EMPro 2011.01
January 2011
EMPro FDTD Simulation

Acknowledgments
Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Java™ is a U.S. trademark of Sun Microsystems, Inc. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC.

The following third-party libraries are used by the NlogN Momentum solver:

"This program includes Metis 4.0, Copyright © 1998, Regents of the University of Minnesota", http://www.cs.umn.edu/~metis , METIS was written by George Karypis (karypis@cs.umn.edu).

Intel@ Math Kernel Library, http://www.intel.com/software/products/mkl

SuperLU_MT version 2.0 - Copyright © 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy). All rights reserved. SuperLU Disclaimer: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

EMPro FDTD Simulation Overview

EMPro enables you to customize and organize projects by providing various features such as scripting and parameterization. These features allow you to quickly and efficiently create or modify projects without using the General User Interface (GUI). The following sections provide a brief description of the major components of creating a simulation.

Constructing the Geometry

EMPro uses the **Feature Based Modeling** concept to create geometries. Using this concept, geometric objects are created as a set of repeatable actions so that operations can be undone and redone quickly without requiring excessive memory. Modeling of objects begins with a simple 2-D cross-section that can modified as per your requirements. For projects that require common geometries, you can create *DefaultProjectTemplate* templates of geometric objects or export them to Libraries to make it easy to import them into new projects. You can also import the CAD files from third-party solid modeling packages.

After building or importing the geometrical objects, you can assign materials to them by creating the material definition objects and applying them by using the drag and drop method.

You can also add discrete Circuit components to the geometry. In previous versions, circuit components were defined in terms of their placement in the mesh, but this method has been revised so that their location (as well as all other physical objects) is defined in terms of their global position in the simulation space. This eliminates the chance that the location of circuit component is altered during meshing, when cells tend to shift.

Defining the Grid and Creating the Mesh

After you have created the geometry objects and applied the material, the grid can be initialized within the Grid Tools interface. While choosing an appropriate cell size for the grid, consider the following factors:

- **Wavelength**: The primary constraint on cell size is wavelength. A cell cannot be larger than 1/10 of the smallest wavelength used to excite the model. Therefore, the maximum cell size can be determined from:

  $$L_{\max} = \frac{c}{10 * f}$$

  Where:
  - $L_{\max}$ specifies the maximum cell dimension
  - c specifies the speed of light, 3x10 ^{8} m/s in free space
  - f specifies the frequency of excitation (Hz)

> **Note:**
> If materials other than good conductors are included in the calculation, the velocity of light will be reduced in those materials and the cell size must be reduced accordingly.

- **Geometry features**: A cell cannot be larger than the smallest feature of your geometry. For example, if the distance between two wires in the geometry is smaller than the maximum cell size, a smaller cell size is needed.
- **Accuracy**: Smaller cell sizes result in greater accuracy in the simulation.

After initializing the grid, you can create a **Mesh** for the project, and then run calculations.

Defining Run Parameters

To run a calculation, it is important to configure the required parameters. You can specify the following run parameters:

- **Circuit Components Definition**: are automatically added to the project as soon as a new Circuit Component is added. The Circuit Component Definition Editor is used to make modifications to this definition.

- **External Excitations**: are added with the **External Excitation Editor**. The source type, whether a discrete source or an external excitation, is set in the Simulations workspace window prior to running the calculation.

- **Waveforms**: are created or edited within the **Waveform Editor**. If a discrete circuit component is already added to the project, a default waveform is automatically added to the project.

- **Outer Boundaries**: are defined within the **Outer Boundary Editor**. Defining the characteristics of the outer boundary enables the calculation engine to provide accurate results.

Requesting Results

Results are collected and stored with objects called **Sensors**. Different types of sensors are available depending on the type of data to be collected.

Running a Simulation

Simulations can be easily created, defined, and stored in the **Simulations** workspace window. Any number of simulations may be queued at one time in this window. They will run one at a time until all simulations are finished calculating. This workspace window is superior to past releases since run parameters are manipulated within one common place so that multiple simulations can be queued without revisiting many different parts of the GUI. Specifications such as Source Type, Parameter Sweeps, S-Parameter Calculations, Frequencies of Interest, Total/Scattered Field Interfaces, and Termination Criteria are defined during this step.

Viewing Output

After running the calculation, view the results from the **Results** workspace window. Some results are displayed in the form of numerical values, while other results are 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.

Other Tools

Several optional tools available that can be used during the simulation creation process.

Scripting

Scripting enables you to customize the GUI to perform any task. Scripts are used to automate tasks that are repetitive or can be done through the GUI with speed and precision.

Parameterization

Parameters are global variables that are defined and stored in one common workspace window and can be referenced anywhere in the interface. Furthermore, they can be used to perform a Parameter Sweep, which is a new feature that increments a specific parameter and perform a calculation at every iteration.

Libraries

Libraries are essentially databases of project definitions that are saved so that they can be used multiple times in subsequent projects. Libraries are useful for users that create multiple similar projects.

Using the Finite-Difference Time-Domain Method

In this section, you will learn about:

- Benefits of using FDTD to perform your electromagnetic simulation.
- Factors to be considered for setting up an electromagnetic calculation in EMPro.

This section 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 great popularity in recent years with the increased performance from computers.

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

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.

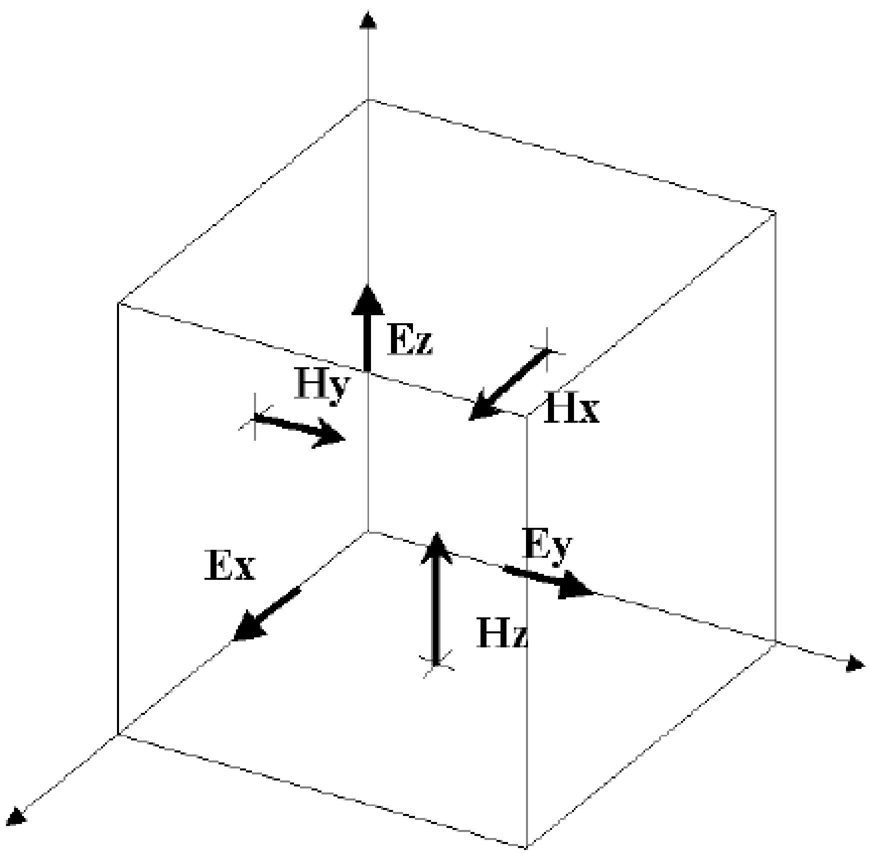
> **Note**
> For more information on field penetration into biological tissues, refer to IEEE publication *C95.3 Recommended Practice for Measurements and Computations with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 100 kHz to 300 GHz*, and the Toflove and Hagness text.

FDTD Overview

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 located on the edges of the box and the magnetic fields are positioned on the faces as shown in the figure below. This orientation of the fields is known as the Yee cell, and is the basis for FDTD.

> **Note**
> For a description of the Yee cell, refer to IEEE publication *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media*.

Yee cell with labeled field components



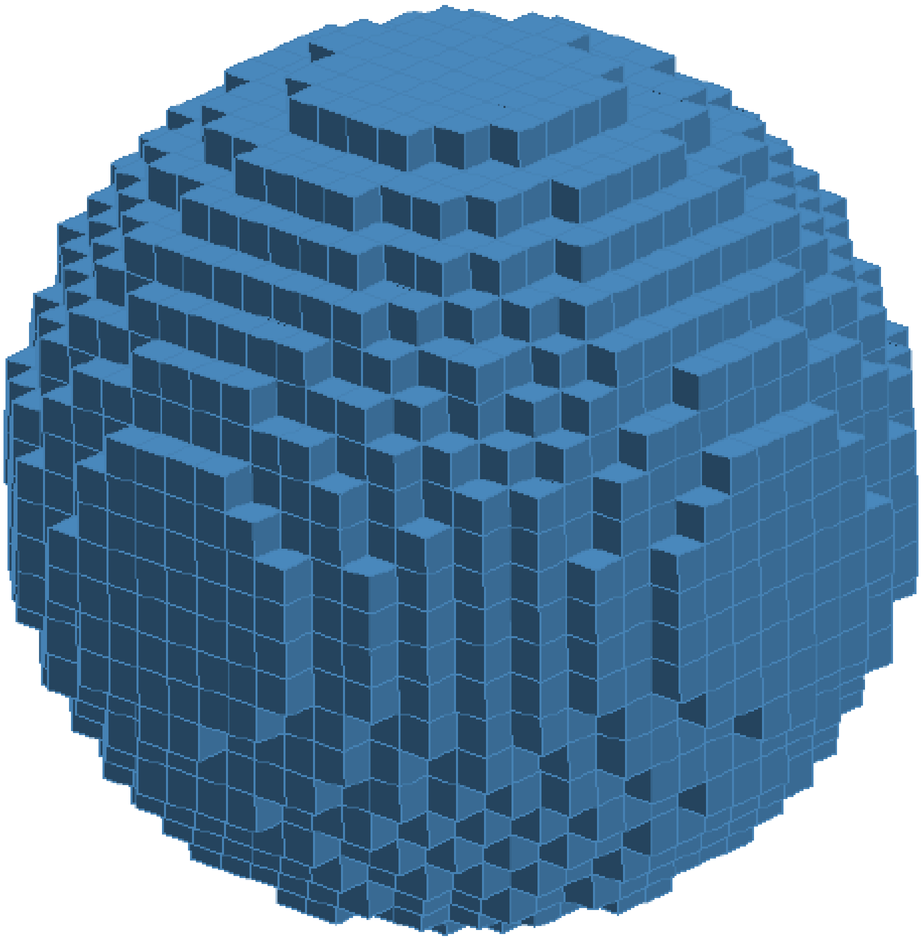
Time is quantized into small steps where each step represents the time required for the field to travel from one cell to the next. Given the offset in space of the magnetic fields from the electric fields, the values of the field with respect to time are also offset. The electric and magnetic fields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

> **Note**
> For a description of how the timestep is calculated, refer to Computer Resources.

When many FDTD cells are combined together to form a three-dimensional volume, the result is an FDTD grid or mesh. 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 the illustration above.

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 identically 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 below. Each small box shown in the figure represents one FDTD cell.

A dielectric sphere as meshed in an FDTD grid



The individual cell edges (electric field locations) in the above image 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.

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. Of course 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 EMPro, 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 EMPro 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.

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.

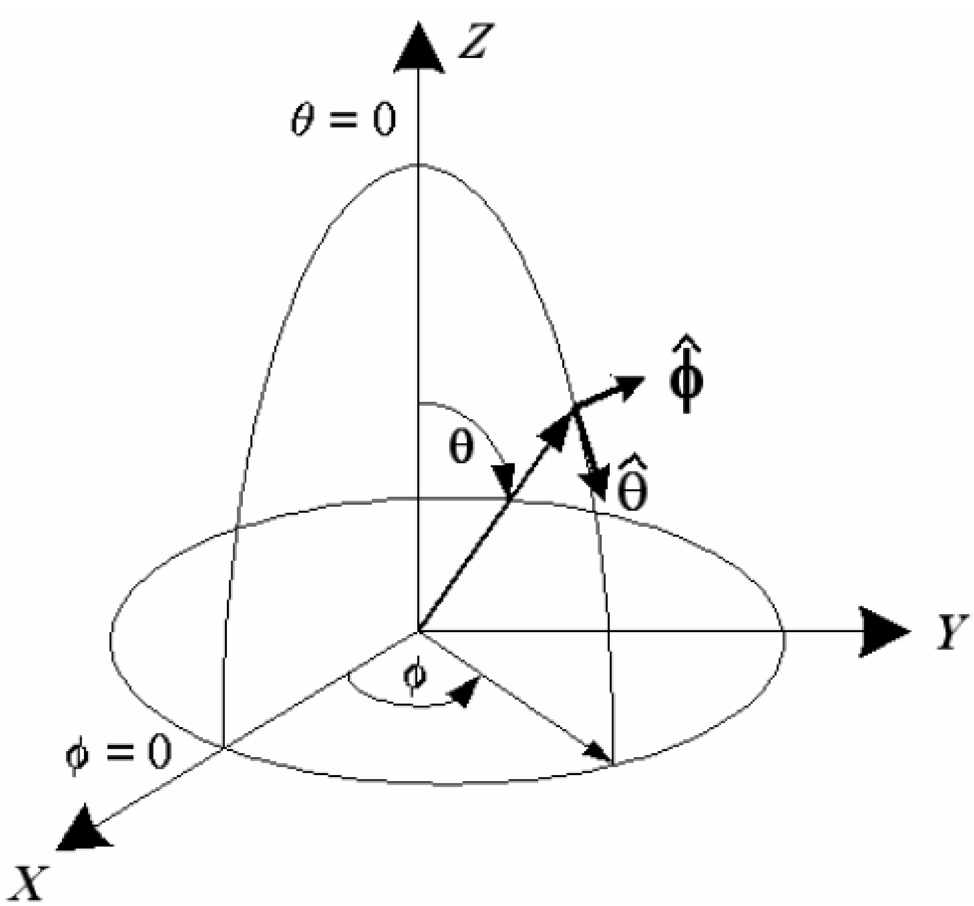
This three-dimensional volume is considered to be the near-zone region in EMPro in terms of the data storage. The field value at any edge in the FDTD grid may be observed as a function of time by saving a near-zone point in EMPro. Other types of data such as steady-state field magnitudes, specific absorption rates, S-parameters, or impedance may be stored as well 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.

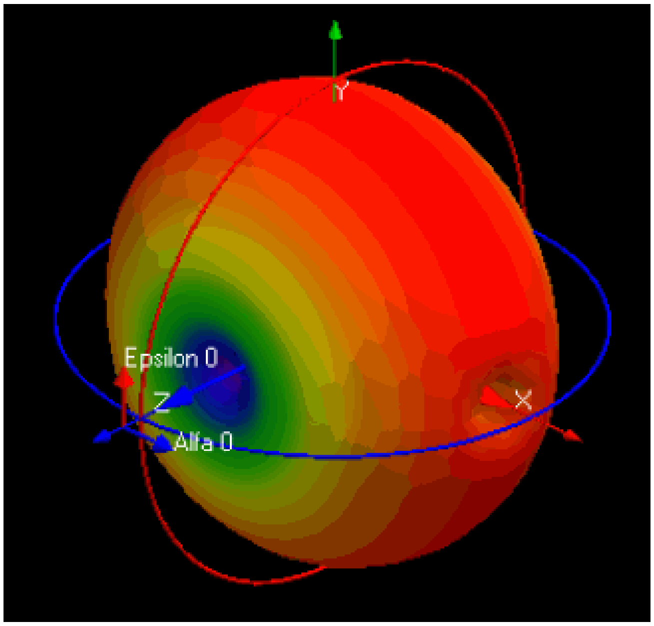
A more practical method for transforming field values to the far-zone and for calculations of radiation gain or radar scattering patterns is to use a transformation to convert the near-zone values in the FDTD grid into a far-field value at some location away from the grid. In EMPro, this is done by enclosing the geometry in a box and storing the fields on the six faces of this box. 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 EMPro geometry must be contained within the box.

The coordinate system used in EMPro 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 the figure below. This coordinate system is used for locating far-zone positions and for defining the incident plane wave direction in EMPro.

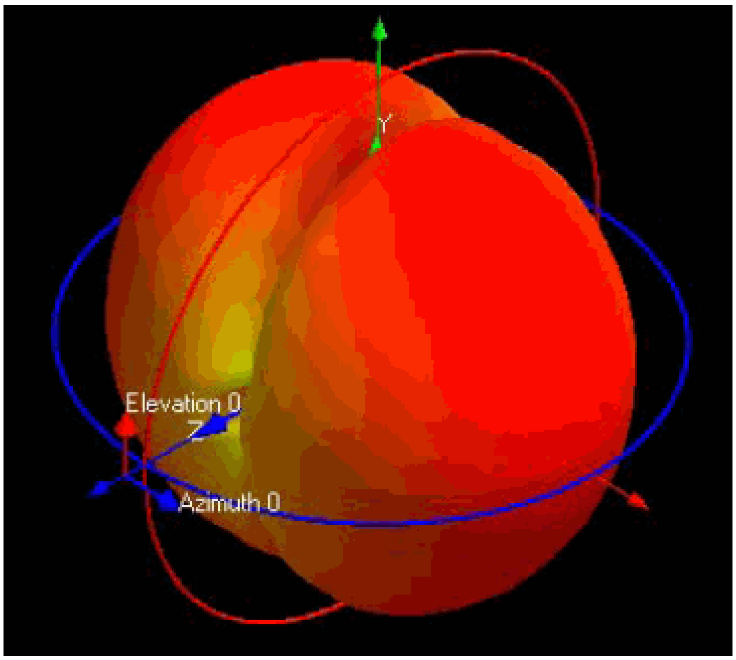
Coordinate System used in EMPro for far-zone and incident plane wave directions



Coordinate System for Alpha, Epsilon far-zone patterns



Coordinate System for Azimuth, Elevation far-zone patterns



Broadband and Steady-State Calculations

EMPro 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 at 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.

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

Outer Radiation Boundaries

A three-dimensional grid of cells forms the EMPro 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 EMPro grid.

The outer radiation boundary is a method for absorbing fields propagating from the EMPro 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 EMPro 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 EMPro calculation.

Computer Resources

> **Note**
> The EMPro 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.

> **Note**
> 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: transient 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 EMPro calculation is more complicated since computer processor performance varies.

> **Note**
> 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 EMPro 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
> Δx, Δy and Δz are the lengths of the cell sides, in meters.

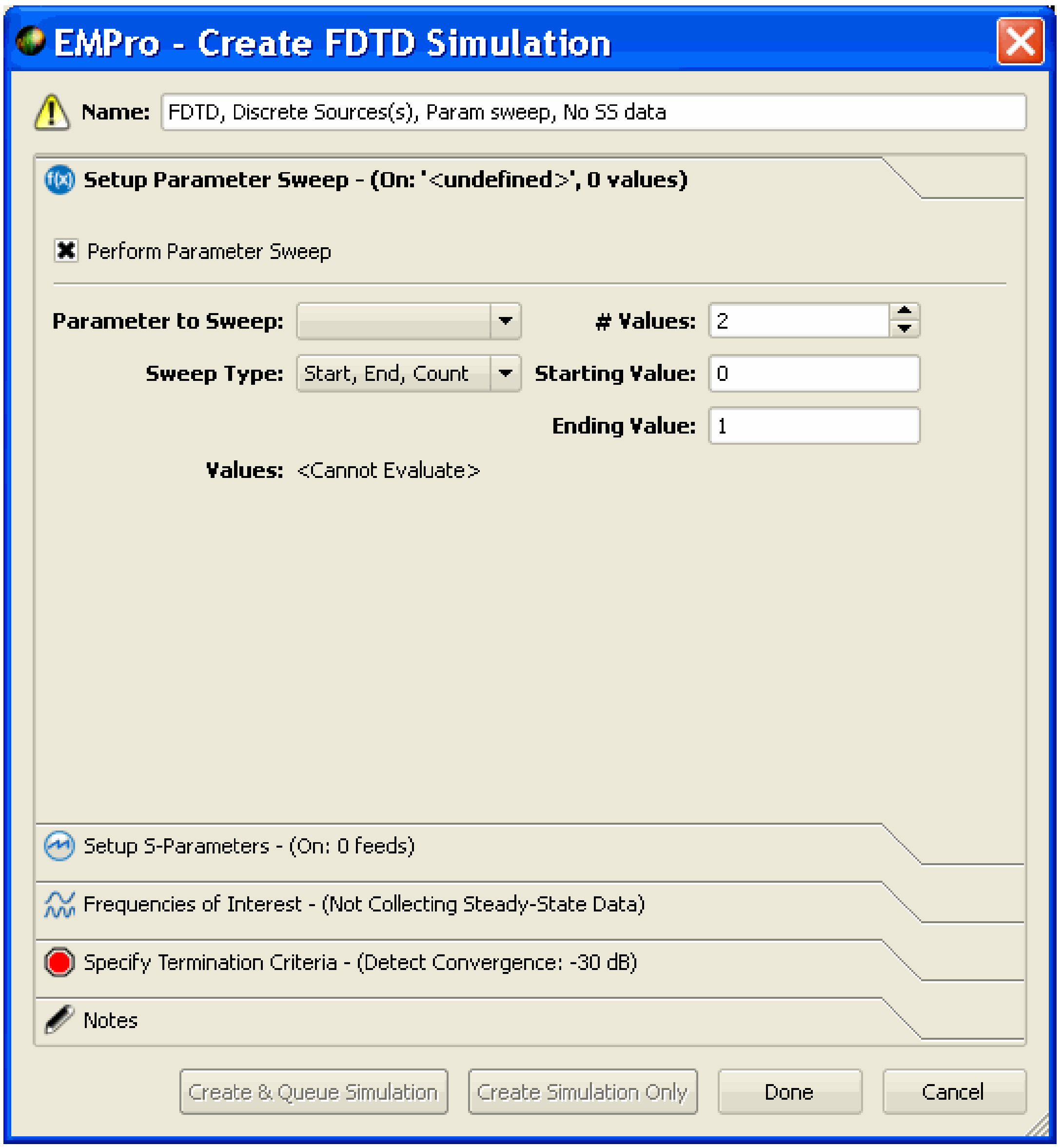
The timestep used for computation must be no longer than the smallest timestep limit for each of the cells in the problem space.

Specifying FDTD Simulation Setup in EMPro

You can specify the simulation options that are specific to the FDTD Simulator in the *Create FDTD Simulation* dialog box. This dialog box enables you to control how the FDTD simulation mesh is generated. In the *Simulations* workspace window, click the **New FDTD Simulation** button.

Specifying the Parameter Sweep Values

The *Parameter Sweep Values* screen enables you to specify the sweep parameter, sweep type, and values, as shown in the following figure:

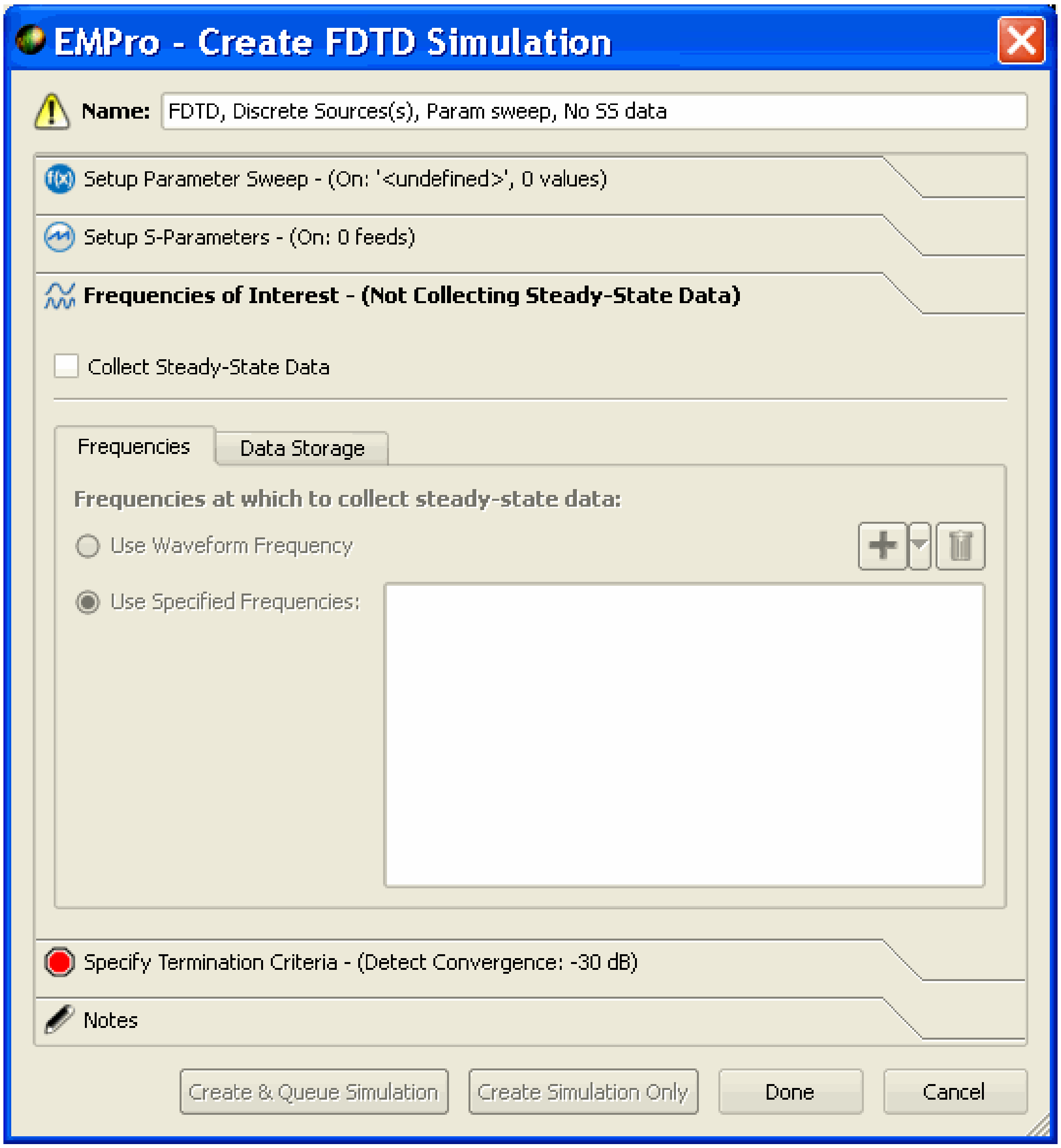


Select the Perform parameter Sweep to enable the options available on the screen:

Option	Description
Parameter to sweep	Specify the required parameter
#Values	Specify the value
Sweep type	Select the required sweep type
Starting value	Specify the start value for the parameter
Ending Value	Specify the end value for the parameter

Specifying Frequency Plans

Using the *Create FDTD Simulation* dialog box, you can specify the frequency settings for your FDTD simulation.



In the *Frequencies of Interest* screen, select **Collect Steady-State data**. This screen consists of the following tabs:

Frequencies

Specify the options listed in the following table:

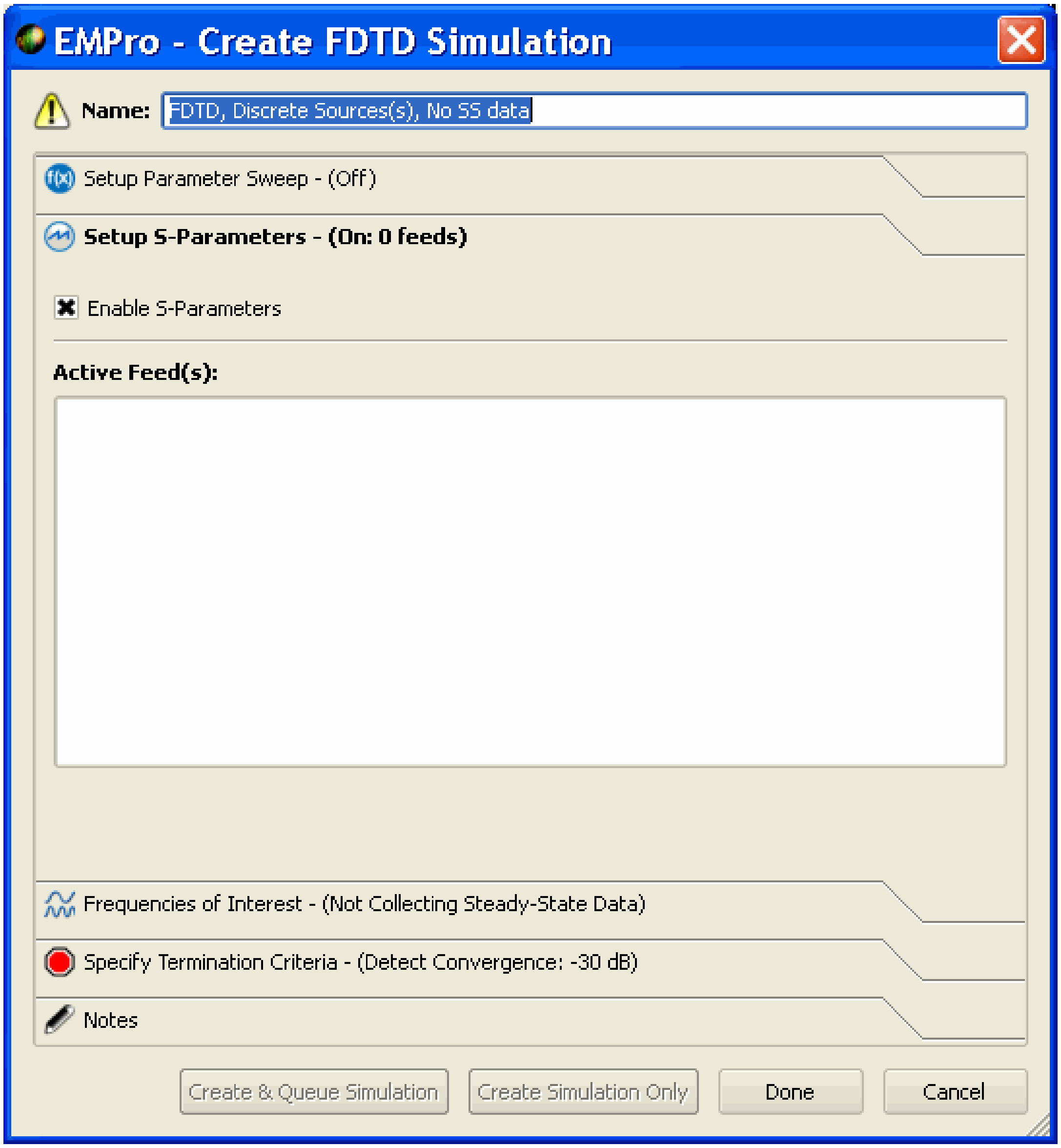
Option	Description
Store Data	Allows you to store data in memory or in the disk.
Compute Dissipated Power	Allows you to calculate the dissipated power.

Data Storage

Option	Description
Save data for post-simulation far zone steady-state processing	Allows you to store data in memory or in the disk.
Normalize fields	Allows you to normalize fields.
Timestep sampling	Allows you to specify the frequency to resolve and sampling interval.

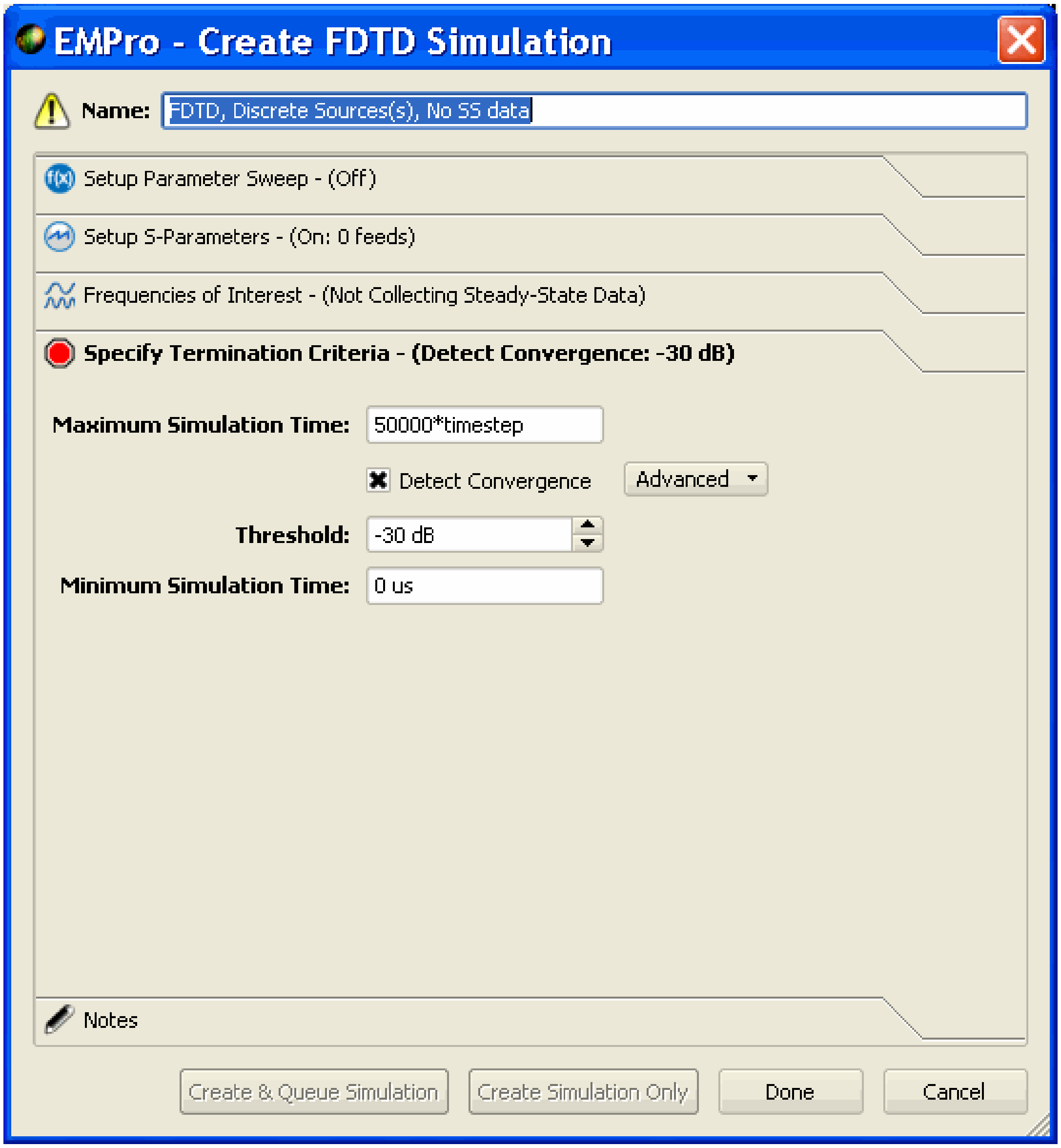
Specifying the S-parameters Settings

Using the *Create FDTD Simulation* dialog box, you can specify the s-parameter settings. Click **Setup S-Parameters** to display the *Setup S-Parameters- (On:0 feeds)* screen, as shown in the following figure:



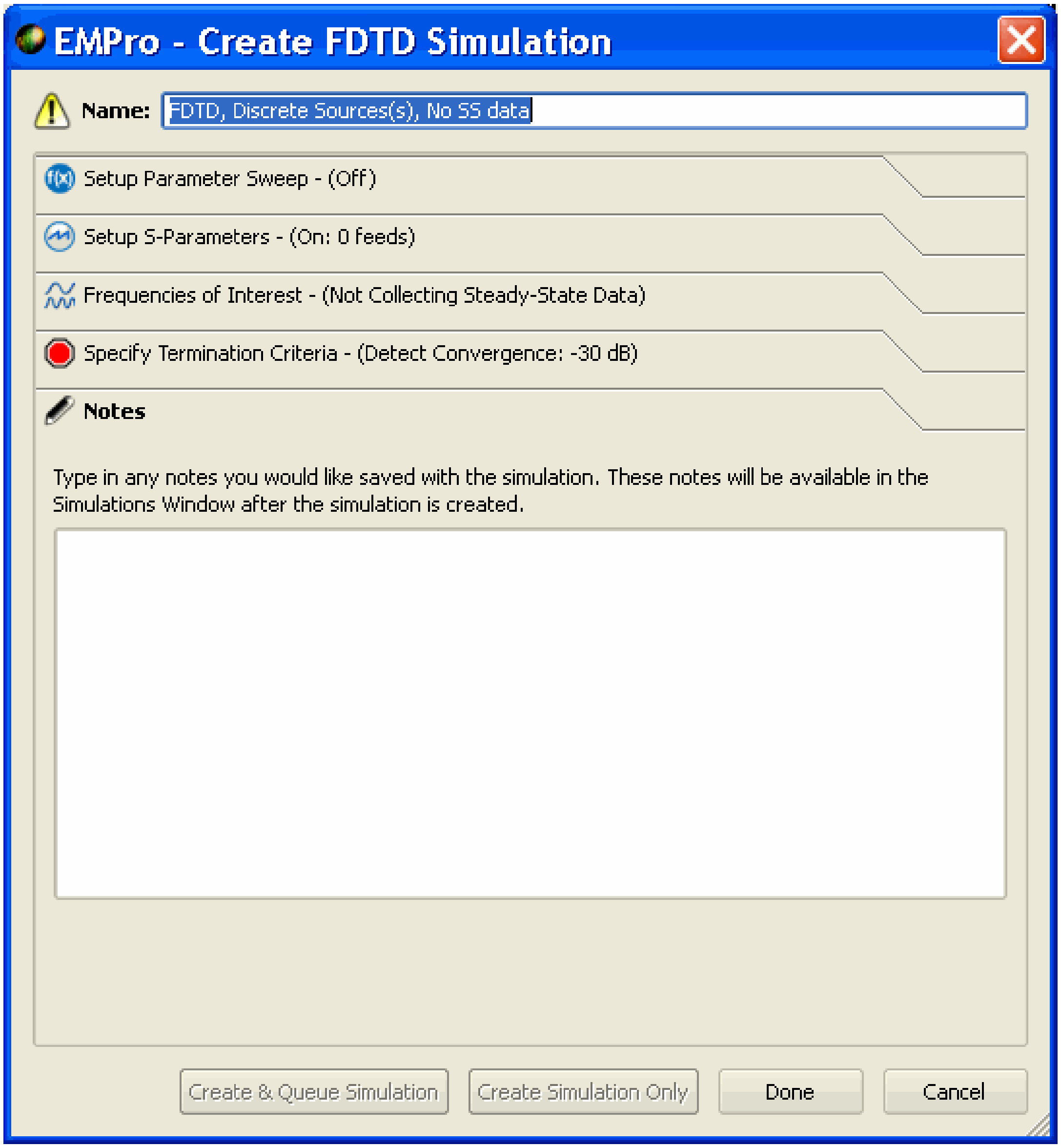
In the *Setup S-Parameters- (On:0 feeds)* screen, select **Enable S-Parameters**. In the **Active Feeds** box, you can specify the required feeds.

Specifying the Termination Criteria



Using the Notes Section

If you want to add any notes or observation with your simulation, you can specify it in the Notes text box. Click **Notes** in the **Create FDTD Simulation** dialog box to display the **Notes** screen, as shown in the following figure:



After you have completed entering your FDTD Simulation options, click the **Done** button to apply the current settings in the *Create FDTD Simulation* dialog box, or click **Create Simulation Only** to accept the settings. You can also click **Create and Queue Simulation**, alternatively, you can click the **Cancel** button to abort the changes and dismiss the dialog box.

Creating a Low Pass Filter Simulation

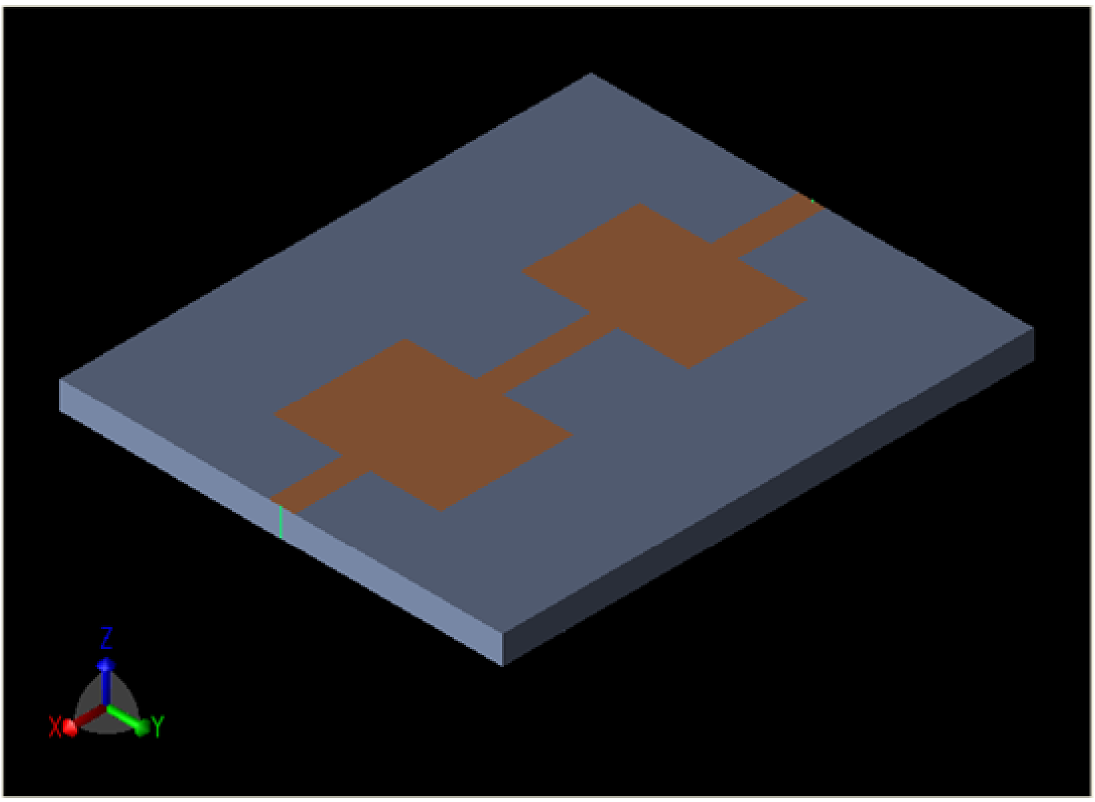
In this project, you will learn how to:

- Set the material properties of a low pass filter and create the geometry through scripts.
- Define the properties of the low pass filter environment.
- Add a feed and a load to the filter and simulate their effects.
- Retrieve port sensor and planar surface sensor data after running the calculation.

Getting Started

This section briefly describes how to set the display units for the low pass filter project.

Low Pass Filter



> **Note**
> To set up a project for the first time, refer to Application Preferences Appendix for instructions about how to configure project preferences and navigate through the display units tab.

In the *Project Properties Editor* window, navigate to the **Display Units** tab:

1. Select **SI Metric** in the Unit Set drop-down list.
2. Change **Length** to millimeters (mm). This changes the Unit Set value to **Custom**.
3. Click **Done**.

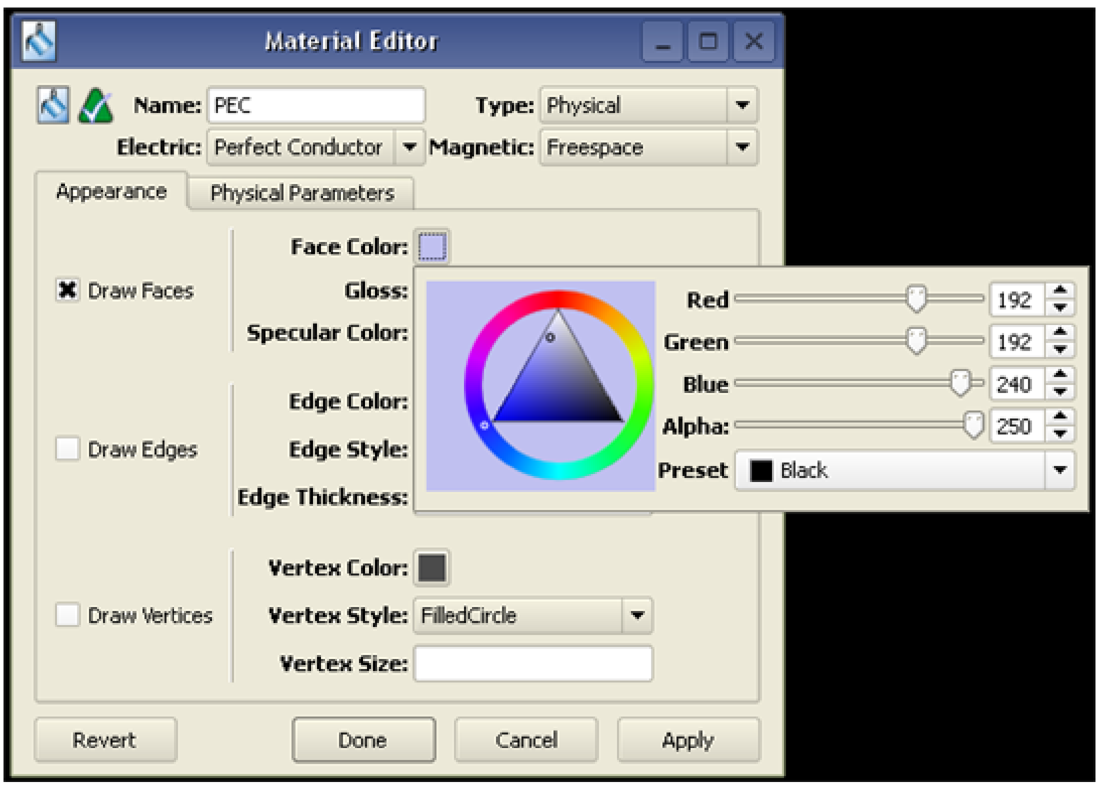
Creating Materials

For this example, you need to create material definitions before creating the geometry so that the script that is executed to build the Substrate block can access the material definitions. The low pass filter will consist of Perfect Electric Conductor and Substrate.

Defining Material (PEC)

1. Right-click **Definitions:Materials** branch of the project tree and select **New Material Definition**.
2. Double-click the new material to edit its properties. Specify the following properties for the electric conductor material:
    - **Name**: PEC
    - **Electric**: Perfect Conductor
    - **Magnetic**: Freespace
    Depending on your requirements, you can also set the display color of the PEC material in the **Appearance** tab, as shown in the following figure.

Editing the color of the PEC material

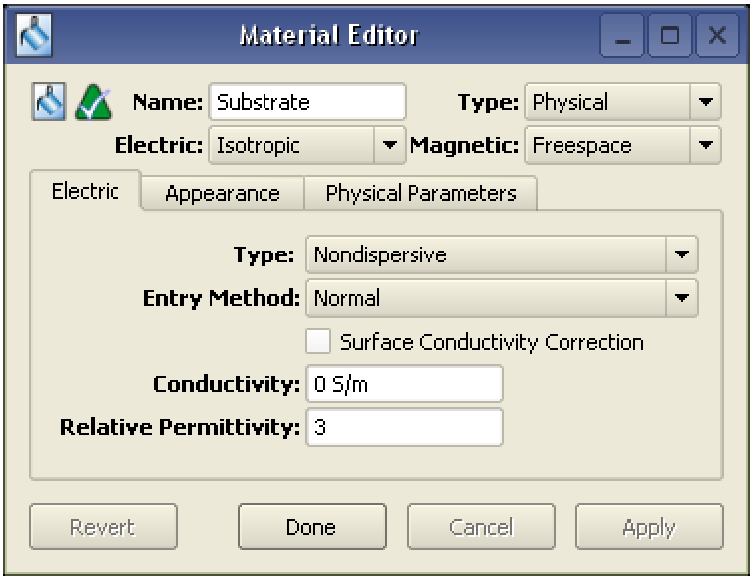


Defining Material, Substrate

1. Right-click the **Definitions:Materials** branch of the Project Tree and select **New Material Definition**.
2. Double-click the new material to edit its properties. Set the substrate material properties as follows:
    - **Name**: Substrate

- **Electric**: Isotropic
- **Magnetic**: Freespace
- Under the Electric tab:
- **Type**: Nondispersive
- **Entry Method**: Normal
- **Conductivity**: 0 S/m
- **Relative Permittivity**: 3

Defining the properties of the Substrate material



3. In the **Appearance** tab, assign a new color to this material to distinguish it from the first material, PEC.
4. Click **Done** to add the new material, Substrate.

Creating the Low Pass Filter Geometry

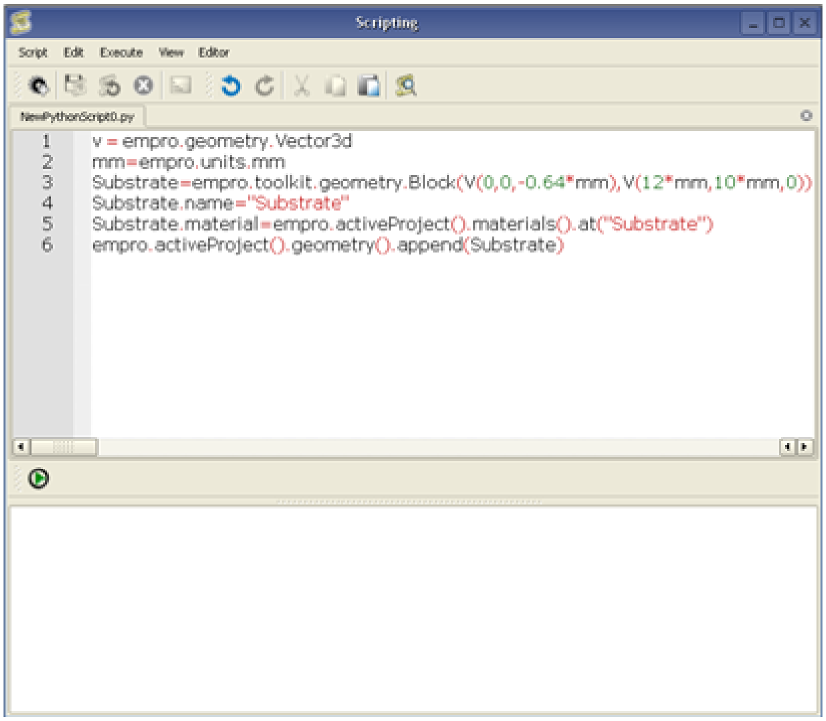
Modeling the Substrate

The Substrate block geometry for the low pass filter is a simple rectangular block. For this example, you will use a script to prompt an interface where we can create a rectangular block with an applied material.

1. Right-click the **Scripts** branch of the Project Tree and select **New Python Script**.
2. This automatically adds a **New Python Script** object to the branch. Right-click the object, select **Rename**, and type **Rectangular_Block**.
3. Copy the following script into the *Scripting* workspace window.

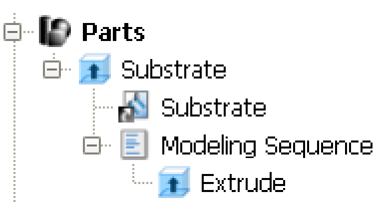
> **Note**
> This script can also be found in the python_scripts/demo/miscellaneous folder of your installation directory, depending on the location where you have installed on your computer.

The substrate script in the Scripting workspace window



4. Choose **Script > Commit** to commit your new script to the project.
5. Click **Execute Python Script** to run the script which will add the Substrate block to the project.
6. Drag-and-drop the object material **Substrate** under Materials on the Substrate block to assign a material to this object.

Substrate block in the Project Tree



Modeling the Strip Line

The Strip Line will be modeled as a polygon Sheet Body. To simplify this operation, you will use another script to add the Strip Line to the project.

1. Right-click the **Scripts** branch of the Project Tree and select **New Python Script**.
2. Right-click the **New Python Script** object, select **Rename**, and type **Strip_Line**.
3. Copy the following script in the *Scripting* workspace window.

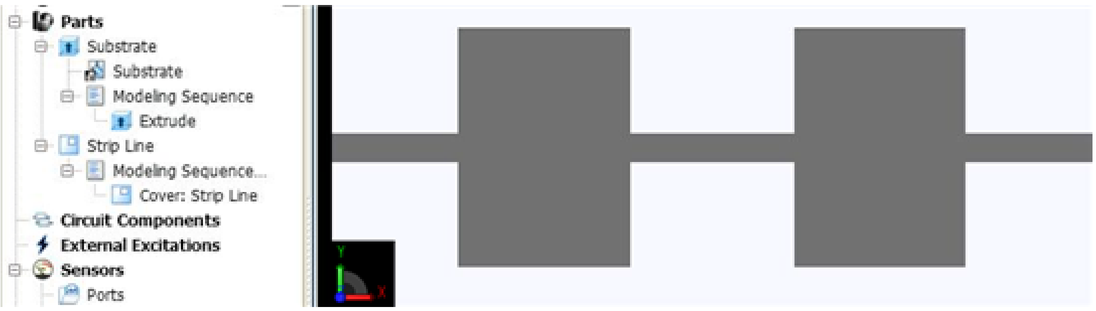
```
V = empro.geometry.Vector3d
E = empro.core.Expression
SCALED_UNIT = E("1 mm")
V1= V(0, 4.77, 0) * SCALED_UNIT; V2=V(0, 5.23, 0)*SCALED_UNIT; V3=V(2, 5.23, 0)*SCALED_UNIT;
V4=V(2, 6.9, 0)*SCALED_UNIT;
V5=V(4.7, 6.9, 0) * SCALED_UNIT; V6=V(4.7, 5.23, 0) * SCALED_UNIT; V7=V(7.3, 5.23, 0) *
SCALED_UNIT; V8= V(7.3, 6.9, 0) * SCALED_UNIT;
V9=V(10, 6.9, 0) * SCALED_UNIT; V10=V(10, 5.23, 0) * SCALED_UNIT; V11=V(12, 5.23, 0) *
```

```
SCALED_UNIT; V12=V(12, 4.77, 0) * SCALED_UNIT;
V13=V(10, 4.77,0) *SCALED_UNIT; V14=V(10, 3.1,0) *SCALED_UNIT; V15=V(7.3, 3.1,0)
*SCALED_UNIT;V16=V(7.3, 4.77,0) *SCALED_UNIT;
V17=V(4.7, 4.77,0) *SCALED_UNIT; V18=V(4.7, 3.1,0) *SCALED_UNIT; V19=V(2, 3.1,0)
*SCALED_UNIT; V20=V(2, 4.77,0) *SCALED_UNIT;
vertices=[
V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,V11,
V12,V13,V14,V15,V16,V17,V18,V19,V20
]
StripLine=empro.toolkit.geometry.PolyPlate(vertices,"Strip Line")
empro.activeProject().geometry().append(StripLine)
```

Note
This script can also be found in the python_scripts/demo/miscellaneous folder of your installation directory, depending on the location where you have installed on your computer.

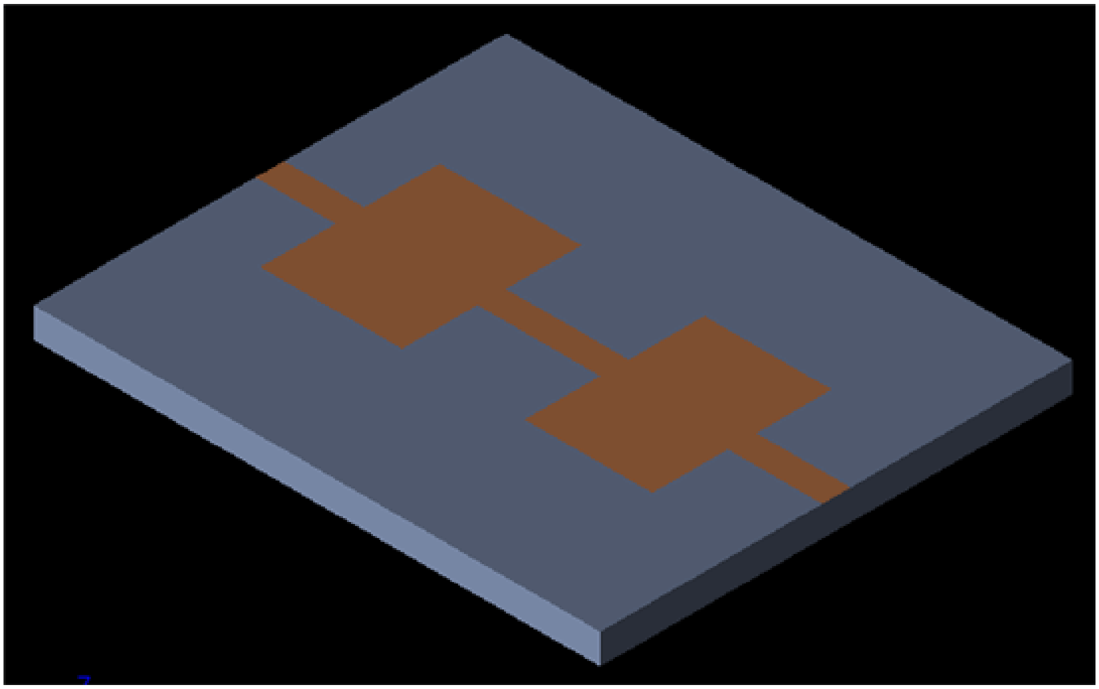
Choose **Script > Commit** to commit your new script to the project.
Click **Execute Python Script** to run the script and create the Strip Line object.

Strip Line geometry created from script



Drag-and-drop the material object PEC onto the object Strip Line to assign a material to this object.
The completed low pass filter geometry will appear in the *Geometry* workspace window.

Low pass filter geometry

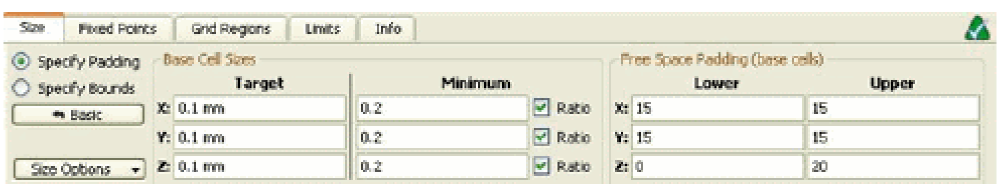


Creating the Grid

Now you will define characteristics of the cells in preparation to perform an accurate calculation.

1. Double-click the **FDTD:Grid** branch of the Project Tree to open **Grid Tools**.
2. Set the **Size** properties of the grid as given below:
    - **Base Cell Sizes**: Target 0.1 mm, Merge 0.2, Ratio boxes checked
    - **Free Space Padding**:
        - 15 for Lower X, Lower Y, Upper X, Upper Y
        - 0 for Lower Z
        - 20 for Upper Z

Defining cell size and free space padding within Grid Tools

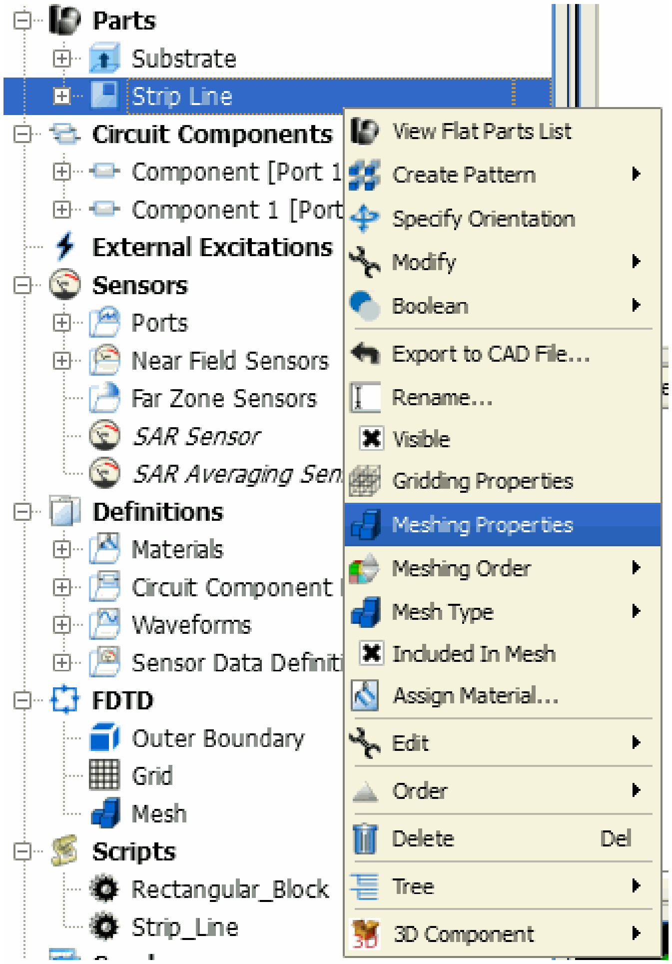


3. Click **Done** to apply the grid settings.

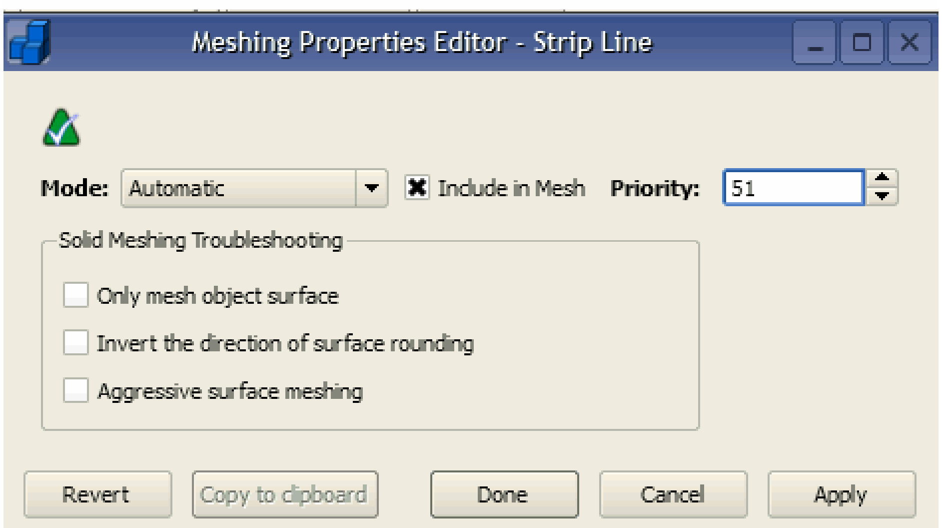
Setting Mesh Priority

For the overlapped geometry regions, you need to define the mesh with one material. Mesh priority also need to be defined because in overlap section the mesh material is decided by mesh priority. In this structure region, stripline is collocated with top plane of substrate as both are at the same z height. In stripline location, you want mesh of the material of this object, that is, PEC, hence mesh priority of stripline should be more that substrate.

1. Right-click **Stripline** and choose **Meshing Properties**.

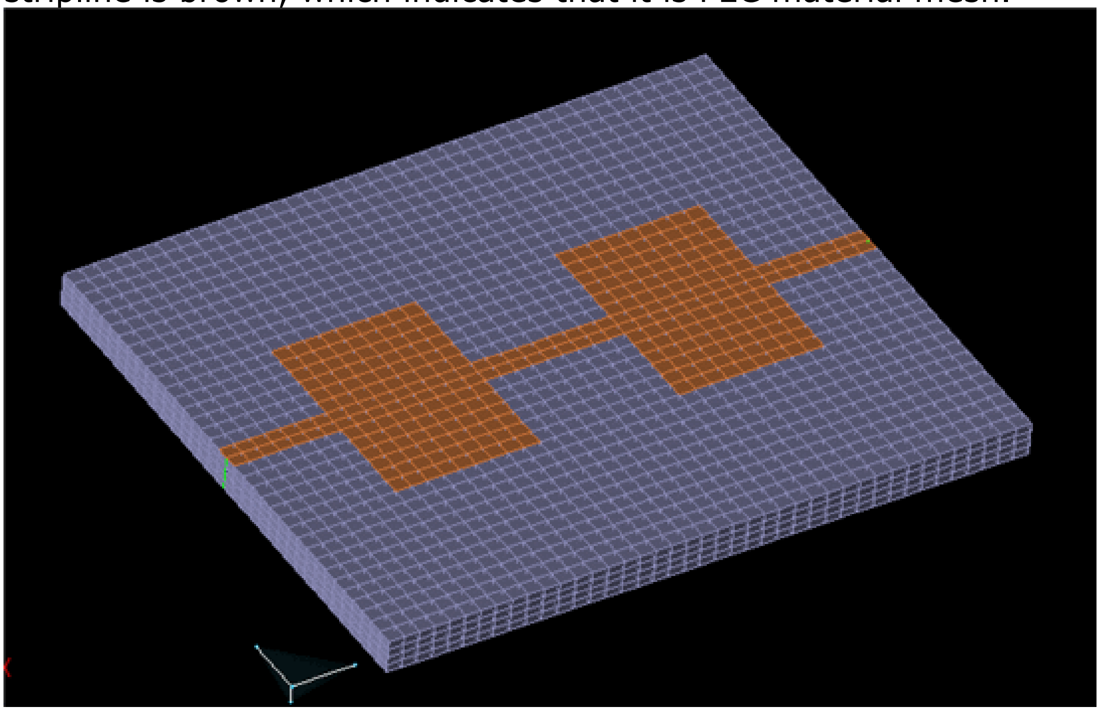


2. Set the **Mesh priority** to **51**. This sets the mesh priority of Stripline higher than the Substrate. By default, the mesh priority of all objects is set to 50.



3. Click **Toggle Mesh Control** on right side of GUI to view the 3D mesh.

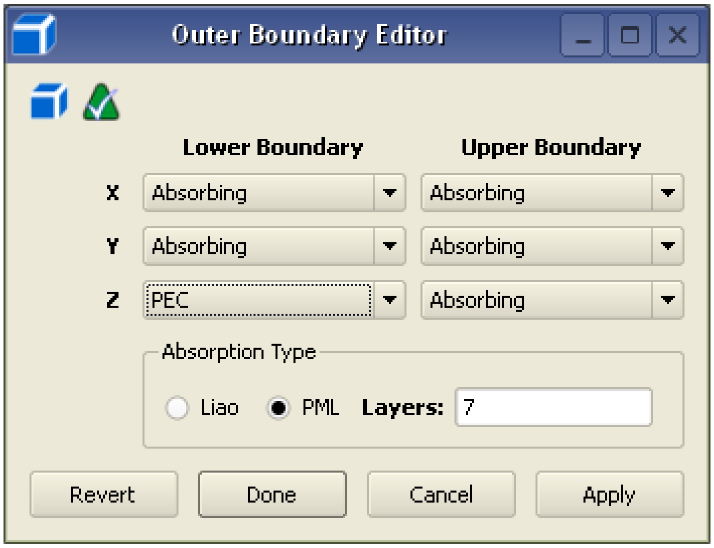
4. Mesh is displayed on the structure. Choose **3D mesh**. The color of mesh on the stripline is brown, which indicates that it is PEC material mesh.



Defining the Outer Boundary

1. Double-click the **Simulation Domain :Boundary Conditions** branch of the Project Tree to open the *Boundary Condition Editor*.
2. Set the outer boundary properties as follows:
   - **Boundary**: Absorbing for all boundaries except Lower Boundary Z, which should be PEC.
   - **Absorption Type**: PML
   - **Layers**: 7

Defining the outer boundary for the low pass filter

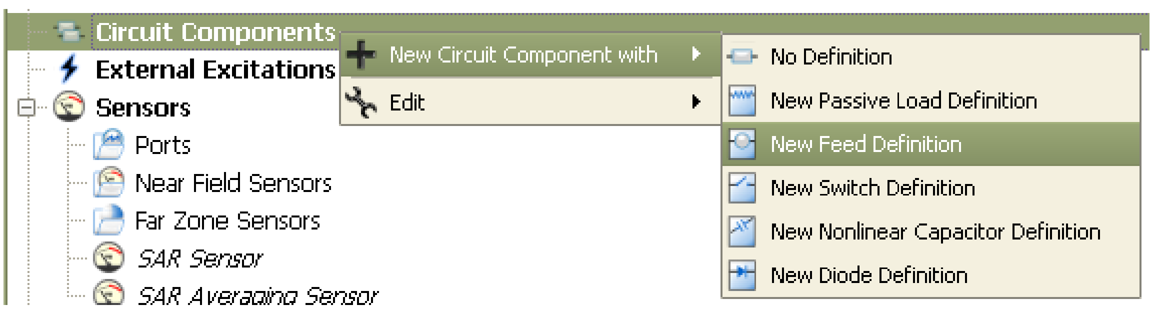


3. Click **Done** to apply the outer boundary settings.

Adding a Feed

1. Right-click the **Circuit Components/Ports** branch of the Project Tree. Choose **New Circuit Component with > New Feed Definition** from the context menu.

Adding a feed to the project



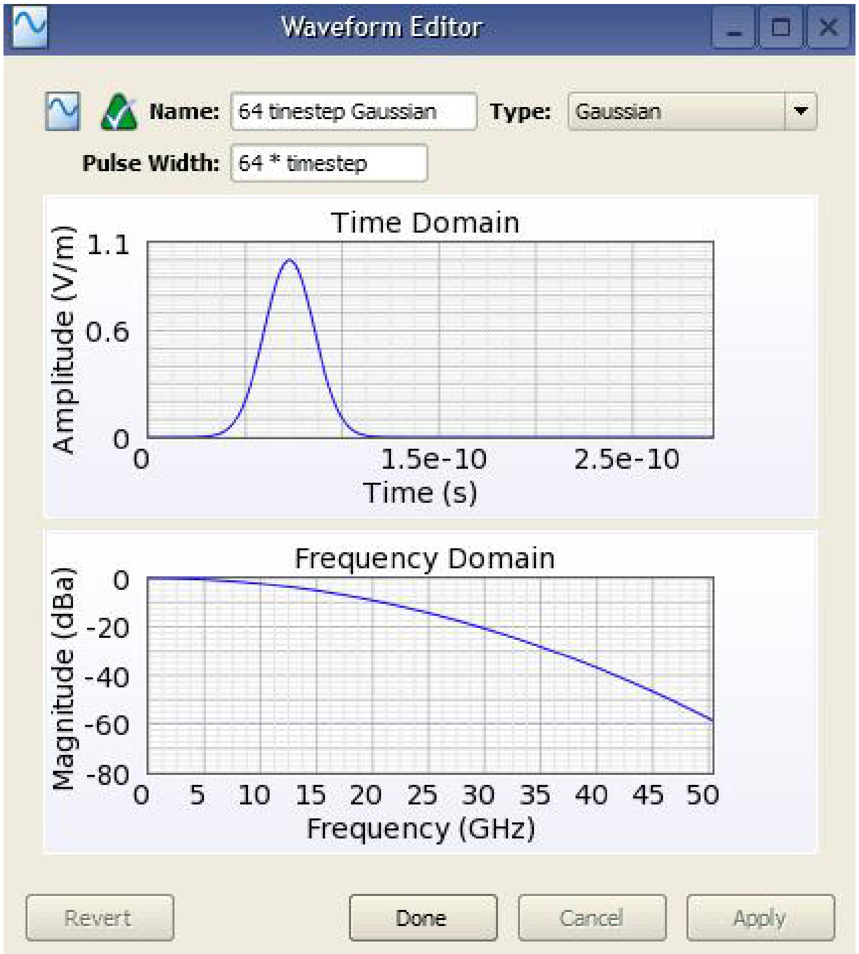
2. Define the endpoints of the feed.
    - Endpoint 1: X: 0 mm, Y: 5 mm, Z: 0 mm
    - Endpoint 2: X: 0 mm, Y: 5 mm, Z: -0.64 mm
3. Click **Done** to add the Feed.

Editing the Waveform

An associated waveform was automatically created for the feed definition.

1. Navigate to the **Definitions:Waveforms** branch of the Project Tree.
2. Double-click the **Broadband Pulse** waveform to edit its properties.
3. Set the properties of the waveform as follows:
    - **Name**: 64 timestep Gaussian
    - **Type**: Gaussian
    - **Pulse Width**: 64 * timestep

Editing the Gaussian waveform



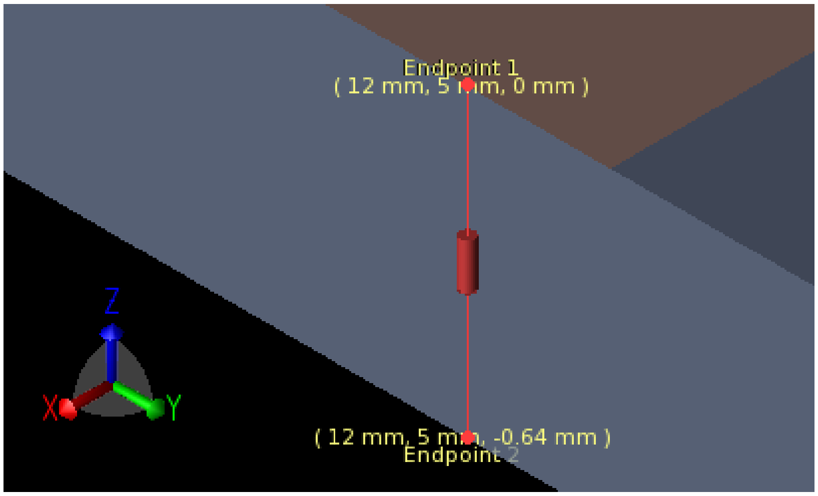
4. Click **Done** to apply the changes.

Adding a Load

The **Load** will be placed at the opposite end of the low pass filter geometry as the Feed.

	Note Before adding a load in FDTD, you should disable FEM.

1. Right-click on the **Circuit Components** branch of the Project Tree, and select **New Circuit Component with > New Passive Load** Definition.
2. Define the endpoints of the passive load.
   - **Endpoint 1**: X: 12 mm, Y: 5 mm, Z: 0 mm
   - **Endpoint 2**: X: 12 mm, Y: 5 mm, Z: -0.64 mm



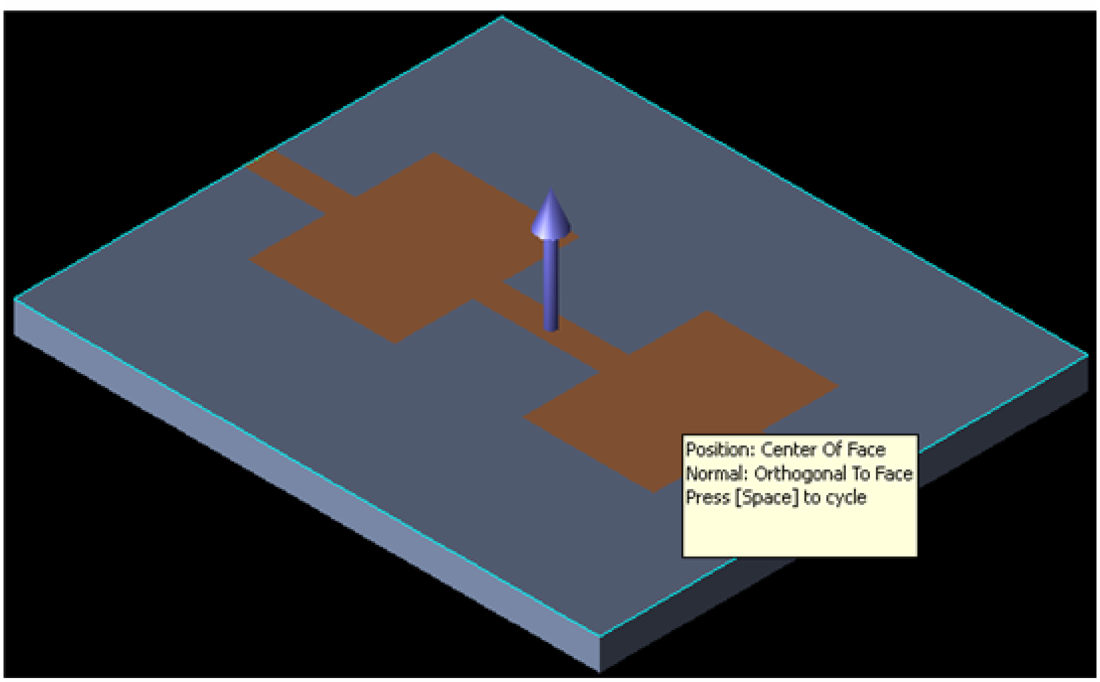
3. Click **Done** to add the Load to the project.

Requesting Output Data

This project already contains one port sensor named **Feed** that will request results. We also wish to collect field samplings at discrete intervals of time throughout the calculation. To retrieve this data, add a **Planar Sensor** at the surface of the Strip Line.

1. Right-click the **Sensors:Near Field Sensors** branch of the Project Tree. Select **New Planar Sensor** from the context menu.
2. Use the **Select** tool (at the top of the View Tools menu) to place the sensor in the middle of the **Strip Line**. Mouse over the Strip Line and press c to center the **Select** tool on the face. This will also set the sensor normal orthogonal to the face.

Centering the planar sensor on the Strip Line

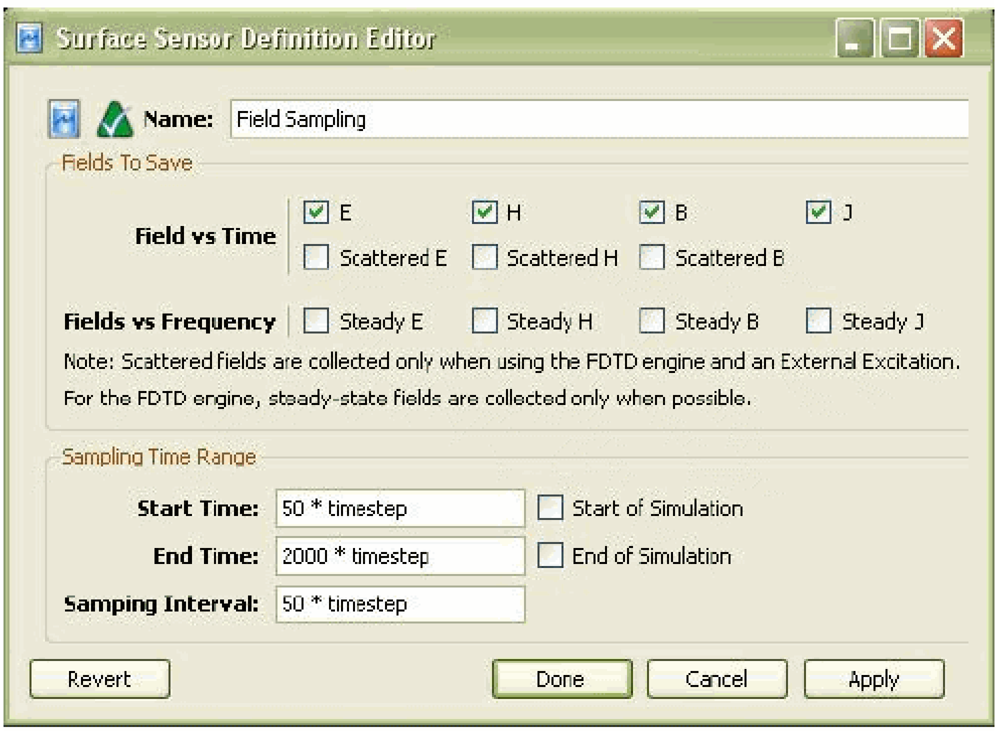


3. Click **Done** to add the Planar Sensor to the project.

This sensor requires a data definition.

1. Right-click the **Definitions:Sensor Data Definitions** branch of the Project Tree. Choose **New Surface Sensor Definition** from the context menu.
2. Set the properties of the surface sensor definition as follows:
    - **Name**: Field Sampling
    - **Field vs. Time**: E, H, B, and J
    - **Start Time**: 50 * timestep
    - **End Time**: 2000 * timestep
    - **Sampling Interval**: 50 * timestep

Adding the sensor definition



3. Click **Done** to finish editing the surface sensor definition.

Now, assign the new definition to the surface sensor.

1. Click and drag the **Field Sampling** definition located in the Project tree and drop it on top of the Surface Sensor in the **Sensors:Near Field Sensors** branch.

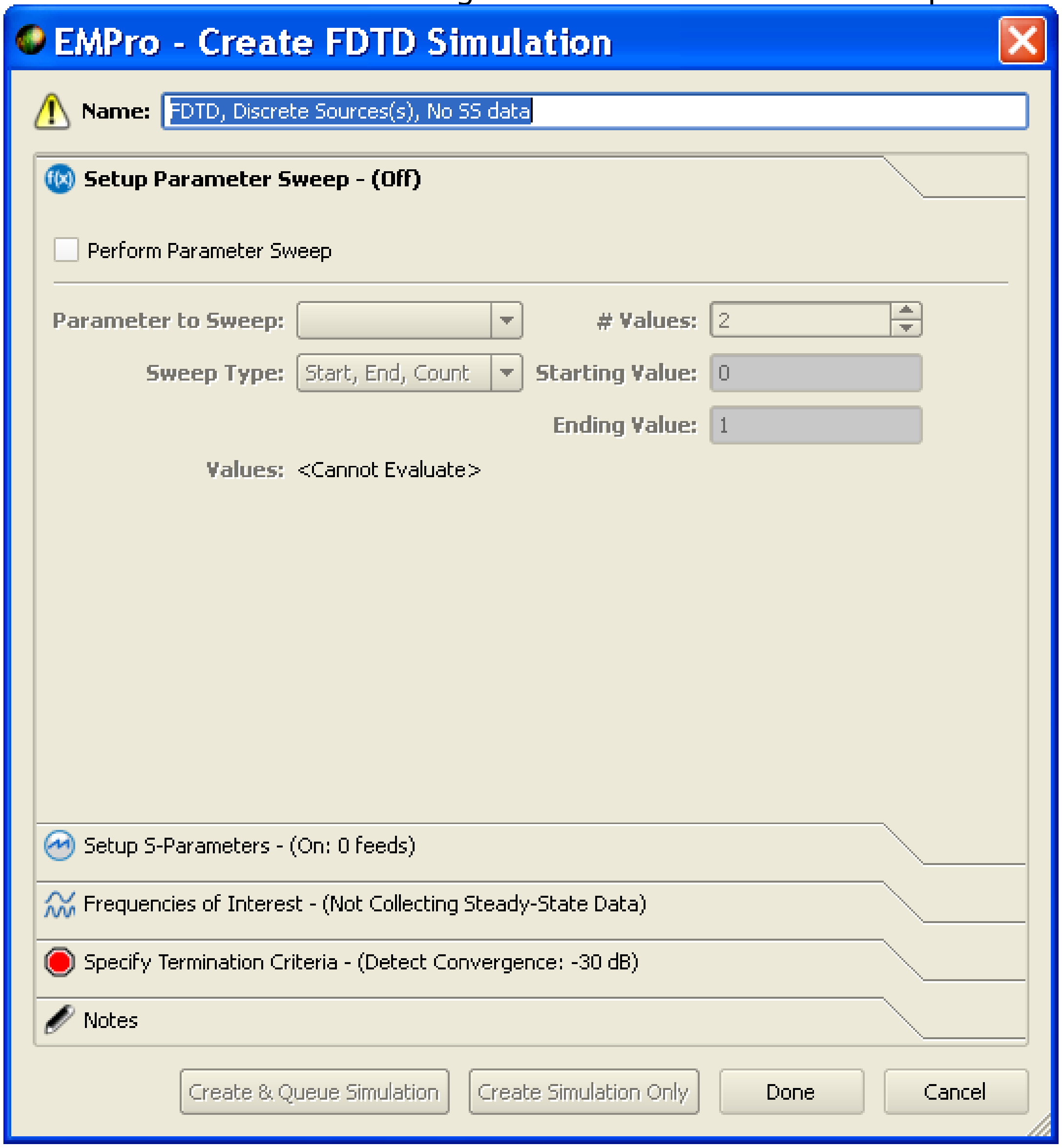
Running the Calculation

If you have not already saved your project, do so by selecting **File > Save Project**. After the project is saved, a new simulation can be created to send to the calculation engine.

1. Open the *Simulations* workspace window. Click the **New FDTD Simulation** button in

the upper-left side to set up a new simulation.

2. Navigate to the **Specify Termination Criteria** tab. Set up the termination criteria as follows:
    - **Maximum Simulation Time**: 10000 * timestep
    - **Detect Convergence**: Selected
    - **Threshold**: -40 dBh6. Adding a new simulation to the low pass filter project



3. Select **Create and Queue Simulation** to close the dialog and run the new simulation.

Viewing the Results

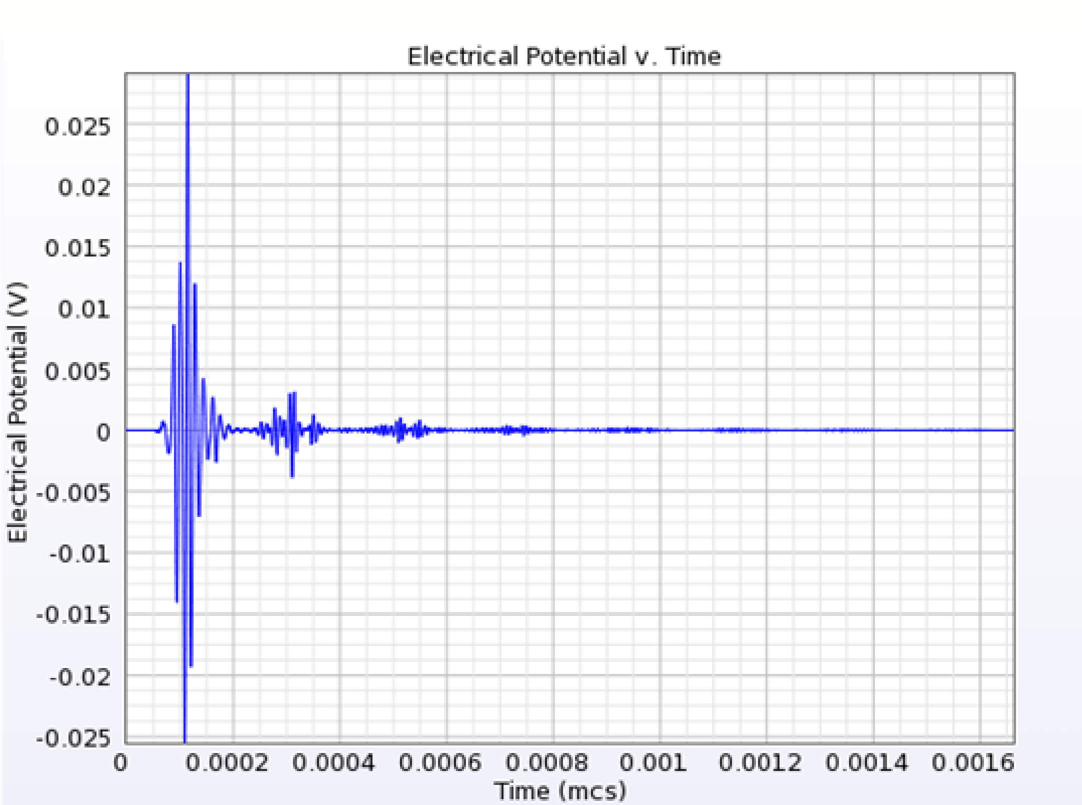
First, you will view the results retrieved with the port sensor placed at the location of the Feed.

Ensuring convergence has been reached

Although automatic convergence has been set, it is good practice to view the waveforms in the model to ensure that the energy has completely dissipated, providing complete convergence.

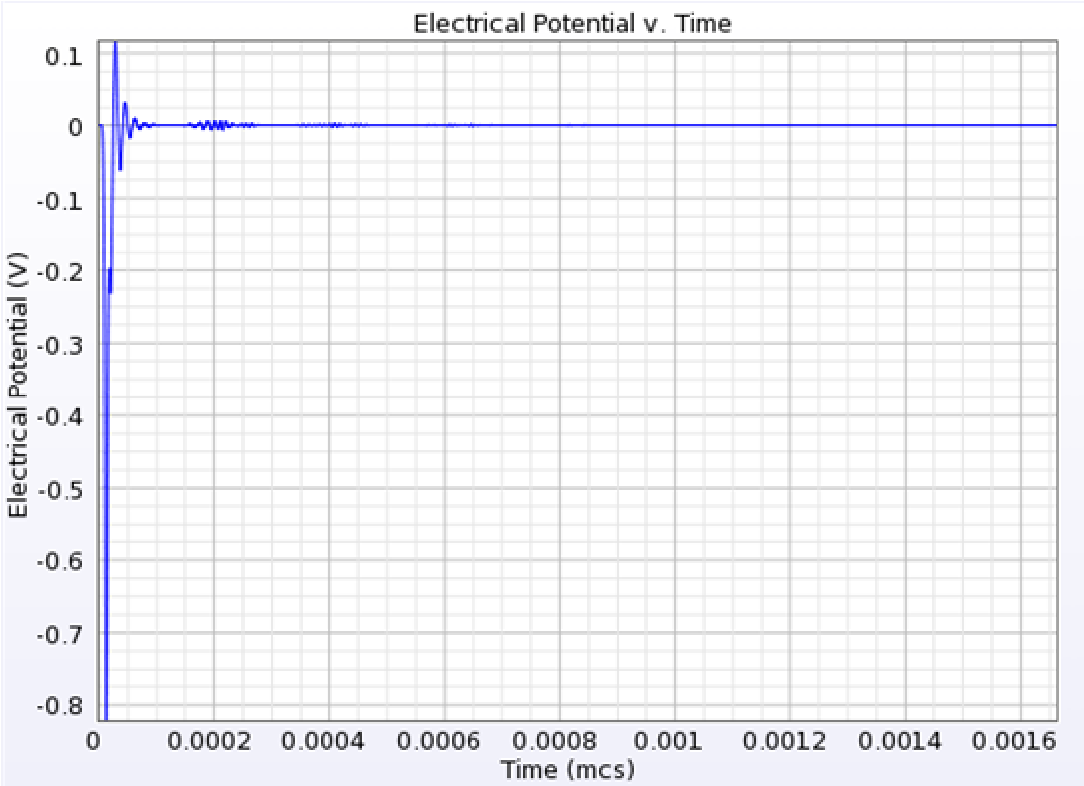
1. To filter the list accordingly, select the following options in the columns in the top pane of the *Results* window. (You may need to change your column headings first).
   - **Data Type**: Circuit Component
   - **Domain**: Time
   - **Result Type**: Voltage (V)
   This will filter all time-dependent voltage data collected by the Feed circuit component.
2. Double-click the **Load** result to view a 2-D plot to ensure convergence has been met.

Viewing results plot to ensure convergence at the Load



3. Repeat to view the results at the Feed.

Viewing results plot to ensure convergence at the Feed

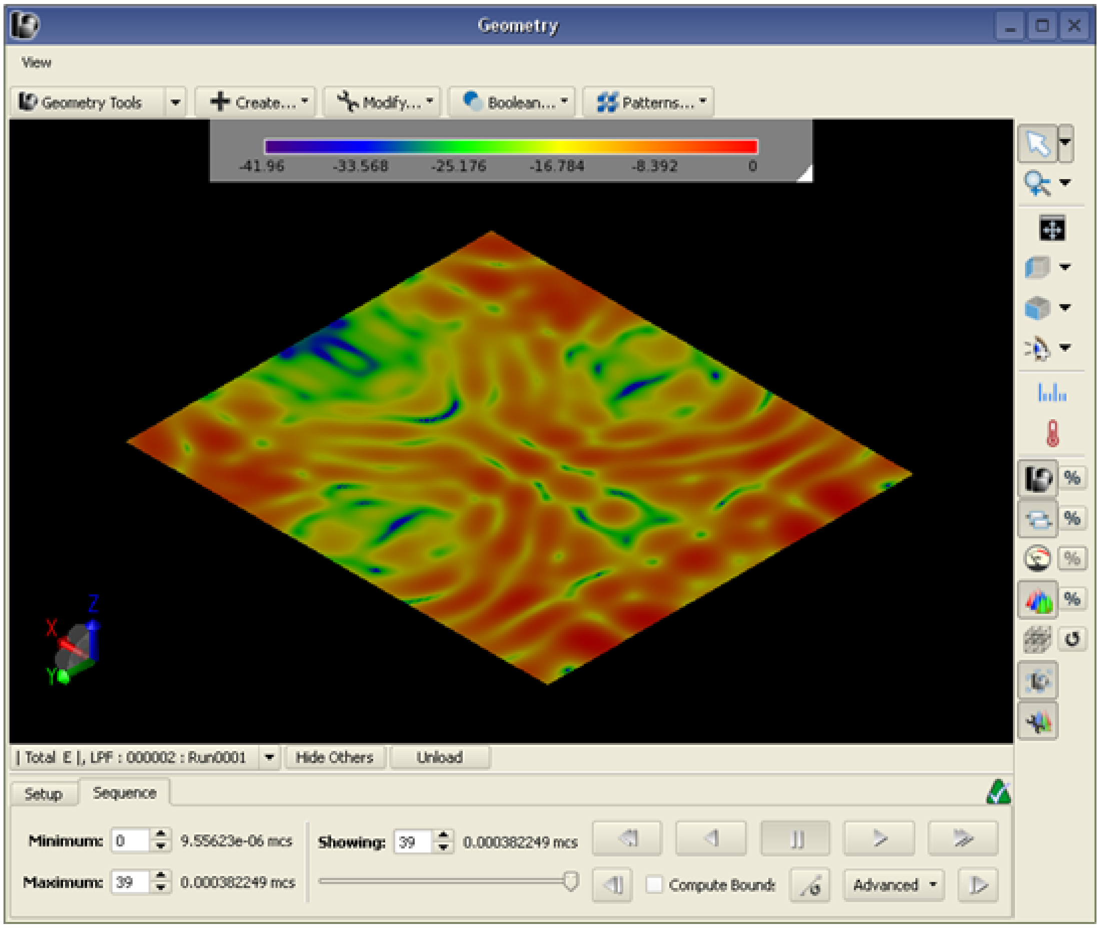


E-Field Results from the Surface Sensor

Now, you will view the field sequences collected by the surface sensor that was placed at the surface of the Strip Line.

1. To filter the E-field results, select:
    - **Data Type**: Surface Sensor
    - **Result Type**: E-Field (E)
2. Double-click the result to open the interface and view the 3-D field sequence.
3. Navigate to the Sequence tab to view the results. You can play back the results as an animation or step through them with the Showing control. If you wish, change the Minimum and Maximum settings to only display a certain range of the sequence.

Viewing E-field results for the surface sensor at the last frame of the sequence


Geometry
View
Geometry Tools
Create...
Modify...
Boolean...
Patterns...
-41.96
-33.568
-25.176
-16.784
-8.392
0
Z
X
Y
| Total E |, LPF : 000002 : Run0001
Hide Others
Unload
Setup
Sequence
Minimum:
0
9.55623e-06 mcs
Maximum:
39
0.000382249 mcs
Showing:
39
0.000382249 mcs
Compute Bounds
Advanced

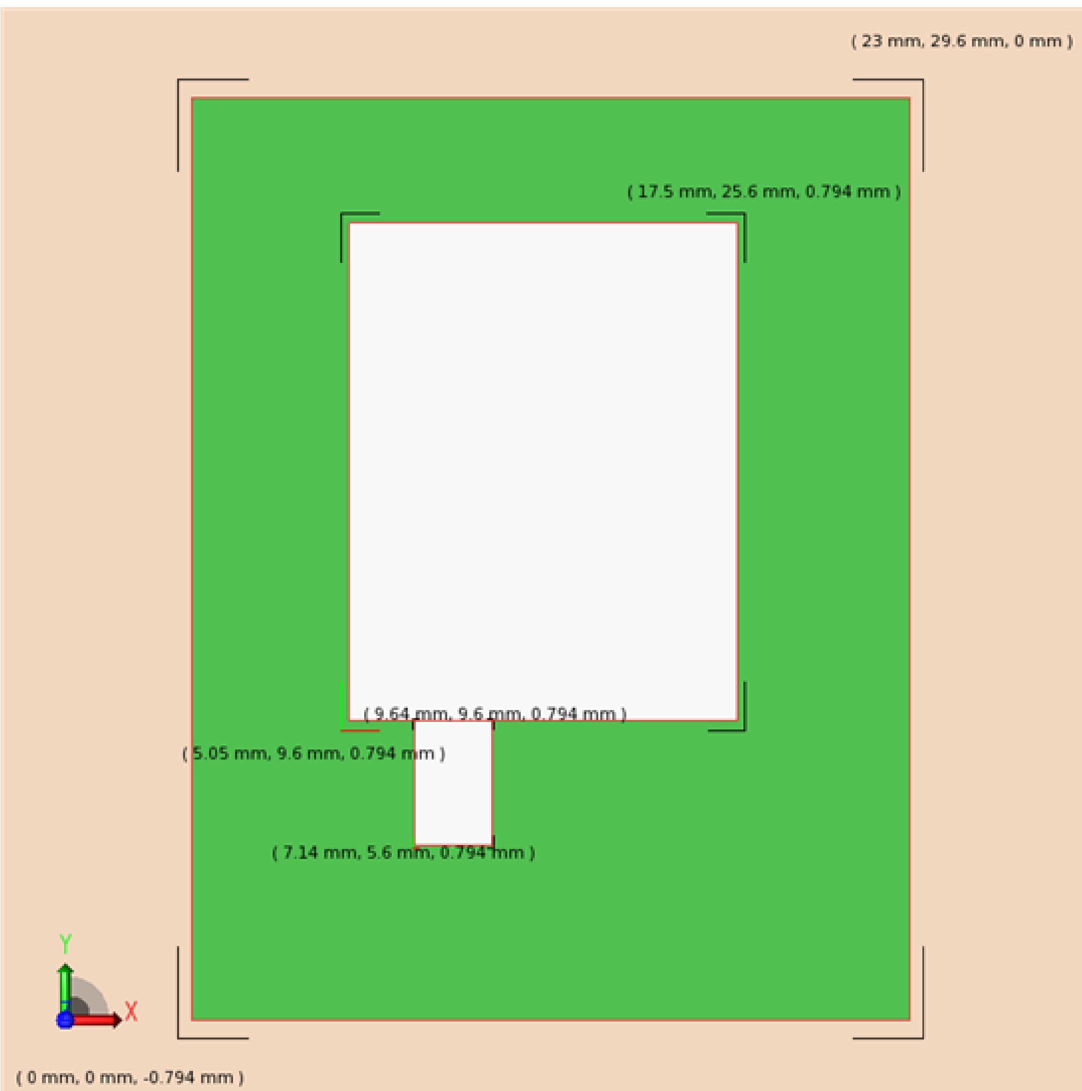
Creating a Microstrip Patch Antenna Simulation

In this example, you will learn how to:

- Set project preferences and display units to initialize any FDTD project.
- Create the shape and set the material properties of your patch antenna.
- Add a feed to the antenna and simulate its effects.
- View the plotted results of your simulation.
- Run an additional simulation with a parameter sweep.

This microstrip patch antenna example is based on a paper by Sheen et al. The patch antenna from Figure 3 of this paper will be constructed and the S-parameters compared with the measured and computed return loss of Figure 5 of the paper. The substrate thickness is 0.794 mm with a relative permittivity $\epsilon_r = 2.2$.

Microstrip Patch Antenna



Getting Started

This section briefly describes how to set the display units for the Patch Antenna project. To set up a project for the first time, refer to for instructions on how to configure project preferences and navigate through the **Display Units** tab.
In the Project Properties Editor window, navigate to the Display Units tab:

1. Select SI Metric in the Unit Set drop down list.
2. Select the **Show all units** checkbox, and adjust the following settings:
   - Change Capacitance to **millifarads (mF)**.
   - Change Current to **milliamperes (mA)**.
   - Change Frequency to **gigahertz (GHz)**.
   - Change Length to **millimeters (mm)**.
   - Change Power to **milliwatts (mW)**.
   - Change Electrical Potential to **millivolts (mV)**.

> **Note**
> The value of Unit Set changes to **Custom** after these settings are adjusted.

3. Click **Done**.

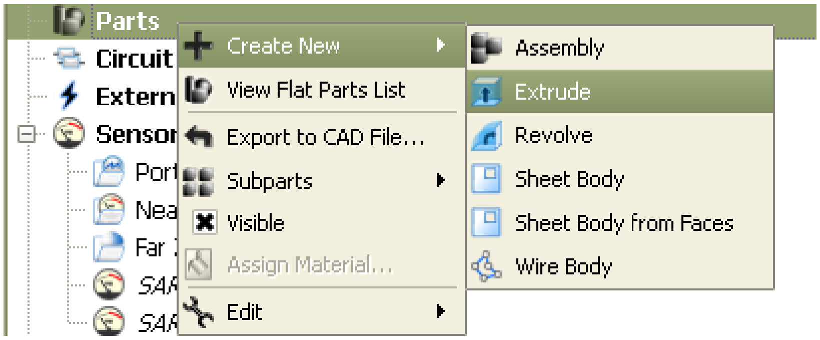
Creating the Patch Antenna Geometry

You will create the patch antenna geometry from of 2 simple components: a rectangular substrate and a microstrip patch. For this example, you will use the Geometry *Tools* interface to create a rectangular Extrusion and a pair of rectangular Sheet Bodies.

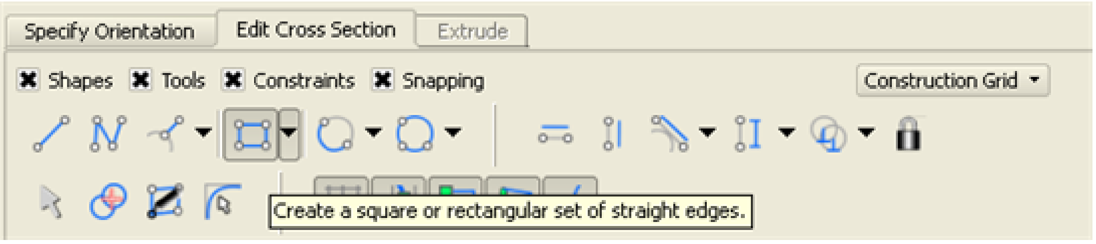
Modeling the Substrate

First, you will create the rectangular substrate named Substrate. This object will stretch from (0, 0, 0) to (23, 29.6, 0) and have a .794mm extrusion in the +Z direction.

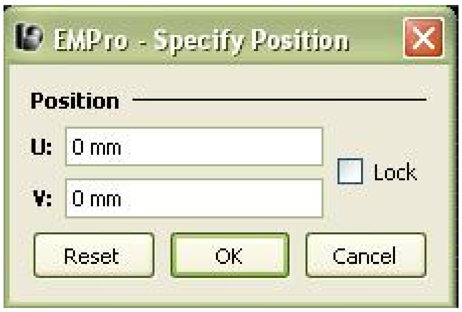
1. Right-click the **Parts** branch of the Project Tree. Choose **Create New > Extrude** from the context menu.



2. Name the substrate by typing **Substrate** in the Name box in the upper-right corner of the window.
3. Choose the **Rectangle** tool from the **Shapes** toolbar.

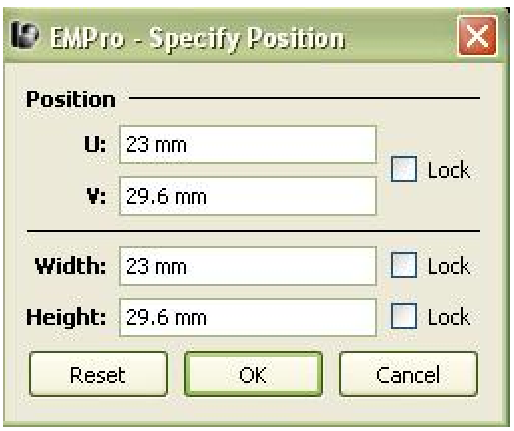
3. 

- The *Creation* dialog box allows exact entry of coordinates. Right-click in the geometry editing space and press the **Tab** key in the geometry space to activate the *Creation* dialog box. Specify the position of the first point.



to **activate** the window

4. Press the **Tab** key to display the *Creation* dialog box for the second point. Enter (23mm, 29.6mm) and press **OK** to complete the rectangle.

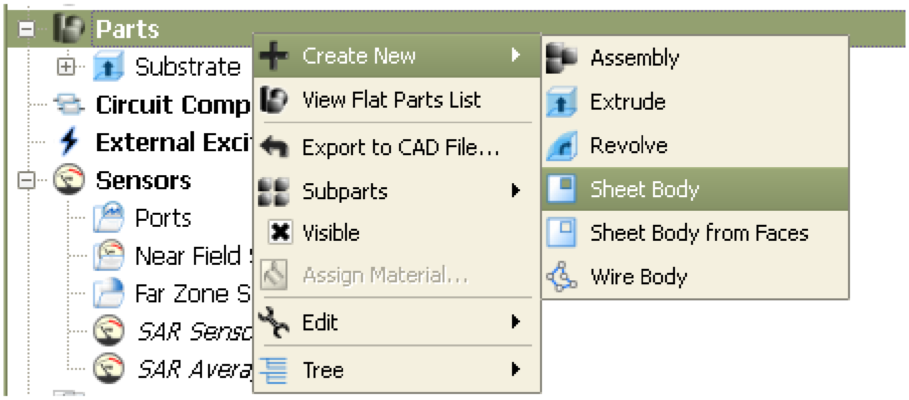


5. Navigate to the **Extrude** tab to extrude the rectangular region. Enter a distance of 0.794mm.
6. Click **Done** to finish the Substrate geometry.

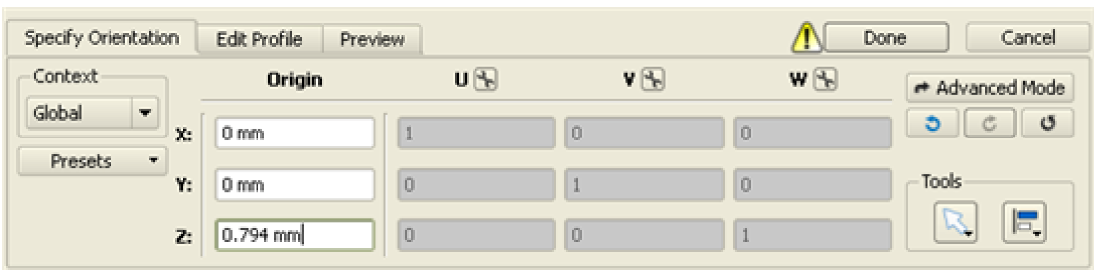
Modeling the Microstrip Patch

The microstrip patch will be created with a Sheet Body object that rests on top of the Substrate. This shape will be comprised of two rectangles. The patch will stretch from (5.05, 9.6, 0.794) to (17.5, 25.6, 0.794). The stub will stretch from (7.2, 5.6, 0.794) to (9.6, 9.6, 0.794).

1. Right-click the **Parts** branch of the Project Tree. Choose **Create New Sheet Body** from the context menu.

1. 

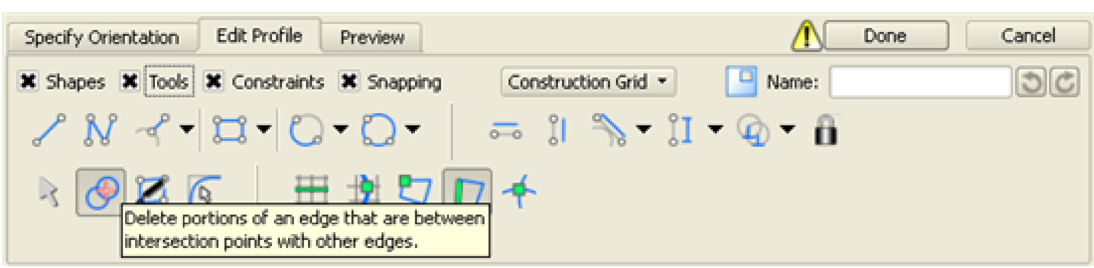
2. Navigate to the **Specify Orientation** tab. Set the origin to (0, 0, 0.794;mm) to place the Sheet Body on top of the Substrate.



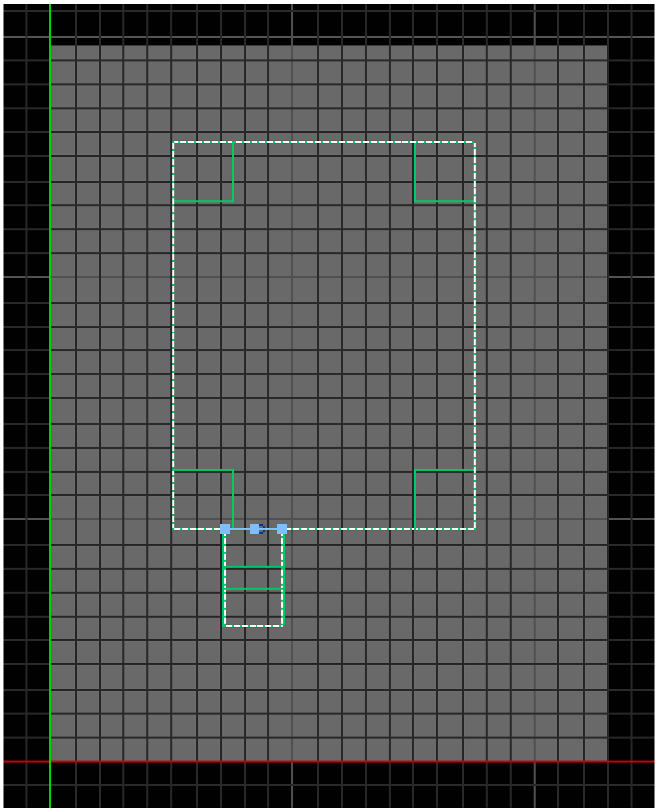
3. Navigate to the **Edit Profile** tab. Type **Microstrip** into the **Name** text box.

You will draw the microstrip and its stub individually and then combine them into a single polygon.

1. Select the **Rectangle** tool. Use the *Creation* dialog box to enter the corners of the microstrip rectangle:
   - **Endpoint 1**: (5.05mm, 9.6mm)
   - **Endpoint 2**: (17.5mm, 25.6mm)
   - Now use the *Creation* dialog box to enter the corners of the stub rectangle:
   - **Endpoint 1**: (7.2mm, 5.6mm)
   - **Endpoint 2**: (9.6mm, 9.6mm)
2. Select the **Trim Curves** tool.

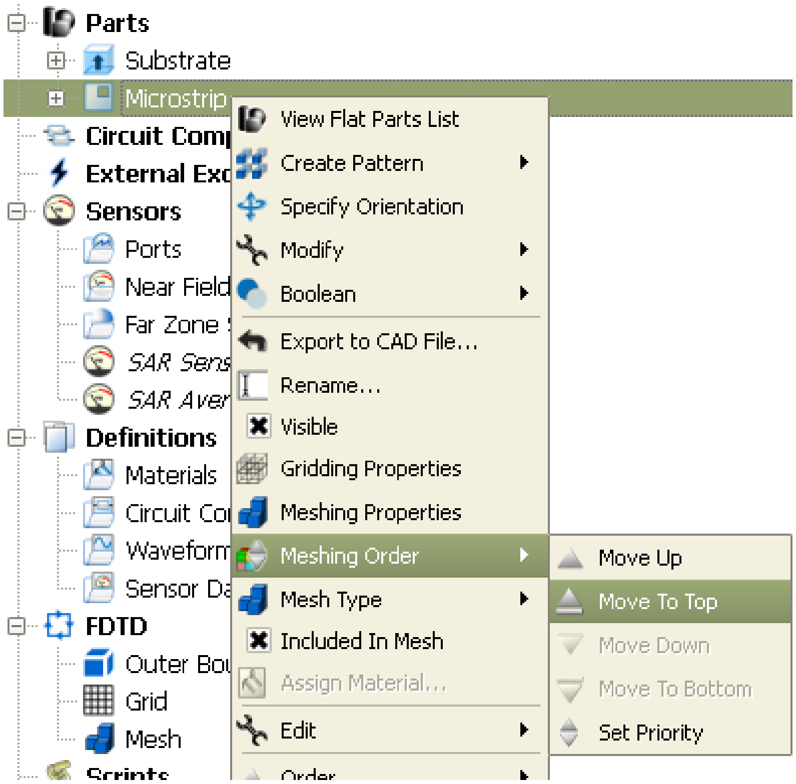


3. Remove the line segment between the microstrip and the stub by clicking on it.

3. 
4. Click **Done** to finish the Microstrip geometry.

Meshing Priority

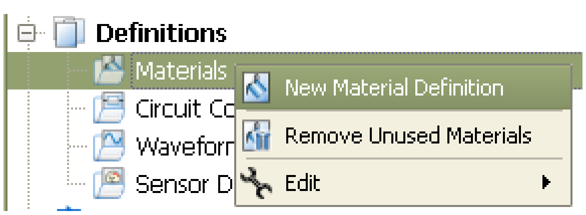
Ensure that the meshing priority of the Microstrip is greater than the Substrate for an accurate calculation.
Right-click the Microstrip in the Project Tree. Under **Meshing Order**, select **Move to Top** if it is an available option.



Creating Materials

Define material, PEC

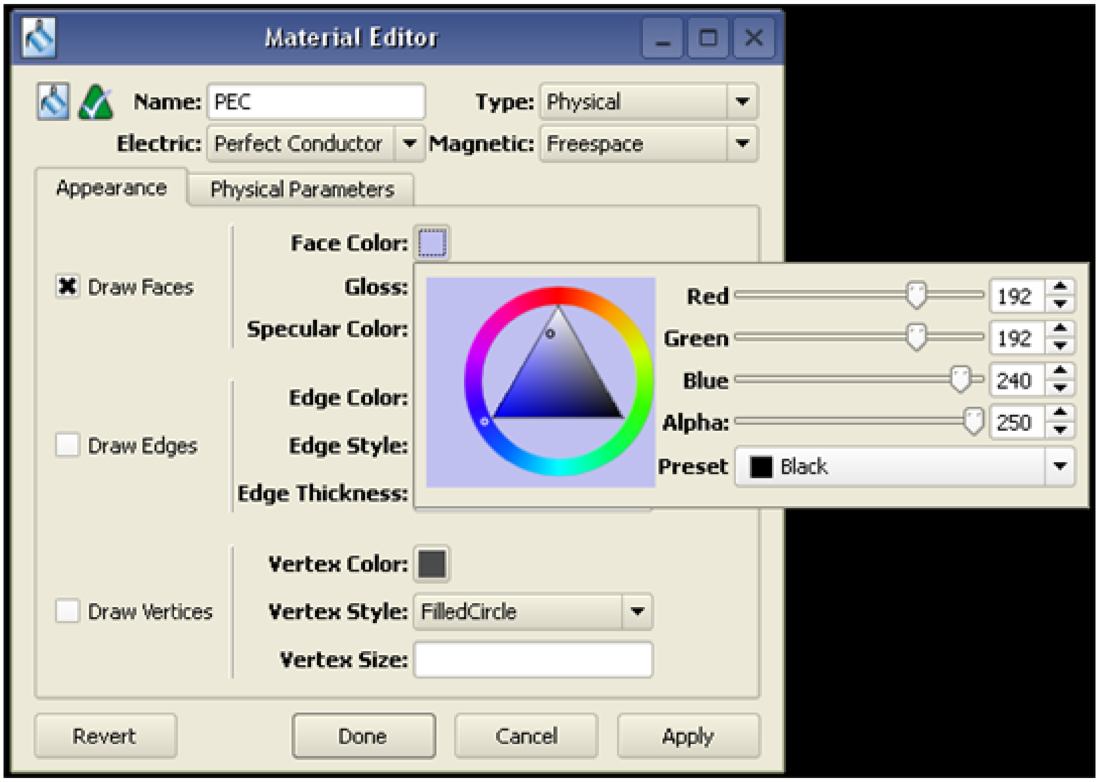
1. Create a perfect electric conductor material by right-clicking the **Definitions:Materials** branch of the Project Tree. Choose **New Material Definition** from the context menu.



2. Double-click the new material to edit its properties. Set the perfect electric conductor material properties as follows:

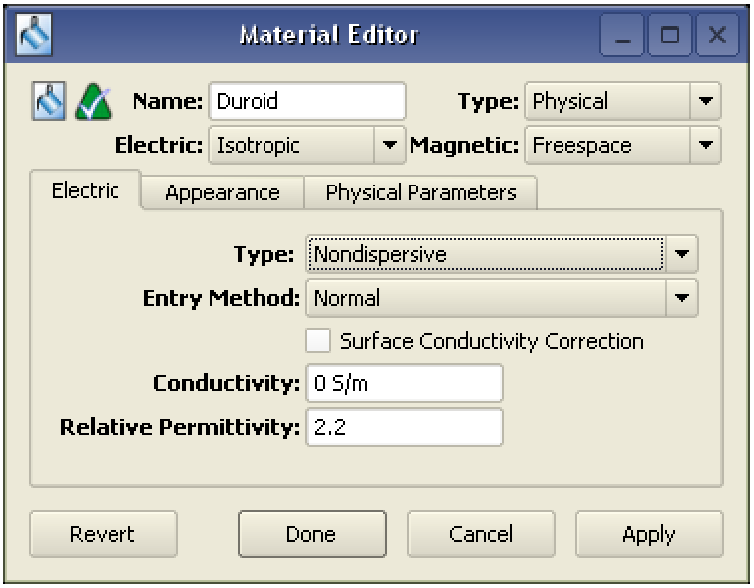
    - **Name**: PEC
    - **Electric**: Perfect Conductor
    - **Magnetic**: Freespace

    If desired, navigate to the Appearance tab to set the PEC material display color.

- 

Define material, Duroid

1. Right-click the **Definitions:Materials** branch of the Project Tree. Choose **New Material Definition** from the context menu.
2. Double-click the new material to edit its properties. Set the duroid material properties as follows:
    - Name: Duroid
    - Electric: Isotropic
    - Magnetic: Freespace
    - Under the Electric tab:
        - Type: Nondispersive
        - Entry Method: Normal
        - Conductivity: 0 S/m
        - Relative Permittivity: 2.2

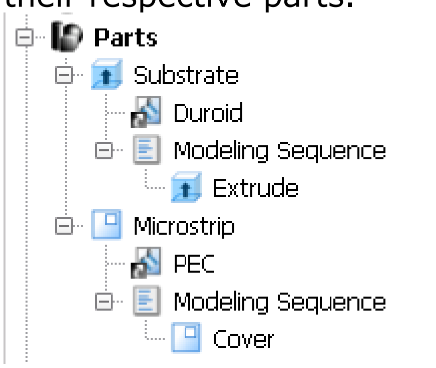


3. If desired, navigate to the Appearance tab to set the Duroid material's display color.

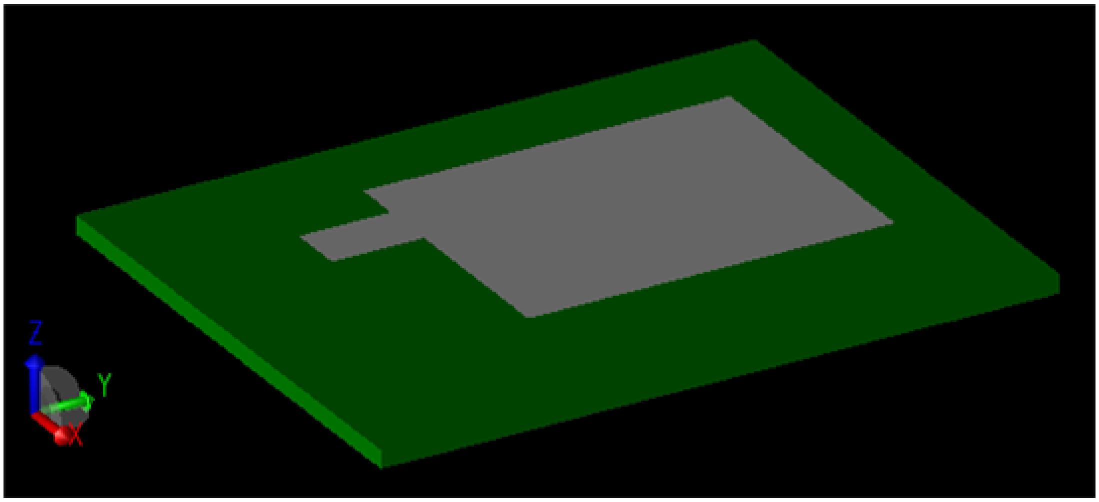
Assigning Materials

1. Click and drag the PEC material object located in the Project Tree and drop it on top of the Microstrip objects in the **Parts** branch of the tree.
2. Assign the **Duroid** material to the **Substrate** object using the same procedure.

The following image shows the Project Tree after material objects have been dropped on their respective parts.



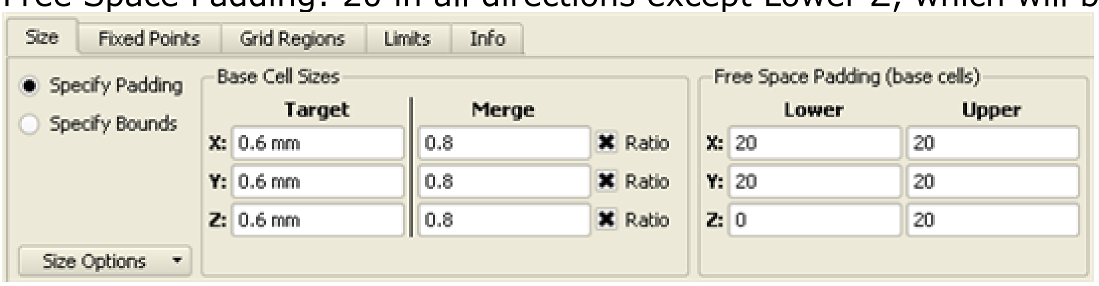
This image shows the microstrip patch antenna geometry with materials applied and colors set for each.



Creating the Grid

Now, you will define characteristics of the cells in preparation to perform an accurate calculation.

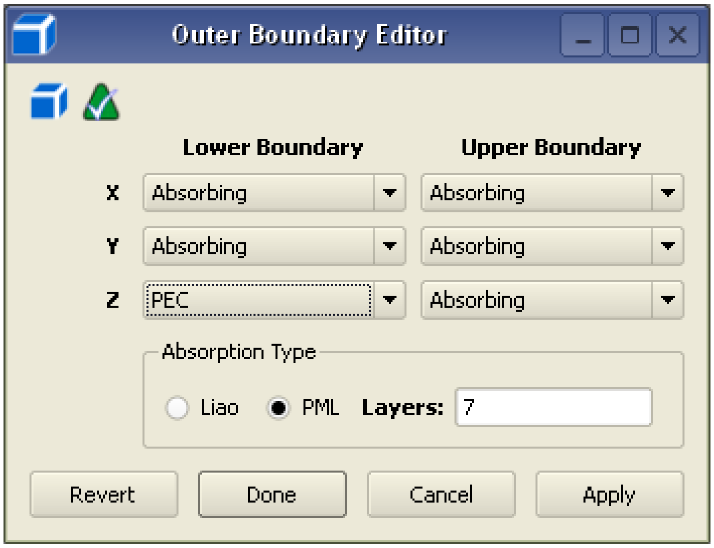
1. Double-click the **FDTD:Grid** branch of the Project Tree to open the **Grid Tools** dialog box.
2. Set the Size properties of the grid as follows:
   - Base Cell Sizes: Target 0.6 mm, Merge 0.8, Ratio boxes selected
   - Free Space Padding: 20 in all directions except Lower Z, which will be 0



3. Click **Done** to apply the grid settings.

Defining the Outer Boundary

1. Double-click the **FDTD:Outer Boundary** branch of the Project Tree to open the Outer Boundary Editor.
2. Set the outer boundary properties as follows:
   - Boundary: Absorbing for all boundaries except Lower Boundary Z, which should be PEC
   - Absorption Type: PML
   - Layers: 7

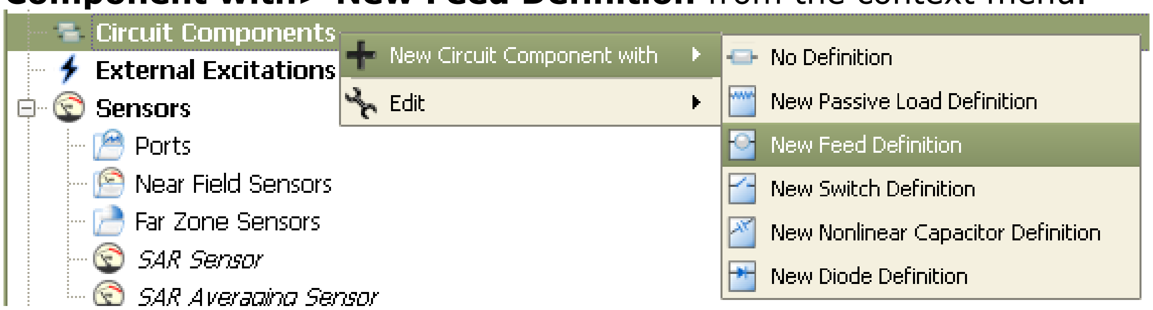


3. Click **Done** to apply the outer boundary settings.

Adding a Feed

You can now add a Feed to the patch antenna geometry. It will consist of a voltage source and series 50Ω resistor connected between the base of the stub portion of the Microstrip and the ground plane. Then, you will then apply a 64-timestep Gaussian waveform to the circuit through this feed.

1. Right-click the **Circuit Components** branch in the Project Tree. Choose **New Circuit Component with> New Feed Definition** from the context menu.

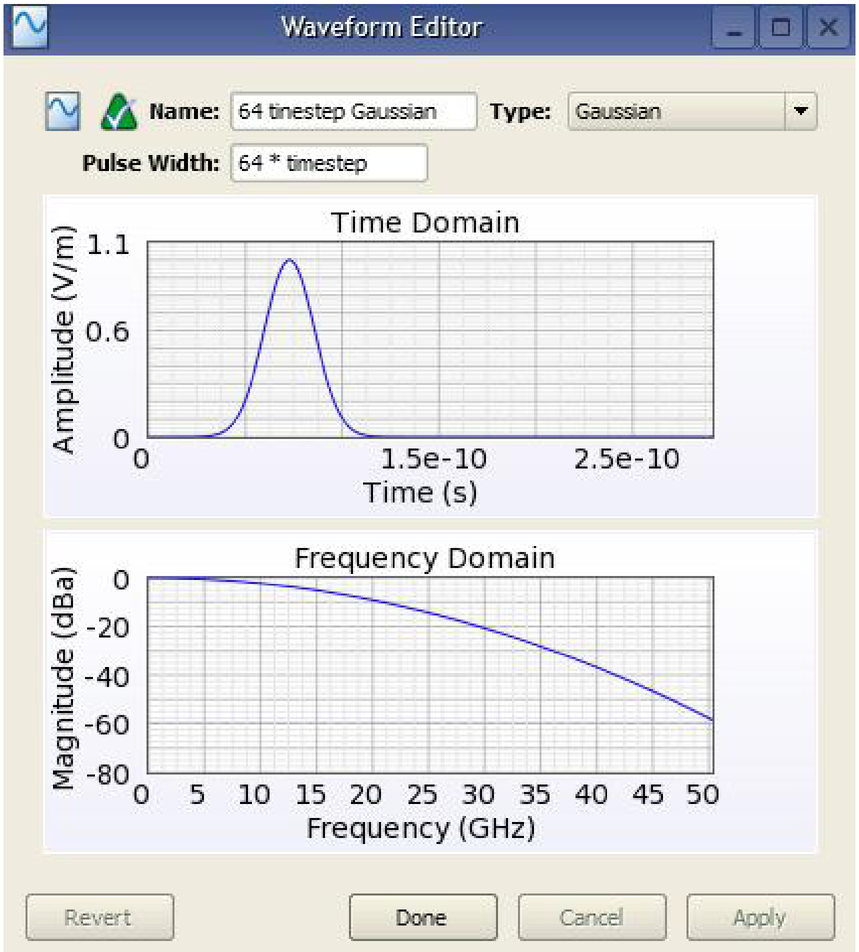


2. Define the endpoints of the feed.
   - Endpoint 1: X: 8.15 mm, Y: 5.6 mm, Z: 0.794 mm
   - Endpoint 2: X: 8.15 mm, Y: 5.6 mm, Z: 0 mm
3. Navigate to the Properties tab, and name the component Feed.
4. Click **Done** to add the Feed.

Editing the Waveform

An associated waveform was automatically created for the feed definition.

1. Navigate to the **Definitions:Waveforms** branch of the Project Tree. Double-click the **Broadband Pulse waveform** to edit its properties.
2. Set the properties of the waveform as follows:
   - Name: 64 timestep Gaussian
   - Type: Gaussian
   - Pulse Width: 64 timestep

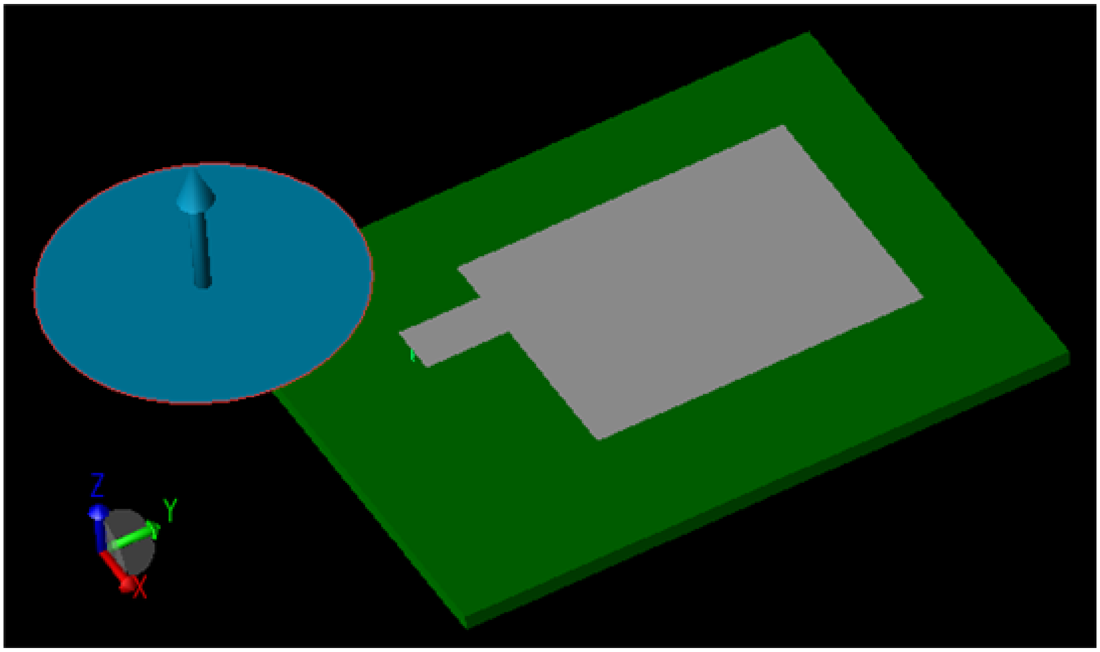


3. Click **Done** to apply the changes.

Requesting Output Data

You will add a **Planar Sensor** at the surface of the Microstrip plate to retrieve electric field sampling data.

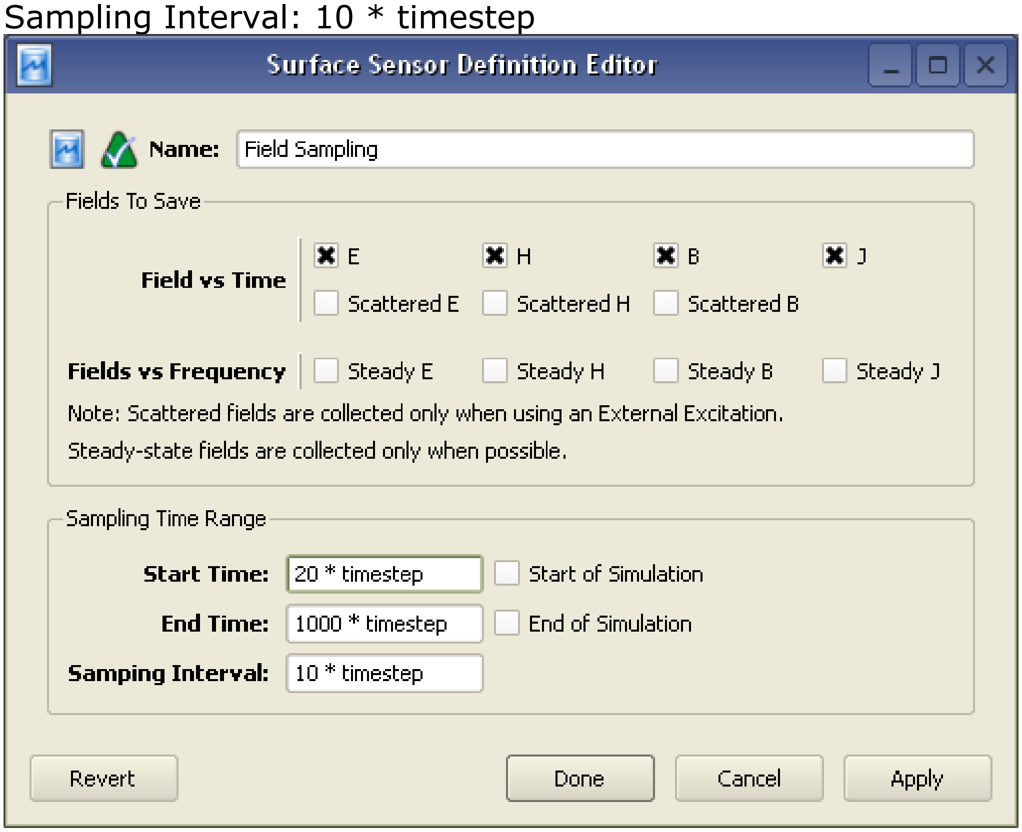
1. Right-click the **Sensors:Near Field Sensors** branch of the Project Tree. Select **New Planar Sensor** from the context menu.
2. Place the Point at (0, 0, 0.794 mm) to position the sensor on top of the Substrate. Set the Normal to (0, 0, 1).



3. Click **Done** to add the planar sensor.

This sensor requires a data definition.

1. Right-click the **Definitions:Sensor Data Definitions** branch of the Project Tree. Choose **New Surface Sensor Definition** from the context menu.
2. Set the properties of the surface sensor definition as follows:
   - Name: Field Sampling
   - Field vs. Time: E, H, B, and J
   - Start Time: 20 * timestep
   - End Time: 1000 * timestep
   - Sampling Interval: 10 * timestep



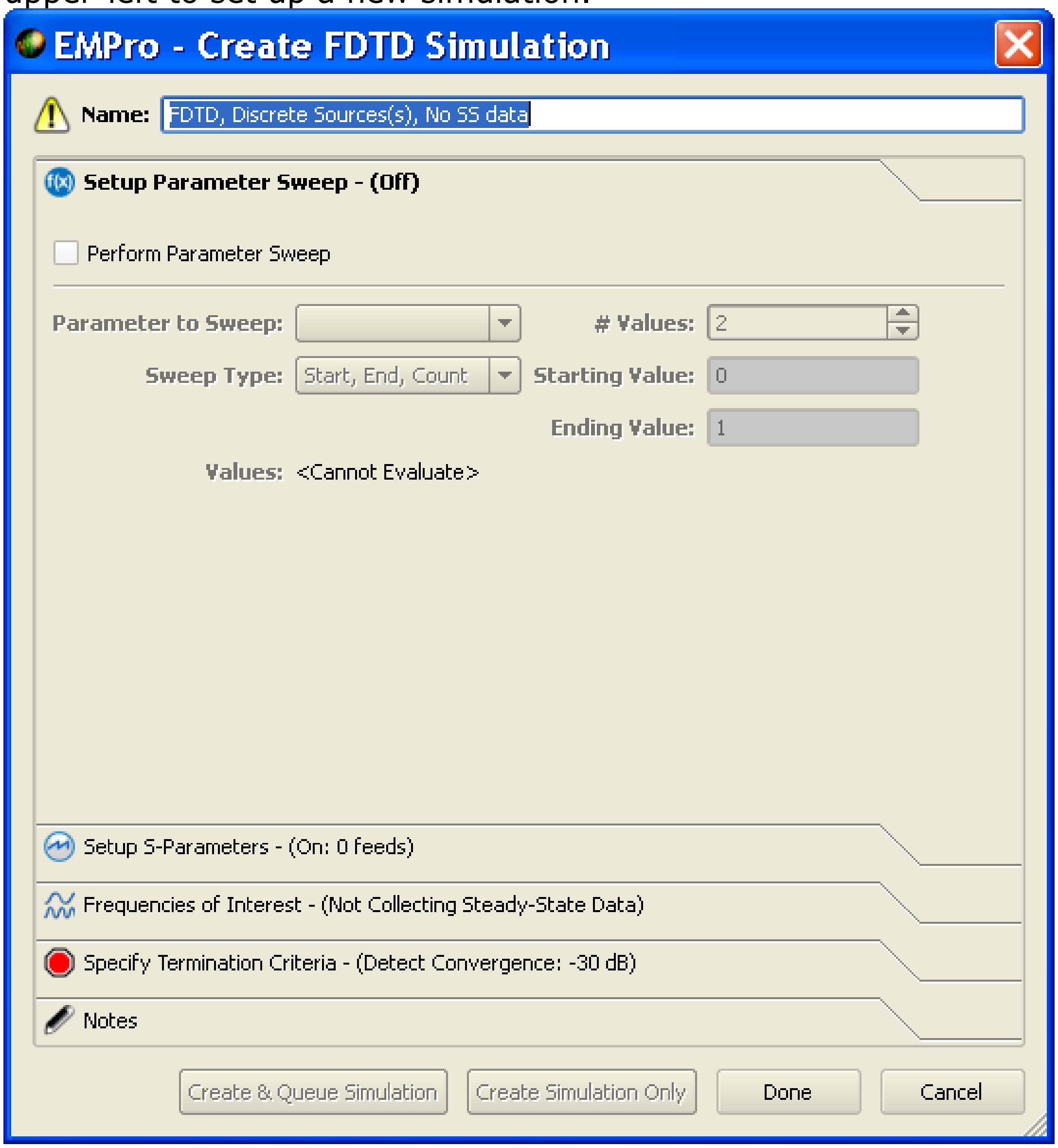
3. Click **Done** to finish editing the surface sensor definition.

Now, assign the new definition to the surface sensor. Click and drag the **Field Sampling** definition located in the Project Tree and drop it on top of the **Surface Sensor** in the **Sensors:Near Field Sensors** branch.

Running a Simulation

If you have not already saved your project, do so by selecting **File>Save Project**. After the project is saved, a new simulation can be created to send to the calculation engine.

1. Open the **Simulations** workspace window. Click **New FDTD Simulation** in the upper-left to set up a new simulation.



2. The default settings are sufficient for this example. Click **Create & Queue Simulation** to close the dialog and run the new simulation.

Viewing the Results

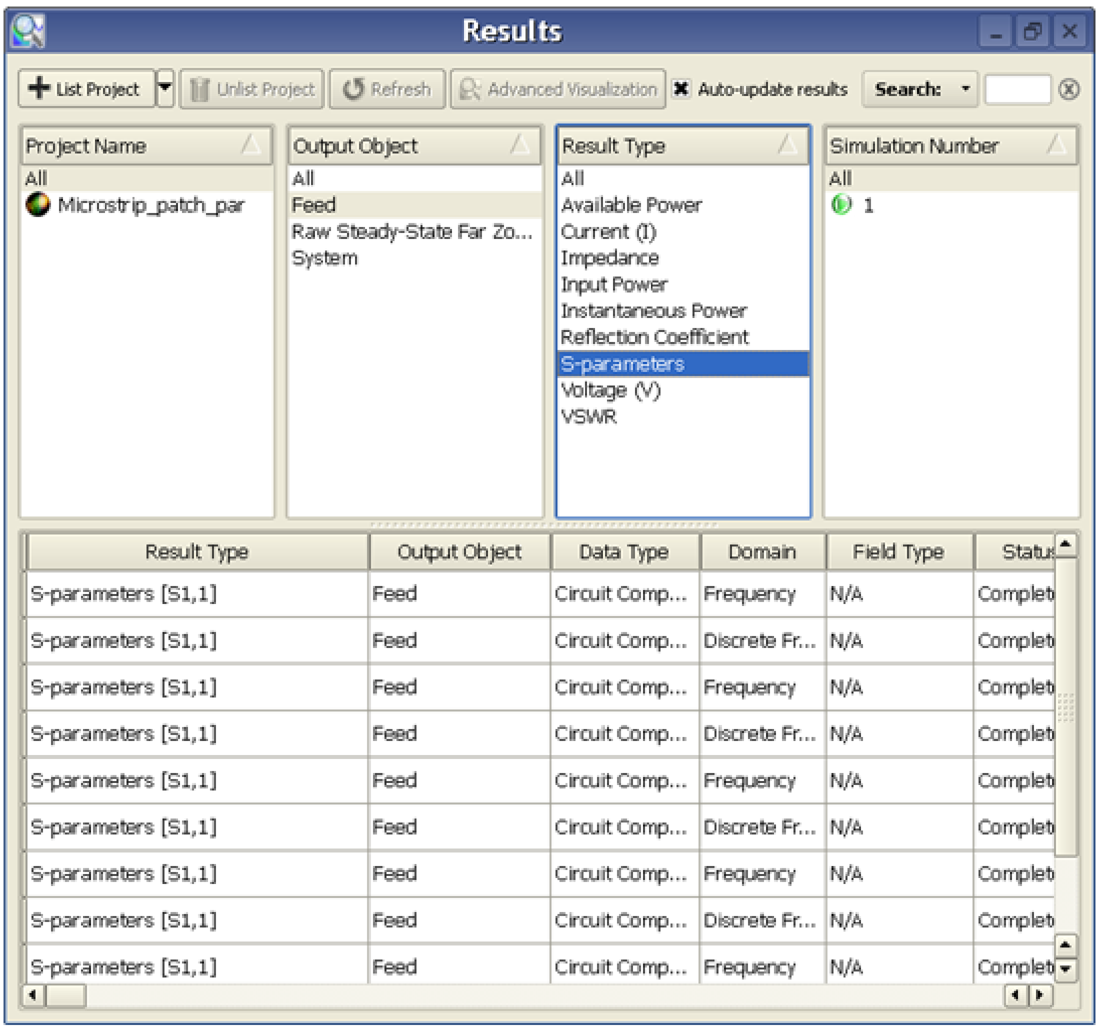
The Output tab of the Simulations workspace window displays the progress of the simulation. After the Status column shows that the simulation has completed, you can view its results in the Results workspace window.

S-Parameter Results from the Port Sensor

First, you will view the S-parameter results retrieved with the port sensor placed at the location of the Feed.

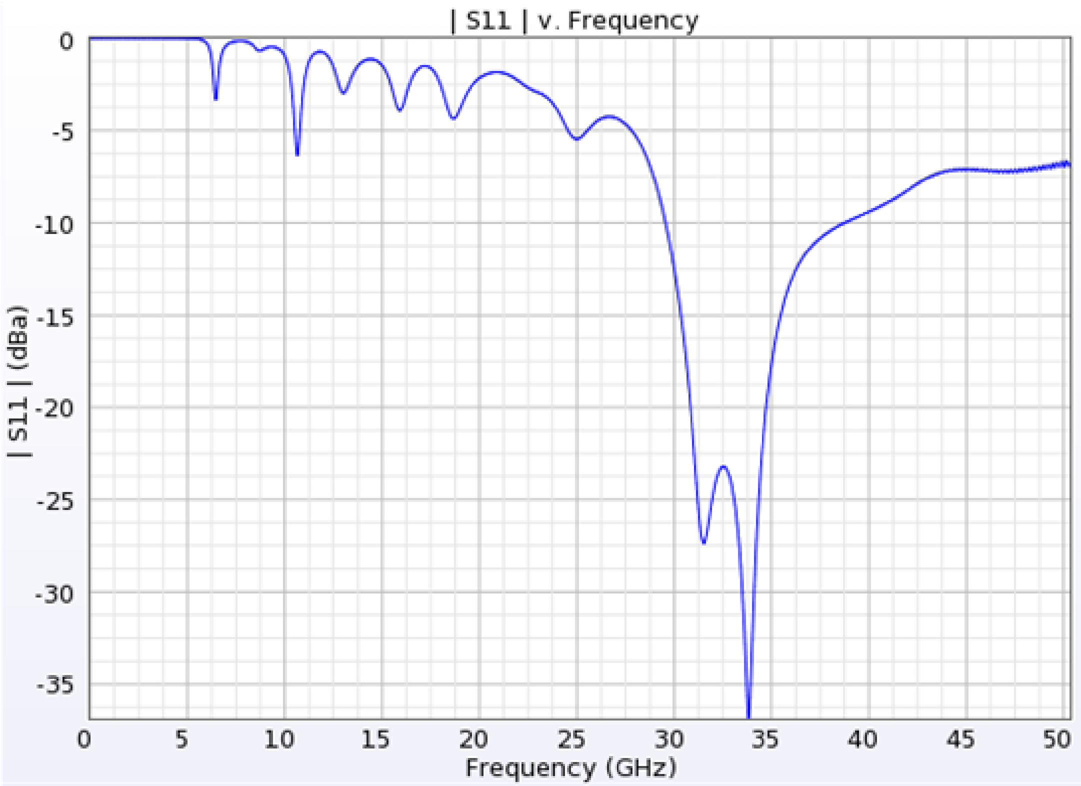
1. To filter the list accordingly, select the following options in the columns in the top pane of the Results window (You may need to change your column headings first).
    - Output Object: Feed
    - Data Type: Circuit Component
    - Result Type: S-Parameters

Viewing S-Parameters in the Results window



2. Double-click the result with a Domain value of Frequency to view transient S-parameter results. The following plot will appear:

Viewing S-Parameters v. Frequency plot

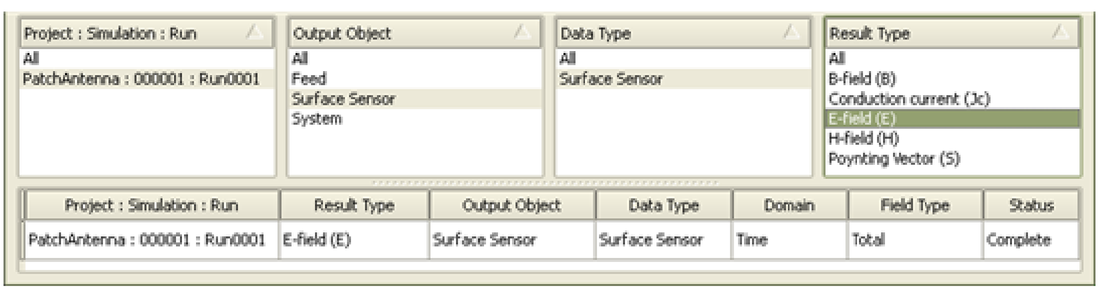


E-Field Results from the Surface Sensor

You can now view the results retrieved from the Surface Sensor.

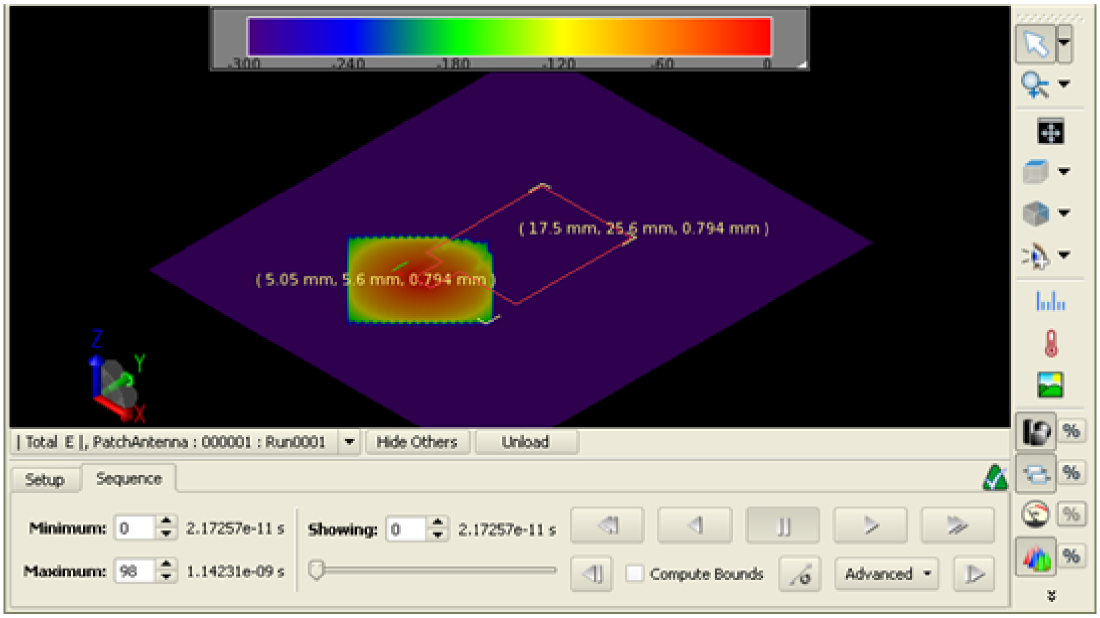
1. To filter the E-field results, select the following options:
    - Output Object: Planar Sensor
    - Data Type: Surface Sensor
    - Result Type: E-Field (E)

Viewing E-Field in the Results window



2. Double-click on the E-Field (E) result in the filtered list. The Results workspace window will appear to view the electric field time sequence.

Viewing E-Field output in the Geometry window



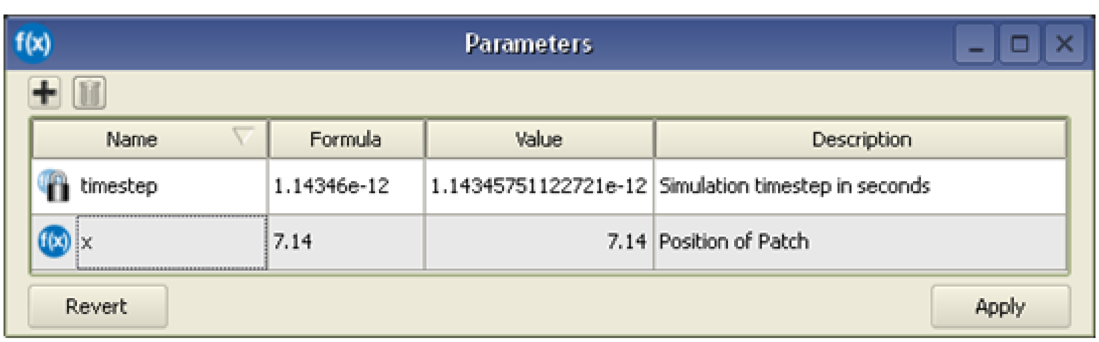
3. Navigate to the **Sequence** tab to view the results. You can play back the results as an animation or step through them with the Showing control. If you wish, change the Minimum and Maximum settings to only display a certain range of the sequence.
4. Click the **Unload** button when you are finished viewing the E-field results.

Adding a Parameter Sweep

You can parameterize your project by defining variables within the Parameters workspace window so that you can reference them in any editor or dialog window. Additionally, it incorporates the ability to perform a Parameter Sweep so that a calculation will increment the value of a variable in order to perform a calculation at every iteration.
For this patch antenna example, we will define a parameter called **x** that will control the position of the Feed and antenna Stub. Later, we will set up a parameter sweep so that the calculation engine will retrieve values for several incremented locations of the feed.

1. Open the Parameters workspace window. Click **Add** to add a new parameter.
   - Name: x
   - Formula: 7.14
   - Description: Position of Patch

Defining a global parameter

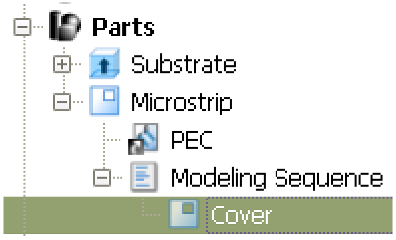


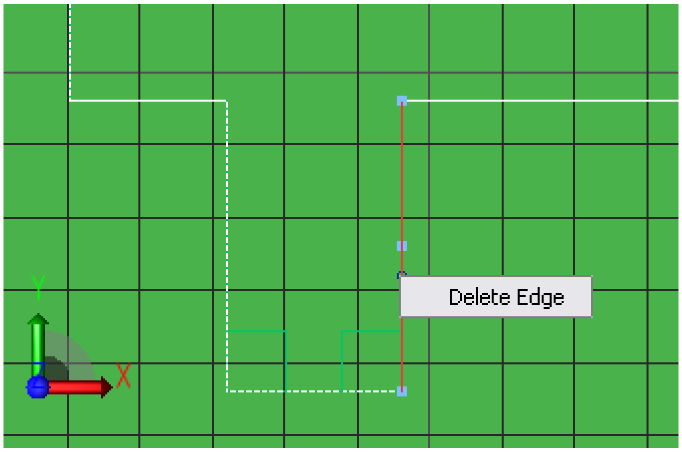
Name	Formula	Value	Description
timestep	1.14346e-12	1.14345751122721e-12	Simulation timestep in seconds
x	7.14	7.14	Position of Patch

2. Press **Apply** to add the parameter to the project.

Parameterizing the Geometry

For brevity, we originally created the antenna Microstrip in one piece. To parameterize the Stub location, we will redraw it as two separate sheet bodies.

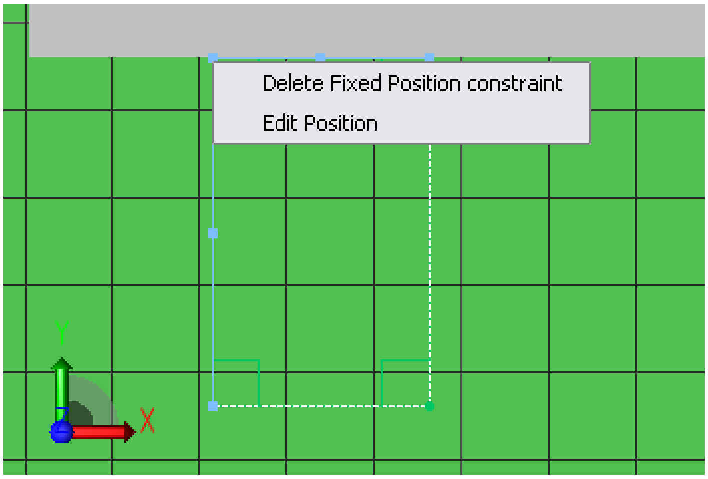
1. In the Project Tree, navigate to the **Microstrip:Modeling Sequence** branch and double-click the **Cover** object.


2. Select the **Select/Manipulate** tool at the top left of the Geometry workspace window. Right-click on an edge of the stub extension, and select Delete Edge to remove it.


3. Repeat this process to remove the other two stub edges.
4. Left-click the left endpoint in the bottom edge of the sketch. Drag it to connect with the neighboring endpoint and close the gap.
5. Click **Done** to apply your changes.

Now, we will add a Stub sheet body with a parameterized location.

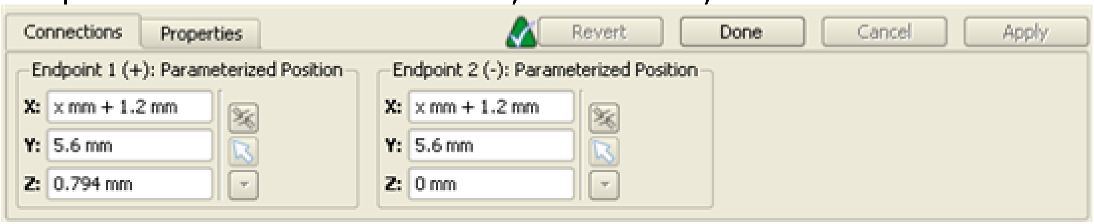
1. Right-click on the Parts branch in the Project Tree. Choose **Create New Sheet Body** from the context menu.
2. Navigate to the **Specify Orientation** tab. Set the origin to (0, 0, 0.794 mm) to place the **Sheet Body** on top of the Substrate.
3. Navigate to the **Edit Profile** tab. Type **Stub** into the Name box.
4. Select the **Rectangle** tool. Use the creation dialog to enter the corners of the stub rectangle:
    - Endpoint 1: (7.2 mm, 5.6 mm)
    - Endpoint 2: (9.6 mm, 9.6 mm)
      Because of the sketcher automatic constraint behavior, you must parameterize all four corners of the stub so it can move whenever the value of **x** changes.
5. To change the position of a vertex, choose the **Select/Manipulate** tool again and right-click the vertex. Choose **Edit Position** from the menu.



6. Set the stub rectangle corner positions as follows (be sure to add mm where necessary):
    - Upper left: (x mm, 9.6 mm)
    - Upper right:(x mm + 2.5 mm, 9.6 mm)
    - Lower left: (x mm, 5.6 mm)
    - Lower right: (x mm + 2.5 mm, 5.6 mm)
7. Click **Done** to finish the Stub geometry.
8. Assign the PEC material to the Stub by drag and dropping it onto the object.

Parameterizing the Feed

1. Locate the **Circuit Components** branch of the Project Tree, and double-click the **Feed** object to edit its position.
2. Set the endpoints of the feed as follows:
    - Endpoint 1: X: x mm + 1.2 mm, Y: 5.6 mm, Z: 0.794 mm
    - Endpoint 2: X: x mm + 1.2 mm, Y: 5.6 mm, Z: 0 mm



3. Click **Done** to finish editing the Feed.

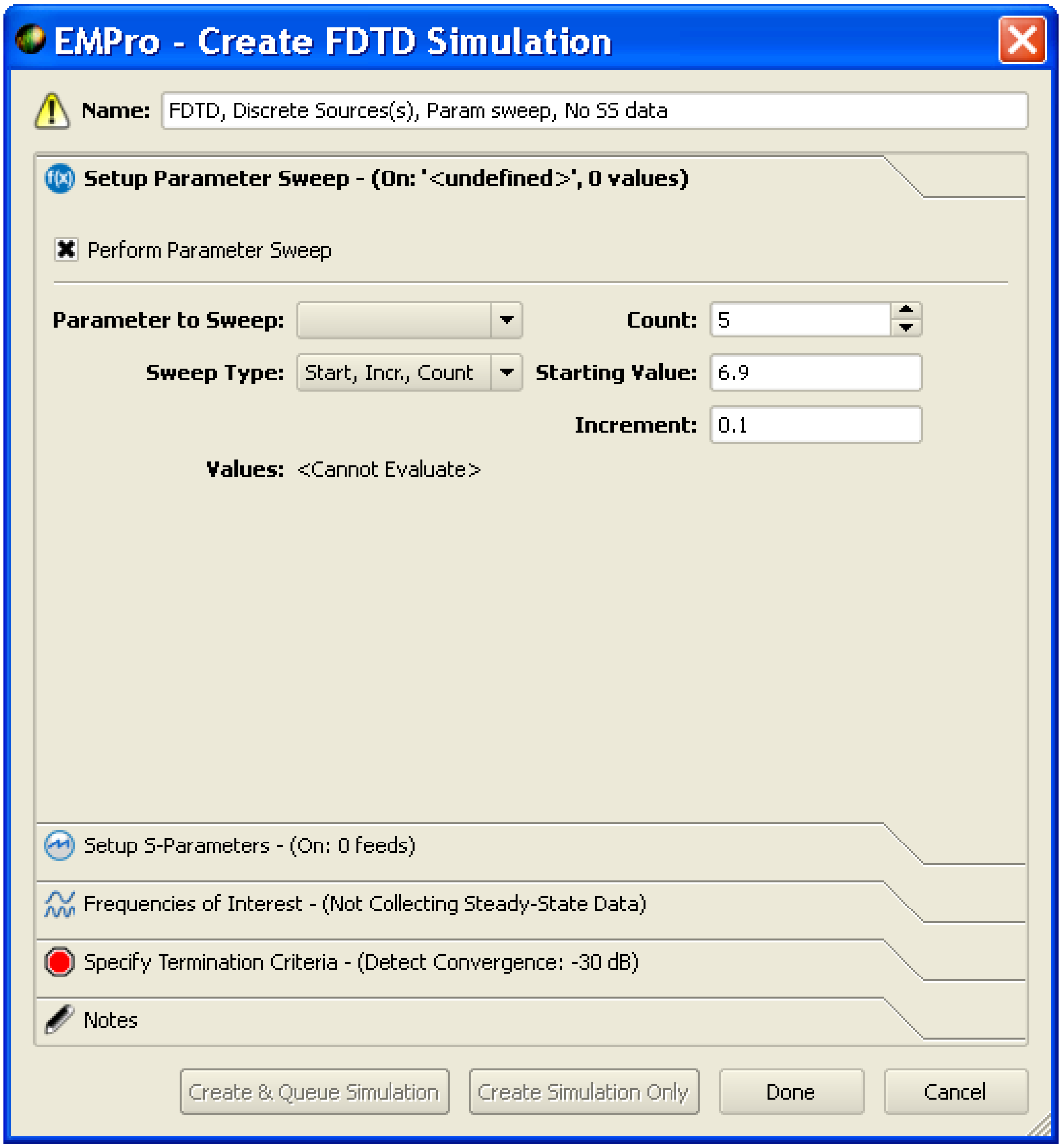
Running a Simulation with Parameter Sweep

You can save the project by selecting **File> Save Project**.

1. Open the **Simulations** workspace window. Click **New FDTD Simulation** in the upper-left to set up a new simulation. Most of the default settings are sufficient. For this simulation, you will define a parameter sweep so that the calculation engine will collect 5 sets of results, each based on an incremented value of our global parameter, **x**.
2. Navigate to the **Setup Parameter Sweep** tab. Select the **Perform Parameter Sweep** check box. Specify the following values:
    - Parameter to Sweep: x
    - Sweep Type: Start, Incr, Count

- Count: 5
- Starting Value: 6.9
- Increment: 0.1

Adding the parameter to the simulation

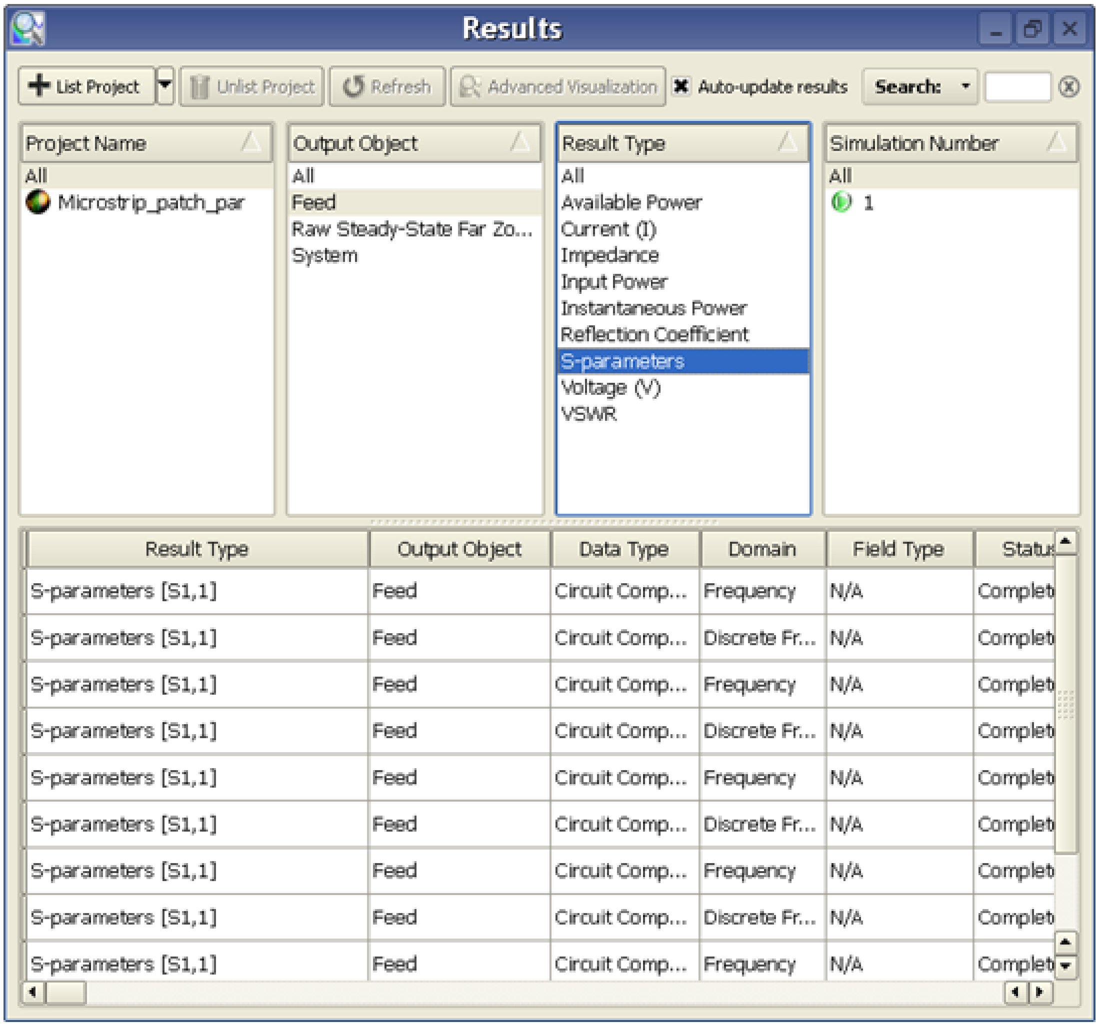


As shown in the Values section, these settings will produce a range of values from 6.9 to 7.3.

3. Click **Create & Queue Simulation** to run this simulation.

Viewing Results of the Parameter Sweep

After the Status column in the Simulations workspace window shows that the simulation has completed, you can view its results in the Results window. Under the **Project:Simulation:Run** column, notice that within the simulation, a new run is created for each parameter value.

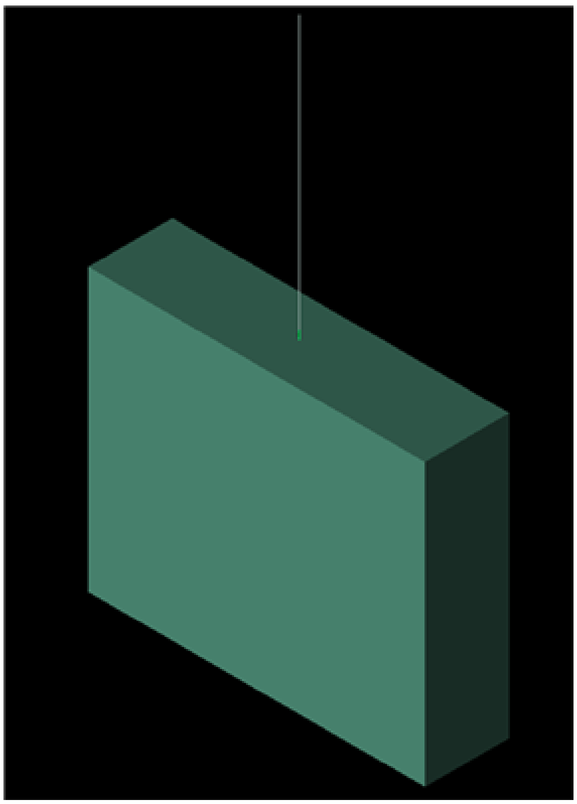

Results
List Project
Unlist Project
Refresh
Advanced Visualization
Auto-update results
Search:
Project Name
All
Microstrip_patch_par
Output Object
All
Feed
Raw Steady-State Far Zo...
System
Result Type
All
Available Power
Current (I)
Impedance
Input Power
Instantaneous Power
Reflection Coefficient
S-parameters
Voltage (V)
VSWR
Simulation Number
All
1
Result Type
Output Object
Data Type
Domain
Field Type
Statu
S-parameters [S1,1]
Feed
Circuit Comp...
Frequency
Discrete Fr...
N/A
Complet

Creating a Monopole Antenna on a Conducting Box Simulation

In this project, you will learn how to:

- Build a monopole antenna using solid modeling techniques.
- Define the properties of the antenna environment.
- Add a feed to the antenna and simulate its effects.
- Add a surface sensor to the box and view the calculated surface current.
- Retrieve far zone results after running the calculation.

In this project, a wire monopole is connected to a conducting box and fed at the junction. The radiation pattern is calculated at a frequency of 1.47GHz.



Getting Started

This section briefly describes how to configure the display units for the Monopole Antenna project.

> **Note**
> To set up a project for the first time, refer to Application Preferences Appendix for instructions about how to configure project preferences and navigate through the display units tab.

In the **Project Properties Editor** window, navigate to the Display Units tab:

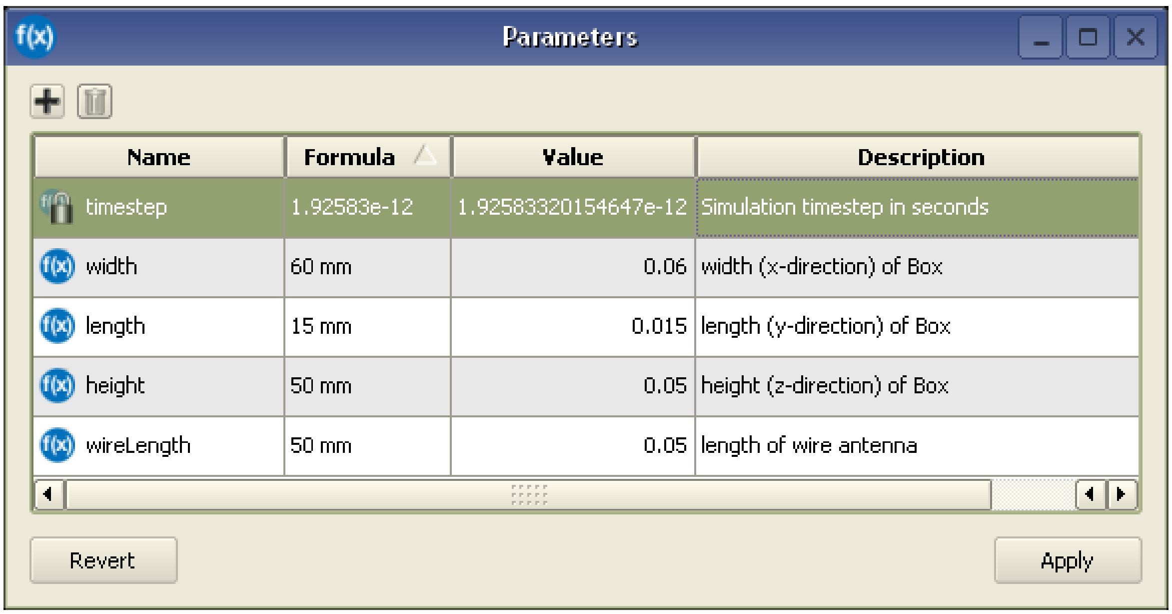
1. Select **SI Metric** in the **Unit Set** drop-down list.

2. Change **Length** to **millimeters (mm)**. This changes the value of Unit Set to **Custom**.
3. Click **Done**.

Parameterizing the Project

In this example, you will parameterize the dimensions of the monopole antenna geometry so that any value can be easily changed in the **Parameters** browser window.

1. Open the **Parameters** workspace window. Click **Add** to add a new parameter.
   - **Name**: width
   - **Formula**: 60 mm
   - **Description**: width (x direction) of Box
2. Add parameters named length, height, and Length in the same manner.
3. Click **Apply** to add the parameters to the project.



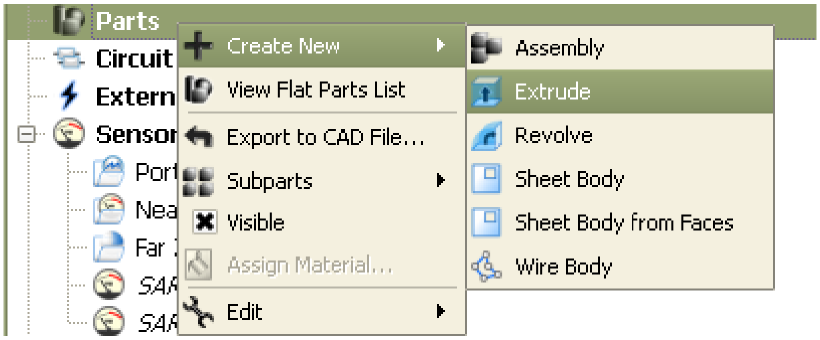
Creating the Monopole Antenna Geometry

The Monopole Antenna geometry is created with a simple Box and a Monopole antenna. The dimensions of the Box and antenna offset location is defined with parameters.

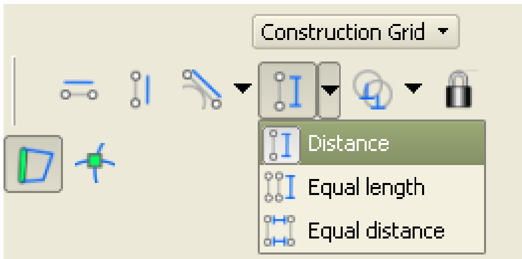
Modeling the Box

First, you need to create the rectangular substrate named Box. This object will use the parameters length, width, and height for its dimensions with an extrusion in the $+Z$ direction.

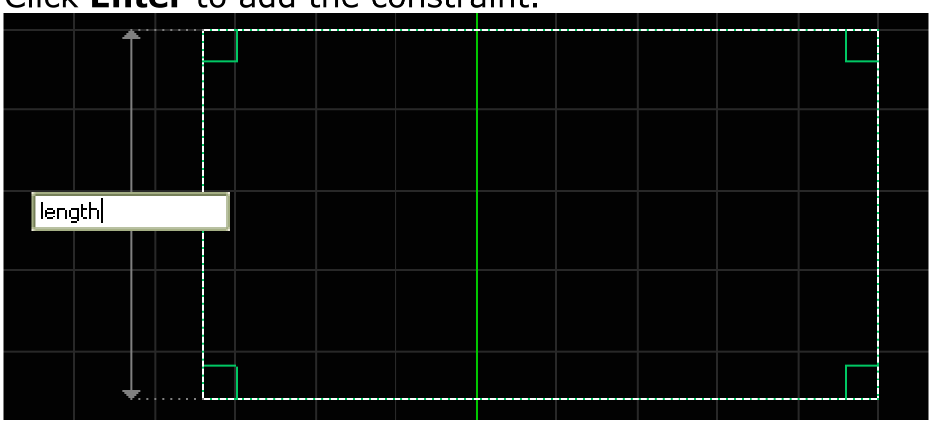
1. Right-click on the **Parts** branch of the Project Tree. Choose **Create New>Extrude** from the **Context** menu.



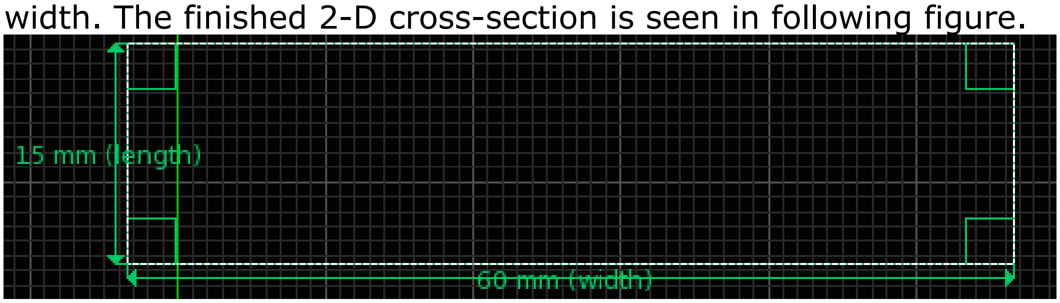
2. Type **Box** into the Name text box.
3. Choose the **Rectangle** tool from the **Shapes** toolbar, and draw a rectangle in the sketching plane. (Dimensions are not important).
4. Select the **Distance** from the Constraints toolbar.



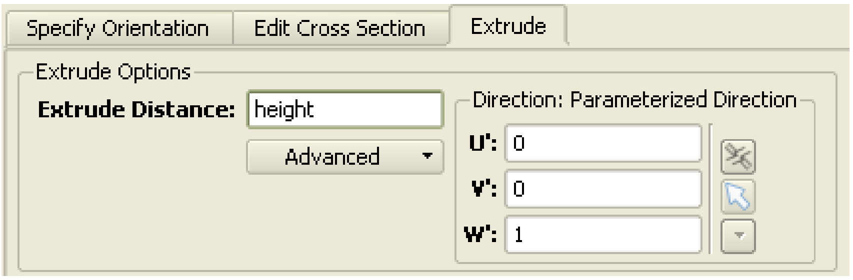
- Click on the left vertical side of the rectangle (it will turn blue), then move the mouse slightly to its left and click again.
- Type **length** in the dialog box, to set its value equal to the length parameter.
- Click **Enter** to add the constraint.



5. Add a constraint to the bottom horizontal side of the rectangle, defining its value as width. The finished 2-D cross-section is seen in following figure.



6. Navigate to the **Extrude** tab to extrude the rectangular region. Enter **height** as the distance, in the _Z direction.



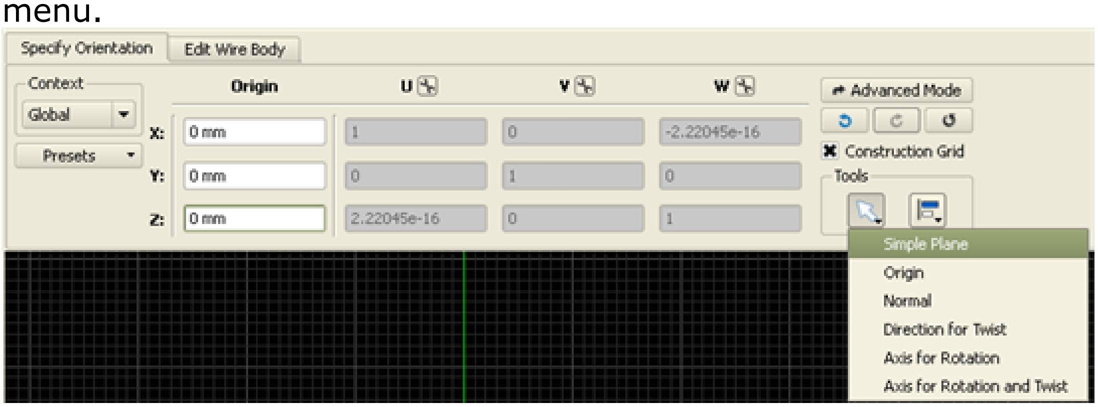
7. Click **Done** to complete the Box geometry.

Now the dimensions of the Extrude Box are completely parameterized and adjustable from the Parameters workspace window.

Modeling the Monopole

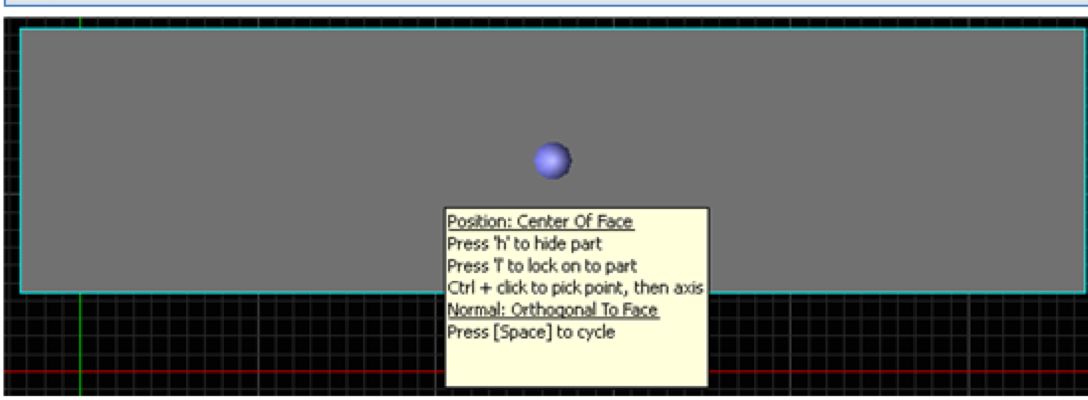
The Monopole will be created with a **Wire Body** object that is locked to the top center of the Extrude Box. Its length will be defined by the parameter Length.

1. Right-click the **Parts** branch of the Project Tree. Choose **Create New> Wire Body** from the context menu.
2. In the **View Tools** toolbar, select the **Top (-Z)** orientation.
3. Navigate to the **Specify Orientation** tab and select Simple Plane from the **Pick** menu.

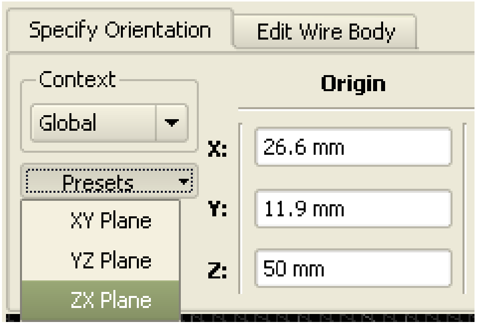


- Place the mouse over the face of the box and click c to center the Origin. Click on this location to set the values.

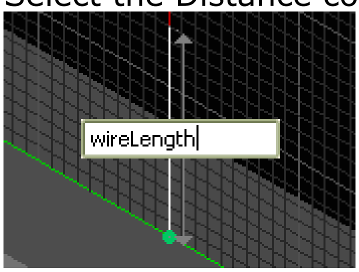
> **Note**
> If the Origin does not move to the center as expected, right-click in the geometry space to activate the window.



4. Redefine the orientation of the sketching plane by selecting the **ZX** Plane under the **Presets** drop-down list.



5. Navigate to the **Edit Wire Body** tab. Type **Monopole** into the Name box.
6. In the **View Tools** toolbar, select the **Front/Right/Top** orientation.
7. Select the **Straight Edge** tool.
    - Click on the origin (where the green and red axes intersect) to place the first point of the wire antenna.
    - Click the second point anywhere along the axis directed normal to the plane of the box.
8. Select the **Select/Manipulate** tool at the top left of the **Geometry** workspace window. Right-click on the end of the wire at the origin and select Lock Position.
9. Select the Distance constraint tool to constrain the length of the wire as wireLength.

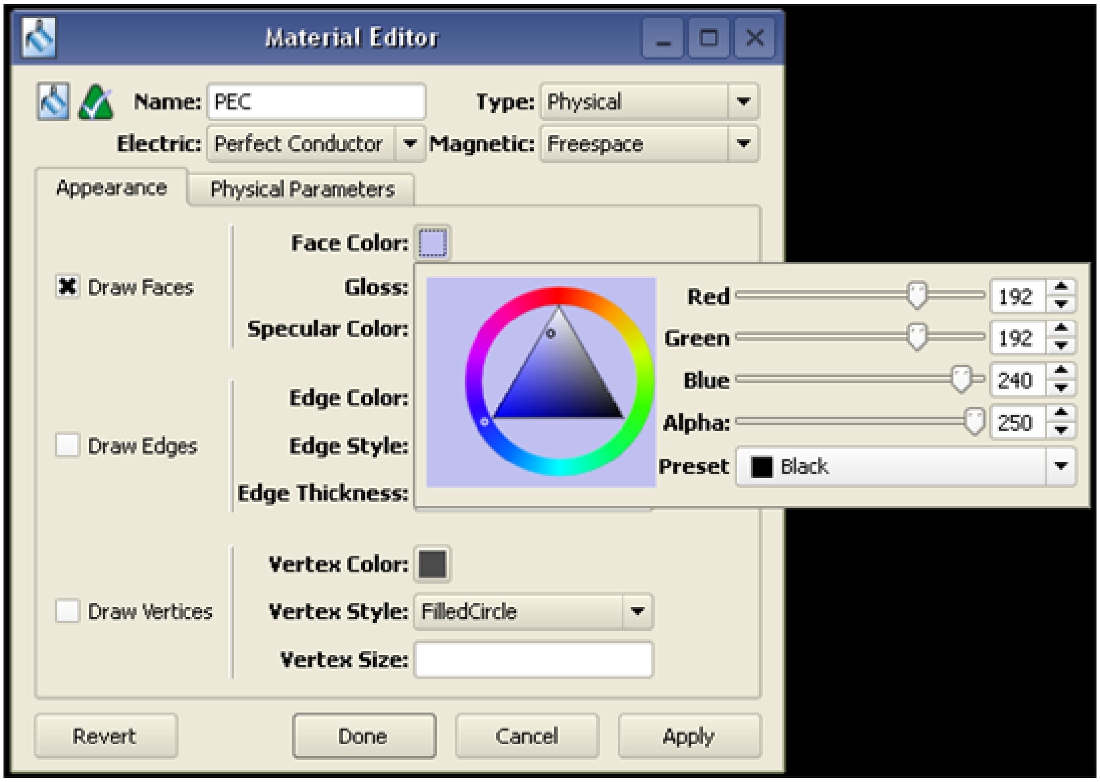


10. Click **Done** to finish the Monopole geometry.

Creating Materials

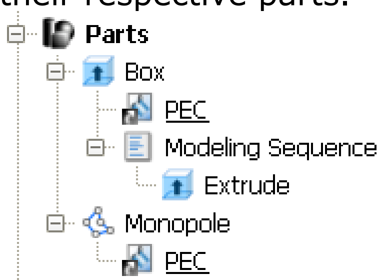
Define Material, PEC

1. Right-click on the **Definitions:Materials** branch of the Project Tree and select **New Material Definition** from the context menu.
2. Set the perfect electric conductor material properties as follows:
    - **Name**: PEC
    - **Electric**: Perfect Conductor
    - **Magnetic**: Freespace
3. If desired, navigate to the **Appearance** tab to set the PEC material display color.

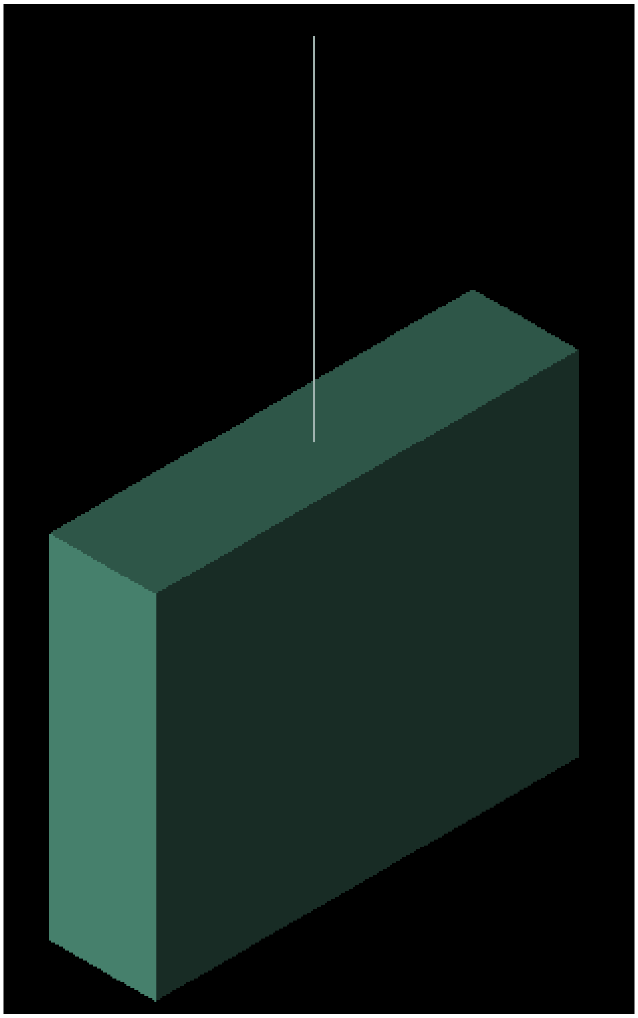


Assigning Materials

Click-and-drag the PEC material object located in the Project Tree and drop it on top of the Monopole and Extrude Box objects.
The following image shows the Project Tree after material objects have been dropped on their respective parts.



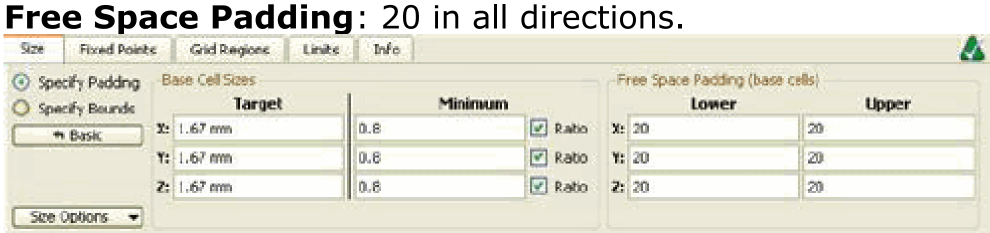
This image shows the monopole box geometry with materials applied and colors set for each.



Creating the Grid

Now, you can define the characteristics of the cells in preparation to perform an accurate calculation.

1. Double-click the **FDTD:Grid** branch to open the **Grid Tools** dialog box.
2. Set the **Size** properties of the grid as follows:
    - **Base Cell Sizes**: Target 1.67 mm, Merge 0.8, Ratio boxes checked
    - **Free Space Padding**: 20 in all directions.



3. Click **Done** to apply the grid settings.

Adding fixed points to the geometry

1. In the **Parts** branch of the Project Tree, right-click on the Extrude Box object and select **Gridding** Properties to open the **Gridding Properties Editor**.
    - Select the **Use Automatic Fixed Points** checkbox.

2. Click **Done** to close the editor.

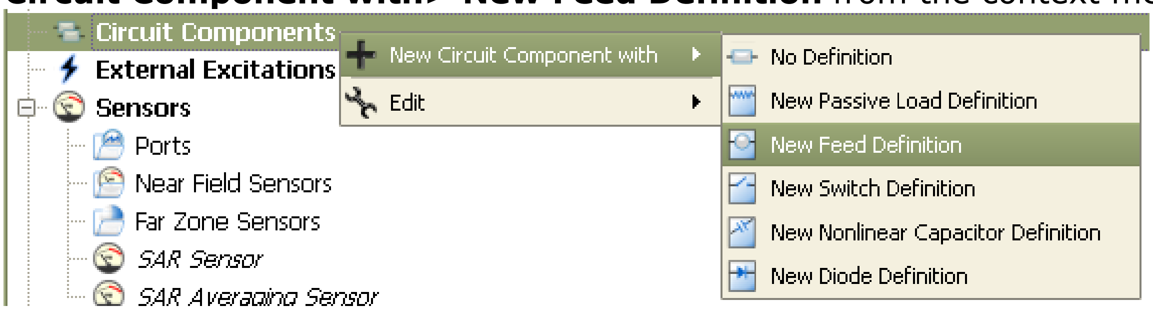
Creating a Mesh

In the **FDTD** branch of the Project Tree, double-click on the **Mesh** icon. This will bring up the mesh view and automatically create the mesh.

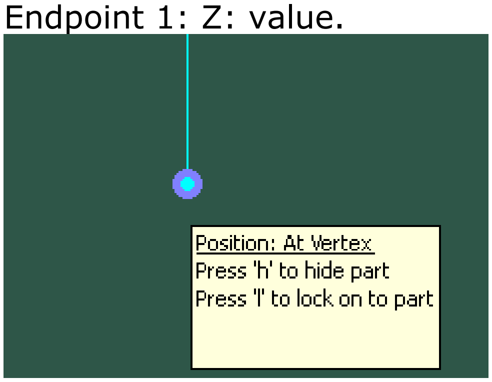
Adding a Feed

You will now add a **Feed** to the monopole geometry at the base of the **Monopole** antenna. The feed will consist of a voltage source and series 50Ω resistor connected at the base of the Monopole. Then, apply a **Ramped Sinusoid** waveform to the circuit through this feed.

1. Right-click on the **Circuit Components** branch in the Project Tree. Choose **New Circuit Component with> New Feed Definition** from the context menu.

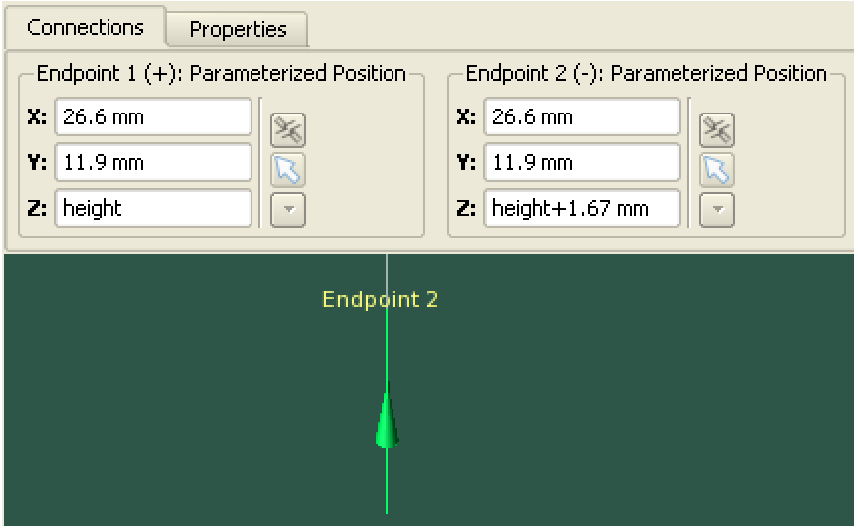


2. Define the endpoints of the feed.
   - Endpoint 1: Zoom in to the area where the Monopole meets the Box. Using the **Pick** tool, click the point at the base of the wire. Then type in height for the Endpoint 1: Z: value.



   - Endpoint 2: Select the **Pick** tool under Endpoint 2, and click a higher location along the wire. Edit Endpoint 2: Z: to be height+1.67 mm. The X: and Y: locations should be the same as the values for Endpoint 1. Your X: and Y: locations may differ from the figure since it is an arbitrary rectangular sketch in the XY plane.

> **Note**
> The **Feed** is the same length as a grid cell, as defined in Creating the Grid.

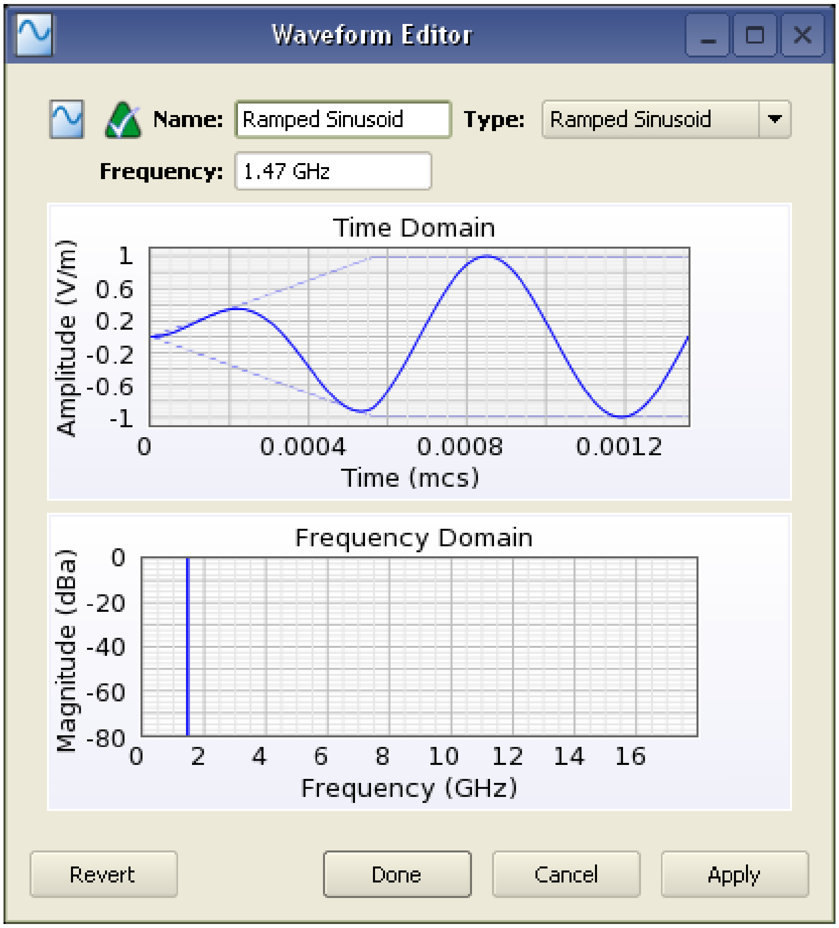


3. Navigate to the **Properties** tab, and enter the following:
    - Name: Feed
    - Component Definition: 50 ohm Voltage Source
    - Polarity: Positive
    - Select the box labeled **This component is a port**.
4. Click **Done** to add the Feed.

Editing the Waveform

An associated waveform was automatically created for the feed definition.

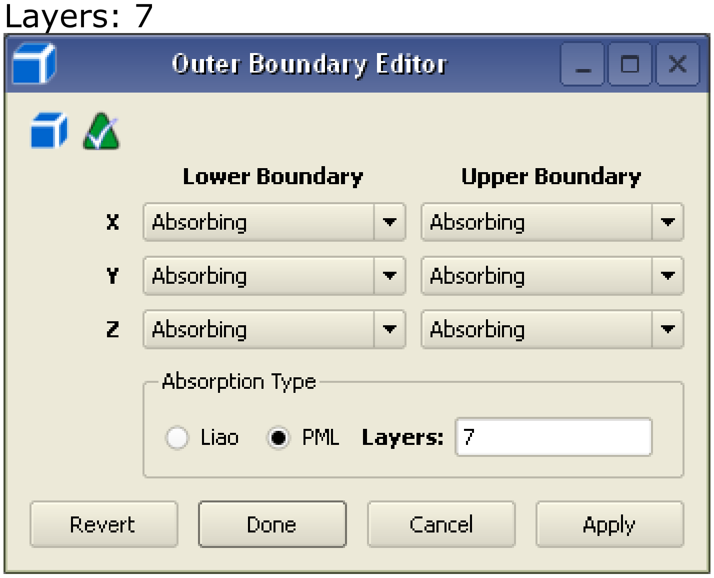
1. Navigate to the **Definitions:Waveforms** branch of the Project Tree. Double-click on the **Broadband Pulse** waveform to edit its properties.
2. Set the properties of the waveform as follows:
    - Name: Ramped Sinusoid
    - Type: Ramped Sinusoid
    - Frequency: 1.47 GHz



3. Click **Done** to apply the changes.

Defining the Outer Boundary

1. Double-click the **Simulation Domain: Boundary Conditions** branch of the Project Tree to open the Boundary Condition Editor.
2. Set the outer boundary properties as follows:
   - Boundary: Absorbing for all boundaries
   - Absorption Type: PML
   - Layers: 7



3. Click **Done** to apply the outer boundary settings.

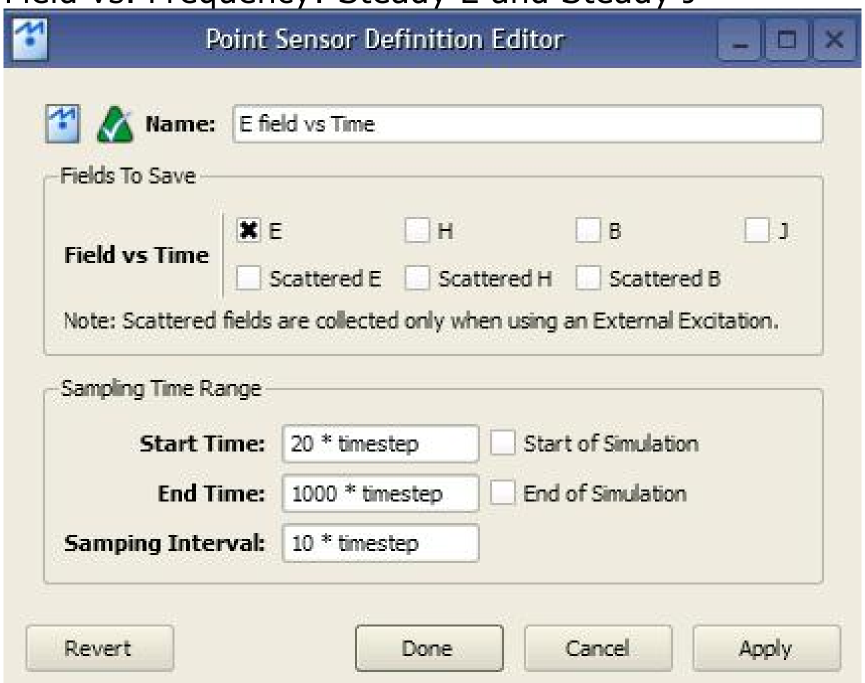
Requesting Output Data

The project already contains one port sensor named **Feed** that will request results. You may also want to collect field samplings at discrete frequencies throughout the calculation. To retrieve this data, add a **Surface Sensor**.

Adding a Surface Sensor Definition

First, create the **Surface Sensor Definition**. Right-click on the **Definitions:Sensor Data Definitions** branch of the Project Tree. Choose **New Surface Sensor Definition** from the context menu.

1. Set the properties of the surface sensor definition as follows:
    - Name: Field Sampling
    - Field vs. Frequency: Steady E and Steady J



2. Click **Done** to finish editing the Field Sampling definition.

Adding a Surface Sensor

1. Right-click the **Sensors:Near Field Sensors** branch of the Project Tree. Select **New Sensor on Model Surface** from the context menu.
2. From the Select Model tab, use the **Select** tool (at the top of the **View** Tools menu) and double-click on the box. You will know that the box is selected when it changes color.
    - Under the Properties tab, enter the following:
        - Name: Surface Sensor
        - Sensor Definition: Field Sampling
        - Sampling Method: Snapped to E-Grid
3. Click **Done** to finish editing the Surface Sensor.

Running the Calculation

If you have not already saved your project, do so by choosing **File>Save Project**. After saving the project, you can create a new simulation to send to the calculation engine.

1. Open the **Simulations** workspace window. Click **New FDTD Simulation** to set up a new simulation.
2. Under **Frequencies of Interest**, select the **Collect Steady-State Data** check box.
   - Under the **Frequencies** tab, select **Use Waveform Frequency**.
3. Select **Create and Queue Simulation** to close the dialog and run the new simulation.

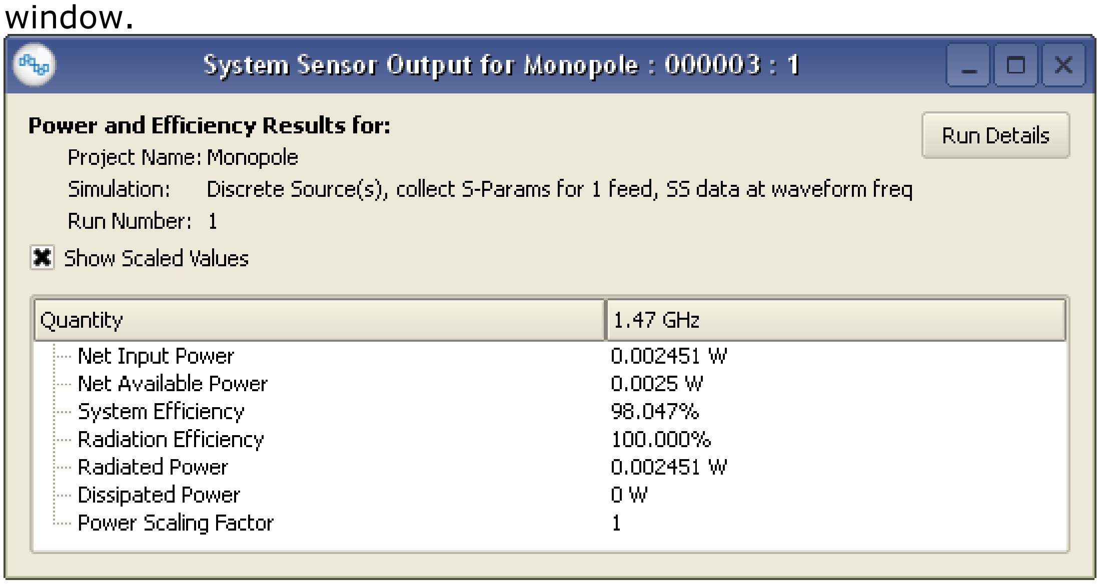
Viewing the Results

The **Output** tab of the **Simulations** workspace window displays the progress of the simulation. Once the **Status** column shows that the simulation has completed, you can view its results in the **Results** workspace window.

System Efficiency Results

First, you will view the System results.

1. To filter the list accordingly, select the following options in the columns in the top pane of the **Results** window. (You may need to change your column headings first.)
   - Output Object: System
   - Result Type: System Efficiency
2. Double-click the result. A list of power and efficiency results will appear in a dialog window.

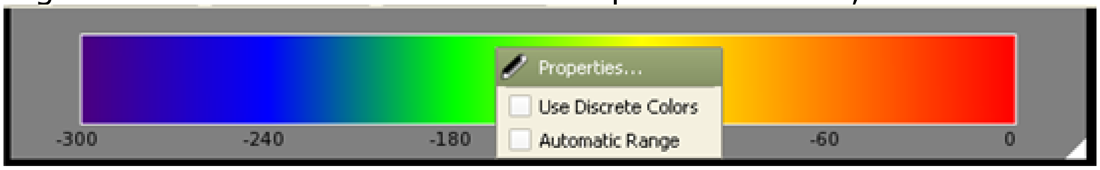


You can close the window when you are finished viewing the results.

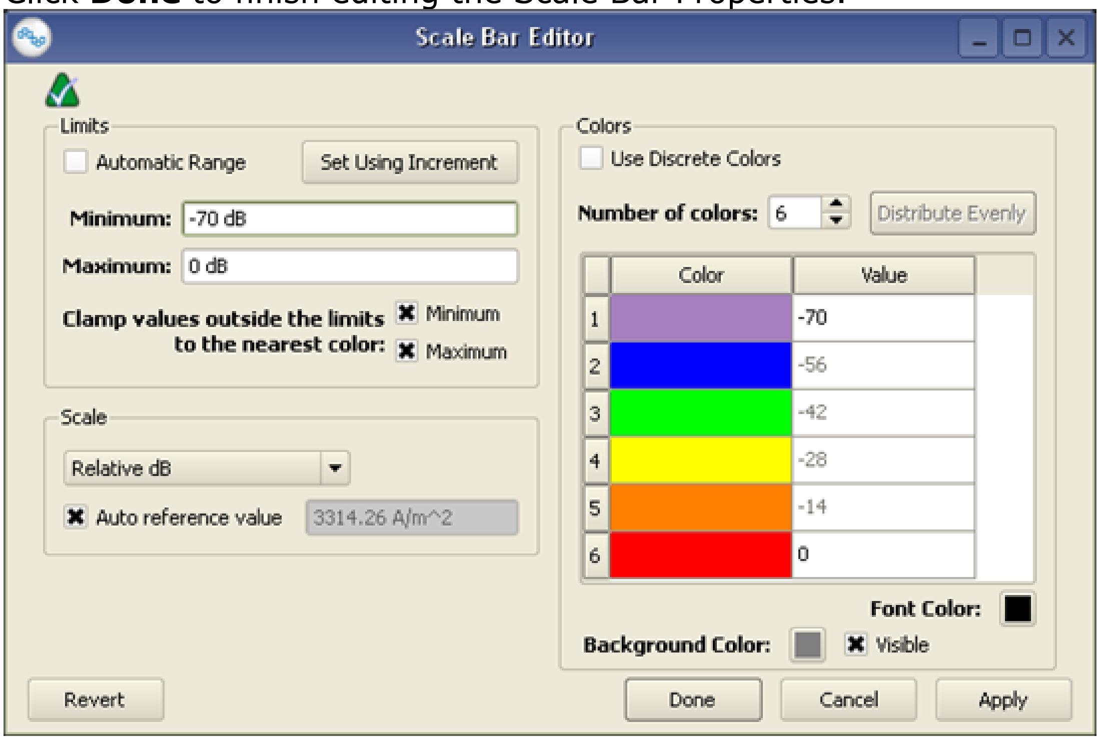
Surface Current Results from the Surface Sensor

Now you can view the surface current results retrieved with the **Surface Sensor** placed on the Extrude Box.

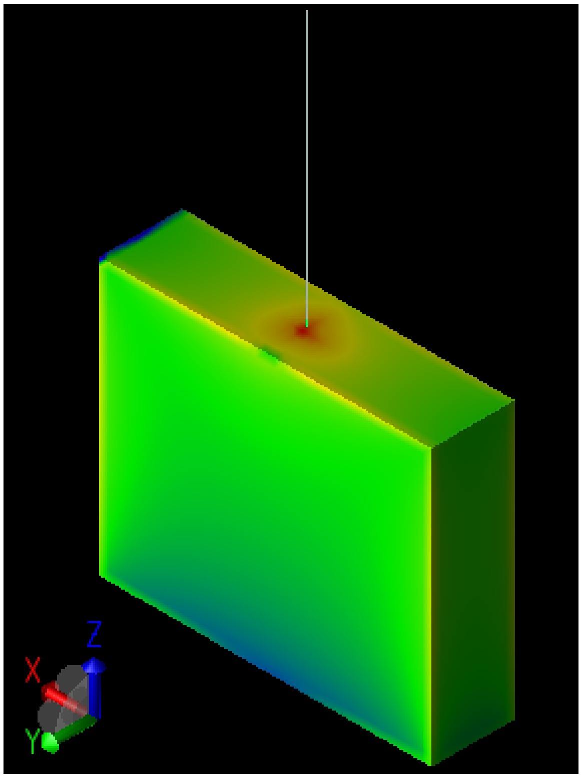
1. Select the following options in the **Results** window:
    - Output Object: Surface Sensor
    - Result Type: Conduction Current (Jc)
2. Double-click on the result. The plot will appear in the **Geometry** workspace window.
3. Right-click on the **Scale Bar** at the top of the screen, and select **Properties**.



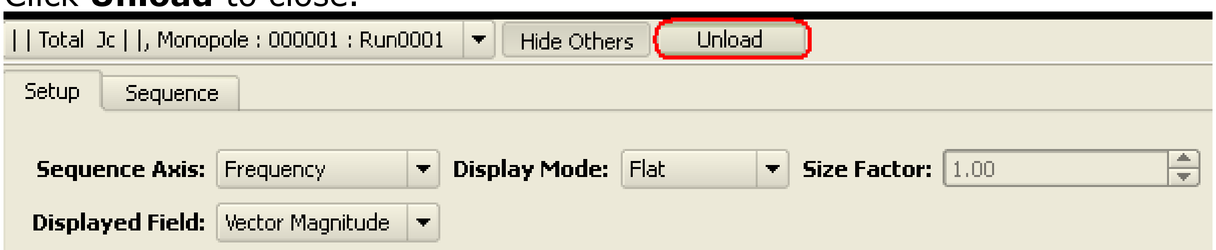
    - Under the Limits Section, remove selection from the **Automatic Range** box.
    - Set the Minimum to -70 dB.
4. Click **Done** to finish editing the Scale Bar Properties.



5. You can see the surface current data on the surface of the box, as shown in the following figure:

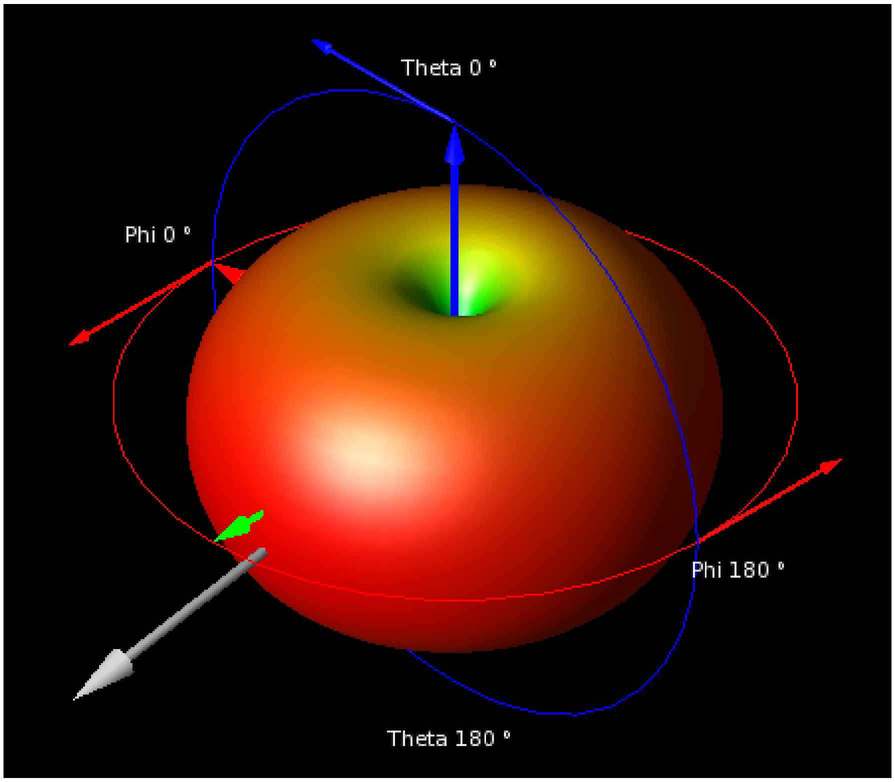


6. Click **Unload** to close.



Far Zone Post Processing

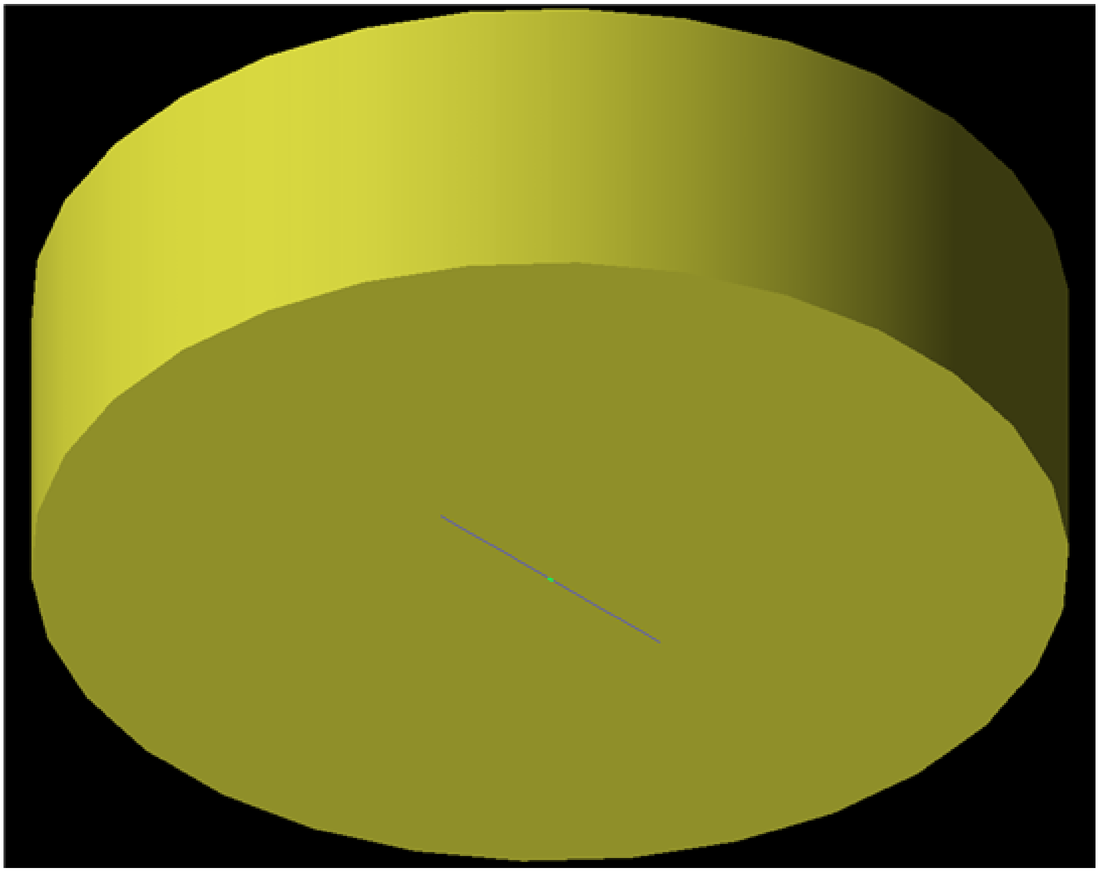
1. To begin the far zone post-processing, select the following:
   - Output Object: Raw Steady-State Far Zone Data
   - Result Type: E-Field (E)
2. Right-click the E-Field (E) result in the filtered list, and select **Post-Process Results**. The **Results** workspace window will appear to set the properties of the Far Zone sensor.
3. The default definition is sufficient for this calculation. Click Done to begin the steady state far zone data transform.
4. Select the following options in the Results window:
   - Output Object: Post Processed
   - Result Type: Gain
5. Double-click the result. The plot will appear in the Geometry workspace window.
6. Right-click on the Scale Bar at the top of the screen, and select Properties.
   - Under the Limits section, uncheck the Automatic Range box. Set the Minimum to -70 dB and the Maximum to 0 dB.
   - Under the Scale section, select Relative dB and check the the Auto reference value box.
   - Click **Done** to finish editing the Scale Bar Properties.
   The following figure displays the Far zone data:


Theta 0 °
Phi 0 °
Phi 180 °
Theta 180 °

Creating a Simple SAR Calculation Simulation

In this project, you will learn how to:

- Model a tissue with a dipole.
- Define the properties of the environment.
- Add a feed to the dipole and simulate its effects.
- Add a point sensor and measure E-field at the center of the tissue.
- Add an SAR sensor and retrieve SAR data.



Getting Started

This section describes how to configure the display units for the SAR project.

> **Note**
> To set up a project for the first time, refer to Application Preferences Appendix for instructions about how to configure project preferences and navigate through the display units tab.

In the **Project Properties Editor** window, navigate to the **Display Units** tab:

1. Select **SI Metric** in the **Unit Set** drop-down list.
2. Change **Length** to **millimeters (mm)**. This changes the value of Unit Set to **Custom**.

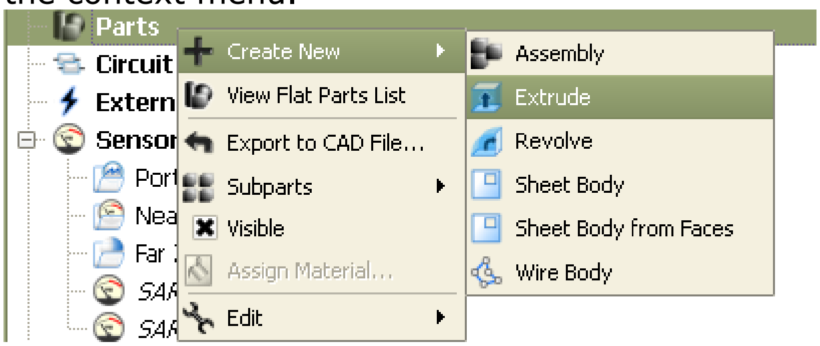
3. Click **Done**.

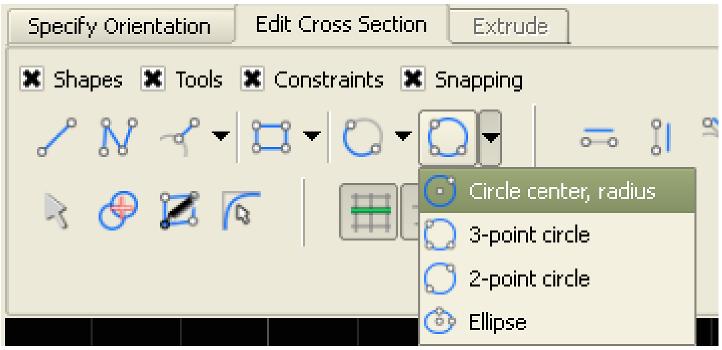
Creating the SAR Geometry

The geometry for this simple SAR calculation consists of a cylinder and a dipole antenna. The cylinder, named Tissue, will be modeled with a simple **Extrusion**. The dipole antenna, named **Dipole**, will be modeled with a **Wire Body**.

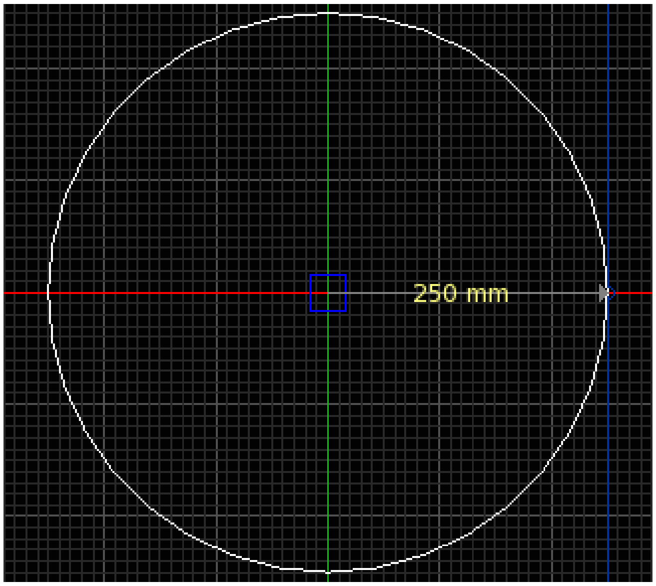
Modeling the Tissue

You will create Tissue with a cylindrical Extrusion with a radius of 250mm in the +Z direction.

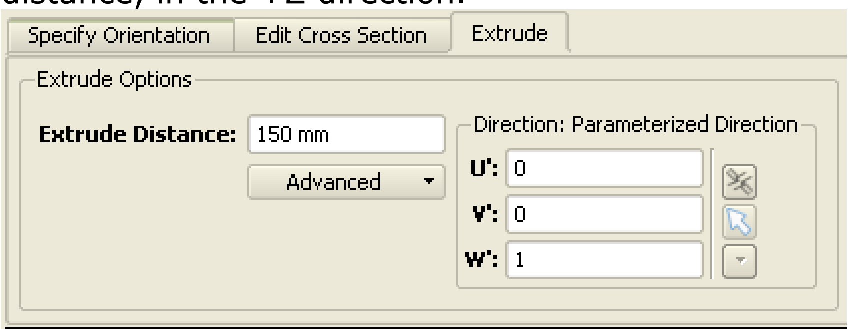
1. Right-click the **Parts** branch of the Project Tree. Choose **Create New>Extrude** from the context menu.


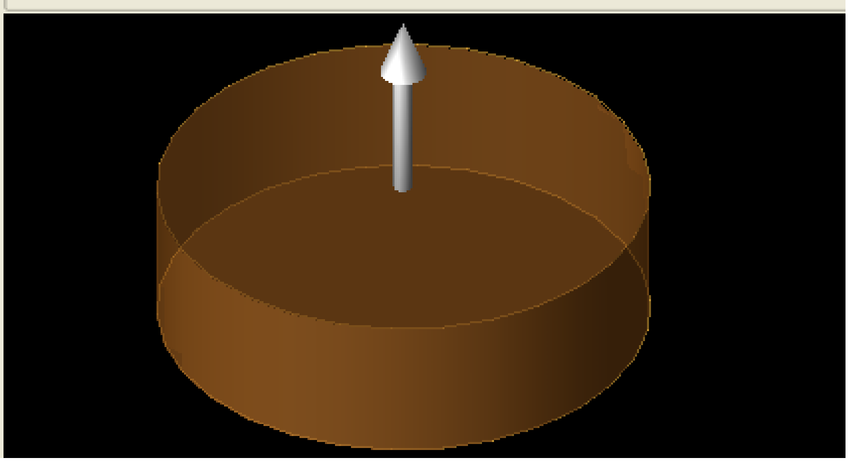
2. Enter **Tissue** as the Name.
3. Click the **Construction Grid** button.
   - Set the minor grid Line spacing to 10 mm.
   - Set Mouse spacing to increments of 0.1 mm.
   - Click **Ok**.
4. Choose the **Circle center** radius tool from the **Shapes** toolbar.


5. Draw a circle by clicking the point (0, 0, 0) and then clicking on the point (250 mm, 0, 0).



6. Navigate to the **Extrude** tab to extrude the cylindrical region. Enter 150 mm as the distance, in the +Z direction.





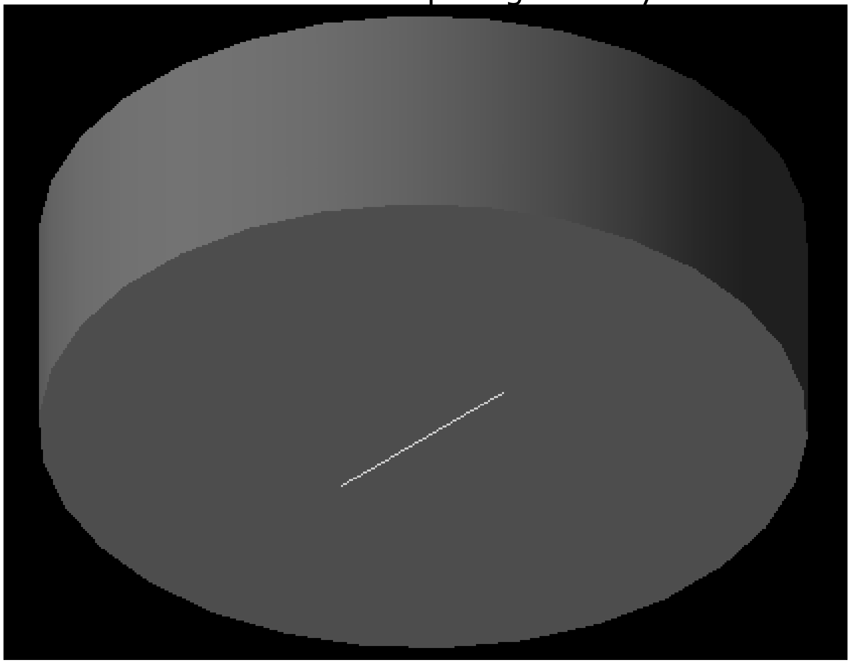
7. Click **Done** to complete the Tissue geometry.

Modeling the Dipole

The Dipole will be created with a 100mm Wire Body object that is centrally located just under the Tissue cylinder.

1. Right-click the **Parts** branch of the Project Tree. Choose **Create New>Wire Body** from the context menu.
2. Under the **Specify Orientation** tab, set the origin to (0, 0, -10 mm).
3. Navigate to the **Edit Wire Body** tab. In the Name box, type **Dipole**.
4. Select the **Straight Edge** tool.

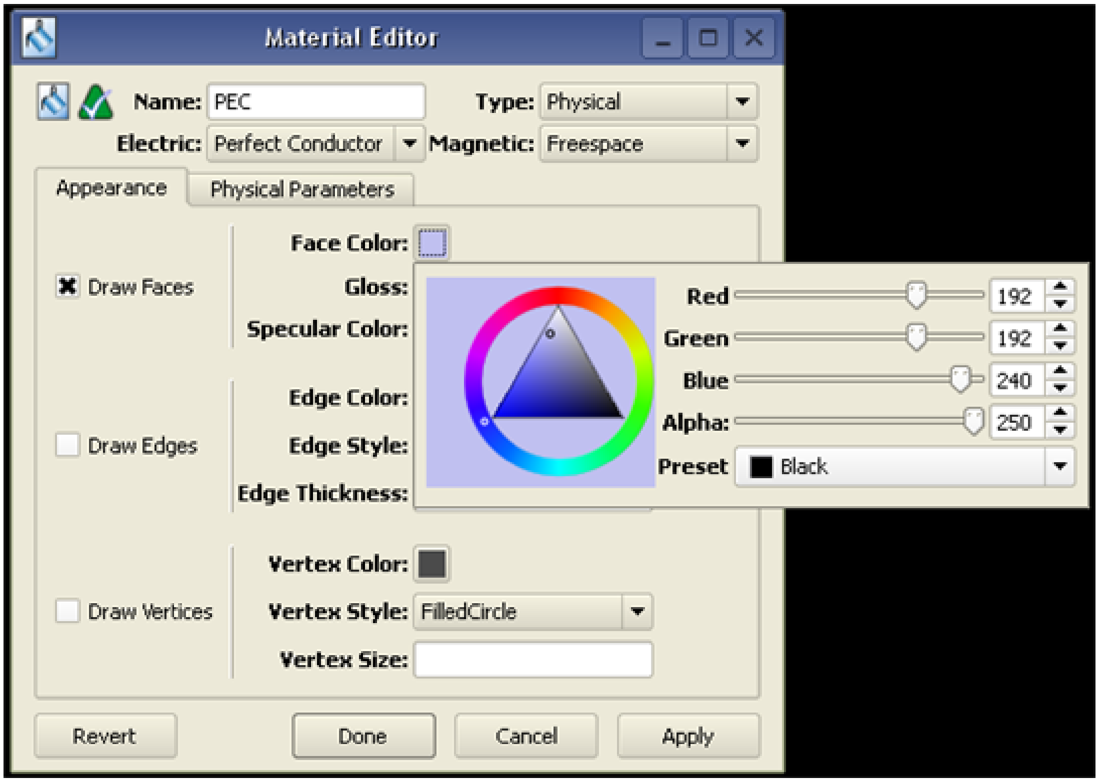
5. In the **View Tools** toolbar, select the Left (+X) orientation.
    - Press the **Tab** key to display the creation dialog for the first point. Enter (-50 mm, 0 mm) and click Ok.
    - Press the **Tab** key to display the creation dialog for the second point. Enter (U : 50, V : 0, Length :100 mm) and click **Ok** to complete the Dipole.
6. Click **Done** to finish the Dipole geometry.



Creating Materials

Define material, PEC

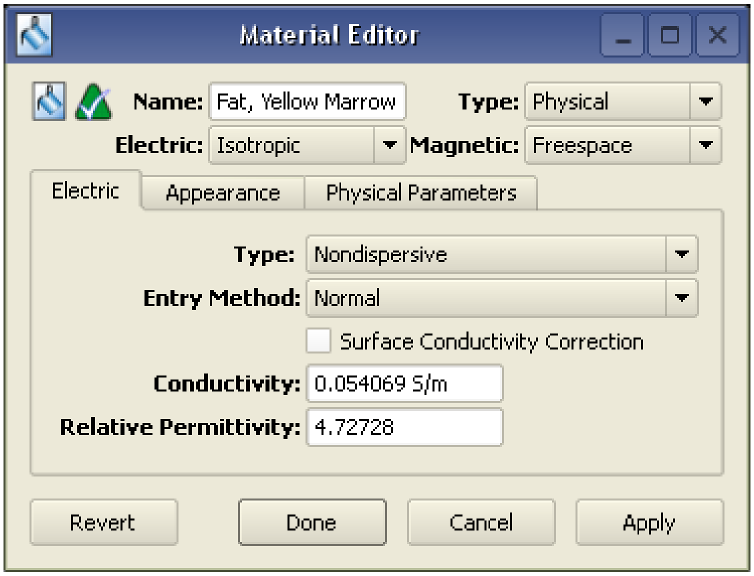
1. Right-click the **Definitions:Materials** branch of the Project Tree. Choose **New Material Definition** from the context menu.
2. Set the perfect electric conductor material properties as follows:
    - **Name**: PEC
    - **Electric**: Perfect Conductor
    - **Magnetic**: Freespace



3. If required, navigate to the **Appearance** tab to set the PEC material display color.

Defining Material as Fat, Yellow Marrow

1. Right-click the **Definitions:Materials** branch of the Project Tree. Choose **New Material Definition** from the context menu.
   Set the material properties as follows:
   - **Name**: Fat, Yellow Marrow
   - **Electric**: Isotropic
   - **Magnetic**: Freespace
     Under the Electric tab:
   - **Type**: Nondispersive
   - **Entry Method**: Normal
   - **Conductivity**: 0.054069 S/m
   - **Relative Permittivity**: 4.72728



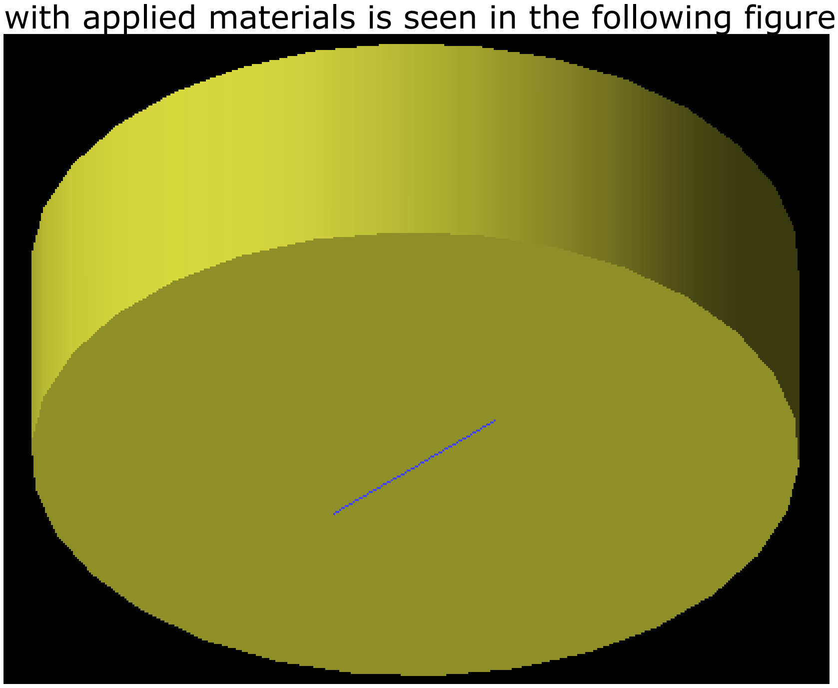
2. Under the **Physical Parameters** tab, enter 943 kg/m^3 as the Density.

> **Note**
> If the FEM option is enabled in EMPro, you will not be able to enter the value of Density. You need to disable FEM to enter the density value.

3. Navigate to the **Appearance** tab and assign the Fat, Yellow Marrow material a new color to distinguish it from PEC.
4. Click **Done** to add the new material Fat, Yellow Marrow.

Assigning Materials

1. Click and drag the PEC material object located in the Project Tree and drop it on top of the Dipole object in the Parts branch of the tree.
2. Assign the Fat, Yellow Marrow material to the Tissue object. The finished geometry with applied materials is seen in the following figure.



Creating the Grid

Now, you can define the characteristics of the cells in preparation to perform an accurate calculation.

1. Double-click the **FDTD:Grid** branch of the Project Tree to open the **Grid Tools** dialog box.
2. Navigate to the **Size** tab:
   - Click the **Advanced** button (if necessary) to access Base Cell Sizes and Free Space Padding. Set the Size properties of the grid as follows:
   - **Base Cell Sizes**: Target 1.67mm, Minimum 0.5, Ratio boxes checked
   - **Free Space Padding**: 20 in all directions

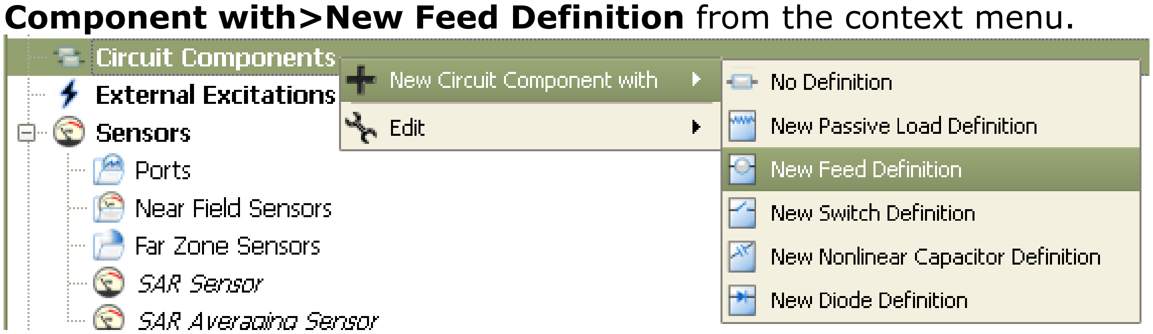
3. Click **Done** to apply the grid settings.

Creating a Mesh

In the **FDTD** branch of the Project Tree, double-click **Mesh**. This opens the mesh view and automatically create the mesh.

Adding a Feed to the Dipole Wire

1. Right-click the **Circuit Components** branch of the Project Tree. Choose **New Circuit Component with>New Feed Definition** from the context menu.

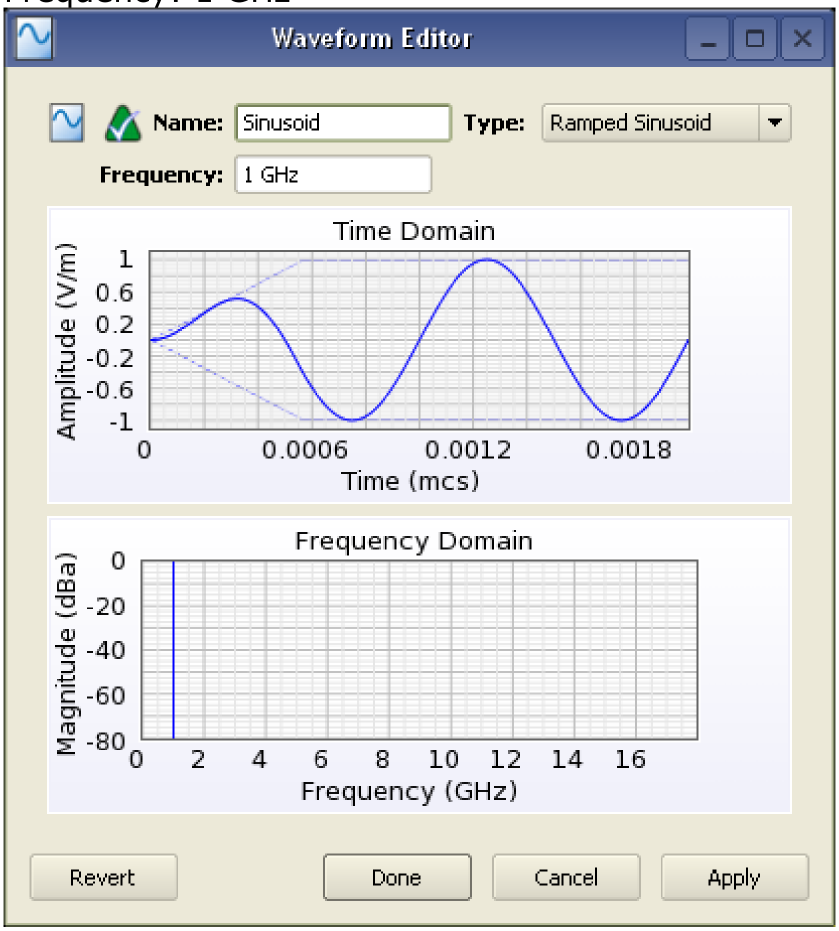


2. Define the endpoints of the feed.
   - Endpoint 1: X: -0.5 mm, Y: 0 mm, Z: -10 mm
   - Endpoint 2: X: 0.5 mm, Y: 0 mm, Z: -10 mm
3. Navigate to the **Properties** tab, and enter the following:
   - **Name**: Feed
   - **Component Definition**: 50 ohm Voltage Source
   - **Direction**: Auto
   - **Polarity**: Positive
   - Select the checkbox labeled **This component is a port**.
4. Click **Done** to add the Feed.

Editing the Waveform

An associated waveform was automatically created for the feed definition.

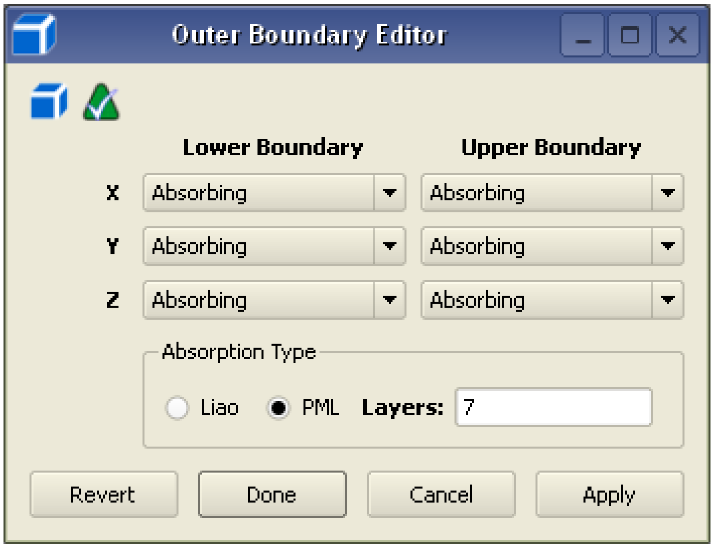
1. Navigate to the **Definitions:Waveforms** branch of the Project Tree. Double-click the Broadband Pulse waveform to edit its properties.
   Set the properties of the waveform as follows:
   - Name: Sinusoid
   - Type: Ramped Sinusoid
   - Frequency: 1 GHz



2. Click **Done** to apply the changes.

Defining the Outer Boundary

1. Double-click the **FDTD :Outer Boundary** branch of the Project Tree to open the Outer Boundary Editor.
2. Set the outer boundary properties as follows:
   - Boundary: Absorbing for all boundaries
   - Absorption Type: PML
   - Layers: 7



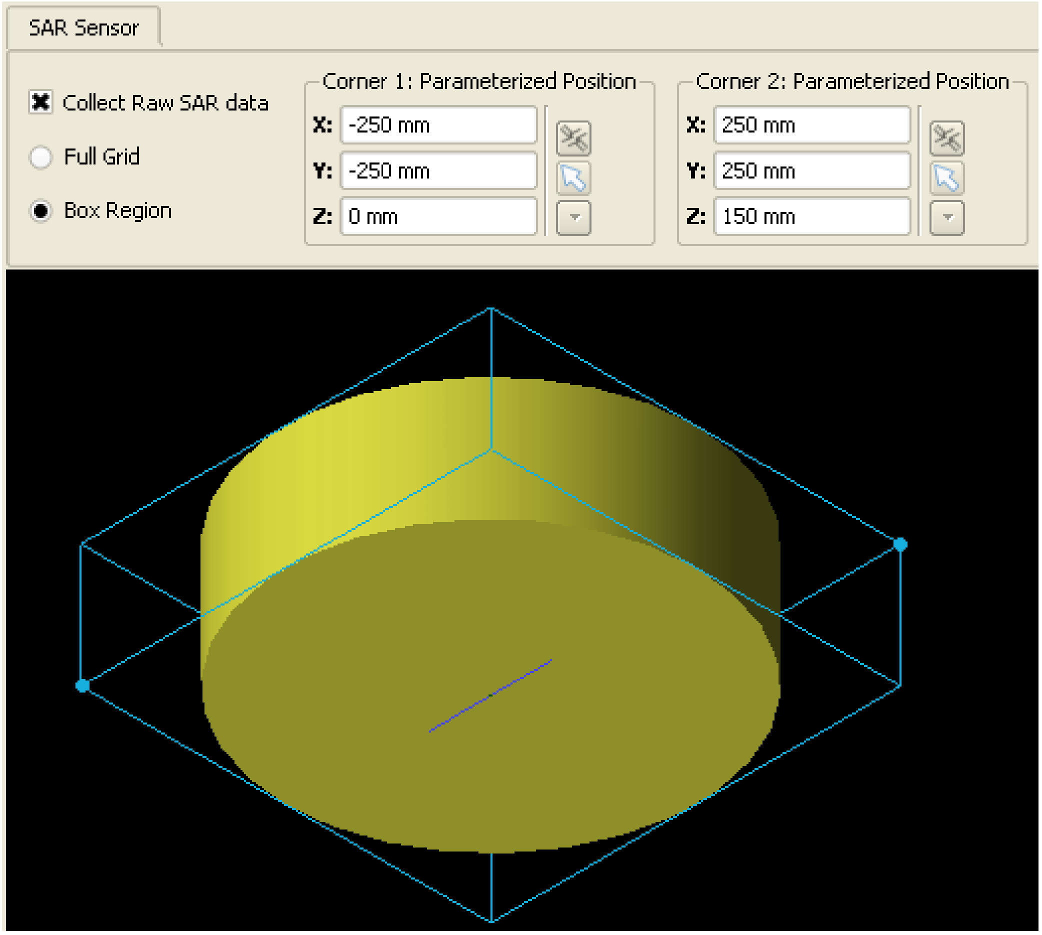
3. Click **Done** to apply the outer boundary settings.

Requesting Output Data

The project already contains one port sensor named Feed that will request results. You may also want to collect SAR results by adding an SAR Sensor.

Adding an SAR Sensor

1. Right-click the **Sensors:SAR Sensors** branch of the Project Tree. Select **Properties** from the context menu.
2. Select the **Collect Raw SAR Data** box.
3. Select **Box Region**, and enter the following coordinates:
    - Corner 1: (-250 mm, -250 mm, 0)
    - Corner 2: (250 mm, 250 mm, 150 mm)

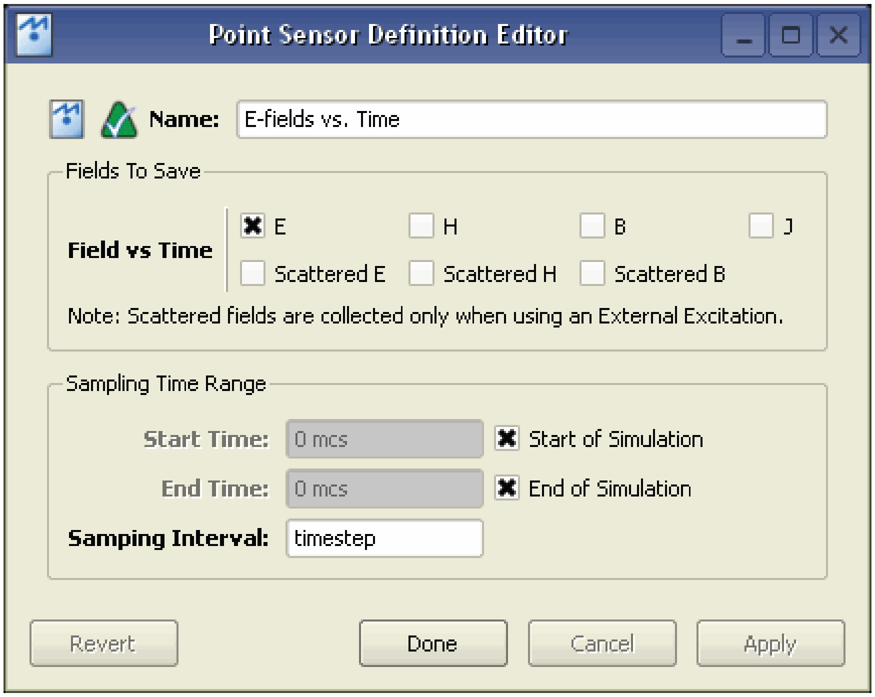


4. Click **Done** to finish editing the SAR Sensor.

Adding a Point Sensor Definition

A Point Sensor may be saved inside the Tissue object to monitor the convergence of the fields during the calculation. First, you will create its definition.

1. Right-click the **Definitions:Sensor Data Definitions** branch of the Project Tree. Choose **New Point Sensor Definition** from the context menu.
2. Set the properties of the surface sensor definition as follows:
    - Name: E-field vs. Time
    - Field vs. Time: E
    - Sampling Interval: timestep



3. Click **Done** to finish editing the Field Sampling definition.

Adding a Point Sensor

1. Right-click the **Sensors:Near Field** Sensors branch of the Project Tree. Select **New Point Sensor** from the context menu.
   - Enter its Location as (0, 0, 75 mm).
   - Under the Properties tab, enter the following:
   - Name: E-field at Tissue Center
   - Sensor Definition: E-field vs. Time
   - Sampling Method: Snapped to E-Grid
2. Click **Done** to finish editing the E-field at Tissue Center Sensor.

Running the Calculation

If you have not already saved your project, do so by selecting **File>Save Project**. After saving the project, you can create a new simulation to send to the calculation engine.

1. Open the **Simulations** workspace window. Click the **Create Simulation** button in the upper-left to set up a new simulation.
2. Type a descriptive Name for the simulation, such as **Tissue Cylinder exposed to 1 GHz Dipole, SAR Saved**.
3. Under **Frequencies of Interest**, select the Collect Steady-State Data checkbox.
4. Under the Frequencies tab, select **Use Waveform Frequency**.
   Most of the default settings are sufficient. Navigate to the **Specify Termination Criteria** tab. Set up the termination criteria as follows:
   - Maximum Simulation Time: 10000 * timestep
   - Dectect Convergence: Checked
   - Threshold: -20 dB
5. Select **Create and Queue Simulation** to close the dialog box and run the new simulation.

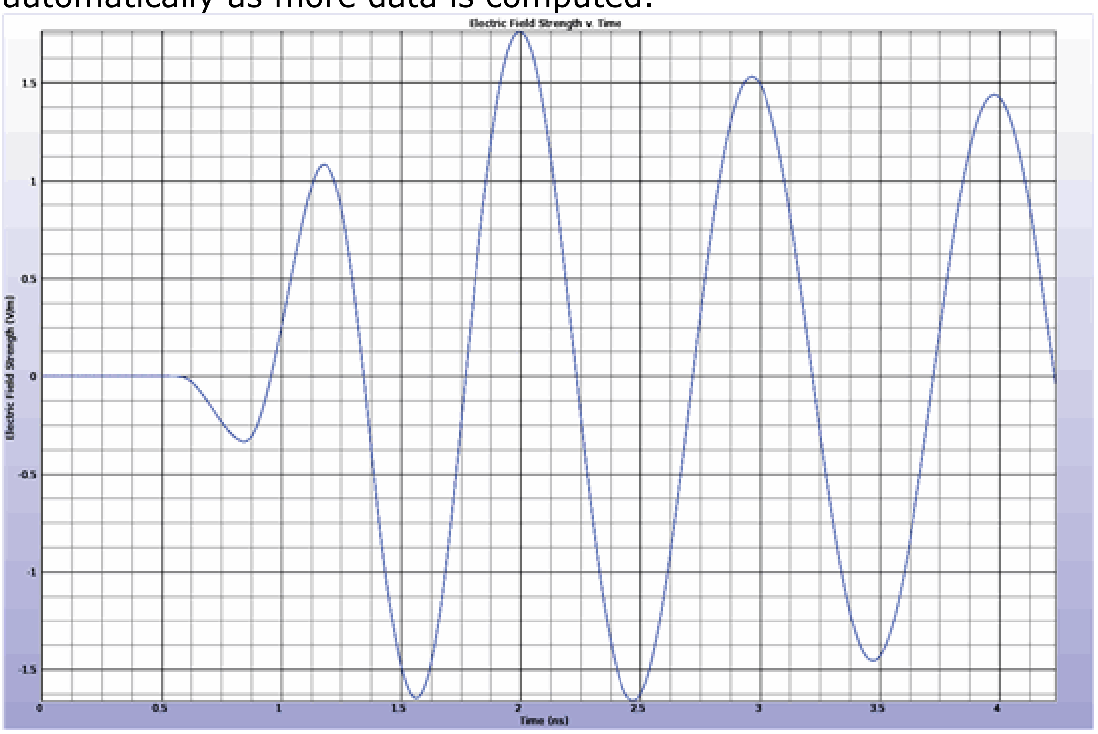
Viewing the Results

The **Output** tab of the Simulations workspace window displays the progress of the simulation. After the Status column shows that the simulation is complete, you can view its results in the **Results** workspace window.

E-field Results

You can view the E-field results retrieved from the center of the Tissue.

1. To filter the list accordingly, select the following options in the columns in the top pane of the **Results** window. You may need to change your column headings first:
    - **Sensor**: E-field at Tissue Center
    - **Result Type**: E-field (E)
2. Right-click the result and select **Create Line Graph**.
3. Select **X** as the Component, and click **View**. The plot of the E-field at the center of the **Tissue** object will appear.
   It is possible to view the data before the simulation is complete. The plot will update automatically as more data is computed.



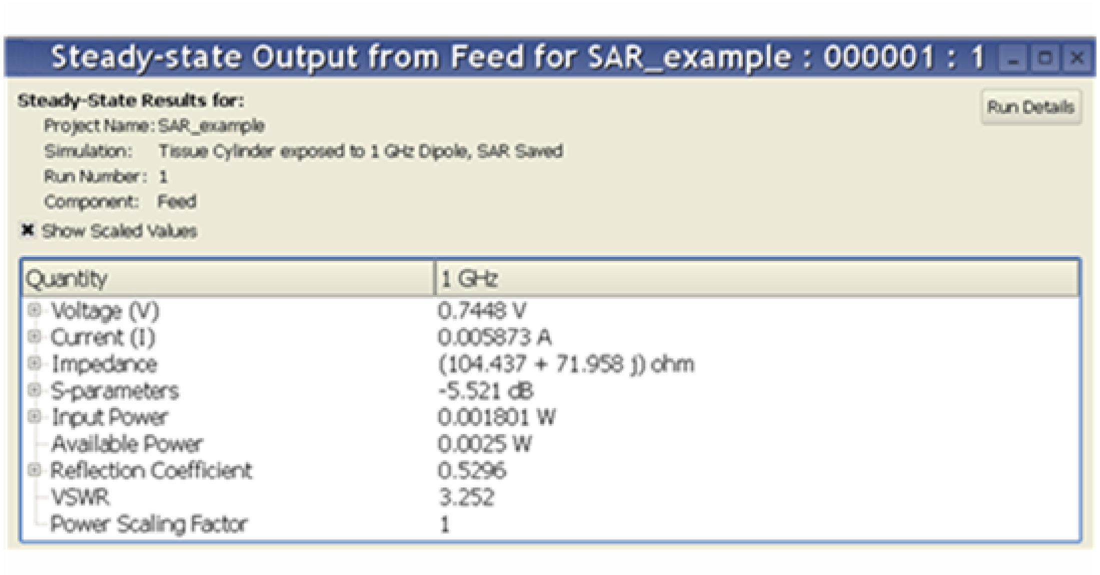
   You can close the window after you have finished viewing the results.

System Efficiency Results

Now, you can view data from the point.

1. To view the Feed results, select the following:
    - **Sensor**: Feed
    - **Result Type**: S-Parameters
2. Double-click the results under the **Discrete** domain. The following results will appear

showing the impedance at the feed, the input power delivered, and the return loss.

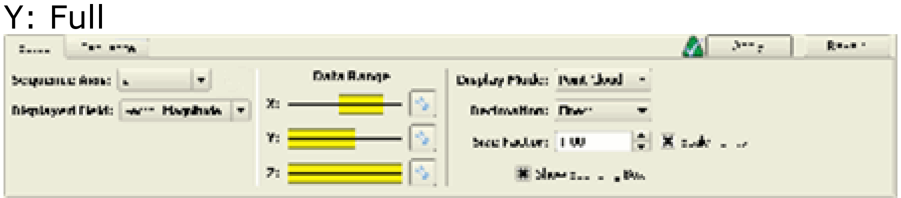


SAR Sensor Data

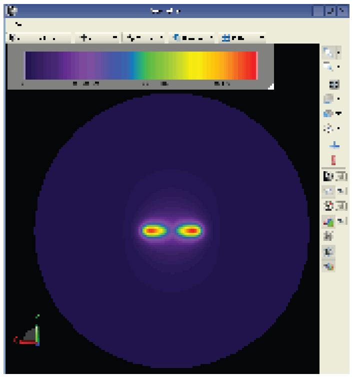
You can load the SAR data into the field viewer.

1. To view the SAR sensor data, select the following:
    - **Sensor**: SAR Sensor (Raw)
    - **Result Type**: SAR (Specific Absorption Rate)
2. Double-click the result in the filtered list. The plot will appear in the Geometry workspace window.
3. Under the **Setup** tab adjust the following settings:
    - Sequence Axis: Z
    - Decimation: Finest

    Under Axis Ranges:
    - Frequency: 1 GHz
    - X: Full
    - Y: Full



4. Click **Apply** to finish editing the SAR sensor setup.
5. Toggle the Parts Visibility to turn off the display of the geometry, and select the Top (-Z) orientation. The resulting image should appear.



6. Click the **Sequence** tab and then click **Play** to view a movie of the SAR slices. To increase the speed of the movie, change the **Decimation** on the **Setup** tab to **Normal**.
7. To review the SAR statistics of the peak and average SAR, click the **Statistics** tab.

Validating SAR Calculations

In this project, you will learn how to:

- Model a tissue-simulating liquid with a dipole.
- Define the properties of the environment.
- Add a feed to the dipole and simulate its effects.
- Add a point sensor and measure E-field at the center of the liquid.
- Add SAR sensors and retrieve raw and averaged SAR data.

Getting Started

This section briefly describes how to set the display units for the SAR Validation project. To set up a project for the first time, refer to Application Preferences Appendix for instructions about how to configure project preferences and navigate through the display units tab.
In the Project Properties Editor window, navigate to the Display Units tab:

1. Select **SI Metric** in the **Unit Set** drop down list.
2. Change **Length** to **millimeters(mm)**. This changes the value of Unit Set to **Custom**.
3. Click **Done**.

Creating the Geometry

The geometry for this example consists of a Flat Phantom, Phantom Shell, and a dipole made of two cylinders.

Modeling the Flat Phantom

First, you will create the rectangular extrusion named Flat Phantom which represents the tissue simulating liquid used for SAR measurements. You will perform the simulation at 835MHz, so the phantom dimensions will be 220x150mm with an extrusion in the +Z direction of 150mm.

1. Right-click the Parts branch of the Project Tree. Choose **Create New > Extrude** from the context menu.
2. Name the object by typing **Flat Phantom** in the Name text box.
3. Choose **View** > **Standard Views** > **Top(-Z)** orientation.
4. Choose the **Rectangle** tool from the Shapes toolbar.
5. Click the mouse on the origin of the coordinate system.
6. Press **Tab** to display the creation dialog for the second point. Enter **225mm,150mm** and click **OK** to complete the rectangle.
7. Navigate to the **Extrude** tab to extrude the rectangular region. Enter the **Extrude Distance** of 150mm.
8. Click **Done** to finish the Flat Phantom geometry.

Modeling the Phantom Shell

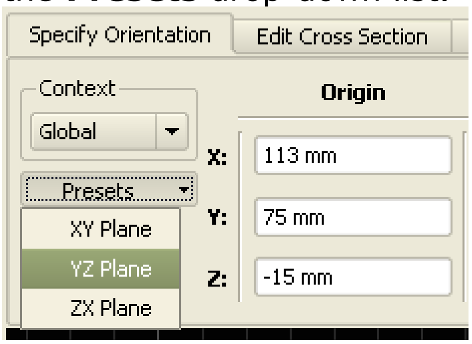
Next, you will create the rectangular extrusion named Phantom Shell. This shell is a plastic vessel that will hold the simulating liquid. For this simulation, you need to add only the bottom of the vessel that separates the liquid from the dipole source. This shell size will match the phantom size in X and Y, and have a thickness of 2mm.

1. Right-click the **Parts** branch and choose **Create New > Extrude** from the context menu.
2. Under the Specify Orientation tab, define the origin at (0, 0, 2mm).
3. Under the **Edit Cross Section** tab, type **Phantom Shell** in the Name text box.
4. Choose **View** > **Standard Views** > **Bottom(+Z)** orientation.
5. Choose the **Rectangle** tool from the Shapes toolbar.
6. Trace the new cross-section over the existing cross-section (of the flat phantom) since they are of equal width and length.
7. Navigate to the **Extrude** tab to extrude the rectangular region a distance of -2mm.
8. Click **Done** to finish the Phantom Shell geometry.

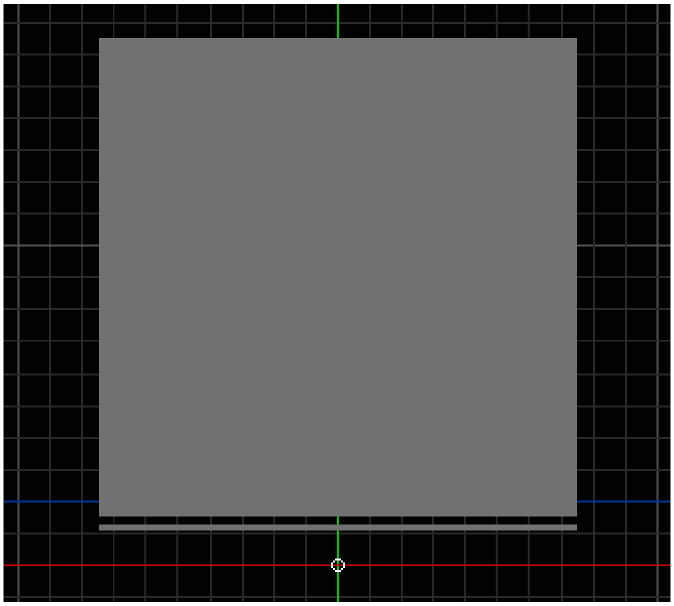
Modeling the Dipole

Now you will create the dipole geometry, which comprises two cylindrical extrusions. Typically the dipole will have a balun structure as well, but we will omit that for simplicity in this example. The dipole will have a radius of 1.8mm and a length of 161mm.

1. Right-click the **Parts** branch and choose **Create New > Extrude** from the context menu.
2. Under the **Specify Orientation** tab, define the origin at (113mm, 75mm, -15mm).
    - Redefine the orientation of the sketching plane by selecting the **YZ** Plane under the **Presets** drop-down list.



3. Under the **Edit Cross Section** tab, type **Cylinder1** in the Name box.
4. Choose **View** > **Standard Views** > **Left(+X)** orientation.
5. Choose the **Circle center** radius tool from the Shapes toolbar.
    - Click the mouse on the origin of the coordinate system.
    - Press **Tab** to display the creation dialog for the radius. Enter 1.8mm and click **OK**.

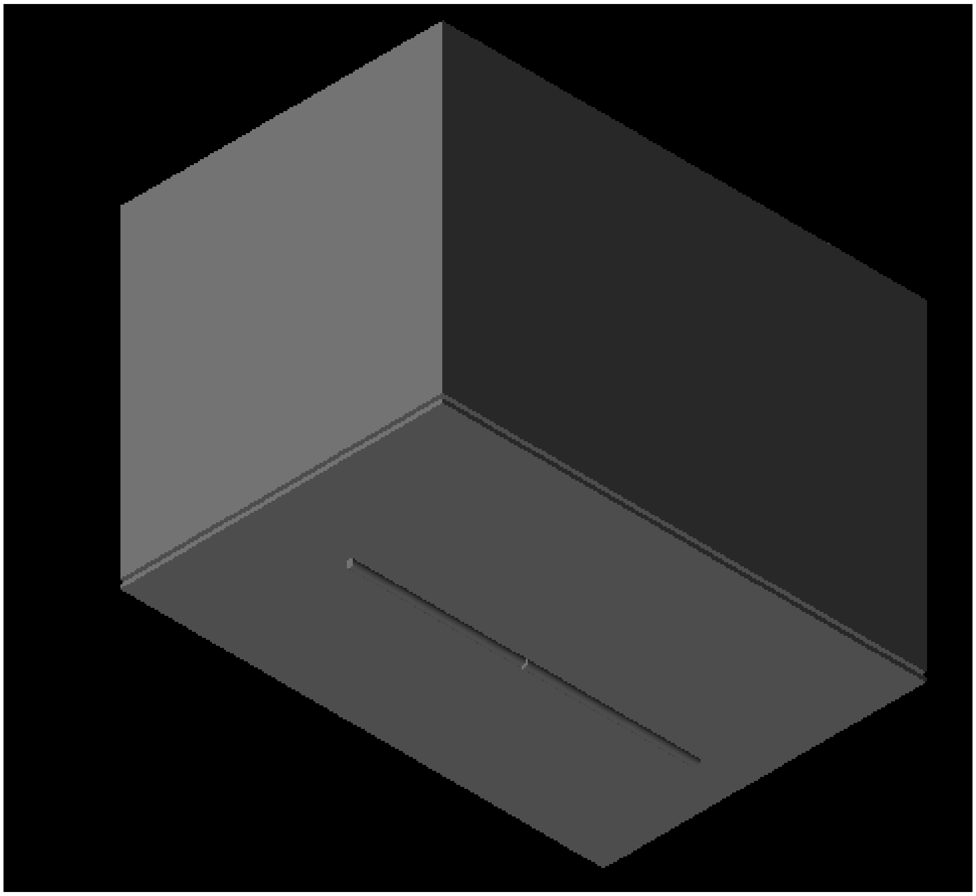


6. Navigate to the **Extrude** tab to extrude the cylinder. Enter a distance of 80mm.
7. Click **Done** to finish the Cylinder1 geometry.

Create the second extrusion

Now you will create the second part of the dipole, Cylinder2.

1. Right-click the **Parts** branch and choose **Create New>Extrude**.
2. Under the Specify Orientation tab, define the origin at (32mm, 75mm, -15mm).
    - Redefine the orientation of the sketching plane by selecting the YZ Plane under the Presets drop-down.
3. Under the **Edit Cross Section** tab, type **Cylinder2** in the **Name** text box.
4. In the **View Tools** toolbar, select the **Right** (-X) orientation.
5. Choose the **Circle center, Radius** tool from the **Shapes** toolbar.
    - Click the mouse on the origin of the coordinate system.
    - Click **Tab** to display the creation dialog for the radius. Enter 1.8mm and click **OK**.
6. Navigate to the **Extrude** tab to extrude the cylinder. Enter a distance of 80mm.
7. Click **Done** to finish the Cylinder2 geometry.
   The following figure displays a view of the finished geometry before materials are added.

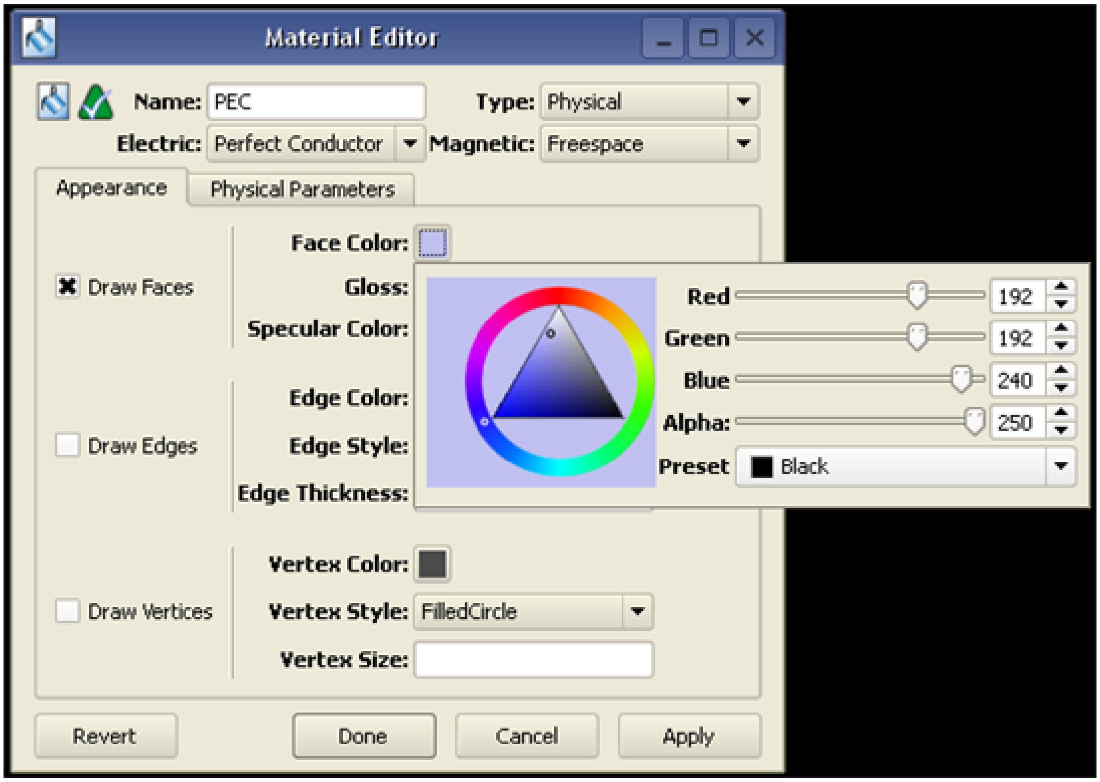


Creating Materials

After creating four new objects, you will assign materials to them. Cylinder1 and Cylinder2 will be perfect electric conductors, PEC. The Flat Phantom and Phantom Shell objects will be isotropic materials named Phantom Liquid and Phantom Shell, respectively.

Define material, PEC

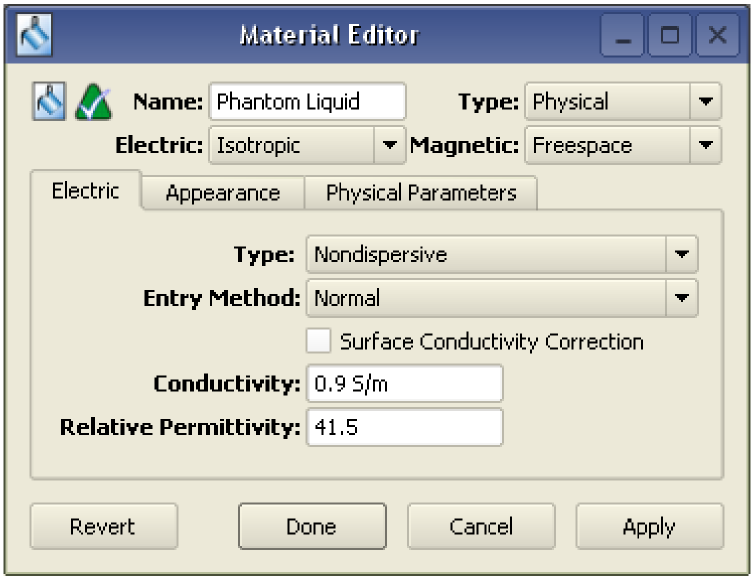
1. Right-click the **Definitions:Materials** branch of the Project Tree. Choose **New Material Definition** from the context menu.
2. Set the perfect electric conductor material properties as follows:
   - Name: PEC
   - Electric: Perfect Conductor
   - Magnetic: Freespace
3. If desired, navigate to the Appearance tab to set the display color of the PEC material.



Define Material, Phantom Liquid

1. Right-click the **Definitions:Materials** branch of the Project Tree and select **New Material Definition**.
2. Set the material properties as follows:
   - Name: Phantom Liquid
   - Electric: Isotropic
   - Magnetic: Freespace
   - Under the Electric tab:
     - Type: Nondispersive
     - Entry Method: Normal
     - Conductivity: 0.9 S/m
     - Relative Permittivity: 41.5

Editing the color of the Phantom Liquid material



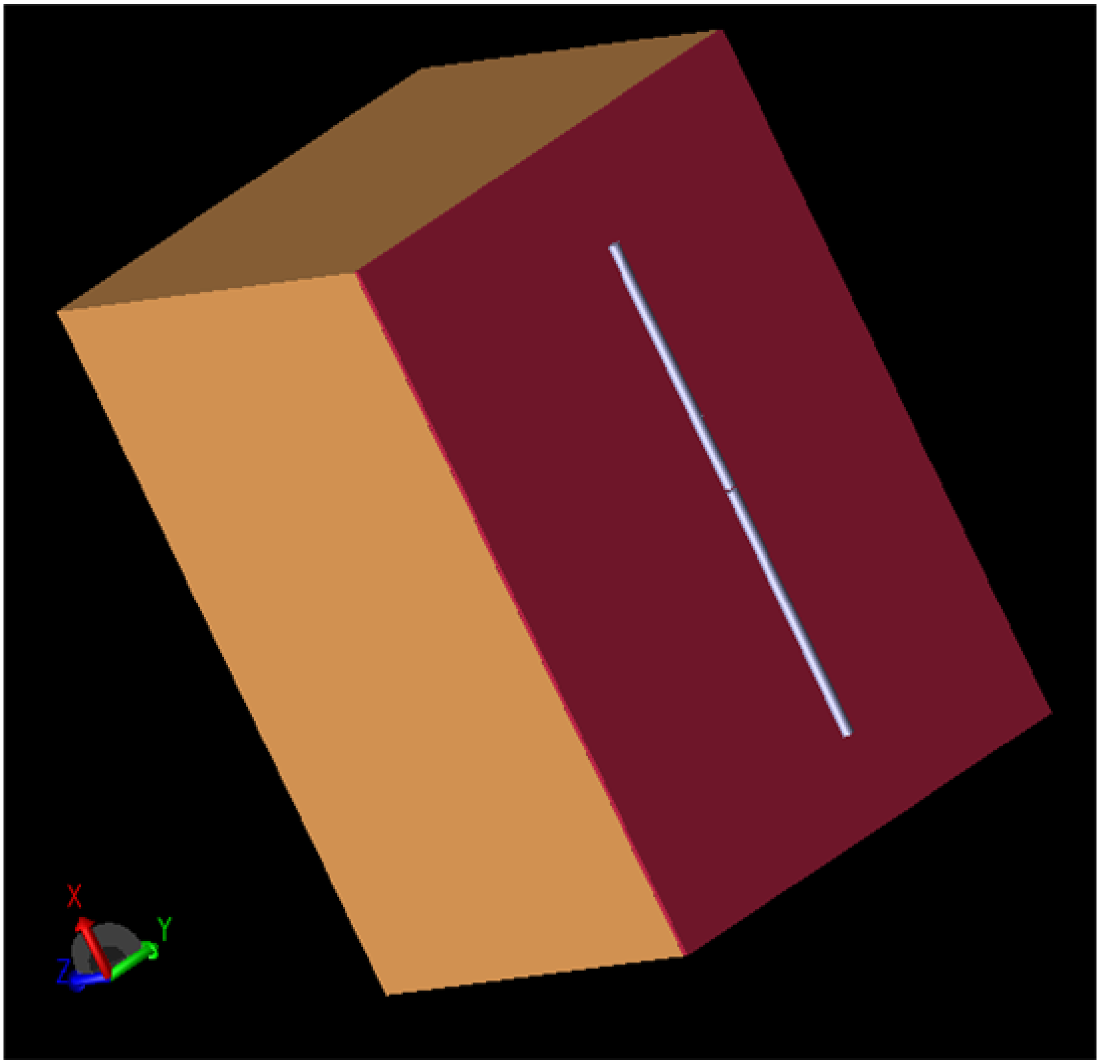
3. Under the **Physical Parameters** tab, enter 1000 kg/m^3 as the Density.
4. Navigate to the **Appearance** tab and assign the **Phantom Liquid** material a new color to distinguish it from PEC.
5. Click **Done** to add the new material, Phantom Liquid.

Define material, Phantom Shell

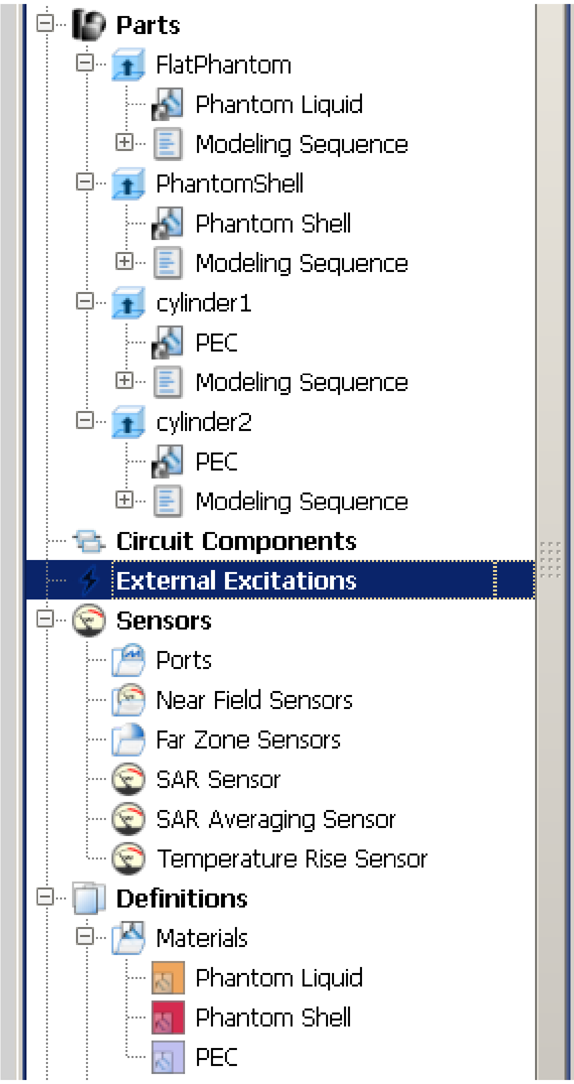
1. Right-click the **Definitions:Materials** branch of the Project Tree and select **New Material Definition**.
2. Set the material properties as follows:
    - Name: Phantom Shell
    - Electric: Isotropic
    - Magnetic: Freespace
    - Under the Electric tab:
    - Type: Nondispersive
    - Entry Method: Normal
    - Conductivity: 0 S/m
    - Relative Permittivity: 3.7
3. Navigate to the Appearance tab and assign the **Phantom Shell** material a new color to distinguish it from PEC.
4. Click **Done** to add the new material, Phantom Shell.

Assigning Materials

1. Click and drag the PEC material object located in the Project Tree and drop it on top of Cylinder1 and Cylinder2.
2. Assign the **Phantom Liquid** material to the **Flat Phantom** object.
3. Assign the **Phantom Shell** material to the **Phantom Shell** object.
   The finished geometry with applied materials is seen in the following figure.



The Project Tree

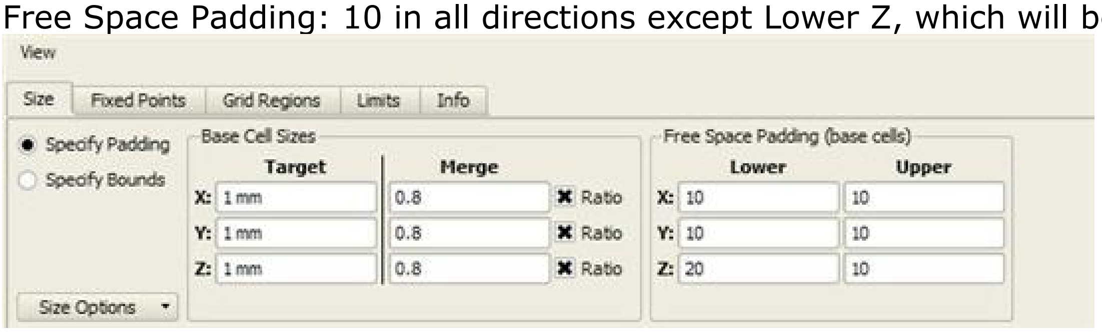


Creating the Grid

Now, you will define characteristics of the cells in preparation to perform an accurate calculation.

Define cell size and padding

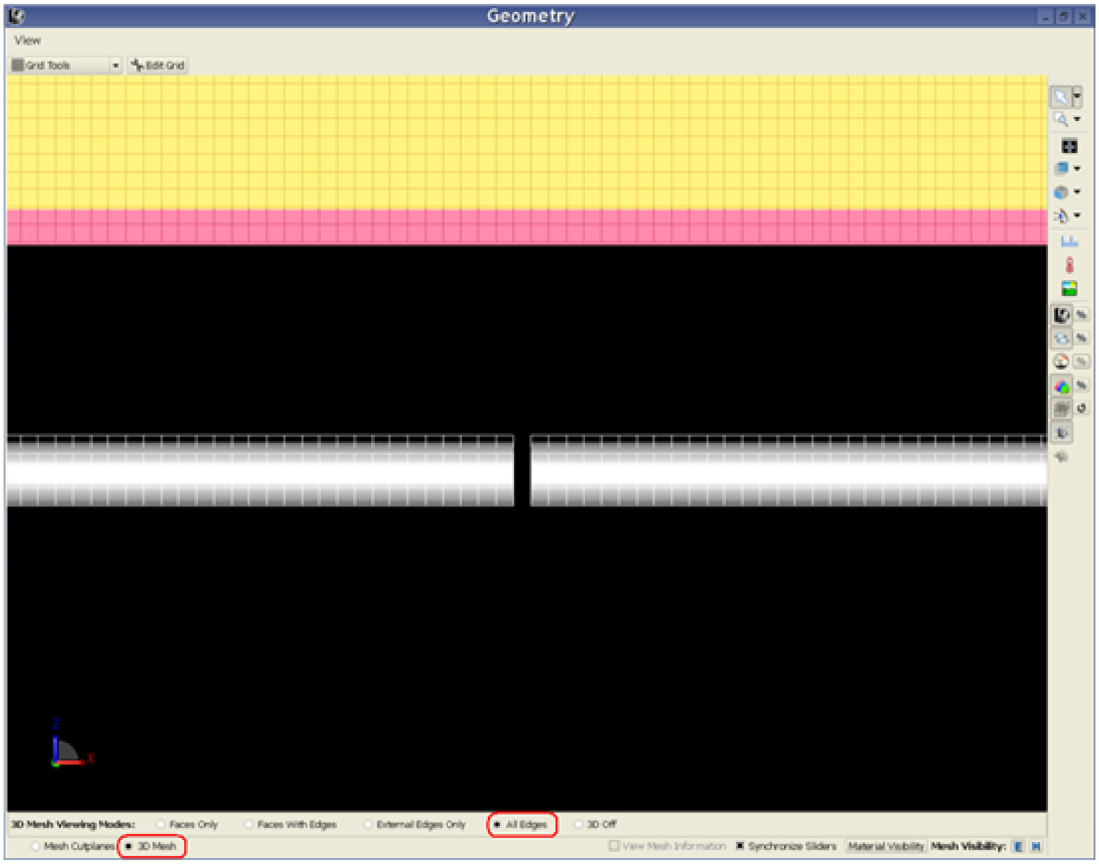
1. Open the **Geometry** browser window, select **Grid Tools** and click **Edit Grid**.
2. Navigate to the **Size** tab.
    - Define Base Cell Sizes as Target 1mm and Merge 0.8 in all directions, with the Ratio boxes selected.
    - Free Space Padding: 10 in all directions except Lower Z, which will be 20.



3. Click **Done** to apply the grid settings.

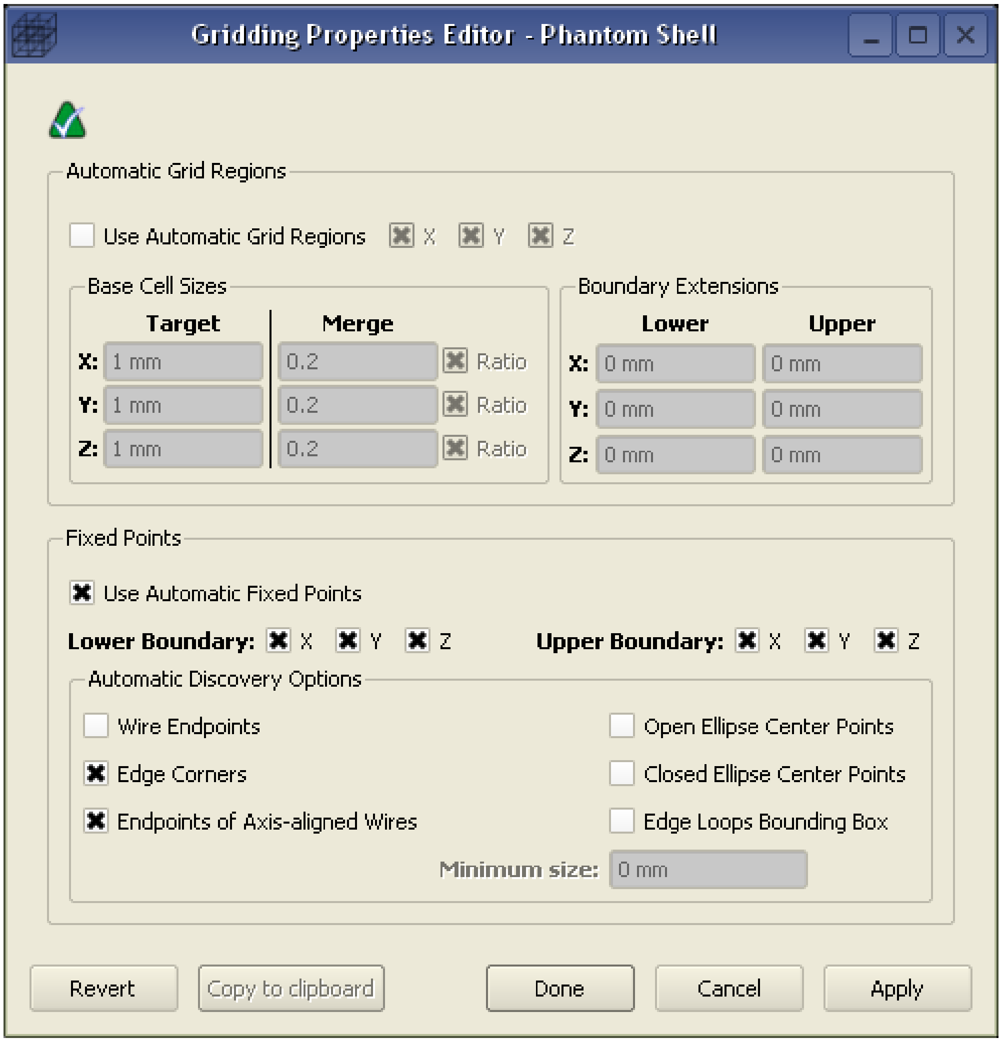
Creating a Mesh

In the FDTD branch of the Project Tree, double-click the **Mesh** icon. This displays the mesh view and automatically create the mesh. If you switch to the 3D Mesh view of **All Edges**, note that the grid does not align with the CAD view of the geometry objects This is because the cell size does not overlap the geometry dimensions exactly.



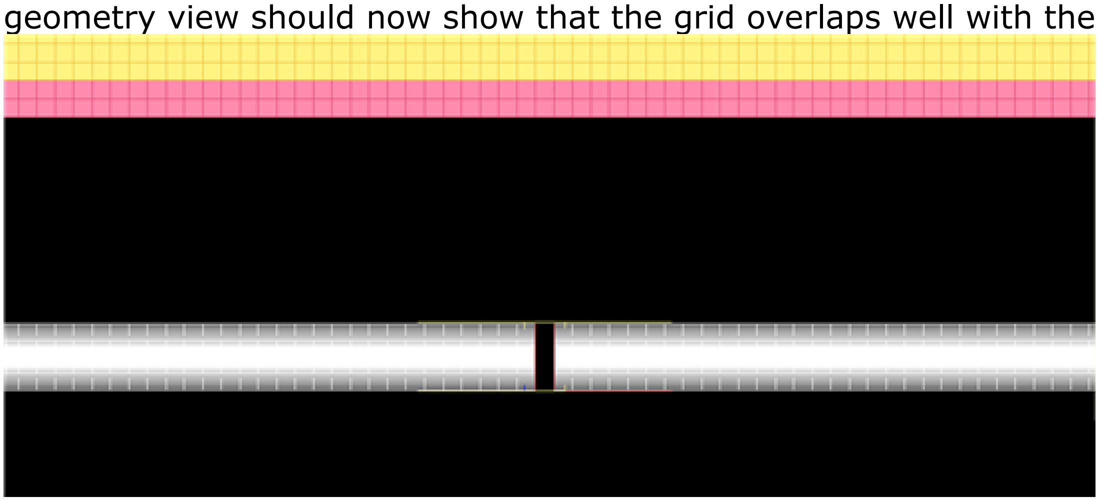
To align the mesh, you can turn on the fixed points for several of the geometry objects. This will adjust the mesh so that the grid lines overlap the edges of the CAD geometry objects.

1. From the **Parts** branch, right-click the **Phantom Shell** object.
2. Select **Gridding Properties** from the menu.
3. In the **Gridding Properties Editor** dialog box, select **Use Automatic Fixed Points**.



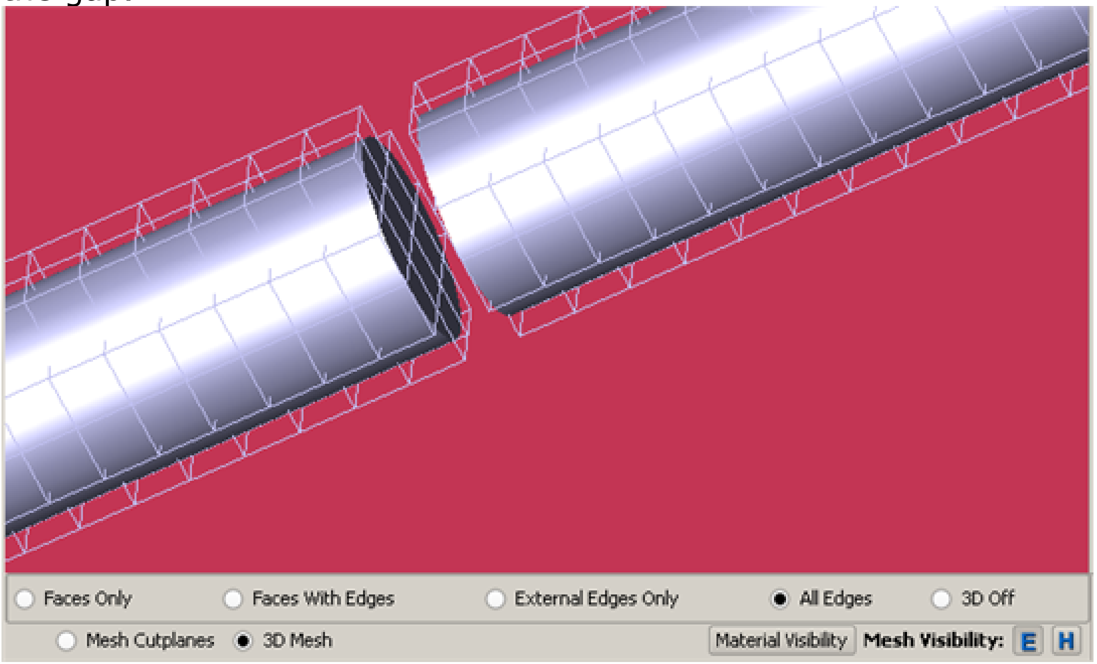
4. Click **Apply** to apply the fixed points extraction to this geometry object.
5. Click **Copy** to clipboard to save these settings.
6. Click **Done**.

Now, you will turn on Fixed Points for the cylinders. Select both Cylinder1 and Cylinder2 from the Parts branch. Right-click and select **Edit> Paste** to copy the clipboard contents to these two objects. This will turn on fixed points for the dipole as well. The resulting geometry view should now show that the grid overlaps well with the CAD objects.

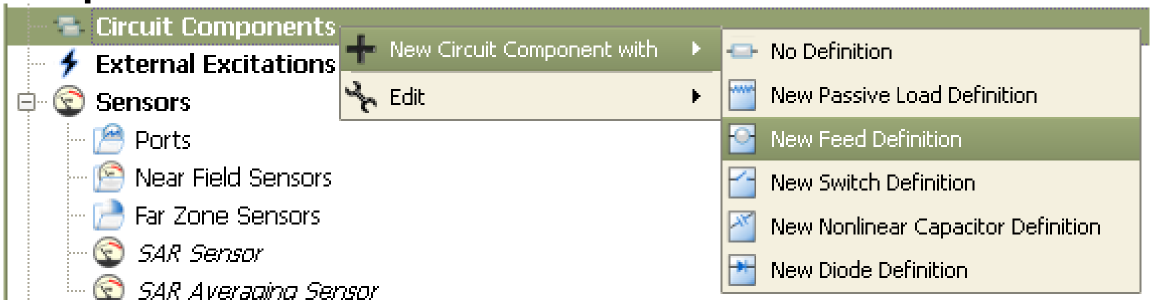


Adding a Feed

Now, you will add a Feed to the geometry. We want to place the feed in the gap between the two cylinders made of PEC materials. The following figure displays a 3D Mesh View of the gap.



1. Right-click the **Circuit Components** branch of the Project Tree. Choose **New Circuit Component with> New Feed Definition** from the context menu.



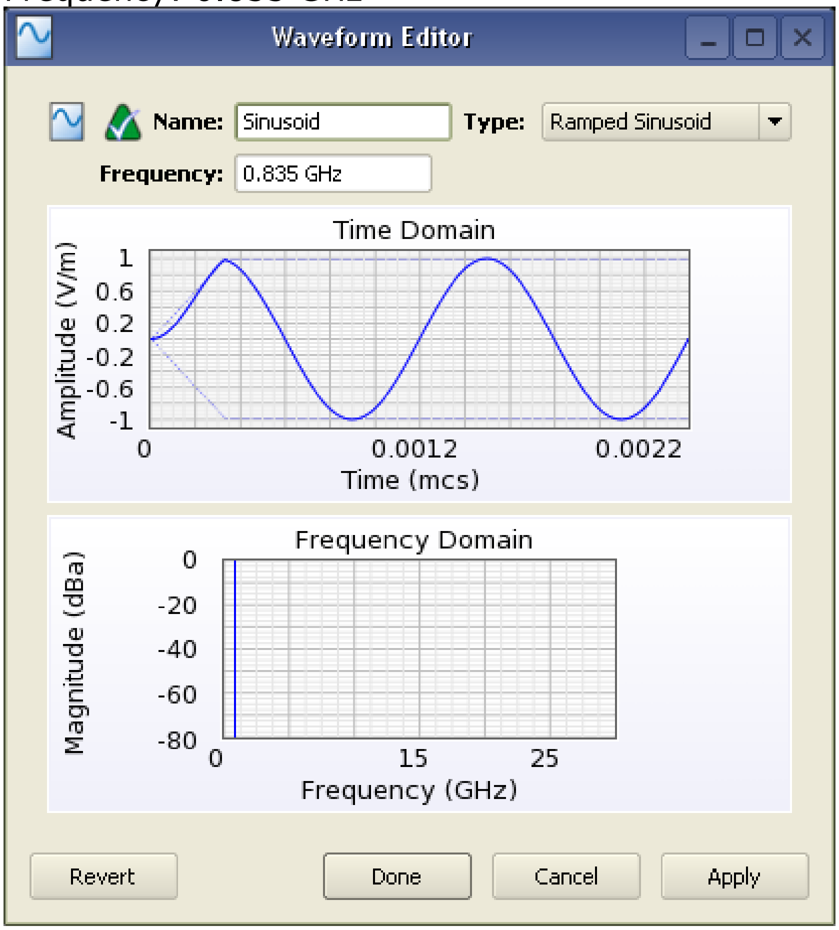
2. Define the endpoints of the feed.
   - Endpoint 1: X: 113 mm, Y: 75 mm, Z: -15 mm
   - Endpoint 2: X: 112 mm, Y: 75 mm, Z: -15 mm
3. Navigate to the Properties tab, and enter the following:
   - Name: Feed
   - Component Definition: 50 ohm Voltage Source
   - Direction: Auto
   - Polarity: Positive
   - Select the **This component is a port** checkbox.
4. Click **Done** to add the Feed.

Editing the Waveform

An associated waveform was automatically created for the feed definition.

1. Navigate to the **Definitions:Waveforms** branch of the Project Tree. Double-click the

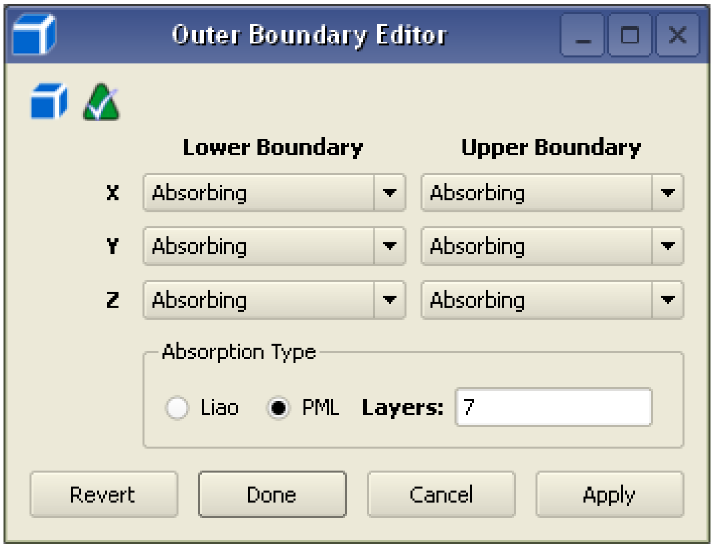
Broadband Pulse waveform to edit its properties.
2. Set the properties of the waveform as follows:
    - Name: Sinusoid
    - Type: Ramped Sinusoid
    - Frequency: 0.835 GHz



3. Click **Done** to apply the changes.

Defining the Outer Boundary

1. Double-click the **FDTD:Outer Boundary** branch of the Project Tree to open the Outer Boundary Editor.
2. Set the outer boundary properties as follows:
    - Boundary: Absorbing for all boundaries
    - Absorption Type: PML
    - Layers: 7



3. Click **Done** to apply the outer boundary settings.

Requesting Output Data

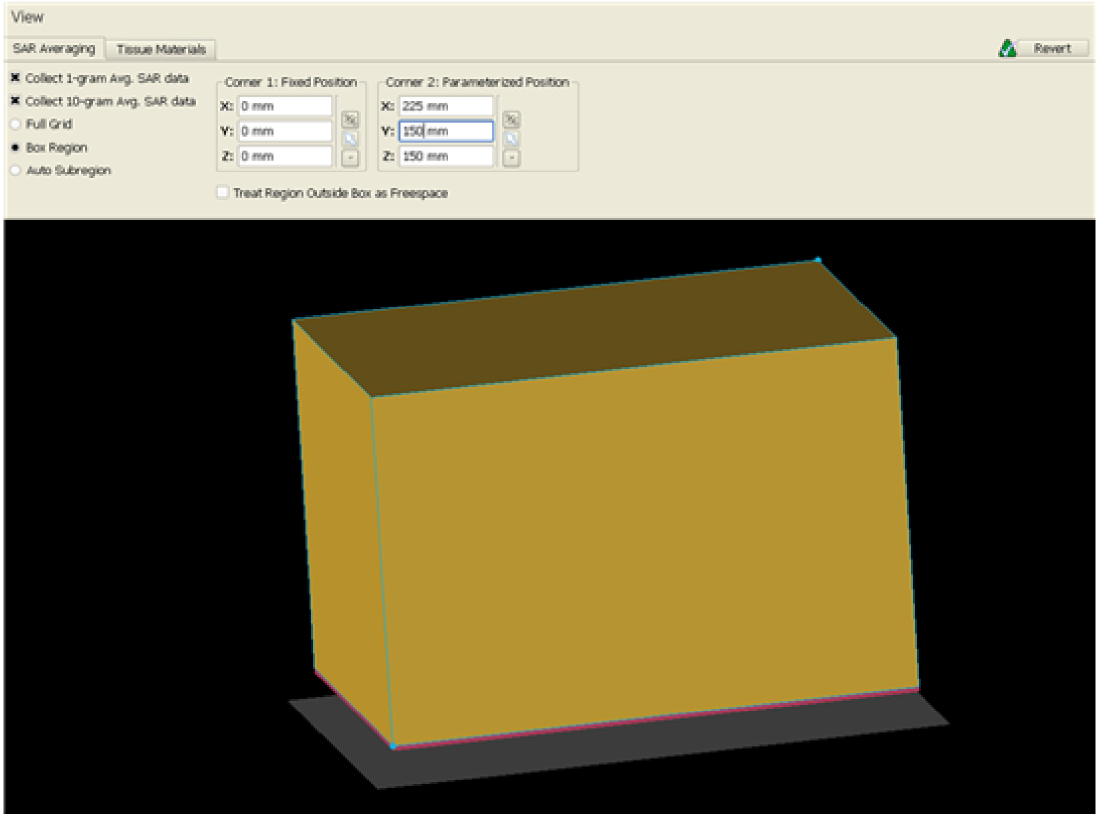
Recall that the project already contains one port sensor named Feed that will request results. You can collect SAR results by adding an SAR Sensor.

Adding an SAR Sensor

1. Right-click the **Sensors:SAR Sensors** branch of the Project Tree. Select **Properties** from the context menu.
    - Select the **Collect Raw SAR Data** checkbox.
    - Select the **Full Grid** box. It requires that the data be saved over the full grid if Averaged SAR values will be computed.
2. Click **Done** to finish editing the SAR Sensor.

To collect averaged SAR data, you must define a sensor.

1. Right-click the **Sensors:SAR Averaging Sensor** branch of the Project Tree. Select Properties from the context menu.
    - Check the Collect 1-gram Avg. SAR data and Collect 10-gram Avg. SAR data boxes.
    - Select the Box Region box, and enter the following coordinates:
        - Corner 1: (0 mm, 0 mm, 0 mm)
        - Corner 2: (225 mm, 150 mm, 150 mm)

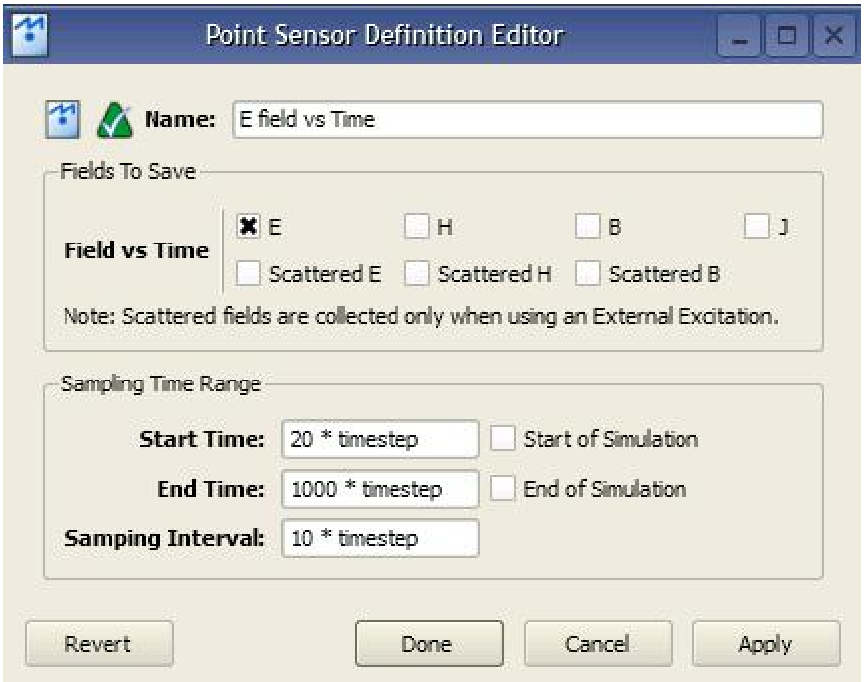


2. Click **Done** to finish editing the SAR Averaging Sensor.

Adding a Point Sensor Definition

A Point Sensor may be saved inside the **Flat Phantom** object to monitor the convergence of the fields during the calculation. First, ypu will create its definition.

1. Right-click the **Definitions:Sensor Data Definitions** branch of the Project Tree. Choose **New Point Sensor Definition** from the context menu.
2. Set the properties of the surface sensor definition as follows:
   - Name: E-field vs. Time
   - Field vs. Time: E
   - Sampling Interval: timestep



3. Click **Done** to finish editing the definition.

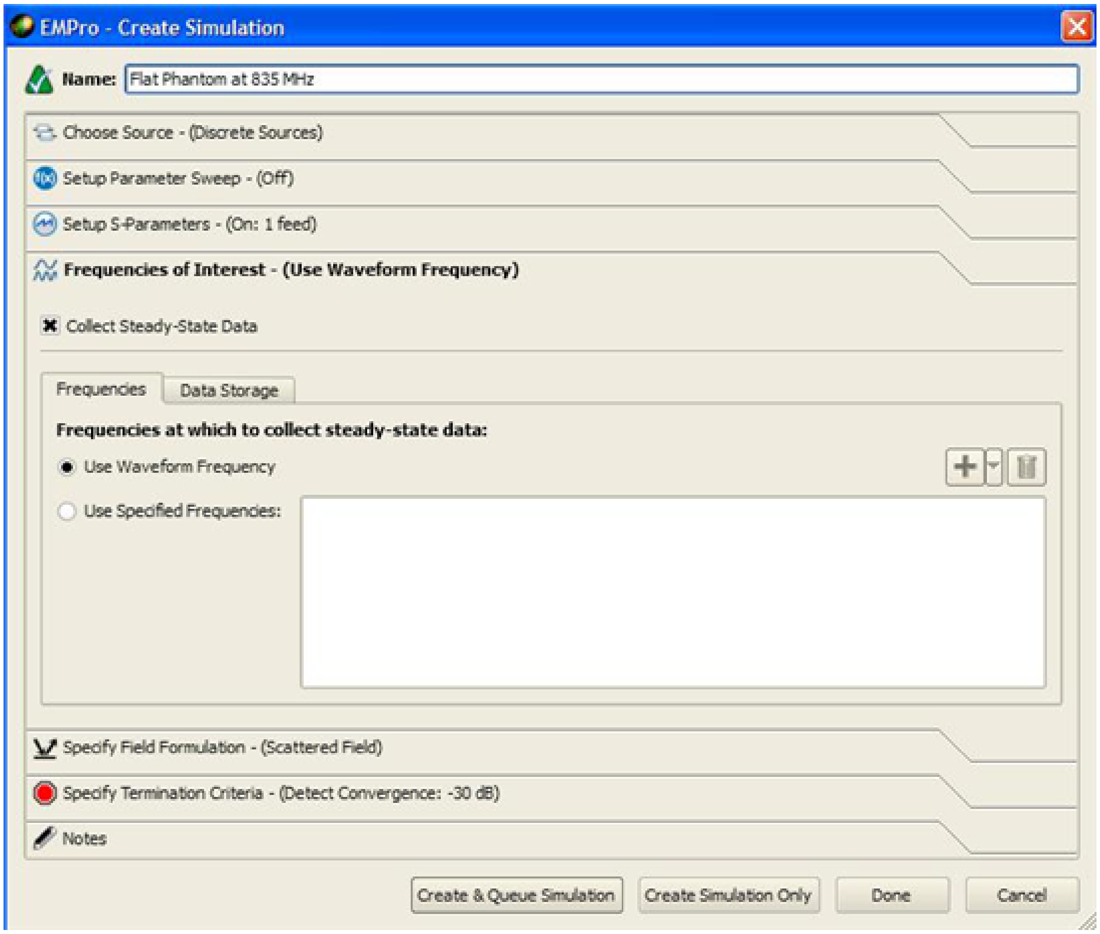
Adding a Point Sensor

1. Right-click the **Sensors:Near Field Sensors** branch of the Project Tree. Select **New Point Sensor** from the context menu.
    - Enter its Location as (112.5 mm, 75 mm, 15 mm).
    - Under the Properties tab, enter the following:
        - Name: E-field
        - Sensor Definition: E-field vs. Time
        - Sampling Method: Snapped to E-Grid
2. Click Done to finish editing the E-field Sensor.

Running the Calculation

If you have not already saved your project, do so by selecting **File>Save Project**. After the project is saved, a new simulation can be created to send to the calculation engine.

1. Open the **Simulations** workspace window. Click the **New Simulation** button in the upper-left to set up a new simulation.
2. Type a descriptive name for the simulation, such as Flat Phantom at 835MHz.
3. Most of the default settings are sufficient. Navigate to the Specify Termination Criteria tab. Set up the termination criteria as follows:
    - Maximum Simulation Time: 10000 * timestep
    - Dectect Convergence: Checked
    - Threshold: -30 dB
4. Select Create and Queue Simulation to close the dialog and run the new simulation.



Viewing the Results

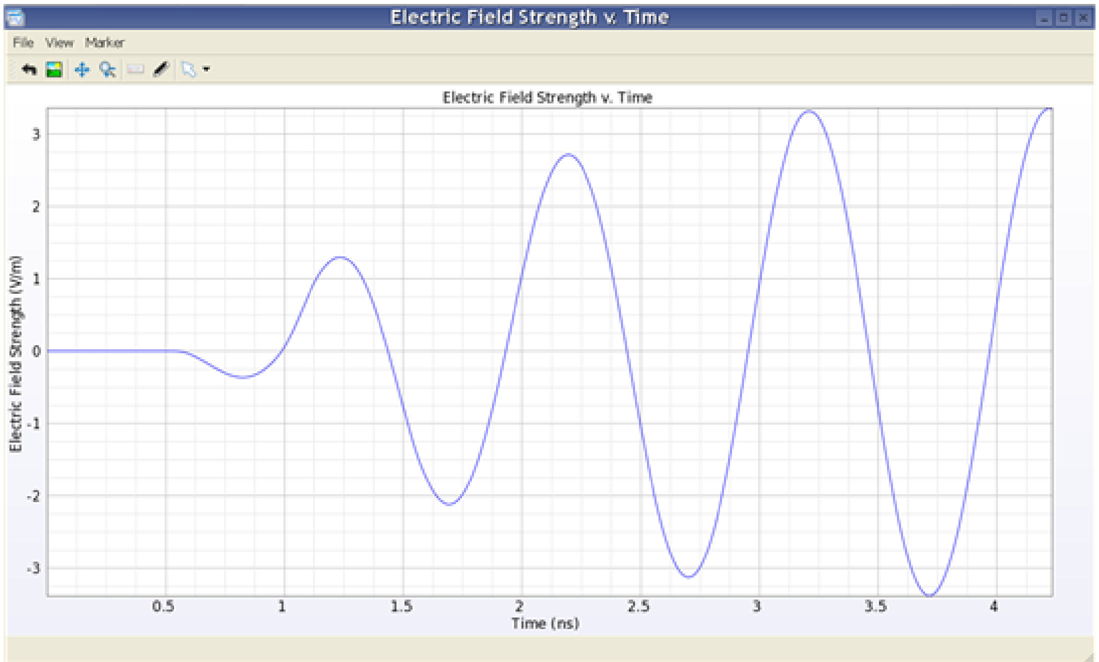
The Output tab of the Simulations workspace window displays the progress of the simulation. After the Status column shows that the simulation has completed, you can view the results in the Results workspace window.

E-field Results

Now, you can view the E-field results retrieved from the center of the Tissue.

1. To filter the list accordingly, select the following options in the columns in the top pane of the Results window. (You may need to change your column headings first.)
    - Output Object: E-field
    - Result Type: E-field (E)
2. Right-click the result and select Create Line Graph.
    - Select X as the Component, and click **View**. The plot of the E-field at the center of the Flat Phantom object will appear.

> **Note**
> It is possible to view the data before the simulation is complete. The plot will update automatically as more data is computed.



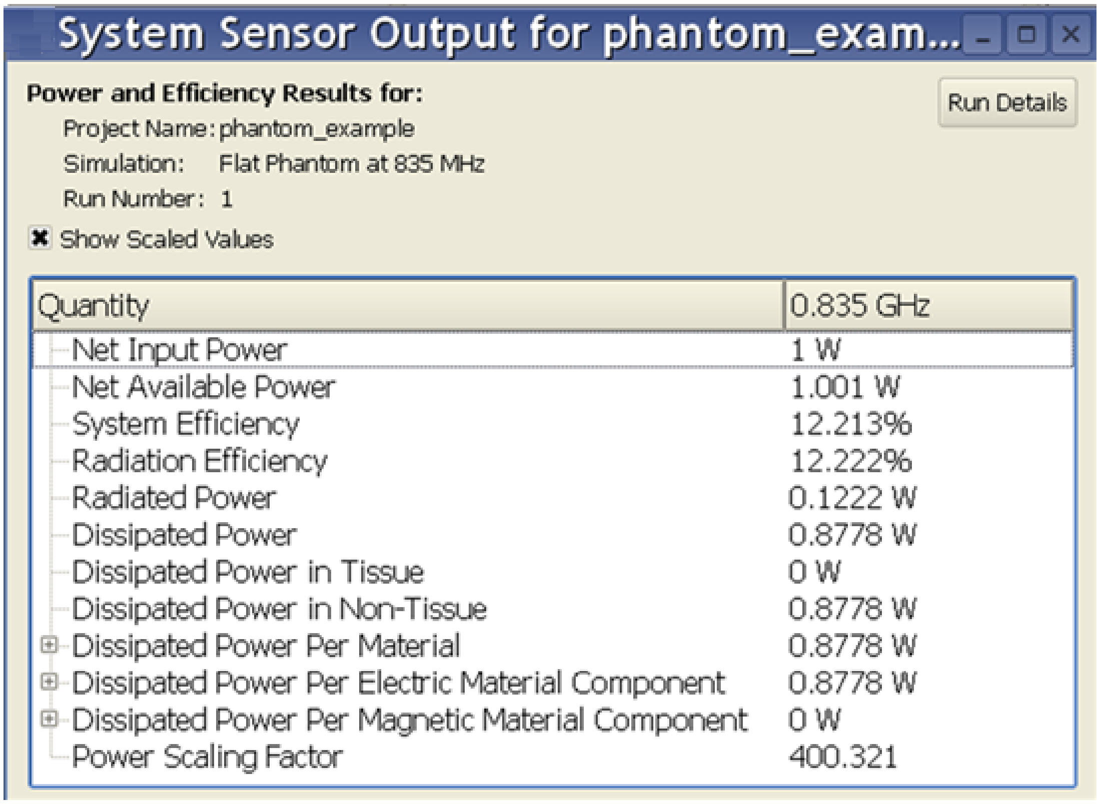
The resulting plot indicates that the fields inside the phantom are at steady-state as a smooth sine wave is visible. This confirms our convergence condition of -30 dB that was set during the simulation setup.

3. You may close the window when you are finished viewing the results.

System Efficiency Results

Now you can view data from the point.

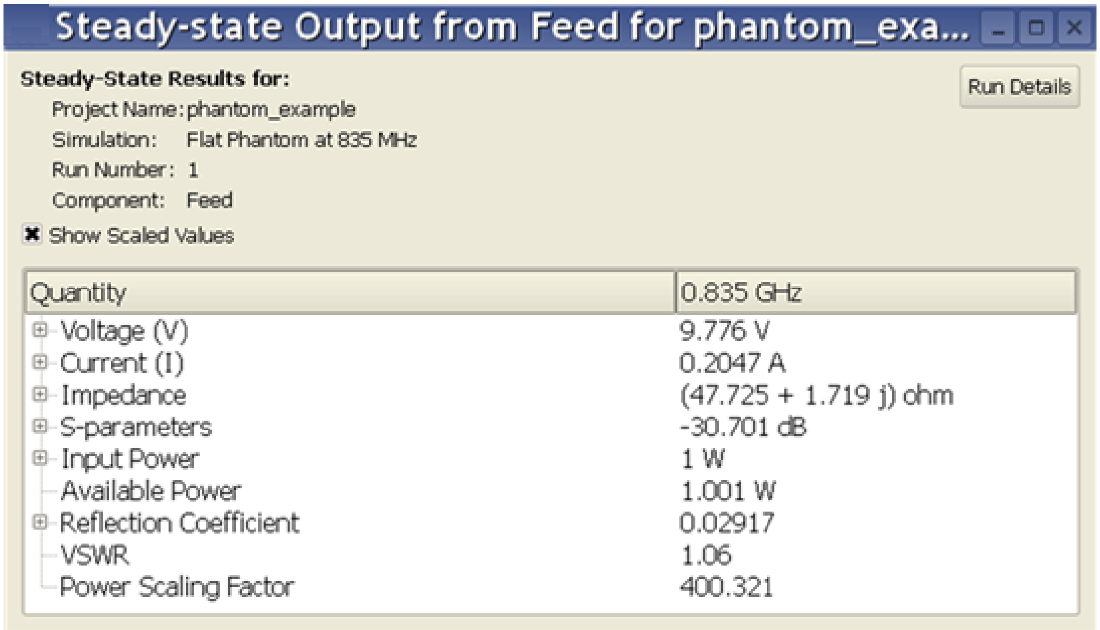
1. To view the system efficiency results, select the following:
    - Output Object: System
    - Result Type: Net Input Power
2. Double-click on the result. The powers in the simulation are displayed. As you can see, the power delivered to the antenna is relatively small, just under 2.5mW. For many SAR analyzes, the power is adjusted to a value such as 1W to normalize all results. You can do this by clicking on the System Sensor Output window.
3. Click the power value to the right of Net Input Power (0.002498 W).
4. Type a value of 1W and click **Enter**. The powers should now scale to the 1W input. This will also scale the SAR value.



5. You can close the window when you are finished viewing the results.

To view the Feed results:

1. In the Results workspace window, select:
    - Output Object: Feed
    - Result Type: S-Parameters
2. Double-click the result under the Discrete domain. The following results will appear showing the impedance at the feed, the input power delivered, and the return loss.



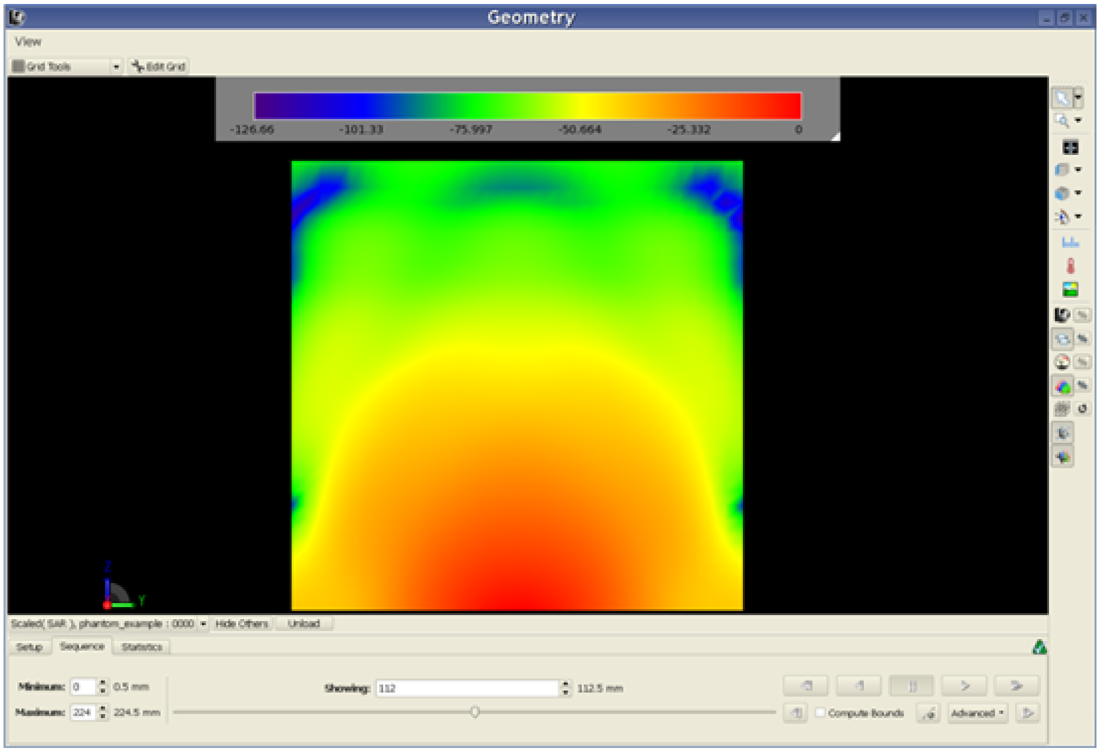
You can see from the table that our return loss is less than -30 dB, so you have a good match at the selected frequency.

3. You can close the window when you are finished viewing the results.

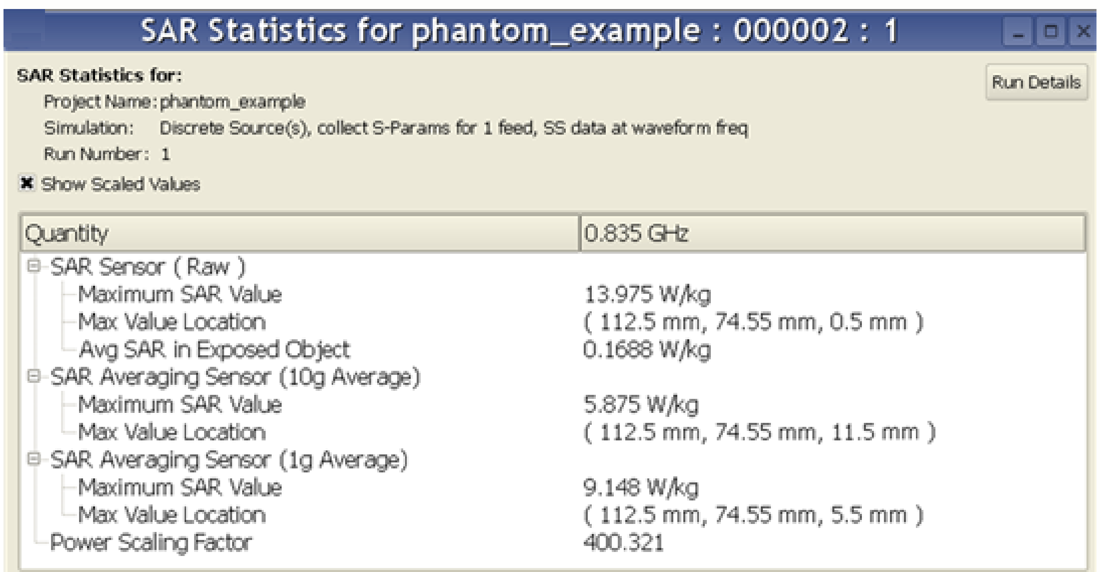
SAR Sensor Data

Now you will load the SAR data into the field viewer.

1. To view the SAR sensor data, select the following:
   - Output Object: SAR Sensor (Raw)
   - Result Type: SAR (Specific Absorption Rate)
2. Double-click the result in the filtered list. The plot will appear in the **Geometry** workspace window.
3. Under the **Setup** tab, adjust the following settings:
   - Sequence Axis: X
   - Display Mode: Flat
   - Decimation: Normal
   - Under Axis Ranges:
     - Y: Full
     - Z: Full
4. Toggle the **Parts Visibility** to turn off the display of the geometry, and select the **Left (+X)** orientation. The resulting image should appear.
5. Under the Sequence tab, define Showing: 112. The following SAR image appears.



6. Under the **Statistics** tab, choose **View all SAR Stats**. A summary table of the SAR values appears, as shown in the following figure:


SAR Statistics for phantom_example : 000002 : 1

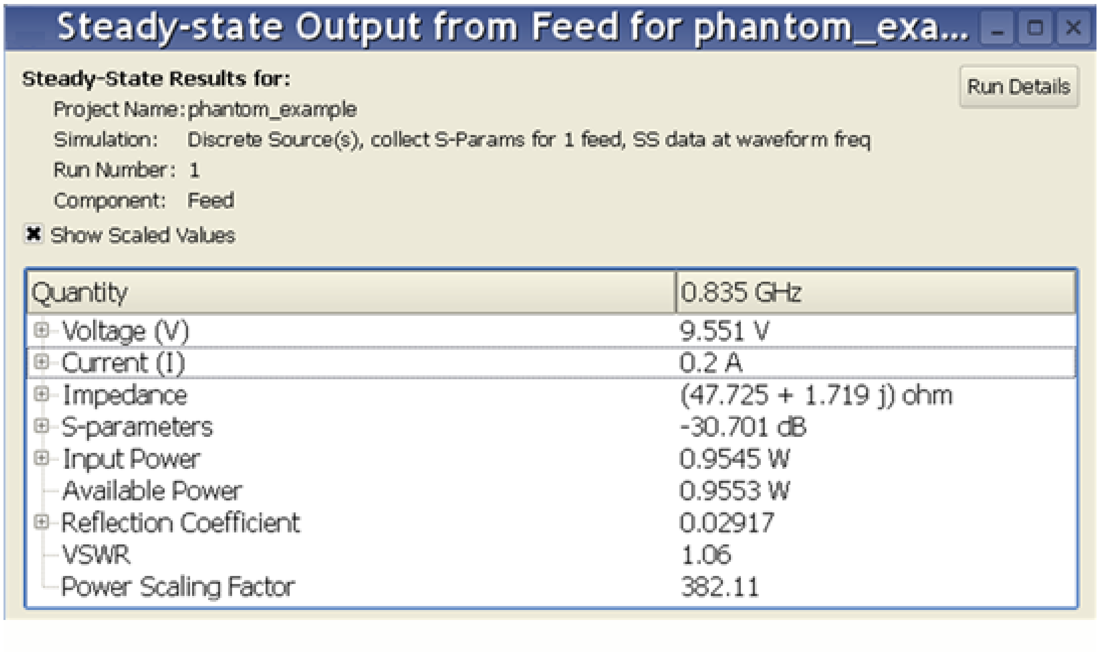
Run Details

SAR Statistics for:
Project Name: phantom_example
Simulation: Discrete Source(s), collect S-Params for 1 feed, SS data at waveform freq
Run Number: 1
Show Scaled Values

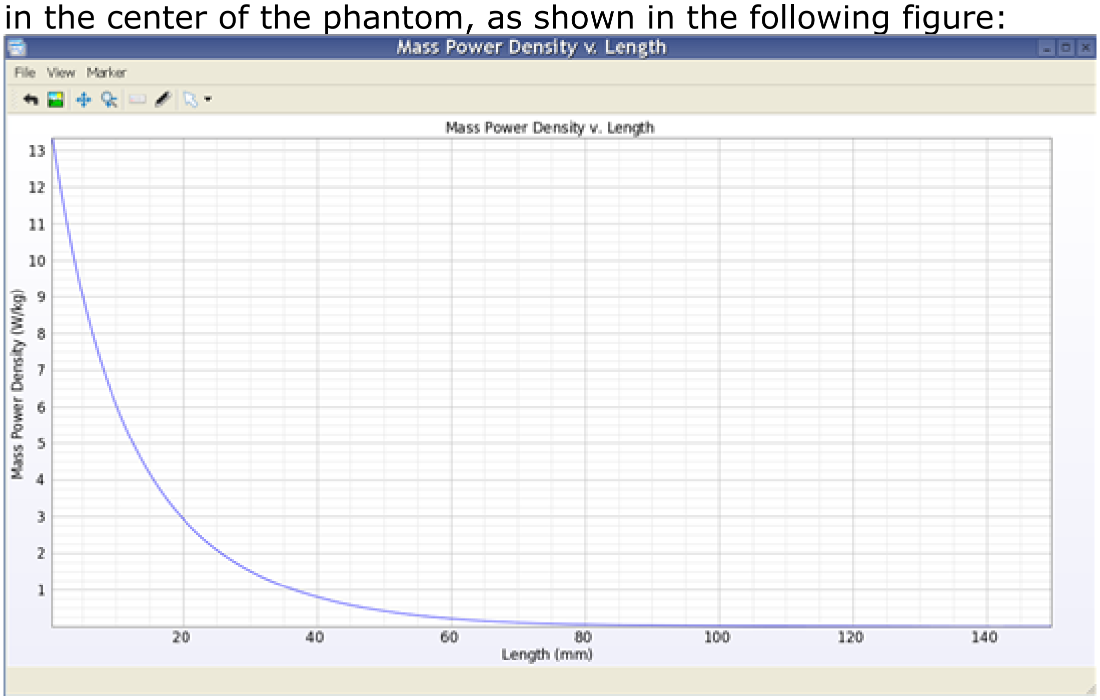
Quantity	0.835 GHz
SAR Sensor (Raw)	
Maximum SAR Value	13.975 W/kg
Max Value Location	(112.5 mm, 74.55 mm, 0.5 mm)
Avg SAR in Exposed Object	0.1688 W/kg
SAR Averaging Sensor (10g Average)	
Maximum SAR Value	5.875 W/kg
Max Value Location	(112.5 mm, 74.55 mm, 11.5 mm)
SAR Averaging Sensor (1g Average)	
Maximum SAR Value	9.148 W/kg
Max Value Location	(112.5 mm, 74.55 mm, 5.5 mm)
Power Scaling Factor	400.321

For some situations, the SAR results should be normalized to the feed point current rather than the forward power.

1. To make this adjustment, return to the **Results** workspace window and select:
   - Output Object: Feed
   - Result Type: Current
2. Double-click the result under the Discrete domain.
3. On this screen, the current value may be edited by clicking on the numerical value. Type in a value of 200mA and all results, including the SAR values, will be adjusted to a feed point current of 200mA.



4. You may close the window when you are finished viewing the results.
   It may be of interest to plot the SAR as a function of distance along a line extending above the feed point.
5. In the Results workspace window, select the SAR data.
6. Right-click on the result and select **Create Line Graph**.
   - Independent Axis: Z
   - X: 112
   - Y: 75 mm
7. Click **View** to see a line plot of the SAR as a function of distance from the feed point in the center of the phantom, as shown in the following figure:



GPU Acceleration for FDTD Simulations

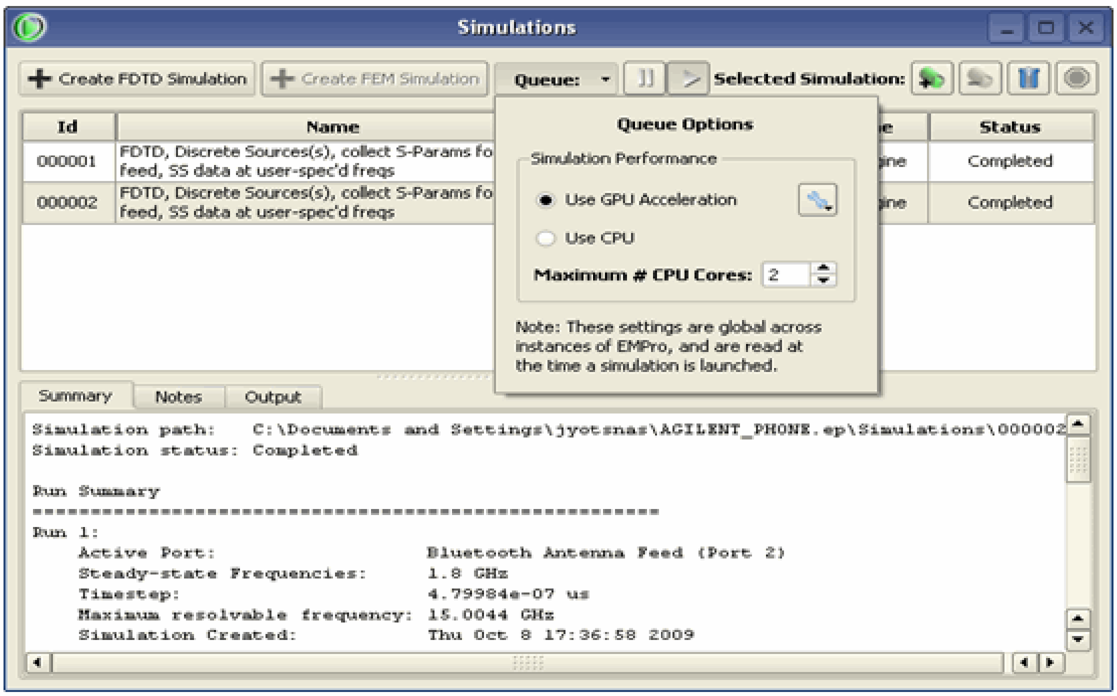
The FDTD simulations can be accelerated by using GPU hardware.

Two types of GPU acceleration solutions are supported:

- Legacy GPU acceleration
- Newer CUDA based acceleration.
  Both acceleration solutions require installation of GPU hardware and drivers. Specification of the required drivers and supported hardware cards is given below.

Selecting GPU Acceleration

You can turn on the GPU acceleration by choosing **Use GPU Acceleration** under **Simulations > Queue**.



To define the acceleration options, click  option and choose **Latest Acceleration** or **Legacy Acceleration**, as appropriate.

Supported Cards

The table below lists the cards supported by EMPro 2010 for GPU acceleration.

Operating System	Legacy Mode for Windows	Legacy Mode for Linux	CUDA for Windows	CUDA for Linux
Quadro 4500X2	Yes	Yes	No	No
Quadro 5600	Yes	Yes	Yes	Yes
Quadro Plex IV	Yes	Yes	Yes	Yes
Tesla S870	No	Yes	Yes	Yes
Quadro 5800	Yes	Yes	Yes	Yes
Tesla C1060	No	Yes	Yes	Yes
Quadro Plex D2	Yes	Yes	Yes	Yes
Tesla S1070	No	Yes	Yes	Yes

Supported Drivers

The table below lists the driver supported by EMPro 2010 for GPU acceleration.

Operating System	Legacy Mode for Windows	Legacy Mode for Linux	CUDA for Windows	CUDA for Linux
Quadro 4500X2	160.02	100.14.19	★	★
Quadro 5600	160.02	100.14.19	★	★
Quadro Plex IV	160.02	100.14.20	★	★
Tesla S870	X	100.14.21	★	★
Quadro 5800	181.20	180.22	★	★
Tesla C1060	X	180.22	★	★
Quadro Plex D2	181.20	180.22	★	★
Tesla S1070	X	180.22	★	★

★ : Use latest available drivers.

The drivers and installation instructions are available on the nVidia website .

Bibliography

1. D. M. Sheen, S. M. Ali, M. D. Abouzahra, and J. A. Kong, ``Application of the three-dimensional finite-difference time-domain method to the analysis of planar microstrip circuits,'' <EM>IEEE Transactions on Microwave Theory and Techniques</EM>, vol. 38, pp. 849-857, July 1990.

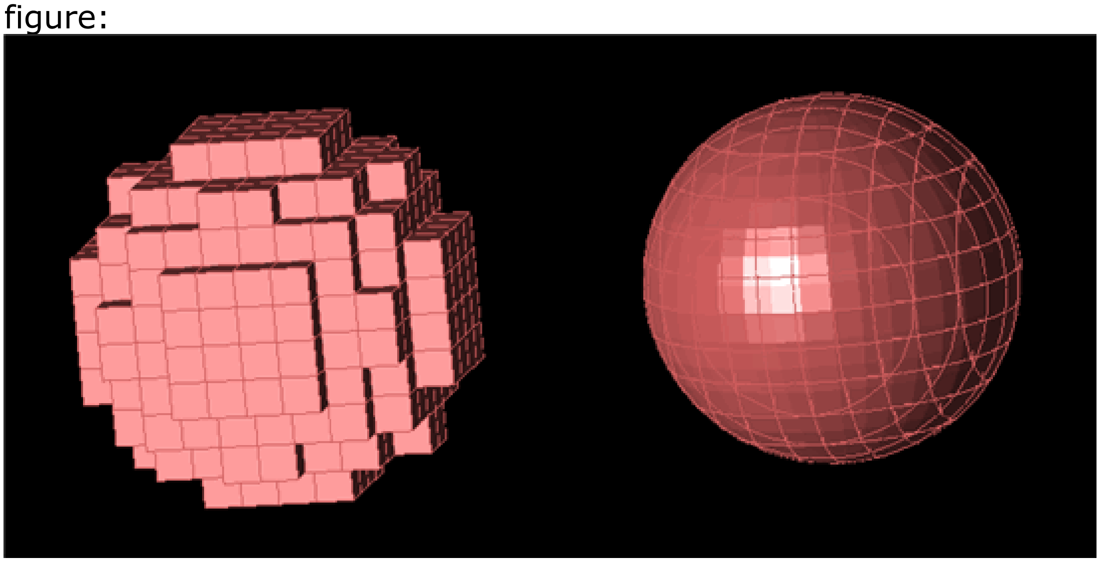
Using Conformal FDTD Meshing

Users can control the discretization of the domain in FDTD with standard meshing by setting appropriate values for the cell sizes. The cell size is often limited by the dimensions and the features of the geometric structure. This in turn determines the runtime cost and the memory consumption of the FDTD simulation. The new conformal meshing feature enables you to specify a coarser grid without compromising the accuracy too much. Conformal meshing consists of a set of techniques that use geometric information to provide a subcellular discretization of the computational domain in order to obtain accurate simulation results for a given grid resolution. In other words, one does not have to specify smaller cell sizes in order to increase the accuracy, or one can reduce the overall simulation time and memory requirements for a given level of the desired accuracy.

For example, a highly curved radiating body requires a fine grid with the standard meshing technique in order to obtain accurate results. Appropriately placed fixed points and grid regions make this grid refinement straightforward, and ultimately provide good accuracy in the simulation results. However, 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 an N^4 increase in simulation time. Therefore, it can be difficult with the standard meshing technique to sufficiently refine the computational grid while keeping the simulations fast and small. In this case FDTD with conformal meshing allows you to specify a coarser grid and thus reduces the computational costs while obtaining a similar level of accuracy.

Using conformal meshing:

- It is possible to represent gaps between conductors in the mesh without having a full grid cell distance between the conductors. This allows you to increase the cell size and timestep, resulting in shorter simulation times.
- Curved structures can be modeled more accurately than can be done with standard meshes only, since the conformal mesh represents the boundary of the object as actually curved inside of a grid cell. Thus, using conformal meshing can improve the accuracy of your simulations for a given grid resolution, as shown in the following figure:



Guidelines for using Conformal Meshing

A conformal mesh uses more geometry information than required by a conventional mesh for computing fields near the boundary of parts. This incurs execution time and memory usage penalties when conformal mesh is enabled. In order to avoid these penalties, you can refer the following guidelines while using the conformal mesh features:

- The conformal meshing feature is most useful for parts with high conductivity or Perfect Conductor electric material properties.
- Conformal meshing is useful 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 mesh will not benefit by using conformal meshing.
- Conformal meshing 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 a conformal mesh that maintains connectivity without enabling aggressive surface meshing as well.

Enabling Conformal Meshing

To enable the conformal meshing feature on a single part, perform one of the following steps:

- Right-Click the *Parts*: *Modelname* branch of the *Project Tree* and select *Meshing Properties* to open the **Meshing Properties Editor** window.
- Right-Click the *Parts* in the Project Tree and select **Enable Conformal Mesh** from the **Gridding/Meshing** sub menu.
- If an assembly is selected in the Project Tree, the right-click menu's **Subparts > Gridding/Meshing > Enable conformal Mesh on all Subparts**.

In order to mesh a part using the conformal meshing feature, the assigned material part 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 Good Conductor choice in the Material editor.

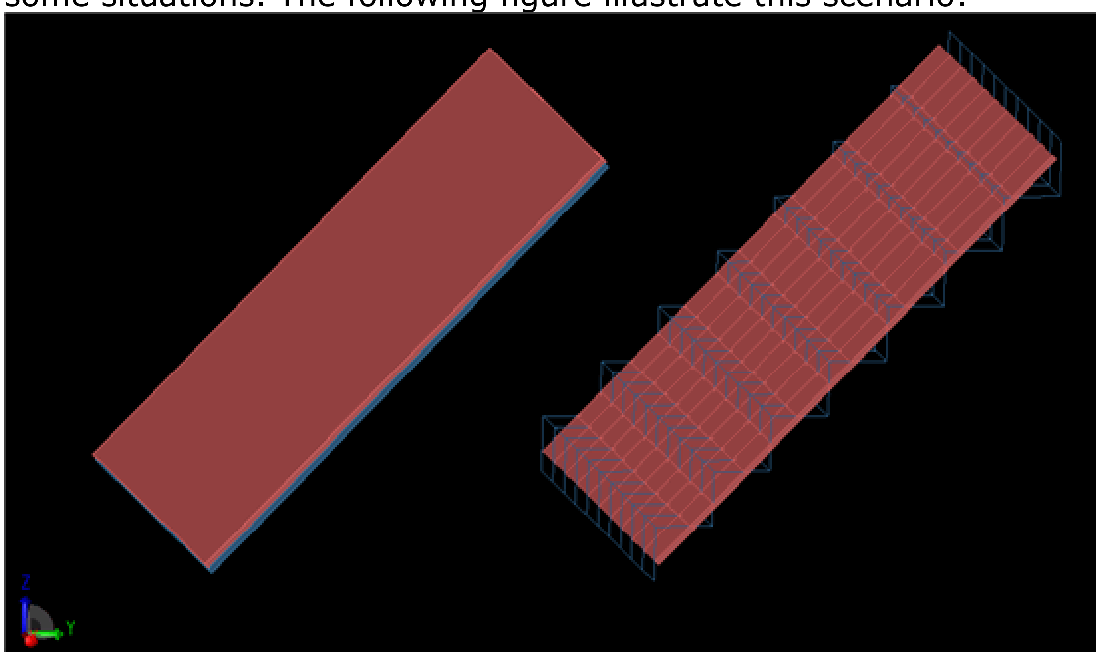
> The Conformal formulation for parts meshed with Good Conductor materials assumes a minimum of five skin depths thickness to the part of all locations, thus blocking all energy transfer through the part.

When the conformal meshing feature is enabled for any part in a simulation, EMPro, by default, reduces 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 Conformal Accuracy setting on the Advanced tab of the Project Properties Editor. This setting allows you to control the 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.

Limitations

In EMPro 2011.01, the conformal meshing feature has the following limitations:

- You can use the conformal meshing feature only with simulations excited by Discrete Sources and Total Field Plane Waves. You cannot use Scattered Field Plane Wave, Gaussian Beam, and Static Voltage Points with conformal meshing.
- You can enable the conformal meshing feature only on parts using nondispersive and isotropic electric materials. Some material types cannot exist on or near a part meshed with the conformal meshing feature in EMPro 2011.01. These include Dispersive, Thin Wire, and Nonlinear electric materials, and any magnetic material type other than Magnetic Freespace. Furthermore, when an Averaged Electric Material is encountered nearby a part meshed with conformal feature, the Dominant Material of the average is used in its place.
- In EMPro 2011.01, conformal-enabled parts cannot come within one cell of the simulation boundary.
- You should take care when a thin part with an Conformal mesh is present 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 conformal meshed parts. In such cases, the best recource may be to enable Conformal feature for both parts, although judicious use of fixed points may correct this behavior in some situations. The following figure illustrate this scenario:



- Parts having conformal meshing use a more faithful meshed representation of the geometry than a conventional Cartesian mesh can provide. This can have unexpected consequences in case where there are small gaps in your parts which a conventional mesh would typically short together. The conformal mesh will never do this. Thus, it is critical when using conformal meshing that the CAD representation of your parts contains no unwanted gaps.