (meshing) can lead to unintended artifacts. One common situation that arises is that two objects in the geometry that are very close together, but not touching, end up intersecting in the mesh. Such an artifact can lead to inaccurate results due to a short-circuiting effect in the calculation. The user traditionally has two options in this case. The first is to move the two objects further apart, which may not be possible given the design of the geometry. The other is to create cells small enough in that region to resolve the separation between the objects, which may lead to an extremely large number of cells and a dramatic increase of computation resources required to run the simulation.

XFdtd provides a third option to alleviate the work required by the user. This is the role of the TOUCHING OBJECTS functionality. There are two tools available for the user to identify and correct any unintended artifacts caused by meshing or geometric positioning.

- The first, TEST OBJECTS IN CAD SPACE, tests two objects for a CAD space intersection. It will report to the user if two objects are actually touching in non-discretized space.
- The second, SEPARATE OBJECTS IN MESH, allows the user to create a relationship between two objects that will ensure they are separated in the final mesh by a user-defined number of cells.

These tools allow the user to maintain a separation while not sacrificing computational complexity or geometric design.

8.4.1 Test Objects in CAD Space

In order to find out whether two objects are actually touching, XFdtd provides the TEST OBJECTS IN CAD SPACE function to test the selected objects. This can be done by holding CTRL and selecting two objects in either the GEOMETRY workspace window or under the PARTS branch of the PROJECT TREE. Then test the objects by selecting TOUCHING OBJECTS > TEST OBJECTS IN CAD SPACE, as shown in Figure 8.26.

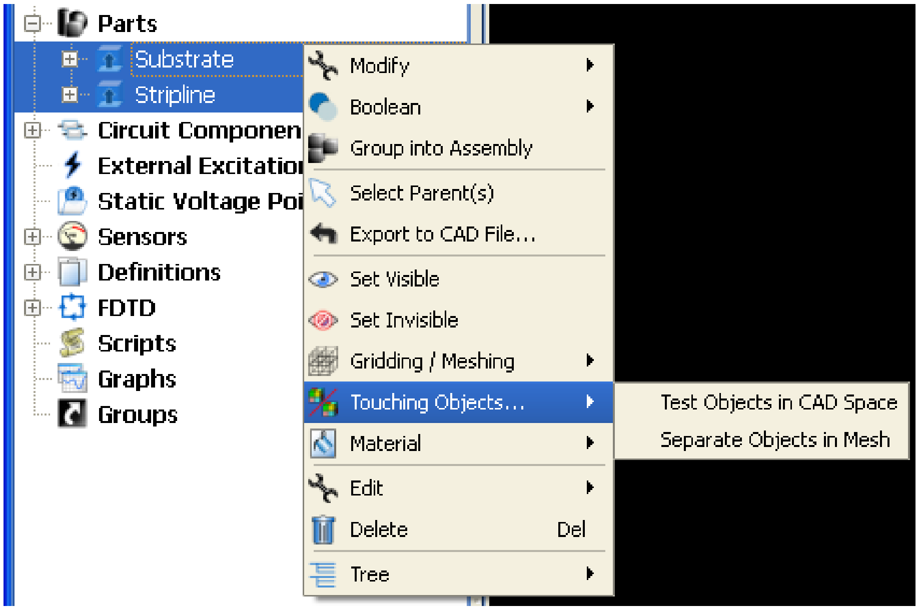


Figure 8.26: The Touching Objects menu

A dialogue will appear with the results. The user will also have the option of ensuring that the objects are separated in the mesh, as seen in Figure 8.27. If YES is selected, the SEPARATE OBJECTS IN MESH dialog will appear, as described in the following section.

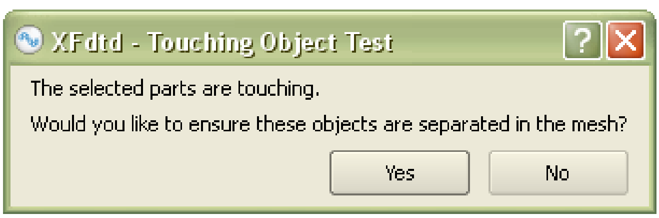


Figure 8.27: The Touching Object Test dialog

8.4.2 Separate Objects in Mesh

Defining a separation operation between two objects ensures that a gap of user-defined size is maintained despite discretization or geometric positioning. To perform this operation, select two objects and navigate to TOUCHING OBJECTS > SEPARATE OBJECTS IN MESH, (seen in Figure 8.26 above). The dialog seen in Figure 8.28 will appear.

- The PRIMARY MODEL is separated from the SECONDARY MODEL. Ensure they are identified appropriately. Pressing SWAP INPUTS will reverse their labels.
- Define the SEPARATION DISTANCE (CELLS) between the two objects. This is the number of cells that is "taken away" from the SECONDARY MODEL in the meshing operation.

- Specify the FILL MATERIAL to use, if any. If no material is specified then the material existing in the mesh under the removed object will be used.
  - ✓ This is a very powerful concept, as it means that any complex configuration of materials may exist in the gap between two objects, and that information is not lost by performing the operation.

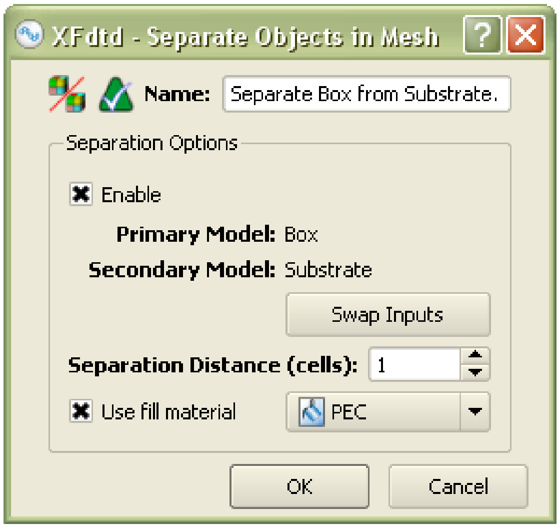


Figure 8.28: The Separate Objects in Mesh dialog

To illustrate this concept, consider the case of two touching cubes made of different materials. A mesh separation of 2 cells and a different (red) material is applied, with the smaller (purple) cube as the PRIMARY MODEL and the larger (blue) cube as the SECONDARY MODEL. Though the geometry is unaffected, the mesh separation is visible when viewing the geometry in MESH VIEW, as seen in Figure 8.29.

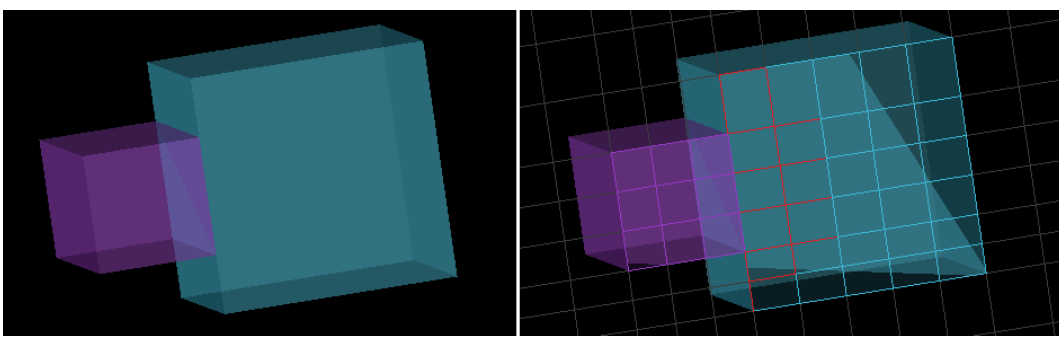


Figure 8.29: Touching objects geometry (left) and mesh view (right) with a 2-cell mesh separation applied

Once this relationship is specified it will be maintained and updated appropriately in the mesh and nothing further is required by the user. If the user decides to delete one of the operands, the operation will be automatically removed; and if the user re-orients or changes materials on the operands, the operation will reflect that appropriately. The mesh separation is managed from the FDTD > MESH branch of the PROJECT TREE.

Chapter 9

Defining Outer Boundary Conditions

In this chapter, you will learn...

→ how to distinguish absorbing and reflecting boundaries in XFdtd

→ how to choose which boundary type to use for your project calculation

Specifying an outer radiation boundary is necessary to indicate how the calculation treats the boundaries of the problem space. During an XFdtd calculation, the fields updated at every cell location are dependent upon the neighboring fields. However, due to memory limitations, the fields on the outer edges of the grid cannot be updated correctly because the grid must be a finite size. To correct this situation, outer radiation boundary conditions are applied at the edges of the XFdtd grid. Thus, the performance of the outer boundaries is a significant factor in the accuracy of the calculation, and it is important to use them correctly. This chapter details several available options for defining the outer boundaries of an XFdtd project.

Outer boundaries are defined in the OUTER BOUNDARY EDITOR, located in the FDTD: OUTER BOUNDARY branch of the PROJECT TREE, as seen in Figure 9.1.

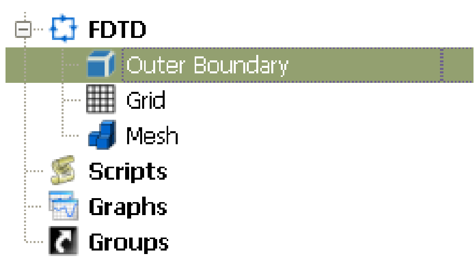


Figure 9.1: Outer Boundary Branch

9.1 Absorbing Boundaries vs Reflecting Boundaries

The outer radiation boundary is a method for absorbing fields propagating from the XFdtd grid toward the boundary. By absorbing these fields, the grid appears to extend infinitely; however, it is actually finite in order to fall within reasonable memory usage. There are two numerical absorbers designed to allow electromagnetic fields radiated or scattered by the FDTD geometry to be absorbed with very little reflection from the boundary. These include a PERFECTLY MATCHED LAYER (PML) and a second-order, stabilized LIAO radiation boundary.

In some cases a reflecting boundary rather than an absorbing one is preferred. A perfectly conducting boundary (either electric, PEC, or magnetic, PMC) may be used in these cases, for example, to provide a ground plane, or to image the fields in an XFdtd calculation.

The Liao and PML boundaries may not be mixed together in the same calculation. Furthermore, PML may not be used with the PMC boundary. The Liao boundary may be used with both PEC and PMC boundaries.

The default boundary condition for XFdtd is PML.

In addition, XFdtd has PERIODIC boundary conditions that allow periodic structures to be modeled. These boundary conditions equate the corresponding outer surfaces of the mesh.

Figure 9.2 shows the OUTER BOUNDARY EDITOR.

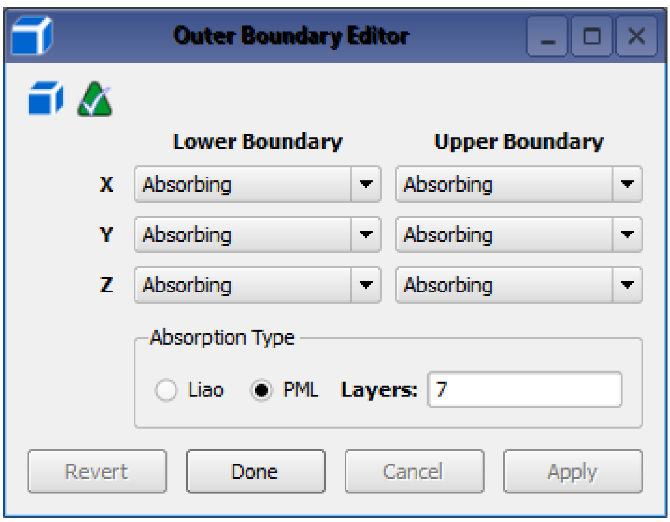


Figure 9.2: The Outer Boundary Editor

9.2 Liao Boundary

The LIAO outer boundary condition is an estimation method, which is makes it fundamentally different from PML boundary conditions. By looking into the FDTD space and back in time, it estimates the electric fields just outside the limits of the FDTD mesh. These estimated values are then used in the FDTD equations inside the space. The Liao estimation assumes that waves are allowed to travel outward from the space but not reflect back in. This method works well provided that there is enough space between the radiating geometry and the outer boundary. Typical limits are at least 10 cells of spacing to ensure that instability does not occur.

- For more on calculation instability, see Section 11.3.5.

A homogeneous dielectric may be located against the Liao boundary. For example, in a lossy earth or strip line calculation, the earth or dielectric layer may touch the outer boundary. Liao will usually function well in this situation provided that there are no air gaps within five cells of the Liao boundary. Liao assumes homogeneous material within five cells, and if this is not the case then the XFdtd calculation will usually be unstable with rapidly rising field amplitudes.

Since Liao is an estimation method, the size of the FDTD mesh is not increased by using it. Some storage is needed for saving electric values at previous timesteps, but this is usually negligible in a typical calculation.

9.3 PML Boundary

The PERFECTLY MATCHED LAYER (PML) boundary condition is offered as an alternative to Liao. PML is an artificial absorbing material that absorbs the incident energy as it propagates through the PML layers. Better absorption, that is, smaller reflection, is obtained by adding more layers at the expense of increasing the size of the FDTD mesh. For example, consider an XFdtd calculation on a mesh using the Liao absorber that is 50 x 60 x 70 cells or a total of 210,000 cells. There is a 15 cell free space border all around the geometry so that the Liao boundaries can provide small reflections. If the Liao is changed to seven PML layers, the geometry mesh will not change. However, outside of this defined mesh region, seven additional FDTD mesh layers are added on each side of the geometry. This means that the actual number of FDTD cells that must be calculated grows to 64 x 74 x 84 or 397,824 cells, almost double. This, coupled with the fact that PML cells require more arithmetic operations than normal cells, means that there is a computational penalty for using the PML boundary condition over Liao.

However, the benefit of using the PML layers is that they provide better absorption than Liao even with only a five-cell border of free space, and perhaps only six PML layers would provide this. In such a situation, calculation time would be saved. Making this comparison would require meshing the object again with a smaller free space margin to the outer boundary. This can be done easily in XFdtd using the mesh tab and choosing a smaller padding around the geometry.

Both PML and Liao boundary conditions are offered to provide flexibility. Both methods should provide similar results when properly used, although in most cases, particularly when low frequencies (compared to the cell size) are used, PML is superior. It is recommended that PML boundary conditions are used whenever possible and therefore this is the default boundary condition in XFdtd.

9.4 PEC Boundary

In some situations, terminating one or more faces of the FDTD geometry space with a PERFECT ELECTRIC CONDUCTOR (PEC) outer boundary is advantageous. For example, the conducting ground plane of a microstrip could be located on one face of the FDTD space.

If all of the outer boundaries of the calculation are not absorbing, a PLANEWAVE should not be used to excite the calculation and the far-zone transformations will not provide correct results for far-zone fields.

The sole exception is in the case of one PEC boundary and five absorbing boundaries, which will compute far zone over infinite PEC ground.

- An edge of the FDTD space should be set to PEC using the PEC Boundary Condition. **Do not** set FDTD cells to PEC material in the geometry *and* set the outer boundary to absorbing, as this will cause instabilities in the calculation.

9.5 PMC Boundary

The PERFECT MAGNETIC CONDUCTOR (PMC) outer boundary condition may be useful in reducing the size of the FDTD mesh, memory requirements, and calculation time by taking advantage of symmetries in the geometry. For example, this condition would be a good choice in a symmetric problem space where magnetic fields are strictly normal to a plane.

9.6 Periodic Boundary

Similar to the PMC boundary condition, the PERIODIC boundary condition may be useful in taking advantage of geometry/field symmetry to reduce the size of the FDTD mesh and therefore the memory and calculation time required. In this case the upper and lower edges of the mesh are forced to be equal during the analysis. This may be useful for cases when small geometries are repeated over and over (i.e. optics examples).

- See Section 11.1.1 for more about using the periodic boundary with PLANEWAVE excitations.

Chapter 10

Saving Output Data with Sensors

In this chapter, you will learn...

→ how to use sensors to save the results of your XFdtd project calculation

→ how to choose the correct sensor to use depending on the type of data you want to save

SENSORS are objects that save data during a simulation. Any type of data that can be saved in XFdtd will be saved with a sensor. The type of data that is saved by a sensor is dependent on the sensor type, as well as the specific data that is requested within various sensor editors. There are various types of sensors that are available within XFdtd, including:

- SYSTEM sensors
- PORT sensors
- NEAR FIELD sensors, including:
  - POINT sensors
  - SURFACE sensors
  - RECTANGULAR sensors
  - PLANAR sensors
  - SOLID PART sensors
  - SOLID BOX sensors
- FAR ZONE sensors
- HEARING AID COMPATIBILITY (HAC) sensors
- SPECIFIC ABSORPTION RATE (SAR) sensors

Result objects are generated based on the sensor objects that are defined in the project.

After a sensor is placed, an editor is used to define its characteristics based on the output data that it is to retrieve. Each type of sensor has its own respective editor window, and is enabled by default. (Disabling the sensor will prevent it from collecting data from the calculation engine.) This chapter details the process of adding sensors to a project and requesting specific results with each type of sensor.

10.1 Sensor Tools

In general, sensors added to the project within the SENSOR TOOLS dialog (with the exception of SYSTEM, PORT and SAR sensors). There are two ways to open this dialog. The first is to choose SENSOR TOOLS from the drop-down list in the GEOMETRY workspace window. The second is to right-click on the SENSORS branch of the PROJECT TREE and the desired sensor type.

- SYSTEM sensors are automatically added to every XFdtd project, as described in Section 10.4.
- Section 10.5 briefly describes PORT sensors and the data the collect. They are added by setting a component property. See Section 7.1.1 for details.
- SAR sensors are added by double-clicking on the sensor in the PROJECT TREE. See Section 10.6 for more on SAR sensors.

Figure 10.1 shows the SENSOR TOOLS menu.

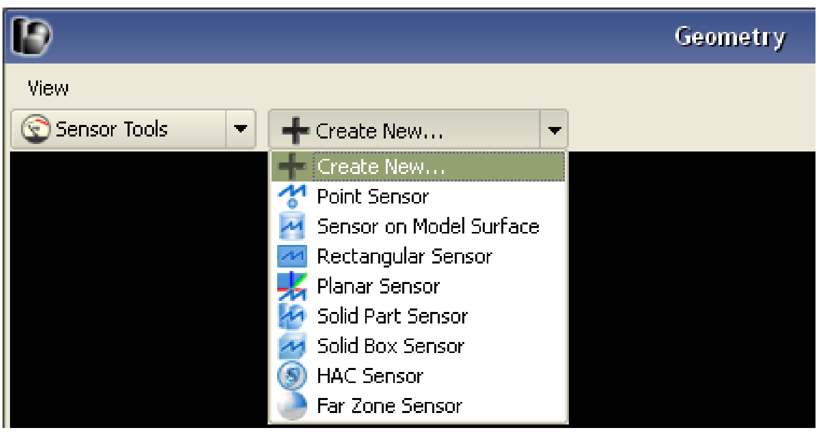


Figure 10.1: The Sensor Tools menu

The following sections detail the process of adding each type of sensor within SENSOR TOOLS, defining its associated characteristics, and requesting the desired output data to be calculated by the sensor.

10.2 Near Field Sensors

NEAR FIELD sensors save time-domain and/or frequency-domain near zone field quantities at specific points, surfaces or volumes within the bounds of calculation space. In general, field data is retrived using the POINT, SURFACE, or SOLID sensors, and hearing aid field values are recorded using the

HEARING AID COMPATIBILITY (HAC) sensor. Solid sensor results can be viewed as 2-D plots, and both Solid and Surface sensors can be viewed as 3-D field sequences (excluding Point sensor data).

- See Section 12.3.1 and Section 12.4.1, respectively, for more on viewing 2-D and 3-D results.

10.2.1 Field Retrieval

The field quantities of X-, Y-, and Z-directed electric (E) and magnetic (H) fields may be saved at a specific point, across a surface, or throughout a volume with a POINT sensor, SURFACE sensor, or SOLID sensor, respectively. Additionally, X-, Y-, and Z-directed conduction current density (J) may be collected with any of these sensors. When a near-zone source is used as the input, the total field values are available. With an incident PLANEWAVE input, the scattered and total electric and magnetic fields may be saved in addition to the total current density.

Samplings of near field data may be saved by specifying SAMPLING TIME RANGE in any of the near field sensor definition windows. Near field data will be collected in specific planes of the geometry during the XFdtd calculation at every interval specified within the definition. A field file containing the electric and magnetic fields and the current will be created for each timestep specified. For example, setting an entry beginning at timestep 100, ending at timestep 1000, with an increment of 100 will create 10 field files which may be viewed as a movie after the XFdtd calculation is performed.

Be aware of the number of field slices to save, as they can store enormous amounts of data. Single field files may contain megabytes of data depending on the number of cells in the specified plane.

10.2.1.1 Point Sensors

A POINT SENSOR is positioned at a specific point-location in the simulation space, and can be defined by the location of a specific vertex in a part object or by a Cartesian 3-D expression. The sensor records data as it occurs at the specified point in space.

Point sensors record data by means of field interpolation or geometric "snapping". When using the interpolated sampling method, the field components are interpolated to the exact location of the point sensor. This is performed by linear interpolation among the surrounding eight appropriate field value sample points (e.g. when measuring E_x, the eight surrounding X-directed *edge centers* are used for the interpolation, and when measuring H_x, the eight surrounding X-directed *cell face center points* are used for the interpolation). When using the snapped sampling method, the location of the point sensor is snapped to the nearest E-grid cell vertex. Field components for snapped point sensors come from the cell whose lowest-index corner is defined by the snapped location of the sensor. The sensor location is thus dependent on the grid definition.

Point Sensor Properties

To define a POINT SENSOR, open the point sensor properties dialog under SENSOR TOOLS. In the LOCATION tab, define the sensor location manually by typing in its coordinates, or automatically by clicking on the intended location in the simulation space with the SELECTION tool. In the PROPERTIES tab, enter the name of the sensor, select the desired POINT SENSOR DEFINITION, choose the sampling method (as described above), and enable the sensor.

Figure 10.2 shows the POINT SENSOR properties dialog.

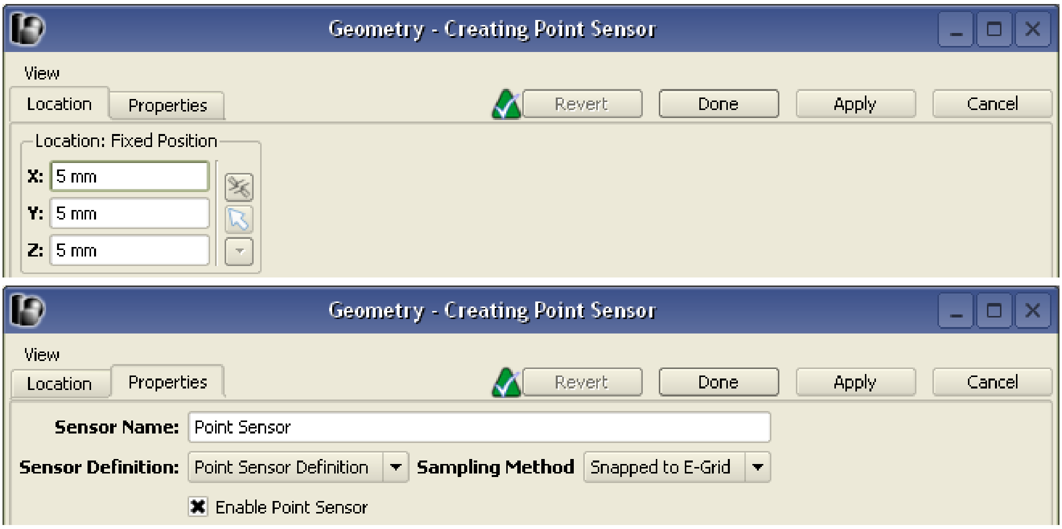


Figure 10.2: Point Sensor properties dialog

Point Sensor Definition Editor

The POINT SENSOR DEFINITION EDITOR is used to create definitions for POINT SENSORS.

To access the editor, double-click on an existing POINT SENSOR DEFINITION in the DEFINITIONS: SENSOR DATA DEFINITIONS branch of the PROJECT TREE. If no point sensor definition is present, right-click on this branch and select NEW POINT SENSOR DEFINITION.

In the FIELDS TO SAVE region of the editor, select the desired point sensor output data to save:

- E: Electric Field Intensity time
- H: Magnetic Field Intensity vs time
- B: Magnetic Flux Density vs time
- J: Current Density vs time
- SCATTERED E: Scattered Electric Field vs time
- SCATTERED H: Scattered Magnetic Field vs time
- SCATTERED B: Scattered Magnetic Induction Field vs time

  Scattered field values can be retrieved only if a GAUSSIANBEAM or a scattered field PLANEWAVE external excitation is used to excite the simulation.

Figure 10.3 shows the POINT SENSOR DEFINITION EDITOR.

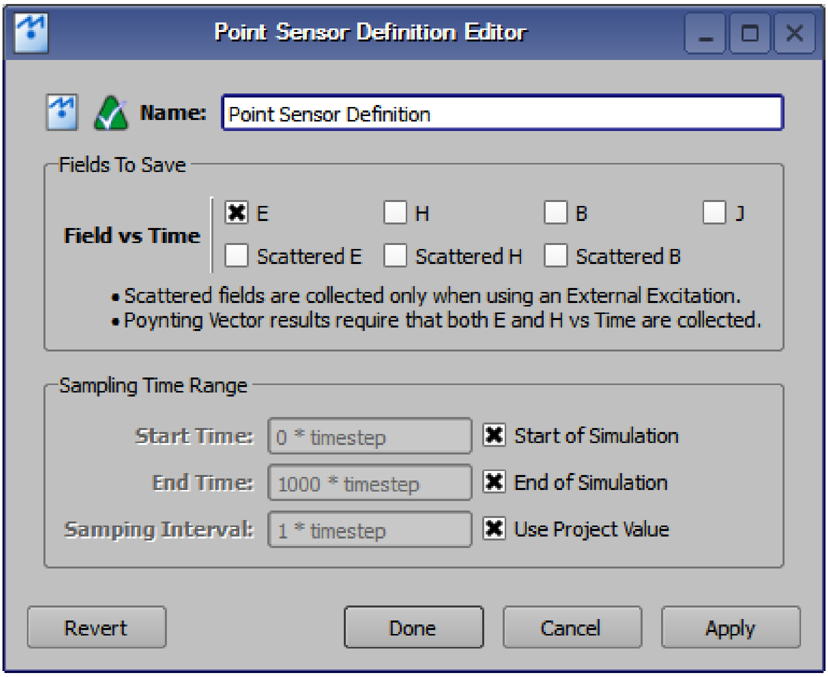


Figure 10.3: Point Sensor Definition Editor

Define the SAMPLING TIME RANGE by entering the START TIME and END TIME, or by simply checking START OF SIMULATION and END OF SIMULATION to automatically assign the sampling time range to these values. The USE PROJECT VALUE option defines the Sampling Interval using the project TIME DOMAIN DATA SAMPLING INTERVAL. Alternatively, the SAMPLING INTERVAL can be used to indicate how often data is saved within this time range.

10.2.1.2 Surface Sensors

Surface sensors collect data on one or more faces of a geometric object in the simulation space. Like POINT SENSORS, they can be interpolated or mesh-snapped.

There are three types of surface sensors in XFdtd:

- SENSOR ON MODEL SURFACE
- RECTANGULAR SENSOR
- PLANAR SENSOR

▶ See Section 10.2.1.2 to reference the output data that can be retrieved by a surface sensor after it has been created within SENSOR TOOLS.

Sensor on Model Surface Properties

To define a SENSOR ON MODEL SURFACE, select the object in the simulation space that the sensor will be attached to by clicking on it in the PICK MODEL tab. In the PICK FACES tab, select the specific face to attach the surface sensor. Finally, in the PROPERTIES tab, assign the new sensor a NAME, DEFINITION and SAMPLING METHOD, and enable the sensor.

✓ All faces of the object chosen in the PICK MODEL tab are selected by default. Skip the PICK FACES tab if you don't need to single out a specific one.

DEFINITION, as mentioned here and in the following two sensor descriptions, refers to definitions stored in the DEFINITIONS: SENSOR DATA DEFINITIONS branch of the PROJECT TREE.

Figure 10.4 shows the SENSOR ON MODEL SURFACE properties dialog.

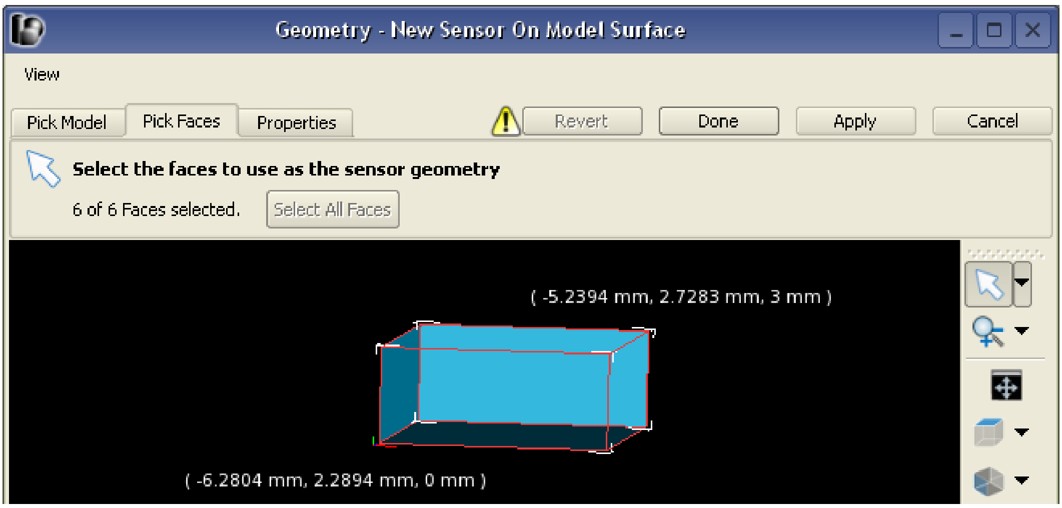


Figure 10.4: Sensor on Model Surface properties dialog

Rectangular Sensor Properties

Define a RECTANGULAR SENSOR by first using the ORIENTATION tab to choose the plane in which the rectangle is defined. Then, use the RECTANGLE tab to define the rectangle's two opposite corners. Finally, under the PROPERTIES tab, assign the sensor a NAME, DEFINITION and SAMPLING METHOD, and enable the sensor.

▶ See Section 5.5 for an explanation of the ORIENTATION tab.

Figure 10.5 shows the RECTANGULAR SENSOR properties dialog.

Figure 10.5: Rectangular Sensor properties dialog

Planar Sensor Properties

The PLANAR SENSOR uses a point and normal direction defined in the ORIENTATION tab to define an entire plane (within the boundaries of the simulation space) to collect sensor data. Select the PROPERTIES tab and assign the sensor a NAME, DEFINITION and SAMPLING METHOD, and enable the sensor.

Figure 10.6 shows the PLANAR SENSOR properties dialog.

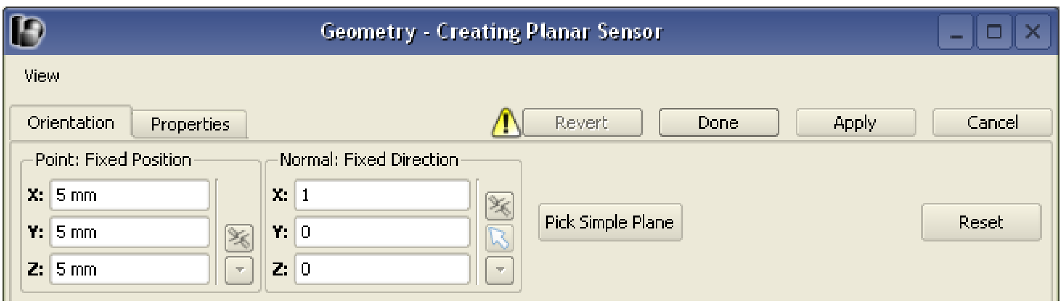


Figure 10.6: Planar Sensor properties dialog

Surface Sensor Definition Editor

The SURFACE SENSOR DEFINITION EDITOR window is used to assign definitions associated with a SURFACE SENSOR.

To access the editor, double-click on an existing SURFACE SENSOR DEFINITION in the DEFINITIONS: SENSOR DATA DEFINITIONS branch of the PROJECT TREE. If no surface sensor definition is present, right-click on this branch and select NEW SURFACE SENSOR DEFINITION.

In the FIELDS TO SAVE area of the editor, select the desired surface sensor output data to save:

- E: Electric Field Intensity time
- H: Magnetic Field Intensity vs time
- B: Magnetic Flux Density vs time
- J: Current Density vs time

- SCATTERED E: Scattered Electric Field vs time
- SCATTERED H: Scattered Magnetic Field vs time
- SCATTERED B: Scattered Magnetic Induction Field vs time

  Scattered field values can be retrieved only if a GAUSSIANBEAM or a scattered field PLANEWAVE external excitation is used to excite the simulation.

- STEADY E: Steady Electric Field vs frequency
- STEADY H: Steady Magnetic Field vs frequency
- STEADY B: Steady Magnetic Induction Field vs frequency
- STEADY J: Steady Current Density Field vs frequency
- DISSIPATED POWER DENSITY: Volumetric dissipated power density vs frequency

Figure 10.7 shows the SURFACE SENSOR DEFINITION EDITOR.

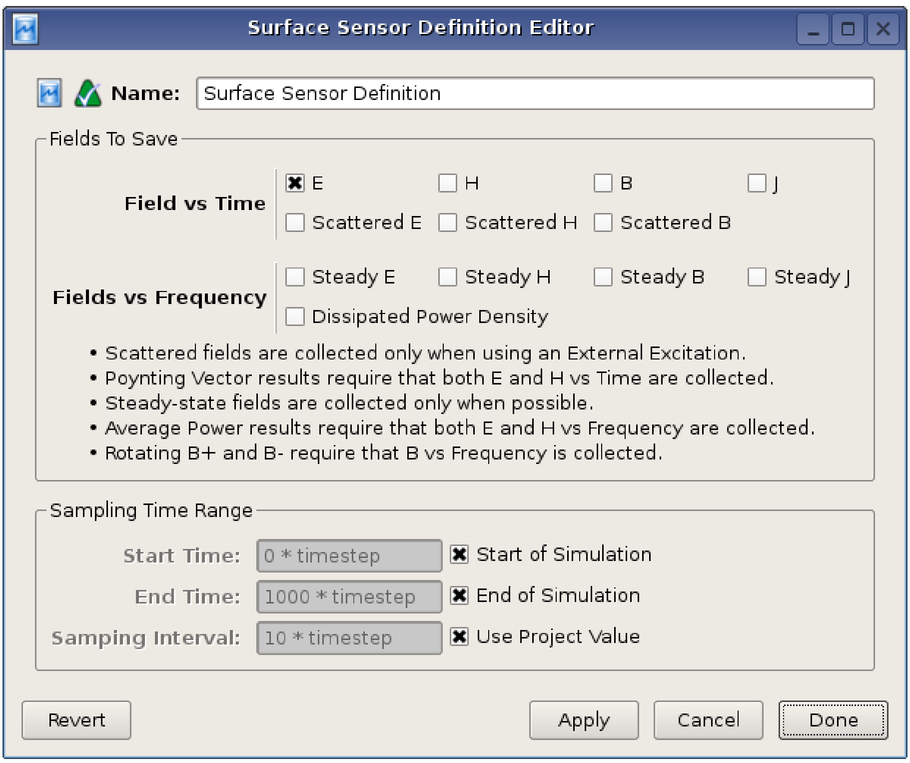


Figure 10.7: Surface Sensor Definition Editor

Define the SAMPLING TIME RANGE by entering the START TIME and END TIME, or by simply checking START OF SIMULATION and END OF SIMULATION to automatically assign the sampling time range to these values. The USE PROJECT VALUE option defines the Sampling Interval using the project TIME DOMAIN DATA SAMPLING INTERVAL. Alternatively, the SAMPLING INTERVAL can be used to indicate how often data is saved within this time range.

10.2.1.3 Solid Sensors

Solid sensors collect data by capturing mesh-snapped fields within a volumetric space (interpolated data is not available).

There are two types of solid sensors in XFdtd:

- Solid Part Sensor
- Solid Box Sensor

▸ See Section 10.2.1.3 to reference the output data that can be retrieved by a solid sensor after it has been created within Sensor Tools.

Solid Part Sensor Properties

A Solid Part Sensor simply assumes the shape of the part that is selected in the Pick Model tab. Enable the sensor and assign it a Name and Definition in the Properties Tab.

Definition, as mentioned here and in the following sensor description, refers to definitions stored in the Definitions: Sensor Data Definitions branch of the Project Tree.

Figure 10.8 shows the Solid Part Sensor properties dialog.

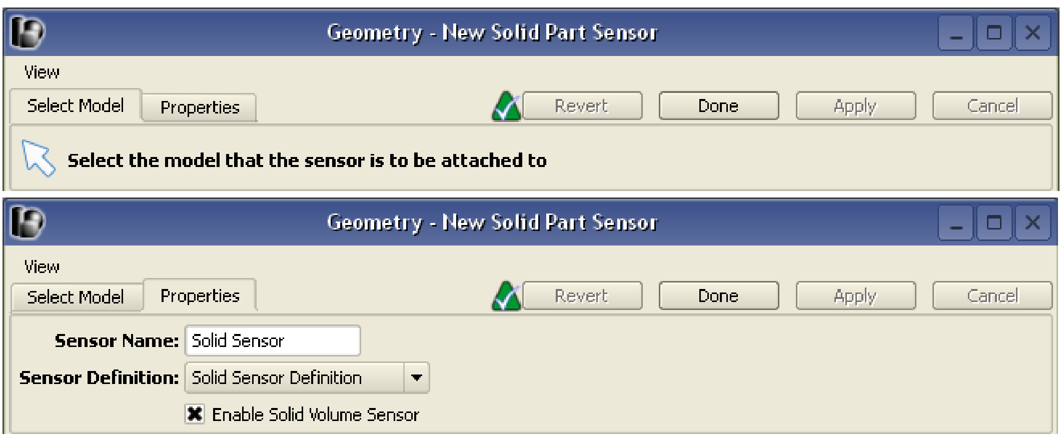


Figure 10.8: Solid Part Sensor properties dialog

Solid Box Sensor Properties

A Solid Box Sensor assumes the shape of a 3-D box. This shape is dictated by the Origin location defined in the Orientation tab, and its farthest corner is defined in the Opposite Corner Tab. Enable the sensor and assign it a Name and Definition in the Properties Tab.

▸ See Section 5.5 for an explanation of the Orientation tab.

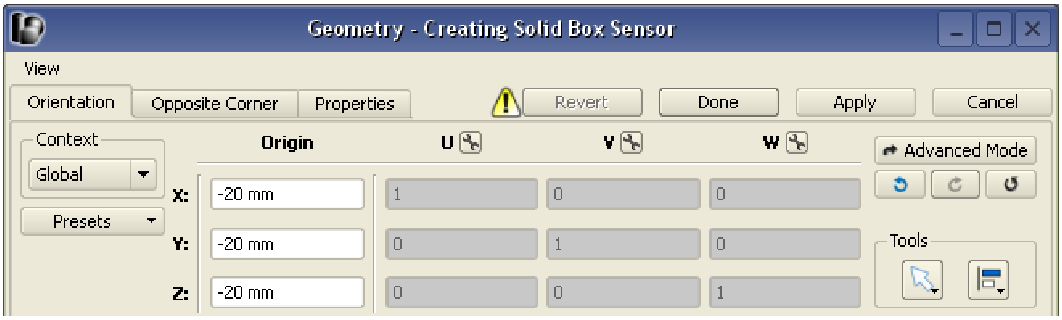


Figure 10.9: Solid Box Sensor properties dialog

Figure 10.9 shows the Solid Box Sensor properties dialog.

Solid Sensor Definition Editor

The SOLID SENSOR DEFINITION EDITOR is used to create a definition for a SOLID SENSOR.

To access the editor, double-click on an existing SOLID SENSOR DEFINITION in the DEFINITIONS: SENSOR DATA DEFINITIONS branch of the PROJECT TREE. If no solid sensor definition is present, right-click on this branch and select NEW SOLID SENSOR DEFINITION.

In the FIELDS TO SAVE area of the editor, select the desired solid sensor output data to save:

- E: Electric Field Intensity time
- H: Magnetic Field Intensity vs time
- B: Magnetic Flux Density vs time
- J: Current Density vs time
- SCATTERED E: Scattered Electric Field vs time
- SCATTERED H: Scattered Magnetic Field vs time
- SCATTERED B: Scattered Magnetic Induction Field vs time

  Scattered field values can be retrieved only if a GAUSSIANBEAM or a scattered field PLANEWAVE external excitation is used to excite the simulation.

- STEADY E: Steady Electric Field vs frequency
- STEADY H: Steady Magnetic Field vs frequency
- STEADY B: Steady Magnetic Induction Field vs frequency
- STEADY J: Steady Current Density Field vs frequency
- DISSIPATED POWER DENSITY: Volumetric dissipated power density vs frequency

Figure 10.10 shows the SOLID SENSOR DEFINITION EDITOR.

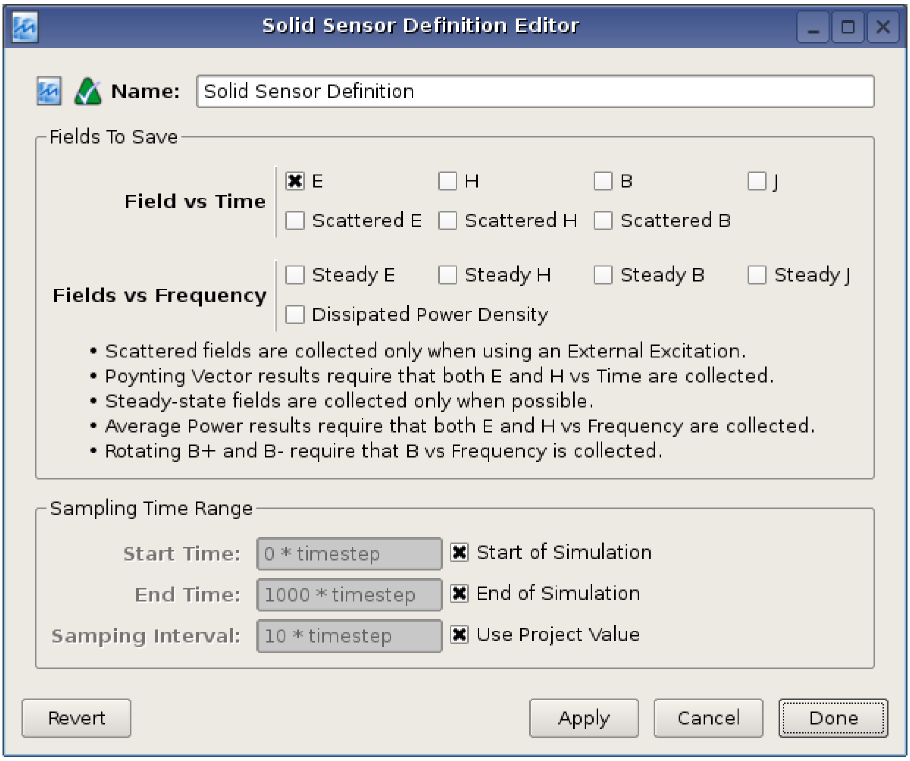


Figure 10.10: Solid Sensor Definition Editor

Define the SAMPLING TIME RANGE by entering the START TIME and END TIME, or by simply checking START OF SIMULATION and END OF SIMULATION to automatically assign the sampling time range to these values. The USE PROJECT VALUE option defines the Sampling Interval using the project TIME DOMAIN DATA SAMPLING INTERVAL. Alternatively, the SAMPLING INTERVAL can be used to indicate how often data is saved within this time range.

10.2.2 Hearing Aid Compatibility Sensors

HEARING AID COMPATIBILITY (HAC) sensors gather data on a 5cm by 5cm arbitrarily-oriented rectangle in free space. They are used to determine if a wireless device (such as a cellphone) will generate electrical and magnetic fields large enough to interfere with a hearing aid. In these cases, they are useful for evaluating the wearer's ability to adjust the position of the phone to a better location.

The HAC sensor is centered at the origin of the coordinate system described in the GEOMETRY tab.

This sensor collects steady-state E and H fields at grid points near the HAC plane at each frequency of interest. These values can be then interpolated onto the plane at a user-defined spatial resolution.

Figure 10.11 shows the HEARING AID COMPATIBILITY (HAC) properties dialog.

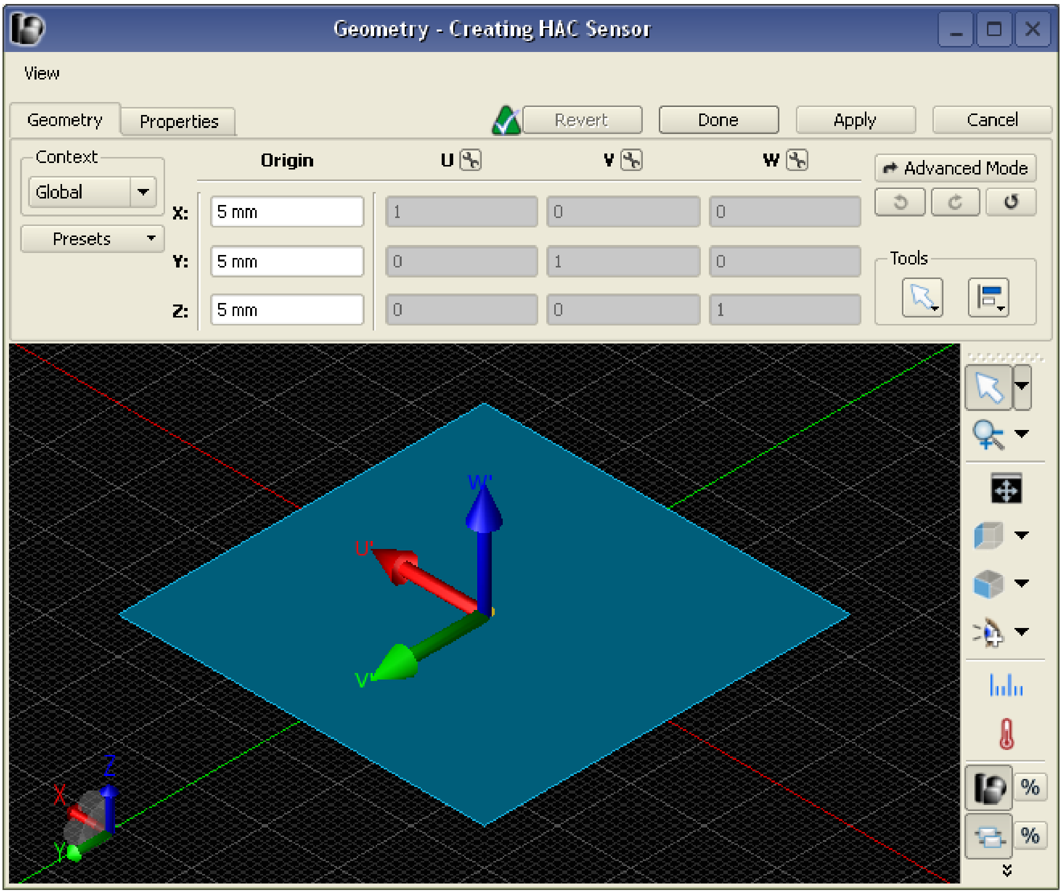


Figure 10.11: HAC Sensor properties dialog

10.3 Far Zone Sensors

FAR ZONE SENSORS are located at theoretical infinite distance from the simulation geometry. They are only available in the absence of PMC or periodic outer boundary conditions, or when more than one PEC boundary is used.

To set up a Far Zone Sensor, under the GEOMETRY tab, choose the THETA/PHI, ALPHA/EPSILON, or ELEVATION/AZIMUTH coordinate system. Far zone sensors can be created with one of the following geometries:

- A range of theta/alpha/elevation over a constant (single) phi/epsilon/azimuth
- A range of theta/alpha/elevation over a range of phi/epsilon/azimuth
- A range of phi/epsilon/azimuth over a constant (single) theta/alpha/elevation

Figure 10.12 demonstrates the transformation of the far zone sensor based on the defined geometry in the THETA/PHI coordinate system.

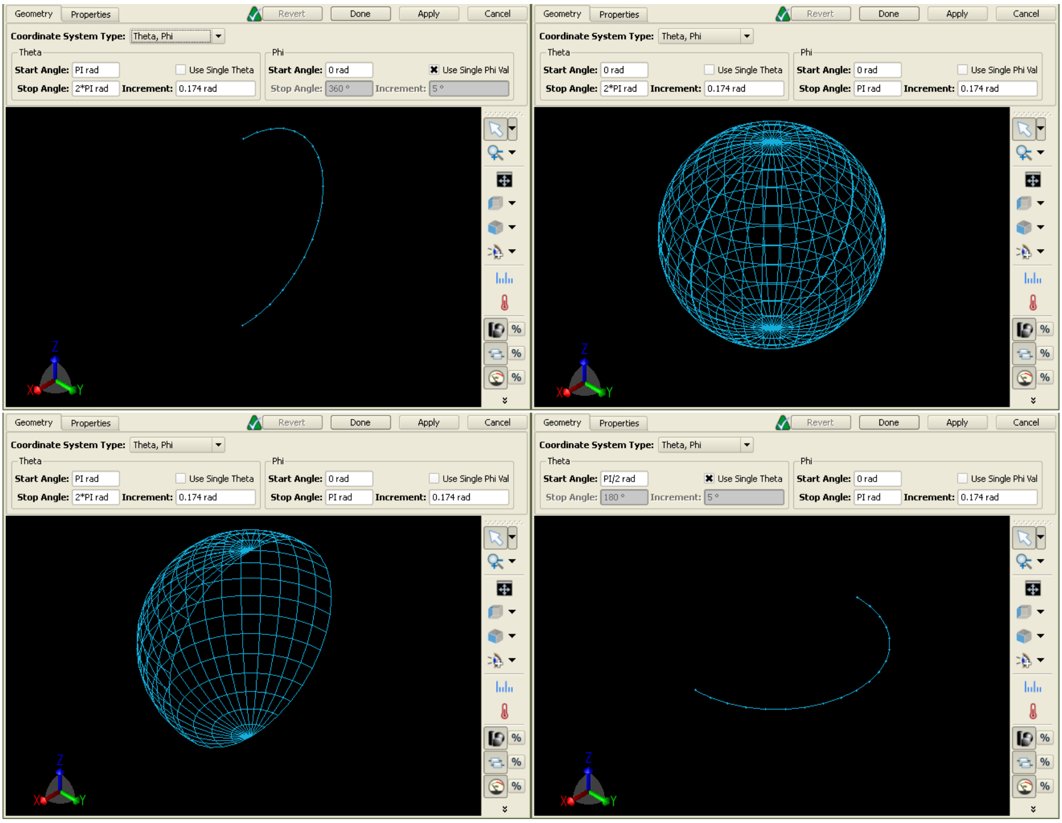


Figure 10.12: Several Far Zone sensor geometries

As seen under the PROPERTIES tab, XFdtd has the ability to compute both a steady-state far-zone (near-to-far) and a broadband far-zone transform. These two options are described below.

Steady-state far-zone transformations

Steady-state transformations are particularly advantageous because the calculation overhead is minimal. They do not require the definition of specific far-zone angles before the FDTD computation, since all patterns are computed in post-processing using data that is automatically stored by the XFdtd calculation engine. Instead, the calculation saves the tangential electric and magnetic fields on the far-zone transformation surface at two timesteps, near the end of the calculation when the system should be in steady state. This sampling determines the complex tangential fields on the far-zone surface at the excitation frequency. These fields are then used in post-processing to provide radiation gain or bistatic scattering in any far-zone direction at any pattern increment. This saves considerable computer time and memory if many far-zone directions are required.

Additionally, the selection of a steady-state far-zone transformation computes the single frequency input impedance, total input power, radiated power, and radiation efficiency. All values computed require that the calculation has reached steady state.

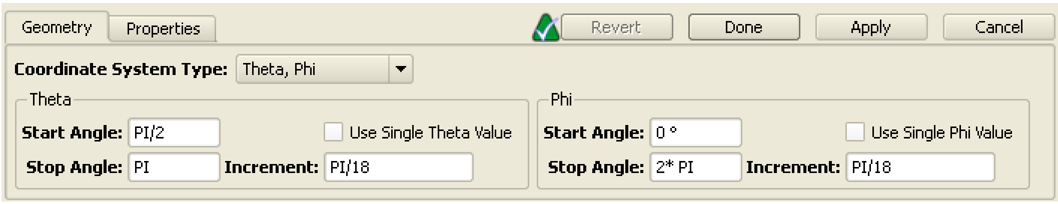


Figure 10.13: Defining a Far Zone sensor

The far-zone gain is displayed in units of dBi. This is the number of decibels of gain of an antenna referenced to the zero dB gain of a free-space isotropic radiator. This value is calculated based on the net power available at the source voltage output. Realized Gain (based on the net available power) and Directivity (proportional to radiated power) are also available for plotting.

Broadband far-zone transformations

The broadband far-zone calculation should be used when broadband results are desired, since the steady-state transform is only performed for a single frequency. This calculation requires extra time for each far-zone angle specified, and unlike steady-state far-zone transformations, all far-zone angles must be defined before running the calculation engine. The broadband far-zone calculation is intended for use in cases where broadband far-zone results at a few points are desired since it is computationally intensive. This calculation will also be desired in instances when far-zone time-domain fields are needed (see below).

Since the calculation is computationally intensive, a sampling interval has been introduced to reduce the number of times it is done during a simulation. The interval can be set manually or computed automatically (the default), and can be found in the Far Zone Sensor editor's Properties tab. When the broadband sampling interval is set to “Use Project Value” and the Project's value is set to “Use Automatic Time Domain Sampling Interval” (in the Project Properties), the project's Frequency Range of Interest and active waveforms are examined to compute the sampling interval.

A feature of the broadband far-zone transformation is that the time-domain far-zone electric fields can also generated and may be plotted. However, a sampling interval greater than one (which the automatic setting will usually do) disables the computation of time domain far-zone results. In order to receive the time domain data, make sure the Broadband Sampling Interval in the Far Zone Sensor's properties is set to “1*timestep”. When an automatic sampling interval is used, the solver's diagnostics file will report the value used in the simulation.

✓ If detailed gain patterns versus angle are necessary, you may reduce calculation time by enabling only steady-state data collection for your sensor and specifying a DFT frequency for your simulation at each frequency you are interested in.

▶ See Sections 12.3.1 and 12.4.2 for more on viewing field results for each angle of the far-zone calculation.

10.4 System Sensors

A SYSTEM sensor is automatically added as a part of each XFdtd project. It collects data within the simulation space which is not applicable to other types of sensors. System sensor results are provided for each run, and these results are defined by the time dependence.

- See Sections 12.2.1 and 12.3.1 to review numerical and plotted results retrieved by System sensors.
- ✓ System sensor results can be rescaled based on user input. See Section 12.4.3.3 for details.

10.5 Port Sensors

PORT sensors save near-zone voltage and current data at the location of a circuit component. They are automatically added to the project when a circuit component is added and its THIS COMPONENT IS A PORT property is checked under the PROPERTIES tab.

- Each Port sensor can have a different source resistance. See Section 7.2.2.2 for more information on specifying the source resistance.
- See Sections 12.2.1 and 12.3.1 to review numerical and plotted results retrieved by Port sensors.

10.5.1 S-Parameter, Group Delay, VSWR and Reflection Coefficient Calculations

When S-Parameter computation is enabled in the CREATE SIMULATION dialog, PORT sensors will also save data used to compute S-Parameters, Group Delay, VSWR, and reflection coefficient. When multiple ACTIVE FEEDS exist in the simulation, enabling S-Parameter computations results in multiple runs where each feed is activated in turn with all other feeds made inactive by replacing them with lumped components with the same impedance configuration. S-Parameters will be computed at each port with respect to each active feed.

When there are multiple active feeds in the simulation and S-Parameter computations are not enabled in the CREATE SIMULATION window, all feeds will be active simultaneously. Results defined for when only one feed is active in the system, such as S-Parameters, will not be available. In this case, Voltage, Current, and Loss are available at each port as well as Active VWSR, which is the same as VSWR but with the allowance for multiple active sources.

- S-Parameters at each Port sensor are calculated using the characteristic impedance retrieved from the circuit component definitions of that port and the active feed.
- See Section 11.1.1 for more on configuring S-Parameter calculations in the CREATE SIMULATION dialog.

10.6 SAR Sensors

XFdtd has several features that fall under the category of biological applications. For compliance with regulations on field absorption in human tissue, the Specific Absorption Rates (SARs) can be computed and averaged. Detailed human body meshes are available for simulations related to effects on realistic heterogeneous models of the body. For some wireless applications, the Specific Anthropomorphic Mannequin (SAM) head is used in addition to the heterogeneous human head (see Figure 10.14).

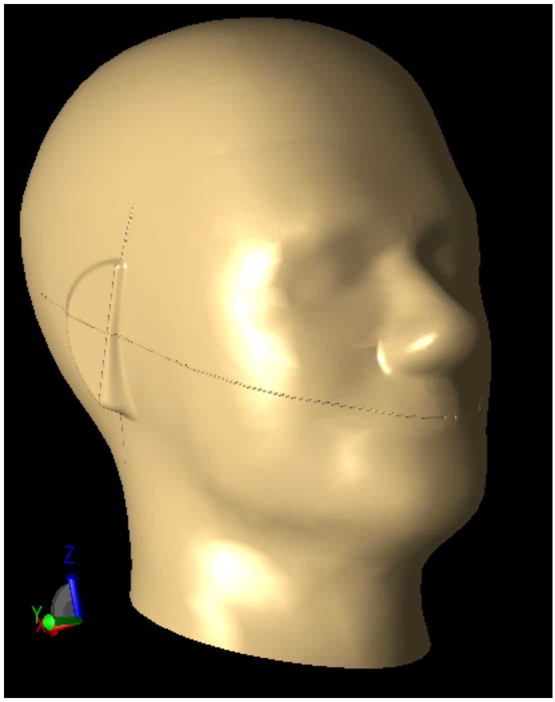


Figure 10.14: The Specific Anthropomorphic Mannequin (SAM) head

The Specific Absorption Rate, or SAR, is the unit of measure commonly used to determine the interaction of electromagnetic fields with human tissue. Most regulations involving devices producing electromagnetic fields must not exceed some exposure limits, typically defined in terms of the SAR averaged over a cubical volume of tissue.

- As an example, the IEEE sets exposure levels in terms of 1 g averaging volumes for most of the body, with a 10 g averaging volume applying to extremities such as the ears and fingers [3].

SAR is defined in terms of the room mean square (RMS) of the electric field magnitude by the relation

$$SAR = \frac{\sigma |E|^2}{\rho}$$

Where

σ is the electrical conductivity in $\frac{S}{m}$, and

ρ is the material density (defined in $\frac{kg}{m^3}$ in XFdtd)

Since the FDTD grid defines the electric fields at the edges of the cells, a single SAR value is formed by summing and averaging the contributions of the 12 electric fields on the edges of the cells. The SAR is then referenced to the center of the FDTD cell.

In XFdtd, the SAR is measured with the SAR sensor and may only be computed in normal dielectric materials. Frequency-dependent materials have a loss term formed by the imaginary part of the permittivity rather than simply by the conductivity, and are not supported for SAR calculations.

The SAR values are saved only in complete voxels (closed FDTD cells) where all 12 edges of the cell are lossy dielectric material (non-zero conductivity) with a non-zero density, therefore steady-state values for SAR and conduction currents will not exist in all planes. To exclude certain materials from a SAR calculation, simply leave the material density as zero. Saving the SAR in a plane of free-space will not produce any useful output as all values will be zero.

✓SAR sensors can be enabled or disabled by right-clicking on the sensor in the PROJECT TREE and changing the SENSOR ENABLED checkbox.

Un-averaged SAR calculation

Un-averaged SAR is measured in XFdtd using the SAR SENSOR. Note that most specifications which involve SAR limits are defined in terms of constant-mass regions, so they will require averaged SAR.

Averaged SAR calculation

The averaged SAR calculation is more meaningful under most circumstances. This calculation is defined by regulations from organizations such as the IEEE and various government bodies. It is computed over cubical volumes of voxels where no face of the averaging volume is external to the body (and thus full of air or other non-tissue material). In certain cases, particularly at the surface of the body, the cubical volume rule can not be satisfied. In those situations, special rules exist for setting the SAR value in a given voxel.

- Refer to the IEEE published standards for regulating SAR calculations and setting SAR values [3].
- Only one SAR averaging region can be defined per calculation run. Additional averaging can be performed as a post-processing step, given that sufficient un-averaged SAR was collected for the region of interest.

Averaged SAR is measured in XFdtd using the SAR AVERAGING SENSOR. There are several ways to compute average SAR values in XFdtd, as shown in the SAR AVERAGING tab. One way is to save 1 gram or 10 gram average SAR regions over the FULL GRID. During the calculation the averaged SAR values will be computed for all appropriate voxels. This process is time consuming, and since the 10 gram SAR is only applicable to the extremity tissues, it is not necessary to compute it for the entire geometry.

As an alternative to computing values over the entire grid, the XFdtd interface also has a tool for computing the average SAR over a BOX REGION. In this mode, the locations of opposing corners of a rectangular box are entered. Only Yee cells that are within the box (once the endpoints have been snapped to the nearest E-grid vertices) will have their average SAR computed. An additional option in this mode specifies whether all data outside of box is treated as free space in the averaging process. Figure 10.15 displays the BOX REGION dialog below.

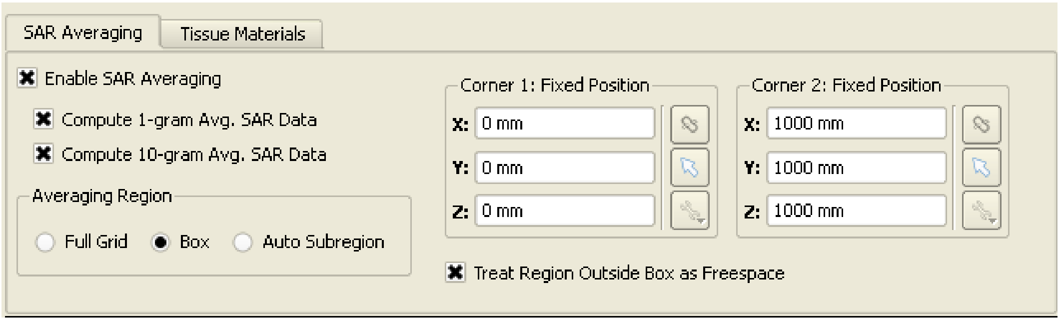


Figure 10.15: Requesting averaged SAR statistics in a boxed region

Another alternative is to select the AUTO SUBREGION option. In this case, the MAX/MIN SAR RATIO is defined in decibels so that the requested 1 gram or 10 gram average is performed only where applicable, thus saving a great deal of calculation time. This quantity must be entered as a unitless ratio (amplitude) or in dBp (a decibel unit with suffix to indicate an absolute unit of electric power). For example, in a typical application, the extremity tissues would be identified by different material types from the body tissues, so indicating this value in the MAX/MIN SAR RATIO would isolate the calculation to that specific region. Figure 10.16 displays this dialog.

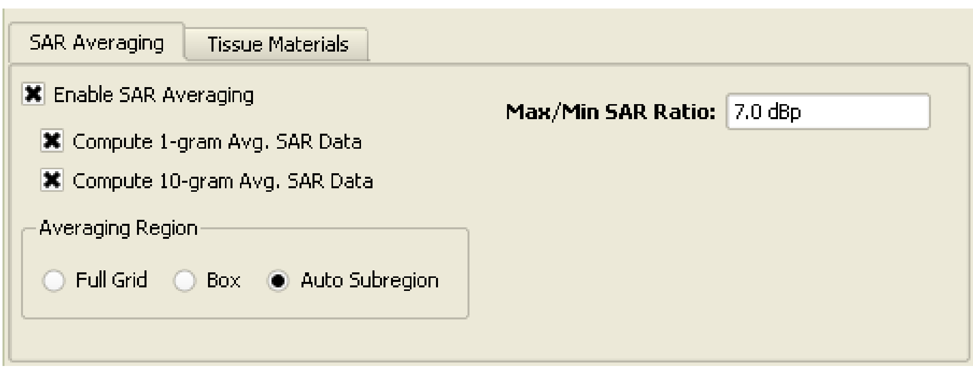


Figure 10.16: Requesting averaged SAR statistics in an automatic subregion

XFdtd also offers tissue selection control under the TISSUE MATERIALS tab. You can compute averaged SAR for ALL TISSUE MATERIALS, or for SELECTED EXTREMITY TISSUE MATERIALS. In choosing the latter option, a dialog box will appear with a list of available pre-defined materials to include in the calculation.

10.7 Sensor Visibility

XFdtd offers the user the ability to toggle the visibility of sensor. To turn off a sensor, un-mark the ENABLE checkbox.

To toggle the visibility of a sensor, right-click on the sensor in the PROJECT TREE and select either SET VISIBLE, or SET INVISIBLE, as seen in Figure 10.17.

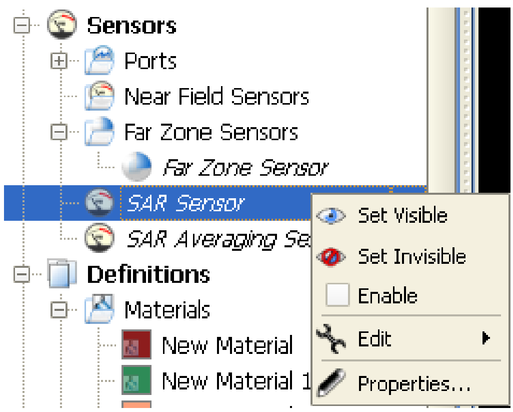


Figure 10.17: Setting the visibility of a sensor

Chapter 11

Running Calculations

In this chapter, you will learn...

→ how to create and run a simulation

→ how to start the XFdtd calculation engine

→ the main factors to consider before beginning a calculation

After the XFdtd project setup is complete, it is time to run calculations on the geometry. The SIMULATIONS workspace window stores the project simulation(s). From this window, the user creates, queues and runs the simulations.

The actual electromagnetic calculations are not made by the XFdtd GUI. Rather, they are run by a separate program called XFsolver once the finished project has been saved. Usually this process is run from the GUI, which calls XFsolver as needed. However, the user may also run XFsolver directly from the command line or from a remote computer. Once the simulations have been run and the desired calculations are complete, the results can be viewed within the GUI.

11.1 Simulations Workspace Window

The SIMULATIONS workspace window provides the interface to queue projects to be run with XFsolver, the calculation engine. Figure 11.1 shows the Simulations workspace window.

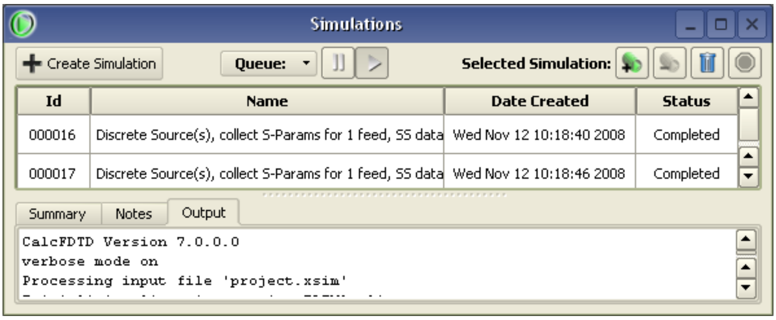


Figure 11.1: The Simulations workspace window

The SIMULATIONS workspace window lists the name of every simulation that has been created for the project. Its STATUS column shows whether the simulation has been created, queued, or completed.

This window also provides the user with the ability to choose how to run the simulation on their computer. Under the QUEUE drop-down list, specify whether to run the simulation on the CPU or on the hardware card with XStream hardware acceleration.

- See Section 11.3.2 for a description of the QUEUE drop-down list.

New simulations are created by pressing the CREATE SIMULATION button in the upper-left corner of this window. The associated definitions are described below.

11.1.1 Creating a New Simulation

Choosing a Source

There are two main options for choosing a source in in XFdtd, as seen in Figure 11.2. When a voltage or current source is used as an input, check USE DISCRETE SOURCES. Alternatively, the user may also SELECT EXTERNAL EXCITATION to use as the source. USE NO SOURCES should be selected in special cases with the Static Solver. Discrete sources and external excitation sources are briefly described below.

- See Section 7.2 for more about discrete sources.
- See Section 7.5 for more about plane wave excitations.
- See Section 7.6 for more about Gaussian beam excitations.
- See Section 7.8 for more about static voltage and the static solver.

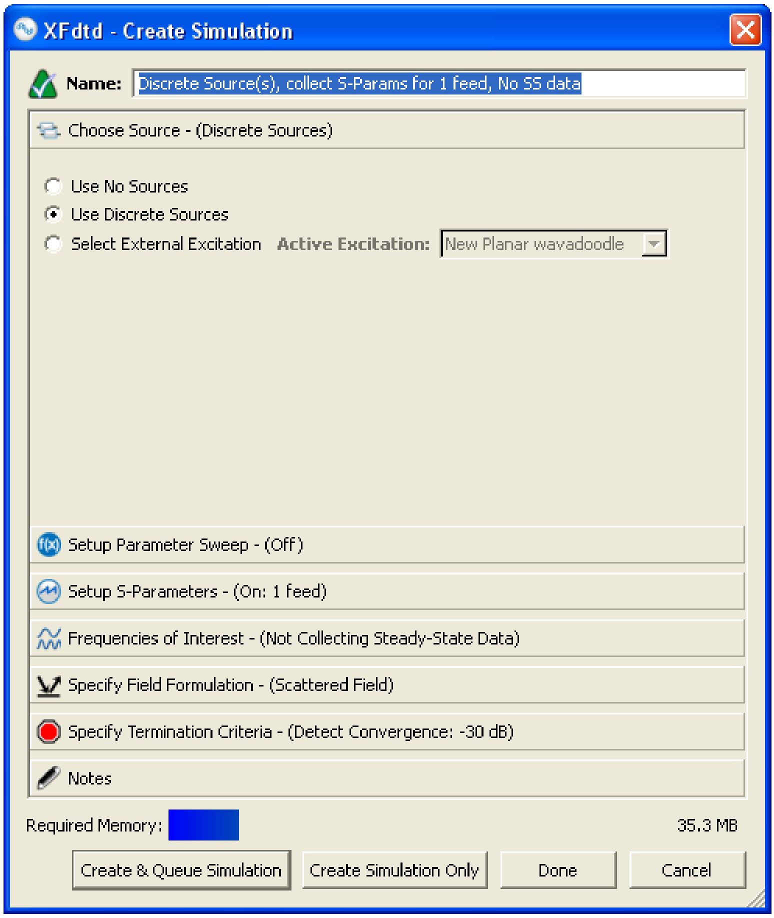


Figure 11.2: Choosing a source for a simulation

Discrete Source. A discrete source is a cell edge on which the electric field is modified by the addition of some type of input waveform. The cell edge can be modified to behave like a voltage or current source. All calculations with discrete source input are performed in total field. Antenna and microwave circuit computations are examples of calculations that may be performed using discrete sources.

Plane Wave. An incident plane wave source is assumed to be infinitely far away so that the constant field surfaces are planar and normal to the direction of propagation. Calculations of radar cross section or scattering may be performed using this input. When a plane wave excitation is selected, additional configuration options will be available on the SPECIFYING FIELD FORMULATION tab below (see Section 11.1.1).

Gaussian Beam. This choice allows for a focused Gaussian beam source in which the incident electric field has a two-dimensional, radially-symmetric Gaussian distribution in planes normal to the incident direction and converges to maximum intensity at the focus point. All calculations with a Gaussian beam source are performed in scattered field, though total field values may also be saved and displayed also. Unlike the plane wave and discrete sources, the Gaussian beam source requires that the source waveform be sinusoidal. Examples where this type of source is useful include structures used at optical frequencies and situations where it is desired to illuminate only a portion of the geometry.

No Source. Use No Source requires that one or more static voltage points already exist in the project, and that the Enable Static Solver checkbox, which appears in the Static Solver tab, is checked. If both of these requirements are not met, and Use No Source is selected, a warning icon will appear.

Parameter Sweep Setup

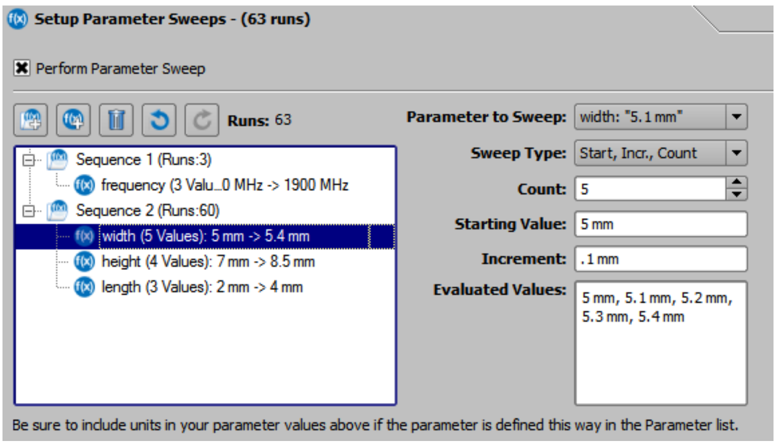


Figure 11.3: Setting up a parameter sweep

A PARAMETER SWEEP defines the values used for a particular Parameter when the simulation is created. Parameter sweeps facilitate simulation creation by methodically altering project parameter values and therefore the project's state. Multiple project states are represented, as an individual run, inside of a single simulation. Each run assumes one of the parameter combinations defined in the parameter sweeps.

PARAMETER SEQUENCES by themselves don't have any settings but instead serve as a container for parameter sweeps. Inside each sequence, a matrix is formed using the values of the contained parameter sweeps. A parameter sequence may only define the values for each project parameter once but multiple sequences can be used if needed.

The interface for creating and editing sequences and sweeps consists of two parts. On the left is a tree for summary, creation and deletion. The right side is used to edit the parameters of the sweeps. The following sections describe these two halves in more detail.

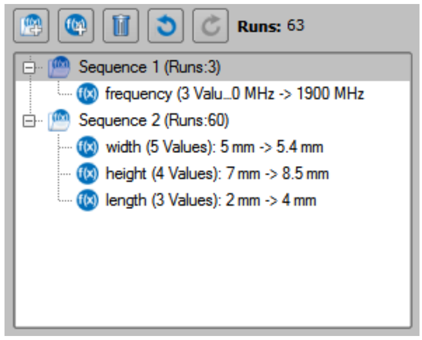


Figure 11.4: Parameter sequences tree interface

The tree structure shows the grouping of sweeps into sequences. Each sequence reports the number of runs defined by the matrix of values in the contained sweeps. The sweeps show the parameter being defined, the number of values and the first and last value. If any of the entries represented by the tree are in an invalid state the yellow error triangle will appear at the right side of the corresponding row. Along the top of the tree are buttons to create a new sequence, create a new sweep, delete, undo and redo.

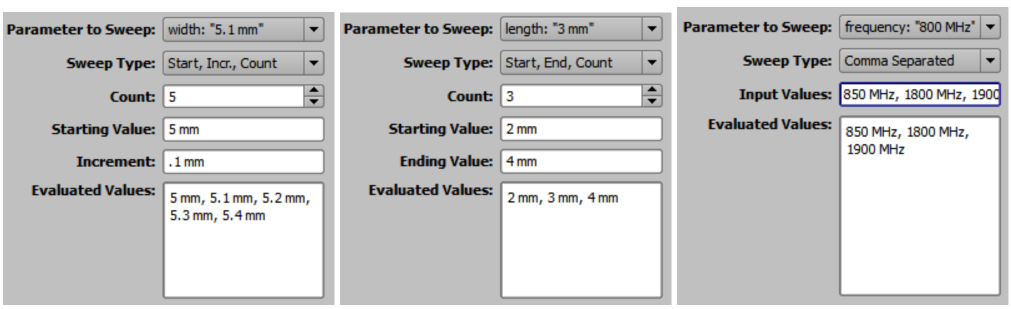


Figure 11.5: Parameter sweep entry methods

Sweep editing has three different entry methods but the overall philosophy remains the same. First the parameter must be chosen, then the SWEEP TYPE specifying how the values are entered. Once the values are entered for the sweep the resultant values will be shown in the evaluated values. Parameter sweeps are defined using one of three types:

- START, INCR., COUNT defines evenly spaced values using start value, count and an increment between values
- START, END, COUNT also defines evenly spaced values but uses a start value, ending value and a count
- COMMA SEPARATED provides a freeform input method to define each value to be used, separated by a comma

It is important to specify the parameter sweep inputs properly. XFdtd allows units to be used when using a parameter or during specification of parameter values. In order to assure correct behavior the input

S-Parameters Simulation Setup

To calculate S-Parameters for a project with multiple ports defined, the simulation will consider one port at a time to be the active port. The computation with that source active will provide data for a column of the S-Parameter matrix. For example, if port one is excited in a three-port circuit, XFdtd will determine S_{11}, S_{21}, and S_{31}. If different ports are to be excited, a separate calculation must be performed with each port active. For example, if the full S-Parameter matrix for a two-port problem is desired, two calculations must be performed with a different port active in each. XFdtd will save the S-Parameters for each run in separate files, differentiated by the active port number.

- See Section 10.5 for more about the data saved with port sensors.

- If a parameter sweep is specified in the SETUP PARAMETER SWEEP tab for multiple parameters in addition to specifying multiple ports within the SETUP S-PARAMETERS tab, the parameter sweep will be performed for each individual port.

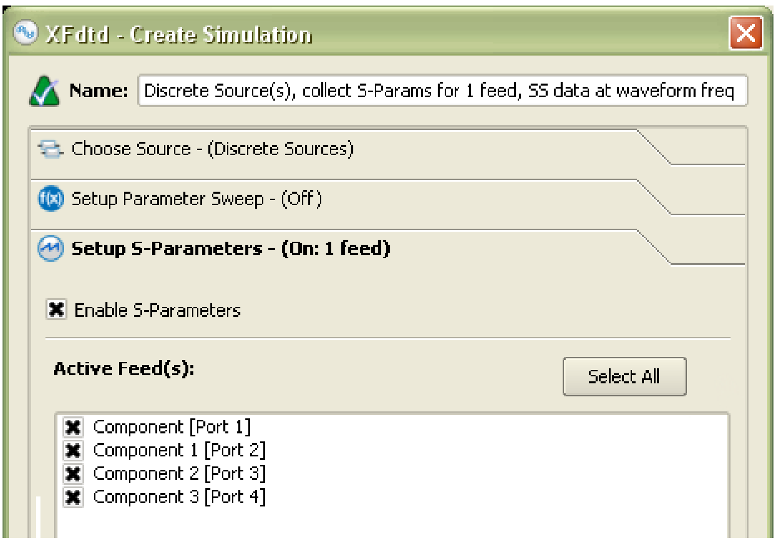


Figure 11.6: Setting up S-Parameters

Frequencies of Interest

Within this section, the FREQUENCIES tab specifies whether the simulation is a broadband (transient) or steady-state calculation. For broadband calculations, uncheck the COLLECT STEADY-STATE DATA box at the top of this tab. For steady-state calculations, check this box and choose whether the calculation is to only USE WAVEFORM FREQUENCY or to USE SPECIFIED FREQUENCIES of interest. By specifying more than one frequency of interest, the calculation engine will essentially run a separate calculation at each discrete frequency by running DFT, saving each as its own run. This will therefore increase the calculation time in comparison to using only the waveform frequency.

In the OPTIONS tab, the user has the ability to specify several settings specific to collection of steady-state data. The “Store Data” option specifies whether to save temporary data IN MEMORY or ON DISK. Saving the data in memory will speed up the calculation because there is no file saving or loading from disk, but it increases the memory requirements. The DISSIPATED POWER ALGORITHM control allows selection of which algorithm is used to determine the dissipated power in the simulation space. The "Far-field Method" (the default) determines it from the power radiated through the edges of the simulation space and is less memory and computationally expensive than the "Near-field Method". The "Near-field Method" determines it by summing the losses on all Yee-cell edges in the simulation space and is more memory and computationally expensive than the "Far-field Method" due to need to sample and store data over the entire geometry. The "Near-field Method" is more accurate. Therefore, if you are interested in accurate losses/efficiencies, use the "Near-field Method" otherwise choose the "Far-field Method." The SAVE DATA FOR POST-SIMULATION FAR ZONE STEADY-STATE PROCESSING checkbox determines whether the complex fields on the far zone box are saved at the end of the simulation. This amount of data and the effort required to save it are generally very small, so it is recommended that this option always be enabled. When the NORMALIZE FIELDS checkbox is checked (the default), when a broadband source is used the steady-state field data computed for each frequency is adjusted so that it produces the same data as for an equivalent sinusoidal source. This ensures that comparisons of broadband steady-state results with any other steady-state results (from simulations using either sinusoidal or broadband sources) agree. Therefore, leave this box checked unless you have a very specific need. SAMPLING INTERVAL specifies how often to sample a data type (in timesteps), and determines the highest possible FREQUENCY TO RESOLVE. The automatic setting analyzes the frequency content of the input waveforms to generate a sampling interval. You can manually choose the maximum frequency by considering the frequency content of your pulse. Choose a value where the frequency content of your pulse has rolled off sufficiently far enough to avoid aliasing issues. Entering the highest frequency that needs to be resolved will automatically adjust the sampling interval. A sampling interval of one requires the most calculation time but also provides the most accurate results because it reduces the effects of aliasing.

- Section 12.5 describes viewing far zone post-processing results.

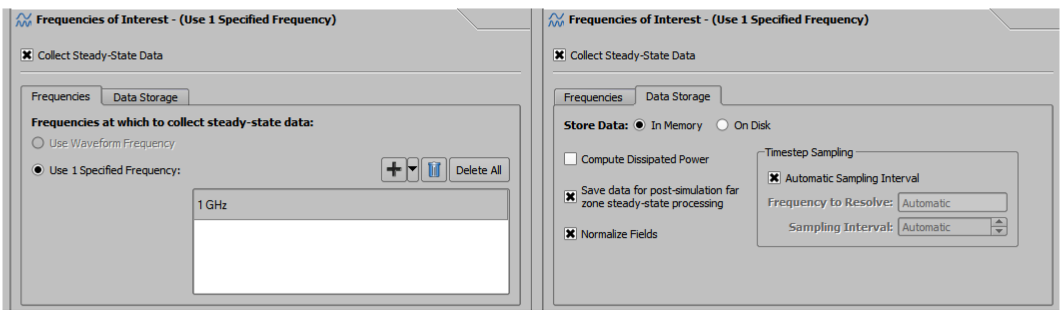


Figure 11.7: Setting up Frequencies of Interest under the Frequencies and Data Storage tabs

Specifying Field Formulation

The SPECIFY FIELD FORMULATION controls are present whenever a plane wave source is selected to excite the simulation space.

The XFsolver application supports both the total-field/scattered-field plane wave simulation technique and the pure scattered-field technique. When the USE TOTAL/SCATTERED FIELD FORMULATION checkbox is checked, the total-field/scattered-field formulation is used.

Selecting total-field/scattered-field versus pure scattered-field plane wave. With the total-field/scattered-field formulation, the total E- and H-fields are simulated in the portion of the simulation space which contains the geometry, but only scattered E- and H-fields are simulated near the outer boundaries. The interface between the total-field and scattered-field computation regions is located eight cells into the FDTD mesh for non-periodic boundaries, and must be free-space.

With the pure scattered-field formulation, scattered E- and H-fields are simulated over the full FDTD mesh. When this formulation is used, plots of both total and scattered field values will be available for display.

In most cases, total-field/scattered-field plane wave is preferable to the pure scattered-field formulation. Figure 11.8 shows total E-fields propagating (from right to left) through a shelled geometry of Perfect Electric Conductor (PEC) material. The image on the left represents a pure scattered-field plane wave source, while the image on the right shows a total-field/scattered-field plane wave source. Since the walls of the box are PEC material, the electric fields within the free space interior of the box should be zero. This is true when the total-field/scattered-field formulation is used, but small fields are present inside of the cavity with the pure scattered-field formulation. Thus, the total-field/scattered field formulation gives a result which is more correct.

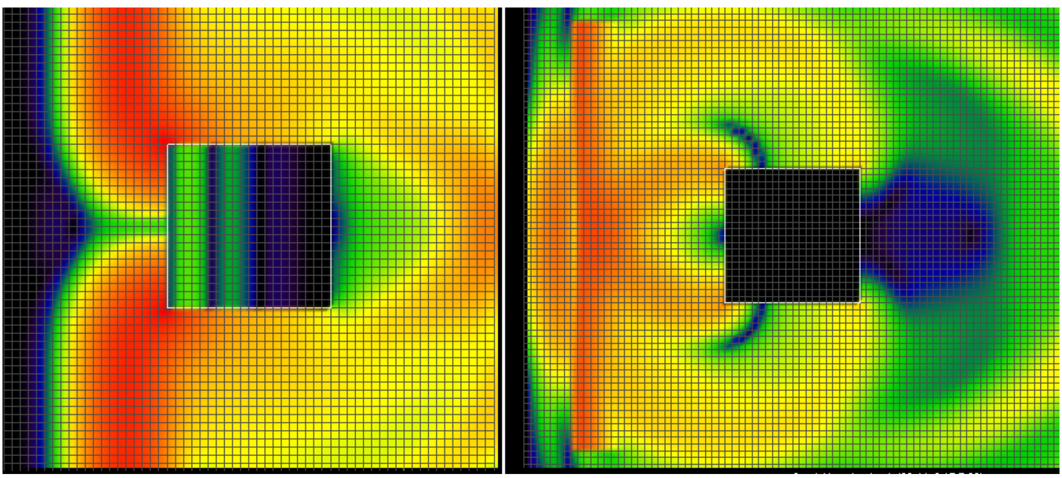


Figure 11.8: Total E-field of a Pure Scattered-Field (Left) vs. Total-Field/Scattered-Field (right) Plane Wave Source

In some cases, however, the total-field plane wave source will also create inconsistencies in the calculations. Figure 11.9 shows the resultant electric field when an object crosses the total-field/scattered-field interface (intersection represented by the white arrows). This will produce incorrect results. To fix this problem, the interface may be moved so that it is sufficient distance from the geometry. However, to move the interface, the bounding box of the project must be increased which will also increase the memory requirements of the project. Alternatively, a scattered-field plane wave can be used with the understanding that the region within the shelled geometry is incorrect.

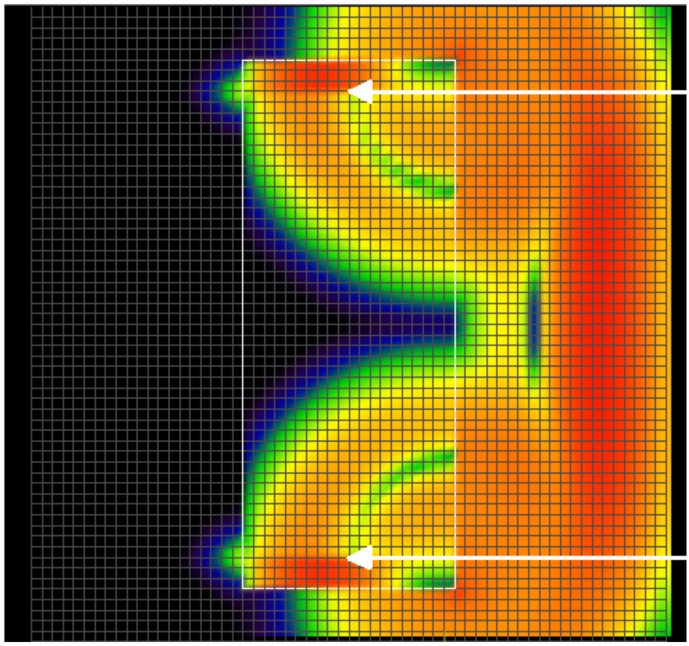


Figure 11.9: Incorrect results due to an object crossing the interface

Figure 11.10 demonstrates a similar problem where the interface is too close to the bounds of the object. In this case the fields fall within the "shadow" region of the object and are not calculated correctly. The image on the left shows the effect of the shadow region early in the field sequence. The image on the right shows the incorrect field values later in the sequence at the interface between the total-field/scattered-field interface. The interface, like in the previous example, must be adjusted if total-field plane wave is to be used.

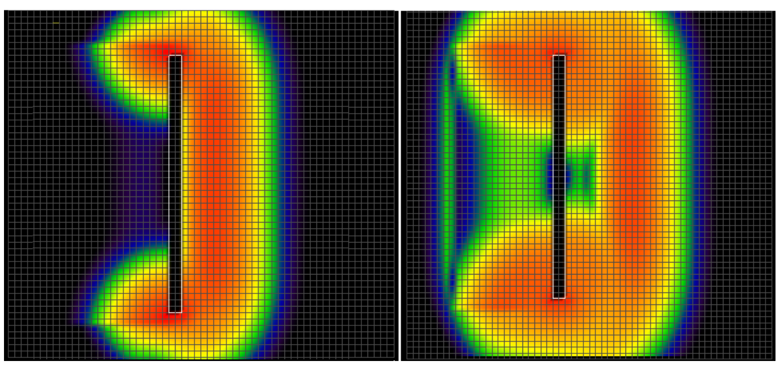


Figure 11.10: Incorrect results when a field propagates within the "Shadow Region" of the geometry

Selecting desired total-field/scattered-field interfaces. The SPECIFY TOTAL/SCATTERED FIELD INTERFACES controls are only applicable when a total-field/scattered-field plane wave source is used in conjunction with periodic boundary conditions.

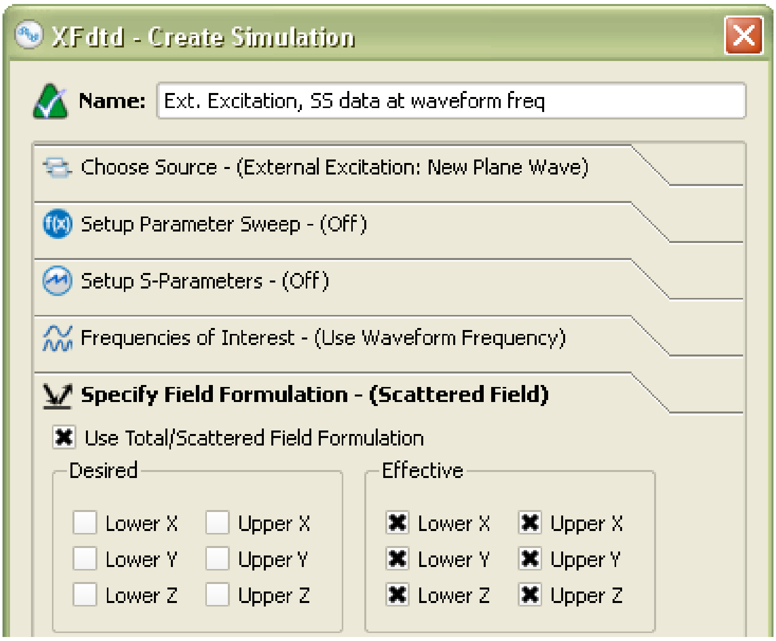


Figure 11.11: Setting up Total/Scattered Field Formulation

There are two sets of X, Y, and Z boundaries listed within this editor. The checkboxes listed under the DESIRED heading are available for users to indicate which sides of the simulation space should include total-field/scattered-field interface. Depending on the boundary conditions, however, the selected DESIRED definitions may not be applicable or may have to be applied in conjunction with other definitions. Thus, the actual definitions that will be applied during the calculation are displayed under the EFFECTIVE heading.

When PERIODIC boundaries are specified, certain sides for the interface may be turned off and the total-field region may extend to the boundary using this definition. Periodic boundaries may be useful for applications such as optics where small geometries are repeated over and over again. Figure 11.12 is an example in which the outer boundaries have been set to PERIODIC in the the Y and Z directions, and the total/scattered field interface has been turned off (unchecked) in the lower X, upper and lower Y, and upper and lower Z directions.

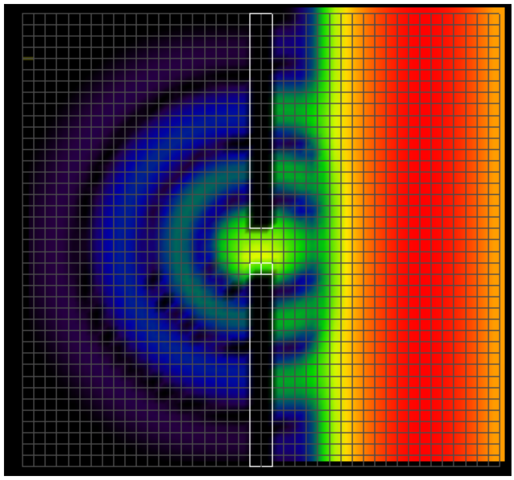


Figure 11.12: The interface may be turned off for problems using periodic boundary conditions.

Static Solver

The Static Solver tab only appears if the project contains static voltage points. To ignore these voltage points, de-select the Enable Static Solver checkbox.

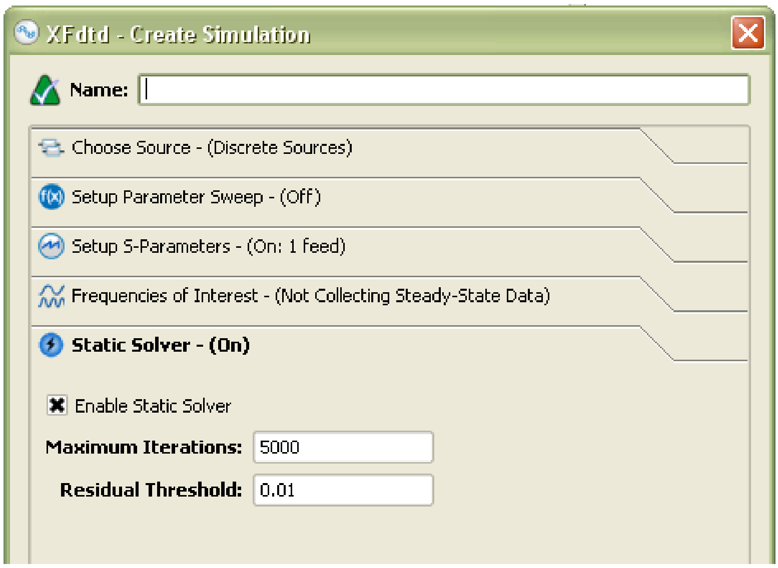


Figure 11.13: Selecting iteration and threshold criteria within the Simulations workspace window

The tab contains two options. MAXIMUM ITERATIONS defines the maximum number of repetitions the solver will compute before exiting if convergence is not reached. RESIDUAL THRESHOLD sets the value to which the iterative Laplace solver will converge. Using the default settings for these parameters is recommended.

Specifying Termination Criteria

The termination criteria specify the conditions which stop the simulation. Many applications require simulations to reach a state of convergence in order for results to be valid. Often the necessary minimum iterations of the solver are unknown so an overly aggressive maximum is suggested. The maximum simulation time is required to prevent simulations which wont converge from running endlessly. Several settings in the termination criteria help simplify the process of finding the middle ground allowing results to run to convergence.

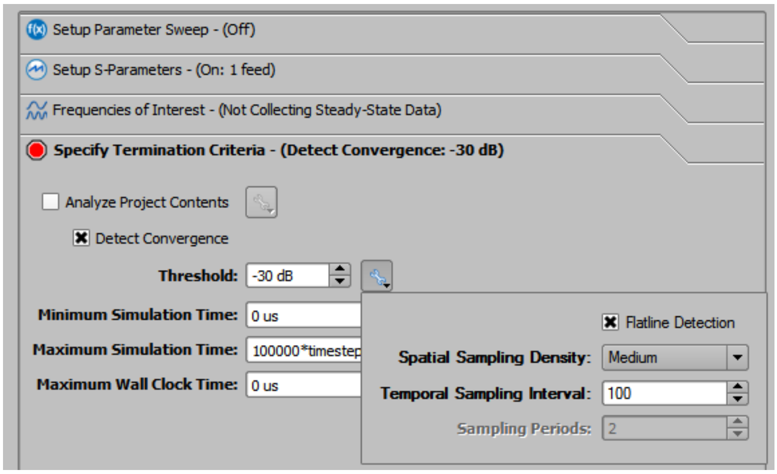


Figure 11.14: Selecting termination criteria within the Simulations workspace window

Traditionally simulation progress is thought of in number of iterations of the solver ending at a maximum. This convention is often useful for providing an estimate of how long the simulation will run but is abstracted from the physical structure of the system under test. The combination of the time step durations and project content variations prevent a one size fits all specification.

The most basic method of ending a calculation is defining a value in the MAXIMUM SIMULATION TIME definition. Once the defined amount of time has passed, the calculation will stop. It is important to note that the calculation will terminate regardless of whether or not convergence has been met, so setting this definition to a proper value is important. If it is too low, results will be of no use.

An advanced method of ending a calculation is defining a value in the MAXIMUM WALL CLOCK TIME definition. Wall clock time is the elapsed time from when a job starts running to the time it completes. Like MAXIMUM SIMULATION TIME, once the defined amount of time has passed, the calculation stops. This field is optional. If any value other than 0 is entered in the field, XFdtd will stop the calculation (assuming it is still running) when it reaches the elapsed time.

Selecting the DETECT CONVERGENCE check-box will automatically stop the calculation if slow convergence is detected, regardless of whether the number of maximum timesteps have completed. Convergence in a broadband calculation is met when all electromagnetic energy has dissipated to essentially zero. Due to numerical noise in the calculation, there may be a trivial amount of electromagnetic energy, even after the calculation has converged. The value defined in the THRESHOLD dialog dictates when the calculation has reached an acceptable value to assume convergence.

✓ A general rule of thumb is that the values should have diminished by at least 30 dB or 1/1000th from the peak values.

For sinusoidally-excited problems, typical values for this setting range from -55 dB to -25 dB depending on the level of accuracy versus runtime desired. For instance, if high accuracy S-Parameters are the goal, then the convergence threshold should be set to lower than -30 dB. If however, the user wants to view antenna patterns, -30 dB to -35 dB is suitable. The trade-off here is run time for accuracy. In general, for sinusoidally excited problems, using automatic convergence with a threshold of about -35 dB will produce very accurate results and will run in the shortest possible time to reach this level.

If a calculation is finished but convergence has not been reached, the output from most calculations will be meaningless. The only option is to decrease the convergence THRESHOLD or increase the MAXIMUM SIMULATION TIME in the calculation and run it again. If a resonance is occurring at a frequency beyond the range of interest, and a broadband input is used, the input waveform can be modified to limit the frequency content and eliminate the resonance. If the resonance is in band, or a sinusoidal input is used, then this is not applicable and the maximum simulation time should be increased or a lower convergence threshold must be chosen.

Advanced options. There are several options available in the ADVANCED button of this window that allow for more specific termination criteria.

Selecting the FLATLINE DETECTION check-box will stop the calculation if a slow convergence is detected. This may occur if the user sets the convergence threshold to a very low value (e.g. <-50 dB). In this case the calculation may converge but to a level higher than specified.

✓ To prevent false convergence, ”slow” convergence can only be detected once the convergence level has reached at least -40 dB.

For a steady-state calculation, convergence is reached when near-zone data shows a constant amplitude sine wave - when all transients have died down and the only variation left is sinusoidal. In this case ”convergence” is tested on the average electric field in the space for its deviation from a pure sine wave. If DETECT CONVERGENCE is turned on, XFdtd automatically places points throughout the space for this purpose. It is particularly important to monitor the results inside high permittivity dielectrics since the field propagation in these materials is much slower than in free space.

To ensure that steady-state calculations converge, XFdtd will allow the user to control the TEMPORAL SAMPLING INTERVAL and SPATIAL SAMPLING DENSITY of the sample points. The TEMPORAL SAMPLING INTERVAL definition is used to control the interval at which convergence is tested during calculations with broadband (pulse) excitations.

✓ Setting this value to 100 or 200 timesteps is typical. Setting this value to much less than that increases the computational overhead a small amount.

SPATIAL SAMPLING DENSITY is used to control how many spatial samples are used to determine convergence. The sample points are equally spaced in all three dimensions of the grid. LOW density samples every 4th point in each dimension, while HIGH density samples every point in each dimension. For very low frequency problems or where the number of timesteps per RF cycle is greater than 200 (e.g. very small cells with a low frequency excitation), this should be set to LOW. For moderate frequencies or where the number of timesteps per RF cycle is less than 200 but greater than 100 this should be set to MEDIUM. For high frequency problems where the number of timesteps per RF cycle is less than 100, a HIGH setting gives the best accuracy. This setting is for both broadband as well as sinusoidal excitations.

The SAMPLING PERIOD is used only when there is a sinusoidal excitation.

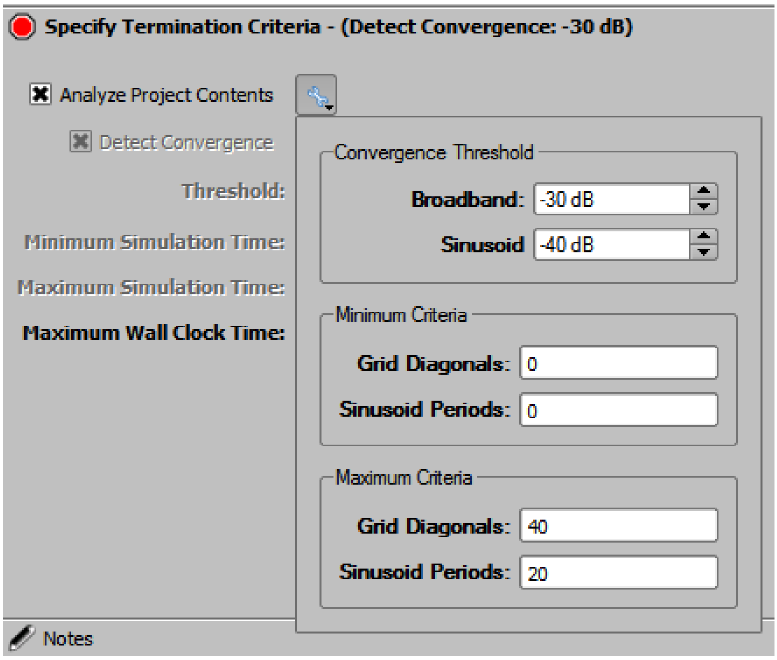


Figure 11.15: Use Project Contents Termination Criteria Properties

Analyze project contents. The ANALYZE PROJECT CONTENTS functionality provides a simplified input method for the termination criteria based on the project setup. The MINIMUM SIMULATION TIME, MAXIMUM SIMULATION TIME and CONVERGENCE THRESHOLD are calculated from the project contents. The default settings work well for most cases but there are several classes of problems which require the settings to be modified for instance, highly resonant structures may require the maximum criteria to be increased.

Under ANALYZE PROJECT CONTENTS settings the CONVERGENCE THRESHOLD is separated into two values for sinusoidal and broadband waveform shapes. The value is chosen based on the active waveforms in the simulation. Often the sinusoidal convergence requires a lower threshold to ensure convergence has been reached. The CONVERGENCE THRESHOLD used will be reported in the THRESHOLD spin box.

Both minimum and maximum criteria may be set using two different inputs. The GRID DIAGONALS entry uses the maximum extent of the simulation space as a relative metric to define time using the speed of light in free space medium. When a sinusoidal waveform is active the sinusoid periods entry method will also be used. The maximum of the two entry methods determines the simulation time which is reported in the disabled editors.

Notes

The NOTES section is simply a tool for users to add any notes to be attached to a project. The notes will be available in the NOTES tab of the SIMULATIONS workspace window after the simulation is created.

Required Memory

The REQUIRED MEMORY section displays an estimate of how much memory is required to simulate the project. Mousing over the estimate, displays a tooltip. The tooltip contains additional estimates (Minimum Usage, Maximum Usage, and Best Estimate), and an explanation of how the estimate was derived.

11.1.2 Queueing and Running Simulations

After defining the necessary components of the simulation in the CREATE SIMULATION dialog described above, click the CREATE AND QUEUE SIMULATION button and the main window will show all of the queued and completed simulations as seen in Figure 11.16. Only one simulation can be run at one time, so as soon as a calulation is complete, another queued simulation will begin.

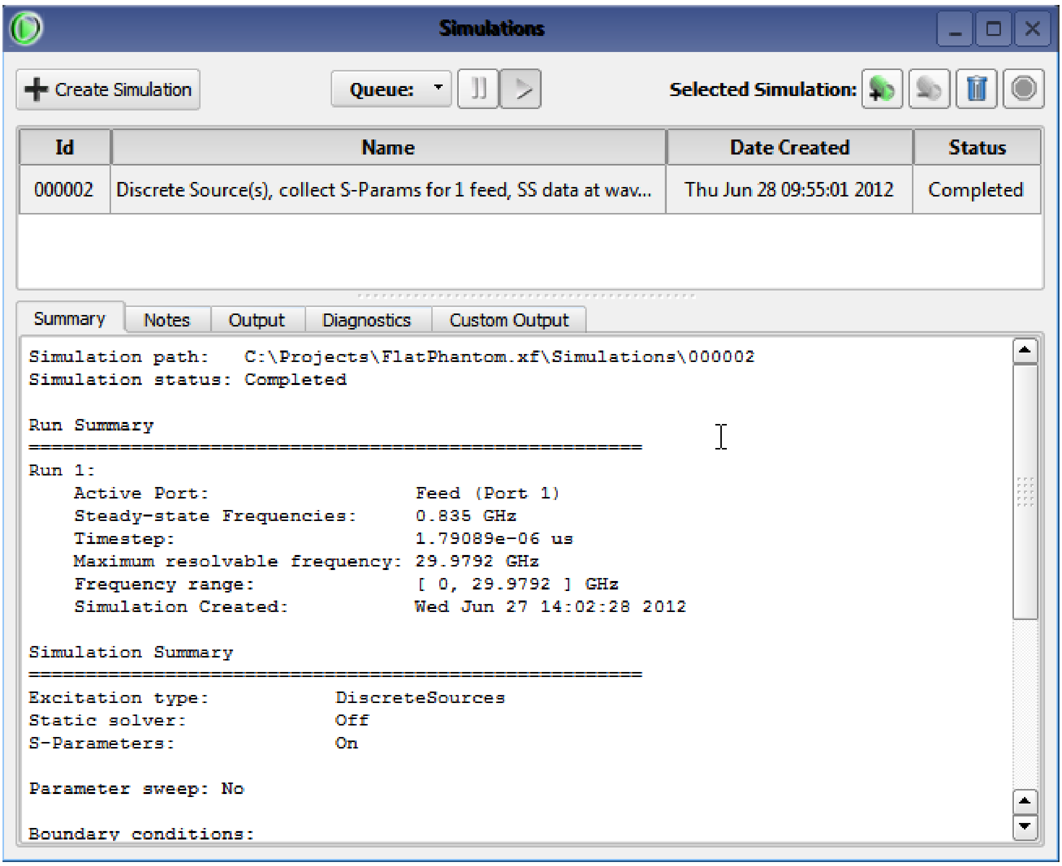


Figure 11.16: Running a calculation

Below this dialog are three tabs:

- SUMMARY, where a basic summary of the simulation is provided, as seen in Figure 11.16.
- NOTES, which simply documents any notes that were added by the user in the CREATE SIMULATION window.

- OUTPUT, which provides the output generated by XFsolver. Statistics such as the percentage of completion, current timestep, convergence, time elapsed, etc, are listed for every simulation in this tab (including output information for every parameter swept).
- DIAGNOSTICS, which provides the diagnostic output generated by XFsolver. This information contains detailed information about each run in the simulation.
- CUSTOM OUTPUT, which displays the contents of `project.clog` files in the simulation directory and its Run directories. These files end-user generated, and are displayed here for convenience.

11.1.3 Creating Simulations

If you only want to create a simulation, but not run it immediately, click the CREATE SIMULATION ONLY button. The simulation will be added to the SIMULATIONS workspace window. To run the simulation, select it and click the ADD TO QUEUE button, located to the right of SELECTED SIMULATION at the top of the window. The XFsolver engine will begin processing the simulation immediately (if no other simulations are running). Alternatively, you can start XFsolver from the command line and run the simulation from there. This process is described in more detail in Section 11.2.2.

11.2 Starting the XFsolver Engine

There are two ways to start the XFsolver engine. The first way is to launch it from XFdtd in the SIMULATIONS workspace window. The second way is to start the calculation from a command line. This is preferable for calculations that require a large amount of memory in order to free memory used by the interface.

11.2.1 Running XFsolver Within the XFdtd Interface

Calculations launched from XFdtd are sent to the XFsolver engine from the SIMULATIONS workspace window. A list of every simulation that has been created within the CREATE SIMULATION window is listed in the main window. When a simulation is ready to run, send it to XFsolver by selecting the simulation and clicking the ADD TO QUEUE button. Although multiple simulations can be queued at once, only one simulation is run in XFsolver at a time. Once the ▶ PLAY button is pressed, each simulation will run one at a time until all of the queued simulations have terminated.

▶ See Section 11.1.1 for more on creating a new simulation.

11.2.2 Running XFsolver From the Command Line

In some cases, it is most useful to run XFsolver from a command line. This will prevent XFdtd from overloading since it will not be allocating memory for XFsolver in addition to its normal memory requirements. This is not an issue for most calculations, but it may cause problems when the memory required to run a calculation approaches the limitations of computer memory. In this case, closing XFdtd and running XFsolver from the command line will free up any memory used by XFdtd.

To run XFsolver, navigate to the appropriate project SIMULATIONS or RUN directory and enter the command (using the appropriate version number):

Windows:

```
"C:\Program Files\Remcom\XFdtd 7.3.0 (64-bit)\bin\Win64.NET2010\xfsolver" [options]
```

Linux:

```
/usr/local/remcom/XFdtd_7.3.0.2/bin/Linux-x86_64RHEL5/xfsolver [options]
```

MacOS:

```
/Applications/remcom/XFdtd_7.3.0.2/xfgui.app/Contents/MacOS/xfsolver [options]
```

The following example is for Microsoft Windows XFdtd users. To follow along with this example, Macintosh and Linux users should substitute the suggested directory paths with paths that are suitable for their computing environment.

Example: Microsoft Windows users can use the Command Line tool to navigate to the Simulations folder of an Remcom-provided example project in the PROJECTS directory. Once there, they can instruct XFdtd to calculate all of the runs for that simulation. In this example, the `Patch.xf` project file will be used.

1. Launch the Command Line editor (**Start** > **Run** > **Cmd**)
2. Navigate to the 000001 directory within the Simulations directory.
   (`cd "C:\Projects\Patch.xf\Simulations\000001\"`)
3. Start XFsolver in verbose mode [-v] to run the simulation and display progress.
   (`"C:\Program Files\Remcom\XFdtd 7.3.0 (64-bit)\bin\Win64.NET2010\xfsolver" -v`)
4. Press the **Enter** key. The simulation should begin.

32-bit users would enter (`32-bit`) instead of (`64-bit`) in the above file path.

To use XFsolver on a specific run within a simulation, users should navigate to the specific run directory (For example, `Patch.xf\Simulations\000001\Run0007\`) and then use the command appropriate to their operating system.

For normal operation no [options] need be specified, but Table 11.1 lists several command line options that may be of interest.

Table 11.1: **Command Line Options**

Option	**Description**
`-h` `--help`	Display a help message containing program option and argument descriptions.
`-v` `--verbose`	Output verbose status messages as a simulation executes. Default value: false
`--version`	Output version information and exit.
`--license-diagnostics`	Output license diagnostics information and exit.

Table continued on next page...

Table 11.1: Command Line Options (continued from previous page)

`--sar-standard={n}`	Specify which version of the IEEE 1528.1 standard to use when computing SAR. Possible values are: 0 == Compute SAR using IEEE 1528.1 prior to Jan. 2010; and 1 == Compute SAR using IEEE 1528.1 ratified Jan. 2010. Default value: 1
`-o` `--output={s}`	Specify zero or more types of output to generate from the simulation and/or post-processing steps. Valid values are `uan` (output a UAN file for each Far Zone sensor), `cpuan` (output a circularly polarized UAN file for each Far Zone sensor), and `advanced-sar-stats` (output a file with advanced SAR statistics). No default value
`-t={n}` `--num-threads={n}`	Specify the number of threads to use for the simulation. Not all computational steps can use more than one thread, but this many threads will be used whenever possible. Default value: 1
`--post-process={s}`	Do not perform static field initialization or timestepping, but instead directly perform the specified post-processing step. Valid values are `sar-project` (post-process SAR sensors defined in the project when the simulation was created), `sar-request` (post-process SAR sensors defined as post-processing requests), `sar-any` (perform post-processing requests or project definitions if requests do not exist) and `farzone` (`sar` is deprecated and is equivalent to `sar-request`). Example: `--post-process=farzone`
`--post-process-off={s}`	Specifies whether or not to perform the specified post-processing step after timestepping. This option has no effect when the `--post-process` option is used. Value values are `sar`. Example: `--post-process-off=sar`
`--post-process-frequencies={n}`	A comma-separated list of which steady-state frequencies to compute if post-processing steady-state data. A single value of `-1` indicates to compute all frequencies. Currently supported only for SAR post-processing. Example: `--post-process-frequency=1,2,4`. Default value: -1
`--use-xstream`	Use XStream resources for computations when possible. If XStream cannot be used, CPU resources will be used instead unless `--use-cpu=false` (or `--no-use-cpu`) is specified. Default value: false
`--use-cpu`	Use CPU resources for computations when use of XStream is not possible or `--use-xstream=false` (or `--no-use-xstream`) is specified. Default value: true
`--xstream-list-devices`	Display a list of all detected XStream devices and exit. Default value: false

Table continued on next page. . .

Table 11.1: Command Line Options (continued from previous page)

`--xstream-memtest` Perform a memory test of XStream devices and exit. Use the `--xstream-use-devices` option to specify which devices to test. Default value: false

`-d={n}`
`--xstream-use-devices={n}`
Enumerate which devices to make available for XStream. Valid device numbers can be determined by using the `--xstream-list-devices` option. If a single value of -1 is specified, similar devices are detected and used as the available hardware devices. This option affects which devices are available for use by the `--xstream-use-number` and `--xstream-availability` options. Example: `--xstream-use-devices=0,3,4`
Default value: -1

`-n={n}`
`--xstream-use-number={n}`
Specify the number of XStream devices to use for the simulation. A value of -1 indicates to use all available devices (as determined from the `--xstream-use-devices`). If the number of specified devices cannot be used, simulating with XStream will fail. Example: `--xstream-use-number=2`
Default value: -1

`-a={s}`
`--xstream-availability={s}`
Specify whether or not to use any XStream device or only those that are currently `free`, meaning that there is not another simulation currently running on the device. Valid values are `any` and `free`. Example: `--xstream-availability=free`
Default value: any

`--fft-size={n}` An integer specifying the FFT size as n where 2^n is the FFT size, e.g. $n == 15, 2^{15} == 32768$. n must be from 15 to 23, inclusive. Default value: 15
`--load-static-fields` Load previously-saved static E-Fields before timestepping starts. Default value: false
`--save-static-fields` Save static E-Fields before timestepping starts. Default value: false
`--static-solver-only` Perform static E-Fields computation, save the E-Fields (`--save-static-fields` is implied) and exit. Default value: false
`--use-cached-license-location` Allow the use of the cached location from any previous successful license checkout by any Remcom product. Default value: true

11.2.3 Running the Calculation Remotely

Running a remote calculation is an alternative way to potentially speed up calculation time and free local computer resources for other operations. This section will explain how to copy files and launch the calculation engine on a remote computer.

If your XFdtd project is stored on a file system which is remotely accessible, you can avoid the process of copying files back and forth from the remote machine. Simply log into the remote machine, navigate to the simulation directory and launch the calculation engine.

Running simulations remotely

1. **Create the simulation.** After creating the simulation from the SIMULATIONS workspace window, click on the CREATE SIMULATION ONLY button to save the simulation to the SIMULATIONS folder where your XFdtd project is stored.

   ✓ You can find the path to this directory in the first line of text under the SUMMARY tab.

   ▶ See Section 11.1 for more about setting up a simulation in XFdtd.

2. **Copy the simulation files to the remote machine.** Copy the whole SIMULATION directory (you'll need to copy it recursively, since it contains subdirectories) except any existing `output` directories, which will exist if the simulation has already been run.

   For Windows users, it may be easiest to zip the entire simulation directory transfer the zipfile to the remote machine over an ftp connection.

   For Mac OS X and Unix users, package the files in a compressed archive with the following command:

   ```
   tar -czf inputFilesForXFSolver.tar.gz * --exclude='output'
   ```

   Then copy `inputFilesForXFSolver.tar.gz` to the remote machine and extract it using:

   ```
   tar -xzf inputFilesForXFSolver.tar.gz
   ```

3. **Run the simulation.** Log in to the remote machine, change directories to the SIMULATIONS folder, and run XFsolver for the project.

   ▶ See Section 11.2.2 for more about running the solver from the command line.

4. **Copy files back to local machine.** In order to view your results from the XFdtd GUI, after running the simulation on the remote machine, the output files will need to be copied back to the local machine. Starting with XFdtd® 7.3, all output files reside in the `output` folder of the Simulation directory and each of its Run directories.
   On Mac and Unix machines, the `tar`, `xargs` and `find` utilities make packaging only the output files easy. On the remote machine, change directories to the SIMULATIONS directory and run

   ```
   find. -type d -name 'output' \
   | xargs tar -czf outputFilesFromXFSolver.tar.gz
   ```

   For Windows users, zip the simulation folder and send it back to your local machine over the ftp connection and extract the files. Overwrite any old files with newer versions of the same file only in the `output` folders.

For Mac OS X and Unix users, copy the `outputFilesFromXFSolver.tar.gz` file to your local machine, change directories and extract it using:

```
tar -xzf outputFilesFromXFSolver.tar.gz
```

5. **Refresh results.** To make your results available from the XFdtd GUI, click REFRESH in the RESULTS workspace window.

Remote SAR post-processing

1. **Setting up post-processing.** After requesting post-processing from the RESULTS workspace window, in the subsequent dialog box, tell the application to not run the calculation right away.
2. **Copy simulation files to the remote machine** The same procedure used above for running the simulation should be used for SAR post-processing.
3. **Run SAR averaging on the remote machine.** On the remote machine, change directories to the SIMULATIONS directory, and execute XFsolver using the `--post-process=sar` command line flag.
   - See Section 11.2.2 for more about running the solver from the command line.
4. **Copy SAR results back to the local project.** The SAR results comprise the following files in the SIMULATIONS directory:

```
Run*/output/status
Run*/output/SteadyStateOutput/*/*gsar.gz
Run*/output/SteadyStateOutput/*/*.infosar1g
Run*/output/SteadyStateOutput/*/*.infosar10g
Run*/output/SteadyStateOutput/*/*.infoseq
Run*/output/SteadyStateOutput/*/*.*gssq
Run*/output/SteadyStateOutput/*/*.stats
```

   It is easiest to transfer *all* results as described in the previous section, but only the new SAR results can be copied using a similar technique.

   Notice that this command sequence retrieves all of the SAR averaging results, including those done with your initial calculation engine run.

5. **Refresh results.** After copying results to the local project, make your results available from the XFdtd GUI by clicking REFRESH in the RESULTS workspace window.

11.3 Calculation Considerations

The calculation portion of XFdtd may be quite lengthy depending on the application. A few guidelines are provided here for estimating computer resources, monitoring the progress of the calculation, and avoiding calculation instability.

- Also refer to Section 11.1.1 for information about defining proper termination criteria to ensure that the calculation has finished.

11.3.1 Computer Resources Estimation

XFsolver will give a time estimate while the calculation is running. This is recalculated every time XFsolver updates its status based on how much time passed since the last update and the remaining number of timesteps. It is not a completely accurate estimation since it does not consider data such as near-field samplings which may only be saved during certain portions of the calculation. Also, the estimate does not include any post-processing which may occur. Of special note are the SAR averages since they are computed in post-processing and may require a significant amount of calculation time.

The amount of memory required for a Simulation (and for each of its Runs) is provided in the `project.info` file in the Simulation and Run directories. It provides the following estimates:

- `estimatedMemory`: A best estimate at the amount of system RAM required for a simulation that is executed using CPU resources
- `minimumMemory`: A lower bound for the amount of system RAM required for a simulation that is executed using CPU resources
- `maximumMemory`: An upper bound for the amount of system RAM required for a simulation that is executed using CPU resources
- `estimatedXStreamMemoryGpuPart`: An estimate of the amount of GPU RAM required for a simulation being executed using XStream®
- `estimatedXStreamMemoryCpuPart`: An estimate of the amount of system RAM required for a simulation

The estimates above are also shown in the tooltip of the Memory Estimation graphic on the right-hand side of the Grid Editor and at the bottom of the Create Simulation dialog.

11.3.2 Multi-Processing Modules

XFsolver has the ability to do both threaded and Message Passing Interface (MPI) calculations in order to utilize computational resources that include multiple CPU cores. The latter requires additional license features, which can be obtained by contacting Remcom.

Multi-threaded simulations use shared memory and are intended for computers with multiple processors and/or cores. This will generally increase performance at close to N times the number of cores, though the overhead for the multi-threading routines can cause very small calculations to run slower when more than one thread is selected. Therefore, the multi-threaded option should not be used when the number of FDTD cells in the geometry is small (less than one million). (The number of cells is computed simply from the X, Y, and Z dimensions (in cells) of the geometry space.)

The number of threads to use for a simulation is defined in the QUEUE drop-down box of the SIMULATIONS workspace window, as seen in Figure 11.17.

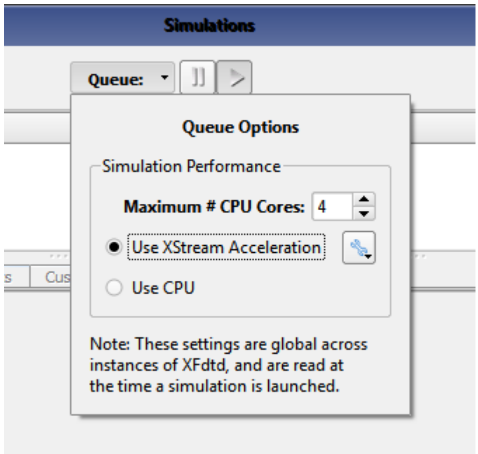


Figure 11.17: Specifying the number of CPU cores to use for a simulation.

Specify the number of processor cores that are to be used for the simulation next to MAXIMUM # CPU CORES.

✓ To specify the number of threads from the command line, the "`--num-threads=N`" option should be used, where `N` is the number of threads.

▶ See Section 11.2.2 for a summary of command-line options.

Simulations can be executed on cluster of computers that are connected by a high-speed network using MPI. At present, HPMPI and OpenMPI on Linux computers is supported. To run a simulation using the MPI version of XFsolver, see the documentation for the MPI library that is installed on all machines in the cluster. This documentation will give instructions on how to start an application using the MPI tools. XFsolver is launched from these tools using the command

```
{path to bin}/xfsolver -openmpi {other xfsolver options}
```

for the OpenMPI implementation, or

```
{path to bin}/xfsolver -hpmpi {other xfsolver options}
```

for the HPMPI implementation.

11.3.3 XStream Hardware Acceleration

XFsolver can also utilize hardware acceleration using its XStream® capability. The hardware is available in the form of graphics cards, which can replace existing graphics cards or act from a stand-alone computer. When this hardware is available, it may be enabled within the SIMULATIONS workspace window under the QUEUE drop-down box. To enable it from the command line, the "`--use-xstream`" option should be used.

11.3.4 Monitoring Simulation Progress

While XFsolver is running, its progress will be updated periodically. When launched from the XFdtd interface, the progress of the calculation will be shown in real-time in the OUTPUT tab located within the SIMULATIONS workspace window, as seen in Figure 11.18. When running from a command line, simulation progress will be printed to the window that was used to start the calculation if the "−v" option is used and will not be updated automatically in the OUTPUT tab. However, the output can updated manually in that tab at any time using the Update button as shown in the figure.

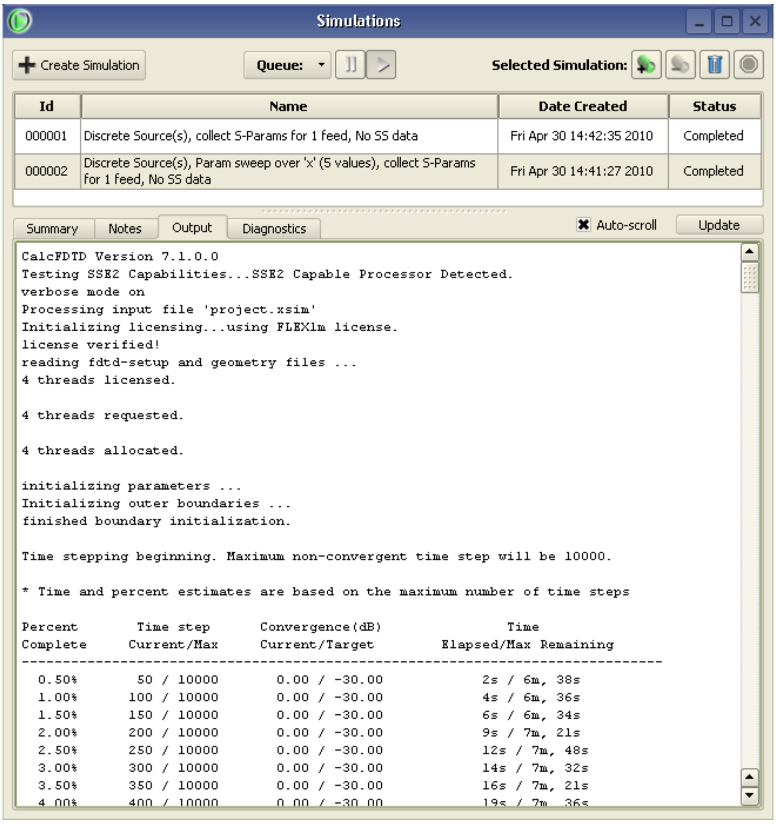


Figure 11.18: Monitoring simulation progress in the Output tab

11.3.5 Calculation Stability

Improper application of the outer boundaries can lead to unstable calculations. Typically the stability of a calculation depends on a few simple guidelines involving boundaries and source placement. Because an absorbing boundary condition like LIAO tries to simulate free space, it requires that a certain amount of continuity be present in the cells leading to the boundary. The cross section at a boundary must be the same for at least 10 cells in from the boundary.

✓ For many problems, a free-space border of 10-20 cells is the best way to ensure stability and accurate performance of the outer boundary.

Another rule for stability is that no source can be placed within 10 cells of an absorbing boundary. An unstable calculation is easily determined by viewing the line plots of time domain data or by viewing the field snapshots. When automatic convergence is enabled, XFsolver can automatically detect an unstable simulation and terminate it.

In some cases instability can be introduced by a frequency-dependent material. If such materials are used in the calculation and an instability results, it may be necessary to change the material parameters or reduce the calculation timestep.

Chapter 12

Viewing Output

In this chapter, you will learn...

→ how to read and filter your calculation results

→ the available tools to analyze 2-D (plotted) and 3-D (field display) results

Following an XFdtd calculation, the results may be reviewed in the RESULTS workspace window. The results that are available depend on the characteristics of the project, such as discrete sources, sensors, external excitations, and other project criteria specified in the SIMULATIONS window.

This chapter details how to review the available results. Some results will be in the form of numerical values. These are typically single-frequency results performed with a near-zone source. Other results will be displayed in the form of plots. There are several types of plots available to view results based on whether they are time-dependent, frequency-dependent, or angle-dependent. Finally, some results will be available to review as colored field displays. Broadband results collected by SURFACE sensors and SOLID sensors are viewed as individual "Field Snapshots" or "Field Sequences" (strings of snapshots). Three-dimensional far-zone fields may also be available depending on simulation criteria.

12.1 Results Workspace Window

Figure 12.1 shows the RESULTS workspace window.

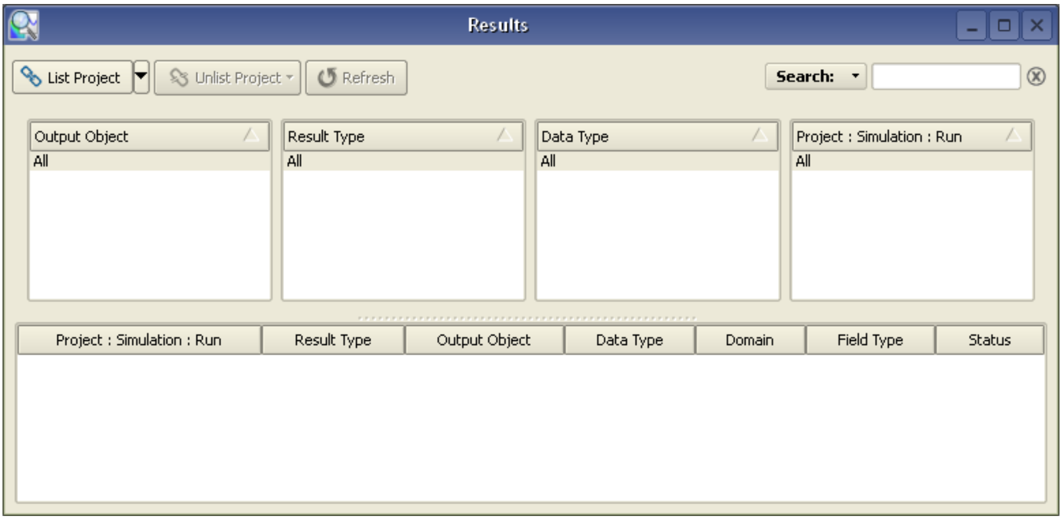


Figure 12.1: The Results workspace window

The RESULTS workspace window stores all of the data collected by the SENSORS during the calculation. Once the simulation is queued for calculation, the project will be listed in this window so that the results can be viewed. Additionally, the results of any other saved project can be loaded by pressing the LIST PROJECT button in the upper-left of the dialog. This makes it possible to view and compare the results of multiple projects without having to load several projects individually. Similarily, the results of any project may be closed by selecting the project and clicking the UNLIST PROJECT button.

By default, the data in the RESULTS workspace window will automatically update as more and more results become available while a simulation is progressing. Any results that are currently being plotted will also update at the same time. An update can also be forced by clicking on the REFRESH button when it is enabled.

If there are results from a large number of simulations loaded in the RESULTS BROWSER, XFdtd® can become unresponsive while the update is occurring. In this case, it will be helpful to uncheck the "Check for Updated Results" checkbox on the General tab of the Application Preferences dialog. After restarting XFdtd®, the REFRESH button (which will always be enabled) will need to be clicked in order to manually update results.

12.1.1 Filtering Data Results

It is possible to filter the data within the RESULTS workspace window by searching or by categorizing. The SEARCH box in the top right corner allows the user to search for the name of any field or data visible within the window. The results will automatically appear in the list below.

The user is also able to customize the four columns at the top of the window, which filter data according to the specified categories. The column headings are controlled by right-clicking on any of the current column headings and selecting one of the available categories. Each category is described in Table 12.1.

Table 12.1: **Data Filtering Options**

Data Filter	Description
Project ID	Displays the XFdtd project ID, which references the location of the loaded project in the file directory.
Project Name	Displays the project name indicated by the user.
Simulation ID	Displays the simulation ID that refers to the simulation's location in the file directory.
Simulation Number	Displays the simulation number that is automatically generated based on how many simulations have been created in a specific project.
Simulation Name	Displays the simulation name specified in the Simulations workspace window.
Run ID	Displays the run ID which references the location of the run in the file directory. Multiple runs are created in the case that a simulation collects data for more than one variable or location (i.e. during parameter sweeps, multiple ports, etc.).
Run Number	Displays the run number.

Table continued on next page. . .

Table 12.1: Data Filtering Options (continued from previous page)

Option	Description
RESULT TYPE	Filters results by type. Any of the following results may be viewed depending on the project, simulation criteria, and the type of data that was requested: Electric Field (E) / Magnetic Field (H) Conduction Current (Jc) / Magnetic Flux Density (B) Rotating B- / Rotating B+ Poynting Vector (S) / Voltage (V) Current (I) / Impedance Available Power / Input Power Instantaneous Power / Component Loss Reflection Coefficient / S-Parameters SAR (Specific Absorption Rate) / Maximum SAR Value Average SAR in Exposed Object / Average Power Net Input Power / Net Component Loss Net Available Power / System Efficiency Radiation Efficiency / Dissipated Power Dissipated Power in Tissue / Dissipated Power in Non-Tissue Dissipated Power per Material / Radiated Power Dissipated Power per Electric Material Component / Dissipated Power per Magnetic Material Component Dissipated Power Density / Axial Ratio Gain / Radar Cross Section Voltage Standing Wave Ratio (VSWR) / Active VSWR
SENSOR	Filters results according to its SENSOR, which refers to a specific sensor by name.
SENSOR TYPE	Filters results according to the type of sensor that collected the data. The sensor is referenced by its general type, rather than by its user-defined name (See SENSOR for filtering data by sensor name). The following is a comprehensive list of the sensors that may be listed within this filter: *Point Sensor* - Retrieves data from any POINT sensor in the project. *Surface Sensor* - Retrieves data from any SURFACE sensor in the project. *Solid Sensor* - Retrieves data from any SOLID sensor in the project. *Far Zone Sensor* - Retrieves data from any FAR ZONE sensor in the project. *Raw SAR Sensor* - Retrieves raw SAR data. *Averaged SAR Sensor* - Retrieves averaged SAR data. *Circuit Component* - Retrieves data from a CIRCUIT COMPONENT. *HAC Sensor* - Retrieves hearing aid compatibility data from HAC sensors. *System* - Retrieves ambient result data (not associated with a sensor object). *External Excitation* - Retrieves data on the EXTERNAL EXCITATION waveform. *Raw Steady-State Far-Zone Data* - Contains information which can be used to generate new steady-state far zone patterns after a simulation is run.

Table continued on next page. . .

Table 12.1: Data Filtering Options (continued from previous page)

PROJECT : SIMULATION : RUN	Displays the Project, Simulation, and Run name in one column.
DOMAIN	Filters data according to Time, Frequency or Discrete Frequencies domains.
FIELD TYPE	Filters data according to total-field or scattered-field.
STATUS	Displays whether a result is complete or still being calculated while the simulation is running. This status can be refreshed manually by pressing the REFRESH button or automatically by selecting AUTO-UPDATE RESULTS.
MISC	Displays query-specific information. For example, for the circuit component voltage when collecting S-Parameters, it could contain the active port. For S-Parameters, it could contain the S designation or simply the active port. For System numbers, it could contain the number (such as efficiency) or net input power.

Note that Rotating B is computed from Magnetic Flux Density (B) using the following convention:

$$B_x^+ = \frac{B\cdot\hat{v}+jB\cdot\hat{w}}{2} \qquad B_x^- = \frac{B\cdot\hat{v}-jB\cdot\hat{w}}{2}$$
$$B_y^+ = \frac{B\cdot\hat{w}+jB\cdot\hat{u}}{2} \qquad B_y^- = \frac{B\cdot\hat{w}-jB\cdot\hat{u}}{2}$$
$$B_z^+ = \frac{B\cdot\hat{u}+jB\cdot\hat{v}}{2} \qquad B_z^- = \frac{B\cdot\hat{u}-jB\cdot\hat{v}}{2}$$

12.2 Viewing Numerical Results

When a single-frequency calculation has been performed with a near-zone source (voltage or current), parameters such as input impedance, S-Parameters, VSWR and the reflection coefficient are displayed as numerical values rather than as line plots since the data only is relevant for the input frequency. Other numerical values are collected by means of a system sensor, which is automatically present in every XFdtd project.

12.2.1 Numerical Data Collected by XFdtd

Table 12.2: Results Displayed as Numerical Data

Sensor Type	Domain	Result Type
Port	Single Frequency	Available Power Current Impedance Input Power Reflection Coefficient S-Parameters Voltage VSWR Active VSWR
System	Single Frequency	Dissipated Power Dissipated Power in Non-Tissue Dissipated Power in Tissue Dissipated Power per Electric Material Component Dissipated Power per Magnetic Material Component Dissipated Power per Material Net Available Power Net Component Loss Net Input Power Radiated Power Radiation Efficiency System Efficiency

12.2.2 Power, Impedance, and Loss

XFdtd provides several quantities for individual ports and for the overall simulated system. These quantities are defined here for reference.

Available Power Available Power is the power which can be delivered by an active port given a conjugate impedance match between the source's impedance (Z_{S1} in Figure 12.2 and the simulation space (Z_{input}).

Net Available Power Net Available Power is the algebraic sum of the Available Power of all active ports.

Input Power Input Power is calculated as the power delivered by an active port into the simulation space.

Net Input Power Net Input Power is the algebraic sum of the Input Power of all active ports, and is therefore the total power delivered by the active port(s) into the simulation space.

Component Loss Component Loss is the power dissipated by an inactive port (Z_{L2} in Figure 12.2).

Net Component Loss Net Component Loss is the algebraic sum of the Component Loss of all inactive ports.

Radiated Power Radiated Power is the difference between the Net Input Power and the dissipative losses from conductive materials and resistive loads in inactive ports (the latter being Net Component Loss).

System Efficiency System Efficiency is the ratio of Radiated Power to Net Available Power.

Radiation Efficiency Radiation Efficiency is the ratio of Radiated Power to Net Input Power.

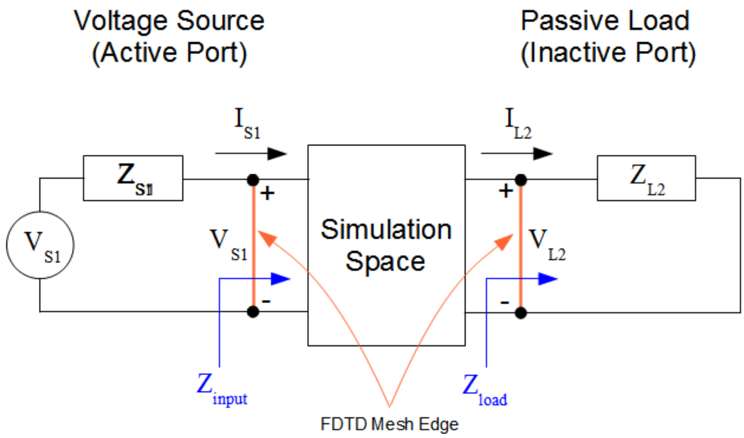


Figure 12.2: Active and Inactive Port schematic

When making calculations that include the input voltage, current and/or power in the calculation formulas, such as antenna gain or input impedance, these quantities are provided at the terminals of the mesh edge. Referring to Figure 12.2, the impedance at the port is the (complex) mesh edge voltage V divided by the (complex) mesh edge current I. The complex values are obtained from an FFT for a broadband calculation or from two samples of the voltage and current (electric and magnetic fields at the port mesh edge) for a sine wave excitation.

The sign convention of these quantities in XFdtd is defined as:

For *active* ports, the quantities are interpreted as "looking" from the port into the simulation space.

For *inactive* ports, the quantities are interpreted as "looking" from the simulation space into the port.

Therefore, if an active port is delivering power into the simulation space, the Input Power at that port will have a real positive part and the Input Impedance is the impedance the port sees looking into the simulation space. Likewise, an active port that is absorbing power from the simulation space will have Input Power and Input Impedance that have a negative real part. In contrast, an inactive port that is absorbing power from the simulation space will have a Component Loss and Impedance that has a positive real part. Note that an active voltage or current source can have an amplitude of zero– it is still considered active for the computations above.

Since Net Input Power is defined as the algebraic sum of the Input Powers of all active ports, switching a component from a Passive Load to an active source with zero amplitude in a simulation where multiple ports are active will, in general, affect both the Radiation Efficiency and the Net Input Power results. To clarify this, consider two different situations. An antenna composed entirely of perfect conductor that includes 2 ports, the first containing a 1 V source with a resistor, the second a passive port containing only the resistor. Assuming that some current flows in the passive resistor, the radiation efficiency will be less than 100%. If we repeat the same calculation, but with a 0 V source with a resistor replacing the formerly passive port, and with both sources set active, the radiation efficiency will now be calculated as 100% even though the 0 V source will have negligible effect on the antenna currents and radiation. This discrimination is done so that active ports may utilize lumped circuit elements to match to an antenna without changing the Radiation Efficiency, Impedance, and Gain, while passive elements may be added to an antenna with their effects included in the Radiation Efficiency and Gain results.

For active ports, Impedance is computed "looking" from the port into the simulation space (and thus the terminology "Input Power"– the power into the simulation space), whereas for inactive ports, Impedance is computed "looking" from the simulation space into the port (and thus a power "loss" from the space). For an active port that is absorbing power from the simulation space, this results in that port having a negative Re(Z), whereas a passive load that is absorbing power from the simulation space will have a postive Re(Z). For Input Impedance calculations ("into" the simulation space), the resistance, capacitance and inductance values of the active port will not be included in the input impedance. I.e., the input impedance of an antenna as calculated using XFdtd should not change regardless of any changes in the active port components (source/R/L/C). This is as it should be, since the antenna impedance is a function of the antenna geometry/materials and not of how the antenna is fed.

12.2.3 TDR / TDT Impedance Results

Time-domain impedance values can be generated for each port, allowing impedance discontinuities to be diagnosed in transmission lines as per the techniques of time-domain reflectometry (TDR). The impedance reported at the active port follows the convention

$$Z_{TDR}(t) = \frac{\int_0^t V(\tau)\,\mathrm{d}\tau}{\int_0^t I(\tau)\,\mathrm{d}\tau}.$$

The time-domain transmission (TDT) impedance seen at a passive port is analagously defined as

$$Z_{TDT}(t) = -\frac{\int_0^t V(\tau)\,\mathrm{d}\tau}{\int_0^t I(\tau)\,\mathrm{d}\tau}.$$

The sign convention of these results is consistent with frequency domain impedance results, such that resistive sources and passive loads will typically see positive TDR and TDT impedance values.

Note that TDR and TDT Impedance results are only available when a single active port is present in a run, and only when the excitation waveform is a Gaussian pulse or an Automatic waveform with a 0 Hz lower frequency bound.

12.2.4 S-Parameter Calculations

S-Parameter results (scattering parameters) are available for display for simulations configured with the COMPUTE S-PARAMETERS checkbox checked in the CREATE SIMULATION dialog (see Section 11.1.1). At discrete frequencies, the S-Parameter data is typically displayed in tabular form, while broadband S-Parameter data can be displayed on an XY Plot or Smith Plot. S-Parameter results can also be exported to several output file formats.

The Common Instrumentation Transfer Interchange (CITI) file format is used to exchange data between instruments and analysis tools. XFdtd can export S-Parameter results from different projects and runs to the same CITI file, provided that the maximum resolvable frequency and time step are the same each run. To export CITI file data, select one or more frequency domain S-Parameter results, then right-click and choose **Export** $>$ **Export to CITI file...** from the context menu.

The Touchstone file format can be used to export an NxN matrix of S-Parameter data from a single simulation to a `.sNp` output file (here, N is the number of ports whose data is written to the file). For this to be possible, each port under consideration must be designated as an ACTIVE PORT in the CREATE SIMULATION dialog's SETUP S-PARAMETERS pane. To export Touchstone file data, select one or more S-Parameter results from an S-Parameter matrix calculation, then right-click and choose **Export** > **Export to Touchstone file...** from the context menu. All of the selected results must have the same Domain (either "Frequency" or "Discrete Frequencies"), and all must come from the same Simulation. The output data contains a full NxN matrix of S-Parameter values for all N ports represented in the selection. Thus, if only the S13 result is selected in the result browser, the output file will contain S11, S31, S13, and S33 data.

Frequency domain S-Parameter data is automatically written to disk in '`.s`' file format by XFsolver. For example, a run with three ports where port 1 is active will typically generate the files '`project.s11`', '`project.s21`', and '`project.s31`' in the Run001 directory for that simulation. These results are computed at a default FFT size. To export '`.s`' data with higher resolution on the frequency axis, right-click one ore more Frequency domain S-Parameter results and choose **Export** > **Export to '.s' file...** from the context menu.

12.2.5 Power Scaling Factor

When viewing single-frequency results collected by the System Sensor, the user can specify a power scaling factor the calculation and determine the scaled result. For example, if the user wanted to know what the output power would have been with an input of one mW, they can enter this value in the Available Power (will be subject to mismatch loss) or in the Net Input Power (after mismatch loss) to see the overall effect. Similarly, any of the output results may be scaled to determine the effect on input and output. Clicking reset will return the values to the un-scaled state.

12.3 Viewing 2-D Plotted Results

There are three basic categories of 2-D plots in XFdtd, depending on the X-axis (abscissa) type:

- XY PLOTS - visualizes data in a Cartesian coordinate system.
- POLAR PLOTS - represents data with an angle-independent axis.
- SMITH CHARTS - displays complex data vs. frequency, such as S-Parameters and reflection coefficient.

The 2-D plots are viewed by right-clicking on the desired result type in the RESULTS workspace window and selecting CREATE LINE GRAPH, as seen in Figure 12.3.

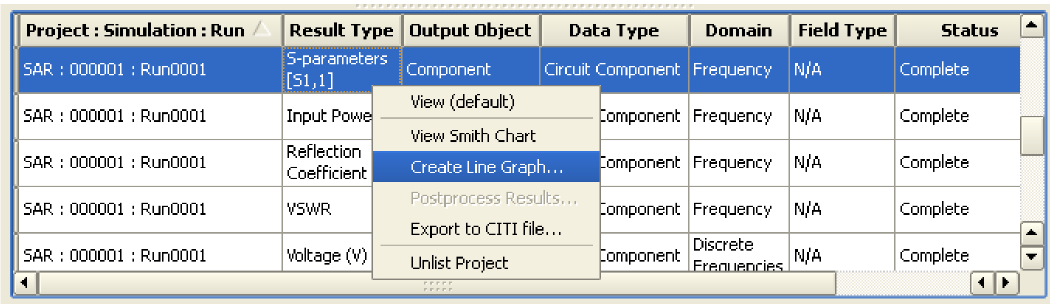


Figure 12.3: Results right-click menu

Before the graph is displayed, the user has the option of adjusting the graph properties such as the component, data transform, and complex part. The TARGET GRAPH option allows the user to view and edit plots that were previously created (from the same data selected in the Results window). Figure 12.4 shows samples of the create graph dialog for each plot.

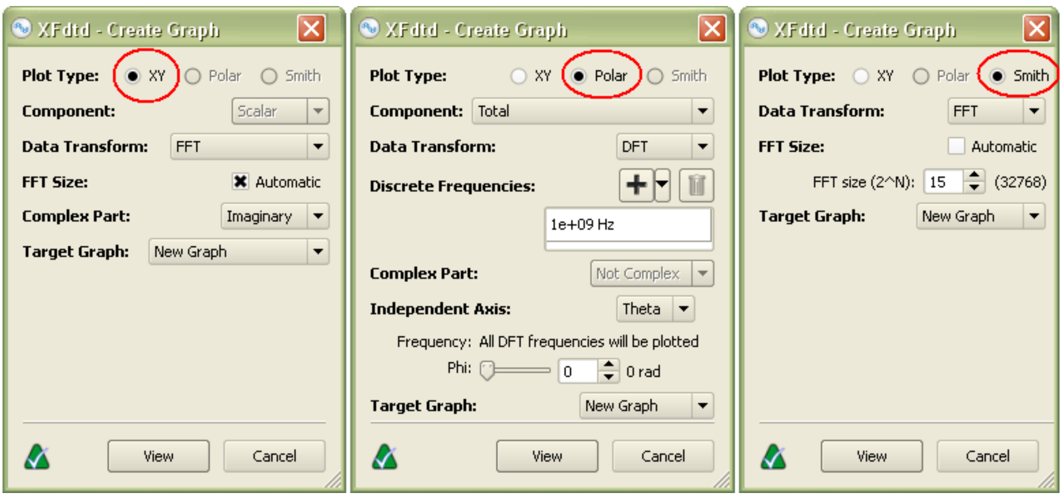


Figure 12.4: Create graph dialog for XY, Polar, and Smith plots

- For all calculations, the most important quantities are the time-domain plots of the fields in the problem space. Always perform a quick review of these values to ensure that a calculation has converged. Without convergence, most other results will be meaningless, particularly any plots converted to the frequency domain such as S-Parameters or impedance.
- Note that input impedance and S-Parameters may be plotted in rectangular form vs. frequency or as a Smith chart.

12.3.1 Plotted Data Collected by XFdtd

Table 12.3: Results Displayed as Plotted Data

Sensor Type	Domain	Result Type
Port	Broadband	Current Impedance Input Power Instantaneous Power Reflection Coefficient S-Parameters VSWR Active VSWR
Point, Surface, Volume	Broadband	E-Field (E) H-Field (H) B-field (B) Conduction Current (Jc) Poynting Vector (S)
System	Broadband	Net Component Loss Net Input Power
Far Zone Sensor	Broadband	Radar Cross Section Gain E-Field (E)
Far Zone	Single Frequency	Radar Cross Section Axial Ratio E Theta, E Phi Circular Polarization Ludwig-2 Az, El Ludwig-2 Al, Ep Ludwig-3
Far Zone Post Processor	Single Frequency	E-Field (E)

12.3.2 Customizing Plots

There are several ways to customize plotted data after it is opened from the RESULTS workspace window. The following sections detail the various tools that are available to modify plotted results.

Keep in mind that the tools available within the plot window depend on the type of graph you are viewing (i.e., XY, POLAR or SMITH plots).

Export Data

Select this option to export graphical data point values to a text file in a specified directory.

Export Image

Select this icon to save an image of the current plot to a specified directory.

The global shortcut key combination *Ctrl+Alt+F12* can also be used to initiate this action. A screen shot of the Geometry View or a Graph Window, whichever is “on top” in the application workspace, will be saved to a file. If neither the Geometry View nor a Graph Window is on top, no action will be taken.

> The global shortcut key combination *Ctrl+Alt+F11* is also defined to save a screen shot as described above to an automatically-named file in the folder defined in the Application Preferences under “Screen Capture.”

Copy Image to Clipboard

This tool takes a screen shot of the geometry as it is currently shown in the GEOMETRY workspace window and saves to to the clipboard for pasting into other applications.

The global shortcut key combination *Ctrl+Alt+F10* can also be used to initiate this action. A screen shot of the Geometry View or Graph Window, whichever is “on top” in the application workspace, will be copied to the clipboard. If neither the Geometry View nor a Graph Window is on top, no action will be taken.

Pan tool

Select this tool, and drag the mouse within the plot to pan to the desired view of the plot. Press the CTRL key and drag to pan along the independent axis. Press the SHIFT key and drag to pan along the dependent axis.

Zoom tool

Select this tool to zoom-in and to zoom-out of the plot. The mouse wheel as well as the right and left mouse buttons, can be used to perform zoom operations as described below.

Using mouse-wheel:

> Roll the center wheel of the mouse forward to zoom-out of both axes simultaneously.
>
> Press CTRL and roll the center wheel forward to zoom-out of the independent axis.
>
> Press SHIFT and roll the center wheel forward to zoom-out of the dependent axis.
>
> Roll the center wheel of the mouse backward to zoom-in to both axes simultaneously.
>
> Press CTRL and roll the center wheel backward to zoom-in to the independent axis.
>
> Press SHIFT and roll the center wheel backward to zoom-in to the dependent axis.

Using mouse buttons:

> Right-click and drag the the mouse anywhere in the plot window to zoom-out of both axes simultaneously.
>
> Press CTRL and right-click/drag to zoom-out of the independent axis.

Press SHIFT and and right-click/drag to zoom-out of the dependent axis.

Left-click and drag to define a rectangular view-window in the plot window to zoom-in to both axes.

Press CTRL and left-click/drag to zoom-in to the selected domain of independent axis values.

Press SHIFT and and left-click/drag to zoom-in to the selected range of dependent axis values.

Legend Visible

This button toggles the display of the legend within the graph.

Graph Properties tool

There are three tabs available for editing plot properties, as shown in Figure 12.5.

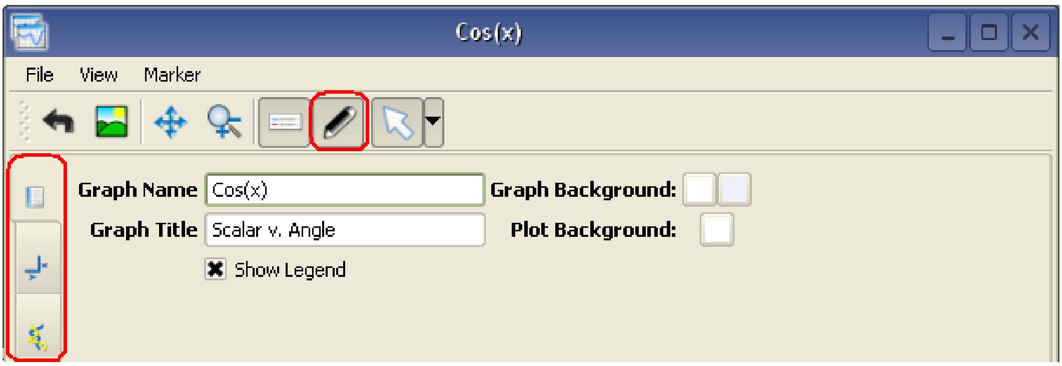


Figure 12.5: Tabs available for editing graph properties

Title properties Define the graph name and title as well as the background color of the graph in this tab. A checkbox also toggles the legend display on and off.

Axes properties Define the title of the axes and the limits of the axes in this tab. The AUTO checkbox may be selected to auto-select these limits.

If a graph only contains continuous data, the X-axis must be manually specified.

The UNITS drop-down menu is used to specify the units and apply a log scale, if necessary. Additionally, users can normalize plots in a 2-D graph by selecting **Relative Log** from the **Unit** drop-down list within the Units drop-down menu in the **X-Axis** section of the **Axes Properties** tab.

When **Relative Log** is selected, XFdtd will determine the maximum value, $maxValue$, from all plots in the graph and apply the following normalization

$$p'(x) = \log(p(x)/maxValue)$$

Where $p'(x)$ is the normalized plot of $p(x)$. $maxValue$ is automatically determined. It cannot be set by the user. A common application for this feature is determining the -3dB level for far-zone radiation patterns.

For certain Smith plots, you will also have the option of modifying the REFERENCE IMPEDANCE.

Figure 12.6 shows the axes properties editor for a 2-D XY graph.

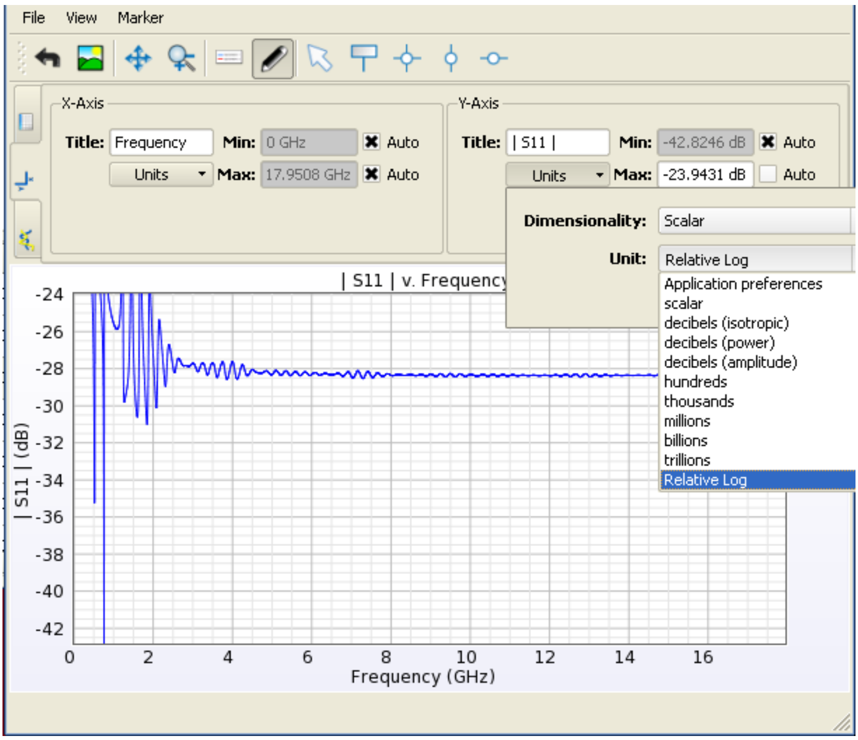


Figure 12.6: Editing the axes of graphs

Plot properties Define the characteristics of the plotted lines in this tab. A list of every dependent variable is listed with a customizable line color, line width, and line style. Any unwanted variables can also be deleted in this tab.

Selection tool

Select this tool to move, delete, or edit a marker's properties. To move a marker, click on the marker with the selection tool (once selected, it will turn yellow) and roll the mouse-wheel forward or backward to move an attached marker along its plot. Pressing CTRL and rolling the mouse-wheel will speed up the movement of the marker.

To delete any marker, simply select the unwanted marker, right-click and press DELETE MARKER. All markers may be deleted at one time by right-clicking anywhere in the plot area and choosing DELETE ALL MARKERS.

To edit a marker's properties, click on the desired marker so that it turns yellow, then right-click and select MARKER PROPERTIES. A window will appear with several different editing options, as seen in Figure 12.7. The location coordinates of the marker can be adjusted by manually typing in the desired values in the REQUESTED LOCATION section of the dialog box.

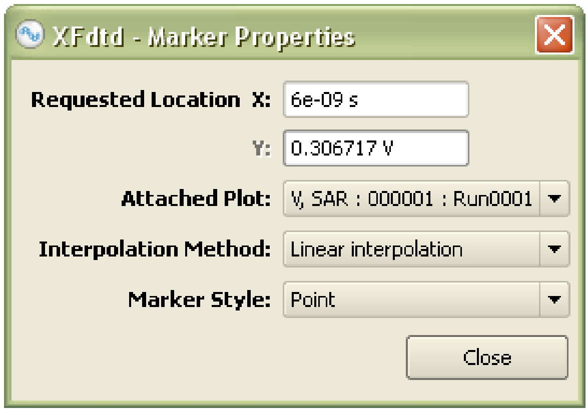


Figure 12.7: Marker properties dialog

If the coordinate box you desire to edit is disabled, select the appropriate option in the ATTACHED PLOT drop-down menu.

The marker may also be attached to a particular plot by selecting its name in the ATTACHED PLOT drop-down menu and selecting an INTERPOLATION METHOD. Depending on this definition, the marker will be shifted to the nearest point on the selected function or linearly interpolated based on the independent axis that is entered by the user in the REQUESTED LOCATION dialog box. Finally, the type of marker can also be redefined in this editor window in the MARKER STYLE drop-down list.

Additionally, this tool can be used to move or close the legend in the graphical space. (Any tool, however, can be used to perform this function.)

Any marker tool may be selected by clicking on its icon or selecting it from the MARKERS menu, as shown in Figure 12.8.

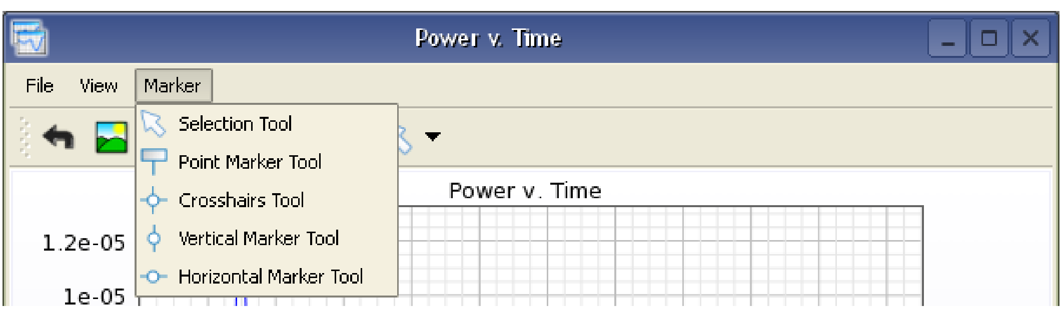


Figure 12.8: Marker drop-down menu

Point Marker tool

Select this tool to mark any point on the plot by clicking on the desired marker location. A marker with the location coordinates will appear above the point, depending on the type of plot:

- XY PLOT: (X-location, Y-location)
- POLAR PLOT: (radius, angle)
- SMITH PLOT: (real part of location, imaginary part of location, frequency)

When the mouse moves close to the plotted curve, it is snapped to the closest location on the interpolated line or sampled point. Holding the CTRL key will disable the snapping action, allowing a point to be placed anywhere. Holding the SHIFT key will snap the marker to sampled points only. Note that markers placed on sampled points are blue, and markers placed on interpolated points are black.

Crosshair Marker tool

Select this tool to mark the location of a single point by two intersecting cross-hairs. The marker is placed at the right edge of the plot. (Snapping actions are the same as the POINT MARKER described above.)

Figure 12.9 shows an XY PLOT with a Crosshair Marker (1.8873e-08 s, -6.0381e-06 A) and Point Marker (1.4636e-08 s, 1.0573e-05 A).

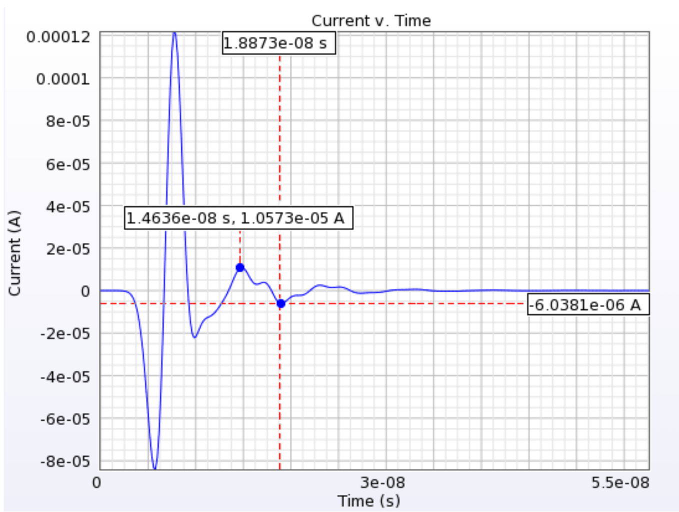


Figure 12.9: 2-D XY graph with Crosshair and Point markers (highlighted in red)

Figure 12.10 shows a polar graph with a Crosshair Marker along a radius.

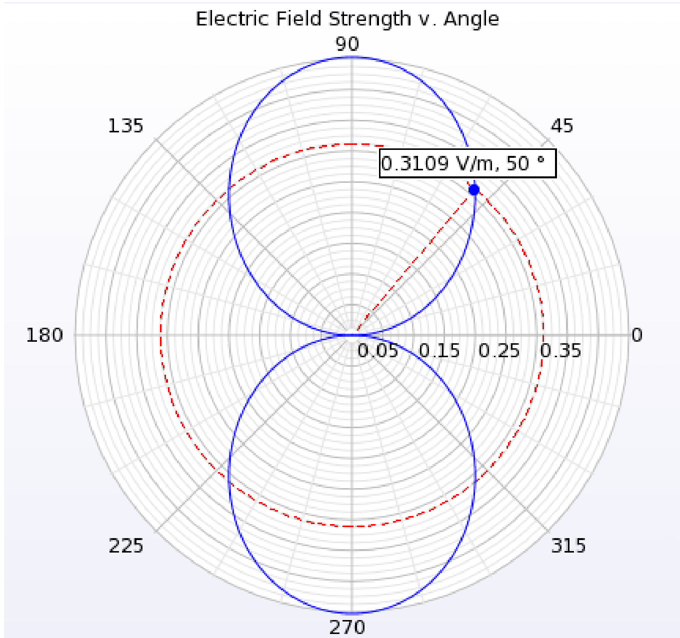


Figure 12.10: Polar graph with Crosshair marker (highlighted in red)

Figure 12.11 shows a smith graph with a Crosshair Marker along a radius.

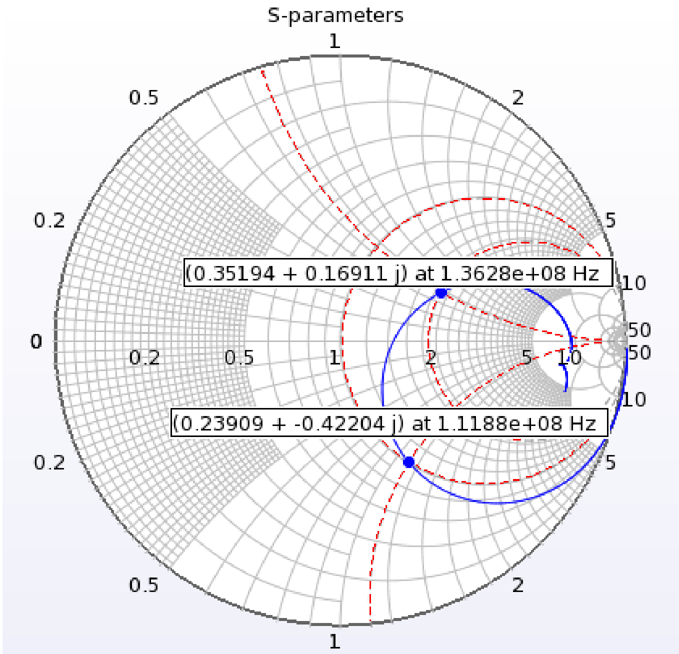


Figure 12.11: Smith graph with Crosshair markers (highlighted in red)

Vertical Marker tool

Select this tool to place a vertical line that intersects with the X-axis. The marker (Y = constant) will be placed along the top-edge of the plot area. (Snapping actions are the same as the POINT MARKER described above.)

Horizontal Marker tool

Select this tool to place a horizontal line that intersects with the X-axis. The marker (X = constant) will be placed along the right-edge of the plot area. (Snapping actions are the same as the POINT MARKER described above.)

Figure 12.12 shows a 2-D XY Graph with a Vertical Marker (at X = 1e-08 s) as well as a Horizonal Marker (at Y = 1.5e-05 A).

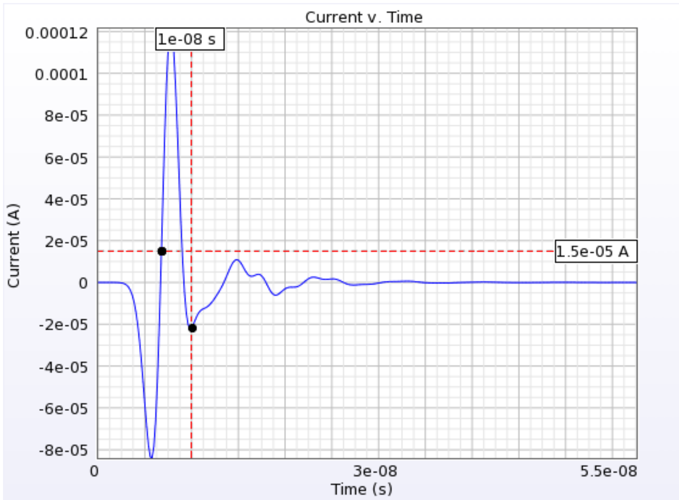


Figure 12.12: 2-D XY graph with Vertical and Horizontal markers (highlighted in red)

12.4 3-D Field Displays

Colored 3-D field displays may be viewed in slices of the geometry by saving either single-frequency or broadband field quantities. The fields may be viewed in with the geometry (solid or meshed) or by themselves.

Three types of fields may be viewed with the geometry:

- **Time Domain Snapshots** - available for any calculation, as they are simply snapshots of the near-zone fields at specific steps in time.
- **Complex Fields or Derived Quantities** (such as SAR) - available at specific frequencies.
- **Three Dimensional Far-Zone Fields** - can be requested either before the calculation with a far-zone sensor, or after a calculation through post-processing.

12.4.1 Data Collected With Near-Field Sensors

The field control panel enables the user to view 3-D results as a slice of data at a specific point in time, or as a sequence. The field sequences are movies of the individual slices as the fields progress with time in a particular slice of the geometry. Depending on what was saved, the results may display electric and magnetic fields, current densities, Poynting vectors stored for each direction (X, Y and Z) and a display of the combined magnitude.

In order to view single-frequency data, the COLLECT STEADY-STATE DATA check box must be checked in the SIMULATIONS workspace window (under the FREQUENCIES OF INTEREST tab).

Table **??** shows all of the available results for each sensor. Note that this depends on what data was saved during the simulation.

Table 12.4: Results Plots Available For Each Sensor Type

Sensor Type	Result Type	Time Domain	Discrete Frequency	Broadband
Point	E-Field (E)	✓		✓
	H-Field (H)	✓		✓
	B-field (B)	✓		✓
	Poynting Vector (S)	✓		✓
	Conduction Current (Jc)	✓		✓
	Scattered E	✓		✓
	Scattered H	✓		✓
	Scattered B	✓		✓
	Average Power			✓
Surface	E-Field (E)	✓	✓	
Solid	H-Field (H)	✓	✓	
	B-field (B)	✓	✓	
	Conduction Current (Jc)	✓	✓	
	Scattered E	✓	✓	
	Scattered H	✓	✓	
	Scattered B	✓	✓	
	Poynting Vector (S)	✓		
	Average Power		✓	
SAR (Avg)	SAR		✓	
HAC	E-Field (E)		✓	
	H-Field (H)		✓	
	HAC Max E-Field (E)		✓	
	HAC Max H-Field (H)		✓	

(Scattered) Electric Fields (E): magnitude, normal, or X, Y or Z components of the (scattered) electric field data at each cell edge.

(Scattered) Magnetic Fields (H): magnitude, normal, or X, Y or Z components of the (scattered) magnetic field data at each cell edge.

(Scattered) Magnetic Flux Density (B): magnitude, normal, or X, Y or Z components of the (scattered) B-field computed from the magnetic fields and associated permeability at each cell edge.

Average Power Density (SAV): magnitudes of the average power density computed from the electric and magnetic fields at each cell edge.

Conduction Current Magnitude (Jc): Conduction Current at an electric field cell edge.

SAR (Specific Absorption Rate): computed for each complete cell containing a lossy dielectric with a non-zero material density.

12.4.2 Data Collected by Far Zone Sensors

The RESULTS workspace window also displays the data collected for Far Zone sensors. Any far-zone request generated using XFdtd's post-processing engine (run separately from the calculation engine) is automatically added once the post-processing is complete. If steady-state far-zone data is enabled for the sensor, the results will include the E-field and axial ratio in the discrete-frequency domain. Gain or discrete-frequency radar cross section (RCS) will also be available when using a feed or external excitation, respectively.

Note that the polarization (Theta/Phi, Ludwig-2, etc.) is selected through the SETUP tab of the Field Editing Toolbar. See Section 12.4.3.4 for details.

If broadband far zone data was requested and the sampling interval for the data was set to one timestep (see the discussion in Section 10.3), time domain far zone fields will be available for each angle that was was defined. To plot that data, right-click on the E-Field result for the far zone sensor in the RESULTS workspace window and choose "Create Line Graph...". The "Independent Axis" in the dialog that appears should be listed as "Time". Near the top next to "Component", the field component to be plotted should be selected and near the bottom, the angle for which the data should be plotted should be chosen. The click "View" to see the selected time domain field.

Steady-state far-zone result data can be exported by right clicking on the far-zone-sensor result type and selecting **Export to UAN file**. The `.uan` file format is unique to Remcom products. It can be read directly by Wireless InSite and XGtd as an antenna file. The file contains far-zone pattern data for a specific frequency. If there is data for more than one frequency, the user must select the pattern data they want to download.

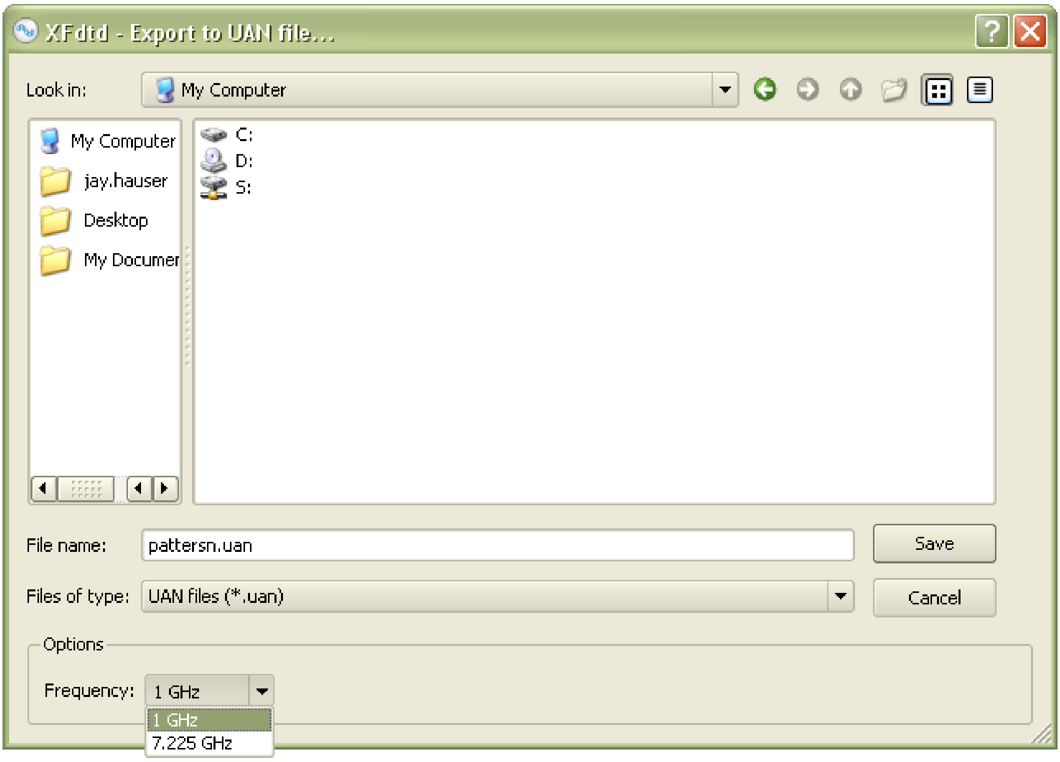


Figure 12.13: Exporting UAN far-zone data

12.4.3 Viewing 3-D Field Displays

XFdtd displays 3-D field data in the GEOMETRY workspace window. The following subsections discuss how to configure and analyze the display, using the SCALE BAR, the FIELD READER TOOL, rescaling, and the field editing toolbar.

12.4.3.1 The Scale Bar

The SCALE BAR, located at the top of the GEOMETRY workspace window, ”paints” the view with a range of colors which correspond to the range of values displayed. By default, the SCALE BAR color palette is shown in continuous mode within a default range of values. You can adjust these properties by right-clicking on the scale bar and selecting DISCRETE MODE, to change the palette to discrete colors, or AUTOMATIC RANGE, to change the range of values to that which is actually present.

There is also a PROPERTIES option under the right-click menu. Clicking this will bring up the SCALE BAR EDITOR. Using this editor, you can manually set the SCALE BAR limits, units and colors to your preference. Take note of several settings that may not be intuitive for the first-time user:

➔ Under the LIMITS section, it is possible to “clamp” values outside the defined scale bar limits to the nearest color. This makes it possible to view outliers that otherwise would not be colored with the Scale Bar.

➔ Under the SCALE section, when RELATIVE DB is selected, you can define its reference value or use the value automatically selected by XFdtd.

➔ Checking USE DISCRETE COLORS makes it easier to view contours in the 3-D results.

➔ FONT COLOR and BACKGROUND COLOR change the display of the SCALE BAR itself.

Figure 12.14 shows the Scale Bar Editor, and Figure 12.15 displays the SCALE BAR with the FIELD READER tool.

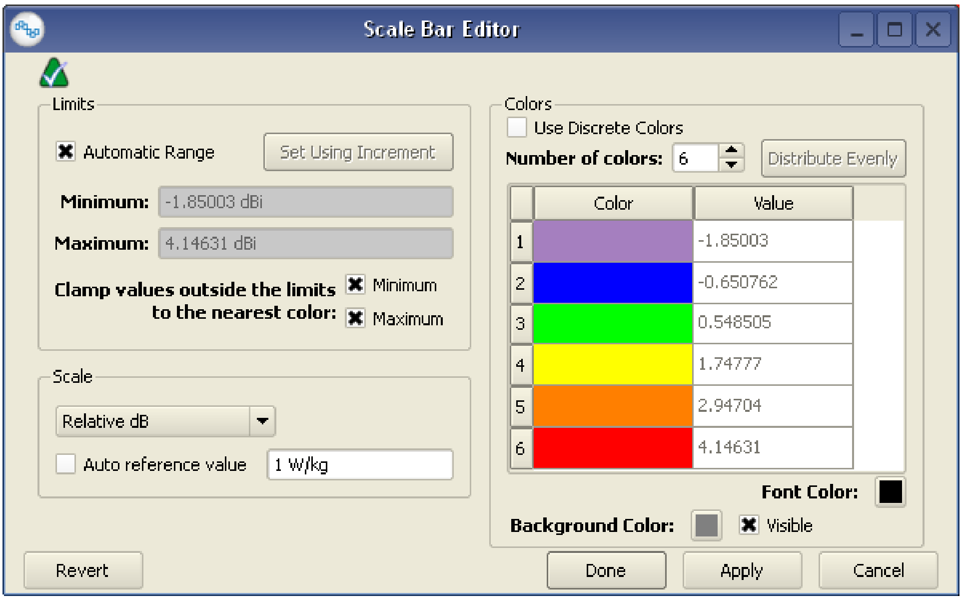


Figure 12.14: The Scale Bar Editor

12.4.3.2 The Field Reader Tool

The FIELD READER tool is located in the toolbar to the right of the GEOMETRY workspace simulation space. After selecting this icon, wheel the mouse over the geometry object to identify its field values. A marker in the SCALE BAR will display the nearest known field value to the location of the mouse. This location is represented by a small dot on the screen.

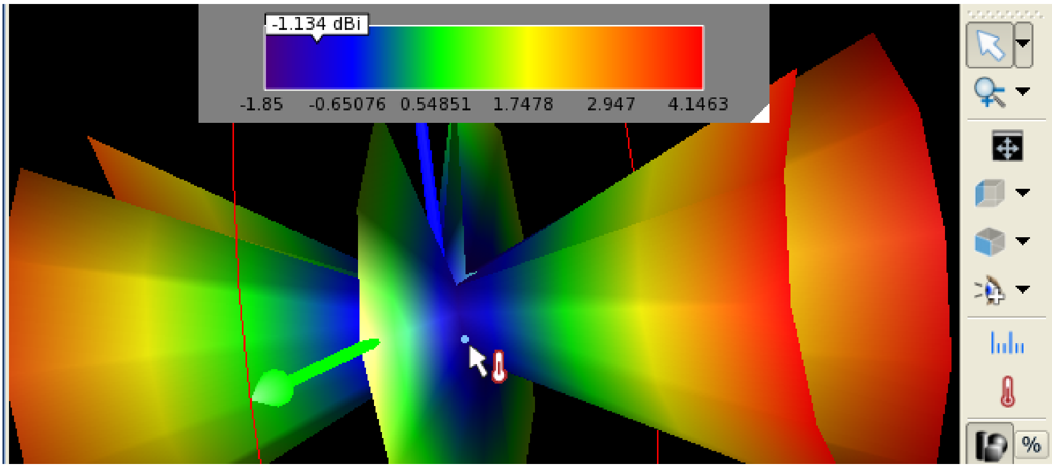


Figure 12.15: The Field Reader tool

12.4.3.3 Rescaling

In the RESULTS workspace window, filter the results in the Sensor column by SYSTEM. Double-clicking on results such as DISSIPATED POWER or SYSTEM EFFICIENCY will bring up the SYSTEM SENSOR OUTPUT dialog. Changing any value in this table will rescale the other values shown.

- ✓ The SHOW SCALED VALUES box must be checked to enable editing in the SYSTEM SENSOR OUTPUT dialog.
- Scaling only affects results which are in the discrete frequencies domain.
- When you change the scaling by editing values in the SYSTEM SENSOR OUTPUT dialog, you only scale the results for that particular calculation engine run at that particular DFT frequency. Any other tables of data and plots associated with this run and frequency will also be affected.

Figure 12.16 shows the SYSTEM SENSOR OUTPUT dialog for a sample SAR sensor project.

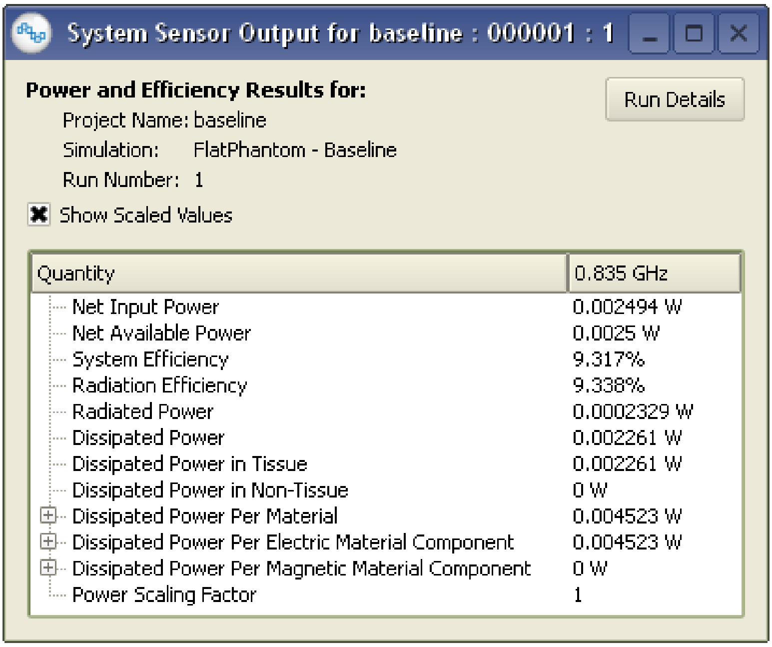


Figure 12.16: System Sensor Output dialog

12.4.3.4 The Field Editing Toolbar

Located at the bottom of the GEOMETRY workspace window, the field editing toolbar is used to configure the properties of the view. In the upper-left corner of this toolbar, a drop-down list will display any view that you have opened from the RESULTS workspace window. You can use the HIDE OTHERS and UNLOAD buttons to single out certain view(s) if necessary. Figure 12.17 shows a drop-down list of such results in the field editing toolbar.

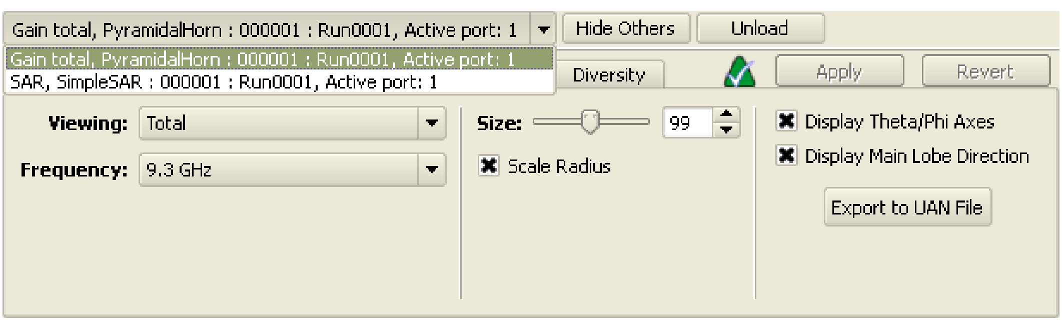


Figure 12.17: The results drop-down list of the field editing toolbar

The tabs and configuration options below the drop-down list will change depending on the active view. A comprehensive list of the available options are described below.

The SETUP tab

There are two main configurations for the SETUP tab, depending on the type of sensor you are evaluating.

Keep in mind that not all fields listed will be available for every view.

Surface, Solid and SAR Sensor configuration

Figure 12.18 shows the SETUP tab for a solid sensor configuration.

- SEQUENCE AXIS - Controls which independent variable which is swept while playing the field sequence from the SEQUENCE tab. Options are:
  - TIME – Each frame in the sequence corresponds to a single instant in time. This option is available for time-domain output data only.
  - PHASE – The sequence mimics the time-domain playback of steady state results captured at the frequency provided by the axis range controls. The sinusoidal response at this frequency is shown over the phase angle range [0, 2π]. Each frame in the sequence corresponds to a fixed phase angle within that range. This option is available for discrete frequency output data only.
  - X – Each frame in the sequence corresponds to a slice value of the X-coordinate. This option is available for Solid Sensor and SAR Sensor results only.,
  - Y – Each frame in the sequence corresponds to a slice value of the Y-coordinate. This option is available for Solid Sensor and SAR Sensor results only.
  - Z – Each frame in the sequence corresponds to a slice value of the Z-coordinate. This option is available for Solid Sensor and SAR Sensor results only.
  - FREQUENCY – Each frame in the sequence corresponds to a single steady-state data collection frequency. This option is available for discrete frequency output data only.
- DISPLAYED FIELD - Lists the available components of a vector field that can be shown in a plot
  - VECTOR MAGNITUDE – Displays the magnitude of the vector field
  - X – Displays the X-directed component of the field
  - Y – Displays the Y-directed component of the field
  - Z – Displays the Z-directed component of the field
  - VECTOR NORMAL – Displays the component of the field vector normal to the surface. This option is available for Surface Sensor data only
- COMPLEX PART - Used to transform complex-valued fields to real-valued data for plotting. This control is not displayed for time domain output data, for real-valued steady state output data, or when the SEQUENCE AXIS is set to PHASE.
  - SPECIFY PHASE – Allows a phase angle to be entered with the PHASE axis range specifier. For a complex field value F and specified phase ϕ, the real part of $Fe^{j\phi}$ is displayed.
  - MAGNITUDE – The magnitude of the complex field value is dipslayed.
  - REAL – The real part of the complex field value is displayed. This is equivalent to choosing SPECIFY PHASE with a PHASE value of 0 radians.

  - IMAGINARY – The imaginary part of the complex field value is displayed. This is equivalent to choosing SPECIFY PHASE with a PHASE value of $3\pi/2$ radians.

- Axis Range Specifiers - These controls enable the user to adjust the range of an independent variable over which the field data will be displayed. Note that the independent variable corresponding to the SEQUENCE AXIS is never represented with these controls.
  - X RANGE / Y RANGE / Z RANGE – Defines the range of the X-, Y-, or Z-coordinate over which field data will be displayed. Click the neighboring MODIFY TOOL button to edit the displayed range for this coordinate. Spatial coordinate ranges are only editable for Solid Sensor and SAR Sensor results.
  - TIME – Defines the single value of time at which the result data is displayed. Click the MODIFY TOOL button to choose a different value of time. Time specification is available for time domain output data only.
  - PHASE – Defines the phase offset within the steady-state sinusoidal response at which to display result data. This setting is only available when SPECIFY PHASE is selected as the COMPLEX PART, and behaves as described above.
  - FREQUENCY – Defines the steady state data collection frequency at which the result data is displayed. Click the MODIFY TOOL button to choose a different frequency. Frequency specification is available for discrete frequency domain output data only.
- DISPLAY MODE - Controls how XFdtd displays field data. The available display modes are VECTOR FIELD, POINT CLOUD, FLAT, and 3-D OFFSET.
- DECIMATION - Used to sample a subset of points for Solid sensors. The available display options are NORMAL, FINE, EXTRA FINE, SUPER FINE, and FINEST.

  By default, data is decimated at the highest level (NORMAL). Each succesively finer value will display more data. FINEST displays all data. The amount of memory and time required to render data is directly related to the amount of data displayed, which could be significant.

- SIZE FACTOR - Used to scale the indicator size for point clouds and vector display modes. This field determies the maximum size of the points.
- SCALE POINTS / SCALE VECTORS - Selects whether or not to scale the size of points (in POINT CLOUDS display mode) or vectors (in VECTOR FIELD display mode) according to the plotted data values.
- SHOW BOUNDING BOX - If selected, the bounding box displays. The bounding box enables the user to see the outer limits of the data being displayed The size and shape of the bounding box changes as users change the values in the DATA RANGE field.
- SURFACE RESOLUTION - Used to sample a subset of points for Surface sensors. This control is only available if the Surface sensor sampling method is configured as FIELD INTERPOLATION.
- ENABLE SCALING - Enables the scaling factor as described in Section 12.2.5

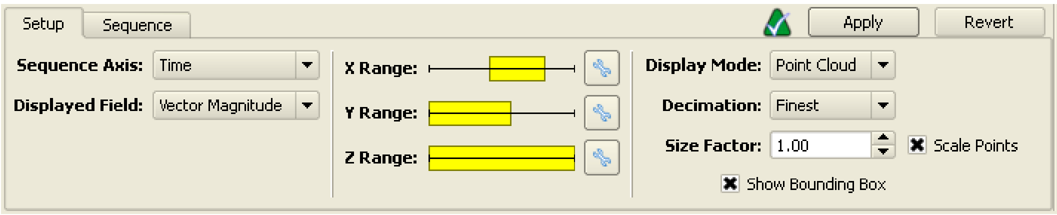


Figure 12.18: The Setup tab for a solid sensor configuration

Far-Zone Sensor configuration

Figure 12.19 shows the SETUP tab for a far-zone sensor configuration.

- In the leftmost section, specify what data to plot with the following controls:
  - VIEWING - When plotting E-fields, Gain, Realized Gain, Directivity, or Radar Cross Section, this control lists field polarizations available for display. Choices include VECTOR MAGNITUDE (for E-field and Radar Cross Section only), TOTAL (for Gain, Realized Gain, and Directivity only), THETA, PHI, ALPHA (LUDWIG-2), EPSILON (LUDWIG-2), ELEVATION (LUDWIG-2), AZIMUTH (LUDWIG-2), HORIZONTAL (LUDWIG-3), VERTICAL (LUDWIG-3), LEFT HAND CIRCULAR POLARIZATION, and RIGHT HAND CIRCULAR POLARIZATION.
  - COMPLEX PART - Used to transform complex-valued fields to real-valued data for plotting. This control is not displayed for real-valued output data.
    - SPECIFY PHASE – Allows a phase angle to be entered in a PHASE editor below. For a complex field value F and specified phase ϕ, the real part of $Fe^{j\phi}$ is displayed.
    - MAGNITUDE – The magnitude of the complex field value is dipslayed.
    - REAL – The real part of the complex field value is displayed. This is equivalent to choosing SPECIFY PHASE with a PHASE value of 0 radians.
    - IMAGINARY – The imaginary part of the complex field value is displayed. This is equivalent to choosing SPECIFY PHASE with a PHASE value of $3\pi/2$ radians.
  - FREQUENCY – Defines the steady state data collection frequency at which the result data is displayed.
- In the second section, configure how the plot is drawn:
  - SIZE - Controls the size of the plot on the screen
  - SCALE RADIUS - Determines whether the radius where each data point is drawn should be scaled according to the size of that value.
  - SHOW SCALED VALUES - Enables the scaling factor as described in Section 12.2.5
- The following options are present in the rightmost section:
  - DISPLAY THETA/PHI AXES - Controls the display of an axis indicator on the graph
  - DISPLAY MAIN LOBE DIRECTION - Shows the direction of maximum total gain
  - The EXPORT TO UAN FILE - Click this button to export far-zone result data (subject to any rotation specified on the COORDINATE SYSTEM tab) to a .UAN file, for use by Wireless InSite

or XGtd. Ranges for Theta and Phi angles to export must be provided on the Export UAN dialog.

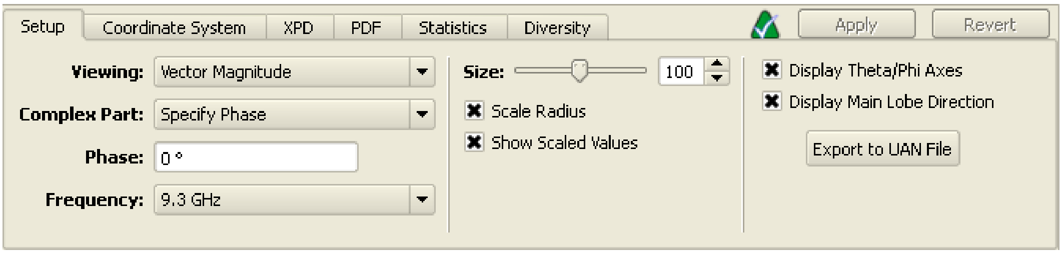


Figure 12.19: The Setup tab for a far-zone sensor configuration

The SEQUENCE tab

This tab is available when viewing output data from SAR, Solid, or Surface Sensors. Configure the settings here to "play through" a simulation. Figure 12.20 displays the SEQUENCE tab.

- MINIMUM/MAXIMUM - values entered here control where the sequence starts and stops.
- SHOWING - shows the currently plotted value of the sequence axis
- The scroll bar at the bottom of the window can be used to view any point along the sequence.
- The simulation buttons allow you to move through the sequence:
  - The and buttons play the sequence forwards and backwards, respectively.
  - The button pauses a playing sequence.
  - The and buttons fast forward and rewind the sequence, respectively.
  - The and buttons jump to the end or beginning of the sequence, respectively.,
- COMPUTE BOUNDS - when this box is checked, XFdtd will automatically compute the bounds before the sequence and adjust the SCALE BAR accordingly.
- - performs the COMPUTE BOUNDS calculation when pressed
- The ADVANCED tab
  - Checking the AUTO-REPEAT SEQUENCE box will automatically replay the sequence continuously.
  - The STEP SIZE determines whether each index of the sequence axis is shown while playing the movie. A value of 1 means show every index, a value of 2 means show every other index, and so on.
- When the SEQUENCE AXIS on the SETUP tab is set to PHASE, an additional INCREMENT control is present. This defines the difference in phase angle between successive frames in the sequence.

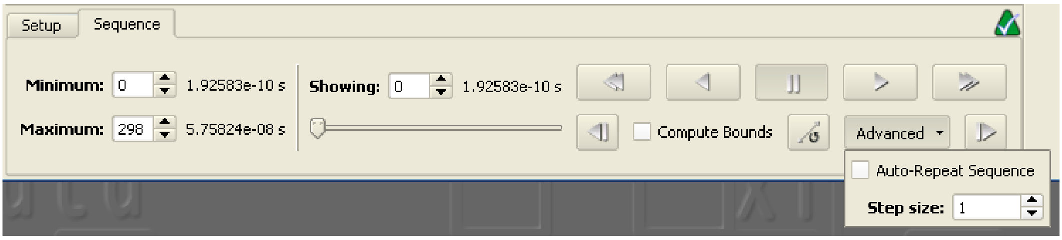


Figure 12.20: The Sequence tab of the field editing toolbar

The Coordinate System tab for far zone resutls

The Coordinate System tab, shown in Figure 12.21, allows the far zone coordinate system to be manipulated. The leftmost controls choose the location of the Center Point of the spherical coordinate system used to plot this far zone result.

- Enter the (X, Y, Z) coordinates of the center, or
- Choose the center using the Pick tool.
- The Reset Center On drop-down allows you to center the plot at the center of the geometry or at the location of any of the ports in your active project.

The Rotations toolset adjusts the orientation of the far-zone results by setting the Up Vector of the view. Changing the Up Vector (which is the Z-axis by default), will set the reference point for the spherical coordinate systems (e.g., theta/phi, alpha/epsilon, etc.) This affects field plots of single polarization components, partial power efficiency and power computations, mean effective gain computations, and antenna diversity. This is useful in cases where the geometry is aligned with an axis of the computation grid, but the real-world position of the geometry is not axis-aligned.

- The Up Vector Presets list allows you to select a pre-defined Up Vector in the X-, Y-, or Z-direction.
- The Pick Up Vector tool allows you to select geometry within the simulation space to create your own Up Vector.
- The button will undo an operation.
- The button will redo an operation.
- The button will reset the Up Vector.

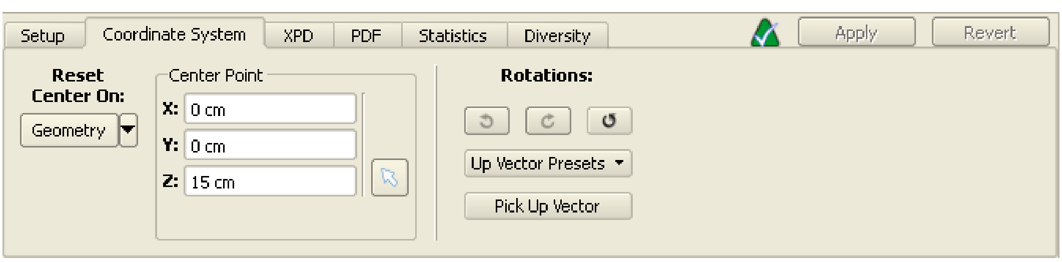


Figure 12.21: The Rotations tab of the field editing toolbar

The XPD tab

The XPD tab allows the cross-polarization discrimination (XPD) value used in mean effective gain and antenna diversity computations to be set. The XPD value gives cross-polarization discrimation of the incident multipath field. It can be expressed by the ratio of time-averaged vertical power to time-averaged horizontal power in the fading environment. A linear value or a decibel value can be entered here.

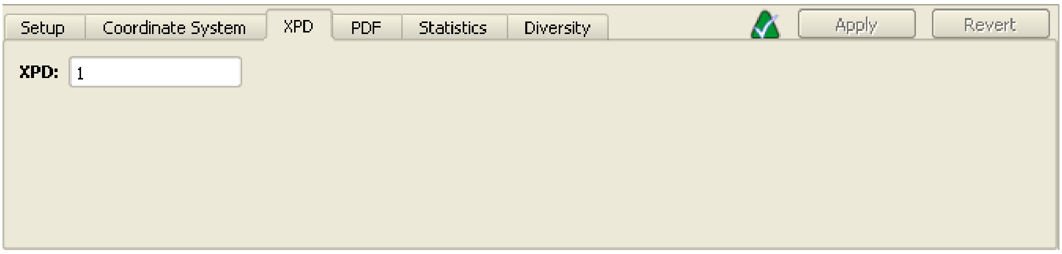


Figure 12.22: The XPD Tab

The PDF tab

The PDF tab controls the Probability Density Function (PDF) settings used in the computation of mean effective gain and antenna diversity far zone results. Figure 12.23 shows the PDF tab. PDF is used to model communciation channels. This function "weights" different directions in the far zone sphere such that certain far zone directions are taken into account more than others in the mean effective gain and antenna diversity calculations. A graphical depiction of the PDF is displayed in place of the field plot while the PDF tab is open. XFdtd supports several varieties of PDF types:

- UNIFORM VALUE – In this case, a constant weighting is used over the far zone sphere.
- GAUSSIAN – This is most commonly used as a model of channels. THETA MAX and PHI MAX are the angles of the maximum incoming field. SIGMA THETA and SIGMA PHI are the standard deviations of the distribution in the θ and ϕ directions, respectively.
- USER-DEFINED – This allows the user to import the probability density data from a text file. The file must contain a grid of theta/phi angles and the corresponding non-negative probability density at each angle. For example, to define the PDF at ten theta angles with twenty phi angles, the text file data would look something like the following:

```
theta0radians phi0radians pdfAtTheta0Phi0
theta0radians phi1radians pdfAtTheta0Phi1
...
theta0radians phi19radians pdfAtTheta0Phi19
theta1radians phi0radians pdfAtTheta1Phi0
...
theta9radians phi19radians pdfAtTheta9Phi19
```

By default, the same Probability Density Function is applied to both theta polarized and phi polarized field data. This is the case whenever the PDF TYPE control for the PHI POLARIZED PDF reads "Use Theta Polarized PDF". Changing this control allows the Theta and Phi PDFs to be defined independently.

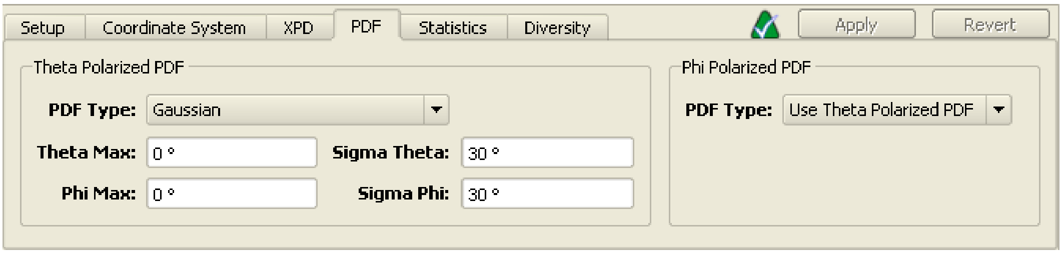


Figure 12.23: The PDF tab in Gaussian mode

The STATISTICS tab

There are two main configurations for the STATISTICS tab, depending on the type of sensor you are evaluating.

Far-Zone Sensor Statistics

Figure 12.24 shows the STATISTICS tab for a far-zone sensor configuration.

When POWER/EFFICIENCY is selected, power and efficiency computed for the far zone data (with rotations applied) is displayed. The option selected in the drop-down box determines the set of angles that are considered for this data.

- FULL PATTERN specifies the use of a full far zone sphere.
- UPPER HEMISPHERE specifies the use of theta = $[0, 90°]$ and phi = $[0, 360°]$.
- OPEN SKY specifies the use of theta = $[0, 80°]$ and phi = $[0, 360°]$.
- PARTIAL PATTERN allows the user to specify an arbitrary solid angle. These angles are defined relative to the coordinate system defined under ROTATIONS.

The OTHER STATISTICS option presents a list of additional statistics.

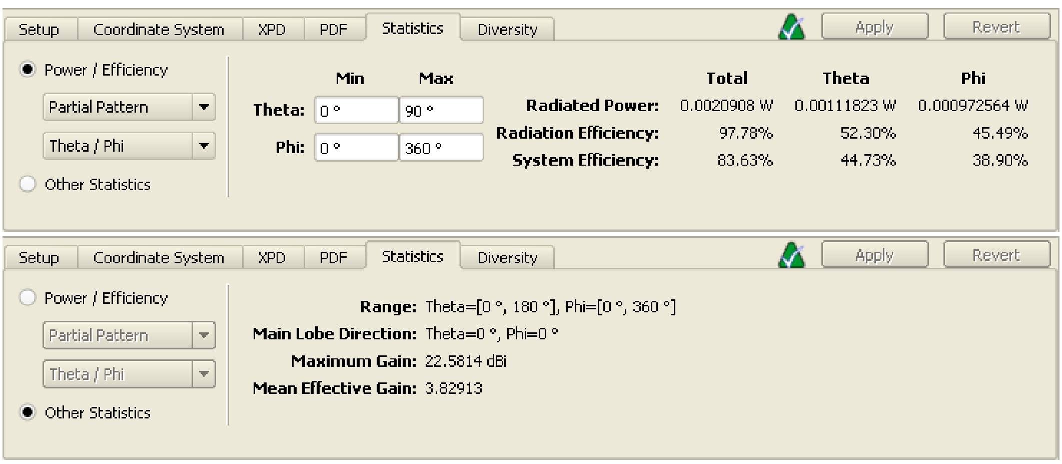


Figure 12.24: The Statistics tab for a Far-Zone sensor, showing Power/Efficiency and Other statistics

SAR Sensor Statistics

Figure 12.25 shows the STATISTICS tab for an SAR sensor configuration. This tab is available when looking at SAR data for both the SAR SENSOR and the SAR AVERAGING SENSOR.

- MAXIMUM LOCATION displays the global coordinates of the location where the SAR maximum values are found for the full-sensor and for the currently displayed slice.
- The MAXIMUM VALUE fields give the maximum SAR value detected over the entire sensor and over the range of the sensor data that is currently being displayed. Depending upon what type of SAR was chosen for plotting in the RESULTS browser, the displayed values are either the maximum raw SAR values, the maximum 1-gram Averaged SAR values, or the maximum 10-gram Averaged SAR values. A value entered here affects the power scaling for this run and frequency, in the same manner as adjusting the POWER SCALING FACTOR in the System Sensor Output dialog.
- The AVERAGE RAW SAR VALUE displays the average raw SAR value over the sensor. This is computed directly from the mass of and power dissipated by the tissues within the sensor region. An adjustment entered here affects the power scaling in the same way as adjusting the MAXIMUM VALUE.
- The VIEW SLICE WITH MAX VALUE button takes you to the planar slice of SAR data which contains the maximum value.
- The VIEW ALL SAR STATS button opens a separate window displaying the statistics for all the SAR sensors defined in the project together in a table, as shown in Figure 12.26. In addition to the quantities above, the mass of and power dissipated by tissues within the sensor region are displayed. In this window, the scaling factor can be adjusted, which affects all SAR values in this window, including data displayed in the SAR Sensor Statistics tab. The scaling factor and SAR statistics values will also automatically update when the user manually changes (rescales) any other statistics available through the RESULTS workspace window, such as power and efficiency values.
  - ▶ See Section 12.4.3.3 for more on rescaling values in the System Sensor Output dialog.

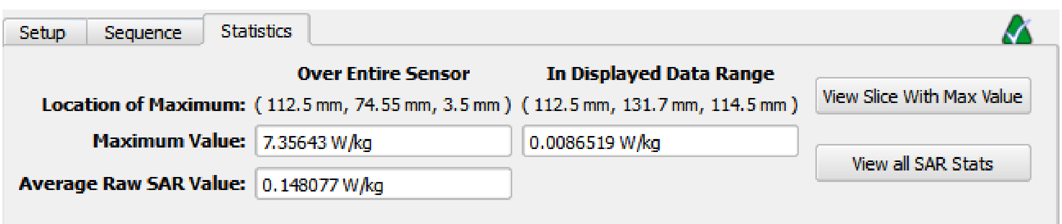


Figure 12.25: The Statistics tab for an SAR sensor

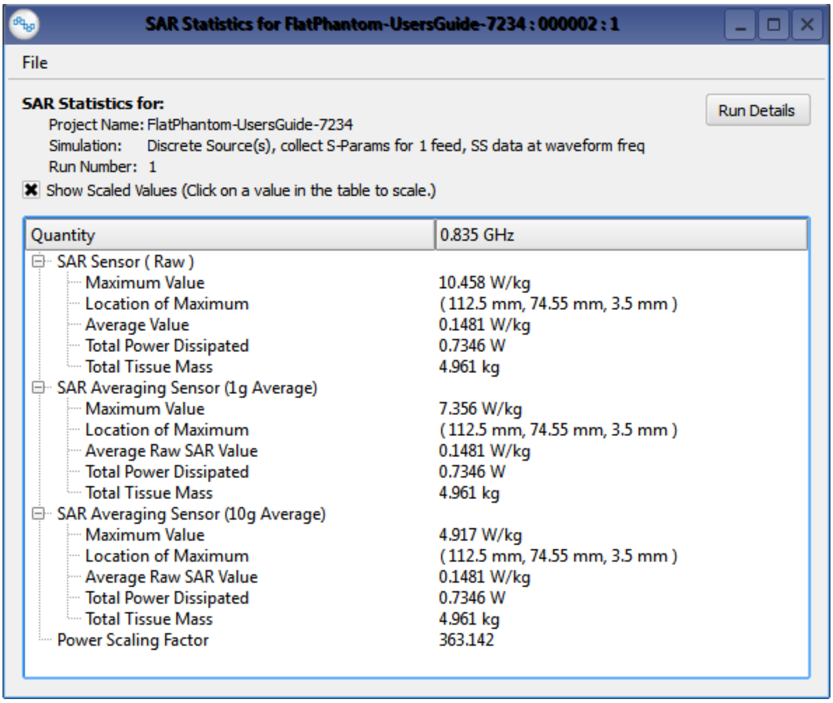


Figure 12.26: The complete list of SAR statistics

The DIVERSITY tab

The DIVERSITY tab is used to compute antenna diversity metrics between two far zone patterns. To perform the diversity computations, you must load the data for both far zone patterns in the RESULTS workspace. Then go to the diversity tab for one of the two patterns. Figure 12.27 shows the DIVERSITY tab.

- The PATTERN 2 drop-down list allows you to select the far zone result which your current far zone result is compared against. The only choices listed here are results which have been loaded into the RESULTS workspace window.
- The statistics next to FROM 3-D PATTERN show the diversity computation for the full far zone sphere. The statistics next to FROM 2-D PATTERN show the diversity computation at theta = $90°$.
- Under DIVERSITY OPTIONS, you can set the ANTENNA 1 PHASE and ANTENNA 2 PHASE controls to specify the phase difference between the two far zone patterns.

Note that the XPD and PDF settings under the PDF tab impact this computation, as well as the settings under the ROTATIONS tab.

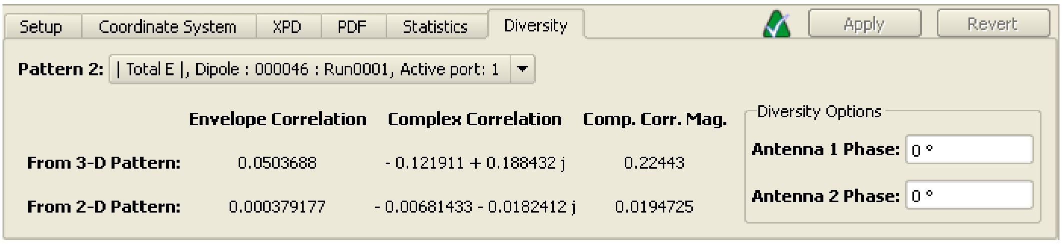


Figure 12.27: The Diversity tab of the field editing toolbar

The HEARING AID COMPATIBILITY tab

Figure 12.28 shows the HEARING AID COMPATIBILITY tab.

The HEARING AID COMPATIBILITY tab allows wireless device manufacturers to determine how well their wireless devices can be expected to perform with a hearing aid given the amount of electomagnetic interference generated by the device. The HAC Sensor feature in XFdtd causes XFsolver to save E- and H-field frequency-domain data in a plane where the hearing aid should be located. XFdtd can then use that field data to perform the M-rating computation as described in the IEEE HAC standard (ANSI C63.19-2006). The M-rating range contains five results (No Rating [worst] - M4 [Best]). The ”No Rating” value may occur if too much electromagnetic interference is generated.

After XFsolver runs, the sensor data is further processed according to the criteria in the standard. The user can determine if the handset will pass based on the rating in the **M-Rating** field.

This tab, as the name suggests, is only available when viewing HAC sensors.

- FREQUENCY - Defines the frequency of interest for the hearing aid.
- DISPLAY MODE - The available diplay modes are POINT CLOUD and FLAT. If POINT CLOUD is selected, the SCALE POINTS checkbox appears. This checkbox enables XFdtd] to scale the points according to the values of the data.
- SHOW SCALED VALUES - When checked, the displayed results are scaled by the scaling factor entered by the user on the system sensor output table.
- SURFACE RESOLUTION - Controls the resolution at which field data is shown for the sensor. The higher the resolution setting, the more data points are shown (and the closer they are together).
- M-RATING - Displays the M-rating of the sensor, or the suitability of the wireless device when used with a hearing aid.
- THRESHOLD - The maximum allowed field value given the current M-rating. The tooltip for this value provides further information about the M-Rating threshold values.
- BAND - Displays the band type. Bands enable the user to create different sets of parameters for the IEEE HAC computation and store them with the project. Under the EDIT button, you can configure the properties of the band, or define a pre-set list of properties using the MANAGE PRESETS button. An editor dialog will appear where you can add a new preset or import presets from a text file. The format of a HAC band definition file is as follows:
  - Lines in the file starting with **#** or **!** are comments and are ignored.

– Otherwise, the line must contain six semicolon-delimited items in this order:

- **Name**: The user-supplied name for the HAC band preset.

- **Probe modulation factor**: Used to scale the measured E-field and H-field maximum values to take into account the effects of the particular input waveform.

- **E-field probe calibration factor**: Used to scale the measured E-field values to take into account the effects of the user's labratory E-field measurement probe (i.e., the probe attenuates signals by a known amount at a given frequency).

- **H-field probe calibration factor**: Used to scale the measured H-field values to take into account the effects of the user's labratory H-field measurement probe.

- **Articulation weighting factor, in dB (must be either 0 or -5)**: Controls whether a -5 dB modification is applied to the threshold E- and H-field values for the M1 through M4 ratings. Checking the box lowers the threshold values, thus making it more difficult to achieve a given M-rating. Whether or not XFdtd should make this -5 dB adjustment is based on the transmission protocol as defined by the HAC standard.

- **Convert peak field values to RMS**: If enabled, the maximum E-field peak values are converted to RMS values (i.e., by multiplying by $\sqrt{2}/2)$ before being used in the HAC M-rating computation.

An example of this format is as follows:

```
!  This is a comment
Band 1 Name; 1.176; 0.9; 0.8; -5; 0
Band 2 Name; 1.0; 0.92; 0.33; 0; 1
```

- The 3x3 GRID REGION - Shows the maximum field values in each of the 9 squares of the HAC grid. The double-line on the top-level of the readout helps you orient in reference to the white grid on the displayed field (it also has a double-line in the top-left corner). Three of the cells in the 3x3 grid will have a dotted red border around them. These are the three cells chosen to be excluded from the HAC rating computation as allowed by the IEEE HAC standard. Also, any cells with red text are cells which exceed the field value threshold amount for the current M-rating.

Figure 12.28: The Hearing Aid Compatibility tab of the field editing toolbar

12.5 Post-Processing

Post-processing gives the user the ability to perform additional computations after the calculation engine run is finished. Three types of post-processing are available.

12.5.1 Far Zone Post-Processing

Far-zone post-processing is enabled by checking the SAVE DATA FOR POST-SIMULATION FAR ZONE STEADY-STATE PROCESSING box under the FREQUENCIES OF INTEREST: DATA STORAGE tab of the SIMULATIONS workspace window. When this configuration is set, there will be an entry for RAW STEADY-STATE FAR ZONE DATA in the RESULTS workspace window. Double-clicking this entry will allow you define a new far zone geometry at which the steady-state far zone pattern will be computed. Edit the geometry exactly as you would edit a far zone sensor. When you click DONE in the far zone sensor editor, the **Far Zone Post Processing**window appears. Select one of the following options.

- **Synchronous**: Far Zone post processing is done via the GUI. Note that computation time increases for a larger computation grid, and increases as the number of desired output far zone angles increases. The calculation engine can generally perform this computation much faster than the GUI if any of the following is true:
  1. XStream acceleration can be used (i.e. the computation is done on the GPU)
  2. Multiple processors can be used to do the computation multi-threaded
  3. MPI can be used
- **Queued**: Automatically begins far zone post processing. This option is not recommended if the following is true:
  - Running XFdtd with MPI
  - Running XFsolver from a different computer or from a different user name
- **Manual**: Enables you to run the Far Zone post processing engine manually

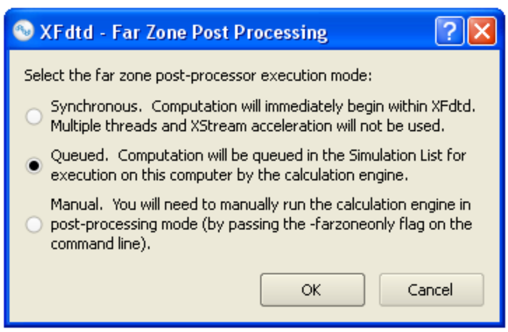


Figure 12.29: Far Zone post processing engine options

After selecting the appropriate option, click the **OK** button. Depending on your selection, the computation of the far zone pattern may begin. Once complete, you can find a new result in the Results workspace window which corresponds to the newly defined far zone pattern.

▶ Section 11.1.1 describes the FREQUENCIES OF INTEREST tab.

12.5.2 SAR Averaging Post-Processing

Right-clicking on any Raw SAR Sensor entry in the RESULTS workspace window will show an option in the context menu to POSTPROCESS RESULTS. This allows you to perform SAR averaging. Edit the geometry and averaging parameters exactly as you would in the SAR AVERAGING SENSOR editor. Once you click DONE in the editor, you will be given the option to automatically run the post-processor. If you say yes, the post-processing operation is queued as if you had created a new simulation. Otherwise, you will have to do the post-processing run manually. Once the SAR averaging is complete, new SAR average data will appear in the Results workspace window.

- Section 10.6 describes the parameters in the SAR AVERAGING SENSOR editor.
- See Section 11.2.3 for instructions on manually running remote SAR post-processing.

12.5.3 SAR Combination

When two or more SAR results of the same variety (e.g. two 1-gram SAR Averages) are selected in the RESULTS workspace window, a COMBINE SAR RESULTS option will become available in the context menu. This command allows multiple scaled SAR results to be summed to find a maximum combined SAR value. A COMBINE SAR RESULTS dialog will display a table of the selected SAR results. For each result, the FREQUENCY of the result should first be selected. If desired, the scaling factor for each result can be specified by editing the NET INPUT POWER value displayed for that run. Press the **OK** button to begin the computation.

A workspace window displays the result of the computation, giving the maximum combined SAR value and a list of one or more Yee cell locations where that maximum was found. The grid indices of the Yee cell are also provided. If the calculation grids for the input runs were not identical (as noted at the bottom of the result window), grid indices are provided for each input run. In that case, SAR for each of the input distributions is considered constant per Yee cell for that run. When Yee cells for different runs are partially overlapping, the combining process tests the SAR value at all possible overlapping combinations of Yee cell voxels.

- Unlike SAR averaging and far zone post-processing, SAR combination results are not saved to disk, and therefore must be recomputed each time they are requested.
- Once computed, a combined SAR result does not respond to scaling changes to its constituent parts.

Chapter 13

Additional Tools for Customizing and Organizing Projects

In this chapter, you will learn how to...

→ use scripts to streamline virtually any task in XFdtd

→ define your own universal parameters

→ store object definitions externally to reuse them in multiple projects

→ organize objects in your XFdtd project for more convenient access

XFdtd provides several convenient tools to facilitate the creation and organization of projects.

Parameterization is a powerful tool that makes it easy to define variables and functions in one convenient workspace window, which can be referenced anywhere within the XFdtd interface. Additionally, parameterization can be used in conjunction with scripting to sweep through a series of parameters (i.e. multiple antenna lengths) to run a calculation at every swept point. In XFdtd, parameters are defined in the PARAMETERS workspace window.

The SCRIPTING workspace window makes it convenient to write scripts to accomplish tasks that are specific to an XFdtd project. It provides users with the ability to create fully-customizable functionality within the XFdtd interface that is specific to their own tasks. Scripting may be useful for quickly performing repetitive tasks, referencing external files, employing a series of modeling operations at once, or virtually any task that is tedious with the standard XFdtd tools.

Libraries provide useful means of storing definitions and any types of objects created within an XFdtd project. They are saved in the LIBRARIES workspace window as files that are not attached to a specific project so that they can be referenced again and again. They are very useful for creating new projects that reuse definitions and objects from past projects. Rather than having to rebuild a project from scratch, pertinent libraries can be imported so that time is not wasted redefining similar objects and properties.

Several other features in XFdtd are available that are also very useful for grouping and organizing definitions and objects. The GROUPS branch of the PROJECT TREE functions to store **shortcut groups**, which are groups of objects that are added and organized by the user. Similarly, ASSEMBLIES are user-defined groups of geometric objects that are added to the PARTS branch of the Project Tree. They are convenient, especially for projects that contain a large number of parts, so that objects in the tree remain organized and easy to access.

13.1 Parameters Workspace Window

The PARAMETERS workspace window allows users to define parameters from a single value or formula, which can be used in most edit fields. Parameterizing provides the ability to change many objects at once by only varying a single value. This window is accessible by right-clicking in the tabbed workspace or selecting the window in the VIEW menu of the Application Menu Bar.

Figure 13.1 shows the Parameters workspace window.

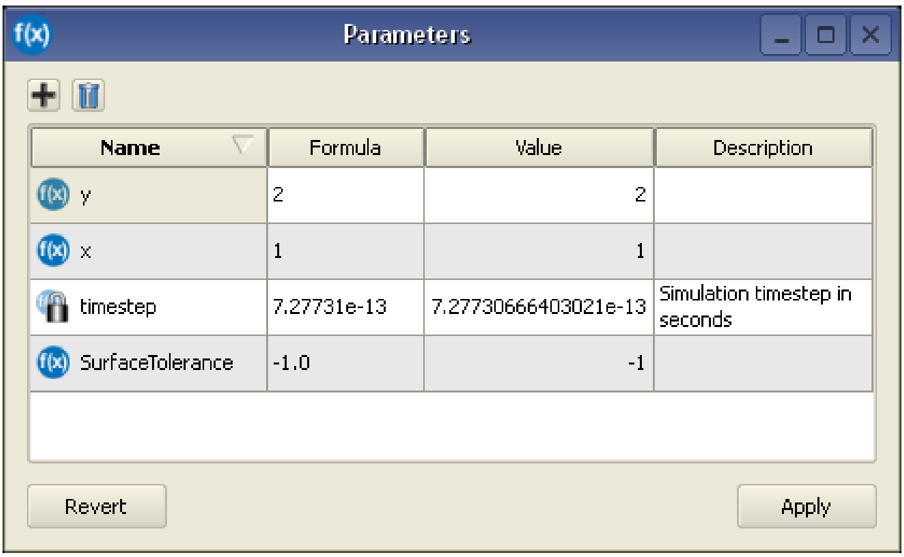


Figure 13.1: The Parameters workspace window

13.1.1 Defining Parameters

The PARAMETERS workspace window contains four fields: NAME, FORMULA, VALUE, and DESCRIPTION. Note that a value named 'timestep' is already present upon opening this window.

A new parameter is added by clicking the + button above the table of parameters. A new line with default values will be added to the list of parameters. These values can edited by double-clicking on any value. The TAB key will scroll through the columns of the table and the ESCAPE key will cancel any changes that have been made to a parameter entry.

The FORMULA contains a mathematical expression or simple number which XFdtd will evaluate. The value for each parameter is a unit-less quantity allowing it to be used in any field of the user interface which accepts parameters. When parameter formulas are given a unit, the value is converted to SI units (meters, seconds, amperes, etc.) ensuring any mathematical operations performed using the parameters maintains a consistent unit, independent of display preference.

Formulas can reference other parameters that have already been defined.

The VALUE column is a read-only column that displays the evaluated value of the parameter. If an invalid formula is entered, an error message will appear within this field with a description of the invalidity. Simply hold the mouse over the error message to view this description.

Similarly, a parameter is deleted by selecting the unwanted parameter and clicking the button above the table. If a parameter is deleted that is referenced within another parameter's definition, an error message will appear since the parameter that is referenced is no longer defined.

Each parameter is referenced by its assigned name defined in the NAME column of this window. For instance, a parameter named '*length*' can be referenced by typing '*length*' into any dialog box within XFdtd, and it will assume this defined value.

▶ A parameter sweep may be set up in the SIMULATIONS workspace window. See Section 11.1.

13.1.2 Parameters Syntax

There are several important syntax rules to keep in mind when defining parameters or formulas in XFdtd.

Parameter Names...

- Cannot start with a number.
- Can only contain alpha-numeric characters and underscores (no spaces).

Parameter Formulas...

- Must have separation between the value and the unit (i.e., “2 mm”, not “2mm”).
- Will bind units with the value immediately to their left (i.e., “2 + 3 mm” = 2.003 mm; “(2+3) mm” = 0.005 mm).
- Cannot be used to modify any scripting objects.
- Should be completely independent of everything but other parameters (i.e., its value should not depend on random numbers or variables).
- Can be built from anything that is legal in the scripting language, QtScript.
  - ▶ For more information on QtScript, visit the Qt Software homepage at www.qtsoftware.com.

13.2 Scripting Workspace Window

The SCRIPTING workspace window allows users to create, edit, manage, and execute user-defined scripts, which are capable of gathering and reporting information from the XFdtd project or making changes to the project. Scripts are blocks of QtScript and are typically used to automate repetitive or tedious tasks (that could otherwise be done through the XFdtd GUI) with speed and precision.

There are two types of scripts in XFdtd:

- MACRO scripts, which typically contain global-scoped code that is only executed on demand.
- FUNCTION scripts, stored in FUNCTION LIBRARIES, which contain function and class definitions that are always available within the scripting engine.

Parameters that have been defined in the active parameter list of the PARAMETERS workspace window may be referenced within any script. A consequence of this ability is that changing a parameter in the parameter list may change the behavior of user-defined functions.

One very powerful capability of XFdtd is the ability to script systematic parameter sweeps. The following is a simple example of a script that performs a single parameter sweep, varying the parameter ”length” from zero to five inches in increments of one:

```
var simData = App.getActiveProject().getNewSimulationData();
var sweep = simData.getParameterSweep();
// Sweep over the parameter 'length'
sweep.parameterName = "length";
// Sweep 'length' from zero to six in increments of one
sweep.setParameterValuesByIncrement( 0, 1, 6 );
// Turn on the sweep
simData.enableParameterSweep = true;
```

▸ For more information on setting up parameter sweeps, refer to Section 11.1.1.

Figure 13.2 shows the SCRIPTING workspace window.

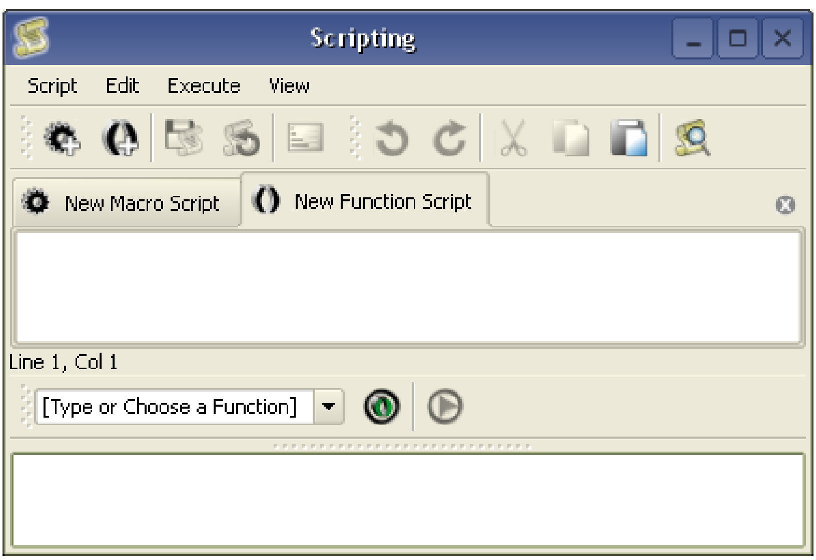


Figure 13.2: The Scripting workspace window

13.2.1 The XFdtd Scripting Language

XFdtd scripts are written in the language QtScript, a variant of ECMAScript.

▸ For more information on QtScript, visit the Qt Software homepage at www.qtsoftware.com.

New Macro Script

Select this icon to bring up a NEW MACRO SCRIPT tab where a new MACRO can be defined. The macro is not executed until the EXECUTE MACRO icon is pressed.

New Function Script

Select this icon to bring up a NEW FUNCTION SCRIPT tab where a new FUNCTION can be defined. The function is not executed until the CALL FUNCTION icon is pressed.

Commit Script

Select this icon to commit a change to a script after an edit. Any changes that have been made to the script will become active. For instance, if a new function was added to the script, it will now appear in the drop-down list of functions when the CALL FUNCTION icon is pressed.

Revert

Select this icon to ”revert” or abandon an edit. Any changes that have been made in the editor will be lost, and the editor will revert back to the script that was stored in the XFdtd project or application.

Clear Output Window

Select this icon to clear text in the output window where error messages and script output are written.

Search and Replace

Select this icon to search for text within scripts. By selecting the REPLACE WITH check-box, text can be located and replaced with the desired text.

Call Function

Select this icon to execute a function from one of the user-defined scripts. Navigate to the tabbed function that is to be called, or select the function from the drop-down menu to the left of this icon. The chosen function will execute and generate any output or error messages in the output window below. If the function takes longer than one half second to finish, a progress dialog box will pop up so that the operation may be cancelled if desired. A message in the output window will indicate when the function call has completed.

Execute Script

Select this icon to execute a macro from one of the user-defined scripts. Macros are executed in the same manner that is described under **Call Function**.

13.3 Libraries Workspace Window

The LIBRARIES workspace window allows users to create libraries or collections of objects grouped by category so that they can be easily referenced during a project and can be accessed in subsequent projects. This makes it very easy to access commonly used objects and definitions so that they do not have to be recreated during every project.

Figure 13.3 shows the LIBRARIES workspace window.

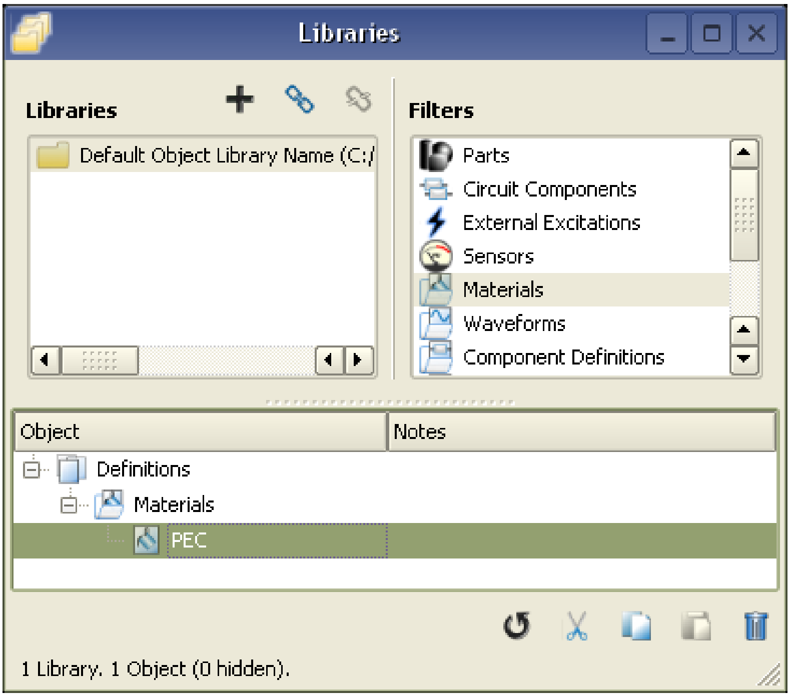


Figure 13.3: The Libraries workspace window

13.3.1 Creating a New Library

To create a new library directory or subdirectory, click on the + button above the LIBRARIES space in the workspace window. Specify the name of the location to store the new library file.

13.3.2 Accessing Existing Libraries

To access an existing library, click on the button, navigate to the appropriate directory, and select the desired library to load into the project.

13.3.3 Adding Objects and/or Definitions to a Library

To add objects of definitions to a library, simply drag the desired object from the PROJECT TREE into lower workspace entitled OBJECT/NOTES. The object will be placed in a library grouping that corresponds to its original position in the Project Tree. For example, a MATERIAL dragged from the Project Tree will be placed in the MATERIALS folder of the Library. Additionally, filters can be applied to library objects to control the visibility of each group. They are controlled in the FILTERS section of the LIBRARIES workspace window.

Appendices

Appendix A

Appendix of Geometric Modeling

In this appendix, you will learn...

→ how to use all of the geometric modeling tools available within XFdtd

A.1 Basic Geometric Shapes

Cone

The CONE creation tool allows the user to add a 2D (sheet body) or 3D (solid body) cone to the project. The dimensions of the object can be specified by keying in numeric or parametric expressions in the HEIGHT and RADIUS fields. The user can interactively edit the cone by dragging the vectors in the geometry view. The CREATE AS option toggles between creating sheet or solid geometry.

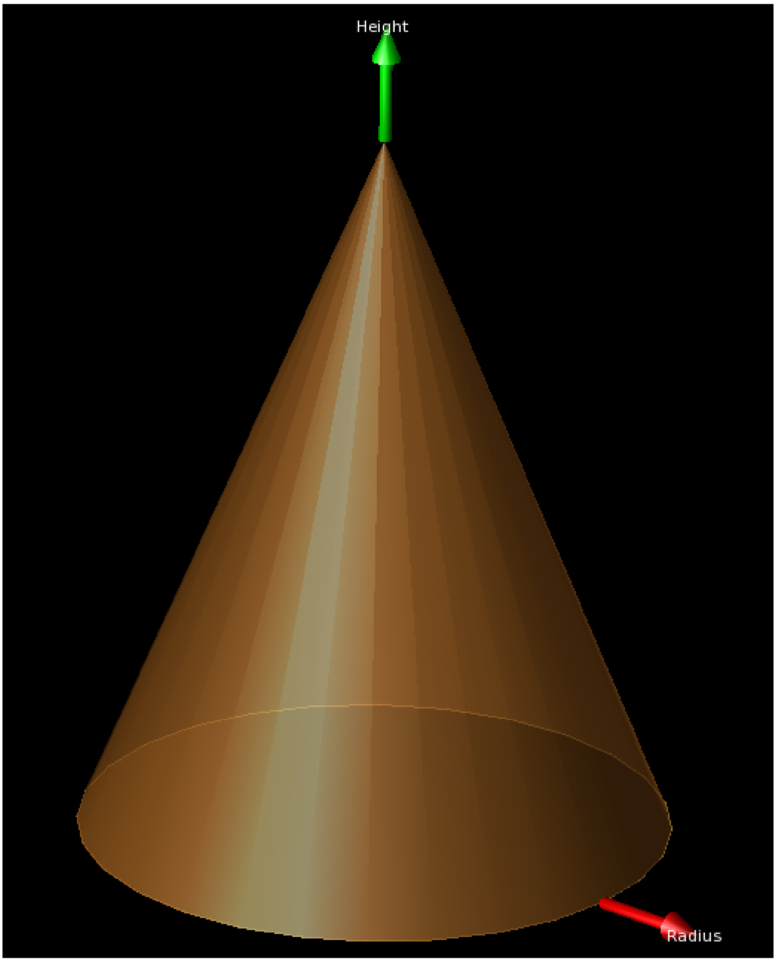


Figure A.1: Interactive vectors for editing a cone

Cuboid

The CUBOID creation tool allows the user to add a 2D (sheet body) or 3D (solid body) cuboid to the project. The dimensions of the object can be specified by keying in numeric or parametric expressions in the WIDTH, DEPTH and HEIGHT fields. The user can interactively edit the object by dragging the vectors in the geometry view. The CREATE AS option toggles between creating sheet or solid geometry.

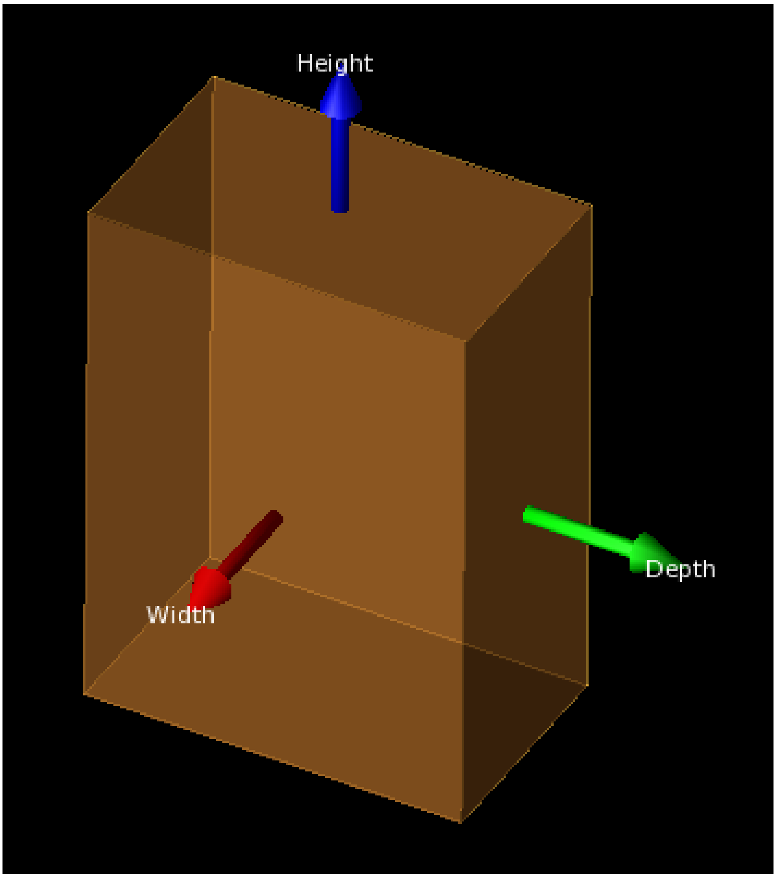


Figure A.2: Interactive vectors for editing a cuboid

Cylinder

The CYLINDER creation tool allows the user to add a 2D (sheet body) or 3D (solid body) circular cylinder to the project. The dimensions of the object can be specified by keying in numeric or parametric expressions in the HEIGHT and RADIUS fields. The user can interactively edit the object by dragging the vectors in the geometry view. The CREATE AS option toggles between creating sheet or solid geometry.

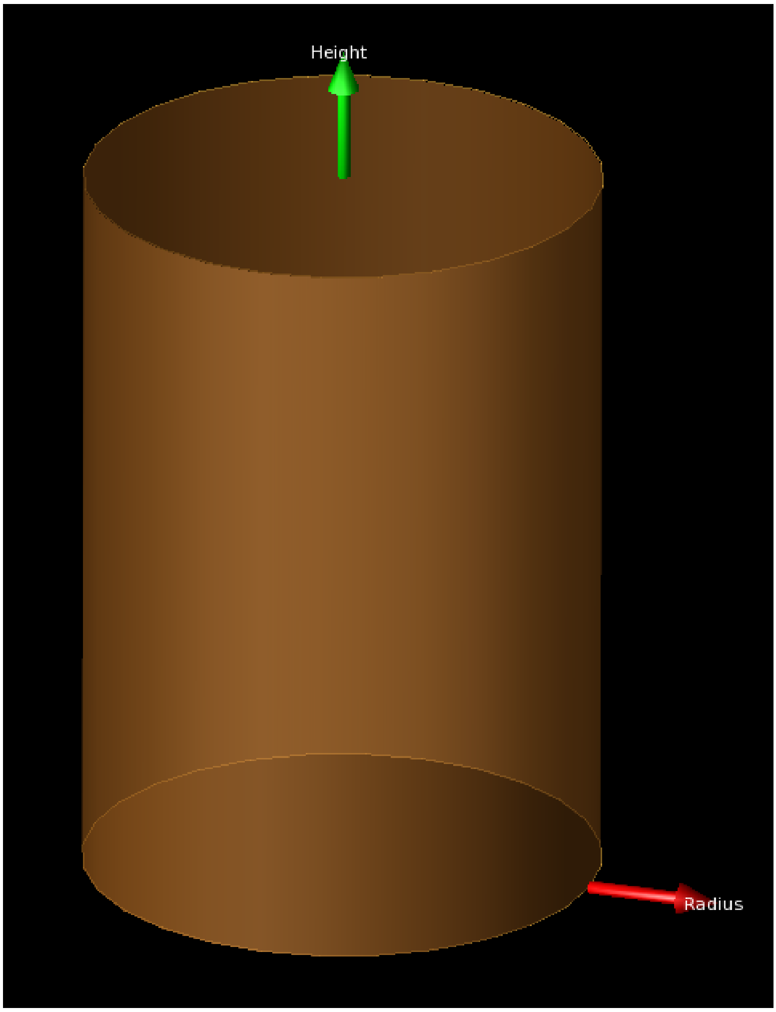


Figure A.3: Interactive vectors for editing a cylinder

Frustum

The FRUSTUM creation tool allows the user to add a 2D (sheet body) or 3D (solid body) elliptical frustum to the project. The dimensions of the object can be specified by keying in numeric or parametric expressions in the HEIGHT, RADIUS 1, RADIUS 2 and TOP fields. The user can interactively edit the object by dragging the vectors in the geometry view. The CREATE AS option toggles between creating sheet or solid geometry.

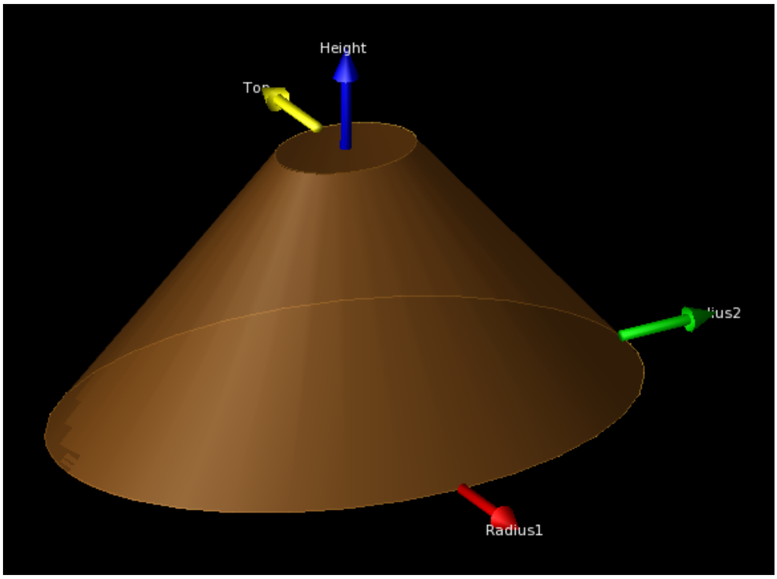


Figure A.4: Interactive vectors for editing a frustum

Helix

The HELIX creation tool allows the user to add a 2D (sheet body) or 3D (solid body) helix to the project. The dimensions of the object can be specified by keying in numeric or parametric expressions in the HEIGHT, MAJOR RADIUS, MINOR RADIUS and THREAD SPACING fields. Additionally, the WINDING option toggles between left and right handedness. The user can interactively edit the object by dragging the vectors in the geometry view. The CREATE AS option toggles between creating sheet or solid geometry.

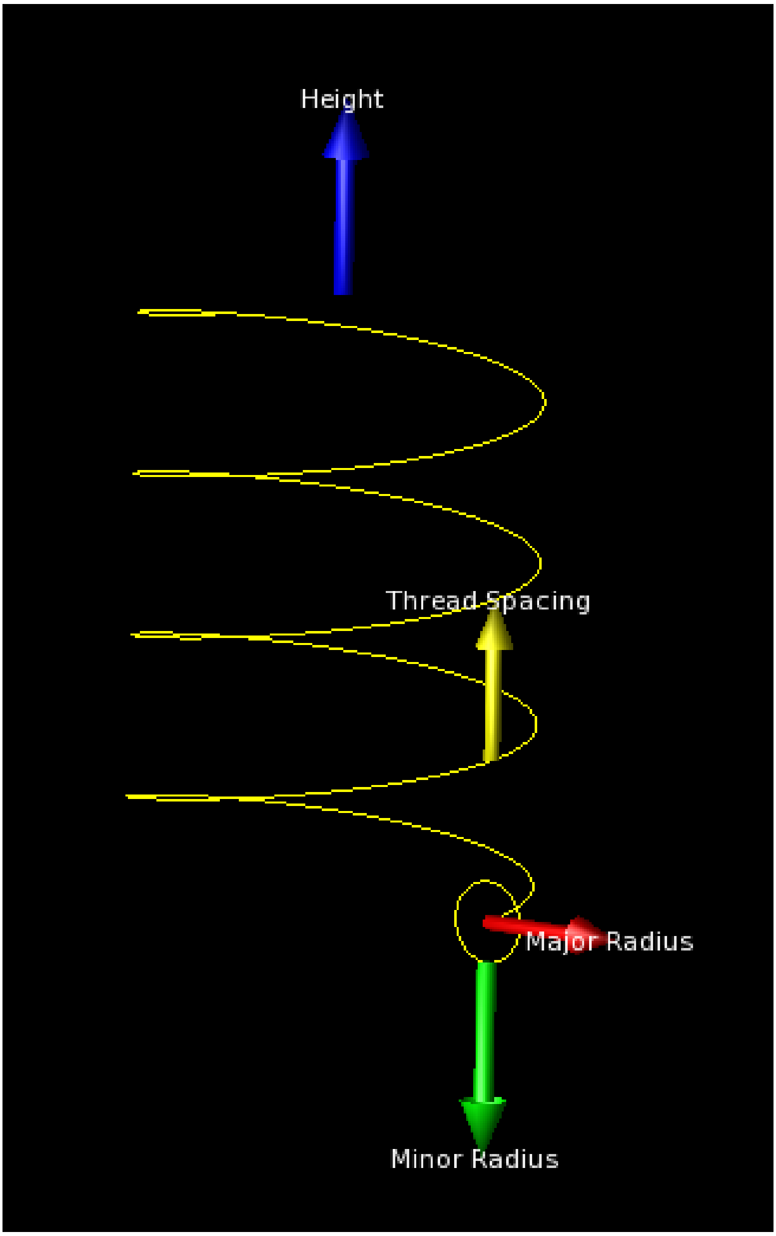


Figure A.5: Interactive vectors for editing a helix

Prism

The PRISM creation tool allows the user to add a 2D (sheet body) or 3D (solid body) prism to the project. The dimensions of the object can be specified by keying in numeric or parametric expressions in the HEIGHT, RADIUS 1 and RADIUS 2 fields. Control the number of sides using the SIDES field. The user can interactively edit the object by dragging the vectors in the geometry view. The CREATE AS option toggles between creating sheet or solid geometry.

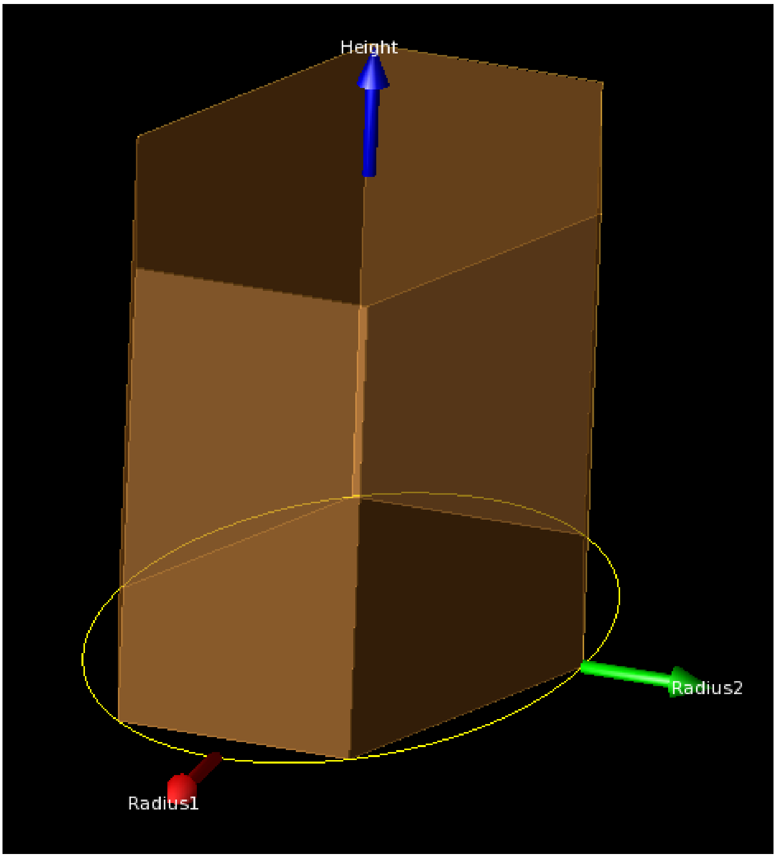


Figure A.6: Interactive vectors for editing a prism

Pyramid

The PYRAMID creation tool allows the user to add a 2D (sheet body) or 3D (solid body) pyramid to the project. The dimensions of the object can be specified by keying in numeric or parametric expressions in the HEIGHT, RADIUS 1, RADIUS 2 and TOP fields. Control the number of sides using the SIDES field. The user can interactively edit the object by dragging the vectors in the geometry view. The CREATE AS option toggles between creating sheet or solid geometry.

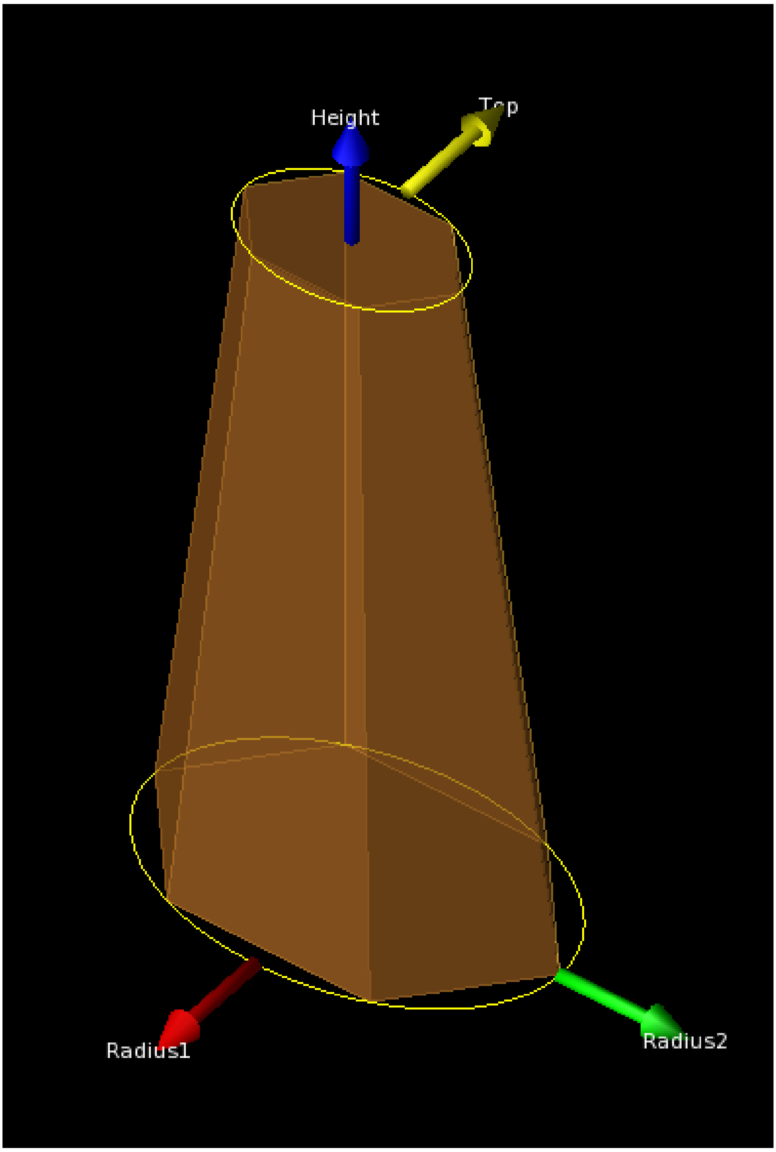


Figure A.7: Interactive vectors for editing a pyramid

Sphere

The SPHERE creation tool allows the user to add a 2D (sheet body) or 3D (solid body) sphere to the project. The dimensions of the object can be specified by keying in a numeric or parametric expression in the RADIUS field. The user can interactively edit the object by dragging the vector in the geometry view. The CREATE AS option toggles between creating sheet or solid geometry.

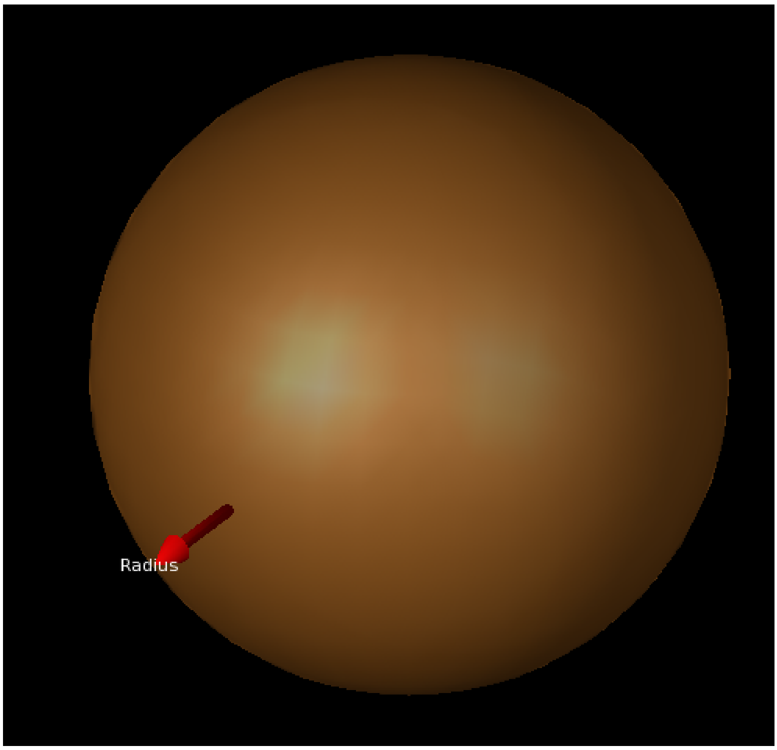


Figure A.8: Interactive vectors for editing a sphere

Torus

The TORUS creation tool allows the user to add a 2D (sheet body) or 3D (solid body) torus to the project. The dimensions of the object can be specified by keying in numeric or parametric expressions in the MAJOR RADIUS and MINOR RADIUS fields. The user can interactively edit the object by dragging the vectors in the geometry view. The CREATE AS option toggles between creating sheet or solid geometry.

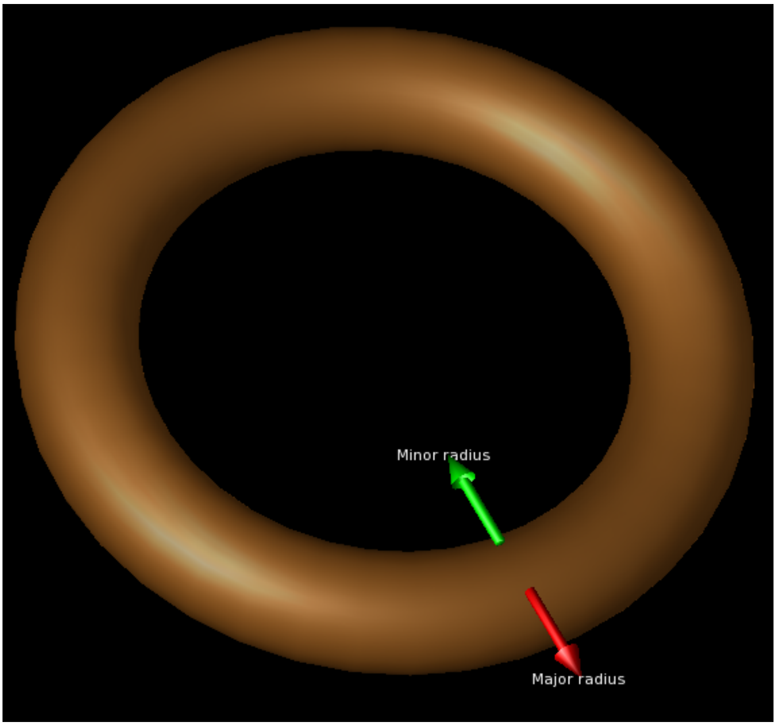


Figure A.9: Interactive vectors for editing a torus

A.2 Editing Cross-Sections for 2-D and 3-D Models

A.2.1 Shapes

Edge tools

Edge tools are used to create lines of various shapes within the XFdtd interface. Figure A.10 displays the Edge Tools including the STRAIGHT EDGE tool (upper left), POLYLINE EDGE tool (upper right), TANGENT LINE tool (lower left) and PERPENDICULAR EDGE tool (lower right).

✓ Pressing TAB while using these tools will bring up the SPECIFY POSITION dialog, which is used to enter relevant properties to the tool being used.

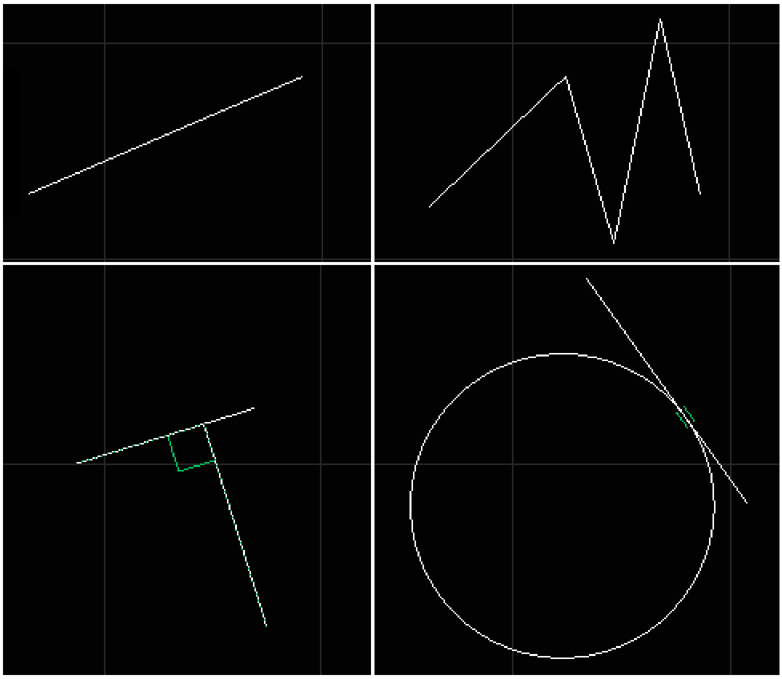


Figure A.10: The Edge Tools

Straight Edge Creates a simple straight edge. To use this tool, click the STRAIGHT EDGE button and click two points in the sketching plane where the endpoints should be located.

Polyline Edge The POLYLINE EDGE is similar to the Straight Edge tool except it allows multiple points to create a series of connected straight edges. Click a starting point in the sketching plane and continue clicking on the locations of subsequent endpoints to create desired polyline edge. Click on the first vertex or press RETURN to finish.

Perpendicular Edge Creates a straight edge perpendicular to an existing edge. To use, select the PERPENDICULAR EDGE button and click on the existing edge that will define the perpendicular direction. This can be a straight or curved edge. Then click on the location of the first and second endpoints of the perpendicular straight edge.

Tangent Line Similar to the Perpendicular Edge tool, but instead draws a line *tangent* to a pre-existing, non-linear edge. To use, select the TANGENT LINE tool, and click on the existing curve that will define the tangential direction. Then click on the location of the first and second endpoints of the tangential straight edge.

Closed Polygon tools

Figure A.11 displays the Closed Polygon tools including the RECTANGLE, POLYGON and N-SIDED POLYGON tools.

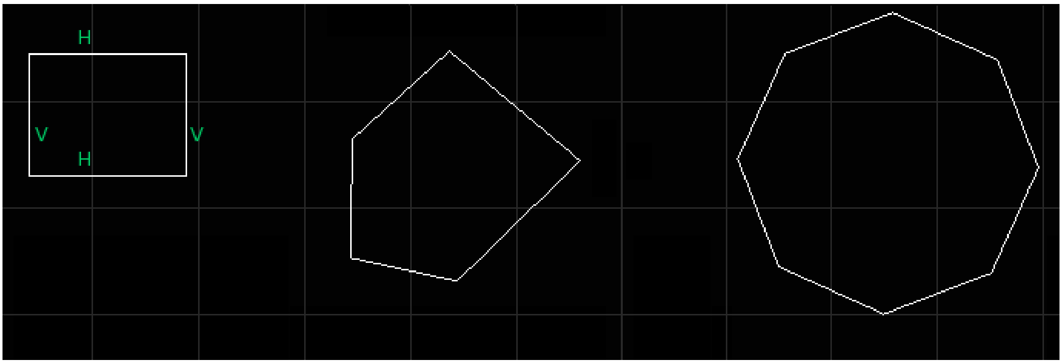


Figure A.11: The Closed Polygon tools

Rectangle Creates a simple rectangle. Click the desired location of the first vertex of the rectangle and drag the mouse to the location of the second vertex.

Polygon Creates a polygon specified by the user. (For regular polygons, see N-SIDED POLYGON). It functions like the POLYLINE EDGE tool. Click the starting point and all subsequent points, then click RETURN to close the polygon. This will draw a line from the last selected endpoint to the first endpoint.

N-Sided Polygon Creates a regular, N-Sided Polygon of a user-specified number of sides. Click the location of the center of the polygon. Then press the left-bracket key '[', or the right-bracket key ']' to decrease or increase the number of sides, respectively. Once the correct number of sides is selected, drag the mouse until the desired size and orientation around the center point is achieved and click again to finish the N-sided polygon.

Arc tools

Figure A.12 displays two of the arc tools: the 3-POINT ARC and 2-POINT ARC tools.

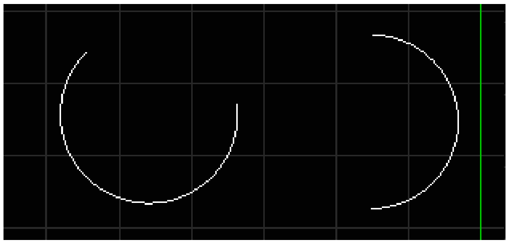


Figure A.12: The Arc tools

3-Point Arc tool Creates an open arc from three points. Click on the location of the first endpoint. Click a second location to specify a point between the two endpoints (which helps determine size), and a third location to specify the other endpoint.

2-point Arc tool Creates a semi-circle from two points. Click on the first endpoint location and drag the mouse until the desired semi-circle size and orientation is achieved. Click this second end point location to finish.

Arc center, 2 points tool Creates an open arc from three points. First, click on the location of the center of the arc. Secondly, click a point to specify the radius of the arc. Finally, click the location of the endpoint to specify the length of the arc.

Circle and Ellipse tools

Figure A.13 displays an example of the CIRCLE CENTER, RADIUS tool and the ELLIPSE tool.

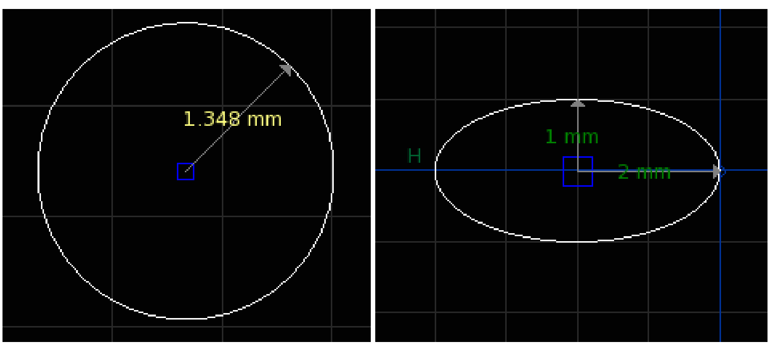


Figure A.13: The Circle Center, Radius and Ellipse tools

Circle Center, Radius Creates a circle defined by its center point and radius. Click the location of the circle's center point, then select another point to define the radius and finish the circle.

3-point Circle Creates a circle based on three user-specified points, similar to the 3-POINT ARC tool. Click the first two points to set the location of the circle and the third to specify its size.

2-point Circle Creates a circle based on the distance between two points. After selecting the first point, choose the second to define the diameter and finish the circle.

Ellipse Draws an ellipse from three points: the center and two perpendicular radii. Click the center point of the ellipse, then select the desired location of the first radii. Finally, select the desired length of the second radii, perpendicular to the first.

A.2.2 Tools

Select/Manipulate

Enabled by default and used as the universal selection tool in geometry creation. It can be used to:

- move an object, edge, or vertex to a new position, by clicking-and-dragging
- select a vertex or edge and lock or edit its position, by right-clicking and selecting LOCK POSTION or EDIT POSITION
- edit the value of an angle or distance constraint, by right-clicking and selecting the edit option
- delete an edge or constraint, by right-clicking and selecting the delete option

Note that this tool is different from the SELECT tool in the VIEW TOOLS menu.

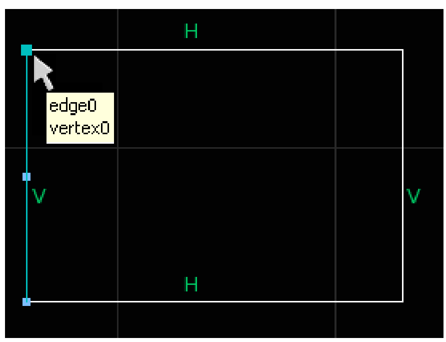


Figure A.14: Select/Manipulate tool

Trim Edges

Deletes segments of edges until they intersect with other edges. To use this tool, click on the section of the edge that is to be deleted.

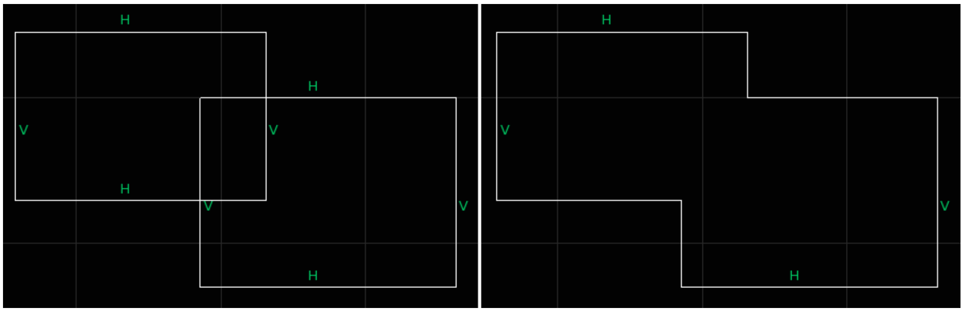


Figure A.15: The Trim Edges tool, used to combine two shapes into one

Insert Vertex

Inserts a vertex onto an already existing edge. Click the desired location of the new vertex on the existing edge.

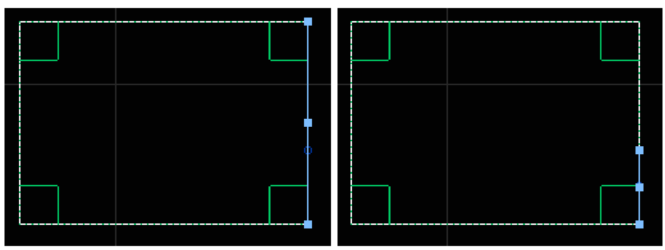


Figure A.16: Insert Vertex tool

Fillet Vertex

Converts a sharp corner into a rounded corner between two curves. Click on any sharp corner and drag until the desired fillet radius is achieved and click to finalize fillet.

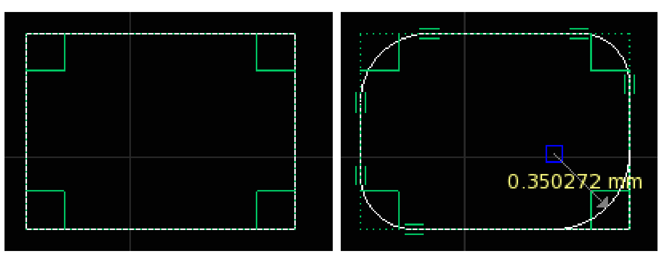


Figure A.17: Fillet Vertex tool

Copy Cross Section Edges

Copies the edges formed by the intersection of the cutting plane and any solid geometry it intersects. Click on a single edge to copy it into the current sketch, or click on the part containing the intersection to copy all of the formed edges.

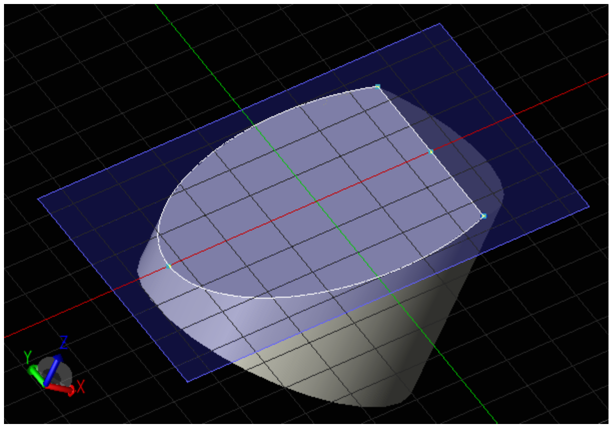


Figure A.18: Copy Cross Section Edges tool

A.2.3 Constraints

The geometry Constraints tools are used to modify pre-drawn shapes to the desired specifications.

Some of the "before" images below have been marked with white arrows to show which edges are constrained in the "after" image on the right.

Horizontal Constraint

Constrains a segment to the horizontal direction.

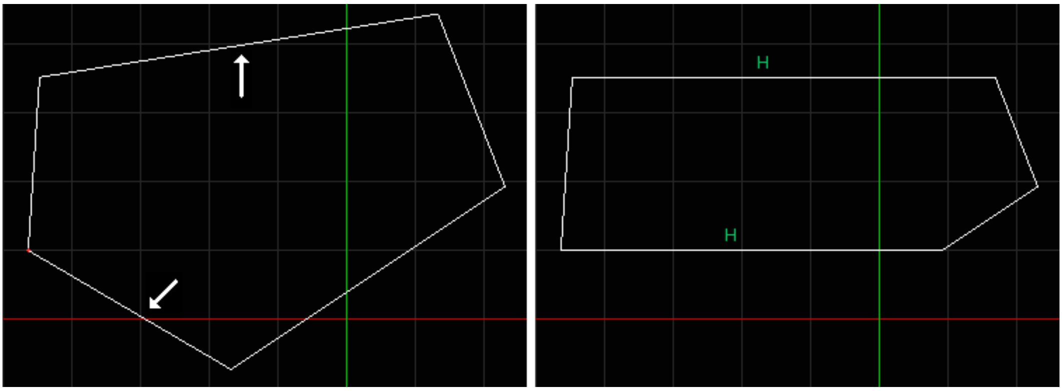


Figure A.19: Polygon before (left) and after (right) two sides are constrained horizontally

Vertical Constraint

Constrains a segment to the vertical direction.

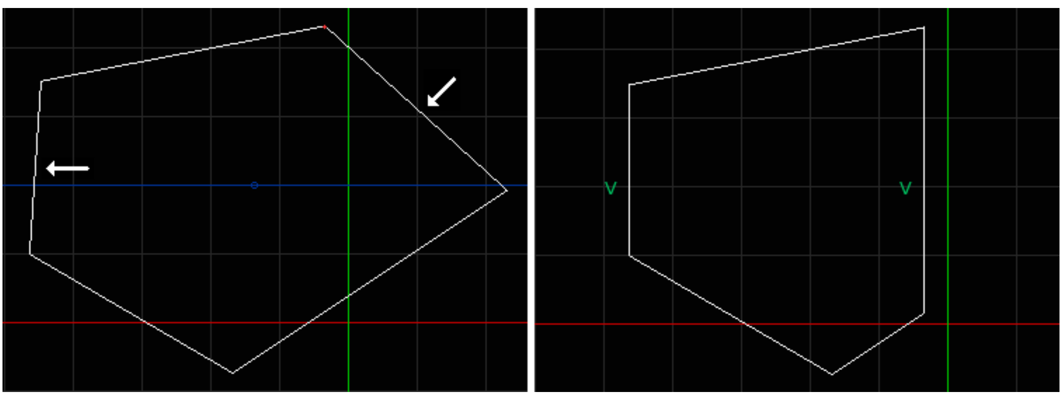


Figure A.20: Polygon before (left) and after (right) two sides are vertically constrained

Collinear Constraint

Constrains two straight segments so that they are in line with each other.

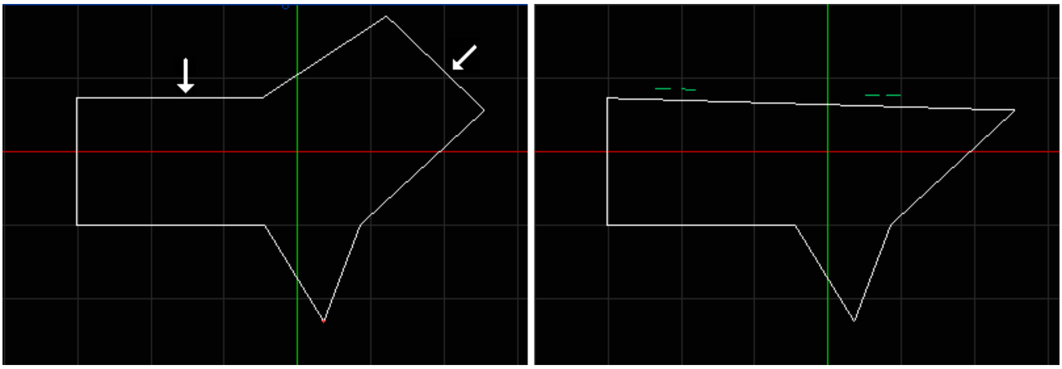


Figure A.21: Polygon before (left) and after (right) after two sides are constrained to be collinear

 **Parallel Constraint**

Constrains two straight segments so that they are parallel to each other.

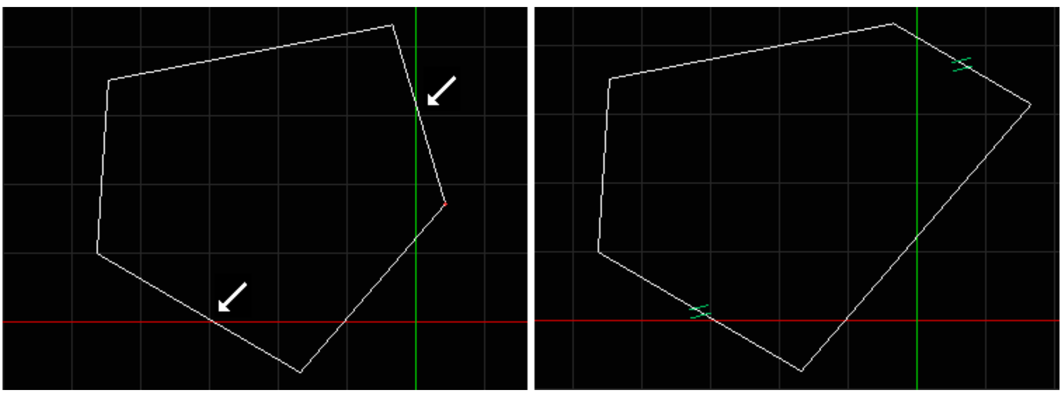


Figure A.22: Polygon before (left) and after (right) two sides are constrained in parallel

 **Perpendicular Constraint**

Constrains two straight segments so that they are perpendicular to each other.

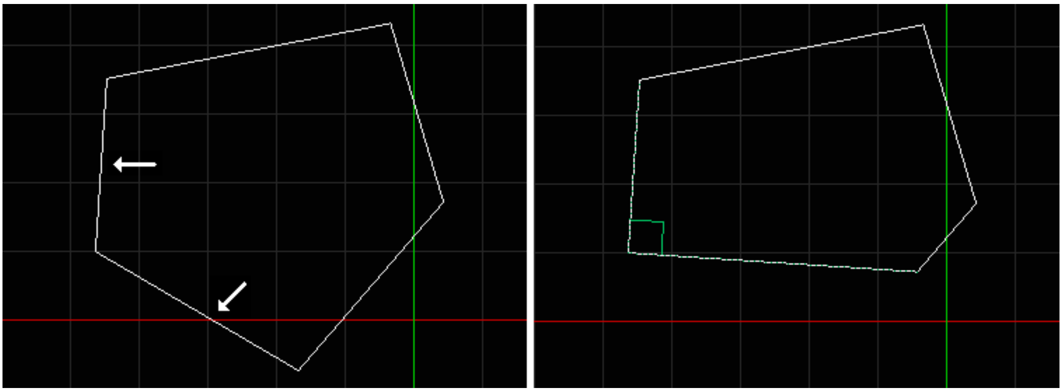


Figure A.23: Polygon before (left) and after (right) two sides are constrained perpendicularly

Tangent Constraint

Constrains a straight segment so that it is tangent to a circular segment at a point.

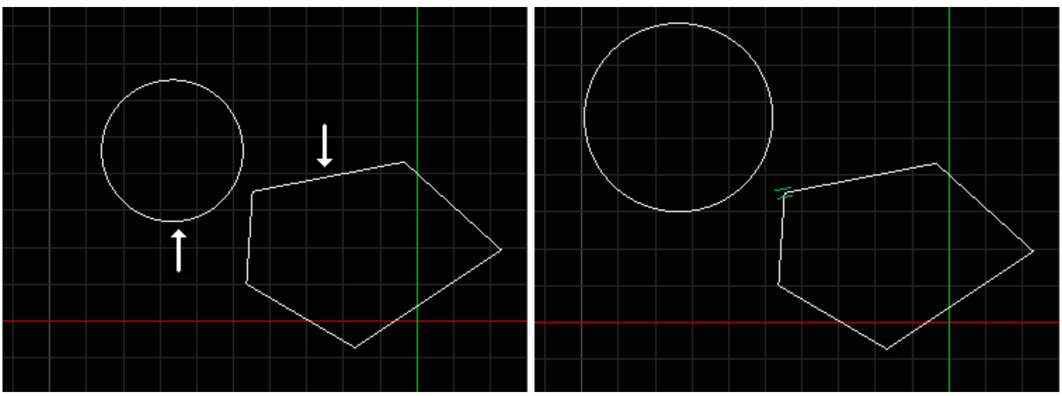


Figure A.24: Circle and polygon before (left) and after (right) a side of the polygon is constrained tangentially in reference to the circle

Concentric Constraint

Constrains two circular segments so that they are centered upon the same point.

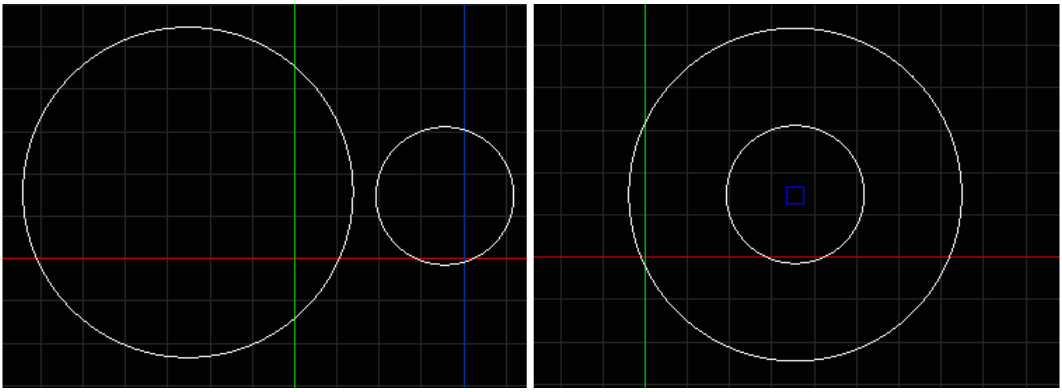


Figure A.25: Two circles before (left) and after (right) they are made concentric

Angle Constraint

Constrains an angle to a user-specified value between two straight lines. Click once to select angle, then click a second time to place label and enter the angle size.

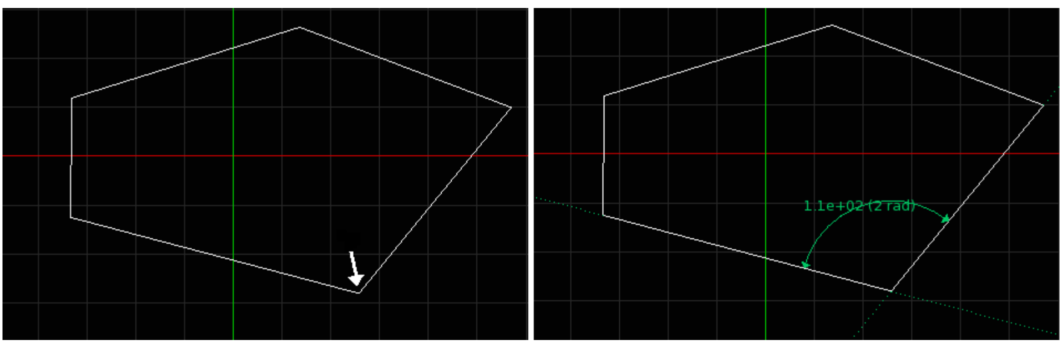


Figure A.26: Polygon before (left) and after (right) an angle has been constrained to a user-defined value

Distance Constraint

Constrains the distance between two points, the distance between a point and a line, or the length of a line to a user-specified value. After selecting the object(s) to constrain, click a final time to place label and enter distance.

As shown in Figure A.27, there are three different constraint "modes": parallel, vertical and horizontal. The mode is determined by the location of the mouse cursor when you click to specify where the constraint should be drawn.

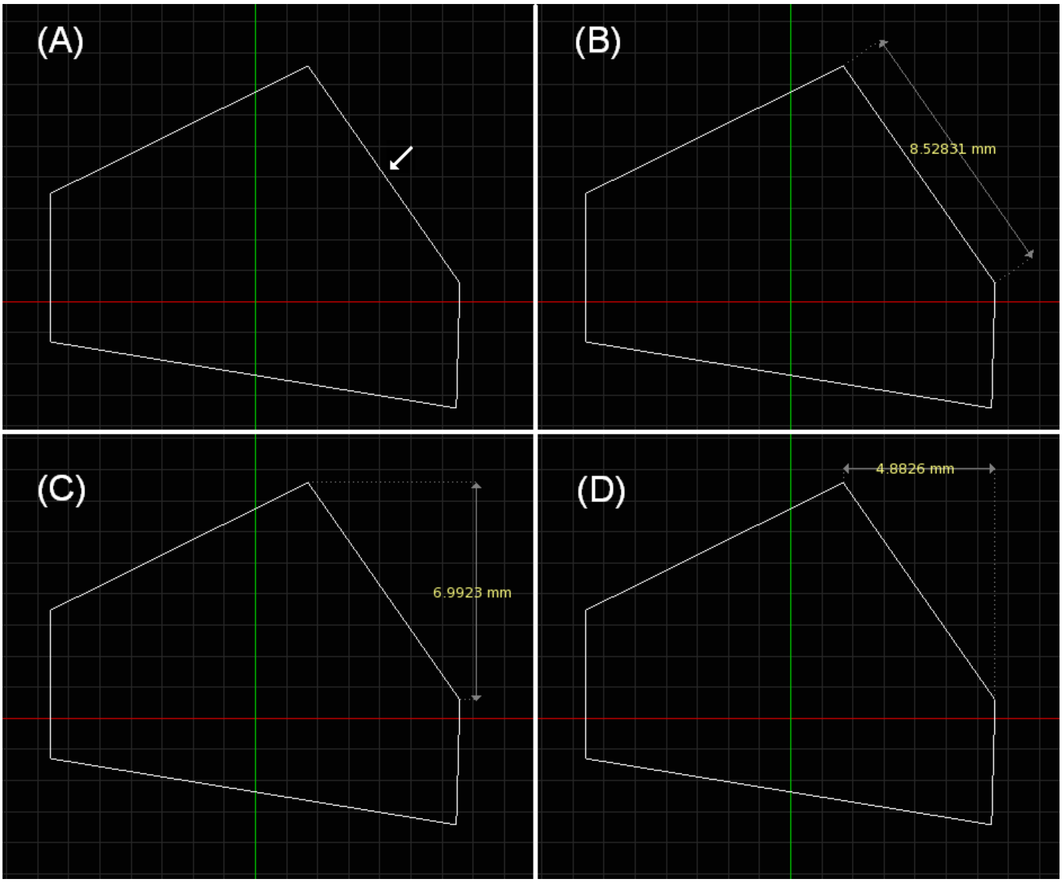


Figure A.27: Polygon (A) before line has been constrained, (B) with a parallel distance constraint, (C) with a vertical distance constraint and (D) with a horizontal distance constraint.

Equal Length Constraint

Constrains selected segments to an equal length (assumes the length of the segment selected second).

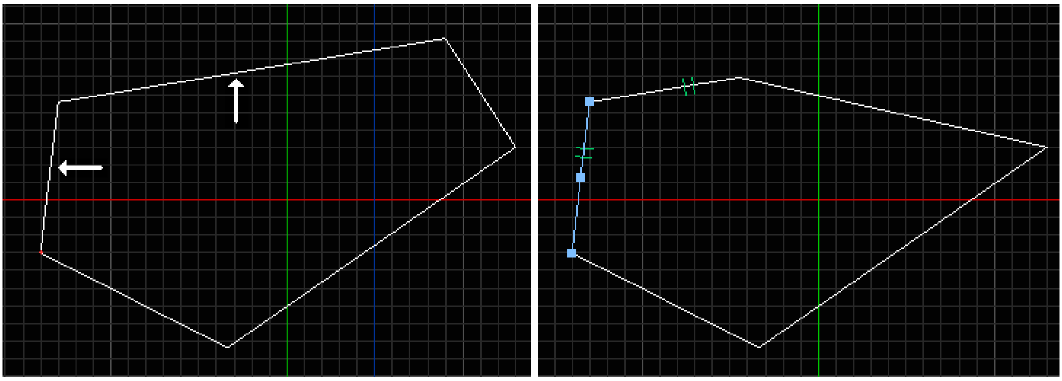


Figure A.28: Polygon before (left) and after (right) two sides are made equal length to one another

Equal Distance Constraint

Constrains two pairs of points so that each pair assumes a distance from each other equal to the distance between the original pair.

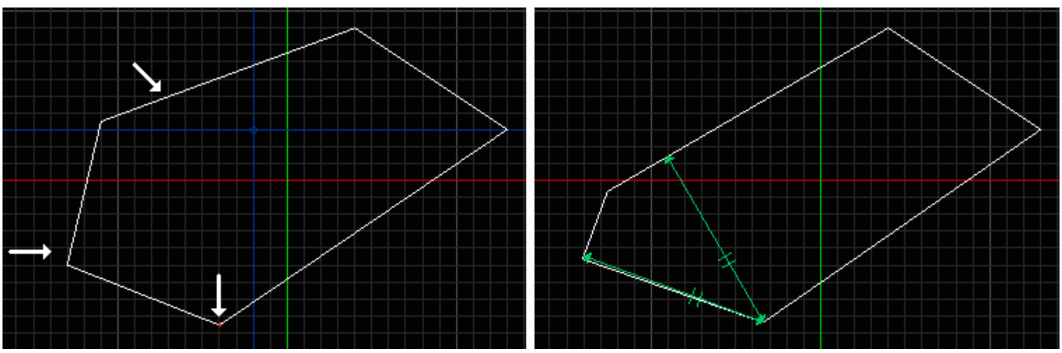


Figure A.29: Polygon before (left) and after (right) two sides are made equal distance from each other

Radius Constraint

Constrains the radius to a user-specified value.

Equal Radius Constraint

Constrains selected radii to an equal length.

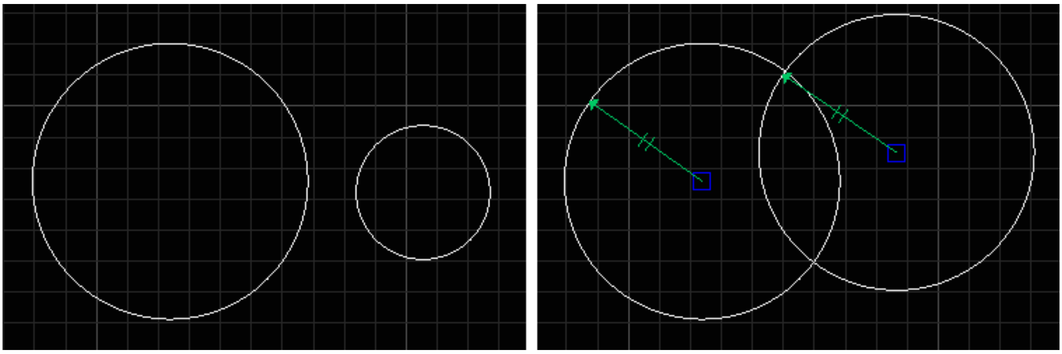


Figure A.30: Two Circles before (left) and after (right) their radii are made equal

A.2.4 Snapping

Snapping tools are used to snap the mouse to a specific point or edge in the XFdtd geometry.

The blue lines in the images below highlight the "snap-to" landmarks.

Snap to Grid Line

Mouse is snapped to the nearest point on the nearest grid line.

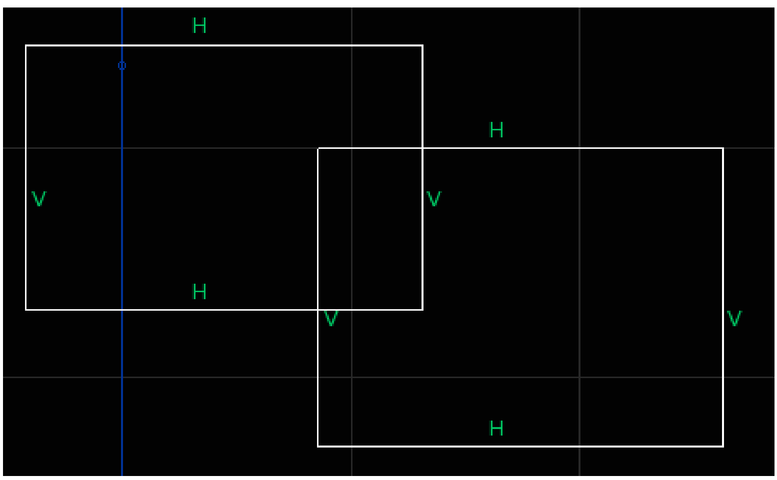


Figure A.31: Snap to Grid Line Tool

Snap to Grid/Edge Intersections

Mouse is snapped to the nearest intersection between the grid and the sketch edge.

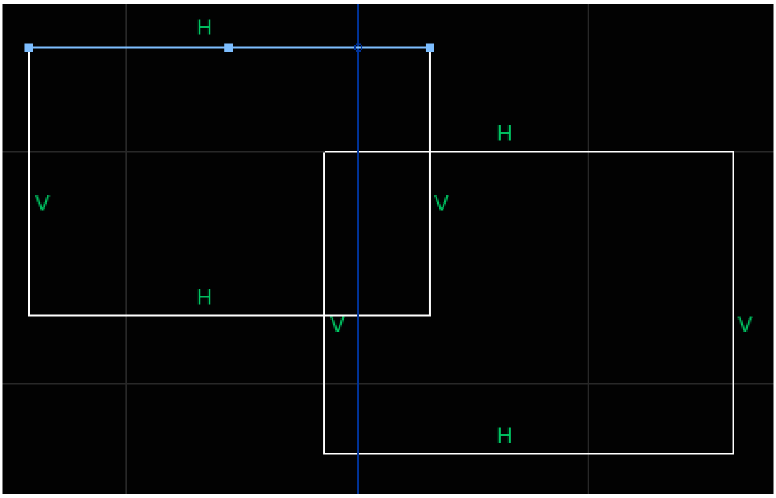


Figure A.32: Snap to Grid/Edge Intersections Tool

Snap to Vertices

Mouse is snapped to the nearest vertex of the sketch or edge mid-point within range.

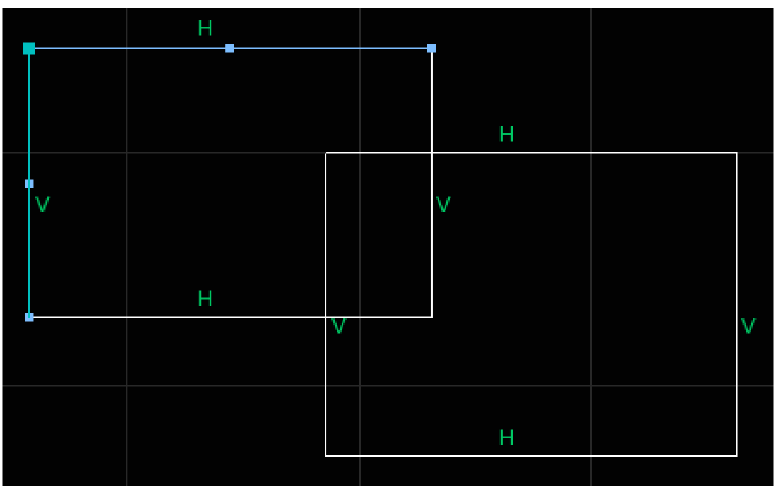


Figure A.33: Snap to Vertices Tool

Snap to Edges

Mouse is snapped to the edges of a pre-defined object.

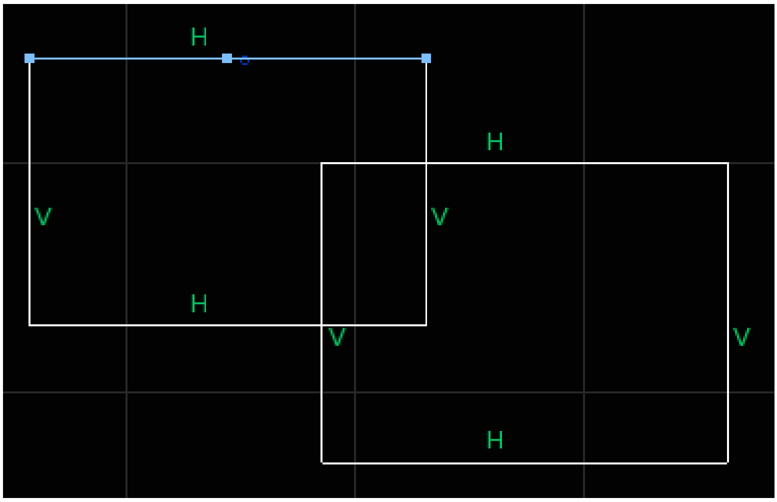


Figure A.34: Snap to Edges Tool

Snap to Edge/Edge Intersections

Mouse is snapped to the vertices of intersecting edges.

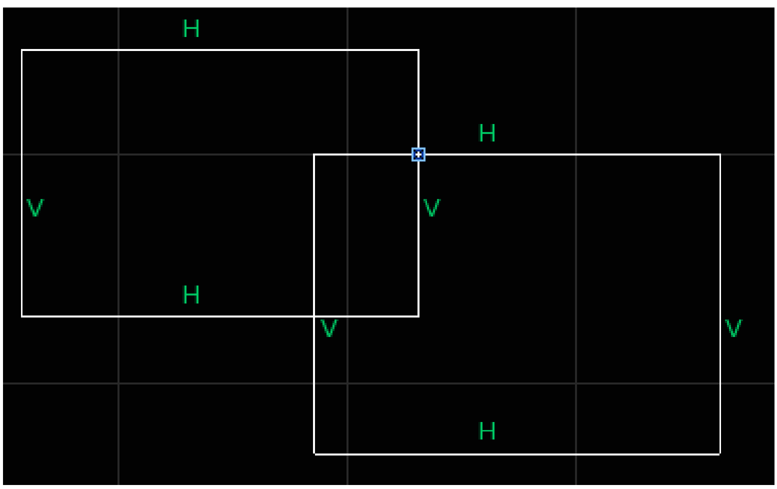


Figure A.35: Snap to Edge/Edge Intersection Tool

Snap to Part Cross Sections

When the cutting plane is active and the sketching plane intersects existing geometry, this tool will snap the mouse location to the edges and vertices formed by the part cross section.

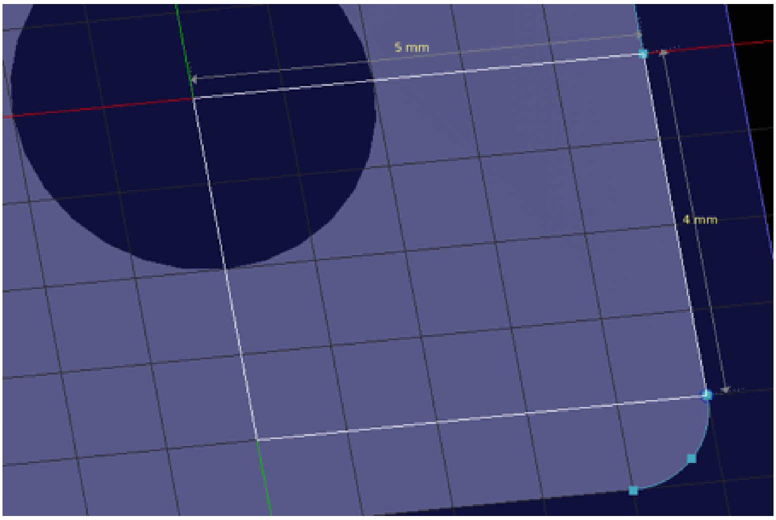


Figure A.36: Snap to Part Cross Sections Tool

A.3 2-D Modeling Options

The 2-D Modeling tools are used to outline or fill-in a simple geometry object.

Wire Body

The WIRE BODY tool is the simplest geometry object. Any of the Shape tools can be used to create the desired wire geometry.

Sheet Body

The SHEET BODY tool is similar to the WIRE BODY tool except its interior is filled with a material.

✓ It is also possible to create a sheet body using advanced options with 3-D modeling operations.

Sheet Body from Faces

The SHEET BODY FROM FACES tool allows the user to create a SHEET BODY from one or more faces of a pre-existing geometry object. The user will prompted to select one or more object faces by selecting them in the view (using CNTL-click for multiple selection) or by clicking on the **Select All Faces** button to select all faces.

If the ”Detach Faces” checkbox on the ”Select Faces” tab of the editor is checked, the new object will be detached from the source geometry. This means that if the source geometry is changed, the new Sheet Body will not change. Conversely, if the checkbox is not checked, the new Sheet Body will change if the source geometry changes.

A.4 3-D Solid Modeling Options

The 3-D Modeling tools are used to create simple solid geometry objects from 2-D forms.

For solid body creation, the 2-D sketch must be closed so that there are no lingering endpoints.

Extrude

EXTRUDE is used to sweep a face in the normal direction from its center. Once a 2-D form in made in the EDIT CROSS SECTION tab, select the EXTRUDE tab to its right to perform an extrusion. For a default extrusion, define the distance in the EXTRUDE DISTANCE dialog box by typing in a numerical value, parameter name, or equation.

If units are not entered next to the numerical value, the default units are assumed.

- See Section 13.1.1 for more information about defining distances with parameter names.

Additionally, the DIRECTION dialog box specifies the axis along which the extrusion will occur. Clicking done after the desired geometry is created will add the object to the project. It can now be seen in the PROJECT TREE.

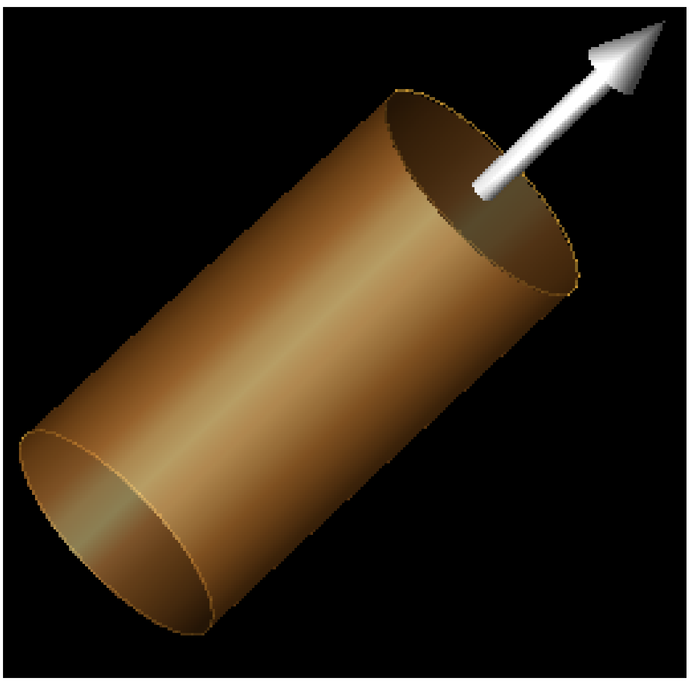


Figure A.37: Extrusion Tool

Revolve

REVOLVE is used to sweep a face in a circular path. Once a 2-D form is made in the EDIT CROSS SECTION tab, select the REVOLVE tab to perform a revolution. For a default revolution, define the angle in the ANGLE dialog box by typing in a numerical value, parameter name, or equation. The AXIS ROOT POSITION dialog specifies the location of the root of the axis around which the shape will revolve. The AXIS DIRECTION box specifies the direction along which the revolution will occur. Clicking DONE after the desired geometry is created will add the object to the project. It can now be seen in the PROJECT TREE.

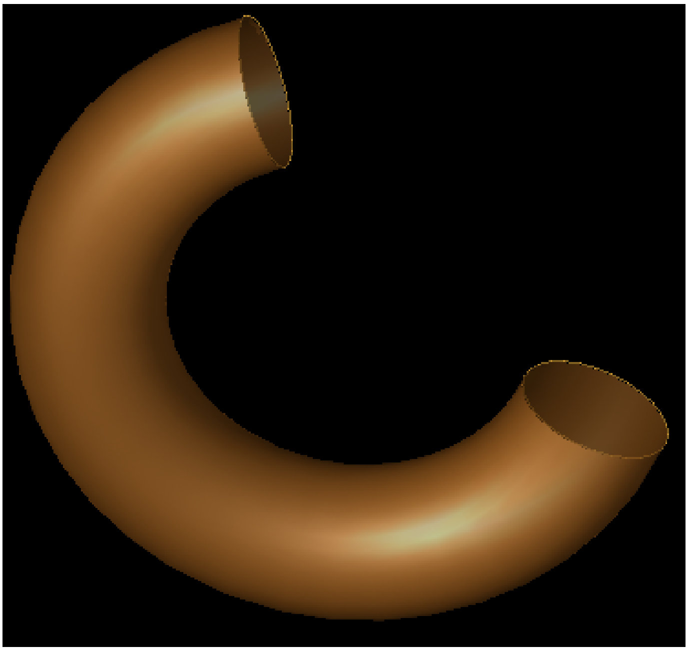


Figure A.38: Revolution Tool

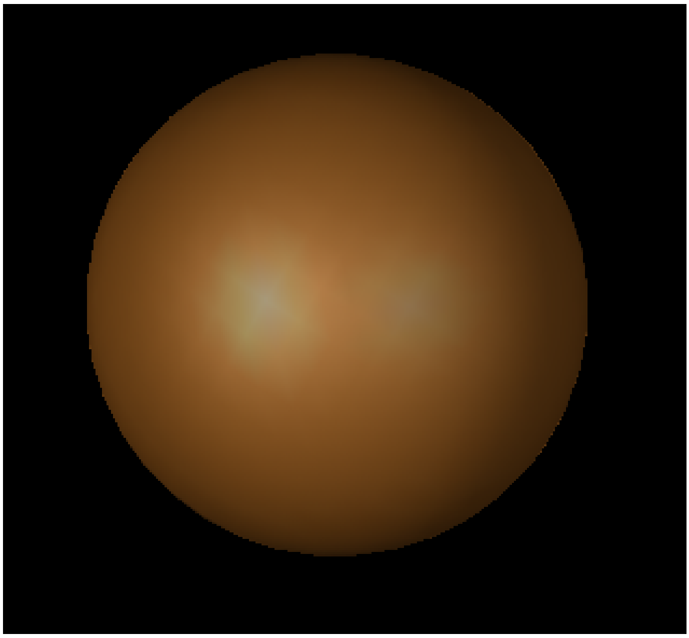


Figure A.39: Creating a sphere with the Revolution Tool

A.5 Advanced 3-D Solid Modeling Operations

The Advanced 3-D Modeling tools are used to modify a pre-defined 3-D geometry object. They are available within the EXTRUDE and REVOLVE operations.

Twist

Twist options control how much the face is twisted as it is swept. They can be specified by angle or law.

BY ANGLE: Specify the total number of degrees that the face will twist while it is swept.

BY LAW: Specify a mathematical expression to control the rate of twist as a function of the variable X.

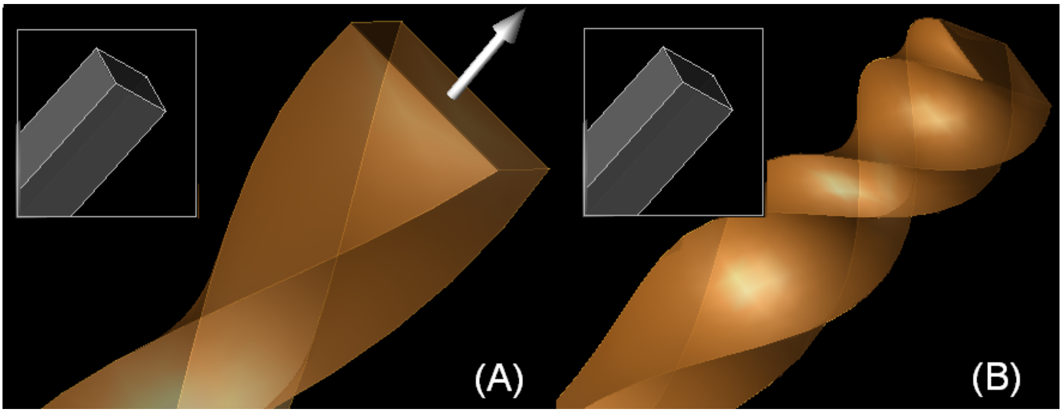


Figure A.40: Twist Tool defined by A) Angle (90 degrees) and B) Law $(0.8x^2)$.

Draft Type

Draft Type options control the expansion or contraction of the edges of the face as it is swept from its initial position.

NO DRAFT: No expansion or contraction of edges during sweep.

DRAFT ANGLE: Specify the expansion or contraction angle from initial position.

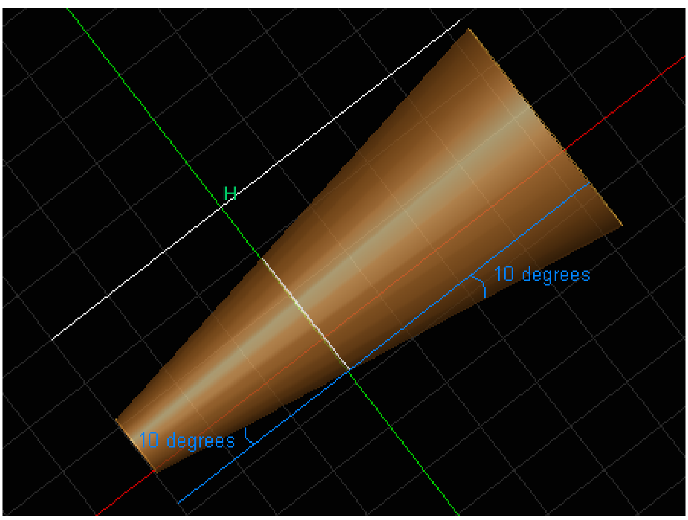


Figure A.41: A cylinder sweep with Draft By Angle (10 degrees)

DRAFT LAW: Specify a mathematical law to control the shape of the sides as the face is swept from initial position as a function of the variable X.

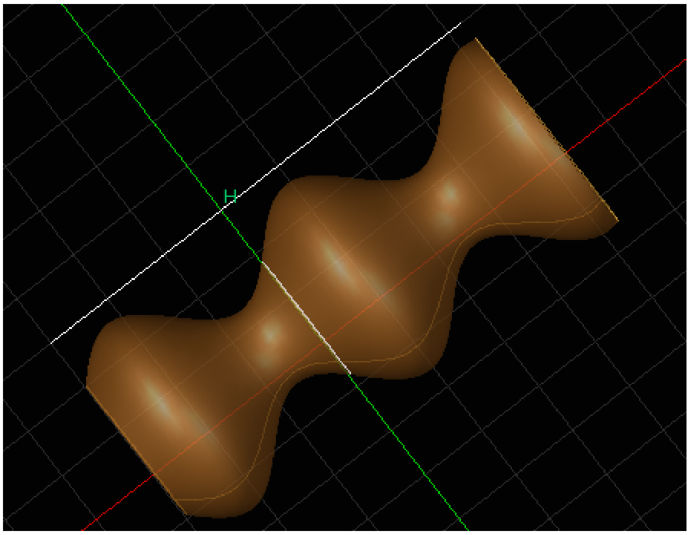


Figure A.42: A cylinder sweep with Draft By Law (.5sin(2x))

END DISTANCE/START DISTANCE: Specify the offset distance in the plane where the sweep ends/begins.

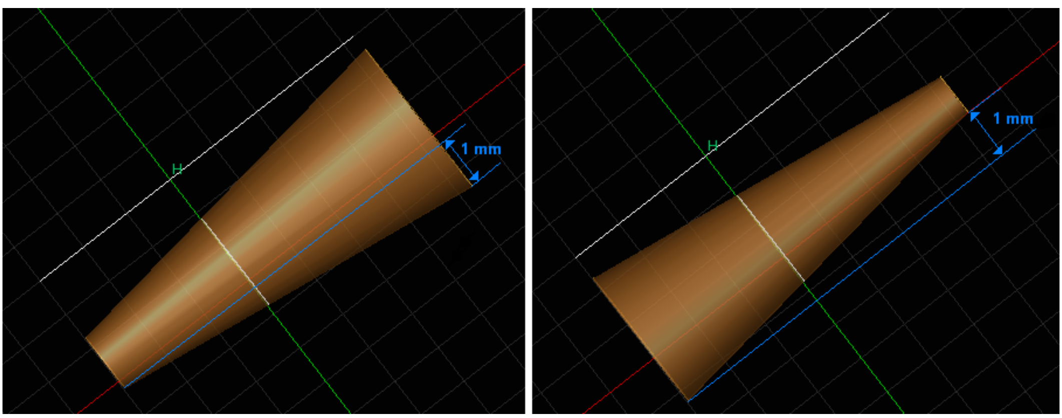


Figure A.43: A cylinder sweep with Draft By End Distance (1 mm) and Start Distance (1 mm)

Hole Draft Type

Hole Draft Type options control the expansion and contraction of a hole. They are therefore only valid during sweeping operations applied to a faces that contain holes. Hole Draft Type can be defined based on the values assigned to the edges in Draft Type options, or by angle.

NO DRAFT: No expansion or contraction is applied to the hole, even if the face has a DRAFT TYPE applied to it.

DRAFT ANGLE Specify the expansion or contraction angle from initial position.

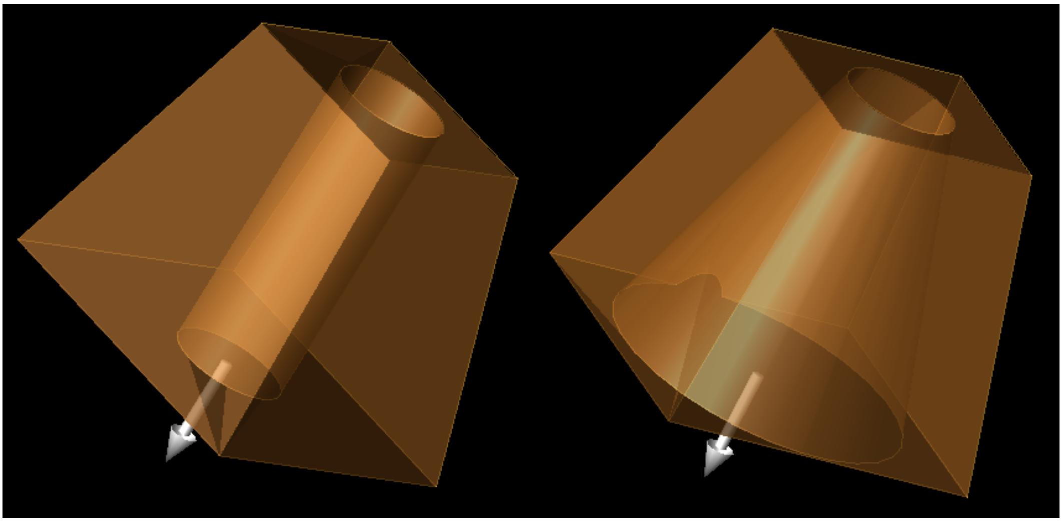


Figure A.44: Hole with no Draft (left) and a defined Draft Angle (right)

WITH PERIPHERY: The expansion or contraction of the hole will be the same as the outside edges of the face as specified in DRAFT TYPE.

AGAINST PERIPHERY: The expansion or contraction of the hole will be the opposite to the outside edges of the face as specified in DRAFT TYPE. (i.e.: the hole will contract as the face expands and expand when the face contracts.)

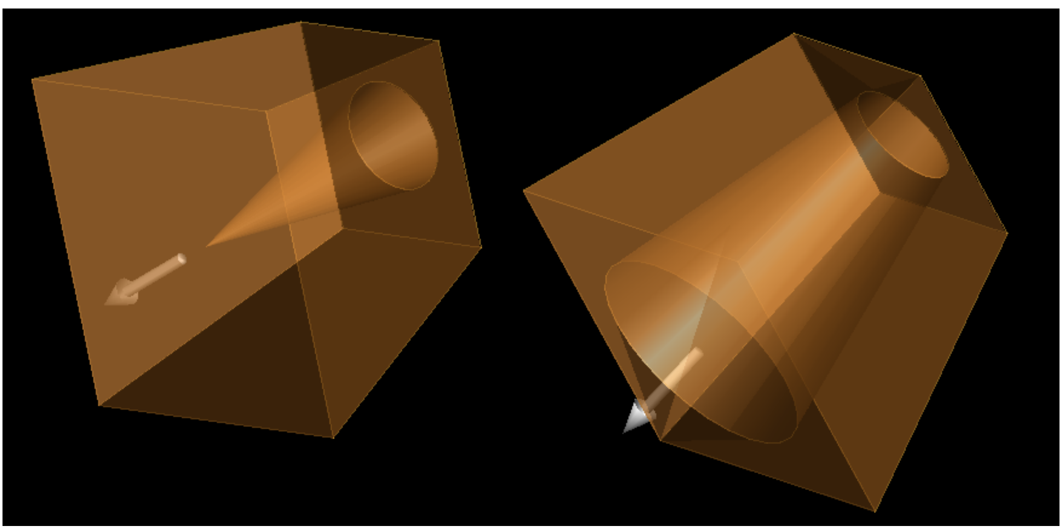


Figure A.45: Hole with Draft Angle against (left) and with the Periphery (right)

Gap Type Modeling Operations

The GAP TYPE specifies how to close the gap created by an offset. Use this feature when specifying a draft angle while creating an extrude. When you click the **Done** button to create the extruded part, the draft angle for the part often creates gaps. The GAP TYPE option you select tells the solid modeler how to fill those gaps. The default gap type is NATURAL, but the following options are available for filling gaps in the geometry.

NATURAL: Extends the two shapes along their natural curves until they intersect.

ROUNDED: Creates a rounded corner between the two shapes.

EXTENDED: Draws two straight tangent lines from the ends of each shape until they intersect.

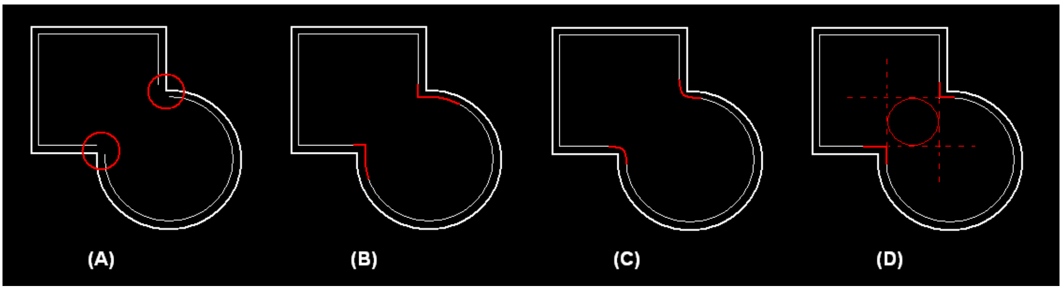


Figure A.46: 2-D Illustration of gap types, showing A) the original gap, B) Natural, C) Rounded and D) Extended

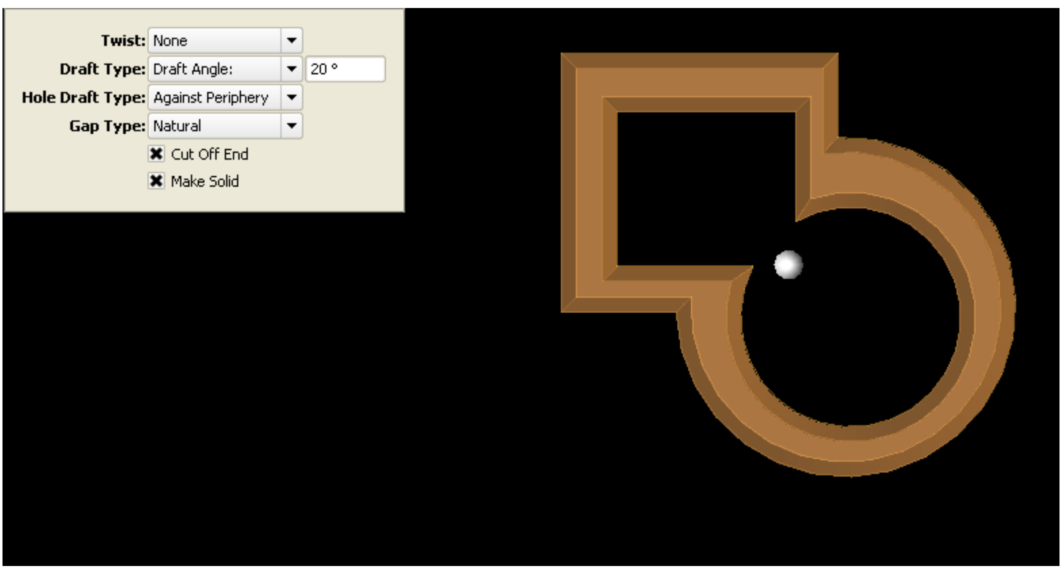


Figure A.47: 3-D Illustration of the Natural gap type

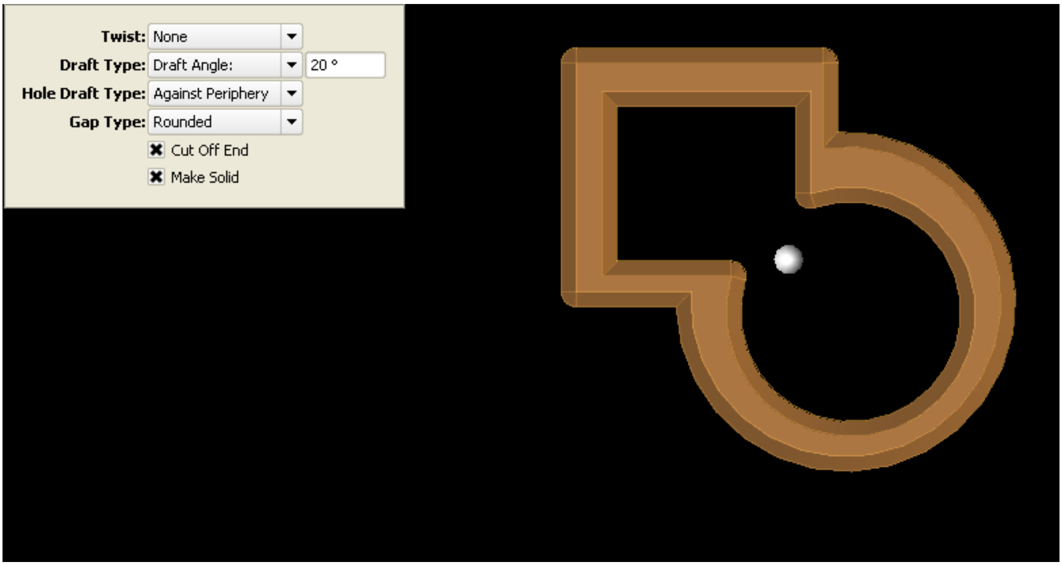


Figure A.48: 3-D Illustration of the Rounded gap type

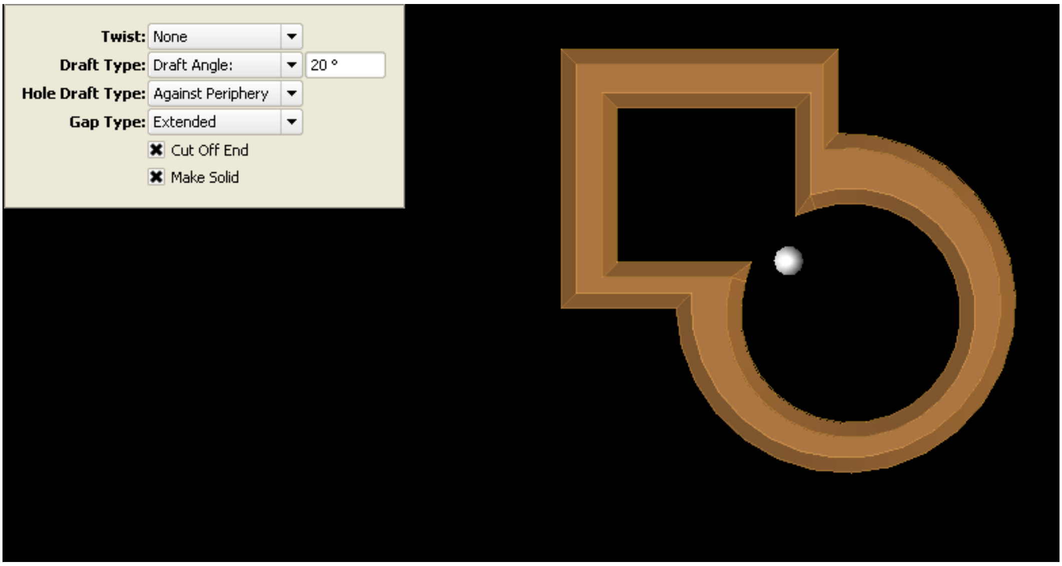


Figure A.49: 3-D Illustration of the Extended gap type

Cut Off End

Controls the orientation of a face that does not follow its normal during a straight sweeping operation. Select this option to chop the end of the swept 3-D object so that the normal of the end face is aligned with the line used for sweeping.

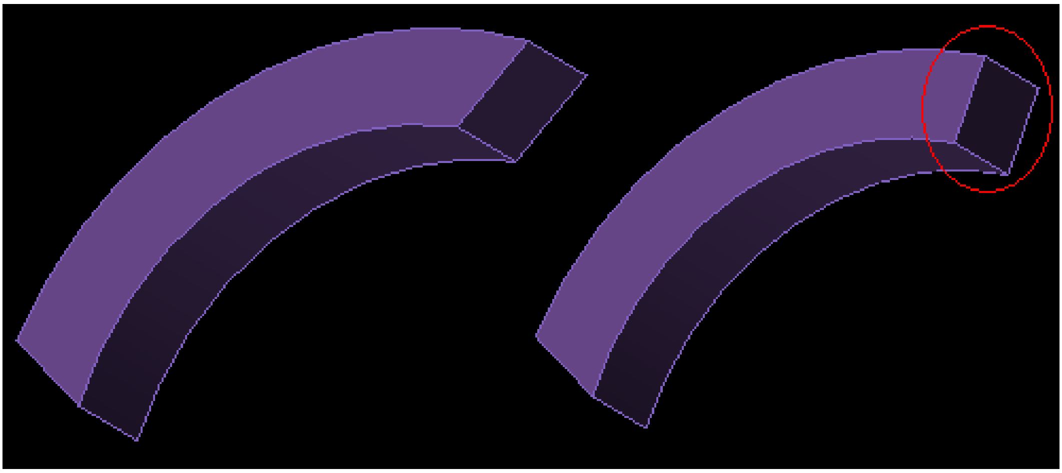


Figure A.50: Original Model (Left) and Model After Cut Off End (Right)

Make Solid

This option makes the model entirely solid. If this option is not selected, the model will be hollow.

A.6 Modifying Existing Geometry

Specify Orientation

The SPECIFY ORIENTATION button is used to position the selected geometry in the simulation space. Clicking this icon will bring up the SPECIFY ORIENTATION tab.

- See Section 5.5 for more on using the SPECIFY ORIENTATION tab.
- See Section 5.2 for descriptions of the tools used to rotate, translate and zoom into the geometry.

Chamfer Edges

CHAMFER EDGES operation creates a beveled edge between two surfaces. After selecting the edge, it will be trimmed at a 45° angle if CONSTANT DISTANCE is selected in the SPECIFY DISTANCE tab. Otherwise, the user enters the chamfer distance for the surfaces on the left and right sides of the edge.

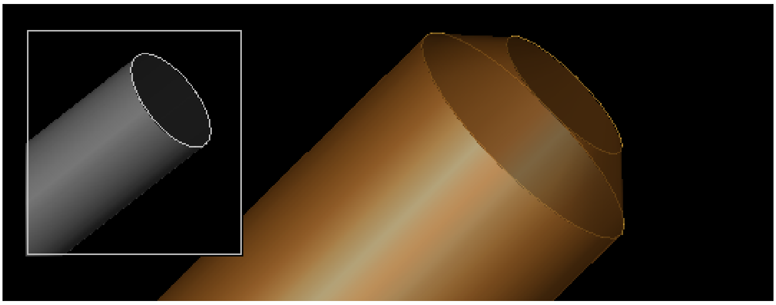


Figure A.51: A Chamfer operation applied to a cylinder edge

Blend Edges

The Blend Edges operation rounds the selected edge of the geometry. Under the Specify Radius tab, the user can enter the Blend Radius to adjust the rounding factor.

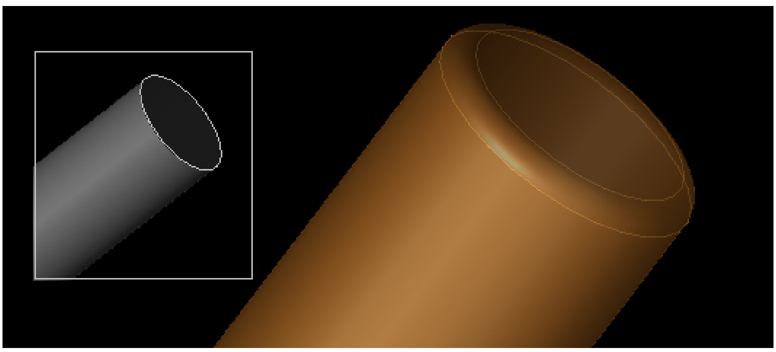


Figure A.52: A Blend operation applied to a cylinder edge

Shell

The Shell operation creates a shell from existing geometry. After selecting the faces to keep open, the user can enter the Shell Thickness under the Specify Thickness tab.

- By definition, the shell operation is used on geometry which is intended to have volume. This operation is not for use an object such as a Sheet Body, whose volume is insignificant in the XFdtd calculation.

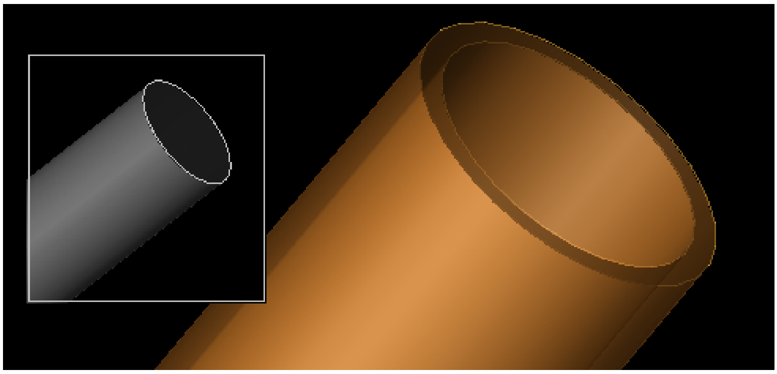


Figure A.53: A Shell operation applied to a cylinder

Offset Sheet Faces

With the OFFSET SHEET FACES operation, the user enters a positive or negative offset distance to increase or decrease the length of the SHEET BODY, respectively.

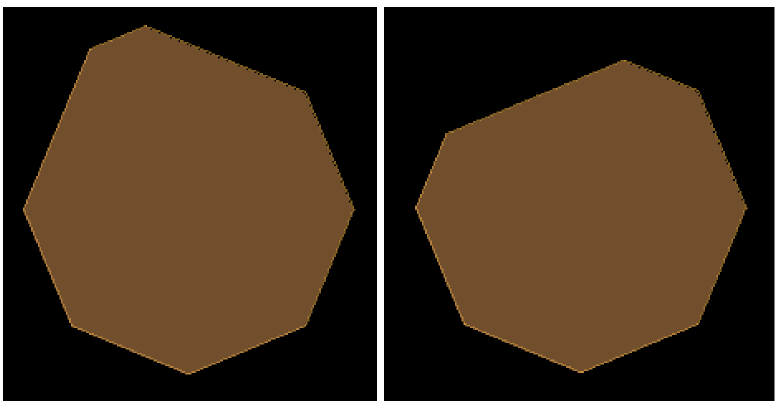


Figure A.54: An octagon with an edge offset in the positive direction (left) and the negative direction (right)

Thicken Sheet

The THICKEN SHEET operation is used to add thickness to a SHEET BODY .

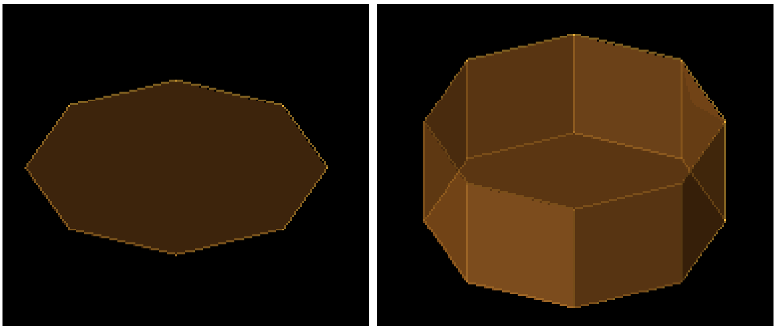


Figure A.55: The Thicken Sheet operation applied to an octagonal sheet body

Loft Faces

The LOFT FACES operation creates a new geometry by making a new solid that connects two chosen faces. Under the SPECIFY LOFT tab, the user can adjust the SMOOTHNESS FACTOR to adjust how closely the newly created edges match the trajectory of the existing geometry. When two faces of a single part are chosen, two options are available in the ADD LOFT AS drop-down list.

- **A Feature**: Adds a new feature to the part's modeling sequence. XFdtd uses this addition to contain the geometry within a single part.
- **A Separate Part**: Xfdtd adds the newly created loft geometry to the project as a new part.

When the faces are chosen from separate parts, two additional options are available in the the ADD LOFT AS drop-down list.

- **Part 1 Addition**: The new loft geometry is combined with the part from the first chosen face.
- **Part 2 Addition**: The new loft geometry is combined with the part from the second chosen face.

After the loft has been created, modifications to the original parts will cause the loft geometry to reapply to match the adjusted shapes.

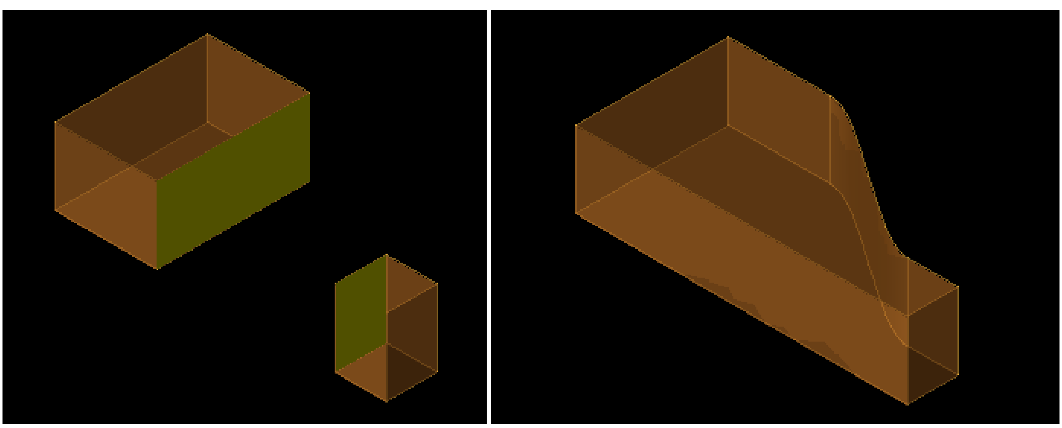


Figure A.56: Two objects within a geometry with faces selected (left) and later connected by a Loft (right)

Twist

The TWIST operation twists an existing part around an axis. The axis is specified by defining two points in space by direct entry of their coordinates or by using the picking tools. These points are rendered in the scene along with a plane at each point normal to the axis that denote the extents of the twist operation. I.e. The twisting will occur between the two planes. The amount of twist to apply at each plane as well as a continuity condition between twisted and untwisted sections of the part are also specified.

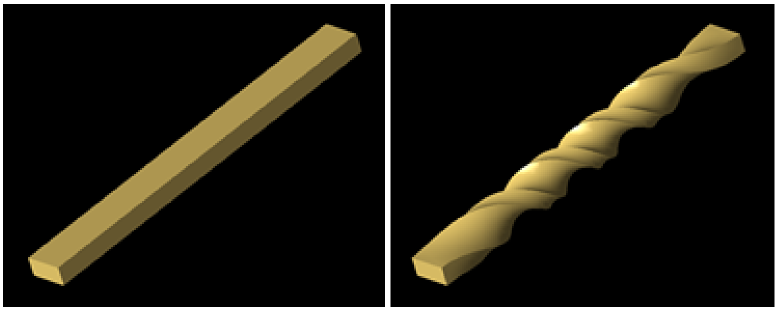


Figure A.57: An object with a twist applied, before (left) and after (right)

Stretch

The STRETCH operation stretchs an existing part along an axis. Like the TWIST operation, the axis is specified by defining two points in space by direct entry of their coordinates or by using the picking tools. These points are rendered in the scene along with a plane at each point normal to the axis that denote the extents of the stretch operation. I.e. The stretching will occur between the two planes. The amount of stretch to apply at each plane as well as a continuity condition between stretched and unstretched sections of the part are also specified.

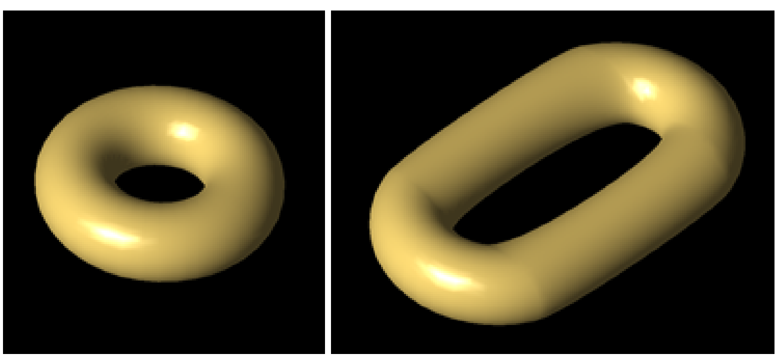


Figure A.58: An object with a stretch applied, before (left) and after (right)

Remove Faces

The REMOVE FACES operation removes a blend or chamfer that was previously applied to a geometry edge. This operation must be applied before the user can offset the length of an object. When a face is removed, a gap occurs and the object becomes a sheet body. To close the gap, click the CLOSE GAPS LEFT BY REMOVED FACES checkbox in the PREVIEW tab. When selected, XFdtd connects the faces around the gap. The part remains a solid object.

✓ This operation is useful for modifying objects that have been imported from CAD files.

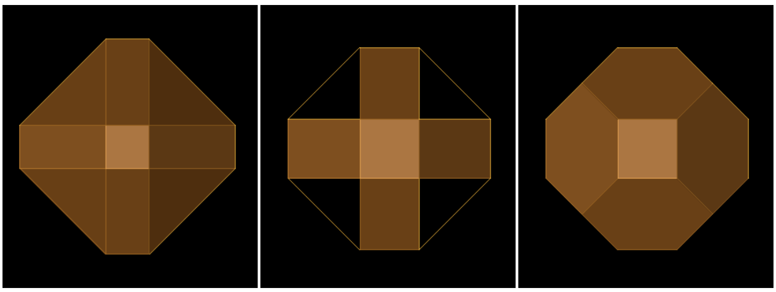


Figure A.59: An object (left) with several faces removed. With open gaps (center) and with closed gaps (right)

Offset Faces

With OFFSET FACES, the user enters a positive or negative offset distance to increase or decrease the length of the selected model, respectively.

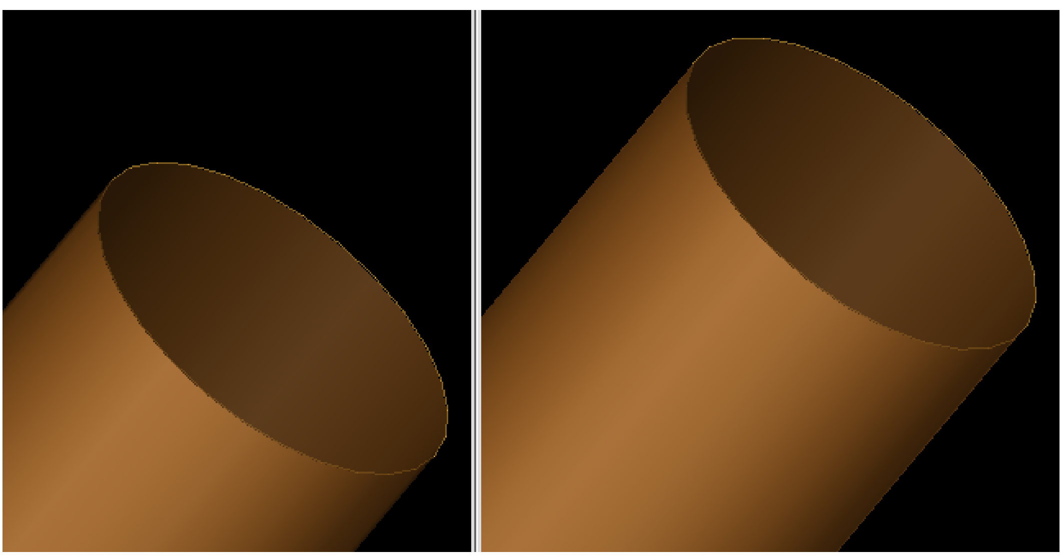


Figure A.60: A cylinder with an applied negative offset (left) and positive offset (right)

A.7 Boolean Operations

Existing Parts Boolean Operation

The EXISTING PARTS Boolean options perform operations on multiple geometric parts. In each case, at least one or more objects are identified as the TOOL (the part(s) used to perform the modification), and at least one or more objects are identified as the BLANK (the part(s) to be modified). There are four types of operations:

- SUBTRACT
- INTERSECT
- UNION
- CHOP

In the SUBTRACT and CHOP operations, the TOOL is subtracted from the BLANK. In the INTERSECT and UNION operations, the part(s) selected first is inconsequential.

With the CHOP operation, the BLANK turns out the same as in a SUBTRACT operation. A new object is added to the PARTS branch, which is the shape of the INTERSECTION.

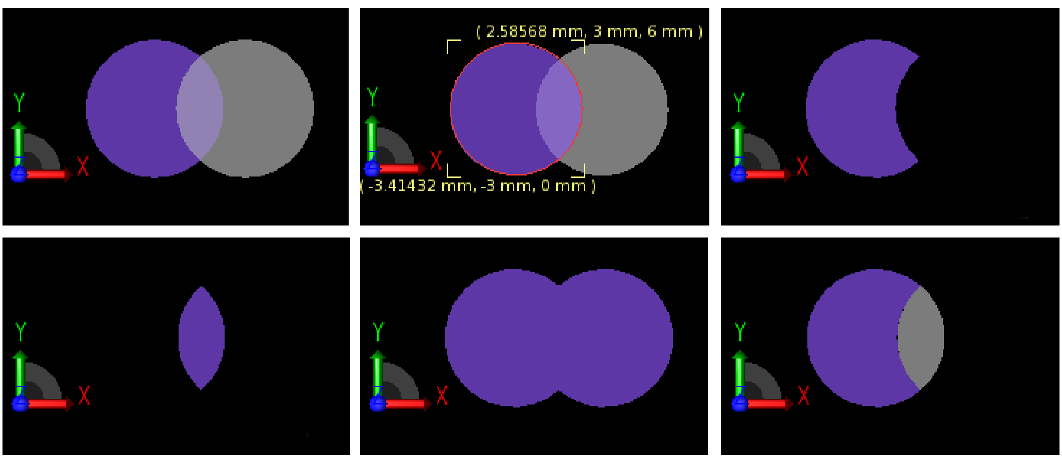


Figure A.61: Original objects (Upper Left), selection of the Blank (Upper Middle), objects after Boolean Subtraction (Upper Right), objects after Boolean Intersection (Lower Left), objects after Boolean Union (Lower Middle), and objects after Boolean Chop (Lower Right)

Extrude Boolean Operation

The EXTRUDE Boolean option performs an operation on a geometric part (or parts). In this case, the user chooses the BLANK, and then creates the object to use as the TOOL. The user then specifies the orientation of the extrusion and the nature of the operation (SUBTRACT, INTERSECT, UNION, or CHOP). In essence, this operation is a shortcut for the EXISTING PARTS Boolean operation.

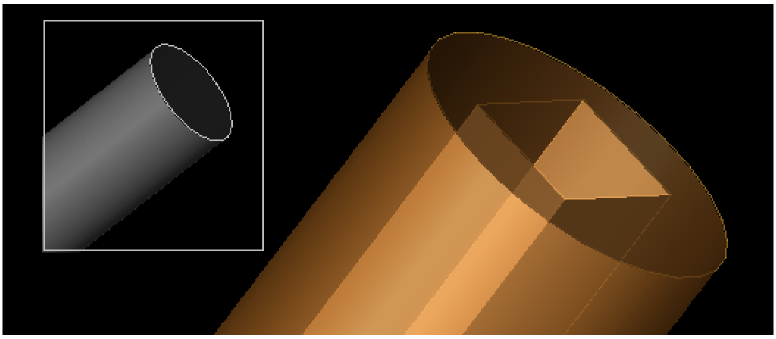


Figure A.62: A boolean extrude operation

Revolve Boolean Operation

The REVOLVE Boolean option performs an operation on a geometric part (or parts). The user chooses the BLANK, and then creates the object to use as the TOOL. The user then specifies the orientation of the revolution and the nature of the operation (SUBTRACT, INTERSECT, UNION, or CHOP).

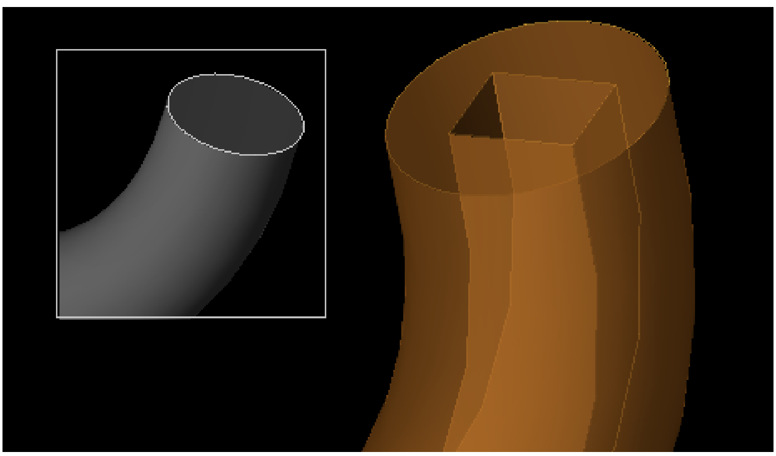


Figure A.63: A boolean revolve operation

A.8 3-D Patterns

Linear/Rectangular Pattern

The LINEAR/RECTANGULAR Pattern option allows the user to select a part in the geometry and replicate it in a linear pattern. After selecting the part to modify, define the SPACING and NUMBER OF INSTANCES in the U', V' and W' directions. Spacing refers to the distance between objects in the specified direction, and Number of Instances refers to the number of objects in the specified direction. For example, if three cylinders are to be spaced at 2-mm intervals in the U'-direction, the Spacing in the U'-direction is 2 mm and the Number of Instances in the U'-direction is 3. Additionally, the STAGGER check-boxes apply a stagger in the specified direction at every other instance in that direction.

Spacing refers to the distance between each object's center point in the specified object. So, for example, if the spacing between two cylinders does not exceed the distance of the cylinder's diameter, the cylinders will overlap.

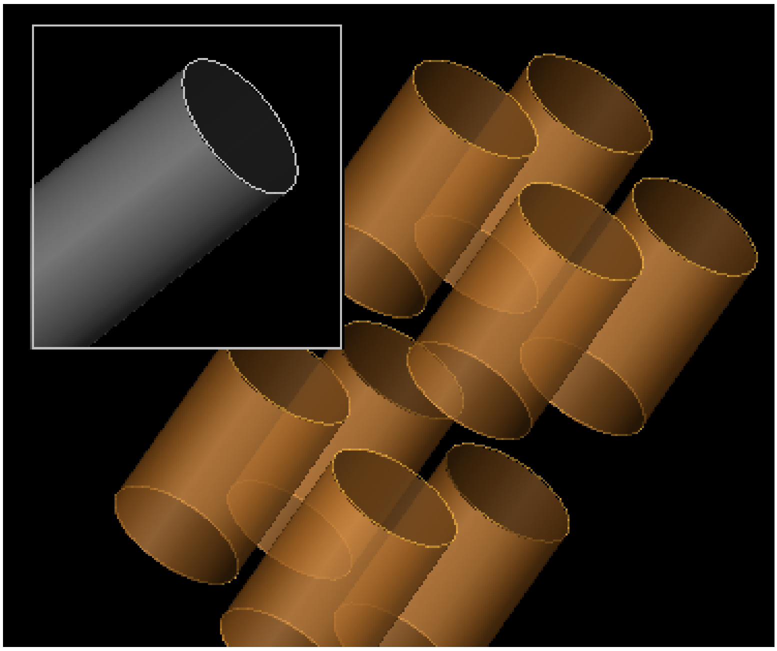


Figure A.64: A linear pattern applied to a cylinder

Circular/Elliptical Pattern

The CIRCULAR/ELLIPTICAL Pattern option allows the user to select a part in the geometry and replicate it in a circular or elliptical pattern. After selecting the part to modify, navigate to the SPECIFY CIRCULAR/ELLIPTICAL PATTERN tab and define the following fields:

- AXIS POINT - specifies the position of the axis
- AXIS NORMAL - specifies values to define the direction of the pattern
- ROOT POSITION (available in ELLIPTICAL MODE) - specifies a point (usually the center of a part) to use as the reference to replicate in the elliptical pattern.
- MAJOR AXIS (available in ELLIPTICAL MODE) - specifies the direction of the major axis
- PATTERN OPTIONS
  - INSTANCES - specifies the number of objects in the pattern
  - ANGLE - specifies the angle across which the objects are patterned (i.e., 180° means that objects are patterned across half of the ellipse)

– RATIO (available in ELLIPTICAL MODE) - specifies the ratio of the minor axis to the major axis

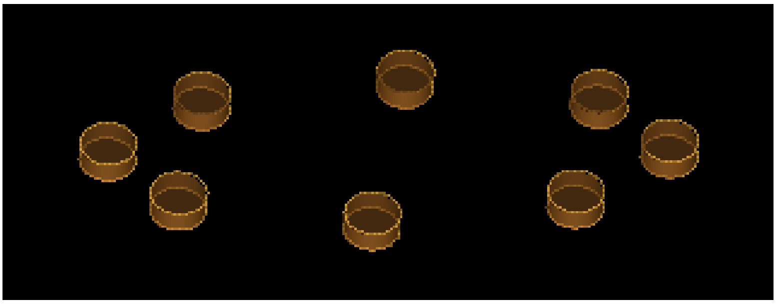


Figure A.65: An elliptical pattern applied to a small cylinder

A.9 Geometric Transformations

TRANSFORMATIONS are geometric operations available for use on imported CAD models and parts created through the feature-based modeling mechanism.

Scale

The object is scaled according to the values entered in the dialog. The user can choose to scale uniformly (all 3 axes at once) or choose the individual axes to scale.

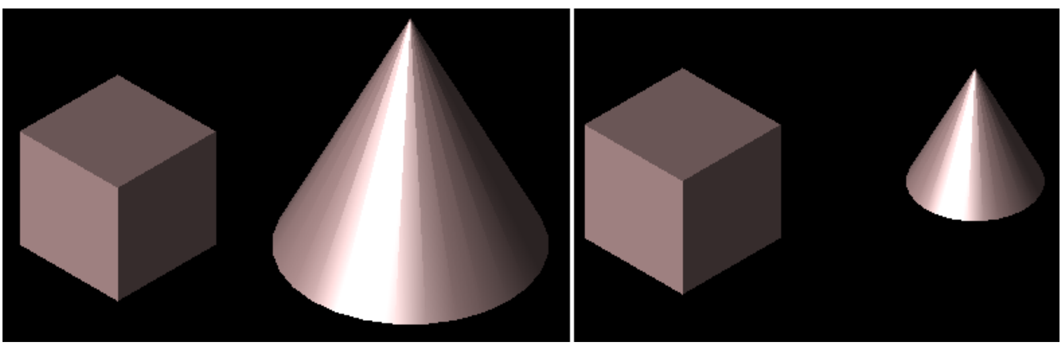


Figure A.66: Scaling an imported CAD cube to 50% of its original size

Translate

The object is translated according to the values entered in the dialog. The user can choose to translate along any or all of the 3 major axes.

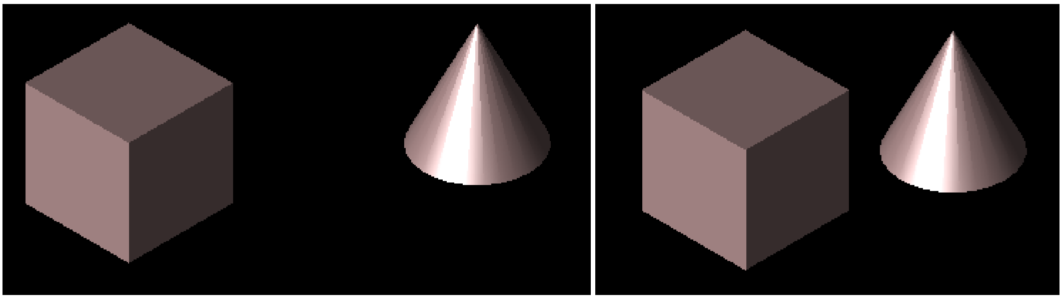


Figure A.67: Translating an imported CAD cube closer to the cone

Rotate

The object is rotated according to the values entered in the dialog. The user can choose to rotate around any or all of the 3 major axes.

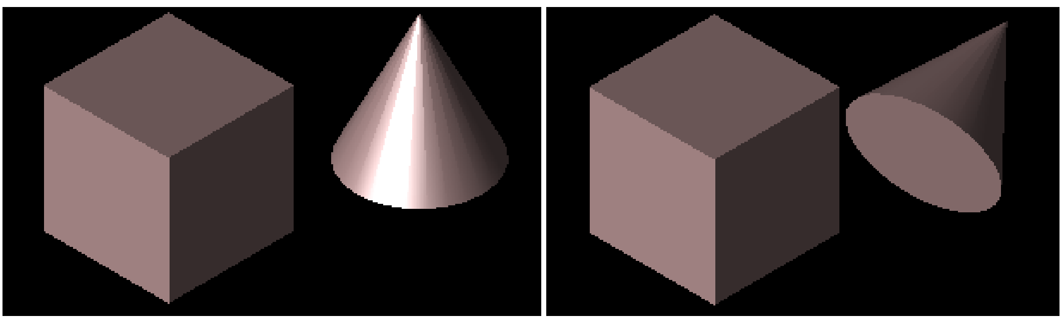


Figure A.68: Rotating an imported CAD cone

Reflect

The object is reflected according to the values entered in the dialog. The user can choose to reflect across any or all of the 3 major axes.

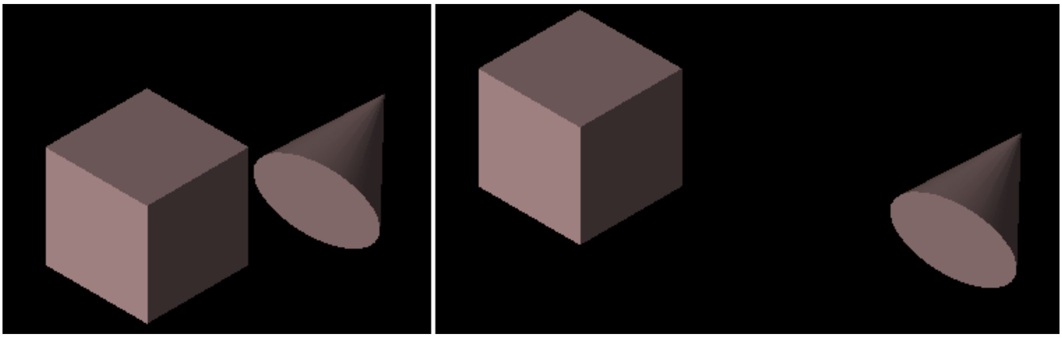


Figure A.69: Reflecting an imported CAD cube across the U' axis

Shear

The object is sheared according to the values entered in the dialog. The user can choose to shear along any or all of the 3 major axes.

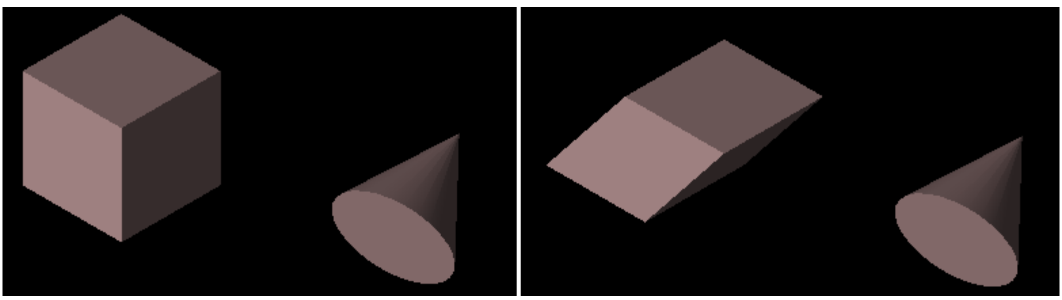


Figure A.70: Shearing an imported CAD cube at a 0.8 ratio

A.10 Flattening a Modeling Sequence

Using the feature-based modeling techniques available in XFdtd, it is possible to generate modeling sequences that are quite long or complicated. In some cases, it may be desirable to apply this entire modeling sequence as a permanent action, treating the resulting geometry as if it were an imported CAD object. This can be done using the FLATTEN operation.

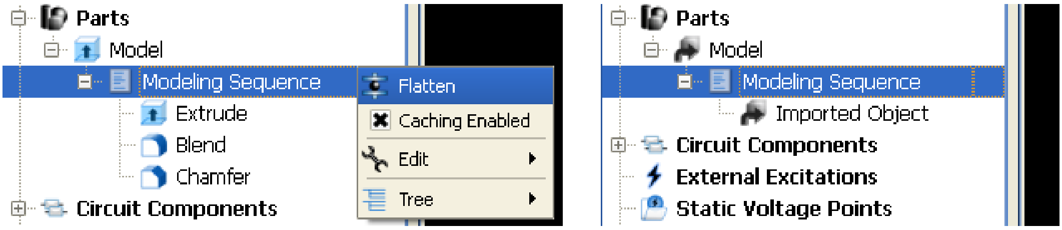


Figure A.71: Flattening a modeling sequence into a CAD object

Flattening a Modeling Sequence has the secondary benefit of allow you to heal the resultant geometry (i.e. remove its geometric irregularities), by right-clicking on the object in the PROJECT TREE and selecting HEAL.

Appendix B

Grid Appendix

In this appendix, you will learn...

→ the theory behind creating a grid in XFdtd

→ how to choose a cell size to optimize your XFdtd project calculation

→ methods for varying the XFdtd grid

→ how to debug the XFdtd grid

B.1 Grid Concepts Overview

The grid consists of three sets of points, one set for each axis: X, Y and Z. At each point is a plane. For example, consider the point at X = 3, which defines a plane in Y and Z. The point Y = 4 defines a plane in X and Z. The point Z = 5 defines a plane in X and Y. Where two planes intersect is a line. Planes from the remaining axis cut that line into edges. These edges are called "cell edges". Building the grid consists of defining the appropriate set of plane-defining points for each axis. Meshing, which occurs after gridding, is the act of assigning materials to each cell edge.

Figure B.1 displays intersecting planes in 3-D with lines and cell edges highlighted.

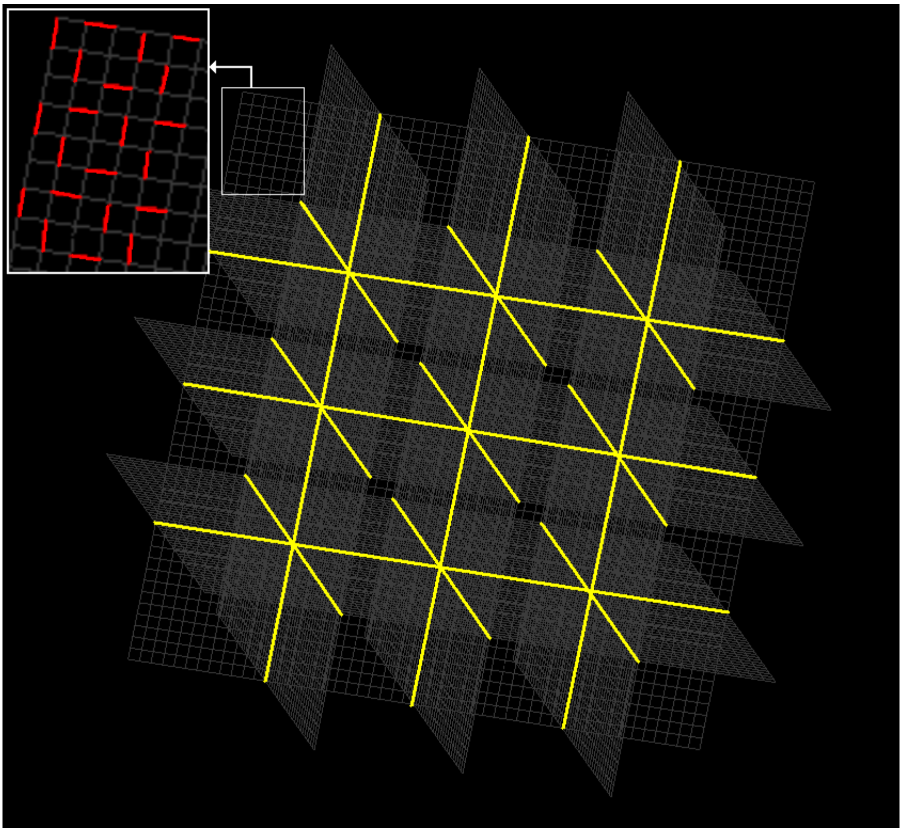


Figure B.1: Intersecting planes with lines (in yellow) and sample cell edges in the upper-left corner of the XY-plane (in red)

Except for some special features associated with components, which are discussed below, the grid is made up of the following elements:

- Fixed points
- Grid regions
- Target cell sizes
- Automatic fixed point minimum distances

A fixed point is a point on an axis at which a plane, in the other two axes, exists. Edge-on, that plane is seen as a line. That is what is meant by a “grid line”. A grid region is a bounded part of the grid. Grid lines are placed at the grid region boundaries. Within a grid region the target cell size and the fixed point minimum distance can be different from the project’s default target cell size and/or minimum distance. One fixed point is placed at the beginning of each grid region and another fixed point is placed at the end of each region.

Figure B.2 displays intersecting planes in 3-D with fixed points, grid lines and a grid region highlighted.

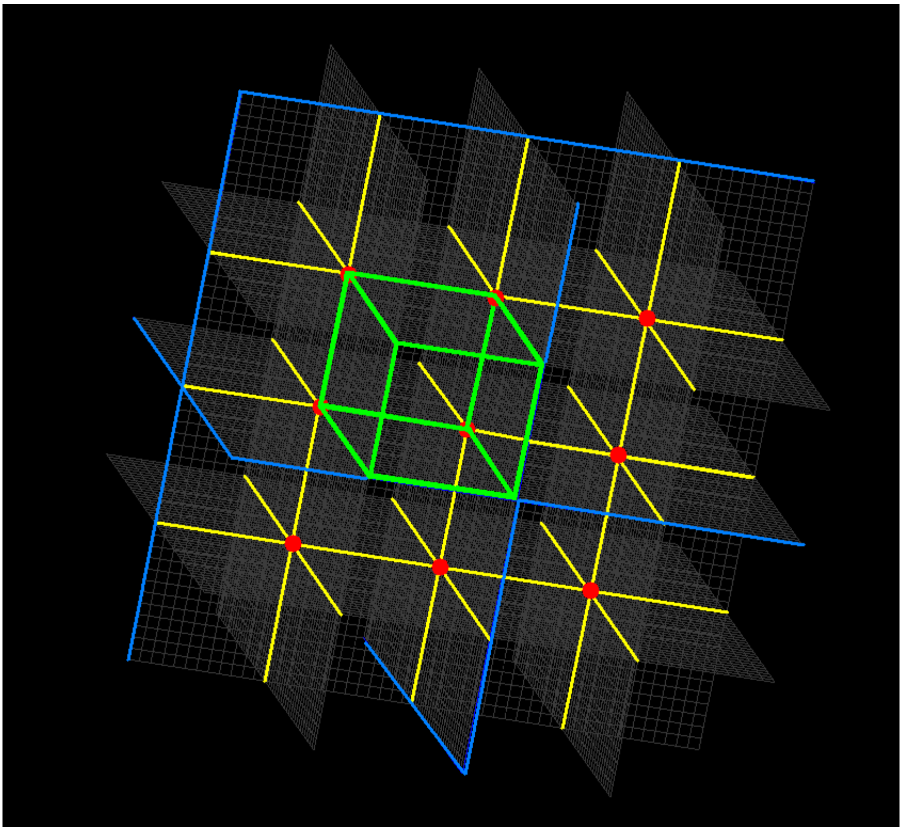


Figure B.2: Intersecting planes with fixed points (in red), grid lines (in blue) and a grid region (in green)

Manual fixed points and manual grid regions are specified in the GRID TOOLS dialog. Other grid regions and fixed points, associated with individual parts, may be specified using the GRIDDING PROPERTIES EDITOR. Grid regions associated with a part are called part grid regions. Fixed points associated with parts are called automatic fixed points, because they are automatically extracted from the geometry of the part.

- See section 8.1 for more on the GRID TOOLS dialog.
- See section 8.2 for more on the GRIDDING PROPERTIES EDITOR.

Taking these elements together, the grid is made up of manual fixed points, automatic fixed points, manual grid regions and part grid regions. Each kind of grid region contains a target cell size and an automatic fixed point minimum distance. Those values are used when creating the grid.

It is important when building a grid to control the size of the smallest cell in the grid. The project's timestep is derived from the smallest grid cell's edge length. Smaller timesteps result in longer runs and larger timesteps result in faster runs. For that reason it is important to prevent the grid from having a cell edge smaller than what is necessary to get the desired results.

Automatic fixed points and part grid regions are extracted from geometry and may result in points so close that the timestep is smaller than desired. The fixed point minimum distance is used to merge automatic fixed points in order to provide control of the timestep. Automatic fixed points are merged so that they are no closer than the minimum distance. If a grid region start or end boundary is too close to another grid point, the grid region may be expanded to prevent the too-small timestep. The grid region is always expanded, never contracted, in these situations.

The grid is created in the following steps. Each axis, X, Y and Z, is considered separately. Note that when considering only a single axis, a cell size is really just an edge size.

➔ Create a set containing manual and automatic fixed points and the fixed points from the borders of manual and part grid regions. If the SPECIFY PADDING option is chosen, then fixed points for overall bounds of the geometry are added to the set.

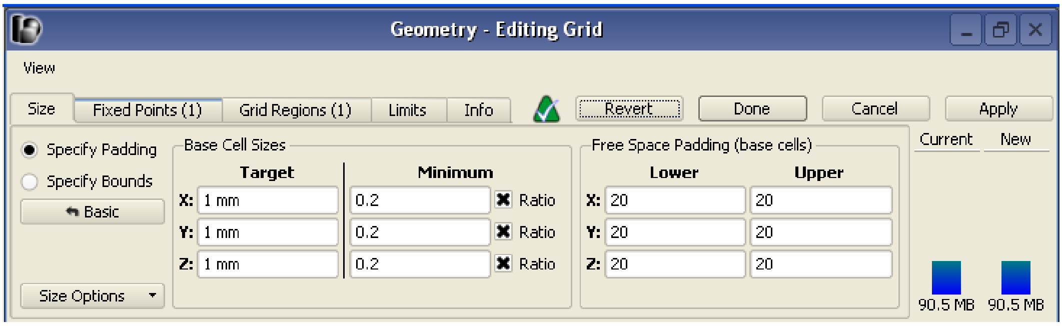


Figure B.3: Editing the grid with the Specify Padding option chosen

➔ Automatic fixed points are merged according to the minimum distance. The fixed point minimum distance can have different values at different points on the grid. There is a "main grid" fixed point minimum distance that is specified on the SIZE tab (see Figure B.3 above). Each grid region, including both manual and part grid regions, has its own fixed point minimum distance. The smallest fixed point minimum distance for a given point on the grid is chosen from all manual and/or part grid regions covering the given point. If no grid region covers the given point, the main grid fixed point minimum distance is chosen. Note that this allows grid regions to specify a minimum distance greater than the main grid fixed point minimum distance. Notice that in Figure B.4, a smaller minimum distance is specified in the grid region than that of the main grid.

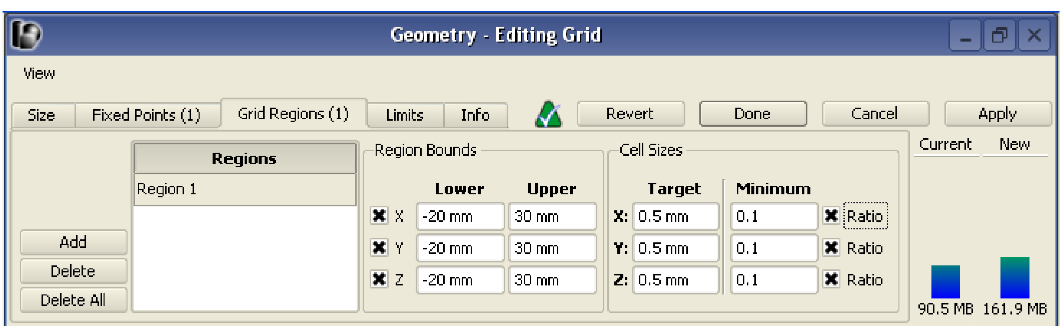


Figure B.4: Adding a grid region with a smaller cell size than the main grid

➔ The movable points, i.e. automatic fixed points and grid region start/end boundaries, are moved away from unmovable points. Unmovable points include manual fixed points and some entries associated with components, as described below. Remember that grid regions never shrink in order to ensure that regions needing improved accuracy get it.

➔ The set of grid points is examined. Transition regions are added to prevent adjacent cell size ratios from violating the maximum cell step factor specified on the LIMITS tab. For example, a transition region is generated if the maximum cell step factor is 2, and 2 adjacent cells have sizes 5 mm and 1 mm. A transition region contains the fewest number of cells required to reach the desired cell size. Each cell in the transition region has a progressively larger size.

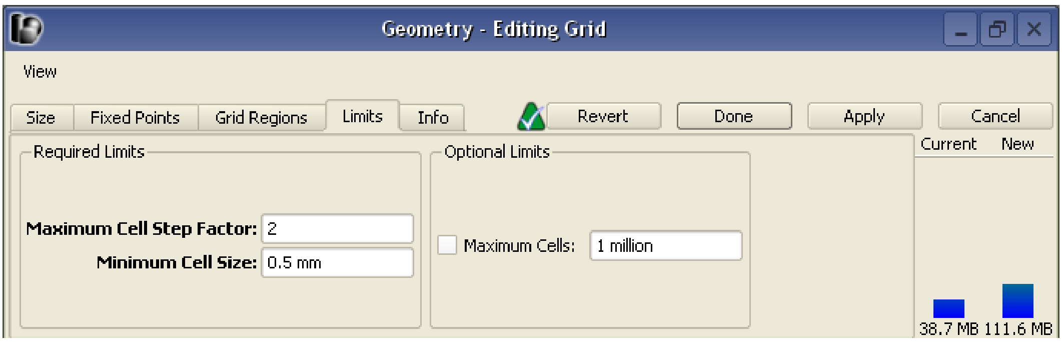


Figure B.5: The Maximum Cell Step Factor shown is never exceeded by adjacent cell size ratios

➔ The set of grid points, including the transition regions, is examined. Gaps greater than the target cell size at the given point are filled evenly with the fewest number of cells required such that the cell size is less than or equal to the target. For example, consider a gap of 9.7 mm with a target cell size of 1 mm. In this case, 8 new points must be added to the 2 points surrounding the gap. The distance is $9.7\ mm - 1\ mm = 8.7\ mm$, bridged by 9 cell edges. Each cell edge will be $\frac{8.7mm}{9edges} = 0.9667\ mm$ per edge.

That completes the calculation of grid points. There are some key facts to note:

- No matter what the fixed point minimum distance may be, it is always possible for there to be two fixed points a distance of $(minimumdistance + \epsilon)$ apart after the merging has been completed. If the target cell size at the given point is less than $(minimumdistance + \epsilon)$, then this distance must be subdivided into two or more cell edges.

  Consider a 0.7 mm minimum distance with a .2 mm target size. Given that $\frac{0.7mm}{0.2mm} = 3$ remainder 1, there must be 3 grid lines in the 0.7 mm gap. Those 3 grid lines create 4 spaces in the gap. The gaps are $\frac{0.7mm}{4} = 0.175\ mm$ each. The formula is:

$$\frac{minimumdistance}{\lceil \frac{minimumdistance}{target} \rceil}$$

- The target cell size may be greater than the automatic fixed point minimum distance. Consider a minimum distance of 0.7 mm and a target distance of 1 mm. If the automatic fixed points are dense then every gap is exactly 0.7 mm. No gaps are greater than 1 mm, so the work is done. If there was at least one region of sparse automatic fixed points then there may be a gap greater than 1 mm. That gap must be bridged using one or more extra grid lines. The smallest such gap is $\frac{target}{2}$, or

0.5 mm in our example.

- The smallest cell in the grid will be the smaller of $\frac{minimumdistance}{\lceil \frac{minimumdistance}{target} \rceil}$ and $\frac{target}{2}$. Any given grid may not encounter one or both of those situations, so that grid's smallest cell size may be larger.

B.2 Choosing an Appropriate Cell Size

Since smaller cells require longer calculation time, it may be advantageous to also define the lower limit of the cell size in the SIZE tab of the GRIDDING PROPERTIES EDITOR window. The minimum cell size can be defined as a MINIMUM distance (i.e., a specific distance with units), or as a ratio of the MINIMUM value to the TARGET base cell size (i.e., a MINIMUM value of 0.8 would restrict the minimum cell size from dropping to a value below 80% of the TARGET base cell size.)

- See the equations at the end of the previous section (B.1) to learn about the factors that affect the smallest cell size.

When defining the TARGET base cell size, ensure that the cell size is much less than the smallest wavelength for which accurate results are desired. A commonly applied constraint is *ten cells per wavelength*, meaning that the side of each cell should be less than one-tenth of the wavelength of the highest frequency (shortest wavelength) of interest. If the cell size is much larger than this, the Nyquist sampling limit is approached too closely for reasonable results to be obtained. Significant aliasing is possible for signal components above the Nyquist limit.

Choosing a cell size of one-tenth of a wavelength is a good starting point, but other factors may require a smaller cell size to be chosen, such as small geometry features and material characteristics.

Grid definitions can be customized for specific objects in the GRIDDING PROPERTIES EDITOR window, so that smaller features are considered, without having to apply smaller, memory-intensive cells to the whole grid.

- See Section 8.2 for more information about assigning grid deinitions to specific object.

Material characteristics will also influence cell characteristics since XFdtd is a volumetric computational method. If some portion of the computational space is filled with penetrable material, the wavelength in the material must be used to determine the maximum cell size. Geometries containing electrically dense materials require smaller cells than geometries that contain only free space and perfect conductors. For this reason, a material definition must be applied to PARTS objects to generate a valid mesh. An error message ⚠ will appear in the case that a material is not assigned to an object.

- See Section 6.1 for information on applying material definitions to objects.

B.3 Grid Regions vs. Fixed Points

There are two primary means of varying the grid in XFdtd. A grid region is a region within the grid that is assigned its own target cell size, which is different from the default grid size defined in the main SIZE tab of the GRID TOOLS button. A fixed point is a point on an axis where a grid line will be placed. Cell sizes are adjusted to flow smoothly between fixed points, never exceeding the TARGET cell size.

The target cell size can vary within different grid regions along a given axis. The main grid's target cell size applies everywhere except within grid regions. A grid region has start and end boundaries on an axis. Grid regions can have target cell sizes and automatic fixed point minimum distances that differ from the main grid and from other grid regions. Grid regions can overlap. When they do, the smallest target cell size and the smallest automatic fixed point minimum distance are chosen from all of the overlapping grid regions at the given point.

Grid regions, like fixed points, can be manual or automatic. Manual grid regions are defined on tabs associated with the main grid editor. Automatic grid regions are associated with parts and so are also called part grid regions. Part grid regions are defined by right-clicking on a part in the tree and choosing Gridding Properties.

Any grid region includes fixed points for the grid region boundaries. Between the bounds, the target cell size and automatic fixed point minimum distances can be different than their values outside the grid region bounds.

The grid flows as evenly as possible between fixed points, using at each point the appropriate target cell size and automatic fixed point minimum distance for that point.

Figure B.6 shows a simple shape that has a uniform grid size of 1 m.

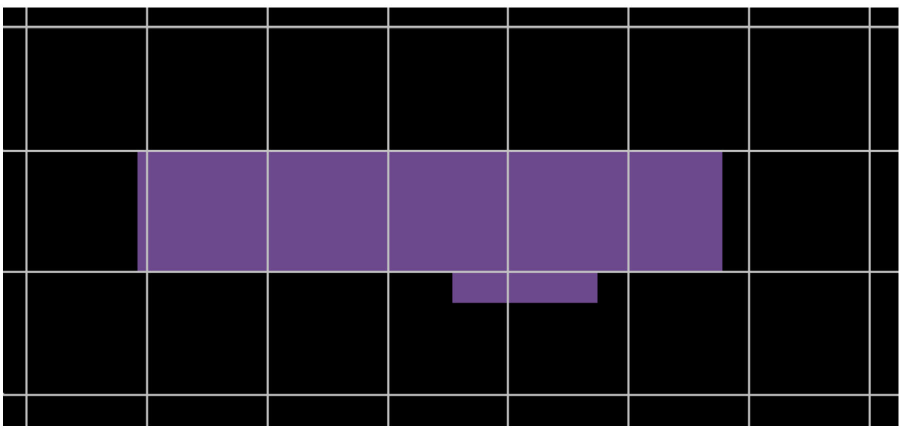


Figure B.6: Simple extrusion with uniform grid

Figure B.7 applies an automatic grid region to this simple shape. Note that the edges of the rectangle are now aligned with the edges of the grid. Also note that, because the height of the rectangle was not evenly divided by 1 m, the main grid spacing adjusted slightly to accommodate for this.

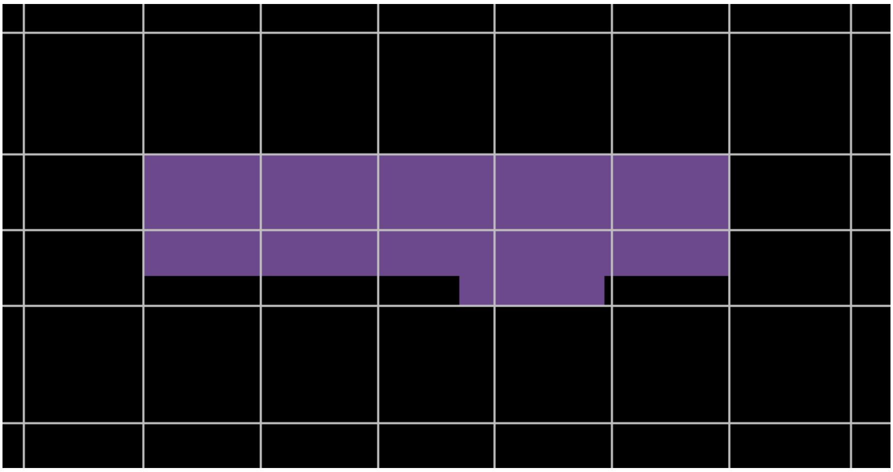


Figure B.7: Simple extrusion with applied grid regions

Figure B.8 applies automatic fixed points to the shape (all default settings in the GRIDDING PROPERTIES EDITOR are applied). Note that the edges, like in the case of the applied grid regions, are aligned with the rectangle's edges. The cells within the shape, however, are auto-generated, and therefore vary from the default target cell size as little as possible.

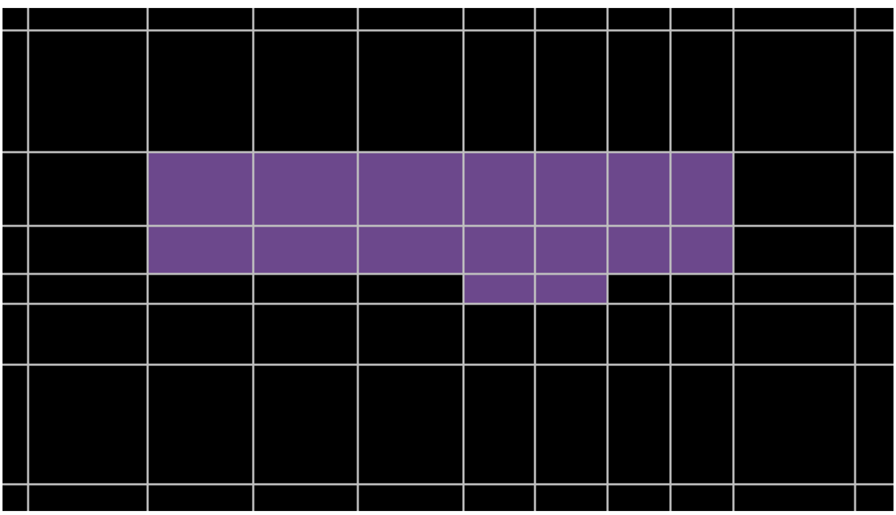


Figure B.8: Simple extrusion with applied fixed points

Figure B.9 applies a user-defined grid region to cover the area of the shape, with cell sizes significantly smaller than the main grid. Note the transition region of cells surrounding the shape. This region contains cells of non-uniform size, which vary from the CELL SIZE defined for the specific grid region to the CELL SIZE defined for the entire grid, at a rate which is limited by the MAXIMUM CELL SIZE STEP FACTOR defined in the LIMIT tab of the EDIT GRID dialog in GRID TOOLS.

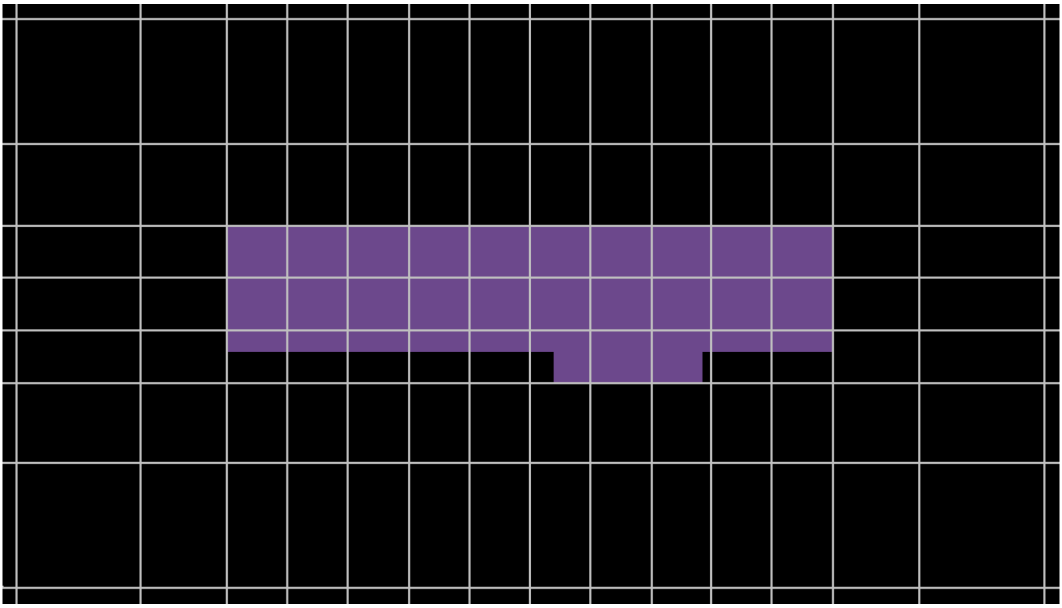


Figure B.9: Simple extrusion with a manual grid region defined in Grid Tools

B.4 Debugging the Grid

There may be times when one does not understand how certain XFdtd grid features in a given grid came to exist. The individual grid features may be turned off so that the grid becomes, in essence, a blank slate. Individual grid features can then be turned back on one at a time. Seeing how the grid changes in this step-by-step fashion makes it possible to locate the cause of any given grid feature.

The following steps outline how to turn off individual grid features.

➔ Be sure to save a copy of the project and open the copy for use in the debugging process.

➔ View the mesh and use the MEASURE tool to measure the geometry's largest extent.

➔ Open the GRID TOOLS dialog:

— Under the SIZE tab, set the X, Y, and Z TARGET sizes to something just larger than the geometry's largest extent.

— Set the MINIMUM sizes for each axis to e.g. 1e-12 mm (not a ratio).

— Set the FREE SPACE PADDING cell values to zero or possibly one.

— Delete all entries under FIXED POINTS and GRID REGIONS.

➔ If your project includes any CIRCUIT COMPONENTS, ensure that the box labeled EVENLY SPACED IN ORTHOGONAL DIRECTIONS is unchecked in the CIRCUIT COMPONENT PROPERTIES: PROPERTIES tab.

> Be sure to turn this property back on (where applicable) later in the debugging process, as accurate results in many cases depend on even component spacing.

➔ Right-click on a part under PROJECT TREE: PARTS and choose VIEW PARTS LIST

— Click on the (Grid Region) column heading to bring parts with Grid Regions to the top of the list.

  – Right-click on the part in the flat parts list and select GRIDDING PROPERTIES. Turn off the part's AUTOMATIC GRID REGIONS and/or FIXED POINTS.

— Click on the (Fixed Points) column heading to bring parts with Fixed Points to the top of the list.

  – As was done for parts with grid regions above, turn off any remaining parts with automatic fixed points chosen.

➔ Under the GRID TOOLS: LIMITS tab, set the MAXIMUM CELL STEP FACTOR to something large, e.g. 20000. Set the MINIMUM CELL SIZE to something very small, e.g. 1e-12 mm.

After performing those steps the grid should be relatively bare, consisting only of the geometry bounds and padding cells, if any were retained.

Now begin turning on grid features one at a time.

➔ Choose an important part and turn on its fixed points (in its GRIDDING PROPERTIES editor). Turn off all AUTOMATIC DISCOVERY OPTIONS except one, clicking APPLY. Turn off that extraction type and turn on the next one, examining each type in turn.

— As you examine the fixed points for the part, experiment with its automatic grid region. Turn it on and initially set its TARGET value to the same value as the main grid's target (under GRID TOOLS). Experiment with different MINIMUM distances, clicking apply each time. At first, leave the part grid region's target cell size the same as the main grid's target cell size.

— Choose the best combination of fixed point extraction types and minimum distances for the part.

➔ Go to another important part, in turn, until all parts for which automatic fixed points have been examined. Ensure that automatic fixed points are turned on for those parts and that an appropriate minimum distance was chosen for each.

➔ Review each important part. Experiment with the target cell sizes for each part, setting the target cell size to the value desired for the final grid.

At this point the grid probably contains adjacent cells whose widths vary by more than is allowed by FDTD theory. Go to the GRID TOOLS: LIMITS tab and change the MAXIMUM CELL STEP FACTOR to 2.0, being sure to tab out of the field to make the new setting take effect. Transition regions will appear. You may wish to experiment with values lower than 2.0, although the value must be greater than 1.0. If the MAXIMUM CELL STEP FACTOR is too low, the grid will report that a grid could not be created. Choose a higher step factor that is not greater than 2.0.

The grid probably will contain large gaps even after the transition regions have been added. Change the main grid's target cell size to its final desired value to see the remaining gaps filled with grid lines spaced appropriately.

Appendix C

Transitioning From Previous Versions

In this appendix, you will learn...

→ where to find tutorials on transitioning to XFdtd 7

→ what to expect when importing Version 6 files

The aim of XFdtd 7 is to improve workflow efficiencies and save time in the process of creating and simulating electromagnetic calculations. For users of previous versions of XFdtd, there will be a transition period to becoming acquainted with the new user interface and XFdtd 7's most notable features. This appendix is meant to help users that are familiar with former versions so that they can quickly transition from the older interface to the newer interface.

C.1 Informational Videos

In order to assist you in learning about XFdtd 7, several transition movies are available which demonstrate finding and executing key tasks in XF7 as compared to how they were done in Version 6.

These narrated movies are three to six minutes in length, and offer an overview of several broad topics:

- The Interface
- Modeling
- Components/Ports
- Sensors
- Gridding/Meshing
- Simulation

To access the transition movies, please visit the following URL address:
http://www.remcom.com/xf7/transition

C.2 Importing Files From XFdtd Version 6.X

XFdtd version 6.X files can be imported into XFdtd 7if they were created in Version 6.5 or later. Files created with earlier versions can be loaded in Version 6.5 and resaved. To import such files into XFdtd, navigate to File > Import > XFdtd Version 6 File and select the desired file. Three types of Version 6.X files can be imported into Version 7:

- Version 6.X project files (*.fdtd). Most project parameters will be imported, including Circuit Components, Waveforms, Boundary Conditions and requested Results (see exceptions below). It will also load the associated geometry file.
- Version 6.X geometry files (*.id). Most geometry parameters will be imported, including Materials, Grid definitions and model information (see exceptions below).
- Version 6.X material library files (*.idm). Loading this file will import all Material definitions in the file.

Surface Conductivity Correction and Evaluation Frequency values for imported materials will be saved as parameters and can be edited in the Parameters workspace window.

Some project parameters will not be imported from Version 6.X. They include:

- Hearing Aid Compatibility (HAC) sensor definitions
- User-defined waveforms
- S-Parameters and Total-Field Scattered-Field (TFSF) PlaneWaves
- Static solver
- Area source specifications
- Temperature rise requests
- Steady-state Far Zone requests

Some geometry parameters will not be imported from Version 6.X. They include:

- Mesh edits
- Mesh objects with adaptive grid regions
- A helix with zero width
- Touching object tests

Appendix D

Output File Formats

In this appendix, you will learn...

→ The format of XFdtd output files.

→ The format of XFsolver output files.

The appendix covers the following topics:

- Point sensors
- Multi-point sensors
- System .ssout files
- SAR sensor statistics
- Volumetric SAR data
- Far zone output file formats
- Miscellaneous XFsolver information
- UAN files

Starting with XFdtd 7.3, all output files, including `project.log` and `project.diag`, are written to the subdirectory `output` of the Simulation and Run directories. This facilitates easy copying and cleanup of output files. The layout, structure and format of the output files is unchanged. Previously, they were all one level up in the directory structure.

D.1 Point Sensors

Each enabled Point Sensor in a project causes XFsolver to write one binary file of output data. This file is written in the `output` directory where `project.input` exists. The file name is NearZonePoint_[*ModifiedSensorName*]_[*uniqueNumber*].bin

Here, *ModifiedSensorName* is the name of the sensor in the XFdtd project, with all characters other than letters and numbers replaced by '_'. The *uniqueNumber* is an arbitrary integer.

The contents of the file are as follows. Multi-byte integers are all little endian.

- 'Rnzp' (4 x 1-byte char)
- version indicator (1-byte unsigned int)

Only version indicator 0 is currently defined. For version 0, the remainder of the file is as follows:

- X,Y,Z location (3 x 4-byte float)
- bitmask of fields (2-byte unsigned int). This tells which field value types are stored in the file. If the corresponding bit is set, the field is contained. From most significant to least significant, the bits are :

  - (0) Scattered E
  - (1) Total E
  - (2) Scattered H
  - (3) Total H
  - (4) Scattered B
  - (5) Total B
  - (6) J
  - (7-15) Zero (reserved for future use).

- A record for each timestep at which field values were recorded by the Point Sensor. Each record contains the following items, most of which are optional, based on whether or not the coresponding bit in the fields bitmask is set:

  - time in seconds (4-byte float)
  - Scattered E_x, E_y, E_z (3x 4-byte float)
  - Total E_x, E_y, E_z (3x 4-byte float)
  - Scattered H_x, H_y, H_z (3x 4-byte float)
  - Total H_x, H_y, H_z (3x 4-byte float)
  - Scattered B_x, B_y, B_z (3x 4-byte float)
  - Total B_x, B_y, B_z (3x 4-byte float)
  - J_x, J_y, J_z (3x 4-byte float)

D.2 Multi-point Sensors

Surface Sensors and Solid Part Sensors both cause XFsolver to write the same type of output files.

The first file to look at for a Multi-point Sensor is the "info" file. A binary file, it is written in the `output` directory where `project.input` exists. The name of this file is: MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_info.bin.

ModifiedSensorName is the name of the sensor in the XFdtd project, with all characters other than letters and numbers replaced by '_'. The *uniqueNumber* value is an arbitrary integer.

All multi-byte ints in the file are little endian. The file contains:

- 'Rmpt' (4 x 1-byte char)
- Version (1-byte unsigned int)

Only file version 0 currently exists. For version 0, the following additional information appears in the info file:

- Fields bitmask (4-byte unsigned int). This determines which field types had values written to disk by the sensor. If the corresponding bit is set, the field values should be on disk. From most-significant to least-significant, the bits are:

  (0) Time Domain Scattered E

  (1) Time Domain Total E

  (2) Time Domain Scattered H

  (3) Time Domain Total H

  (4) Time Domain Scattered B

  (5) Time Domain Total B

  (6) Time Domain J

  (7) Discrete Frequency Total E

  (8) Discrete Frequency Total H

  (9) Discrete Frequency J

  (10) Discrete Frequency Total B

  (11) Dissipated Power Density

  (12 - 31) Zero, reserved for future use

- Number of points (4-byte unsigned int). This indicates the number of grid indices at which field values were recorded (the grid indices themselves are given in a separate geometry file which is described below).

Version 0 creates many additional files, all of which are currently stored in the same directory as the "info" file. For version 0, information about the geometry of the sensor is stored in a binary file, which is named

- `MultiPoint_[`*ModifiedSensorName*`]_[`*uniqueNumber*`]_geom.bin`

The geometry file consists of a list of locations where the sensor has collected data and written it to file. It contains:

- For each grid index where field values are written out, a record is written to the file. The (i)th record contains:

  - X-index in the calculation grid for the (i)th vertex (4-byte unsigned int)

  - Y-index in the calculation grid for the (i)th vertex (4-byte unsigned int)

  - Z-index in the calculation grid for the (i)th vertex (4-byte unsigned int)

Note that the ordering of the grid indices as they are found in this and the other result files is not in general lexicographic, and is not guaranteed to follow any predictable pattern.

If time domain data has been requested for the sensor, a timestep file is written. The timestep file lists every timestep for which time-domain field data has been reported. The name of the file is

MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_timesteps.bin

For each timestep where field data is reported, the timestep file contains the time in seconds, as a 4-byte float.

Time domain field data is stored in many files, such that different field types, different timesteps, and different components of the field vector are in different files. For instance:

MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*] \
_transient_Ex_scattered_[*timestepIndex*].bin

Note: *timestepIndex* is not the number of the timestep computed by XFsolver. It corresponds to a zero-based index into the time values reported in the "timestep" file described above.

The full set of possible files at any reported timestep depends on which bits are set in the fields bitmask:

- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Ex_scattered_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Ey_scattered_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Ez_scattered_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Ex_total_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Ey_total_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Ez_total_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Hx_scattered_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Hy_scattered_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Hz_scattered_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Hx_total_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Hy_total_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Hz_total_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Bx_scattered_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_By_scattered_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Bz_scattered_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Bx_total_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_By_total_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Bz_total_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Jx_[*timestepIndex*].bin

- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Jy_[*timestepIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_transient_Jz_[*timestepIndex*].bin

Each time-domain-field data file contains one field value for each grid index listed in the "geometry" file described above:

- Field value (4-byte float)

If single-frequency (steady-state) data has been requested for the sensor, a "frequencies" file is written. This file lists every frequency that steady-state field data has been reported for. The name of the file is:

MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_frequencies.bin

The contents of the file are as follows:

- For each steady-state frequency:
  - Frequency in Hz (4-byte float)

Single-frequency field data is stored in many files as different field types, different frequencies, different components of the field vector, and the real and imaginary parts of complex values are in different files. For instance:

MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*] \
_steady_Exr_total_[*frequencyIndex*].bin

gives the real part of the X-component of the total E-field. Dissipated power density result files are a special case – the result is not complex, and is split among components centered on the Ex, Ey, Ez, Hx, Hy, and Hz grid edges sharing the same index.

The *frequencyIndex* value is the zero-based index into the frequency values reported in the "frequencies" file described above.

The full set of possible files at any reported frequency depends on which bits are set in the fields bitmask:

Note: "`r`" is the real part of the complex field value. "`i`" is the imaginary part of the complex field value.

- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Exr_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Exi_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Eyr_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Eyi_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Ezr_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Ezi_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Hxr_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Hxi_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Hyr_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Hyi_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Hzr_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Hzi_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Bxr_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Bxi_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Byr_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Byi_total_[*frequencyIndex*].bin

- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Bzr_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Bzi_total_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Jxr_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Jxi_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Jyr_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Jyi_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Jzr_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_Jzi_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_PddEx_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_PddEy_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_PddEz_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_PddHx_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_PddHy_[*frequencyIndex*].bin
- MultiPoint_[*ModifiedSensorName*]_[*uniqueNumber*]_steady_PddHz_[*frequencyIndex*].bin

Each single-frequency field data file contains one field value (4-byte float) for each grid index listed in the ”geometry” file described above.

D.3 system.ssout Files

Each steady state output frequency directory has a `system.ssout` file which contains a number of useful pieces of steady-state data for that frequency.

Note: There is some possibility that this file may be phased out in the future in favor of a different file format or multiple files.

The `system.ssout` file contains ASCII data with four main sections:

- Power Data
- Efficiency Data
- Port Data
- S-Parameter Data

D.3.1 Power Data

Power data in the file is contained in a block beginning with the following line:

```
begin_<ComputedPower>
```

It ends with the line:

```
end_<ComputedPower>
```

In between, each line reports a power value in Watts, in the followng format: [*Name*] [*Value*]. The *Value* field provides the numeric value in Watts.

- AvailablePower (The net available power)

- NetInputPower (The net input power)
- FeedLoss (The total loss from all inactive ports)
- DissipatedPower (The amount of power dissipated by lossy materials in the simulation space)
- DissipatedPowerInTissue (The amount of power dissapated due to losses in tissue materials)
- DissipatedPowerInNonTissue (The amount of power dissipated due to losses in materials other than tissue materials)
- RadiatedPower (The amount of power radiated from the simulation space)

D.3.2 Efficiency Data

Efficiency data is contained in a block beginning with the following line:

`begin_<Efficiency>`

It ends with the line:

`end_<Efficiency>`

In between, each line reports an efficiency percentage in the followng format: [*Name*] [*Value*]. The *Value* field contains either the radiation efficiency or the system efficiency.

- RadiationEfficiency (The radiation efficiency)
- SystemEfficiency (The system efficiency)

D.3.3 Port Data

For each port in the simulation, there is a block that begins with the following line:

`begin_<Feed_N>`

It ends with the line:

`end_<Feed_N>`

Here, *N* is the zero-based index of the port. This is one fewer than the port index reported by the XFdtd GUI, and also one fewer than the index numbers in the S-Parameter values reported below.

The first line within the block gives the name of the circuit component, as

`Name [The Component Name]`

Each subsequent line within the block contains a name-value pair in the following format: [*Name*] [*Value*]. The *Value* field contains a complex number in the appropriate SI unit for the value. Complex numbers are reported in two possible formats:

- RIFormat: [*RealPart*] j[*ImaginaryPart*]
- RIMFormat: [*RealPart*] j[*ImaginaryPart*] = [*Magnitude*]

The following name-value pairs exist:

- AvailablePower (RIMFormat analytic available power in Watts)
- MeasuredPower (RIMFormat computed input power in Watts)
- Voltage (RIMFormat voltage in Volts)
- Current (RIMFormat current in Amperes)
- Impedance (RIFormat impedance in Ohms)
- CharacteristicImpedance (RIFormat analytic characteristic impedance value in Ohms)

D.3.4 S-Parameter Data

S-Parameter data is reported in a block that begins with the following line:

`begin_<SParameters>`

It ends with the line:

`end_<SParameters>`

Within the block are a number of name-value pairs, in the following format: [*Name*] [*Value*]. Each pair gives an S-Parameter value in the RIMFormat described above.

Each *Name* is in the following format: S[*PortNumber*][*ActivePortNumber*]. Thus, if the active port is port #1, and there are two ports total, you would have values for ”S11” and ”S21.”

D.4 Transient Voltage and Current Data

For each circuit component designated as a port in the simulation, XFsolver writes out a file with time domain voltage and current values. This file is written in the `output` directory where `project.input` exists. The file name depends on the name of the circuit component in the XFdtd project. For example: `CircuitComponent_[`*name*`].vc.`

The file has the following format:

- '!remcomfdtd' (11 x 1-byte char)
- 'L' (indicates little endian multi-byte integer formats follow)
- 3, as a 2-byte unsigned int
- Version number, as a 2-byte unsigned int

Only version number 0 is presently defined. For version 0, the remainder of the file contains 12-byte records giving the time, voltage, and current at each timestep. Each record is:

- Time in seconds, 4-byte float
- Voltage in Volts, 4-byte float
- Current in Amperes, 4-byte float

D.5 SAR Sensor Statistics Files

A Raw SAR statistics file SAR_Raw_Sensor.stats is written out for each steady state frequency for each SAR Sensor in the project. XFsolver places the file(s) in the appropriate steady-state frequency directory.

The file contains the following data:

- '!remcomfdtd' (11 x 1-byte char)
- 'L' (indicates little endian multi-byte integer formats follow)
- 9, as a 2-byte unsigned int
- Version number, as a 2-byte unsigned int

Only version number 0 is defined. For version 0, the remainder of the file is:

- Max SAR value, W/kg (4-byte float)
- X, Y, Z zero-based cell index of the cell with the maximum SAR value (3x 4-byte unsigned int)
- Average SAR value, W/kg (4-byte float)
- Number of voxels (4-byte unsigned int)
- Total mass in grams (4-byte float)

A SAR statistics file is written out for each steady-state frequency for each Averaged SAR Sensor in the project. XFsolver places the file(s) in the appropriate steady-state frequency directory. For 1-gram averaging, the Averaged SAR Sensor statistics file is named:

SAR_Averaging_[*ModifedSensorName*]_[*uniqueNumber*].1g.stats

For 10-gram averaging, the Averaged SAR Sensor statistics file is named:

SAR_Averaging_[*ModifedSensorName*]_[*uniqueNumber*].10g.stats

Its contents are:

- '!remcomfdtd' (11 x 1-byte char)
- 'L' (indicates little endian multi-byte integer formats follow)
- 10, as a 2-byte unsigned int
- Version number, as a 2-byte unsigned int

Only version number 0 is defined. For version 0, the remainder of the file is:

- Max SAR value, W/kg (4-byte float)
- X, Y, Z zero-based cell index of the cell with the maximum SAR value (3x 4-byte unsigned int)
- Total mass in grams (4-byte float)
- Mass of the max constructed cube, grams (4-byte float)
- Lower corner X, Y, Z zero-based cell index (3 x 4-byte unsigned int)
- Upper corner X, Y, Z zero-based cell index (3 x 4-byte unsigned int)

D.6 Volumetric SAR Data Files

XFsolver writes multiple output files of volumetric SAR data per steady-state frequency. Each file gives the SAR value (raw or averaged) for any Yee cell where SAR was computed in a single planar slice of the mesh. This file is written to the appropriate steady-sate frequency directory. Each file is named as follows:

For raw SAR data:
`SAR_Raw_Sensor.[`*SliceDirection*`][`*SliceIndex*`].sar.bin`

For 1-gram averaged SAR:
`SAR_Averaging_[`*ModifiedSensorName*`]_[`*uniqueNumber*`].` \
`[`*SliceDirection*`][`*SliceIndex*`].1gsar.bin`

For 10-gram averaged SAR:
`SAR_Averaging_[`*ModifiedSensorName*`]_[`*uniqueNumber*`].` \
`[`*SliceDirection*`][`*SliceIndex*`].10gsar.bin`

Here, the *SliceDirection* is either 'xy', 'yz', or 'xz', indicating the planar direction of this slice of the mesh. The *SliceIndex* indicates the magnetic calculation grid index of that planar slice. For instance SAR_Raw_Sensor.xy_5.sar.bin contains raw SAR data for the Yee cells with magnetic calculation grid index k = 5.

ModifiedSensorName is the name of the sensor in the XFdtd project, with all characters other than letters and numbers replaced by '_'. For SAR Averaging sensors, this is typically 'SAR_Averaging_Sensor', although a different name may have been assigned during SAR post-processing. The *uniqueNumber* is an arbitrary integer.

The contents of any planar slice SAR data file (raw or averaged) are as follows:

- '!remcomfdtd' (11 x 1-byte char)
- 'L' (indicates little endian multi-byte integer formats follow)
- 13, as a 2-byte unsigned int
- Version number, as a 2-byte unsigned int

Only version number 0 is defined. With version 0, all SAR values reported in the file must be contained in a single planar slice of Yee cells (either grid index i, j, or k is constant for the file). The remainder of the file data is as follows:

- Plane normal direction (1-byte unsigned integer): A value of 0 means that this is a plane of constant x (grid index i is fixed). A value of 1 is used for a plane of constant y (grid index j is fixed). A value of 2 indicates a plane a constant z (grid index k is fixed). This value is similar to the 'xy', 'yz', or 'xz' *SliceDirection* used in the output file naming convention given above.
- Plane normal grid index (4-byte unsigned integer): This value tells where the planar cut exists in the mesh, defined in the same manner as the *SliceIndex* of the file naming convention given above.
- Number *N* of SAR value records reported in this file (4-byte unsigned integer).
- *N* SAR value records, specifying a location and computed SAR value. Each record is 12 bytes, comprising the following:

  - X-index in the magnetic calculation grid for the measurement point (4-byte unsigned integer). This value is not present if the plane normal direction is 0 (in which case the X-index is the plane normal grid index).
  - Y-index in the magnetic calculation grid for the measurement point (4-byte unsigned integer). This value is not present if the plane normal direction is 1 (in which case the Y-index is the plane normal grid index).
  - Z-index in the magnetic calculation grid for the measurement point (4-byte unsigned integer). This value is not present if the plane normal direction is 2 (in which case the Z-index is the plane normal grid index).
  - SAR value in W/kg (4-byte float)

Note that the SAR records are not in any predefined order. Also, some Yee cells within the plane have no corresponding value for SAR (e.g. cells in freespace, or cells which touch a perfect conductor edge), and as such, are omitted.

D.7 Far Zone Files

When a far-zone sensor with steady-state data collection enabled is placed in a project, it will generate a data file for each steady-state frequency used in the computation. Each data file is written in the appropriate steady-state frequency directory (e.g. SteadyStateOutput/f0).

The name of a single-frequency far zone output file is

FarZone_steady_[*ModifiedSensorName*]_[*uniqueNumber*].fz

where

[ModifiedSensorName] is the name of the sensor in the XFdtd project, with all characters other than letters and numbers replaced by '_', and *[uniqueNumber]* is an arbitrary integer.

The file is binary and contains the following data:

- '!remcomfdtd' (11 x 1-byte char)
- 'L' (indicates little endian multi-byte integer formats follow)
- 1, as a 2-byte unsigned int
- Version number, as a 2-byte unsigned int

Only version number 0 is presently defined. For version 0, the remainder of the file contains *N* 24-byte records, where *N* is the number of requested far zone angles. Each record contains:

- Theta in radians (4-byte float)
- Phi in radians (4-byte float)
- Real part of Etheta (4-byte float)
- Imaginary part of Etheta (4-byte float)
- Real part of Ephi (4-byte float)

- Imaginary part of Ephi (4-byte float)

Note: It is possible to predict the order in which far-zone angles will appear in the output file. If the far-zone sensor is defined in the theta/phi coordinate system, angles are represented in the following order:

- theta0 phi0
- theta0 phi1
- theta0 phi2
  ...
- theta1 phi0
- theta1 phi1, etc.

Here, *theta0* and *phi0* are the Theta/phi start angles defined for the sensor. The theta-increment and phi-increment for the sensor determine the differences in angle between successive theta or phi values.

For sensors defined in Alpha/Epsilon or Elevation/Azimuth coordinate systems, angles appear in similar order, either

- alpha0 epsilon0
- alpha0 epsilon1
- alpha0 epsilon2
  ...
- alpha1 epsilon0
- alpha1 epsilon1, etc.

or

- elevation0 azimuth0
- elevation0 azimuth1
- elevation0 azimuth2
  ...
- elevation1 azimuth0
- elevation1 azimuth1, etc.

Broadband Far Zone Sensor Output Files

A far zone sensor with broadband data collection enabled will generate a single data file in the `output` directory where `project.input` exists.

The name of the broadband far zone output file is:

`FarZone_transient_[`*ModifiedSensorName*`]_[`*uniqueNumber*`].fz` where

[ModifiedSensorName] is the name of the sensor in the XFdtd project, with all characters other than letters and numbers replaced by '_', and *[uniqueNumber]* is an arbitrary integer.

This text file contains one line for each timestep in the simulation. Each line of text gives space-delimited values for Ephi and Etheta for each requested angle (e.g. Ephi-angle-0, Etheta-angle-0, Ephi-angle-1, Etheta-angle-1, Ephi-angle-2, Etheta-angle-2).

In order to make sense of this data, users need to know which far-zone angle corresponds to each set of Ephi/Etheta values. See the Steady State Far Zone Sensor Output File explanation (above) for a description of how angles are ordered in the output files.

D.8 Geometry.input File Format

The `geometry.input` file is an ASCII file that contains grid size and deltas and material information. It uses the following template:

`Line 1: Geometry file version 3.20 (indicates file format)`

`Line 2: (always empty)`

`Line 3: mesh_defined (ignore)`

`Line 4: 0 0 0 (ignore)`

`Line 5: 0 0 0 (ignore)`

`Line 6: 0 0 0 (ignore)`

`Line 7: 0 0 0 (ignore)`

`Line 8: 0 0 0 (ignore)`

`Line 9: 0 (ignore)`

`Line 10: begin_<materials> (the list of materials are contained within this delimited section)`

`...`

`Line ??: end_<materials>`

`Line ??: begin_<GridDefinition>`

`Line ??: GridOriginInMeters # # # (indicates the spatical location of the lower bounding box of the grid)`

`Line ??: NumberOfCellsInX # (indicates the number of grid lines in the X direction)`

`Line ??: NumberOfCellsInY # (indicates the number of grid lines in the Y direction)`

`Line ??: NumberOfCellsInZ # (indicates the number of grid lines in the Z direction)`

Line ??: end_<GridDefinition>

Line ??: begin_<Deltas>

Line ??: begin_<DelX>

Line ??: 0 #Del (indicates the length of the electric cell edge at that index and all following indices until another index is provided. always starts at 0)

...

Line ??: end_<DelX>

Line ??: begin_<DelY>

Line ??: 0 #Del

...

Line ??: end_<DelY>

Line ??: begin_<DelZ>

Line ??: 0 #Del

...

Line ??: end_<DelZ>

Line ??: end_<Deltas>

begin_<AveragedMaterialBins> (always empty)

end_<AveragedMaterialBins>

D.9 Mesh.input File Format

The `mesh.input` file is a binary file that contains mesh information.

It starts with a 16-byte file header to specify its file type (similar to the Transient Voltage and Current Data section in this appendix):

- '!remcomfdtd' (11 x 1-byte char)
- 'L' (indicates little endian multi-byte integer formats follow)
- '0' (as a 2-byte unsigned int to indicate the mesh.input file)
- '#' (as a 2-byte unsigned int to indicate the version number of the mesh file. Always 0 for now)

Next, the file provides a mesh file header containing information on how many mesh data points to read:

- '0' (as a 1-byte unsigned int to indicate the mesh data is in BinaryEndPoints format. Only 0 is supported now)
- '#Ex' (as a 4-byte unsigned int to indicate the number of electric x-directed mesh points to read)
- '#Ey'

- '#Ez'
- '#Mx' (as a 4-byte unsigned int to indicate the number of magnetic x-directed mesh points to read)
- '#My'
- '#Mz'
- '#AveEBins' (as a 4-byte unsigned int to indicate the number of electric averaged material bins to read)
- '#AveMBins' (as a 4-byte unsigned int to indicate the number of magnetic averaged material bins to read)
- '#AveEEdges' (as an 8-byte unsigned int to indicate the number of electric averaged material edges to read)
- '#AveMEdges' (as an 8-byte unsigned int to indicate the number of magnetic averaged material edges to read)

Mesh data appears next. Read it in the same order as in the mesh header. For example, first read all **#Ex** points, then read all **#Ey** points, and so on. **#M*** will be zero unless magnetic materials are used in the mesh. **#Ave*** will be zero unless a voxel object exists in the mesh and averaged material meshing is enabled. The **#Ave*** format is not documented in this appendix.

Each **#E*** and **#M*** point consists of 17 bytes. For an **#Ex** or **#Mx** point,

- `'#xStart'` (as a 4-byte unsigned int to indicate the x cell edge start index)
- `'#y'` (as a 4-byte unsigned int to indicate the y cell index)
- `'#z'` (as a 4-byte unsigned int to indicate the z cell index)
- `'#xStop'` (as a 4-byte unsigned int to indicate the non-inclusive x cell edge stop index)
- `'#material'` (as a 1-byte unsigned int to indicate the material number)

For an **#Ey** or **#My** point,

- `'#x'` (as a 4-byte unsigned int to indicate the x cell index)
- `'#yStart'` (as a 4-byte unsigned int to indicate the y cell edge start index)
- `'#z'` (as a 4-byte unsigned int to indicate the z cell index)
- `'#yStop'` (as a 4-byte unsigned int to indicate the non-inclusive y cell edge stop index)
- `'#material'` (as a 1-byte unsigned int to indicate the material number)

For an **#Ez** or **#Mz** point,

- `'#x'` (as a 4-byte unsigned int to indicate the x cell index)
- `'#y'` (as a 4-byte unsigned int to indicate the y cell index)
- `'#zStart'` (as a 4-byte unsigned int to indicate the z cell edge start index)
- `'#zStop'` (as a 4-byte unsigned int to indicate the non-inclusive z cell edge stop index)
- `'#material'` (as a 1-byte unsigned int to indicate the material number)

Here the material integer maps to the list of materials read from the geometry.input file.

Sample code for reading Ex edges on the mesh

```
for( p = 0; p < #Ex; ++p )
{
    read( xStart );
    read( y );
    read( z );
    read( xStop );
    read( material );
    for( i = xStart; i < xStop; ++i )
    {
        xElectricMesh[x][y][z] = material;
    }
}
```

The following are all read from geometry.input.

- numberOfCellsInY
- numberOfCellsInZ
- list of materials

D.10 Miscellaneous XFsolver Information

Introduction: This section contains miscellaneous XFsolver information. Remcom will add more content as it becomes available.

How to retrieve the steady state frequency values from XFsolver:
To retrieve steady state frequency values from XFsolver, follow the procedure below.

1. Open the `project.input` file
2. Search for the following line of text:

   ```
   begin_<SteadyStateFrequencies>
   ```

This line begins a block of text, wherein each line in the block lists both a steady state frequency and the name of the directory containing many of the output files for that frequency.

Each line of text has the following format:

   `FrequencyFolder` [*frequencyHz*] [*directoryName*]

(e.g., `FrequencyFolder 1.5e+10 SteadyStateOutput/f0/`).

The block of steady state frequency lines ends with:

```
end_<SteadyStateFrequencies>
```

D.11 UAN File

The user-defined antenna format is primarily used to share antenna pattern information between Remcom products (XFdtd, Wireless Insite®, XGtd®). However, since it is an open, text-based format, users of these products are free to read amd create files in this format or use it for their own applications.

The full (`.uan`) format specification is described below. Note that XFdtd only writes `complex` output in `mag_phase` format. XFdtd outputs an additional output keyword of `NetInputPower` followed by the Net Input Power in Watts for the simulation from which this pattern was obtained. This parameter is currently ignored by Wireless Insite and XGtd.

A `.uan` file starts with a parameter section:

```
\begin_<parameters>
    format free
    phi_min 0
    phi_max 360
    phi_inc 4
    theta_min 0
    theta_max 180
    theta_inc 3
    [complex, real]
    [mag_phase, real_imag]
    pattern [gain, power, field]
    magnitude [dB, linear]
    maximum_gain 0
    phase [degrees, radians]
    direction [degrees, radians]
    polarization [theta_phi, phi_theta, theta, phi]
\end_<parameters>
```

where brackets ([ ]) indicate that one of the set of words within the brackets should be chosen. Note that the `[mag_phase, real_imag]` keyword is required only when complex is being used and that the `maximum_gain` keyword is optional and is measured in dBi. Also note that the `*_min, *_max and *_inc` values must be in integer format.

Following the parameter section, the data section follows which provides angle-based pattern data, one row per angle.

For example, the first four lines of `complex` data specifiation might be:

```
90.000  0.000  -10.0  -25.0  90.0  172.50
90.000  4.000  -11.5  -24.0  92.0  174.50
90.000  8.000  -12.0  -24.0  94.0  176.50
90.000  12.00  -13.5  -20.0  97.0  178.50
```

where the data format is theta, phi, gain (theta component), gain (phi component), phase (theta component), phase (phi component).

If phase is not given, the keyword `complex` should either be omitted from the parameter section, or replaced by `real` (which is the default). Further, the `mag_phase` or `real_imag` keyword, which specifies the format of the complex values, is not required in the header.

The data in the file would be:

```
90.000  0.000  -10.0  -25.0
90.000  4.000  -11.5  -24.0
90.000  8.000  -12.0  -24.0
90.000  12.00  -13.5  -20.0
```

Appendix E

Application Change Log

This appendix lists the the changes, bug fixes and enhancements (in reverse chronological order) for each release of XFdtd. It is intended to inform the user of newly added features and, if the user has experienced any issues or bugs in previously releases, to aid in determining if those issues or bugs have been addressed.

XFdtd is versioned with four numbers as *major.minor.feature.bugfix*. A change in any number except the *bugfix* number indicates that new features have been added to the software.

7.3.x

7.3.0.x

7.3.0.4

Fixes

- Added missing scripting methods *getCollectSteadyBFieldsVsFreqOfInterest()* and *setCollectSteadyBFieldsVsFreqOfInterest()* on both *SensorDataDefinition* and *SurfaceDataDefinition*.
- Fixed issue where display of scaled results would not update after changing the scaling factor.
- Removed messages about "PETSC ERROR:" on stderr when the UI application aborts unexpectedly.
- Fixed issue where only the first N time samples of results were being used when computing FFT results for viewing where N is determined by the Minimum Plot FFT Size field in the Application Preferences.
- Fixed issue where XFsolver could sometimes not start with the message "There was a problem obtaining system resource usage" (extremely rare).
- Resolved situation where the application could exit unexpectedly if the grid was modified so that the lower corner of a waveguide boundary was outside the grid.

- Fixed issue where steady state results for waveguides and the system sensor could be incorrect for some DFT frequencies when MPI is used.

7.3.0.3

Fixes

- Added support for NVIDIA's Kepler architecture.
- Included additional third-party licenses.
- Resolved issue where solving for waveguide modes would cause the application to exit unexpectedly on Linux.
- Resolved issue on Linux where non-`.sat` or non-`.sab` CAD files could not be imported due to a missing executable in the distribution.
- Resolved issue where geometry picking tools might not be deactived properly when picking waveguide port integration line endpoints.
- Fixed issue where attempting to use the object returned by *View.getCamera()* could cause the application to exit unexectedly.
- Resolved an issue where the meshing process could continue after cancellation and subsequently

  possible cause the application to exit unexpectedly.
- Fixed issue where a simulation could go unstable when a matched termination was present.
- Resolved issue in 7.3.0.2 only that would cause XFsolver to crash when an MPI rank was located entiredly within the PML boundary. The solver now exits cleanly with an error message instead as in 7.2.x and previous versions.
- Fixed issue where some number of MPI ranks could cause bad simulation resuls. This is like to affect only y- or z-directed waveguides or simulations where the number of MPI ranks is more than half the number of cells in the x-direction.
- Fixed issue that could cause slight differences in waveguide port S-Parameters and input power results depending on the number of MPI ranks. This affected only simulations being executed using MPI.
- Resolved issue where XFsolver could crash when running a simulation with averaged materials present. This problem was fixed in 7.2.3.4 as well.
- Fixed issue where the Degradation Level meshing parameter for voxel objects was not restored properly when reloading a project. This problem was fixed in 7.2.3.4 as well.

7.3.0.2

Important Notes

All output files, including log files, from XFsolver are now placed in a single folder named `output` in the simulation and run folders to facilitate being able to easily copy or remove simulation results. The names and structure of the files remain the same. If you have outside processes that look for files from XFsolver, they will need to be modified accordingly.

Legacy XStream® is no longer supported. If you encounter difficulties because of this, please contact Remcom tech support.

Licenses issued for previous releases of XFdtd® will not work for this release. Users with current support contracts should automatically receive new licenses; otherwise contact Remcom tech support to update or obtain your new license file.

The Linux platforms for which XFdtd® is delivered have been changed due to changes in usage as well as technical issues with building for older platforms. Dropped platforms are SLES9 64-bit, RHEL4 64-bit and RHEL5 32-bit. Added platforms are SLES10 64-bit. The RHEL5 64-bit builds should work for all Linux platforms that are no older than RHEL5, including Mandriva, Ubuntu, Fedora, CentOS and RHEL. If you encounter difficulties because of this, please contact Remcom tech support.

New Features

- Waveguide ports are now available in addition to lumped components. See Section 7.4 to get started with Waveguide ports.
- The creation of simulations can now be cancelled.
- Dissipated Power Density is now an available result quantity for surface and solid sensors.
- Added ability to scale steady-state results across all Runs in a Simulation. This is achieved by choosing “Entire Simulation” from the “Apply Edits to Which Scope” dropdown in the steady-state results editors.
- Magnetic Ferrite and Thin Wire materials are now supported for XStream® simulations.
- Periodic Boundary Conditions are now supported for XStream® simulations.
- Added new result type of Active VSWR. See Section 10.5 for more information.
- The static electric field solver has been updated and optimized, including the ability to calculate and assign static potentials to uninitialized conductors. This new setting can be selected when creating an XFdtd simulation in the Create Simulation dialog on the Static Solver tab.
- Licensing has been simplified so that only the license location need be specified. XFdtd® will check out a Pro XFdtd® license, and then automatically attempt to check out a BioPro if a Pro license is unavailable or when it is needed. The end result is less confusion about which license to choose in the License dialog if both Pro and BioPro licenses are available. Additionally, there is no longer a need to install a license server except for the case of floating licenses.
- Screen capture capability has been enhanced by 1) Providing the global hotkey *Ctrl+Alt+F12* to open the existing export-to-file dialog for the Geometry View and Graph Windows; 2) Adding the global hotkey *Ctrl+Alt+F10* and View menu items to copy a screen shot to the clipboard; and 3) Adding the global hotkey *Ctrl+Alt+F11* to save a screenshot to an automatically-named file in the folder specified in the “Screen Capture” section of the Application Preferences. See Sections 4.1.2, 5.2 and 12.3.2.
- Voxel models for the Visible Male and Visible Female are now provided as part of the installation. They can be imported into a Project using File→Import→Voxel Object→MMF from the main menu and browsing to the `data/BioModels` directory of the XFdtd® installation folder.

Usability/Performance Improvements

- Simulation memory estimator now better estimates amount of memory required for simulations that use Debye/Drude materials.
- Extra licensing information at the end of XFsolver's `.diag` file was removed as it provided no useful information.
- Tooltips for the pre-defined views have been clarified.
- Dramatically improved performance of loading results for solid volume and SAR sensors.
- Ability to manipulate the camera in the Geometry View by selecting the triad in the lower left has been disabled in order to eliminate a common source of confusion. To re-enable, select "Interact with Triad" on the Modeling tab of the Application Preferences.
- Added "Visible when Enabled" checkbox to the cutting plane menu in order to be able to hide the cut plane even when it is active. Also added the ability to change how the cutplane is rendered under "Cutting Plane" on the Modeling tab of the Application Preferences.
- Added Zoom-to-Selection to augment Zoom-to-Extents. This new button is available by clicking on the down-arrow next to Zoom-to-Extents. These two zoom buttons work the same as the other "view" buttons in that the last-used operation stays on top and using ctrl-click iterates through them. Additionally, using shift-click on either of these will perform the other operation but not cause it to come to the top.
- Meshing now utilizes multiple CPU cores, improving performance by nearly N (where N is the number of available logical cores).
- An appropriate Sensor definition is now automatically created and/or assigned when creating a new Sensor.
- Added tooltips to both the "View Slice with Max Value" and the "View All SAR Status" buttons of the SAR field viewer to describe what they do.
- Added ability to add the Simulation Notes as a column in the Results Browser. Right-click on the header of the lower part of the Results Browser to show it.
- Average Power (Sav) is reported as a complex number in the Field Viewer. Viewing the phase of this number doesn't make sense, so the ability to specify phase as the sequence axis or viewing the phase is disabled in the Field Viewer.
- Viewing circularly polarized far zone results with the sequence axis as phase and viewing the phase is disabled since these quantities do not make sense.
- Renamed the option "Compute Dissipated Power" to "Dissipated Power Algorithm" to more accurately reflect its purpose and remove confusion.
- Static solver now outputs time elapsed and time estimates similar to that shown when FDTD timestepping.
- Added the ability to add frequency lists to Frequencies of Interest in addition to replacing existing ones in the Create Simulation dialog's Frequencies of Interest tab.
- The description of output ordering in the file format for multi-point sensors has been clarified in Appendix D.2.
- Removed redundant text "Locator:" in the name of Locators in the Project Tree.

- Improved performance of selecting faces when creating a sheet body from faces. Now geometrical operations are performed when selecting the “Preview” tab instead.
- ODB++ import no longer requires the step attribute list and now supports rotation and mirror of user-defined symbols.
- Added ability to specify which XStream® devices should be available for simulation and whether or not to fall back to CPU when XStream® is now available in the Advanced Options for the simulation queue. These options map to the `--xstream-use-devices` and `--use-cpu` command line options of XFsolver.

Scripting

- *NewSimulationData.getFieldInterfaces()*, *NewSimulationData.getStaticSolverEnabled()* and *NewSimulationData.getSParametersEnabled()* are now properly documented.
- Added *Assembly.getAncestorsOfPart()* and *Assembly.getFullPartName()*.
- Expanded the documentation for *LinearPattern* to make usage of its various methods more clear.
- Several methods of *TerminationCriteria* are now documented that previously were not. Additionally, fixed typos in method names (old names will still work) and fixed an error where calling *TerminationCriteria.{s,g}SpatialDensity()* would result an error about an invalid value.
- Resolved crash when cloning the a *Discrete1DForwardFftDataSet*.
- Added *Camera*, *CameraList*, *Project.getCameraList()*, *View.setCamera()* and *View.getCamera()* to allow manipulation of the camera in the view through scripting.
- Added the global *System* object to provide system information. Initially, this object provides the method *getOsType()* to determine what operating system is being used.
- Added numerous methods to the global *View* object to allow the visibility and opacity of objects view, analogous to the visibility toggle buttons in the user interface. Also added methods to capture images of the view to file. See the online reference of the *View* object for a full listing.
- Added *FOIParameters.getSavedFrequencyListNames()* and *FOIParameters.getSavedFrequencySet()* to access the frequency of interests sets that have been saved through the user interface (on the Frequencies of Interest tab in the Create Simulation dialog).

Fixes

- The message from XFsolver when “flatline” is detected is now more descriptive and consistent with other simulation termination messages, which should make it more clear how the simulation was terminated.
- Resolved issue where the convergence computation for sinusoidally-excited simulations could be slightly different between CPU and XStream® simulations.
- Resolved issue where interopolated point sensors reporting H-Fields could report results that were slightly shifted from the actual requested location.
- Resolved issue where the application could crash when modifying the direction of a planar sensor if the specified normal’s length was extremely small.

- Fixed problem where the Maximum Simulation Time in timesteps wasn't always obeyed when XACT objects were present. In these cases, the number of timesteps may be been reduced so that the simulation did not run as long as the specified real time.
- Increased the precision of floating point numbers written to `project.input`. One manifestation of the truncated precision was a change in phase of far zone results between result computed by XFsolver and those computed synchronously in the user interface.
- Resolved issue where an "Unexpected Condition" was encountered when reloading a project that used a boolean parameter type with a value of `true` or `false`.
- Fixed issue where whether or not ports/non-ports could be outside the simulation space without causing project invalidity. Now, for a project to be valid, all enabled lumped components must be inside the simulation space.
- Fixed crash when attempting to load SAR results where some of the datafiles are corrupt.
- Changed handling of lumped components that are not ports so that their losses are now always included in system loss calculations (Net Component Loss, Gain, etc.). Effectively, "Port" status now make S-parameters computation available and ensures that time domain voltage and current and broadband frequency quantities, like impedance and component loss, are available.
- Fixed incorrect dielectric thicknesses in certain cases for ODB++ import.
- Fixed crash when creating a sinusoidal simulation with a large maximum number of timesteps (1e9).
- Removed choice of export to CATIAv4 on Mac since it is not supported.
- Fixed issue where the Twist and Stretch geometry editors did not change the cursor when the attachment type was toggled, status messages were not displayed and the picking buttons were not rendered as depressed during picking.
- Resolved issue where the validity of a Sampled Passive Load could be incorrect.
- Resolved an issue where an incorrect edge could be highlighted when sketching.
- Fixed problem where the rendering of manual fixed points was not updated while editing the fixed points.
- Fixed incorrect units of the amplitude of an External Excitation. (The unit conversion is the same, so results are unaffected.)
- Resolved problem where it could be difficult to grab an edge or vertex in the sketcher and move it.
- Fixed issue where selecting more than one mesh operation and deleting them would leave at least one mesh operation in place.
- Fixed issue where the title for a dataset in the Field Viewer would not use "Scaled(X)" correctly in all cases. Datasets are now never named this way to make their use consistent throughout. Reference the "Show Scaled Values" checkbox instead.
- Fixed issue where picked endpoints of the Twist operation are interpreted in the coordinate system of the object being twisted, not the coordinate system of the picked object. This would be seen only if the two object had different working coordinate systems.

- Resolved issue where exporting an image from the Geometry View or a Graph Window resulted in an image with what was in the background of the window rather than the window itself when certain themes (such as the Aero theme) were used in Windows Vista and above.
- Resolved issue where the evaluated list of values for parameter sweeps could be edited when they are not actually editable.
- Fixed problem where the description text in the Macro Descriptions window would sometimes be wider than the window, forcing the user to scroll to read the entire description.
- Resolved issue where Locators could not be pasted. Locators are allowed to be pasted into the same list from which they came.
- Resolved issue where a copy and paste of a Part would cause the application to exit unexpectedly on Linux.
- Modified Locator picking so that they are now not attached by default. There are known issues with Locators attached to Parts that are pulled from Libraries or pasted from other instances (matching them may not produce the correct result); detaching them works around those issues.

7.2.x

7.2.3.x

7.2.3.4

Fixes

- Resolved issue where a Component's lead could be meshed in certain circumstances if the lead was connected to a Part meshed with XACT.
- Resolved issue where XFsolver could crash when running a simulation with averaged materials present.
- Fixed issue where the Degradation Level meshing parameter for voxel objects was not restored properly when reloading a project.

7.2.3.3

New Features

- Though this release is technically a bugfix-only release, a change was made so that the simulations' `project.info` files now contain the version of XFdtd that wrote the simulation. The new line is similar to: "`writtenWithVersion 7.2.3.3 (64-bit)`".

Fixes

- Resolved issue where simulations could become unstable when run using XStream® if XACT geometry existed in a cell adjacent to a lumped component.

- Resolved issue where adding a result to the Field Viewer did not make it the currently selected result.
- Fixed issue where a warning of an unexpected condition would appear when switching from a Sampled Passive Load to a Feed in the Circuit Component Definition editor.
- Resolved issue where materials were not assigned during import of non-'af_man' MMF files.
- Fixed issue where ODB++ crop functionality could incorrectly crop surfaces with curved vertices near the edge of the substrate.
- Fixed issue where plots of single frequency data would not update accordingly when the scaling factor for those results was changed.
- Resolved issue where the 'Status' field in the Results Browser would not be refreshed until the window was redrawn.

7.2.3.2

Fixes

- Fixed issue where sketcher constraints would be missing when editing Parts in projects created in previous versions of the application.
- Fixed issue with migration tool where newer projects could be migrated when they really didn't need to be.

7.2.3.1

Important Note on Migrating to Version 7.2.3

Changes to the way XFdtd® handles geometrical features have been made in 7.2.3 to resolve a specific class of modeling problems. In order to do this, certain geometry in projects last saved with 7.2.2 or earlier needs to be prepared for the transition to 7.2.3. Such geometry may be in existing projects, libraries or templates. This migration process is implemented in a special Migration Tool which is automatically launched from XFdtd if needed when loading projects. If XFdtd determines that the project you are loading requires migration, you will be presented with a dialog asking if the migration tool should be launched. If so, the migration tool will have the Source and Destination fields pre-populated with the input path and a suitable output path. In many cases, you will only need to click the "Migrate" button. Once the migration is complete, click "Close" to dismiss the migration tool and resume using XFdtd.

If you do not perform this migration, or if the migration process fails, objects that reference faces, edges and vertices (such as Modeling Features and relative positions and directions) may become invalid or may not update as expected.

The migration tool can be used to update libraries and templates as well as projects and can perform "batch" processing. If you manually start the migration tool (found in the "`migration_tools`" subdirectory of the XFdtd installation directory or in XFdtd's start menu in Windows), you will be able to specify the Source and Destination fields. When using objects from a library or using templates that need to be migrated, a dialog will notify you of the need to manually run the migration tool on that library or template. For detailed information on how to use the migration tool, use your mouse cursor to point at the various input fields to display tooltips.

New Features

- The Parts List now contains a column that indicates whether Parts have Locators or not. A tooltip listing the names of the Locators will appear when hovering over the indicator.
- Two new Modeling Features have been added: Twist and Stretch. See Section A.6 for more information.
- Added the ability to specify RLC values sampled as a function of frequency for Passive Loads. See Section 7.2.1 for more information.
- Added the ability to display user-generated (custom) log information from simulations in the Simulations Window in the "Custom Output" tab. All information in "project.clog" files in a Simulation directory and/or its Run directories will be displayed in this tab.
- SAR Averaging was updated to comply with recent changes to the IEEE 1528.1 standard with regards to more stringent mass accuracy in the averaging volume and the way "unused" voxels are handled.
- Simulations with Debye/Drude materials are now supported on XStream®.
- Plane wave excitation can now be used in conjunction with active sources. Note that results are limited to voltage, current and near zone field data in this configuration.

Usability/Performance Improvements

- Improved performance of writing simulation data up to 4x (depends upon what capabilities are being used).
- Resolved case where the Material context menu item "Voxel Material Assignments" would open a dialog entitled "Tissue Material Assignments", which was inconsistent. The latter text is now used in both places.
- Implemented a change where XFsolver now uses the FFT size defined in the Application Preferences when creating that S-Parameter and `.cti` file output. Additionally, XFsolver also has a new commane-line option, `--fft-size={n}` to override the input file's FFT size setting.
- Plot Point Markers can now show the name of the plots they are attached to. Turn this on in the marker's properties by selecting "Show Plot Name when Point and Attached."
- Added an Application Preference to turn off automatic checking for updated results. Please refer to the tooltip on this item before disabling it. Additionally, removed the "Auto-update Results" checkbox in the Results Browser window to remove confusion.
- XFsolver now always prints "Processing finished. xfsolver exiting." as its last line of output before exiting, under all conditions.
- The "Choose Line Graph..." dialog now displays more significant digits when displaying the actual value for axis values, and a tooltip has been added to show the full precision when there still are not enough digits being displayed.
- XFsolver now prints out what version of of XFdtd® was used to write a simulation as well as the version being used to run the simulation.

- The Simulation Window's Output tab will now concatenate all available `project.log` files in Run subdirectories in the same way that `project.diag` files have been treated for the Diagnostic Output tab. This is useful when simulation runs are executed independently on remote machines.
- The performance of the Output tab of the Simulation Window has been improved. Previously, significant delays could be experienced by the user when the Output tab was updated if a simulation's output log was large.
- When a simulation cannot be created due to material/XACT conflicts, the names of the Parts involved are now reported in the error message.
- The list of Projects loaded in the Results Browser is now saved with a Project. When the Project is reloaded, the user is asked if those Projects should be reloaded into the Results Browser. This is especially useful for the case where results from other projects are being plotted since otherwise those plots would be invalid until the user manually re-adds those projects to the Results Broswer.
- The format of `.uan` files is now documented in Appendix D.11.
- Improved performance when refreshing the Results Browser when graphs with many FFT-based plots exist. Previously, the user interface could become unresponsive for a significant amount of time while the Results Browser refreshed due to the creation of a simulation or from the user clicking on the "Refresh" button in the Results Browser.
- Drastically improved meshing performance when meshing geometry consisting of many individual lumps, such as Parts created as patterns and via Parts created from importing a PCB design.
- Added "Macro Information" menu item to Macros menus to display information about the macros provided in the menu. See Section 4.1.4 for more information.
- Section 7.7.7 on User-Defined Waveforms has been updated to be more clear.

Scripting

- Added *Simulation.getDiagnosticOutput()* and *Simulation.getCustomOutput()*.
- *RampedSinusoidWaveformShape* has been modified to be configured automatically to match the configuration required by the solver. Previously, it was the responsibility of the user to properly configure the ramping time.
- Resolved issue where sometimes running a script would result in the error message "Cannot use defunct object" that could occur when deleting objects from the project between two script executions.

Fixes

- Resolved issue where XFsolver would cause a Qt Warning message to appear stating "QString::arg: Argument missing: Failed to write file '%s' ". This was not a fatal message and occurred only when there was a problem writing an output file.
- Resolved an issue where a hollow, closed PEC object meshed with XACT could have fields propagate into the body if it was touching a PEC boundary.
- Fixed issue where reorienting Assemblies that have Parts relative to objects within the Assembly were not rendered correctly when previewed, but would be placed in the correct location after the orientation change was applied.

- Resolved issue where a face that was part of a sphere in an imported CAD model was misrepresented after import.
- Resolved issue where the application would exit unexpectedly while re-adding blends on geometry that had previously had blends removed.
- Corrected the computation of the characteristic impedance for RL-parallel-C components.
- Fixed issue where the Grid would be valid even though the specification of a Part Grid Region was invalid due to a bad formula, for example. In this case, the Part's Grid Region specification was silently ignored.
- Resolved issue where an invalid entry for the Boundary Extensions of a Part's Grid Region specification would cause a dialog to appear stating that an Unexpected Condition was encountered. After that, the application would not behave well, such as not being able to save a Project.
- Fixed problem where the Mesh would be invalid when XACT objects were near PEC boundaries, even though it should have been valid.
- Resolved issue where Parts with self-intersecting lumps would not mesh properly if XACT was enabled for those Parts.
- The scripting methods *Extrude.setDirection()* and *Extrude.getDirection()* can now be called without receiving an error message.
- Fixed Group Delay equation that caused result to be off by a factor of roughly 40.
- Improved performance of adding fixed points and grid regions. This improvement is particularly noticable when making changes to the grid through scripting.
- Fixed problem where the top-level "Component Loss" field in the State State Component results table was not editable.
- Resolved issue where sorting by date in the Project Load and Save dialogs would sort alphanumerically instead of numerically.
- Resolved issue where clicking the "Enter" or "Open" button to load a project in the Project Load dialog would enter the project directory rather than load it. There are currently two known issues with entering paths: 1) UNC paths on Windows do not appear to work properly. The workaround is to map UNC paths to a drive letter. 2) Typing in a path using "C:\{path...}" does not work properly. The workaround is to use forward slashes instead of backslashes if possible (e.g. "C:/{path...}"
- Fixed issue where resizing the scale bar for a nano-scale project when in the Field Viewer would cause the scale bar to disappear.
- Resolved issue where the application would fail when right-clicking in the Parts List when items in the project tree that were not selectable objects (such as the Project item) were selected.
- Resolved issue where convergence detection could become disabled when using the "Analyze Project Contents" setting for the Termination Criteria of a Simulation. When this happened, the simulation would run to the maximum number of timesteps.
- Due to user feedback, the computation of Rotating B-Fields has been reverted back to the same that was used in 7.2.1 and before. Previously, B+/B- could be visualized in ways that did not have real meaning since they are scalar quantities. Therefore, the following additional changes have been made to reduce the confusion of how B+/B- field results are presented: 1) Vector Magnitude is not

available; 2) Vector display is not available; 3) Phase cannot be used as a Sequence Axis; and 4) Phase cannot be selected in the Complex Part.

- Resolved problem where application would fail when using the measuring tool when a cut plane intersected the location of a Part that had been previously deleted.
- Fixed issue where XACT contours would not be rendered correctly in mesh cutplane views for small objects.
- Resolved issue where editing the Raw SAR Total Power Dissipated would not adjust the scaling factor as it should have.
- Fixed issue where the first feature in a modeling sequence could be deleted using the delete key shortcut.
- Resolved crash caused by double clicking on Model with an empty modeling sequence.
- Fixed crash activing Select Shortcut Target right click menu aciton when both Shortcut Group and child Shortcut were selected.
- Resolved issue where application was failing when changing the dimensions of a parameterized piece of geometry.

7.2.2.x

7.2.2.8

Fixes

- Changes to ensure that projects/templates/libraries saved in 7.2.2.8 are properly migrated when loaded in 7.2.3 and above. Added a tool to migrate projects/templates/libraries from XFdtd® 7.2.2.7 and before to 7.2.3 and above.

7.2.2.7

Fixes

- Resolved issue where saving a valid Project could result in an error message during save that said “Failed to save recipe for *part name*”.
- Resolved issue where selecting geometry could result in the application to exit unexpectedly.

7.2.2.6

This version was not released.

7.2.2.5

Fixes

- Fixed issue where projects saved in 7.2.2.4 would be reported as corrupt when reloaded if they contained sensors that referenced more than one face (in total) of geometry. No data is lost, however there is no way for the user to proceed. This release fixes the cause of the problem as well as allows projects saved in 7.2.2.4 to be reloaded normally.

7.2.2.4

New Features

- Though this release is technically a bugfix-only release, an XFsolver for Open MPI has been added to the Linux distribution. See the Installation Guide for more information.

Fixes

- Fixed issue where Parts extracted from a Boolean operation using Extract Parts would lose their Modeling Sequence when the undo stack was cleared of that operation. If the project was saved in this state, XFdtd® would crash when reloading the project. This fix includes handling reloading such a project so the application will not crash, but the previously-extracted Parts are not recoverable. However, their existence is detected and a placeholder Part named "Unrecoverable Part" is added in their place.
- Updated the saving of sensors for proper migration to 7.2.3 of modeling fixes. Previous 7.2.2 releases addressed these changes for geometry.

7.2.2.3

Fixes

- Resolved issue where simulations with a SAR Averaging Sensor and a PEC boundary condition would fail.
- Fixed issue where running a secure script would result in a parse error.

7.2.2.2

Note: 7.2.2.2 is the initial public release of the 7.2.2.x series

New Features

- Simulations excited by the static solver are now able to be run with XStream®.
- XFdtd's CAD Merge functionality has been extended for ODB++ import as well. Multiple simultaneous ODB++ databases are supported, as well as a "crop-to-substrate" capability. See Section 5.4.4 for more information.
- New options for XFsolver allow more control of over result post-processing. `sar-any`, `sar-request` and `sar-project` were added to the `--post-process` option, and the option `--post-process-off` was added. See Table 11.1 or use `xfsolver --help` for more information.

- Ordinary TDR/TDT and Group Delay are now provided for ports. See Section 12.2.3 for more information.
- XFdtd now supports the import ICRP, NICT and Virtual Population voxel models, with the assignment of tissue-appropriate EM parameters. See Section 5.4.2 for more information.
- Average SAR, mass and dissipated power are now reported for the volume over which raw SAR is computed as well as the volume over which averaged SAR is computed.

Usability/Performance Improvements

- DFT sampling information is now printed to the XFsolver log files.
- Section 10.3 was expanded to clearly document how to visualize time-domain far-zone electric fields.
- Library names are now displayed so that they are easier to read.
- The License Dialog now shows hardware keys in the Host ID section.
- Added materials to the Material Library that is provided with XFdtd for the SAM phantom head and hands. The EM parameters for these materials was obtained from the CTIA OTA Test plan rev. 3.1; the materials are named “CTIA Phantom Head [ND]” and “CTIA Phantom Hand [ND]”.
- Improved performance when loading very large projects.
- Added a Notes tab to the Material editor to view and edit notes for a material.
- When many plots are added to a graph there are now more combinations of colors and linetypes to distinguish them from each other.
- Improved performance when changing the meshing priority of Parts.
- Lock files named `writing.lock` are now created in the Simulation and Run directories while a Simulation is being created indicating if that directory (and its subdirectories) is still being modified.
- Feedback is now provided in the Orientation Tab by an icon on the left-hand side of both the Basic and Advanced modes to indicate whether the WCS is attached to any geometry. Mouse over the icon to get a tooltip describing exactly what part of the WCS is attached.
- Added support for being able to save and subsequently load and run encrypted scripts.

Scripting

- Fixed point settings for *CircuitComponent* are now properly documented.
- The *enabled* attribute of UI elements is now properly documented.

Fixes

- Fixed condition where the S-Parameters reported in the `*.cti` and `*.s*` output files from XFsolver could be incorrect if the components involved in an S-Parameter matrix had different impedances or had reactive impedances. Data displayed and exported from the UI was not affected.
- Resolved issue where SAR results could be incorrect if SAR results were requested for a very large volume. When this happened, the results were obviously incorrect.
- Resolved several issues where reflections from the boundary were higher than expected, were not symmetric, or varied between CPU and XStream® simulations.

- Resolved issue where static field solutions gave different results depending upon number of threads used.
- Fixed an issue where specifying the bounding box for raw SAR computations and not requesting averaged SAR would cause SAR results to be unavailable.
- Resolved issue where simulations using Negative Index Materials (NIM) could go unstable when run on Windows.
- Fixed scripting APIs *LinearPattern::getRepeatCounts()* and *getSeparationDistance()*, which previously reported errors when they were called.
- Fixed bug where descrete frequency results may be missing in the Result Browser when one or more runs in a parameter sweep fail to generate XFsolver input files due to invalidity.
- Resolved issue where the focus point for a Gaussian Beam excitation was not provided to XFsolver.
- Resolved issue where *ExtendedComponentArithmeticDataSet* was not serialized properly such that when it was read back into the application it would be in an invalid state.
- CAD merge now properly positions parts even if the main assembly was moved prior to the merge operation.
- Resolved issue on Windows where pasting from the clipboard would result in an item being pasted that we previously copied to the clipboard instead of the latest item copied to the clipboard. Once this happened, the application would need to be restarted for copy-n-paste to work again.
- Fixed situation where, under certain circumstances, snapping and automatic constraints could interfere with each other causing ”Profile not closed” or ”Profile has self-intersecting edges” errors when sketching that were difficult to find and correct.
- Fixed issue where sometimes interferring Parts were rendered when they shouldn't be during picking operations when using the 'lock-on-Part' capability. Now, during picking, once a Part is hidden it stays hidden until picking is complete.
- Resolved issue where the icon for a Part in the Parts List could be different from the one displayed in the Parts Tree.
- The vector representation of Rotating B-Fields has been modified such that the vectors lie and rotate in the XY plane. The equations implemented in XFdtd also now properly documented in the Reference Manual.
- Fixed problem where manipulating graph markers could sometime generate a pop-up message box stating that “XFdtd has encountered an unexpected condition....Attempting to use flat index access with an index larger than the size.”
- Wire Body tree item no longer expands when double clicked, but instead opens the editor for it.
- Fixed an issue where the maximum SAR values from several simulations could not be plotted together in a single operation.
- The text color of the field value readout of the scale bar in the Geometry View is now always black. Previously, it followed the font color for the scalebar, which could make it unreadable.
- Fixed a problem where the rendering of External Excitations and Far Zone sensor and field data was sized incorrectly when using the nanomater-scale template.

- Resolved issue where if multiple waveforms of different types exist in the project but the first one in the list isn't used, the incorrect convergence algorithm could be selected for a simulation.
- Resolved issue where the context menu for the Scale Bar may be shown when right-clicking in the view even when output fields have been hidden.
- Fixed problem where after undoing the deletion of an Assembly with relatively positioned Parts, not all Parts would be rendered until selected or modified.
- Resolved issue where very small mesh edges might not be rendered.

7.2.1.x

7.2.1.2

Note: 7.2.1.2 is the initial public release of the 7.2.1.x series

New Features

- Geometry creation has been enhanced with the addition of tools to directly create and manipulate the following primitive geometric shapes: Sphere, Cylinder, Cone, Cuboid, Torus, Pyramid, Prism, Frustum and Helix.
- Nanometer-scale geometry can now be properly modelled. Previously, such geometry could be created and simulated but could not be visualized. To create a project for nanometer-scale geometry, select "New Project from Template" from the "File" menu and select the "Nanometer" template.
- A standard materials database is now included with the XFdtd installation. The materials can be found by opening the Libraries window and selecting one or more of the "Materials" libraries in the list in the upper left.
- SAR can now be computed for multiple transmitters by choosing "Combine SAR Results..." from the context menu when two or more SAR results are selected in the Results Browser.
- Independent PDFs can now be specified for E_ϕ and E_θ when computing antenna diversity.
- New options for the Termination Criteria of simulations are now available (and are used by default) to provide better and more automatic convergence control. See the Reference Manual for more information.
- Voxel models can now be scaled. To do so, right-click on the Model in the Project Tree or the Geometry View, select "Modify", "Transform" and "Scale."
- Time-dependent materials are now supported for XStream® simulations.

Usability/Performance Improvements

- Sensors collecting time domain data can now automatically determine an appropriate sampling interval or be set manually globally. See "Use Project Value" in each Sensor Definition and the Time Domain Sampling settings in the Project Properties.
- Manual fixed points are now visualized when selected in the Manual Fixed Point table in the Grid Editor.

- Wire bodies can now easily be Covered, Revolved or Extruded by right-clicking on them in the Project Tree.
- Tweaked the graphical range display in the Field Viewing controls to be more clear when there is only one value available in the range.
- Added button to swap Circuit Component endpoints.
- Simulation Notes are no longer copied automatically from the last Simulation that was created. Use the “Use Last” button on the Notes page of the Create Simulation dialog instead.
- Rotated Far Zone patterns can now be exported directly when viewing them by clicking on the “Export to UAN File” button on the Far Zone Viewer’s Setup tab.
- Made optimizations to unit processing to provide an overall improvement in performance.
- HAC values are now shown in multiple units in their tooltips in addition to those specified by the Project’s Unit Preferences.
- The validity of a Part is no longer dependent upon the validity of the Material assigned to that Part (with the exception of Voxel and Grid Space Volume Parts).
- One can now also align along the global X-, Y- or Z-axis when using the “Align Directions” tool to specify the orientation of a Part by pressing ’x’, ’y’, or ’z’ when choosing the second direction instead of picking another reference with the mouse.
- The features in a Part’s Modeling Sequence and now be edited directly through the context menu of a Part under “Edit Feature.”
- Changed the behavior of the auto-reference value calculation in the Field Viewer so that when “Auto reference value” is checked in the scale bar settings, when one toggles between showing scaled values and unscaled values, the reference value also changes as expected. Previously, this occured only if “Automatic Range” was also selected.
- Feed names are now output to the `system.ssout` file.
- Steady-state frequency results for multiple frequencies can now be scaled simultaneously by selecting the “Apply Edits to All Frequencies” checkbox in a Steady-state result window before editing a value.
- The process of shutting down the application has been improved so that situations where it may take a long time for the application to actually stop running after the window goes away should be resolved.
- A context menu item (Extract Parts) has been added to Boolean modeling features to break the boolean back into its constiuent Parts.
- Materials using the Sampled parameter type can now be used with XACT.
- The Project will now be invalid if the geometry being too close to a Liao boundary.
- XFsolver will now pause and wait when its connection to the license server is lost rather than exit.
- Made optimizations to the results scaling to reduce dependence on file operations.

Scripting

- *Simulation.getCreationDate()* is now available.

- Added methods to *CoordinateSystem* to convert between Global, Local and Reference coordinate systems.
- Added methods to *CircuitComponent* to query and modify a component's fixed point settings.
- Added getter methods to *PartGridParameters* and *GridRegionParameters*.

Fixes

- Fixed bug where simulations using XACT may go unstable or give incorrect results when XStream® is requested but cannot be used (e.g. due to unsupported materials or boundary conditions).
- Fixed crash that could occur in certain situations when writing sensor data for a simulation being executed with multiple threads.
- Fixed a problem where the loss computed for a lumped component that was a resistor in series with an inductor was incorrect. This was causing incorrect results for the Net Component Loss and therefore for efficiency calculations.
- Resolved bug where the reported percent complete during SAR processing was incorrect when more than 1 thread was specified.
- Removed incorrect solver message about multi-threads applying only to XACT update process when multiple threads is specified and XACT is being used.
- SAR postprocessing will now properly handle Simulations with multiple Runs where the Run being postprocessed isn't the first one.
- Far Zone postprocessing now prints an appropriate error message when the required input file is not available rather than failing silently.
- Fixed issue where the mesh was updated twice when viewing a mesh plane.
- Fixed meshing of Circuit Components for several cases where connectivity was broken between a Component and a Part that the Component touched.
- The scripting API now properly takes *Total* intead of *VectorMagnitude* for *RadarCrossSection*.
- *DataSetOperation.isCompatible()*, which gave an error about not being callable when used.
- The *DraftOption* enumeration for *SweepOptions* is now properly documented.
- Fixed bug where the parameters E_s and E_o for non-linear electric parameters were reversed.
- The simulation memory estimate has been modified to fix the case where, in certain situations, the actual amount of memory required to run a simulation could be significantly underestimated.
- Fixed issue where creating a surface sensor from invalid geometry would silently fail when creating the simulation, and no results would then be available for that sensor after simulation. The problem will now be reported at simulation creation time.
- Resolved issue where using *saveCurrentProjectAs()* without having ".xf" at the end of the specified filename resulted in the error "Failed to create new simulation directory" when creating a simulation.
- Resolved case where Parts with the default name would not be named the same as in the Project Tree when listed using "click-through" in the Geometry View (using the middle mouse button in the view).

- Fixed several issues with visual representation of SAR and SAR Averaging sensors.
- Simulation memory estimates no longer go away when only the mesh is invalid.
- Fixed bug where one would need to click twice on the Advanced button to enter Advanced mode when in the Orientation tab if the coordinate system was parameterized.
- The HAC field viewer now provides full controls for how the data is displayed (flat/point/vector with sizes, etc.) just like all other field viewers.
- Fixed an issue where dragging and dropping Parts in the tree would sometimes delete them instead of moving them to another position.
- Fixed problem where post-processing SAR would always compute both 1g and 10g averages if the current project's SAR Averaging Sensor was disabled.
- Fixed incorrect name and tooltip for the ”Evaluation Frequency” of the Sampled Parameters material types. The Evaluation Frequency is always used to determine the properties of the material.

7.2.0.x

7.2.0.5

Usability/Performance Improvements

- Improved performance when using the Pick Tool with very large/complicated geometric models.
- Projects created in XFdtd 7.1 or before that reference faces on booleaned geometry (such as features on a boolean or a surface sensor on a boolean) may be invalid when loaded in XFdtd 7.2 due to changes in the solid modeling kernel used by XFdtd. While XFdtd can detect this situtation, it cannot automatically correct it. A dialog is now shown on project load when this situation is detected. There are two possible solutions for this situation. One is to open the project in the previous version of XFdtd, perform a flatten operation on the booleans in the project, save and then load that into XFdtd 7.2. The other is to respecify the items that have become invalid in XFdtd 7.2 as the dialog suggests.

Fixes

- Fixed problem where Shift-Click on a View Orientation button did not work. The fix also addressed issues where rotating in the view while zoomed in did not behave as expected.
- Fixed situation where a message stating “Cannot pass null widget if rect is set” was shown, followed by a crash after that popup was dismissed.
- Resolved issue that caused a message to appear on stdout under Linux stating “Application asked to unregister timer 0xnnnnnnnn which is not registered in this thread. Fix application.”
- Corrected XFsolver messages about the required CUDA version that stated CUDA 3.1 was required when really CUDA 3.2 is required.
- Resolved XFsolver crash attempting to process steady state fields for a sinusoidal run when only a small number of timesteps were complete.
- Resolved issue where in rare instances, a Gaussian Beam-excited simulation could go unstable.

- Fixed crash after undoing the deletion of a Separate Objects in Mesh operation.
- Tweaked simulation memory estimator to provide better estimates.

7.2.0.4

Note: 7.2.0.4 is the initial public release of the 7.2.0.x series

New Features

- The direction of incidence of External Excitations is now rendered in the Geometry View, which greatly aids in correctly identifying and specifying the correct direction. As part of this change, the properties of an External Excitation are now edited in the Geometry View in the same way as Circuit Components, for example, instead of in their own window.
- The solid modeler has been upgraded to ACIS R21sp2, which is now the default format when saving and exporting geometry. With this upgrade, XFdtd now supports the import of Pro/E Wildfire 5.0 files.
- Several improvements to the XACT algorithms make it more accurate and stable.
- A new view alignment button has been added to the Geometry View to orient the view such that the user is looking straight-on (normal) either to the Top or Bottom of the cutting plane. Since the cutting plane defines the sketching plane when creating geometry, this is particularly useful when sketching.
- XFsolver now uses the same syntax as the user interface for specifying command-line options; therefore a new set of command-line options is now used. Although all previous options are still supported, users are encouraged to begin using the new options. See the Reference Manual or type `xfsolver --help` for a list of command-line options.
- XFsolver has a new capability to track device usage when using XStream®. This new capability allows one to simply specify the number of XStream® devices to use (with the `--xstream-use-number=` option) and whether or not to use devices that are already in-use (with the `--xstream-availability=` option). Along with the ability to specify which devices to make available to XStream® (the `--xstream-use-devices=` option), complete flexibility is provided to choose what devices are used for a simulation. For example, specifying `xfsolver --xstream-use-devices=0,1,3 --xstream-use-number=2 --xstream-availability=free --no-use-cpu --use-xstream` will use 2 XStream® devices that do not already have a simulation running on them and only devices 0, 1 and 3 will be used. If 2 free devices cannot be allocated, the simulation will not fall back to running on CPU.
- When in the Sketcher with the cutting plane visible, previous versions of XFdtd allowed the sketching tools to snap to the edges and vertices created by intersecting existing geometry with the cutting plane. This version now adds a new tool in the Sketcher to copy those edges into the current Sketch.
- Added the command-line option `--fft-size` to XFsolver in order to specify the size of the FFT to use when computing data for its frequency output data (such as S-Parameter output files). This option can be used to increase the frequency resolution of the output data.
- XFsolver now outputs CITI files along with `.s??` files.

- Simulations using averaged materials (volume mesh objects meshed with a Meshing Method of "Averaged Material") are now supported on XStream®.
- Added automatic DFT sampling interval computation based on the frequency content of the active waveforms and the simulation's specified Frequencies of Interest. New projects have the automatic sampling interval enabled by default and older projects retain the sampling interval previously specified.

Usability/Performance Improvements

- Optimized broadband far-zone computations by introducing a sampling interval that can either be set manually or computed automatically. This setting can be found in the Far Zone Sensor editor's Properties tab. When the broadband sampling interval is set to "Use Project Value" and the Project's value is set to "Use Automatic Time Domain Data Sampling Interval" (in the Project Properties), the project's Frequency Range of Interest and active waveforms are examined to compute the sampling interval. A sampling interval greater than one (which the automatic setting will usually do) disables the computation of time domain far-zone results. In order to receive the time domain data, make sure the Broadband Sampling Interval in the Far Zone Sensors' properties is set to "1*timestep". When an automatic sampling interval is used, the solver's diagnostics file will report the value used in the simulation. Old projects will retain their sampling interval of "1*timestep" but new projects will enable the automatic computation of the sampling interval.
- Made several optimizations for CAD import to improve performance.
- Added a line to XFsolver output stating which Run it is on, which is useful when performing parameter sweeps.
- Added the ability to specify how many XStream® devices to use when running simulations from the UI. This is specified in advanced Queue options of the Simulations window. Devices that do not have simulations currently running on them are allocated first. Additionally, the items of the Queue options have been reordered to reduce confusion about which items pertain to which.
- The licensing subsystem no longer uses cached license locations to find licenses for XFdtd, but instead uses only what is provided by the `LM_LICENSE_FILE` and `REMCOMINC_LICENSE_FILE` environment variables and what is provided in the licensing dialog. This change will not affect users whose licenses are properly configured, but will alleviate confusion when licensing is misconfigured but XFdtd still launches.
- Along with the change to licensing above, the "Browse" button in the License Dialog now browses for a '.lic' file instead of a folder path. The specifications of a path to a folder containing one or more '.lic' files and 'port@server' are still also accepted.
- Along with the new feature of External Excitation visualization, Static Voltage Points and External Excitations now also have a Visibility attribute.
- The behavior of the cutplane (when applicable) is now consistent when editing Sensors, Circuit Components, etc. The cutplane no longer ever cuts an object being edited.
- Improved XFsolver performance for simulations with steady-state output by optimizing the on-disk storage format for temporary steady-state field files.
- A simulation can no longer be created in the Create Simulation dialog if Parameter Sweeps are enabled but no sweeps are defined. Previously, one could click to Create a simulation and receive the message "No xfsolver files were written."

- Unchecking the Automatic context menu for the scale bar in the Field Viewer now disables the automatic reference value as well as the automatic data range. This behavior should now match the intent of setting all automatic behavior when that context menu item is chosen.
- The scripting editor now has an option to enable/disable word wrap. This option can be found in the “Edit” menu of the Scripting Editor and defaults to ‘enabled’.
- Added appropriate tooltip text for Assembly Transform undo/redo operations.
- Added more information about HAC thresholds to the tooltips for the M-rating and Threshold labels when viewing HAC data in the Field Viewer.
- The default for storing data on the Data Storage tab under Frequencies of Interest in the Create Simulation dialog has been changed from “On Disk” to “In Memory”.
- The size of bounding boxes is now displayed in the view addition to its corner coordinates.
- When the formula for a constraint in the Sketcher is invalid, the contraint is rendered in red and “<Invalid Formula>” is displayed for the value. Additionally, the formula is now shown in parentheses as well, just like when the formula is valid.
- Added text to Meshing Properties and Set Meshing Priority dialog to indicate that a higher meshing priority number means a higher meshing priority.
- Added complete memory estimate information for XStream® simulations to the tooltip message for the memory estimate shown in the Grid Editor and the Create Simulation dialog.
- Added a button in the Geometry View to flip the cutting plane, which can be found in the dropdown menu of the Cutting Plane button. This is a handy shortcut to inverting the cutting plane by editing it and rotating it 180 degrees.
- Improved the ‘Find’ capability in the Scripting Editor by adding a “Find Again” capability, binding Ctrl+F to open the Find dialog, Ctrl+G to Find Again, and remembering the settings of the Find dialog.
- Added the ability to create a Sheet Body from Faces and have the faces remember their source so that if the source changes, the faces change as well. To completely detach the new Sheet Body from the source (as was the previous behavior), check the “Detach Faces” checkbox on the “Select Faces” tab when creating the Sheet Body from Faces.
- Improved the Frequencies of Interest table in the Create Simulation dialog by displaying the current number of frequencies defined, adding a “Delete All” button, and enabling multiple selection.
- Passive Loads with R=L=C=0 are now treated as PEC shorts, which matches the schematic diagram that is presented in the UI as well as users’ expectations.
- Additional notes have been added to Sensor Definition editors regarding the availability of Poynting Vector, Average Power and Rotating B-Fields.
- Renamed “Transient Far-Zone” to “Broadband Far-Zone” in the user interface to eliminate confusion with the term “transient”.
- With the change to the command-line options for XFsolver noted above, the type of XStream® implementation has been divided into separate executables. Which type of XStream is to use, specified in the advanced XStream® settings in the Simulation Queue, has been expanded. Users desiring to use Acceleware version 9-based XStream® should select “Use Legacy XStream Acceleration

(Ax9)". At the command line, the Ax9 XStream executable can be found in the `ax9` subdirectory of the appropriate platform's `bin` directory. The XFsolver launch script for Linux has been updated to accept the `-ax9` option in addition to the `-ax8` option that has existed for some time.

- The search dialog in the Scripting Window now remembers its previous settings.

Scripting

- Added ability to create Shortcuts and Shortcut Groups through scripting. New API classes: *Shortcut*, *ShortcutGroup* and *ShortcutTree*.
- Added ability to get and set the edges of Blends with *Blend.{get,set}Edges(Array)*.
- Added method *getNumberOfMarkers()* to *XYGraph*, *PolarGraph* and *SmithGraph*.
- Added ability to access the parameters (all parameters as well as swept parameter values) used in a simulation. See *Project.getSimulationIds()*, *Simulation.getParameterNames()* and *Simulation.getParameterValue()*.
- Added the accessors for the project's timestep manipulation attributes. See the following methods on *Project*: *isUsingCustomTimestepMultiplier()*, *setUseCustomTimestepMultiplier()*, *getCustomTimestepMultiplier()*, *setCustomTimestepMultiplier()*, *getXactTimestepMultiplier()* and *setXactTimestepMultiplier()*
- Resolved issue where *LawEdge* was not using the correct units (they were 'mm' instead of 'm').
- The Grid Region sort order in the Parts List was modified to be more useful– it is now "No Grid Regions" followed by "Grid Region on Parent" and then "Grid Region on Part".
- Documentation for some global objects (such as *FarZoneUtils* and *ResultUtils*) was missing.
- Resolved issue where attempting to create an *AutomaticRangeBasedWaveformShape* resulted in an error.
- Added methods to the global *ResultUtils* object to obtain tabular sensor data similar to what is shown in the user interface. See the following methods on *ResultUtils*: *getSystemSensorTable()*, *getHacTable()*, *getSarStatisticsTable()*, *getExternalExcitationTable()* and *getComponentTable()*.
- Added methods to the global *ResultUtils* object to easily convert between the various *ResultQuery* enumeration values and a string. These are useful when presenting these values in dialogs for user input. See the set of methods *StringTo()* and *ToString()* on *ResultUtils*.
- *Simulation* is now derived from *Selectable* which provides access to a Simulations's name, notes and metadata.
- Project and Simulation notes as well as parameter values are now available for all Projects loaded in the Result Browser. Previously, they were available only for the currently loaded Project.
- Added methods to access the list of deleted CAD IDs that is used during CAD merge operations. See *Assembly.getDeletedCadIds()* and *Assembly.clearDeletedCadIds()*.

Fixes

- Resolved crash in specific situation when geometry intersected $\pm$X boundary, planar J data was requested, scattered-field formulation was used and excitation was a plane wave propagating in the -Z direction.

- Fixed bug (introduced 7.1.2) where a switch component always behaved as open when running the simulation using XStream®.
- Fixed bug (introduced in 7.1.2) which *may* cause XFsolver to crash of all of the following are true: XACT is being used; a point, solid or surface sensor is being used to capture time-domain J-Field data; and XStream® is being used.
- Resolved issue where the dissipated power of a good conductor meshed with XACT was too low.
- Fixed issue where XFsolver always opened `project.log` file, even if XFsolver was called with the `-version` option to print the version an exit.
- Fixed several places where a message was printed by all nodes of during an MPI simulation when it should have been printed only by one node.
- Fixed issue where sometimes the CPU memory estimate for XStream® simulations could be less than zero in the `project.info` file.
- The FDTD meshing algorithm now accounts for facets that share only a single endpoint edge. Previously, certain CAD geometry could cause a single mesh line to extend across large gaps in the geometry.
- Resolved issue where the Automatic waveform did not update properly when the Grid was invalid. (This was not a common situation.)
- Resolved issue where accessing graph markers through scripting could cause an exception that blocked all project modifications.
- Fixed problem where Scale Transform Features were sometimes not applied after loading project.
- The Net Available Power dataset no longer has the name “Unknown Result Type”.
- Fixed crash when enabling XACT on a Part while the Grid Editor was open.
- Fixed crash when opening an empty Graph.
- Resolved issue where running simultaneous CAD imports on the same machine could cause the import(s) to fail.
- Fixed issue where importing some CAD files would take a very long time and use an inordinate amount of system RAM or would result in a Project that would take a very long time to save and use a great deal of disk space.
- Some `.sat` CAD files would import into XFdtd with names “Object 1”, “Object 2” ... “Object N” instead of the names they had in the tool from which they were exported. This has been resolved for certain cases.
- Fixed issue where the size of vectors or points could be too small to be seen when rendering field data.
- An object that was cut from the project could be pasted into a Shortcut Group, which created the situation where one could have Shortcuts to objects that weren’t in the project.
- Fixed issue where, when multiple results of different units were loaded into the Field Viewer, the Field Reader tool would report the highlighted value on whichever scale bar was visible, even if it was the wrong scale bar.

- Added a scroll bar to the warning dialog that is shown when simulation creation is complete in order to handle the case where many warnings are generated (e.g. for a large parameter sweep).
- Fixed problems where the wait cursor was always displayed while interacting with dialogs generated from scripting and where that cursor never switched back to the default cursor after running scripts where multiple levels of dialogs were used.
- Clicking the Reset button on the Far Zone Rotations tab no longer moves the display origin to (0,0,0), but only resets the direction (as it should).
- Fixed crash when specifying a Radius Constraint with an undefined parameter.
- Resolved issue where the label of the Sequence axis on the Sequence tab of the Field Viewer was not always correct after changing the setup on the Setup tab until the sequence index was changed.
- Fixed problem where dragging/pasting a Circuit Component attached to a Part from a Library into a project would bring the Part into the project but not the Component.
- Fixed problem where Static Voltage Points couldn't be copied (the Copy menu item was not enabled).
- The Level field of the Broadband waveform type now has a unit of 'dBa', which was previously missing.
- The Fixed Point indicator in the Parts List has been fixed to properly handle Assemblies that have Fixed Point extraction on for their bounding boxes.
- For consistency, renamed occurences of "Fixed Location" throughout the application to "Fixed Point".
- Fixed issue where steady-state field results in the Field Viewer would not be scaled when Phase was chosen on the Setup tab, and where clicking on the "Show Scaled Values" checkbox in the Field Viewer the first time did not cause it to be toggled.
- The Frequencies of Interest tab in the Create Simulation dialog now properly handles the case when an Automatic Waveform is a sinusoid by auto-selecting "Use Waveform Frequency".
- Fixed issue where plotting multiple datasets on XY Graphs and Smith Charts resulted in a different plot order and therefore different colors being assigned to each plot.
- Resolved issue where if Parts Visibility was off, selection in the View was disabled. This resulted in several problems: the Measuring Tool could not be used, one could not right-click on a Part to bring up its context menu, and one could not click on a single Part to select only it when several Parts were currently selected.
- Resolved issue where application responsiveness was degraded when any Orientation tab was open and the cutting plane was enabled.
- Cross-section snapping in the Sketcher now always obeys the state of the cross-section snapping toggle button. Previously, snapping occurred only if the cutting plane was enabled while in the Sketcher.
- Highlights when using a Picker Tool are now cut by the cutting plane when it is enabled. Previously, the geometry highlights were not cut even though the geometry they were highlighting was.
- Fixed crash when hovering the mouse over an edit box for an invalid constraint in the Sketcher.

- Fixed issue where the schematic diagram for a Nonlinear Capacitor with a resistor in parallel was not correct.
- Resolved issue where the Detach buttons for positions and directions were not initialized to the correct state and so were enabled even with the position or direction was not attached to anything.
- Fixed issue where one could perform a multiple selection of Parts in the Project Tree for many modeling and sensor operations that take only one Part. This resulted in unexpected behavior when the operation respected only the first Part that was selected.
- Fixed issue where pasted mesh objects were not visible.
- Resolved issue where definitions could not be pasted onto Circuit Components, Near Field Sensors, or External Excitations.
- When the cutting plane is enabled, it is now always rendered even if there is no geometry in the Project.
- The cutting plane editor now shows an invalidity icon when the cutting plane specification is invalid.
- The Locator editor now shows an invalidity icon when the Locator specification is invalid.
- Fixed issue where which faces were selected would sometimes be forgotten between steps when specifying a Loft.

7.1.x

7.1.2.x

7.1.2.6

Fixes

- Fixed issue where a certain combination of R and L values for a series RL passive load produced incorrect dissipated power results for the component.

7.1.2.5

Usability/Performance Improvements

- Modified algorithm when assigning materials during CAD Import with “Automatically assign materials” enabled but “Refresh and merge existing parts” disabled so that the materials existing Parts with the same name have a higher priority than (and are therefore used instead of) materials of the same color.
- Modified behavior of CAD Import with “Refresh and merge existing parts” enabled so that merged Parts are always placed in the global location defined by the imported Parts rather than only by their location in their own working coordinate systems. The byproduct of this change is that if one moves an imported Part using Specify Orientation and then performs a CAD import and merge for that Part, the Part will move (back) to its original location. If one desires to perform the move operation and have it persist through import and merge operations, the move should be done with the Translate

and/or Rotate tools so that the movement is applied to the Part as a Modelling Sequence Feature instead of by using Specify Orientation.

- Added support for bulge parameter in LWPOLYLINE section for DXF files. Imported designs with straight lines instead of curves in some places may now import as expected.

Fixes

- Fix issue where Default Materials were not added at the end of the Material List when performing CAD Import with “Automatically assign materials” enabled.
- Resolved intermittent crash of the ‘converter’ program at the end of the first phase of CAD Import.

7.1.2.4

Fixes

- Updated XStream® to use CUDA 3.2 runtime to resolve issue where 4GB of RAM was being resported for nVidia C2070 hardware instead of 6GB. This means that users of XStream® will need to have a video driver that is new enough to support CUDA 3.2, which would mean driver version 260 or higher. Please visit nVidia’s website at www.nvidia.com to obtain the latest video driver.

7.1.2.3

Usability/Performance Improvements

- Additional logic to handle multiple Parts with the same name in an Assembly during CAD Merge.

7.1.2.2

Usability/Performance Improvements

- Aligned terminology for Custom HAC Band Settings to that of ANSI C63.19. “Waveform Modulation” has been renamed to “Probe Modulation”, “Electric Probe Modulation” to “Electric Probe Calibration” and “Magnetic Probe Modulation” to “Magnetic Probe Calibration”. Additionally, the value entered for the new “Probe Modulation” is the inverse of that entered for the old “Waveform Modulation”, aligning this value to what is expected to be entered.
- Improved CAD Merge capability: to include matching Part names in the algorithm to assign Materials and other Part settings, and to better detect existing geometry for merge operations and CAD Merge Result reporting.
- Added ability to Import/Merge a CAD Model into a specific existing Assembly by right-clicking on that Assembly and choosing Import.
- Added Alt+M shortcut to open Macros Menu.

Scripting

- Added:

  CustomHacBandSettings::setProbeModulationFactor()

CustomHacBandSettings::setElectricProbeCalibrationFactor()

CustomHacBandSettings::setMagneticProbeCalibrationFactor()

HacBandSettings::getProbeModulationFactor()

HacBandSettings::getElectricProbeCalibrationFactor()

HacBandSettings::getMagneticProbeCalibrationFactor()

Deprecated:

CustomHacBandSettings::setWaveformModulationFactor()

CustomHacBandSettings::setElectricProbeModulationFactor()

CustomHacBandSettings::setMagneticProbeModulationFactor()

HacBandSettings::getWaveformModulationFactor()

HacBandSettings::getElectricProbeModulationFactor()

HacBandSettings::getMagneticProbeModulationFactor()

Please note: Backwards compatibility for the deprecated methods is maintained, but the meaning of the first parameter of the *CustomHacBandSettings* is now the inverse of what is was (see the previous section).

Fixes

- Fixed issue with CAD Merge where the counts reported in the CAD Merge Results shortcuts were not correct sometimes.
- Fixed problem where changing selection could take a very long time if the Parts List was open and many Parts were selected.
- Resolved issue where importing a `.sat` file would result in the Parts being imported in a random order.
- Resolved issue where some warning messages about possibly invalid S-Parameters were still being shown after Simulation creation. These messages are now in the `project.diag` file.
- Fixed issue where DXF models weren't being rendered after import.
- Fixed crash when loading a project that had a cut plane attached to geometry. Also changed the default for the picking tools when editing a cut plane to not attach to geometry.

7.1.2.1

Note: 7.1.2.1 is the initial public release of the 7.1.2.x series

New Features

- The performance of XStream® simulations using XACT has been significant improved. Users may see up to a 15x improvement in speed!

- Added the ability to perform sweeps over multiple parameters within a single simulation. This is an extension of the capability previously provided on the Parameter Sweep tab of the Create Simulation dialog. See the XFdtd Reference Manual for more information.
- Added Frequency Range of Interest for a Project. This range allows XFdtd to automatically formulate an excitation waveform, provide additional project validation and show result plots more tailored to the user's interests. See the XFdtd Reference Manual for more information.
- Added arbitrary cutting planes to the 3-D geometry view. Cutting planes can be activated one at a time, and using the standard orientation tools, cutting planes can be located anywhere in 3-D space. Additionally, when the cutting plane is active, any cross sectioned edges and vertices can be used as snapping points within the sketch (cross section) editor and all location picking tools, such as the measuring tool, and circuit component end-point pickers.
- Added the ability to run Macros without the need to import them into the Project. To use, simply put the `.xmacro` scripts you wish to use in a directory (defined by an Application Preference) and then access them using the "Macros" menu on the main menu bar of the application. See the XFdtd Reference Manual for more information.
- Added the ability to export S-parameters to Touchstone file format. To export to Touchstone, right-click on the S-Parameter results that you wish to export and select "Export→Export to Touchstone file..."
- XFsolver now implements a Time Dependent Material, previously known as Bulk Switch. This material type allows the definition of electric materials with time dependent conductivity and permittivity and magnetic materials with time dependent permeability and conductivity. Though this capability has not yet been exposed in the UI, the solver input files can be amended to enable this capability. Please contact Remcom Technical Support if you would like to use this new capability.
- Raw SAR computations have been updated to the latest standard, IEEE 1528.1 (IEC 62704-1) draft dated 19-Jan-2010.

Usability/Performance Improvements

- Major improvements were made to viewing 3D steady-state fields, resulting in the ability to specify the phase for viewing and animate the data. The selection of what field components and complex part combinations are allowed has been optimized to reduce confusion.
- When mousing over an invalidity indicator, the tooltip describing the invalidity now pops up immediately rather than after the tooltip timeout period.
- Picking tools were changed in version 7.1.0 to not attach to geometry by default (pressing the 'a' key during picking then causes attachment). When placing the endpoints of Circuit Components, however, it is usually desirable to have them attached to the selected geometry. Therefore, the default when picking endpoints for Circuit Components has been changed to attach to the picked geometry. Throughout the application, whether or not whatever is being picked will be attached to geometry is always indicated by the cursor- when not attaching, a simple arrow cursor is shown. When attaching, an arrow with a thumbtack is shown.
- Picking tools have the capability to attach what is being picked to geometry, so that when that geometry moves the attached point moves with it. Examples of this are attaching one Part's working coordinate system (WCS) to another Part or attaching the endpoint of a Circuit Component to a Part. One can see the attachment state of a WCS by going to the Advanced mode of its Orientation tab

or to the Circuit Component editor and noticing the title of the group box containing the information for the picked item. If it is unattached, the title will be something like “Endpoint 1: Fixed Position” or “Endpoint 1: Parameterized Position”. If it is attached, it will be something like “Endpoint 1: “Center of Edge”. If the geometry referenced by the attachment is deleted or modified in such a way that the element that is referenced no longer exists, the attachment is still remembered but is in a “Frozen” state- it will simply stay in space at its last known good position. Previously, there was no indication in the UI of this state, which made it difficult to diagnose problems. Now, the title of the group box is appended with “(!)” when the attachment is in the “Frozen” state. Additionally, a tooltip has been added to the title that explains the current state- that is is frozen (and potential reasons why) or it names the geometry that is being reference.

- The default location for the template directory has been moved to the appropriate platform-specific location for user data. On Windows, this is in the Remcom/XFdtd folder of the users's Documents (or My Documents) folder, while on Unix this is `$HOME/Remcom/XFdtd`. The new location applies to new installations of XFdtd only; if the template directory was already specified, it will not be changed. The template directory can always be changed manually in the Application Preferences.
- When XFsolver cannot detect GPU hardware for XStream® simulations, it now prints a more informative message to the output about the problem and possible solutions.
- When XFsolver encounters unrecoverable errors such as the inability to allocate GPU memory (for XStream® simulations), an informational message describing the problem is now written out to both `project.log` and `project.diag` to inform the user of the problem.
- Several improvements where made to the measuring tool, including printing dx, dy and dz in the view, removing markup when ESC is pressed, providing more markup at the measured locations, adding the total length to the status bar display, and optimizing the status bar display for both Distance and Position to be more readable.
- Version 7.1.1 added some warning messages during simulation creation about using components in S-Parameter computations that have reactance or inconsistent resistances. This was causing undue concern for many users, so these messages have been moved to the `project.diag` file instead. These messages now also include warnings when Switches, Nonlinear Capacitors and Diodes are classified as ports.
- Added the ability to perform regular expression searches in the Result Browser and in the Parts List. To turn on RegExp searches, press the Search button and select “Regular Expression” (the default is “Simple”). When using this type of search, “Enter” must be pressed after entering the search terms to initiate the search (when in Simple mode, the search is immediate). The most common usage for the regular expression search would be search for multiple items, such as XXX and YYY and ZZZ. This would be specified in the search field as: “XXX|YYY|ZZZ”. Additionally, the search fields in both the Result Browser and the Parts List now remember the last few searches that were entered.
- Added the ability to have shortcuts to Locators and to move and copy Locators within a single Part.
- A plot of probability density function (as used for mean effective gain and antenna diversity computations) is now provided on the PDF tab of the far zone field viewing controls.
- Significantly improved performance when setting all subparts visible/invisible or including/excluding them from the mesh.
- The spatial coordinates of data values when using the Field Reader tool are now displayed on the status bar.

- Added the ability to have shortcuts to Locators and to copy/move Locators within a single Part.
- Resolved issue where Linux 64-bit executables could not be licensed with a USB dongle.
- Information about the selected Part(s) is now shown at the top of the Parts List. When one part is selected, its full assembly path is shown (and is selectable for copy/paste operations). When two or more parts are selected, the number of selected items is shown. This space also has a tooltip that provides additional information on the selection.
- The Edit context menu for Parts now has an additional item “Copy Full Assembly Path Name” that will copy the full assembly path as a string to the clipboard to make it convenient to include this path in other applications (such as an email message or report).
- For CAD Import operations when “Automatically Assign Materials” is selected, a fallback for assigning the material, gridding, meshing and visibility attributes is to search for a Part with the same name as the newly imported Part.
- Renamed “Wideband Eval Frequency” in the “Sampled” electric and magnetic properties in the Material editor to “Broadband Eval Frequency” for consistency. Also added a tooltip to that field to describe what the field means and when it is used.
- The Locators editor now uses the appropriate cursors to indicate the attachment state for picking.
- The toolbar in the Geometry View containing the view controls, which was previously always to the right of the view, is now movable to any of the 4 sides of the view.
- The default action when “Return” or “Enter” is pressed when in the Create Simulation Dialog is now to do nothing. Previously, “Create and Queue” was the default action which caused simulations to be created accidentally when “Enter” was pressed unintentionally.
- The “Diagnostics” tab of the Simulations window is now automatically updated when it is switched to so that the user no longer needs to click “Update” manually after navigating to that tab.
- When Compute Dissipated Power and Save Fields on Far Zone Box for Post Processing are both unchecked for a simulation, some System output is not available. Previously, this resulted in the values for Dissipated Power, Radiated Power and Efficiency to be displayed as “NaN”. Now, “No data” is displayed instead and a tooltip for that field is displayed with more information about why “No data” is shown.
- Added new File menu item “New Window” that can open a new or existing project in a new instance of XFdtd. The new instance will be the same version that is currently running.
- Added ‘s’ hotkey for Picking Tools to show all parts that were previously hidden by pressing ‘h’.
- New SAR binary file format added to Appendix D of the Reference Manual.

Scripting

- Added several methods to *Sketch* to facilitate working with vertices: *getEdgeIds()*, *getVertexIds()*, *getStartId()*, *getEndId()* and *getPosition()*.
- Added the ability to manage preferences from scripting through the following methods on *App*: *getPreference()*, *setPreference()*, *clearPreference()*, *clearAllPreferences()* and *getDefinedPreferenceNames()*. These methods allow the user to easily store and retrieve persistent data that can be used to prepopulate dialog choices, for example.

- Added methods to *Grid* to obtain automatically extracted fixed points: *getAutomaticFixedPoints()*, *getAllAutomaticFixedPoints()*, *getGriddingParameters()*. At the same time, the methods *getGridRegionParameters()* and *getPartGridParameters()* were deprecated- use *getGriddingParameters()* instead.
- Added method *Project.getSimulation()* to access existing simulations. The returned object can be queried for status, for example.
- Added method *App.getPartVisibility()* to obtain the visibility attribute of a Part
- Implemented *move()* for the following "container" objects: *CircuitComponentDefinitionList*, *CircuitComponentList*, *ExternalExcitationList*, *GraphList*, *FarzoneSensorList*, *NearFieldSensorList*, *SensorDataDefinitionList*, *StaticVoltagePointList*, *Assembly*, *MaterialList*
- Fixed problem where a Part's material could not be set through scripting.
- Fixed bugs in the two-argument and three-argument versions *UnitClass.convert()* that prevented conversion to linear scalar units.
- *LocatorList* is now properly documented and accessible. Previously, *Part.getLocatorList()* would result in an error message.
- Added implementations of *Loft*, *Reflect*, *Rotate*, *Shear*, *Scale* and *Translate*
- Fixed crash when *Recipe.flatten()* is used before the Recipe has been added to a Model.
- *MaximumFrequencyWaveformShape.AllFrequenciesSpecificationType* and *MaximumFrequencyWaveformShape.SpecifySpecificationType* are now defined properly. Previously, they were not defined properly such that *MaximumFrequencyWaveformShape.setSpecificationType()* would not result in the correct setting.
- Fixed problem in all "container" objects (e.g. *WaveformList*, *MaterialList*, etc.) where the "replace" methods would not properly update the name of the replaced item in the Project Tree until it was edited.
- Fixed crash when using *Model.setRecipe()* when the passed-in Recipe contains no Features.
- *GroupBox.title* is now properly documented.
- *PointSensorDataDefinition.getSampledTimeRange()* is now properly documented.
- Added more documentation to *CoordinateSystem::getOrigin()* to describe its use and limitations.
- Fixed bug that would sometimes cause the application to abort when *GraphAxis::getTheme()* was called.

Fixes

- Fixed situation where a simulation with XACT would run on CPU but would crash using XStream® before time-stepping began when more than one thread was specified.
- Changed the field interpolation mechanism XFsolver uses to compute the HAC sensor maximum E-field and H-field values.
- Fixed XFsolver crash that may occur for some geometries when using an active feed with a series RLC arrangement combining more than one of the resistor, capacitor, and inductor.

- HAC sensors are now supported for MPI simulations. Previously, MPI simulations would crash when computing HAC.
- Fixed problem where some files were not being closed properly, which could cause messages about “Sensor Output Error” during post-processing when there were many components.
- Fixed bug which could cause XFsolver to abort when running with more than one thread when certain combinations of surface or solid sensors collecting steady-state data were specified.
- Fixed issue where polar Directivity data would not be plotted.
- Resolved case where the application would crash in some cases when using Zoom to Extents with a sketch that was not valid.
- Fixed issue where a very large project would become corrupt when saved. When reloading the project this problem resulted in an application crash or a corrupt project message.
- Fixed a problem where components that shared the same Yee cell node were incorrectly identified as being on the same edge.
- Made further corrections to simulation memory estimation for XStream® simulations.
- Addressed the situation where results were listed as “Complete” when a simulation failed with an error condition. Trying to plot those results would then give the error message “No timestep data is available”. Now the results will be properly “Unavailable”.
- The equation for magnetic conductivity where fixed to use μ_0 instead of ϵ_0. This caused the conductivity computed from complex relative permeability to be incorrect.
- The CAD Import and Merge process was setting parts as “Meshing Disabled” rather than unsetting “Include in Mesh” when parts that were previously deleted are reimported. Similarly, the XFdtd 6 import process was doing the same when importing unmeshed parts. This caused a confusion when checking these parts as the “Include in Mesh” did not seem to have any effect- even though the part is included in the mesh, it wasn’t being meshed because its meshing type was disabled. The Import and Merge processes now set “Include in Mesh” instead.
- Fixed issue where XFdtd version 6 geometry that contained wire bodies would sometimes result in wires that were not oriented properly after import.
- Plotting different types of results with the “View (Default)” action at the same time no longer creates multiple undo items; pressing ‘Undo’ once will now undo the creation of all graphs created by that action.
- Fixed issue where Graphs would sometimes not draw properly until they were resized.
- Fixed crash while creating a simulation with a small number of XACT edges during the “Analyzing XACT Mesh” phase of simulation creation.
- Resolved issue where application would crash when clicking on a Part in the geometry view when a very large model was loaded.
- Addressed issue where the application would crash due to running out of memory when the Construction Grid was visible and the view was oriented such the the grid would grow very large. The number of grid lines that can be drawn is now limited; if that limit is reached, only the bounding box of the construction grid is now displayed.

- Fixed problem where CAD files could not be imported if the path contained UNICODE (non-ASCII) characters.
- The Distance readout in the status bar when using the Measuring Tool now always goes away and highlighted edges in the view disappear when the Measuring Tool is deactived.
- Fixed crash when undoing the deletion of assemblies and subparts when both the assembies and subparts were selected when deletion occurred.
- Fixed issue where copy-n-paste of a component with both ends attached to the same part to another instance would cause two nodes in the Project Tree for that Part. Deleting one of the nodes would then cause the application to crash.
- Fixed crash when deleting a Part that has a SAR sensor attached to it while the sensor was being edited.
- Fixed issue where the Position coordinates displayed in the status bar when using the Picking tools were incorrect in many cases.
- Fixed a problem in tabular result windows (e.g. System Output) where if the scaling factor is set to zero (such as by entering '0' for a scaled quantity like Net Input Power), scalable quantities are no longer editable.
- Fixed incorrect loading/interpretation of user-defined probability density function files.
- Removed 'Delete' context menu item from many items in the Project Tree that should not have had it.
- Fixed bug that caused object deletion from the context menu to fail in rare cases after using Select Shortcut Target.
- Resolved intermittent crash during the creation of a 3-point arc in the Sketcher.
- Fixed bug where copy-n-paste of once-attached circuit components could crash the application.
- Resolved crash using using "Modify→Transform→Rotate" or "Modify→Transform→Rotate" from the Geometry View.
- Resolved issue where in certain cases, the Field Reader would report incorrect values when viewing XZ-planar slices of volumetric data.
- Fixed problem where in a special case dragging a circuit component between application instances could add an empty part to the Project Tree of the target instance, making the target instance unstable.

7.1.1.x

7.1.1.2

Note: 7.1.1.2 is the initial public release of the 7.1.1.x series

New Features

- Simulations using External (Plane Wave) Excitations now fully exploit GPU hardware when using XStream®. Users of this capability should see a significant performance improvement.

- The ability to define arbitrary locations and orientations on any part has been added called “Locators”. Any two locators can then be “matched” in order to quickly align two parts. This is a generalization of XFdtd’s “Match Points and Directions” capability. See “Using Locators to Orient Parts” in the Reference Manual for more information.
- The NVIDA Fermi architecture is now officially supported for XStream® simulations. All XStream® users (except Legacy XStream®) will need to upgrade to the latest NVIDIA video driver to use XStream® or XFsolver will report that it cannot find CUDA-capable hardware.
- Geometry meshed with XACT can now touch or extend past PEC boundaries.
- Added the new lumped component type RL||C, useful for modeling a lossy inductor. This new component type can be used as a Passive Load and is defined just like the previously available RLC and R||L||C combinations.
- Peak SAR values in the currently displayed data selection are now available on the “Statistics” tab when viewing SAR data.
- Added circularly polarized UAN file output for far zone data from XFsolver when the `-outputcpuan` command-line option is used. The new output file name ends with `.cpuan`. Also added active feed location and frequency to the header section of `.uan files`.

Usability/Performance Improvements

- SAR data is now written out from the engine in a binary format rather than gzipped ASCII. This improves performance when reading the data for display, making display of SAR data significantly faster.
- Added an option to ODB++ Import to automatically add Manual Fixed Points on the edges and inside layers of ODB++ geometry.
- Warnings generated during simulation creation are now appended to the Simulation Summary available in the lower pane of the Simulation Window.
- Added the ability to reorder plots in a graph.
- When plotting data versus material via the Create Line Graph dialog, the dialog now shows the material name. An example of this is plotting Dissipated Power per Material on a line graph.
- Component names have been added to XACT feed-edge error messages in order to aid in determining to location of the problem.
- In the Waveform Editor, the text “Excite up to a Maximum Frequency” has been changed to “Specify Frequency Roll-off” to avoid confusion about what frequencies would be excited by the waveform and to more accurately describe what is being specified.
- The visibility and opacity settings for Parts, Components, Sensors, Output and Bounding Box (as controlled by the buttons along the right-hand side of the Geometry Window) are now saved and restored with the project.
- Added simulation-creation time checks to warn about cases where S-Parameters may not be correct due to complex port impedances or non-uniform resistances across all ports.
- Added simulation-creation time check to validate that a sinsoidal waveform’s frequency or specified Frequencies of Interest do not exceed the maximum frequency for the simulation (as defined by the FDTD limitation of 10 cells/wavelength).

- The Simulation Name edit field in the New Simulation Window now features auto-complete for previously entered names.
- Additional result types are now plotted on a single graph when "View (Default)" is chosen for multiply selected results. Specifically, plotting Far Zone and RCS results should now behave more as expected.
- Wheel mouse no longer inappropriately activates pull-down comboboxes.
- Component and Sensor validity behavior has been changed so that they are not always simply valid when they are disabled.
- Incident Amplitudes reporting in the External Excitation Editor has been reformatted for better display.
- Added more information to a diagnostic message that reports when there is difficultly writing data during a simulation. The previous message was not clear that it was for informational purposes only and that data integrity was not compromised.
- The tooltip for the Project node in the Project Tree now shows the full path to the project.
- The sinusoidal waveform name has been changed from "Ramped Sinusoid" to "Sinusoid" since the ramping part is generally inconsequential and generated confusion.
- Changed error message when grid bounds are invalid to be more specific about why there is an error.
- Shortcut Groups are now created at the selected level of the the Groups tree instead of always being created at the top-level.

Scripting

- Fixed problem where *CadImport.exportCad()* was failing improperly with the message "Invalid Assembly to import into".
- Fixed issue where calling *ResultScalingManager::setScalingFactor()* or *ResultScalingManager::getScalingFactor()* would result in an error message.
- Modified *App.saveCurrentProject()*, and added *App.saveCurrentProjectAs()* and *Project.saveCurrentProjectCopyAs()* to match the behavior provided by the UI for "File→Save", "File→Save As..." and "File→Save Copy As...".
- Added *ResultTopology.getScaleResults()* and *ResultTopology.setScaleResults()*.
- Added *FarZoneUtils.rotateAndInterpolateEFields()* and *FarZoneUtils.exportRotatedGainAndPhaseToUANFile()*.
- Removed *ResultQuery.SurfaceCurrentSensor* as a documented Sensor type since it is not available.
- Fixed issue where Booleans created through scripting did not result in the same object as when created through the UI in some cases when one of the source objects had a non-default working coordinate system.
- Added missing method and attribute documentation for *Color*.
- Fixed problem that could result in the application exiting unexpectedly when using *SweepPath*.

- Fixed scripts/primitives/helix.xmacro to not have units for “Start Direction” and “Wire Normal Direction”.

Fixes

- Fixed incorrect reporting of time stepping start time in project.diag file.
- Fixed an issue where simulations run on CPU may go unstable when using more than 4 threads.
- Fixed problem where far zone data was missing at the Z=0 plane for simulations where the lower Z-boundary was PEC.
- Resolved problem where conduction current for anisotropic electric materials with non-zero conductivity may have been incorrect.
- Corrected computation of port voltage (polarity) for a Diode.
- Fixed issue where XFsolver would exit unexpectedly for a simulation using the static solver and a maximum timestep setting of one.
- Added checking for intrinsic capacitance and inductance for voltage sources to ensure that appropriate values are always being used.
- Fixed problem where dissipated power computations could be incorrect in certain situations when XACT is enabled for lossy dielectrics.
- Resolved issue where using XACT with XStream® could give different results than running the same simulation on CPU, especially when objects meshed with XACT are close the the boundary.
- Fixed problem with unexpected data in volume sensor *_geom.bin and *_info.bin files.
- Fixed an issue where the application could exit unexpectedly when viewing a slice of the mesh if there was a meshed object very close the edge of the simulation space.
- Fixed a problem where the relative permittivity of a material could be set to “inf” but then a simulation using that material would go unstable. Now the material will become invalid if “inf” is specified.
- Fixed incorrect computation of Elevation and Azimuth polarizations of far zone data.
- Restored .mesh file import to pre-7.1 performance (7.1.0.x imports were very slow compared to previous versions).
- Fixed issue where adding a Frequency of Interest through scripting would cause a recursive repaint when subsequently going to the Frequencies of Interest tab in the Create Simulation window.
- Fixed problem where the tooltip for an Anchor Point was not converting coordinates for the selected display units.
- Resolved an issue where viewing results of a tiny surface sensor, small enough so that it has no mesh edges at all associated with it, caused an “unexpected condition” warning message.
- Fixed an issue where in rare circumstances an invalidity message stating that all “Ports must be contained within the grid” even when this was already true.
- When creating a simulation fails completely, files that had already been written are now properly cleaned up (they were previously left behind).

- Fixed issue where the name entered in the editor when creating a Sheet Body from Faces was ignored.
- Fixed problem where text could be cut off in Meshing Parameters editors.
- Fixed issue where some graph color changes were not reverted upon undo.
- Fixed a problem where changes while editing project properties (such as Project Notes) would be discarded whenever a simulation was running and a graph of data from that simulation was being displayed whenever the graph updated.
- Fixed issue where with some window themes, selecting an object in the Geometry View would apparently not select that object in the Project Tree. In actuality, it was selected in the tree but the windowing theme was rendering the selection differently since the Project Tree did not have keyboard focus. Now highlighted text in the Project Tree is always displayed as though that window has focus.
- Net Component Loss and Dissipated Power (total and per material) are now available for external excitation simulations when the simulation option “Compute Dissipated Power” is checked.
- Fixed issue where the display of the data bounding box for sensor data did not obey the “Output Visibility” setting; instead it was always visible.
- Resolved problem where the alpha channel of a material’s color was being reset when it color was re-edited.
- Resolved issue where a progress dialog would always briefly pop up when selecting the “Chop” operation when creating a Boolean.
- When a plot is made invisible, its entry in the legend is now removed as well.
- Fixed a case where under rare circumstances, viewing Normal mesh edges could cause the application to exit unexpectedly.
- Corrected error message about running simulations when closing XFdtd when a simulation was not actually running.

7.1.0.x

7.1.0.5

Note: 7.1.0.5 is the initial public release of the 7.1.0.x series

New Features

- The major feature added in this version is Accurate Cell Technology (XACT), which improves simulation speed and accuracy by representing geometric features at sub-cell resolution in the mesh. See the XFdtd Reference Manual for details.
- Invalid Features in a Modeling Sequence now indicate they are invalid and cause the Modeling Sequence and therefore the Model to be invalid when the feature cannot be applied properly. This change means that problems in Models will be much easier to locate and parameter sweeps that sweep through values that cause such an invalidity will not be quietly ignored.

- Transform Entity operations (Scale, Translate, Rotate, Shear and Reflect) are now Modeling Sequence features of Parts. Previously, one would need to Flatten a modeling sequence before being able to apply a transformation and that transformation wasn't parameterizable. This is no longer the case. Additionally, the transforms can be applied to Assemblies.
- The executables that comprise XFdtd have been renamed. This affects the commands that one uses to start XFdtd from the command line (the Windows Start Menu is unaffected). To start the XFdtd user interface, use the command `xfui` from the appropriate platform bin directory. To start the calculation engine use, use the command `xfsolver`.
- xfsolver now uses CPML for its PML boundary condition instead of UPML. This primarily resolves issues with long simulation initialization times (and in rare cases a hang), and also allows dispersive materials to touch the outer boundary.
- For Linux, improved the xfsolver launch script to be able to start the MPI version of xfsolver. Now this script should always be used to start xfsolver from the command line on Linux instead of running the executable directly. In addition to the command line options excepted by the executable, the launch script also understands the `-ax8` option to use Legacy XStream® (Acceleware version 8) and the `-hpmpi` option to use the HPMPI version of xfsolver.
- Added an epoxy dielectric layer to ODB++ Import to fill in space between boards where traces do not exist.
- Added ability to perform Far Zone postprocessing-only with the calculation engine with the `-farzoneonly` command line option. Additionally, when post-processing Far Zone from the UI, the user will have the option of performing the post-processing synchronously in the UI, asynchronously using the engine, or manually using the engine.
- Some Boolean operations are now available from the context menu when selecting more than two Parts. For example, one can now select several Parts, right-click, select "Boolean" and then "Union" in order to easily union together several objects.
- Added ability to scale B-Field steady-state output the same as H-Field output.
- Added new output types "Realized Gain" and "Directivity" for far zone results.
- The solid modeler has been upgraded to ACIS R20sp3, which is now the default format when saving and exporting geometry.
- XStream® now fully supports PMC outer boundaries (the use of PMC was previously restricted to the XStream® Acceleware solution).

Usability/Performance Improvements

- Picking Tool behavior has been modified to no longer automatically attach the picked item (plane, direction, vertex) to the geometry it references. To attach the picked item, press 'a' before picking (see the tool tip that pops up while the Pick Tool is active). This will cause the cursor to change to a pointer with a thumb tack, indicating that the picked item will be attached to the geometry. To restore the previous behavior (of automatically attaching), see the "Compatibility" tab in the Application Preferences.
- Added simplified Grid settings controls in the Grid Editor and in Gridding Properties editors. The simplified, or "Basic", controls are now the default. To use the previous controls, use "Advanced Mode" in the Grid Editor or Gridding Properties.

- Moved the location of some context menu items to make the menu smaller and more consistent. Some of the changes: Create Pattern moved to the Modify submenu; Group into Assembly moved up. Gridding and Meshing menu items grouped together into a submenu.
- Added context menu item in entry fields to insert units and parameters from the Parameter List.
- Select-through functionality in the view was moved to the center mouse button. Previously, one needed to “left click, then right click”, which was non-standard and confusing.
- Added ability to not scale vectors and points with their values when viewing fields. This feature is available in the new “Scale Points” and “Scale Vectors” checkboxes on the Setup tab in the Field Viewer. Far Zone data has a similar new feature, “Scale Radius”, which when disabled will display field values on a sphere rather than scaling the radius at which the values are rendered. By default, these options are enable to maintain the same behavior as previous versions.
- Modified behavior of automatic scale settings in the Field Viewer for quantities shown in dB so that it no longer will have a range of [-300,0], but generally [-70,0]. The default color settings also now generate colors in 10 dB increments for this default range.
- Relaxed sensitivity of edge and vertex selection. The selection was previously very sensitive to mouse location and movement, resulting in the inability to either select or drag them as one would expect. Feedback was that sometimes it worked, and sometimes it didn’t; now it should always work.
- The XFdtd UI and simulation engine now properly handle UNICODE characters in all cases except in Plots and in the View, where they will appear as rectangles (this will be addressed in a future release). Previously, there were issues with type UNICODE characters in a few fields in the UI and with UNICODE characters in project directory path and file names.
- Resolved issue in the Sketcher where snapping to edges and vertices did not result in exact placement. This led to many “open profile” and incorrect trimming problems.
- The “Face Normals” tool was renamed to “Align Directions” and the “Match Faces (Point & Normal)” was renamed to “Match Points & Directions” in order to more clearly describe their actions.
- The “3-D” View Mode in the Field Viewer has been renamed “3-D Offset” to help avoid confusion. The Size Factor is also now applied to the offset for this mode.
- Renamed “Load Project” to “Open Project” and added “Close Project” to be consistent with the terminology of other tools. “Close” unloads the current project and loads an “Empty Project” into the application, whereas “New Project” unloads the current project and loads a new project using the default template into the application.
- When licensed using a license server, the application no longer exits unexpectedly if connection to the license server is lost. The application will pause with a message box until the connection is reestablished or terminated by the user by dismissing the dialog.
- The performance of material visibility changes has been improved, especially for imported mesh/voxel data.
- Materials are now created with a unique color when possible. The previous behavior of always using the same default color as defined in the Application Preferences can be set in the Application Preferences Modeling Tab under Default Appearances, Material.

- Added a button to force a refresh of calculation engine output from the filesystem. This is useful when running simulations at the command line or remotely over a shared file system, i.e. not through the Simulation Queue within the application.
- The Application Preference “Clear Results on Project Load” now defaults to being checked based on user feedback.
- Added Application Preference for the color of the Construction Grid.
- Added Application Preference for the color of grid lines in planar mesh displays where there is no material defined (freespace).
- The project Open and Save dialogs now show only directories.
- Added “Select All Faces” button and text indicator of the number of faces picked when creating a sheet body from an existing part. Also added an text statement of what Model is selected when performing Model selection for Modify operations.
- Improved progress reporting for ODB++ Import.
- ODB++ Import improved to handle setting and restoring the New Part Modeling Unit properly, and also where to find this setting in the Application Preferences.
- Allow nets in ODB++ to be broken into multiple parts on import according to layer.
- No longer show frequency spectrum results for simulations excited with a sinusoid, which was causing confusion. For the minority of situations where this data is useful, this data is still available through the scripting API.
- The “Set Only This Material Visible” context menu item for materials now works when multiple materials are selected (and says “Set Only Selected Material Visible”).
- The “View” context menu item for the Mesh node in the Project Tree has been replaced with a checkbox to make its use consistent with its behavior.
- Dramtically improved performance of Mesh Slice rendering when Normal edges are turned on.
- Results that are loaded in the Field Viewer are now automatically unloaded if the project they came from is unlisted in the Results Browser. This resolves an issue where the result selection dropdown box in the Field Viewer would have an invalid entry in such situations.
- The pulsewidth for projects imported from XFdtd v6 is now properly adjusted for use in version 7 instead of being used verbatim. In the case where a Gaussian waveform with a pulsewidth of 32 is imported, a Broadband Pulse will be used in version 7 instead, which is the intent of the imported waveform.
- The DFT Sampling Interval entry field was artificially limiting the interval to 1024. This limit has been raised to 1M.
- The waveform that was used for a simulation is now reported in the Simulation Summary.
- The ‘XFdtd (Direct X)’ shortcut in the start menu was removed since it was causing confusion among users and is rarely ever needed.
- A Part now becomes invalid if its bounding box is empty.

- Added text to the Scale Bar in the Field Viewer to indicate what quantity and units are being displayed.
- Project validity now detects when two or more components are meshed onto the same cell edge.
- Added detection and loading of an XFdtd project that is inside a folder of the same name. This is usually the result of unzipping a zipped project, since most zip utilities automaticaly create a subfolder with the base name of the zipfile and place the contents of the zipfile in there. For example, zipping a project `MyProject.xf` results in `MyProject.xf.zip`. Unzipping this file by default will create `MyProject.xf/MyProject.xf`.
- When editing, Cancel buttons throughout the application are now always enabled; this makes their behavior consistent with most other applications.
- Selection of Display Units in the Project Properties when "Show All Units" is *not* checked is modified to include only units most useful for EM simulations.
- Added units "uW", "nW", "mV/m", "dB(mV/m)", "uV/m" and "dB(uV/m)"
- Changed the Pick Position tool's tooltip to be more accurate about what can be picked.
- Added tooltips to all fields in the Create Line Graph dialog to clarify the use of each field.
- Added explanatory note to the Default Appearances tabs in the Application Preferences.
- Added number of manual fixed points and grid regions that are defined to the appropriate tab text in the Grid Editor.
- The Remesh Now item in the Mesh node's context menu is now always enabled when the mesh is out-of-date to allow the user to force the entire mesh to be recomputed instead of allowing incremental meshing to occur automatically. Previously, this item was enabled only when "Automatic Remeshing" was not selected.
- Provide a warning message when a completed simulation is queued to run again.
- Added a filter in the Import CAD dialog to show only Pro/E Assembly files. This is useful for large models where there are many parts and assemblies mixed together.
- Renamed the "New Simulation" button in the Simulations Window to "Create Simulation" for consistency with the title of the window that opens when this button is clicked.
- Modified Smith chart markers to show $x - yj$ instead of $x + -yj$.
- Fixed labels for Impedance plots created through Create Line Graph to be more explicit rather than just being "Resistance".
- Create Line Graph now allows adding plots to empty graphs.
- Create Line Graph now puts all selected S-Parameter or Impedance results on the same graph rather than putting each in its own graph.
- Added ability to specify relative dB on 1-D graphs.
- Changed all uses of the term "Fixed Planes" to "Fixed Points" to provide consistency across the application.
- Corrected order when tabbing through the entry fields of Sensor Definitions.

- Rendering of components in the mesh representation now display the component along the whole edge. Previously, there was a small gap between the representation of the component and the next mesh edge, which caused concern that the component was not connected (even though it was).
- The Scale Bar Properties for the results shown in the Geometry View now as a “Default” button to reset the color settings to their default values.
- Improved performance of shutting down XFdtd. In some cases, XFdtd would continue running for a while after the UI disappeared.

Scripting

- Added *Project.updateMesh()* to force the entire mesh to be recomputed instead of allowing incremental meshing to occur automatically. This is equivalent to the “Remesh Now” item in the context menu of the “Mesh” node in the Project Tree.
- Added *Project.setAutoUpdateMesh(bool)* and *Project.getAutoUpdateMesh()* to set and query automatically updating the mesh when needed. This is equivalent to the “Automatic Remeshing” checkbox in the context menu of the “Mesh” node in the Project Tree.
- Added *ThickenSheet*.
- *Add Recipe.flatten()* to provide the ability to flatten a Modeling Sequence, as available in the UI.
- *SmithGraphMarker* and *PolarGraphMarker* were documented but not available in the scripting interface.
- Fixed issue where *LinearGraphAxis.setUnit()* and *CommonGraphAxis.setUnit()* were overloaded and therefore could not be disambiguated when used with UnitClass.Dimensionality. New methods *.setUnitClass()* where added to these classes to provide that functionality.
- Added *DataSource.getDataSetIds()*
- Fixed issue where *SensorDataDefinitionList::getSensorDataDefinition()* did not return an object that allowed subclass-appropriate methods to be called.
- Fixed issue where *LinearGraphAxis.setUnit()* and *CommonGraphAxis.setUnit()* could not be called with *UnitClass.Dimensionality*. In each of these classes, a new method, *.setUnitClass()* was added for this purpose.
- Added scripting API for Part tranformations: *Scale*, *Translate*, *Rotate*, *Reflect* and *Shear*.
- Added script `SetTissuesForVariPoseMan.xmacro` to the scripts library. After loading a `.mmf` file from VariPose®, this script can be used to automatically setup the electric parameters of all the tissues.
- Added an optional pass-parameter to *Part.publishSAT()* to specify the ACIS version with which to save the file.
- Added *PartFacesSensorGeometry.setAllFaceIdsOnPart()* to tell the sensor to collect on all faces of the geometry.
- Added *Model.getNumberOfLumps()* and *Model.getLumps()*, the latter of which can be used to to separate a Part into muliple Parts (as the “Separate” context menu item on a Model does).

Fixes

- Fixed issue where simulations using Total/Scattered Field formulation with Periodic Boundary Conditions would fail with a message about not being able to compute far zone information, even though far zone should not have been computed.
- Resolved issue in the simulation engine where in some cases XStream® simulations would not fall back to CPU when they should have.
- Fixed an issue where postprocessing would not be performed if XStream® was used and the number of threads specified by the user was more than the number licensed. This resulted in what appeared to be a successful simulation, just with missing results.
- Fixed an issue where RCS transient far zone was incorrect when using more than one thread and XStream®.
- Fixed issue where a warning message was printed from the engine `N` times when `-proc N` was used.
- Fixed issue where a simulation using an External Excitation and a Feed would excite both of these sources even when set to use only the External Excitation.
- Resolved issue where very small XStream® simulations may have become unstable when using multiple GPUs.
- Addressed an issue where convergence might not be triggered for simulations with a very wide broadband pulse. This resulted in the simulation running to the maximum specified timestep with 0% convergence.
- Improved performance of writing transient far zone files. This post-processing step when running simulations was taking much longer than it should have.
- Fixed an issue where rerunning a simulation did not properly reset the simulation status files, which in some cases could make it impossible for the UI to properly display results from that simulation.
- Corrected computation of max HAC E-Field value, which was using Ex in place of Ey in the formulation.
- Resolved an issue where sensors of the same type with names that were not unique when compared case-insensitively (such as “Port XY” and “Port xy”) caused the results for those sensors to be written incorrectly and therefore displayed incorrectly.
- Resolved issue where XFsolver would crash when more than 20 layers of PML was being used with XStream®. Now, a warning message will be printed to the effect that more than 20 layers of PML is not support and XFsolver will attempt to fall back to the next supported computational method for the simulation.
- Fixed XFsolver crash and/or bad when performing SAR postprocessing for individual tissue types.
- Fixed several issues related to results updating that resulted in invalid plots or application crashes.
- Fixed issue where an corrupt mesh file that was corrupt caused the application to fail upon project load.
- Resolved issue where the Loft operation did not properly validate the Smoothness parameter.
- Resolved issue where scaling an imported object by -1 caused the application to abort.
- Fixed issue where the export of a single far zone slice to a UAN file caused an unhandled exception.

- Fixed grid generation algorithm to properly handle situation where a Grid Region having a bounds smaller than the smallest allowed cell size resulted in an infinite loop. In this situation, the grid region that is too small will be handled as if it were at least as large as the minimum cell size. (This condition is very rare.)
- Fixed a situation where the grid generation algorithm could enter an infinite loop if conditions existed that required that a grid region to be moved slightly. This condition is now detected and the grid is set invalid instead. (This condition is very rare.)
- FFT and DFT data are now always normalized using the simulation timestep. This means that now that the magnitude of FFT and DFT data can be compared across simulation with different timesteps. Since such data is usually viewed in relative terms, this is not a change that most users should notice.
- Resolved an issue where `.mesh` files imported from XFdtd 6 would sometimes not have the correct cell size.
- Corrected SAR relative dB conversion to use $10log10()$ instead of $20log10$.
- Fixed an issue where if a parameter name contained unicode, the parameter list would not be restored properly when reloading a project.
- Fixed issue where an Elliptical pattern in a saved project was not reapplied when the project was reloaded.
- Material colors are now exported with the object when exported to a CAD format.
- Import of Pro/E CAD models no longer includes suppressed parts when imported. Including them produced unexpected results.
- Fixed a problem when saving cached geometry with the project that sometimes resulted in geometry not reappearing when the project was reloaded.
- Features that operate on multiple lumps no longer leave a partial model if the feature does not succeed on at least one lump (such as Thicken Sheet). The Model now becomes invalid instead.
- Using the Lock Constraint on a vertex in the sketcher and using “Lock Position” from a vertex’s context menu now both result in a “Lock Fixed Position” constraint for consistency. Previously, both methods would properly lock the vertex’s position, but the constraint was handled differently in the UI.
- Wire bodies, previously ignored, are now properly imported from CAD objects.
- Fixed issue where saving a project on Linux to a filepath with a symbolic link in it prevented the project from being reloaded properly.
- Corrected issue in UAN file output where the same value of net input power was being used for all DFT frequencies instead of the correct net input power at each frequency.
- Fixed an issue in scripting that could cause property accessors and method calls on local variables to unexpectedly stop working in rare cases, typically after removing objects like Parts or Materials from the Project.
- Resolved issue where SAR results would not load if there was no distance between a SAR sensor using a manually specified box and the edge of the simulation space.

- Resolved issue where a consituent parts of a Boolean operation would not be valid after saving and reloading a project.
- Added unit of Magnetic Conductivity (ohms/meter) and fixed units of magnetic conductivity in the material editor, which was erroneously using electric conductivity units.
- Fixed issue where modeling operations involving Sheet Bodies were not handled correctly if the operation was created in XFdtd 7.0.2 or before. This resulted in a Model that appeared to be correct when loaded in a later version of XFdtd but went invalid when further operations were applied to the Model.
- Corrected the computation of interpolated surface sensor data. In certain situations, this issue resulted in the displayed data being shifted by 1/2 of a Yee cell.
- Fixed problem where some unit conversions were occurring at a lower precision, causing rounding errors when converting back and forth between units. This was most evident when specifying values in non-metric units.
- Fixed issues where performing a boolean on Part referenced by sensors or circuit components would result in that Part having two representations in the Project Tree, which then generally resulted in application failure when further manipulating that Part.
- Fixed issue where non-axis aligned Planar Sensors would not render correctly in certain situations.
- Fixed problem where the timestep would not update immediately after disabling the Custom Timestep Multiplier in the Project Properties.
- Removed “Separate” context menu item from the main Parts node in the tree, which did nothing.
- Fixed problem where progress went backwards when preparing multiple plots to be shown.
- Fixed situation where deleting Static Voltage Points could cause the Create Simulation Window to not allow a simulation to be created with the invalidity message “Static solver points must be defined when no sources is chosen”.
- Fixed issue where the Create Simulation Window did not check to see if edits were in progress before opening, allowing changes to the project during editing.
- Fixed issue where entering an invalid formula in any field of Marker Properties caused an exception that produced a warning message about the application possibly becoming unstable.
- Fixed problem where deleting a Waveform Definition while it is open in an editor causes the application to abort.
- Fixed issue where the axis of rotation for a Circular/Elliptical pattern was rendered in the wrong location when the working coordinate system for the Part was not the same as the global coordinate system.
- Fixed issue where the `materialFrequency` parameter was created for some, but not all, imports of XFdtd version 6 files. This resulted in the project being invalid after importing some files (`.idm` and `.mesh`).
- Fixed issues in ODB++ Import where drill extended too far into other layers and where drill were at incorrect z-locations in some cases.

- Fixed issue in ODB++ Import where the drill pads incorrectly had constant radius throughout all the layers.
- Fixed missing icons for sensors in libraries.
- Resolved issue of not being able to import DXF file with the file extension `.DXF` (`.dxf` did work).
- Fixed issue where the list of Simulations was being being unecessarily refreshed when saving a project. This resulted in extra project save time for a project with many simulations.
- Fixed issue where a simulation could be created while in the process of editing something. Now, the user will be asked for the disposition of the edit before opening the Create Simulation dialog.
- Fixed issue where the slider button of the Opacity controls of the Geometry View would sometimes be hidden under the adjacent spin box.
- Fixed the rendering of the axis and root position for Elliptical patterns, which were drawn in the wrong location in some cases.
- Fixed issue where performing a Boolean Intersection of multiple Parts showed correct preview but resulted in only one of the Parts being intersected.
- Resolved issue where Booleaning two parts when one is inside a repositioned Assembly resulted in incorrect geometry.
- Fixed issue where the application crashed when redoing the creation of a Boolean Extrude.
- Fixed problem where right-clicking to bring up the context menu for the Scale Bar in the Field View caused the context menu for any geometry under the scale bar to be shown.
- Removed UniGraphics and ParaSolid as potential filetypes to Import or Export since XFdtd does not currently support these formats.
- The Create Sheet Body editor now properly validates when on the “Edit Profile” tab.
- Fixed issue where adding a plot to a Smith chart was not done such that it was undoable.
- Fixed issue where some text in the Application Preferences window was being truncated.
- Resolved issue where the crosshairs in the sketcher did not always follow the mouse.
- Resolved issue where picking tools sometimes did not select the correct geometry when the geometry was oriented and the view was not normal to one of the primary planes.
- Fixed issue where antenna diversity statistics would not be computed when using a Gaussian PDF in certain situations.
- Resolved issue where circular dependency between working coordinate systems of Parts caused Parts to be invalid with the message “Object is updating”, with eventual failure of the application later on. Includes a change that will now not allow the creation of circular dependence of this type between Parts.
- Fixed issue where Axis and Twist description text was lost between closing and reopening the Specify Orientation editor (Advanced Mode) when they were referencing other geometry.
- Materials can now be dropped on imported DXF parts.

- Setting volumetric objects, such as imported VariPose® meshes, invisible and turning on/off their bounding box representations now works as expected.
- Pasting multiple items in the Project tree now pastes them in the same order they were copied rather than in the reverse order.
- Fixed Advanced Convergence Settings to correctly name the Temporal Sampling Interval (which was incorrectly labeled Spatial Sampling Interval).
- Fixed issue where the Specify Position dialog where in some cases parameters were immediately converted to values rather than kept as parameters.

Glossary

Anchor Point

Defined by its location within the reference coordinate system. This location along with the offsets defined by Translations determines where the origin (0,0,0) or the local coordinate system is located 87

Assembly

User-defined groups of geometric objects that are added to the Parts Branch 265

Broadband Calculation

A calculation performed in XFdtd yielding results for a range of frequencies 11

Cell Edge

The edge created when two planes cut a line normal to those planes 323

Circuit Component Definition Editor

Defines the properties associated with discrete components 111

Circuit Components Branch

Organizes discrete circuit components of the active project within the Project Tree 36

Circuit Component

A discrete component which functions as part of a circuit in the geometry 107

Component Tools

Tools used to add discrete components to the XFdtd project 108

Constraints

Restrictions placed on geometric parts that must be satisfied in order to consider the model valid 60

Construction Grid

Controls the spacing and appearance of the visible grid lines in the 2-D sketcher. Changes made with this tool do not affect the properties of the grid (fixed points, grid regions, target cell sizes, etc.) 61

Convergence

A condition met in a broadband calculation when all electromagnetic energy has dissipated to essentially zero 212

Definitions Branch

Stores definitions that can be applied to or shared with other objects within the active project, located in the Project Tree 37

Discrete Source

A voltage or current source whose generated electric field is modified by the addition of an input waveform 108

Error Icon

Indicates when an invalid definition is created within a project 43

Excitation

A source applied to the project defined as a discrete source, incident plane wave, or Gaussian beam 107

External Excitations Branch

Organizes the external excitations applied to the active project within the Project Tree 36

Extrude

A geometry operation used to sweep a face in the normal direction from its center 301

FDTD Branch

Stores definitions associated with the outer boundaries of the active project, including the grid and mesh, within the Project Tree 38

FDTD

Finite-Difference Time-Domain: A method used to calculate elecromagnetic field values in the time domain 5

Far Zone Sensor

A sensor located at theoretical infinite distance from the simulation geometry 193

Far Zone

The region at theoretical infinite distance from the geometry 193

Field Editing Toolbar

Used to configure the properties of the calculation results display 251

Field Reader Tool

Used to identify field values in the calculation results display 249

Field Sequence

A a movie of individual field snapshots for each slice of the geometry over a time interval 245

Field Snapshot

A display of near-zone fields at a specific step in time 245

Fixed Point

A point on an axis at which a plane, in the other two axes, exists 324

Free Space Material

A material with relative permittivities and permeabilities set to one and conductivities set to zero 94

Frequency Range of Interest

Defines the frequency range over which the user is interested in obtaining results 21

Function Library

A library containing function and class definitions that are always available within the scripting engine 267

GUI (Graphical User Interface)

The XFdtd display screen 13

Gaussian Beam Editor

Defines the properties of a Gaussian beam excitation 138

Gaussian Beam

An incident electric field that has a two-dimensional, radially-symmetric Gaussian distribution in planes normal to the incident direction, converging to maximum intensity at the focus point 138

Geometry Tools

Tools used to create and edit various aspects of the project geometry 45

Geometry Workspace Window

The main project viewing area where users add and edit the fundamental elements of a project 41

Global Coordinate System

The default orientation and location of all objects in the project; defined in terms of X, Y and Z 80

Global Triad

The set of orientation vectors representing the Global Coordinate System 80

Graphs Branch

Organizes the graphical output associated with data collected during a calculation within the Project Tree 39

Grid Line

A plane viewed edge-on 324

Grid Region

A region within the global FDTD grid, bounded by planes, with a specified cell size 160

Grid Tools

Tools used to assign general and customizable definitions to the FDTD grid 153

Gridding Properties Editor

Defines grid regions and fixed points associated with an individual object 158

Grid

A block of FDTD cells without applied materials 8

Groups Branch

Organizes the objects of an active project in user-defined and auto-generated groups within the Project Tree 39

Hearing Aid Compatibility (HAC) Sensor

Saves electric and magnetic field data relevant to a hearing aid on a 5cm by 5cm rectangle 192

Libraries Workspace Window

Allows users to create libraries and save collections of project-independent objects grouped by category 42

Library

A collection of objects grouped by category for easy reference 269

Local Coordinate System

Displays adjustments from the Reference Cooordinate System during an editing session; defined in terms of U', V' and W' 80

Macro

A script containing global-scoped code that is only executed on demand 267

Material Editor

Defines materials to be applied to objects in an FDTD simulation 93

Material

A definition applied to the project geometry that defines each part as electric, magnetic or free space 93

Maximum Cell Step Factor

The adjacent cell size ratio that cannot be exceeded within the grid 327

Mesh Object

A subsection of the XFdtd grid including a collection of edges with applied materials 74

Meshing Properties Editor

Assigns the properties of the mesh for individual objects 163

Meshing

The act of assigning materials to each cell edge 323

Mesh

A block of FDTD cells with applied materials 8

Minimum Distance

The minimum distance between fixed points 325

Near Field Sensor

Saves time-domain and/or frequency-domain near zone field quantities at specific points within the calculation space 182

Near Zone

A block of FDTD cells with applied materials 8

Orientation Triad

The set of orientation vectors representing the Local Coordinate System 80

Origin

The center of the Orientation Triad 80

Outer Boundary Editor

Defines the properties of the Outer Boundary 177

Outer Boundary

The outer edge of the XFdtd grid 177

Parameter Sweep

Defines the values used for a particular parameter when the simulation is created. 204

Parameters Workspace Window

Allows users to create, edit and delete parameters that are referenced universally throughout the project 42

Parameter

A global variable or function used to define properties or run parameter sweeps, stored in the Parameters Workspace Window 265

Parts Branch

Organizes the physical geometry of the active project within the Project Tree 34

Parts List

A dialog window used to view, define and organize characteristics of geometric parts 35

Perfect Conductor

An electrical or magnetic conductor with no resistivity, or whose field is defined as zero 8

Plane Wave Editor

Defines the properties of a plane wave excitation 138

Plane Wave

A source assumed to be infinitely far away so that the constant field surfaces are planar and normal to the direction of propagation 138

Port Sensor

Saves near-zone voltage and current data at the location of a circuit component 196

Post-Processing

A computation performed after the calculation engine run is finished which typically saves results to disk 262

Project Properties Editor

Defines the default display units, custom timestep multiplier, and new part modeling units in the active XFdtd project 21

Project Tree

A set of categorized branches used to display and organize the contents of the active XFdtd project 20

QtScript

The language used to write Scripts in XFdtd 268

Reference Coordinate System

The orientation and location of objects within their native assembly; defined in terms of U, V and W 80

Results Workspace Window

Stores all of the results available for a selected project 41

Revolve

A geometry operation used to sweep a profile in a circular path 301

Scale Bar

Displays colors to distinguish between the ranges within the calculated result values 248

Scripting Workspace Window

Allows users to create, edit, manage and execute user-defined scripts 42

Scripts Branch

Stores user-defined scripts within the Project Tree 38

Script

A block of QtScript used to quickly automate tasks that could otherwise be done through the XFdtd GUI 267

Sensor Tools

Tools used to define sensors and the data they collect 182

Sensors Branch

Organizes the sensors defined in an active project within the Project Tree 37

Sensors

Objects that save data during a simulation 181

Shortcut Group

Groups of objects that are added and organized by the user under the Groups Branch 265

Simulations Workspace Window

The main interface used to define simulations to send to the calculation engine 41

Single-Frequency Calculation

A calculation performed in XFdtd yielding results for a single frequency 11

Snapping Tool

Facilitates the exact placement of vertices on the sketching plane 60

Specific Absorption Rate (SAR) Sensor

Used to measure the interaction of electromagnetic fields with human tissue 196

Static Voltage Point Branch

Organizes the static voltage points defined in an active project within the Project Tree 37

Steady State

The point in time when near-zone calculation results show that all transients have died down and the only variation left is sinusoidal 214

Steady-State Calculation

See Single-Frequency Calculation 11

System Sensor

Automatically saves simulation data not applicable to other sensors 195

Target Cell Size

The maximum cell size within a grid region 327

Timestep

The amount of time required for the electromagnetic field to travel from one cell to the next 6

Transient Calculation

See Broadband Calculation 11

Transition Region

A region between a grid region and the global grid, containing the fewest number of cells required to progressively reach the desired cell size 327

View Tools

Tools used to manipulate the view of the simulation space 45

Voxel

A volumetric pixel point in space with specific volume characteristics 66

Waveform Editor

Used to edit waveforms to inject energy into the space for an FDTD simulation. 140

Waveform

Determines the time variation of an excitation; defined as a pulse, a sinusoidal source, or user-defined 140

Waveguide Tools

Tools used to add waveguides to the XFdtd project 119

XFdtd

A three-dimensional full-wave electromagnetic solver based on the Finite-Difference Time-Domain method 1

XStream

A hardware acceleration option available for XFdtd in the form of a graphics card 224

Yee Cell

The cell used as the basis for FDTD calculations 6

XFsolver

A calculation engine, separate from the XFdtd GUI, which performs an FDTD calculation 201

Bibliography

[1] K. S. Kunz and R. J. Luebbers, *"The Finite Difference Time Domain Method for Electromagnetics"*. Upper Saddle River: CRC Press, 1993.

[2] A. Taflove and S. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method, Third Edition*. New York: Artech House Publishers, 2005.

[3] C95.3-2002, "Recommended practice for measurements and computations of radio frequency electromagnetic fields with respect to human exposure to such fields, 100khz to 300ghz," *IEEE Standards and Coordinating Committee 28 on Non-Ionizing Radiation Hazards*, pp. i–126, April 2002.

[4] K. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation 14*, pp. 302–307, 1966.

[5] C. Clement, ed., *Annals of the ICRP, ICRP Publication 110, Adult Reference Computational Phantoms*. Elsevier Ltd, 2009. Volume 39, Number 2. Accompanied by ICRP CD 4. ISSN 0146-5453. ISBN 978-0-7020-4186-0.

[6] http://wmc.nict.go.jp/bio/model/index_e.html.

[7] http://www.itis.ethz.ch/services/anatomical-models/virtual-population/.

[8] P. Hasgall, E. Neufled, M. Gosselin, A. Linkgenböck, and N. Kuster, "IT'IS Database for thermal and electromagnetic parameters of biological tissues." http://www.itis.ethz.ch/database, September 2011.

[9] C. Balanis, *Advanced Engineering Electromagnetics*. New York: Wiley, 1989. Section 2.8.3.

[10] S. Ramo, J. Whinnery, and T. Van Duzer, *Fields and Waves in Communication Electronics*. Wiley, 1994.

[11] R. E. Collin, *Foundations for Microwave Engineering*. New York: Wiley-IEEE P, 2000. Section 6.7.

[12] B. Lax and K. J. Button, *Microwave Ferrites and Ferrimagnetics*. McGraw-Hill, 1962. Sections 4.1, 4.2.

[13] F. Kung and H. T. Chuah, "A finite-difference time-domain (fdtd) software for simulation of printed circuit board (pcb) assembly," *Progress in Electromagnetic Research, PIER 50*, vol. Elsevier book series, pp. 299–335, 2005.

[14] J. D. Jackson, *Classical Electrodynamics*. Wiley, 1999.

[15] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York: John Wiley and Sons, 1989.

[16] D. F. Williams and R. B. Marks, “A general waveguide circuit theory,” *J. Res. Natl. Inst. Stand. Technol*, vol. 97, pp. 533–562, 1992.

[17] D. F. Williams and B. K. Alpert, “Causality and waveguide circuit theory,” *IEEE Tr Microwave Theory and Techniques*, vol. 49, pp. 615–623, 2001.

[18] K.-L. Wong, Y.-W. Chi, and S.-Y. Tu, “Internal multiband printed folded slot antenna for mobile phone application,” *Microwave and Optical Tech. Lett.*, vol. 49, pp. 1833–1837, 2007.

Index

Remcom provides innovative electromagnetic simulation software and consulting services. Our products simplify the analysis of complex EM problems and lead the market in FDTD-based modeling and simulation. We offer accessible and responsive support provided by a staff of experts and comprehensive training programs.

XF	**XFdtd**®: 3D electromagnetic simulation solver that provides engineers with powerful and innovative tools for modeling and EM field simulation.
XG	**XGtd**®: A high frequency GTD/UTD based package for the design and analysis of antenna systems on complex objects such as vehicles and aircraft.
WI	**Wireless InSite**®: A radio propagation analysis package for analyzing the impact of the physical environment on the performance of wireless communication systems.
VP	**VariPose**®: A geometric modeling package for the manipulation and refining of high-resolution human mesh models for the medical and biomedical markets.
RL	**Rotman Lens Designer**®: A tool for the design, synthesis, and analysis of Rotman Lenses.

- **Remcom Professional Support: www.remcom.com/support**
  One year of Remcom Professional Support (RPS) is included with each product you purchase. Maintaining your RPS license ensures that you will always have access to technical support and free upgrades. To contact Remcom Support, call 814.861.1299. Or, enter your support request online via our convenient customer support portal. Please visit **www.remcom.com/support** to get started.

- **Remcom Discussion Forum: www.remcom.biz/forum**
  Remcom's Discussion Forum allows you to engage with other EM Simulation professionals and Remcom's own experts. You may view discussions as a guest, or join the community to participate and post your own comments. Registration is fast and free! Visit the forum at **www.remcom.biz/forum** or **www.remcom.com/discussion-forum**

Visit **www.remcom.com** for more information

Remcom, Inc.
315 S. Allen St., Suite 416
State College, PA 16801 USA

+1.888.7.REMCOM (US/CAN)
+1.814.861.1299 phone
+1.814.861.1308 fax