Agilent Technologies

EMPro 2011.01
January 2011
Using EMPro

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:

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

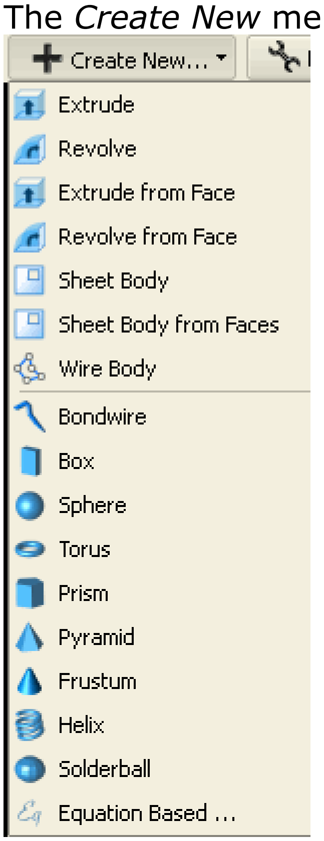
Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for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

Creating a New Geometry

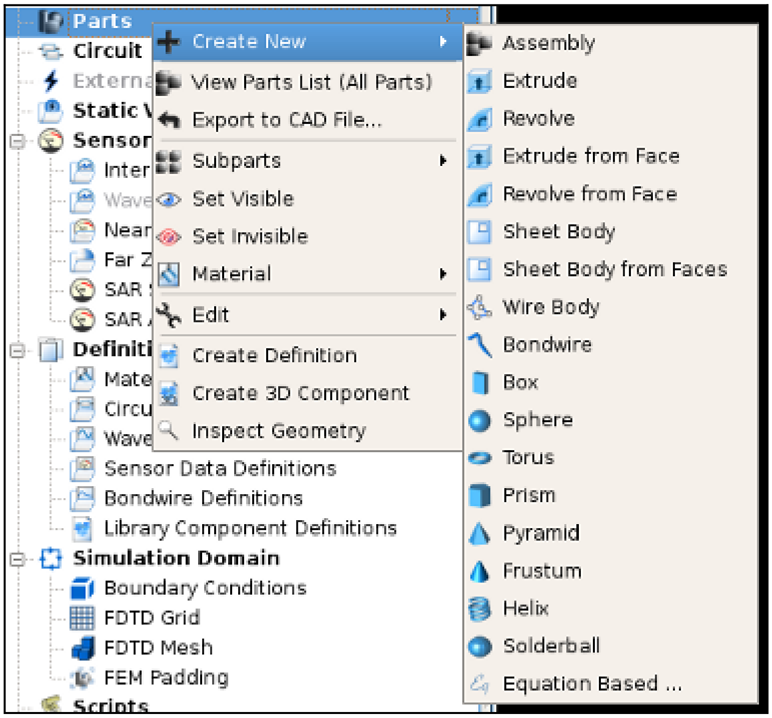
After selecting *Geometry Tools* in the drop down list of the *Geometry* workspace window, click **Create** to prompt a drop-down menu to appear. This menu includes the following modeling operations:

- Extrude
- Revolve
- Extrude from Face
- Revolve from Face
- Sheet Body
- Sheet Body From Faces
- Wire Body
- Bondwire
- Box
- Sphere
- Torus
- Prism
- Cylinder
- Pyramid
- Frustum
- Helix
- Solderball
- Equation based

The *Create New* menu items are displayed in the following figure:



Additionally, these tools are accessible from the Project Tree by right-clicking the Part branch, as seen in the following figure:



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

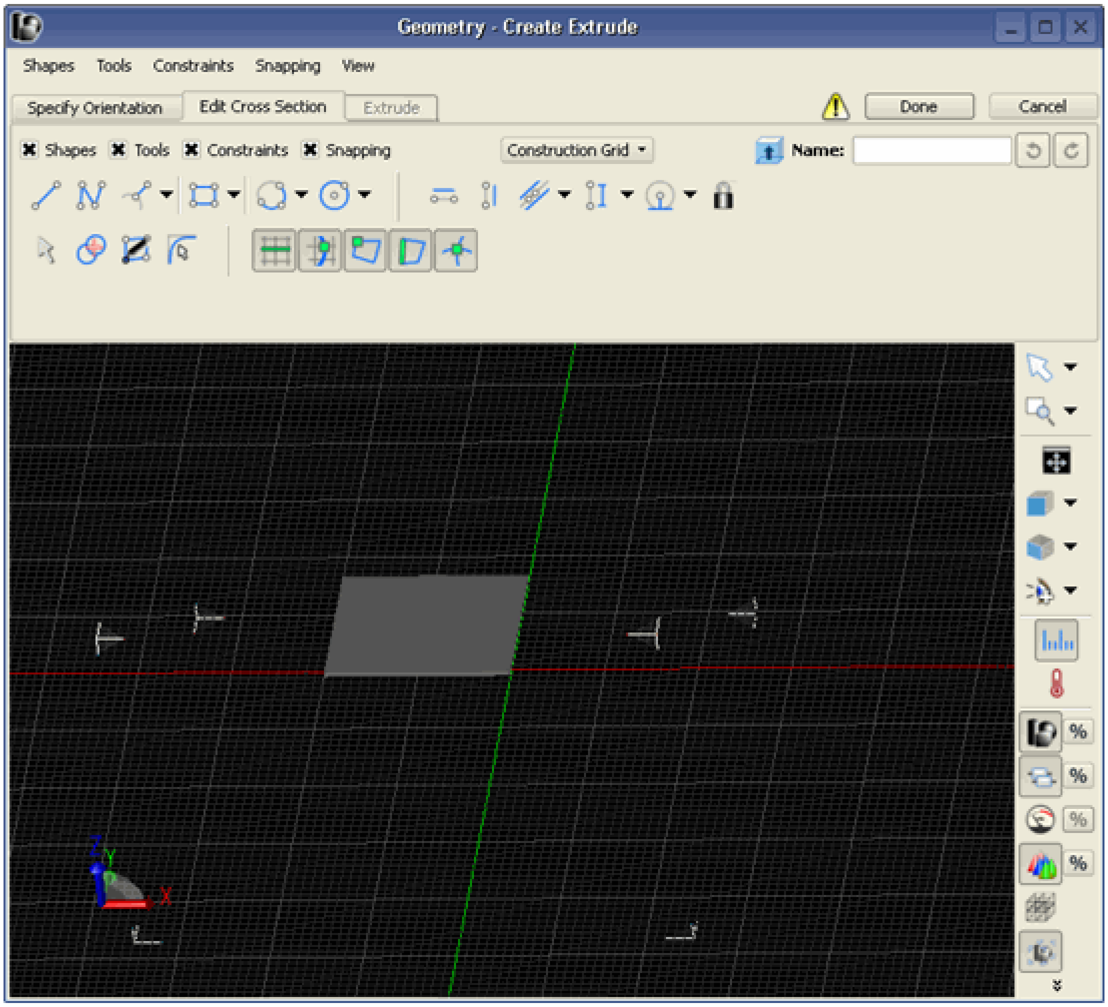
Extrude

Specify Orientation Tab

The *Specify Orientation* tab provides tools for orienting geometric parts in the simulation space. For a detailed discussion of this tab, refer to *Orienting Objects in the Simulation Space* (quickstart).

Edit Cross Section

In the *Edit Cross Section* tab, four toggle buttons including *Shapes*, *Constraints*, *Tools*, and *Snapping*. Each have a corresponding series of buttons in its own toolbar below. Clicking these labeled buttons will toggle the corresponding toolbars on and off. All of these buttons, in addition to the *View Tools* detailed earlier, are also available in the drop-down menus in the upper-left part of the screen. Additionally, a button labeled *Construction Grid* is available to edit the spacing of the visible grid lines in the 2-D sketcher. (This has no impact on the FDTD grid definition).



The current object is named in the box labeled *Name* in the upper right section of the *Edit Cross Section* tab. If a name is not defined in this box, the object is assigned a default name in the *Project Tree* when it is added to the project. The object can be renamed at any time in the *Project Tree* by right-clicking the object and selecting **Rename**.

To the right of the *Name* dialog box are two buttons, *Undo* and *Redo*. Clicking the **Undo** button will undo all actions carried out in the *Edit Cross Section* tab. Similarly, the Redo button will repeat any actions mistakenly erased during an undo operation.

Shapes

The *Edit Cross Section* tab contains a number of *Shapes* sketching tools that are useful for creating simple 2-D geometries for wire bodies and sheet bodies. They also serve as a common starting point to define 2-D cross sections for 3-D bodies such as extrusions, revolutions, and more complicated solid modeling operations. The *Shapes* tools are selected by clicking their respective icon.

- Pressing |**Esc**| or |**Backspace**| will back-up one step when using a multi-step creation tool.
- Pressing |**Esc**| a second time will deactivate the edge creation tool and activate the default *Select* tool.
- Pressing |**Tab**| will bring up a dialog to specify the position.
  The following is a list of the 2-D shapes available in EMPro:

- Straight Edge
- Polyline Edge
- Perpendicular Edge
- Tangent Line
- Rectangle
- Polygon
- N-Sided Polygon
- 3-Point Arc
- Arc Center, 2 Points
- 2-Point Arc
- Circle Center, Radius
- 3-Point Circle
- 2-Point Circle
- Ellipse

> **Note**
> For a detailed description of each shape tool, refer to *Shapes* (using) in the "Appendix of Geometric Modeling".

Tools

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

- Select/Manipulate
- Trim Curves
- Insert Vertex
- Fillet Vertex

> **Note**
> For a detailed description of each 2-D sketcher tool, refer to *Tools* (using) in the "Appendix of Geometric Modeling".

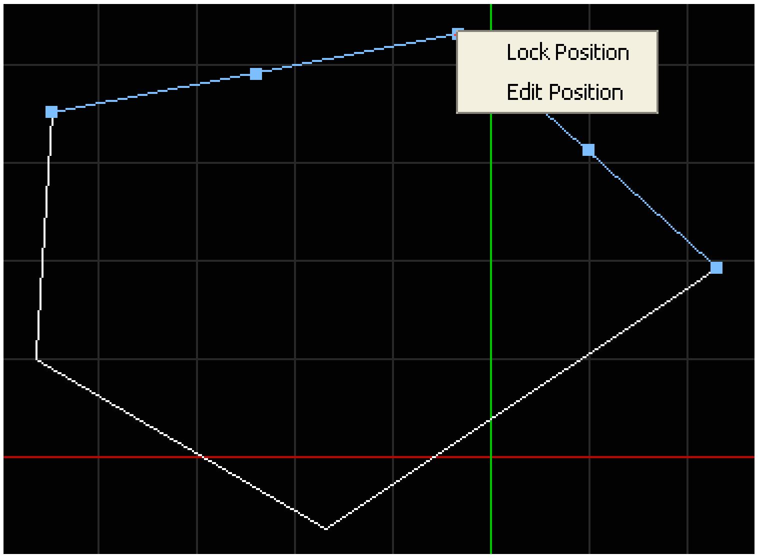
Constraints

Constraints are restrictions placed on geometric parts that must be satisfied in order to consider the model valid. They ensure that the user's intent is sustained throughout a calculation when parameters may change. Some objects are created with constraints already embedded. For instance, a rectangle is composed of four straight edges that are constrained perpendicularly as seen in preceding illustration. Other constraints are user-defined by means of *Constraint* tools.

Applying a constraint to an object will often affect other characteristics of the object. For instance, applying a horizontal constraint to one side of an irregular quadrilateral will most likely change the length of one or more sides and the angles that form with those connecting sides. Thus, it is important to lock any points that are intended to stay static. There are two main ways to do this:

- By selecting the *Lock Constraint* tool and clicking the the appropriate *vertex* or *side*.
- By selecting the *Select/Manipulate* tool, right-clicking the appropriate *vertex* or *side*, and selecting **Lock Position**, as shown below.

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



Note
For more about the *Select/Manipulate* tool's functionality, refer to *Select/Manipulate* (using) in the "Appendix of Geometric Modeling".

Each type of the following types of *Constraint* tools has its own green symbol or letter that is visible when the mouse is held over the constrained segment.

- Horizontal
- Vertical
- Collinear
- Parallel
- Perpendicular
- Tangent
- Concentric
- Angle
- Distance
- Equal Length
- Equal Distance
- Radius
- Equal Radius

Note
For a detailed description of each constraint, refer to *Constraints* (using) in the "Appendix of Geometric Modeling".

Snapping

Snapping tools are available to facilitate the exact placement of vertices on the sketching plane. When snapping is enabled, the mouse will be **snapped** to the closest of one or more snapping landmarks if one comes within range. For example, if *Snap To Grid Lines* is selected, the mouse is moved or snapped to points on the closest grid line as it is moved around in the sketching plane. This makes it much easier to place a vertex in the desired position without having to zoom in to a discrete position. Blue dots and blue lines represent the snapped location of the mouse when snapping is enabled.

In the case that the mouse is not within sufficient range of a selected landmark listed below, a vertex will be placed at its exact location on the sketching plane as if snapping were not turned on. (For example, if the mouse is dragged to the middle of a cell and the *Snap To Grid Lines* option is selected, the vertex will be placed in the center of the cell because it is not close enough to a surrounding grid line.)

Several snapping options can be selected at a time, in which case, the vertex will be snapped to the closest landmark that is within range of the mouse.

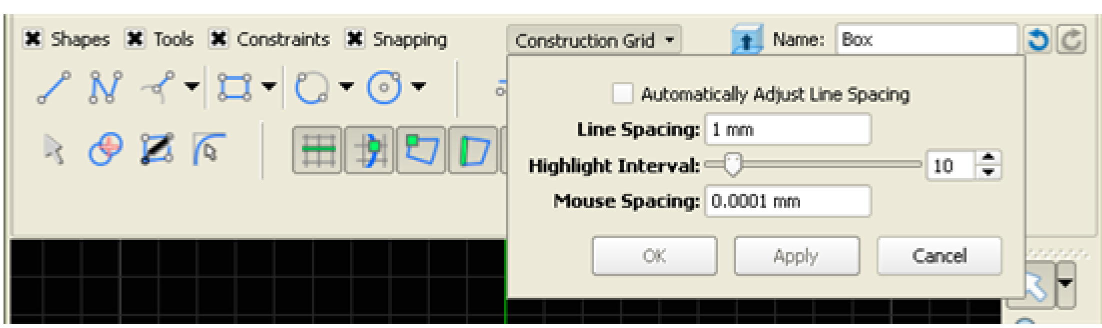
- Snap To Grid Lines
- Snap To Grid/Edge Intersections
- Snap To Vertices
- Snap To Edges
- Snap To Edge/Edge Intersections

Note
For a detailed description and image of each, refer to *Snapping* (using) in the "Appendix of Geometric Modeling".

Customizing the Construction Grid
The *Construction Grid* drop-down controls the appearance of the grid without impacting its actual cell size.

- *Automatically Adjust Line Spacing* causes the construction grid to adjust its line spacing with the current zoom level. As you zoom in, the lines are moved to be closer to each other. As you zoom out, they decimate and become further apart.
- *Line Spacing* is available when automatic isn't checked. This is the spacing between adjacent lines of the construction grid.
- *Highlight Interval* controls the interval which lines are highlighted. Every "Nth" line will be made bold.
- *Mouse Spacing* controls the minimum resolvable distance by the mouse. As you move the mouse, you will be unable to move between two points closer than this specified distance.

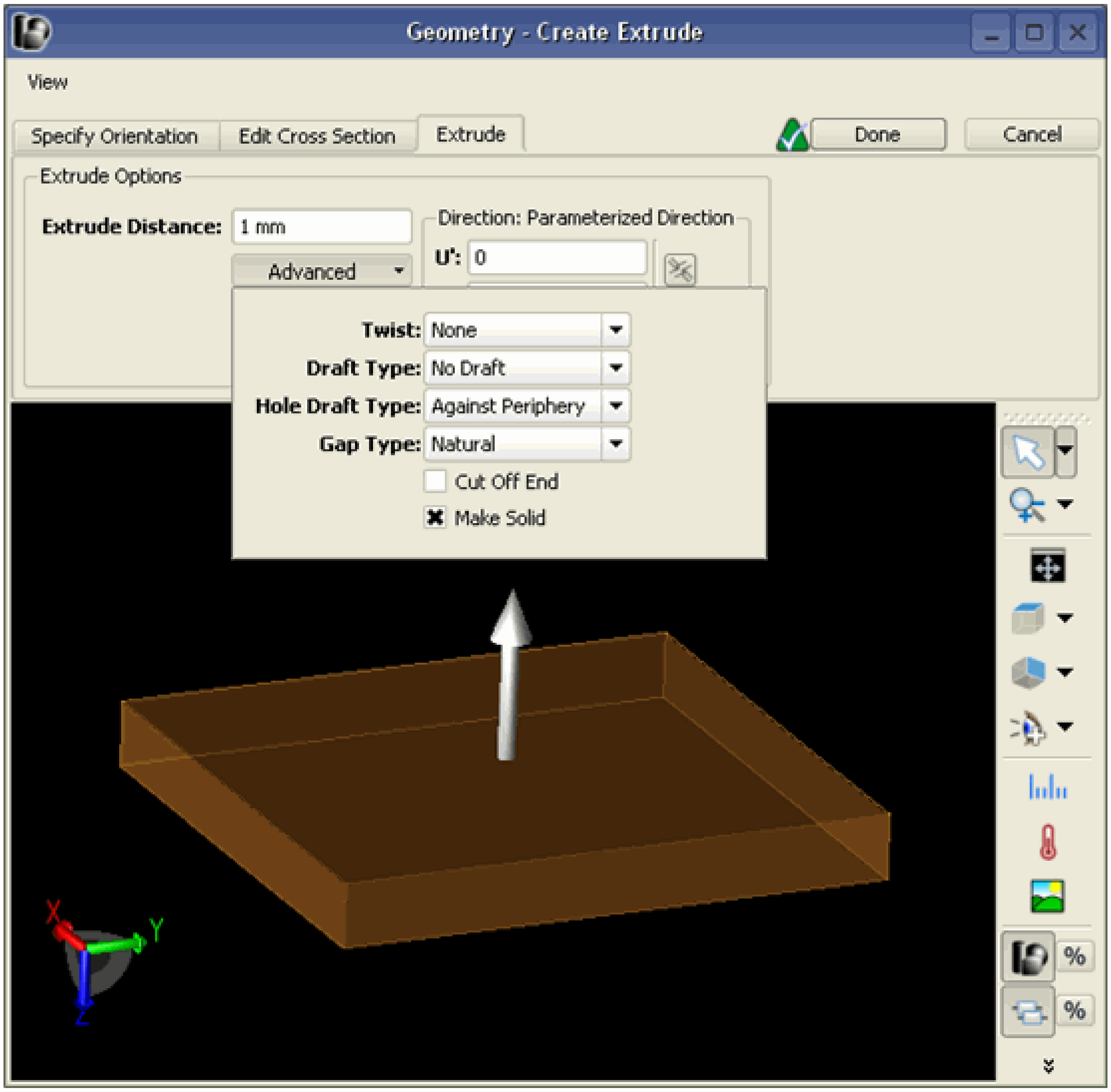
Editing the Construction Grid



3-D Operation Tabs

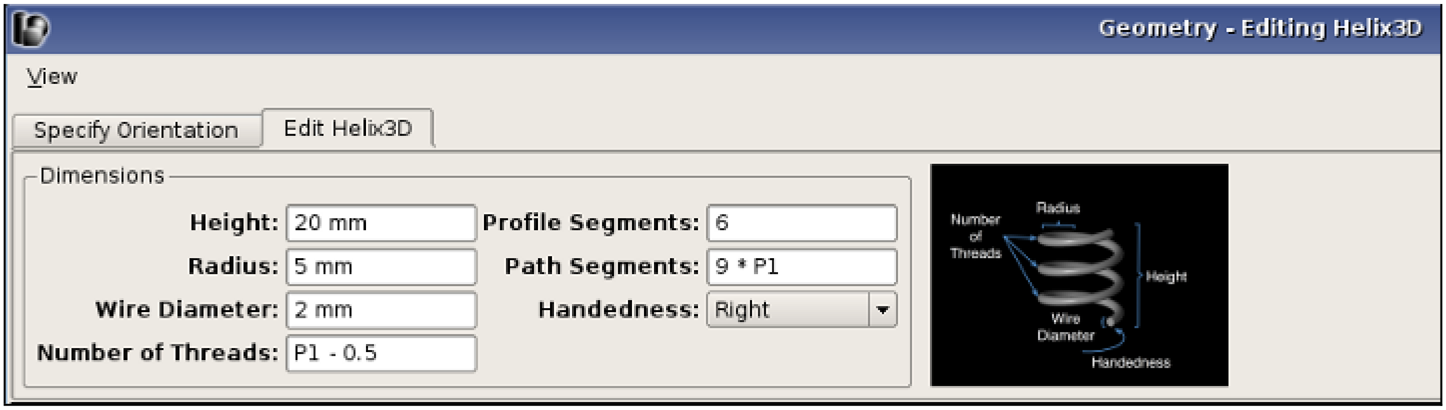
If subsequent tabs are available to the right of the *Edit Cross Section* tab, continue on to complete a 3-D operation. (These tabs are not available for 2-D objects.)

The figure below shows the *Advanced* drop-down menu inside of the *Extrude* tab, available when an *Extrude* operation is selected. This menu contains operations that can be applied to the 3-D object. For more information on these operations, refer to *Advanced 3-D Solid Modeling Operations* (using) in the "Appendix of Geometric Modeling".



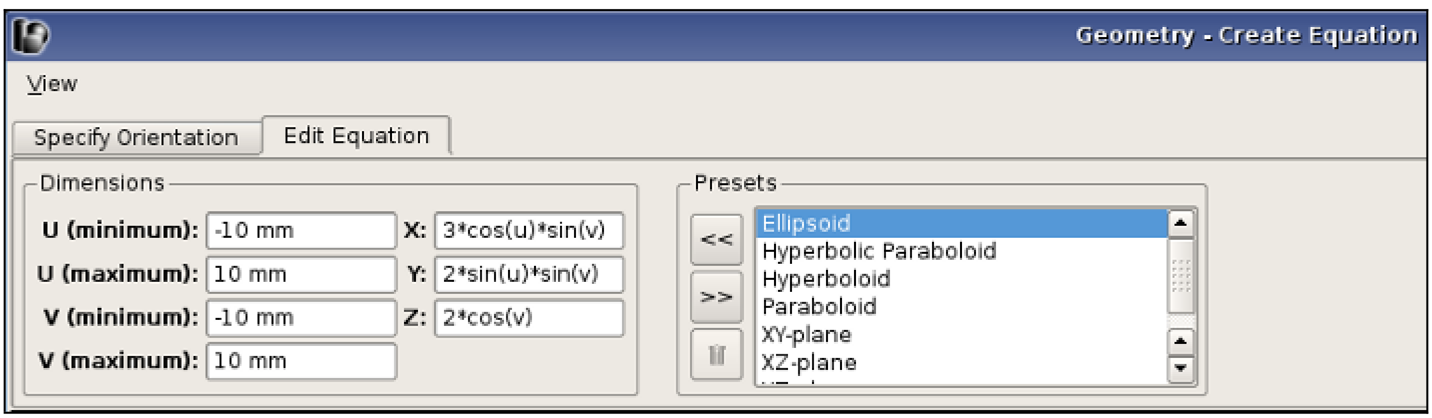
Primitive Building Blocks

EMPro has a built-in library of parameterized 3D objects that includes the following types of shapes: Bondwire, Box, Sphere, Torus, Prism, Pyramid, Frustum, Helix, Solder Ball, and Equation Based. A shape can be inserted by selecting the appropriate item from the *Create New* menu. With each of these shapes corresponds an editing tool, that lets you specify the shape's parameters, see folowing figure. By double clicking on the building block in the *Part* node of the Project Tree, one can (re)edit the parameters after insertion. Note: the editing boxes are all parameterizable, i.e., use of EMPro parameters, see for example parameter P1 in the figure below. The object can be positioned after insertion by using the *Specify Orientation* menu.



Equation Based Sheet Objects

Select *Equation Based ...* from the *Create New* menu to insert an equation. There are a number of presets available in EMPro, which can be loaded using the **<<** button. Users can add presets to EMPro which will also be available for other projects, i.e., presets are application (EMPro) specific and not project specific. Hereto the user must specify the **X-**, **Y-**, and **Z-** parameters and save the preset using the **>>** button. Note: these parameters may contain refrences to **U** and **V**, and/or mathematical functions/operators such as: **+, -, *, /, mod, abs, ceil, floor, pi, e, exp, log, ln, sqrt, arccos, arccosh, arcsin, arcsinh, arctan, arctanh, arccot, arccoth, arcsec, arcsech, arccsc, arccsch, cos, cosh, sin, sinh, tan, tanh, cot, coth, sec, sech, csc, csch**. User defined presets can be removed from the list, EMPro defined presets cannot be deleted.



Modifying Existing Geometry

The *Modify* button in the *Geometry* workspace window may be selected to modify the geometry of existing objects in the project.

- Specify Orientation
- Chamfer Edges
- Blend Edges
- Shell Faces
- Loft Faces
- Remove Faces
- Offset Faces

> **Note**
> For images of each of these operations, refer to *Modifying Existing Geometry* (using) in the "Appendix of Geometric Modeling".

Boolean Operations

The following boolean operations are available in EMPro:

- Two Parts
- Extrude
- Revolve

The *Two Parts* tool provides several boolean operations to subtract, intersect, or unite two objects. For these operations, one object must be selected to be the *Blank*, and the other the *Tool* which acts on the blank.

Holes may also be extruded or revolved through any part with its respective tool in this menu. An object is selected in the *Pick Blank* tab and the cross section of the hole is sketched and oriented in the *Edit Profile* and *Feature Orientation* tabs, as described in the *Edit_Cross_Section_Tab* and Specify_Orientation_Tab, sections respectively. Then the shape of the removed section is specified in the *Extrude Boolean* tab, or *Revolve* tab depending on which operation is selected. The *Preview* tab shows a preview of the object before the changes are formally applied to the project. For more information on defining extrusions or revolutions, refer to *3-D Solid Modeling Options* (using) in the Appendix of Geometric Modeling. An image of each boolean operation is available in *Boolean Operations* (using) in the "Appendix of Geometric Modeling".

Patterns

Patterns are created in EMPro by replicating a single selected object multiple times in one of the organized arrangements listed below.

- Linear/Rectangular
- Circular/Elliptical

> **Note**
> For the definitions and images associated with these patterns, refer to *3-D Patterns* (using) in the "Appendix of Geometric Modeling".

Creating Materials

In this section, you will learn how to:

- Add materials to your EMPro project
- Define materials and apply them to geometric objects

Once objects are created and situated correctly in the simulation space, material definitions must be added or else the project *will not* be considered valid. The electrical and magnetic materials available within EMPro are detailed in this section.

The *Material Editor* window is the main interface used to define materials to be applied to objects in a simulation. The series of tabbed windows within the editor are used to define a material based on its constitutive parameters. After adding materials to the project, simply drag-and-drop the material in the *Project Tree* onto the desired geometry to apply it to that object.

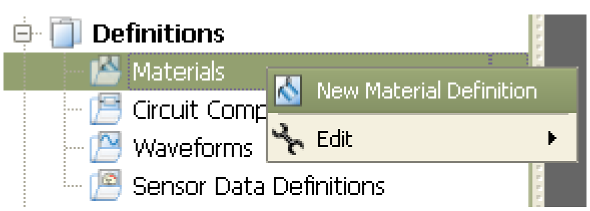
The following section describes the options under each tab within the *Material Editor*.

Adding a New Material

To add a new material, right-click **Definitions: Materials** branch of the *Project Tree* and select *New Material Definition*, as seen in the following illustration. A *Material* object will be added to this branch. Depending on the project preferences, the *Material Editor* window will appear automatically. If not, simply double-click on this object to bring up the editor. Similarly, double-click on any existing *Material* icon to edit an existing material within the *Material Editor*.

> For more information about project preference definitions, refer to *Application Preferences* (using).

Adding a New Material Definition to the project



Once the *Material Editor* window is open, type in the name of the new material in the *Name* dialog box. Define the *Type* of material as *Physical* or *Freespace*. *Freespace* is the most basic material definition. Every other type of material is included within the *Physical* definition, in which case the *Electric* and *Magnetic* types should be assigned in their respective drop-down lists below.

There are five electrical and magnetic material types available in EMPro.

- Freespace

- Perfect Conductor
- Isotropic
- Diagonally Anisotropic
- Anisotropic (Electric only)

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

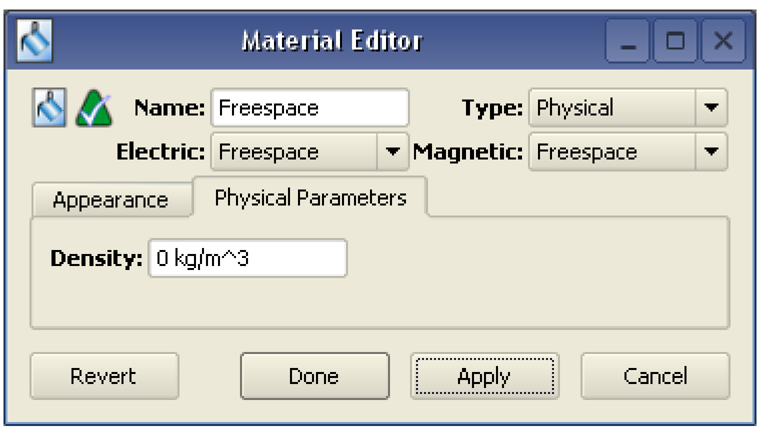
The following sections will detail the various types of materials.

Freespace

Freespace is the most basic material. By default, the EMPro problem domain is initialized to free space. This material sets relative permittivities and permeabilities to one, and conductivities to zero.

The following figure shows the *Material Editor* when the *Freespace* material is defined. Notice that no *Electric* or *Magnetic* tab is available, since both are defined as *Freespace* material.

Defining a Freespace material



Perfect Conductors

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

Note
You can read more about the *Surface Conductivity* box in the *Complex Permittivity, Loss Tangent and Surface Conductivity Correction Overview* section.

Electric Materials

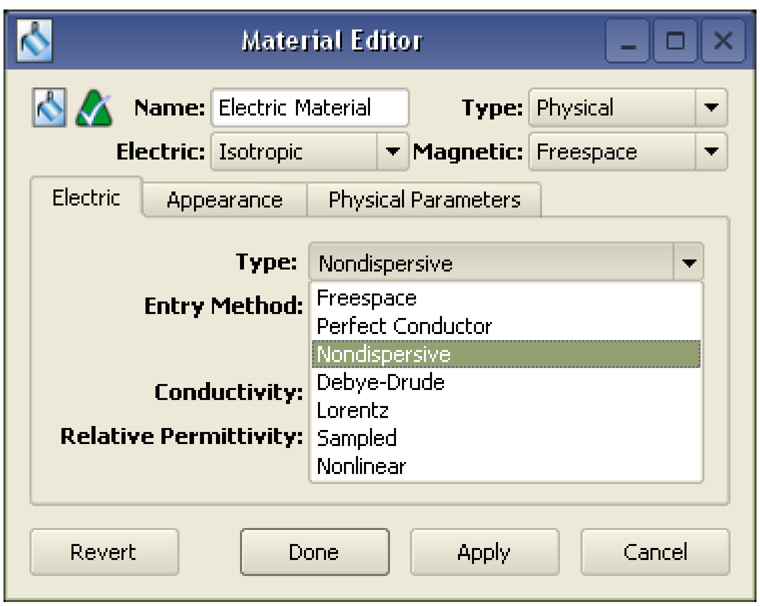
Isotropic Materials

EMPro includes several *Isotropic* materials:

- Nondispersive
- Debye/Drude
- Lorentz
- Sampled (FDTD only)
- Nonlinear (FDTD only)

The next figure shows the *Material Editor* when an *Isotropic* material is defined. Note that only the *Electric* tab is available since *Magnetic* is defined as *Freespace*. If *Magnetic* was defined as another type, a *Magnetic* tab would be available as well.

Defining an Electric material



Nondispersive

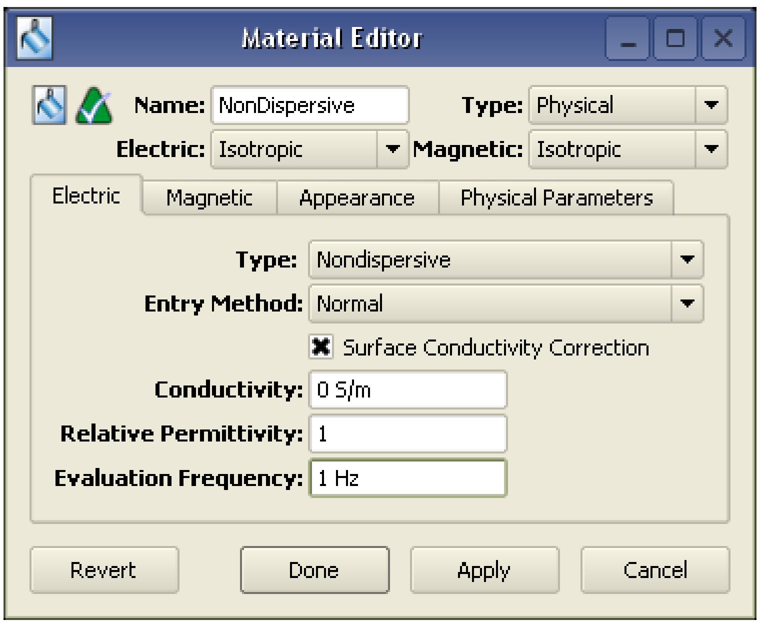
Nondispersive material properties do not very with frequency. The continuous-time expressions of Maxwell's equations for linear, isotropic, nondispersive materials which will be discretized in EMPro are:

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

where:

- ϵ represents the electric permittivity,
- σ represents the electric conductivity,
- μ represents the magnetic permeability, and
- σ^{*} represents the magnetic conductivity.

Defining a Nondispersive material

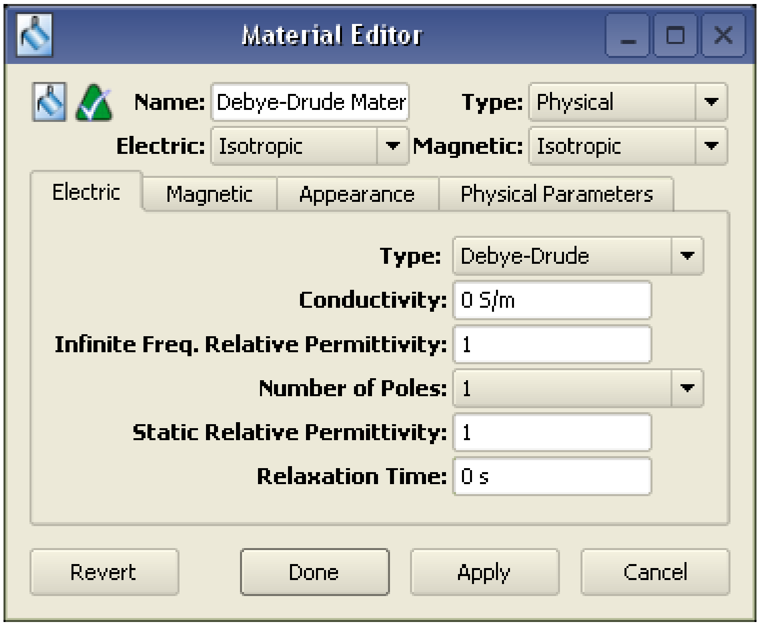


Debye/Drude

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

> **Note**
> This is discussed in detail in Chapter 8 of the Kunz and Leubbers text [*1* (using)].

Defining a Debye/Drude material



These parameters cannot be set arbitrarily or instability can occur in FDTD simulations. One constraint is that the FDTD timestep must be small enough to accurately calculate the transient behavior of the material. If the timestep is 3% of the relaxation time or smaller, the time variation of the material parameters should be sufficiently resolved. Typically, the timestep is a very small fraction of the relaxation time. In order to be clear about the signs in the following discussion, note that we are using the engineering time variation of:

$$-j\omega t$$

and we are defining the complex permittivity as:

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

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

Note
See equation 8.29 of the Kunz and Leubbers text [*1* (using)].

For a *Debye* material ($\sigma = 0$), stability is assured by setting ϵ_r to a larger value than ϵ_∞ .

In order to have realistic behavior at high frequencies, ϵ_∞ should be no less than one and should not be much larger than one. Thus the condition for strictly *Debye* material to be stable for FDTD simulations is:

$$\epsilon_\infty \geq 1$$

$\epsilon_r > \epsilon_\infty$
$\sigma = 0$

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

$\epsilon_\infty \geq 1$
$\epsilon_r > \epsilon_\infty$
$\sigma \geq 0$

These conditions are, however, too restrictive to specify general *Drude* materials. The more general *Drude* conditions are: $\epsilon_\infty \geq 1$

If ($\epsilon_r < \epsilon_\infty$),

then: $\sigma \geq \epsilon_0 \frac{\epsilon_\infty - \epsilon_r}{\tau}$

otherwise: $\sigma \geq 0$

where ϵ_0 is the *Freespace Permittivity* of 8.854e-12 !img227.png!.

> **Note**
> More general conditions for *Drude* materials can be determined from the discussion in Chapter 8, Section 3 the Kunz and Leubbers text [*1* (using)].

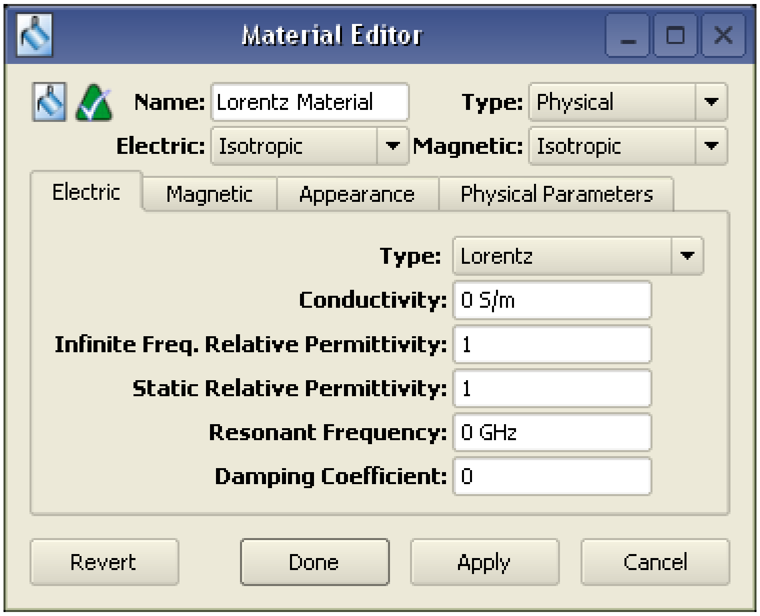
Lorentz

Stability in *Lorentz* materials for FDTD simulations should be obtained as long as *Conductivity* ≥ 0 and the FDTD timestep is 3% of the relaxation time or less. The limits on the material parameters are:

$\sigma \geq 0\frac{S}{m}$
$\epsilon_\infty \geq 1$
$\epsilon_r \geq \epsilon_\infty$

$ResonantFrequency\ (\Omega) > 0$
$DampingCoefficient > 0$

Defining a Lorentz material

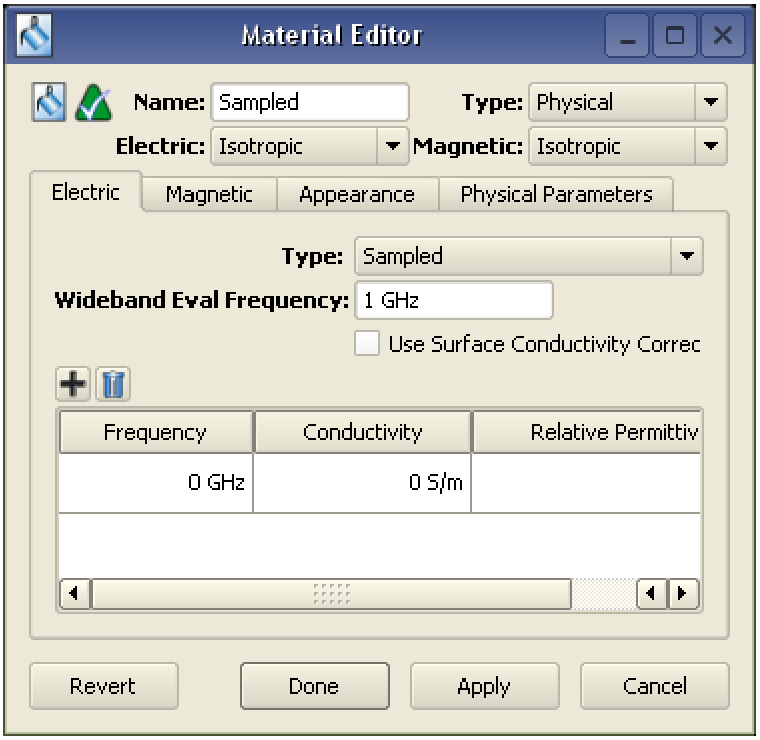


Sampled

This material enables you to enter multiple relative permittivities and conductivities at one time. It will behave like a nondispersive material when the calculation engine is called and the *Wideband Eval Frequency* dictates what parameters to use.

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

Defining a Sampled material



Nonlinear

The relative permittivity of a nonlinear isotropic dielectric material satisfies:

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

Where:

ϵ_r is relative permittivity

E is instantaneous cell edge E-field

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

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

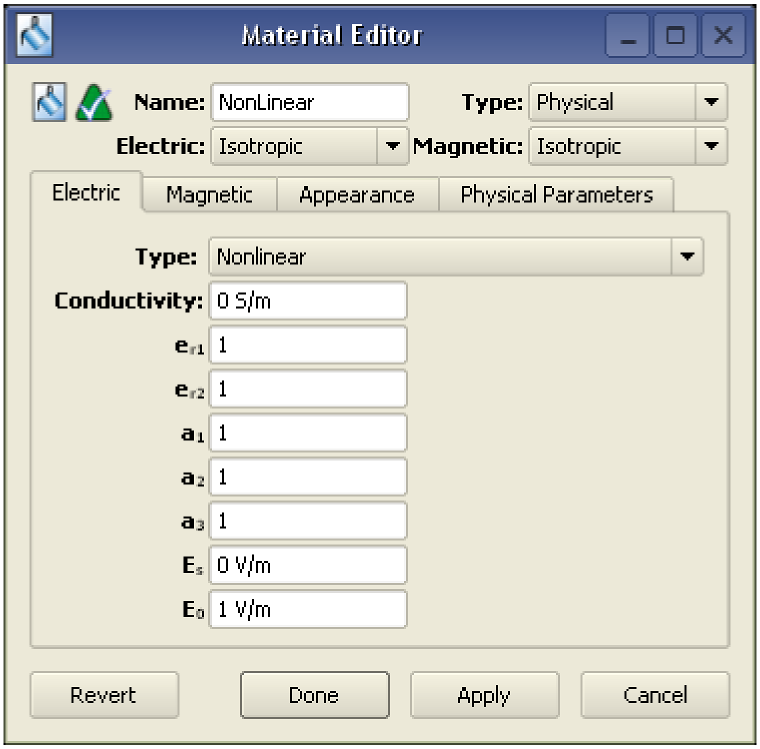
E_s is the E magnitude above which the material becomes non-linear

E_0 is a scaling term

a_1, a_2 and a_3 are coefficients

Note
Nonlinear materials are not supported in FEM simulations.

Defining a Nonlinear material



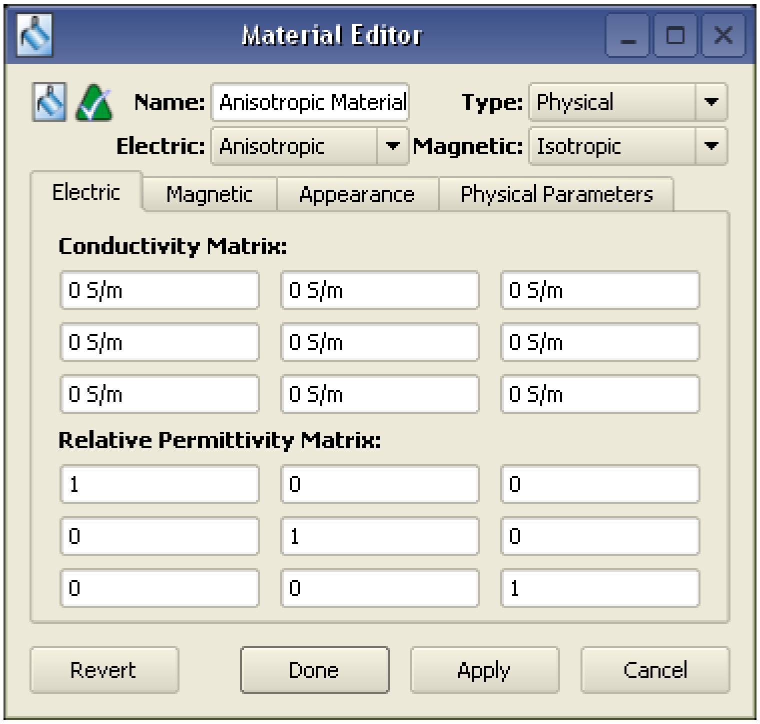
Diagonally Anisotropic

The definitions for a *Diagonally Anisotropic* are equivalent to those corresponding definitions detailed for *Isotropic* materials, except the definitions in each of the principle directions are independently specified.

Anisotropic

Frequency-independent *Anisotropic* materials are defined in EMPro by the relative permittivity, $\bar{\bar{\epsilon}}$, and *Conductivity*, $\bar{\bar{\sigma}}$, tensors.

Defining an Anisotropic material



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

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

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

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

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

Complex Permittivity

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

$$\sigma = \omega_o \epsilon'' \epsilon_o$$

Loss Tangent

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

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

Surface Conductivity Correction

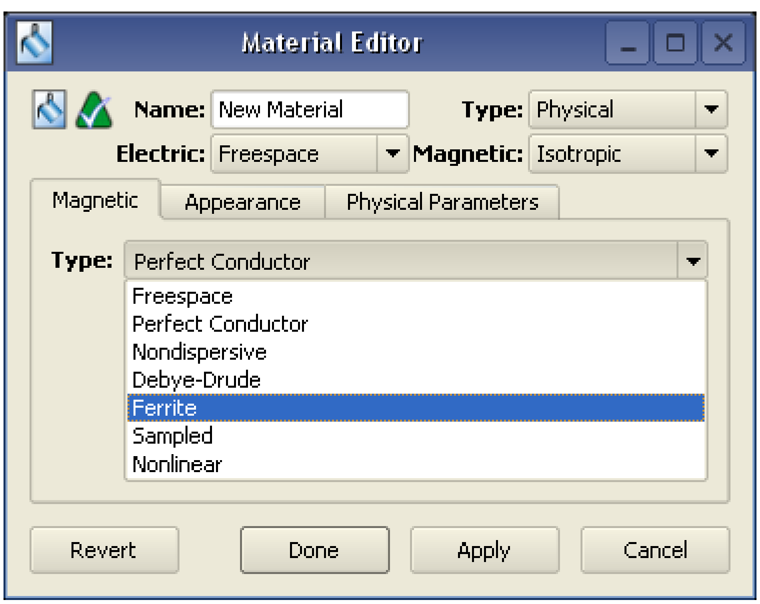
In the case of a frequency-dependent conductor, the *Surface Conductivity* Correction_ box can be checked to correct the conductivity of a material for a single frequency of a sinusoidal excitation. This is necessary in cases where the penetration and loss in good conductors needs to be included in the calculation. Enabling this option allows for the reduction of wavelength in these materials without reducing the cell size to maintain the 10 !img249.png! limit.

Magnetic Materials

EMPro also includes several types of magnetic materials. Many of these materials are simply the magnetic counterpart to the dielectrics described in Electric Materials. All restrictions noted in the *Electric Materials* section apply to their magnetic counterparts.

The figure below shows the *Material Editor* when a Magnetic *Isotropic* material is defined. Note that only the *Magnetic* tab is available since *Electric* is defined as *Freespace*. If *Electric* was defined as another type, a *Electric* tab would be available as well.

Defining a Magnetic material



Isotropic

Nondispersive

- See Nondispersive in the *Electric Materials* section.

Debye/Drude.

- See Debye/Drude in the *Electric Materials* section above.

Magnetized Ferrites.

The first parameter related to magnetized ferrites is the *Applied Field*, (H_o). Enter its value in units of $\frac{A}{m}$. This number will be used to calculate the Larmor precession frequency (ω_o), $\omega_o = \gamma H_o$

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

Next, enter the *Internal Magnetization* (M_o) in units of T . This number is used to calculate the saturation frequency (ω_m), $\omega_m = \gamma \frac{M_o}{\mu_o}$.

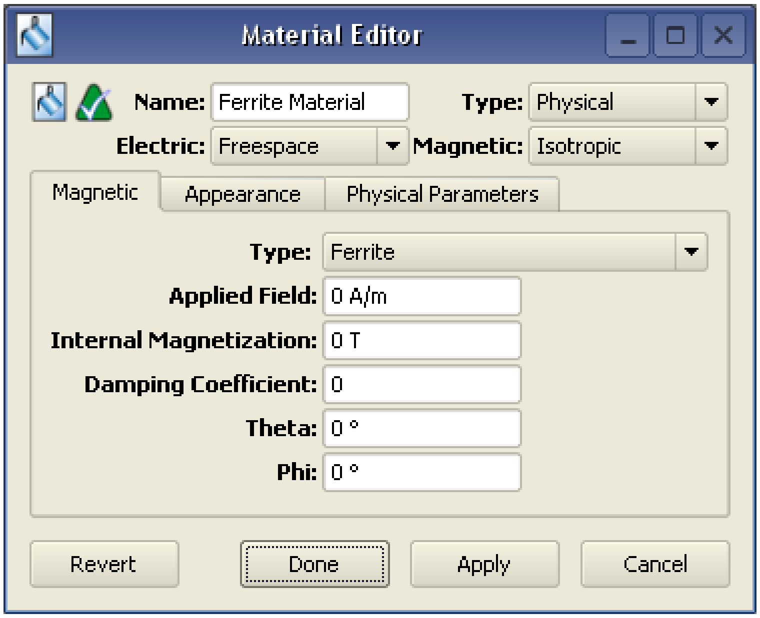
Then, use the *Damping Coefficient* to account for damping in the ferrite or of any absorption of power due to the ferrite. Finally, enter the direction of the biasing field using the spherical direction fields THETA and PHI.

Note
There are several informative references that discuss the form of the permeability tensor used for the ferrites [*5* (using),*6* (using),*7* (using),*8* (using)]. (The first two references do not discuss the damping coefficient.)

See the Kung text for parameters for some commercially available ferrites [*8* (using)].

Note
Ferrite materials are not supported in FEM simulations.

Defining an Magnetized Ferrite material



Sampled.

- See Sampled in the *Electric Materials* section above.

Nonlinear.

- See Nonlinear in the *Electric Materials* section above.

Diagonally Anisotropic

- See Diagonally Anisotropic in the *Electric Materials* section above.

Appearance

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

Physical Parameters

The *Physical Parameters* tab governs the definitions most commonly associated with biological tissue. These definitions are thus necessary when performing biological calculations. These values are computed automatically for tissues in Agilent Technologies high fidelity meshes.

Defining Outer Boundary Conditions

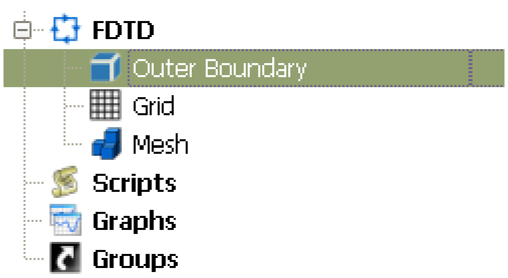
In this section, you will learn about how to:

- Distinguish absorbing and reflecting boundaries in EMPro
- Choose which boundary type to use for your project calculation

Specifying an outer radiation boundary is necessary to indicate how the calculation treats the boundaries of the problem space. During an EMPro calculation, the fields updated at every cell location are dependent upon the neighboring fields. However, due to memory limitations, the fields on the outer edges of the grid cannot be updated correctly because the grid must be a finite size. To correct this situation, outer radiation boundary conditions are applied at the edges of the EMPro grid. Thus, the performance of the outer boundaries is a significant factor in the accuracy of the calculation, and it is important to use them correctly. This chapter details several available options for defining the outer boundaries of an EMPro project.

Outer boundaries are defined in the *Outer Boundary Editor*, located in the *Fdtd: Outer Boundary* branch of the *Project Tree*, as shown below.

Outer Boundary Branch



Absorbing Boundaries vs Reflecting Boundaries

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 infinitely; however, it is actually finite in order to fall within reasonable memory usage. There are two numerical absorbers designed to allow electromagnetic fields radiated or scattered by the FDTD geometry to be absorbed with very little reflection from the boundary. These include a *Uni-Axial Perfectly Matched Layer* (PML) and a second-order, stabilized *Liao* radiation boundary.

In some cases a reflecting boundary rather than an absorbing one is preferred. A perfectly conducting boundary (either electric, PEC, or magnetic, PMC) may be used in these cases, for example, to provide a ground plane, or to image the fields in an EMPro calculation.

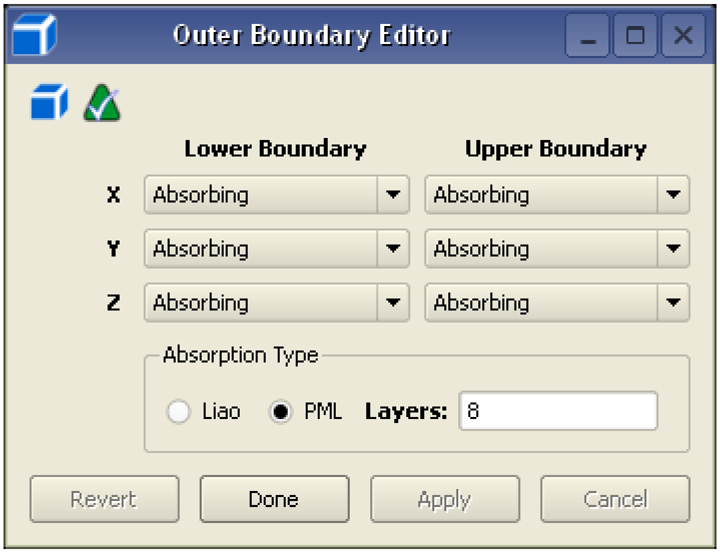
The Liao and PML boundaries may not be mixed together in the same calculation. Furthermore, PML may not be used with the PMC boundary. The Liao boundary may be used with both PEC and PMC boundaries.

> **Note**
> The default boundary condition for EMPro is PML.

In addition, EMPro has *Periodic* boundary conditions that enable periodic structures to be modeled. These boundary conditions equate the corresponding outer surfaces of the mesh.

The figure below shows the *Outer Boundary Editor*.

The Outer Boundary Editor



Liao Boundary

The *Liao* outer boundary condition is an estimation method, which is makes it fundamentally different from PML boundary conditions. By looking into the FDTD space and back in time, it estimates the electric fields just outside the limits of the FDTD mesh. These estimated values are then used in the FDTD equations inside the space. The Liao estimation assumes that waves are allowed to travel outward from the space but not reflect back in. This method works well provided that there is enough space between the radiating geometry and the outer boundary. Typical limits are at least 10 cells of spacing to ensure that instability does not occur.

> **Note**
> For more on calculation instability, refer to the section on *Calculation Stability* (using).

A homogeneous dielectric may be located against the Liao boundary. For example, in a lossy earth or strip line calculation, the earth or dielectric layer may touch the outer boundary. Liao will usually function well in this situation provided that there are no air gaps within five cells of the Liao boundary. Liao assumes homogeneous material within five cells, and if this is not the case then the EMPro calculation will usually be unstable with rapidly rising field amplitudes.

Since Liao is an estimation method, the size of the FDTD mesh is not increased by using it. Some storage is needed for saving electric values at previous timesteps, but this is usually negligible in a typical calculation.

PML Boundary

The *Perfectly Matched Layer* (*PML*) boundary condition is offered as an alternative to Liao. PML is an artificial absorbing material that absorbs the incident energy as it propagates through the PML layers. Better absorption, that is, smaller reflection, is obtained by adding more layers at the expense of increasing the size of the FDTD mesh. For example, consider an EMPro calculation on a mesh using the Liao absorber that is 50 x 60 x 70 cells or a total of 210,000 cells. There is a 15 cell free space border all around the geometry so that the Liao boundaries can provide small reflections. If the Liao is changed to eight PML layers, the geometry mesh will not change. However, outside of this defined mesh region, eight additional FDTD mesh layers are added on each side of the geometry. This means that the actual number of FDTD cells that must be calculated grows to 66 x 76 x 86 or 431,000 cells, more than double. Since PML cells require more arithmetic operations than normal cells, the time penalty is actually greater.

This time penalty for PML is also increased because the PML cells have special equations for both electric and magnetic fields. For an EMPro calculation with no magnetic materials present, the magnetic fields are computed very quickly. However, when PML is added, the magnetic field update equations are more complicated even when no actual magnetic fields are present and this adds to the time penalty.

The benefit of using the PML layers is that they provide better absorption than Liao even with only a five-cell border of free space, and perhaps only six PML layers would provide this. In such a situation, calculation time would be saved. Making this comparison would require meshing the object again with a smaller free space margin to the outer boundary. This can be done easily in EMPro using the mesh tab and choosing a smaller padding around the geometry.

Both PML and Liao boundary conditions are offered to provide flexibility. Both methods should provide similar results when properly used although in some cases, particularly when low frequencies (compared to the cell size) are used, PML is superior. It is also recommended that PML boundary conditions are used wherever possible when using the adaptive meshing feature.

PEC Boundary

In some situations, terminating one or more faces of the FDTD geometry space with a *Perfect Electric Conductor* (PEC) outer boundary is advantageous. For example, the conducting ground plane of a microstrip could be located on one face of the FDTD space.

If all of the outer boundaries of the calculation are not absorbing, a plane wave should not be used to excite the calculation and the far-zone transformations will not provide correct results for far-zone fields. The sole exception is in the case of one PEC boundary and five absorbing boundaries, which will compute far zone over infinite PEC ground.

> **Note**
> An edge of the FDTD space should be set to PEC using the PEC Boundary Condition. **Do not** set FDTD cells to PEC material in the geometry *and* set the outer boundary to absorbing, as this will cause instabilities in the calculation.

PMC Boundary

The *Perfect Magnetic Conductor* (PMC) outer boundary condition may be useful in reducing the size of the FDTD mesh, memory requirements, and calculation time by taking advantage of symmetries in the geometry. For example, this condition would be a good choice in a symmetric problem space where magnetic fields are strictly normal to a plane.

Periodic Boundary

Similar to the PMC boundary condition, the *Periodic* boundary condition may be useful in taking advantage of geometry/field symmetry to reduce the size of the FDTD mesh and therefore the memory and calculation time required. In this case the upper and lower edges of the mesh are forced to be equal during the analysis. This may be useful for cases when small geometries are repeated over and over (i.e., optics examples).

> **Note**
> For more information about using the periodic boundary with *Plane Wave* excitations, refer to *Creating a New Simulation* (using).

Saving Output Data with Sensors

In this section, you will learn how to:

- Use sensors to save the results of your EMPro project calculation
- Choose the correct sensor to use depending on the type of data you want to save
- Mesh voxel objects

Sensors are objects that save data during a simulation. Any type of data that can be saved in EMPro is saved with a sensor. The type of data that is saved by a sensor is dependent on the sensor type, as well as the specific data that is requested within various sensor editors. There are various types of sensors that are available within EMPro, including:

- **Port** sensors
- **Near Field** sensors, including:
  - **Point** sensors
  - **Surface** sensors
  - **Rectangular** sensors
  - **Planar** sensors
  - **Solid Part** sensors
  - **Solid Box** sensors
- **Far Zone** sensors
- **Specific Absorption Rate** (SAR) sensors
- **Hearing Aid Compatibility** (HAC) sensors

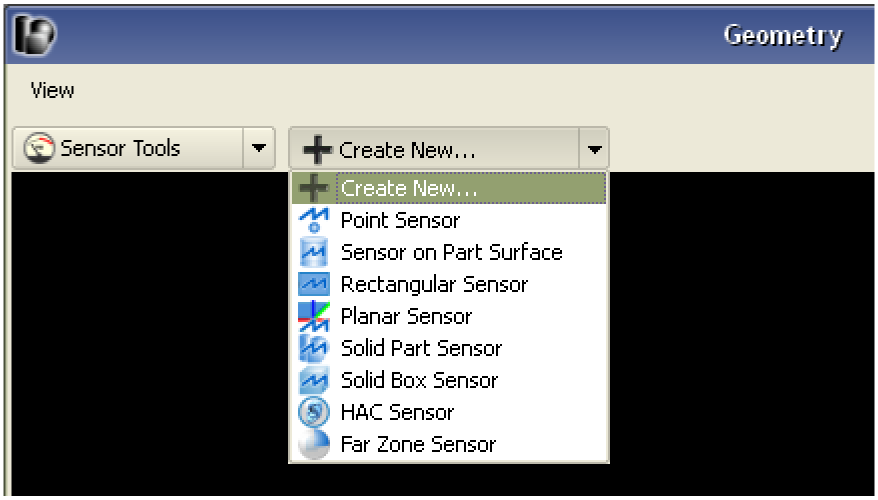
Result objects are generated based on the sensor objects that are defined in the project. After a sensor has been placed, an editor is used to define its characteristics based on the output data. Each type of sensor has its own respective editor window. This section details the process of adding sensors into an EMPro project and requesting specific results with each type of sensor.

Sensor Tools

In general, sensors added to the project within the **Sensor Tools** dialog (with the exception of *Port* and SAR sensors). There are two ways to open this dialog. The first is to choose **Sensor Tools** from the drop-down list in the **Geometry** workspace window. The second is to right-click on the **Sensors** branch of the **Project Tree** and select the branch that corresponds to the desired sensor type.

> **Note**
> *Port* sensors are added by setting a component property. For more information on how to add a *Port* sensor, refer to Adding a New Component.
>
> SAR sensors are added by double-clicking on the sensor in the *Project Tree*. For more information on SAR sensors, refer to SAR Sensors.

Sensor Tools menu



The following sections detail the process of adding each type of sensor within *Sensor Tools* , defining its associated characteristics, and requesting the desired output data to be calculated by the sensor.

Port Sensors

A *Port* sensor saves near-zone voltage and current data at the location of a circuit component. Port sensors are automatically added to the project when a circuit component is added and the *This Component Is A Port* property box is checked in the *Properties* tab of the *Circuit Component Properties* dialog.

Each Port sensor can have a different source resistance. For more information on specifying the source resistance, refer to Specifying the Source Resistance.

S-Parameter, Group Delay, VSWR and Reflection Coefficient Calculations

When S-parameter computation is enabled in the *Simulations* workspace window, *Port* sensors will also save data used to compute S-parameters, Group Delay, VSWR, and reflection coefficient. When multiple *Active Feeds* are used in the simulation, S-parameters will be computed at each Port sensor with respect to each active feed.

Note
S-parameters at each Port sensor are calculated using the characteristic impedance retrieved from the circuit component definitions of that port and the active feed. For more information on defining S-Parameter calculations with single or multiple ports, refer to S-parameters Simulation Setup.

Near Field Sensors

Near Field sensors are used to save time-domain and/or frequency-domain near zone field quantities at specific points within the bounds of calculation space. In general, field data is retrieved using the *Point*, *Surface*, or *Solid* sensors, and hearing aid field values are recorded using the *Hearing Aid Compatibility* (HAC) sensor. Solid sensor results can be viewed as 2-D plots, and both Solid and Surface sensors can be viewed as 3-D field sequences (excluding Point sensor data).

Field Retrieval

The field quantities of ***X*** -, ***Y*** -, and ***Z*** -directed electric (E) and magnetic (H) fields may be saved at a specific point, across a surface, or throughout a volume with a *Point* sensor, *Surface* sensor, or *Solid* sensor, respectively. Additionally, ***X*** -, ***Y*** -, and ***Z*** -directed current density (J) may be collected with any of these sensors. Current densities are determined by multiplying the conductivity of the material at the specified location by the electric field in the given direction. When a PEC material is present, the current density will be computed by the loop of magnetic fields surrounding that cell edge. Thus, the current density only includes the conduction current. When a near-zone source is used as the input, the total field values are available. With an incident *Plane Wave* input, the scattered and total electric and magnetic fields may be saved in addition to the total current density.

Samplings of near field data may be saved by specifying *Sampling Time Range* in any of the near field sensor definition windows. Near field data will be collected in specific planes of the geometry during the EMPro calculation at every interval specified within the definition. A field file containing the electric and magnetic fields and the current will be created for each timestep specified. For example, setting an entry beginning at timestep 100, ending at timestep 1000, with an increment of 100 will create 10 field files which may be viewed as a movie after the EMPro calculation is performed.

Note
Be aware of the number of field slices to save, as they can store enormous amounts of data. Single field files may contain megabytes of data depending on the number of cells in the specified plane.

Point Sensors

A *Point Sensor* is positioned at a specific point-location in the simulation space, and can be defined by the location of a specific vertex in a part object or by a Cartesian 3-D expression. The sensor records data as it occurs at the specified point in space.

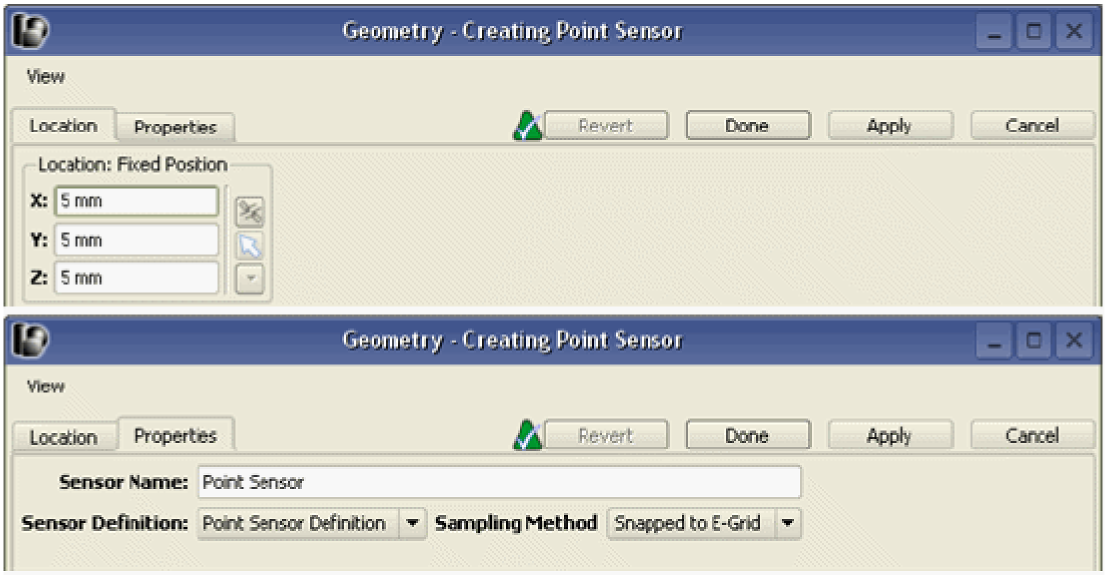
Point sensors record data by means of field interpolation or geometric "snapping". When using the interpolated sampling method, the field components are interpolated to the exact location of the point sensor. This is performed by linear interpolation among the surrounding eight appropriate field value sample points (e.g., when measuring E_x, the eight surrounding ***X*** -directed *edge centers* are used for the interpolation, and when measuring H_x , the eight surrounding ***X*** -directed *cell face center points* are used for the interpolation). When using the snapped sampling method, the location of the point sensor is snapped to the nearest E-grid cell vertex. Field components for snapped point sensors come from the cell whose lowest-index corner is defined by the snapped location of the sensor. The sensor location is thus dependent on the grid definition.

Point Sensor Properties

To define a *Point Sensor*, open the point sensor properties dialog under *Sensor Tools*. In the *Location* tab, define the sensor location manually by typing in its coordinates, or automatically by clicking on the intended location in the simulation space with the

Selection tool. In the *Properties* tab, enter the name of the sensor, select the desired *Point Sensor Definition*, and choose the sampling method as described above.

Point Sensor properties dialog



Point Sensor Definition Editor

The *Point Sensor Definition Editor* window is used to assign definitions associated with a *Point Sensor*.

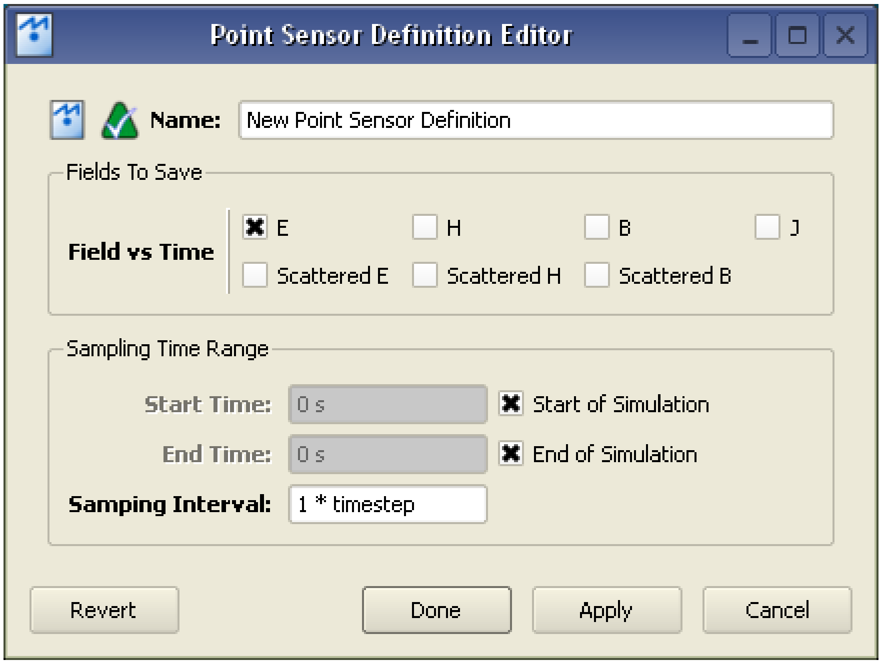
To access the editor, double-click on an existing *Point Sensor Definition* in the *Definitions: Sensor Data Definitions* branch of the *Project Tree*. If no point sensor definition is present, right-click on this branch and select *New Point Sensor Definition*.

In the *Fields to Save* region of the editor, select the desired point sensor output data to save:

- **E:** Electric Field Intensity time
- **H:** Magnetic Field Intensity vs time
- **B:** Magnetic Flux Density vs time
- **J:** Current Density vs time
- **Scattered E:** Scattered Electric Field vs time
- **Scattered H:** Scattered Magnetic Field vs time
- **Scattered B:** Scattered Magnetic Induction Field vs time

> **Note**
> Scattered field values can be retrieved only if a Gaussian beam or a scattered field plane wave external excitation is used to excite the simulation.

Point Sensor Definition Editor



Define the *Sampling Time Range* by entering the *Start Time* and *End Time*, or by simply checking *Start of Simulation* and *End of Simulation* to automatically assign the sampling time range to these values. Choose a *Sampling Interval* to indicate how often data is saved within this time range.

Surface Sensors

Surface sensors collect data on one or more faces of a geometric object in the simulation space. Like *Point Sensors*, they can be interpolated or mesh-snapped.

There are three types of surface sensors in EMPro:

- **Sensor On Part Surface**
- **Rectangular Sensor**
- **Planar Sensor**

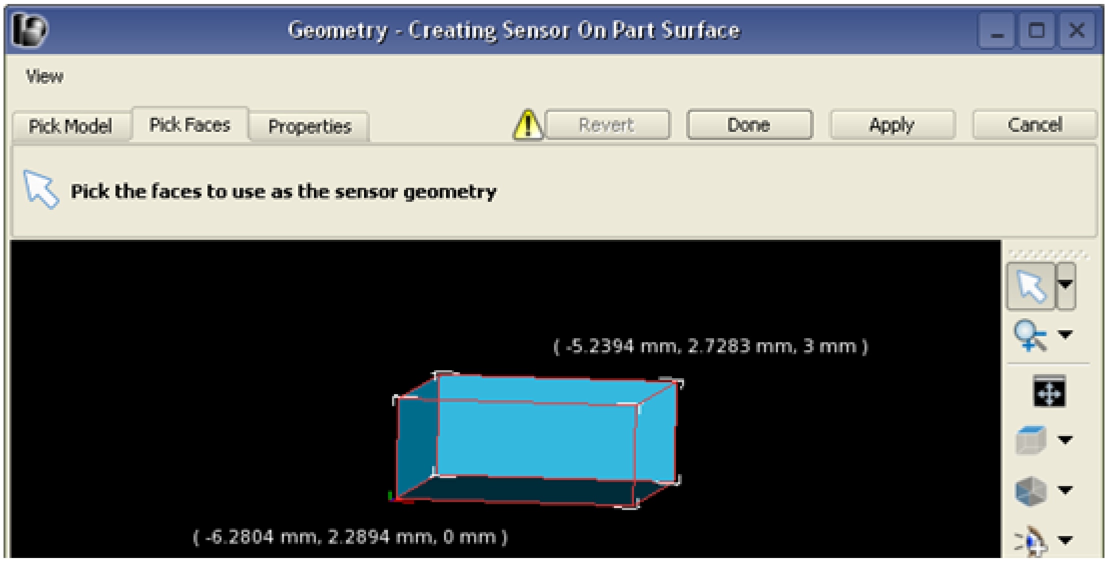
> **Note**
> Refer to the Surface Sensor Definition Editor section to reference the output data that can be retrieved by a surface sensor after it has been created within *Sensor Tools*.

Sensor on Part Surface Properties

To define a *Sensor On Part Surface*, select the object in the simulation space that the sensor will be attached to by clicking on it in the *Pick Model* tab. In the *Pick Faces* tab, select the specific face to attach the surface sensor. Finally, in the *Properties* tab, and assign the new sensor a *Name*, *Definition* and *Sampling Method*.

> **Note**
> *Definition*, as mentioned here and in the following two sensor descriptions, refers to definitions stored in the *Definitions: Sensor Data Definitions* branch of the *Project Tree*.

Sensor on Part Surface properties dialog

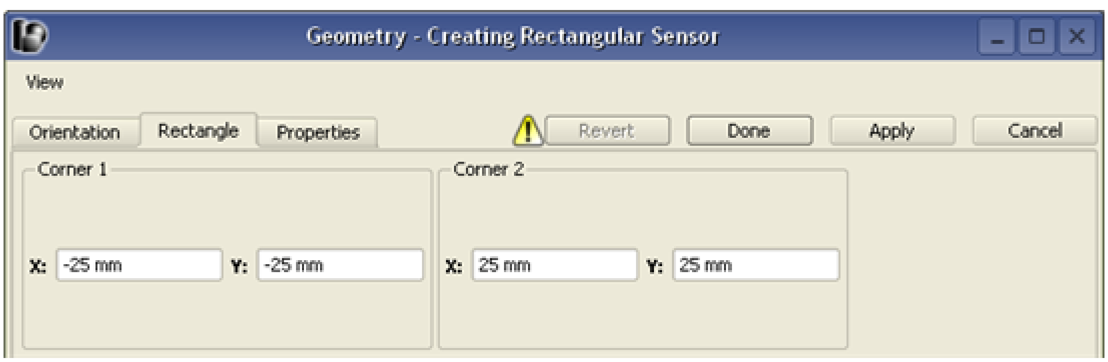


Rectangular Sensor Properties

Define a *Rectangular Sensor* by first using the *Orientation* tab to choose the plane in which the rectangle is defined. Then, use the *Rectangle* tab to define the rectangle's two opposite corners. Finally, under the *Properties* tab, assign the sensor a *Name*, *Definition* and *Sampling Method*.

Note
For an explanation of the *Orientation* tab, refer to the section Specify Orientation Tab.

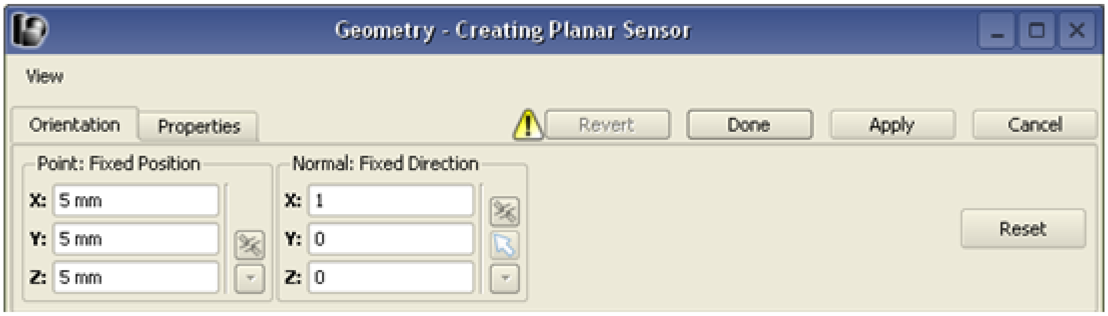
Rectangular Sensor Properties dialog



Planar Sensor Properties

The *Planar Sensor* uses a point and normal direction defined in the *Orientation* tab to define an entire plane (within the boundaries of the simulation space) to collect sensor data. Select the *Properties* tab and assign the sensor a *Name*, *Definition* and *Sampling Method*.

Planar Sensor Properties dialog



Surface Sensor Definition Editor

The *Surface Sensor Definition Editor* window is used to assign definitions associated with a *Surface Sensor*.

To access the editor, double-click on an existing *Surface Sensor Definition* in the *Definitions: Sensor Data Definitions* branch of the *Project Tree*. If no surface sensor definition is present, right-click on this branch and select *New Surface Sensor Definition*.

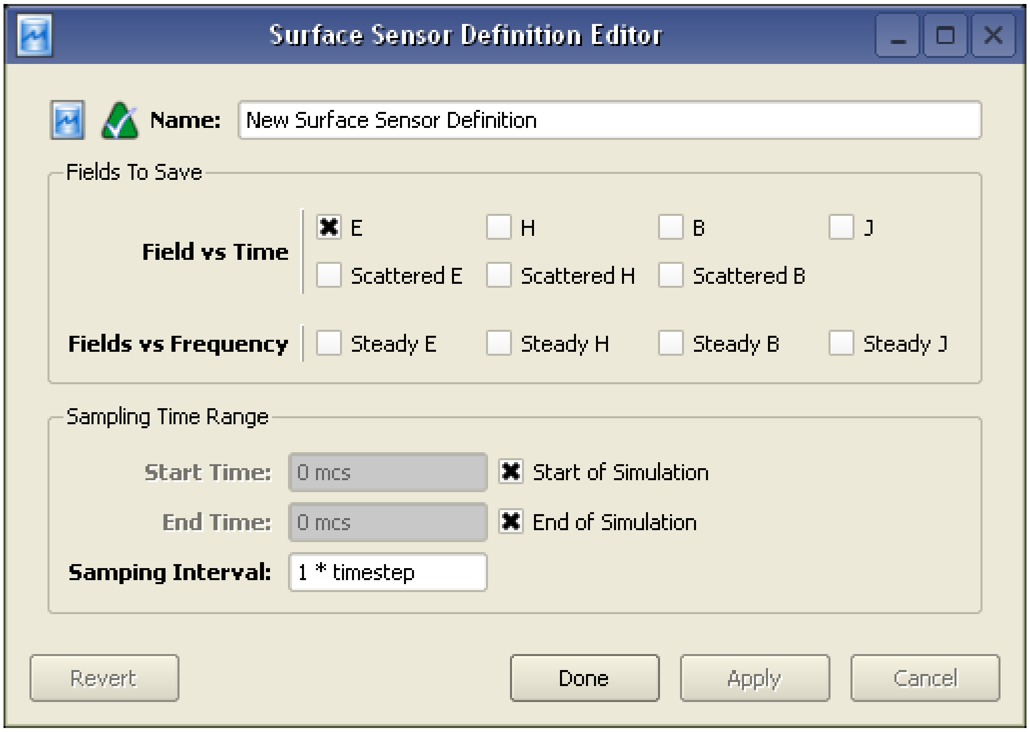
In the *Fields to Save* area of the editor, select the desired surface sensor output data to save:

- **E:** Electric Field Intensity time
- **H:** Magnetic Field Intensity vs time
- **B:** Magnetic Flux Density vs time
- **J:** Current Density vs time
- **Scattered E:** Scattered Electric Field vs time
- **Scattered H:** Scattered Magnetic Field vs time
- **Scattered B:** Scattered Magnetic Induction Field vs time

  Scattered field values can be retrieved only if a Gaussian beam or a scattered field plane wave external excitation is used to excite the simulation.

- **Steady E:** Steady Electric Field vs time
- **Steady H:** Steady Magnetic Field vs time
- **Steady B:** Steady Magnetic Induction Field vs time
- **Steady J:** Steady Current Density Field vs time

Surface Sensor Definition Editor



Define the *Sampling Time Range* by entering the *Start Time* and *End Time*, or by simply checking *Start Of Simulation* and *End Of Simulation* to automatically assign the sampling time range to these values. Choose a *Sampling Interval* to indicate how often data is saved within this time range.

Solid Sensors

Solid sensors collect data by capturing mesh-snapped fields within a volumetric space (interpolated data is not available).

There are two types of solid sensors in EMPro:

- **Solid Part Sensor**
- **Solid Box Sensor**

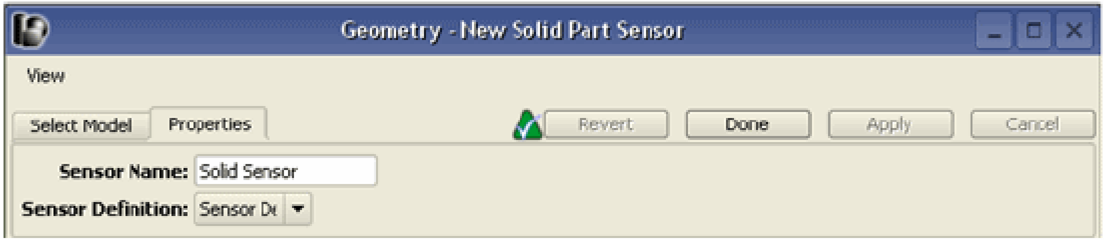
> **Note**
> Refer to Solid Sensor Definition Editor to reference the output data that can be retrieved by a solid sensor after it has been created within *Sensor Tools*.

Solid Part Sensor Properties

A *Solid Part Sensor* simply assumes the shape of the part that is selected in the *Pick Model* tab. Assign the sensor a *Name* and *Definition* in the *Properties Tab*.

> **Note**
> *Definition*, as mentioned here and in the following sensor description, refers to definitions stored in the *Definitions: Sensor Data Definitions* branch of the *Project Tree*.

Solid Part Sensor Properties dialog

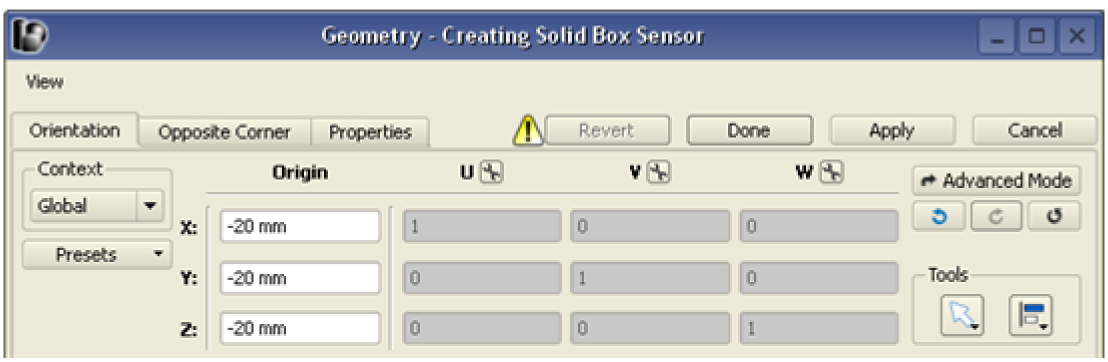


Solid Box Sensor Properties

A *Solid Box Sensor* assumes the shape of a 3-D box. This shape is dictated by the *Origin* location defined in the *Orientation* tab, and its farthest corner is defined in the *Opposite Corner Tab*. Assign the sensor a *Name* and *Definition* in the *Properties Tab*.

> **Note**
> Refer to the section Specify Orientation Tab for an explanation of the *Orientation* tab.

Solid Box Sensor Properties dialog



Solid Sensor Definition Editor

The *Solid Sensor Definition Editor* window is used to assign definitions associated with a *Solid Sensor*.

To access the editor, double-click on an existing *Solid Sensor Definition* in the *Definitions: Sensor Data Definitions* branch of the *Project tree*. If no solid sensor definition is present, right-click on this branch and select *New Solid Sensor Definition*.

In the *Fields to Save* area of the editor, select the desired solid sensor output data to save:

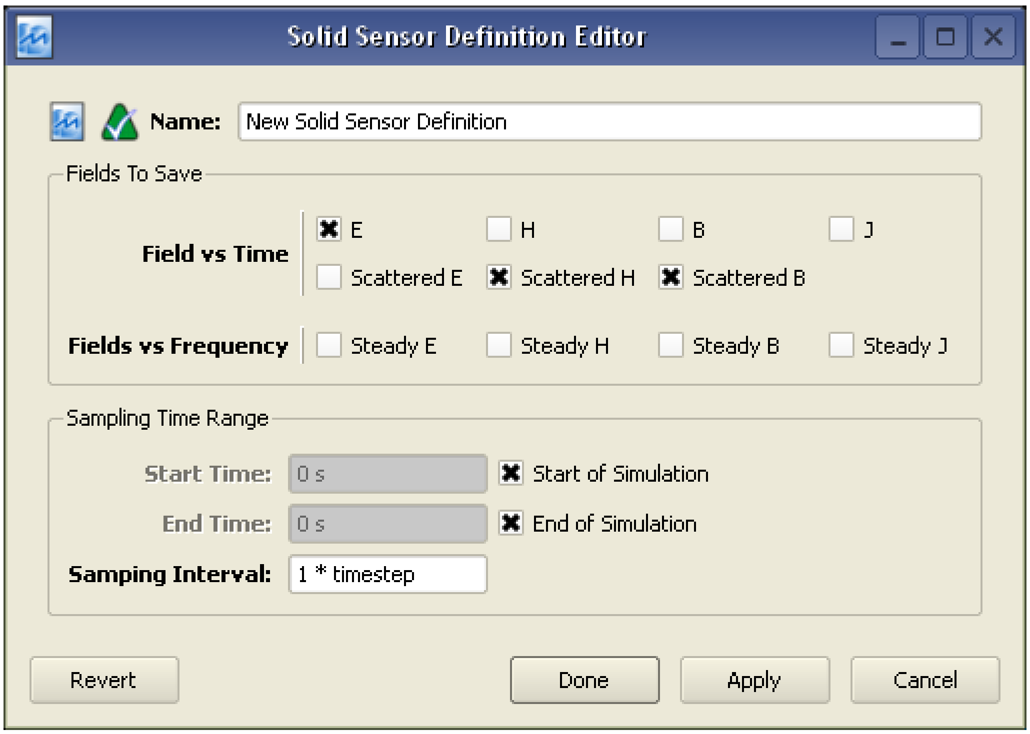
- **E:** Electric Field Intensity time
- **H:** Magnetic Field Intensity vs time
- **B:** Magnetic Flux Density vs time

- **J:** Current Density vs time
- **Scattered E:** Scattered Electric Field vs time
- **Scattered H:** Scattered Magnetic Field vs time
- **Scattered B:** Scattered Magnetic Induction Field vs time

  Scattered field values can be retrieved only if a *Gaussian Beam* or a scattered field *Plane Wave* external excitation is used to excite the simulation.

- **Steady E:** Steady Electric Field vs time
- **Steady H:** Steady Magnetic Field vs time
- **Steady B:** Steady Magnetic Induction Field vs time
- **Steady J:** Steady Current Density Field vs time

Solid Sensor Definition Editor



Define the *Sampling Time Range* by entering the *Start Time* and *End Time*, or by simply checking *Start of Simulation* and *End of Simulation* to automatically assign the sampling time range to these values. Choose a *Sampling Interval* to indicate how often data is saved within this time range.

Hearing Aid Compatibility Sensors

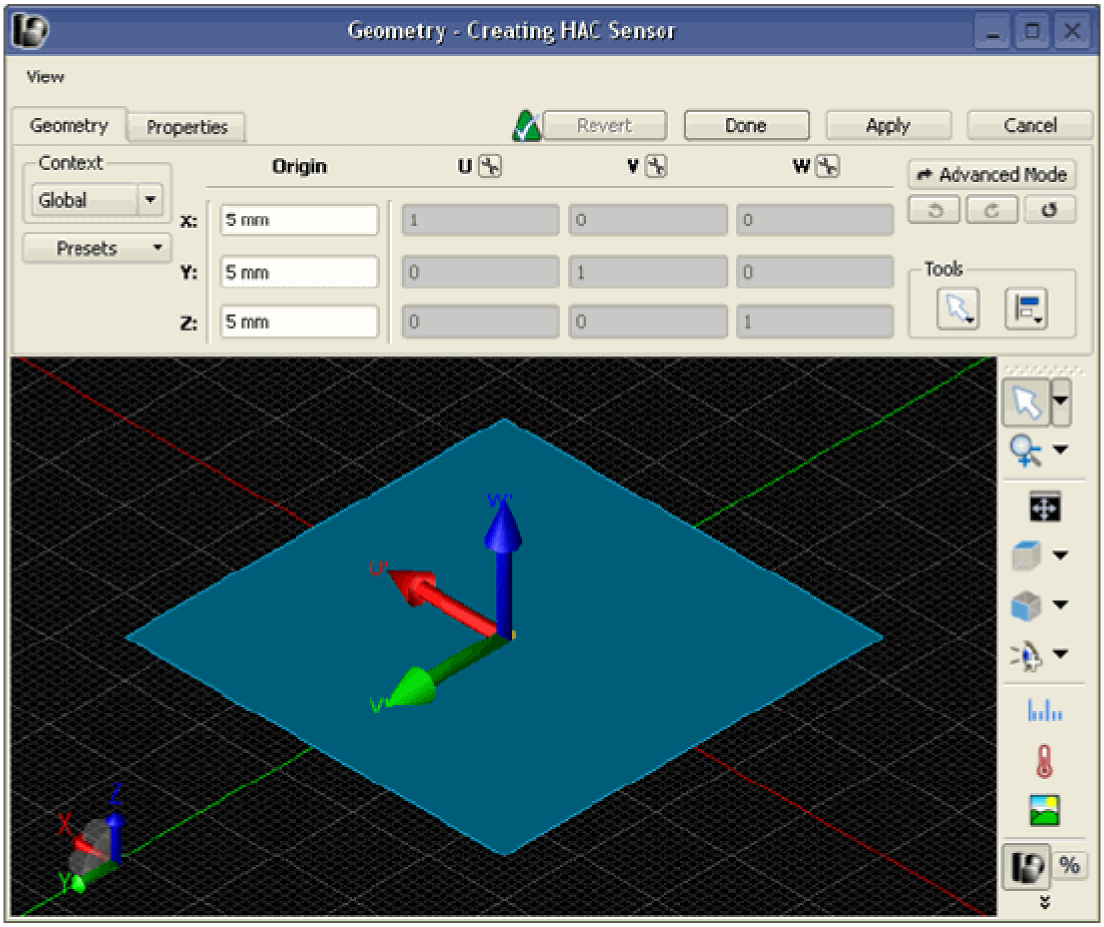
Hearing Aid Compatibility (HAC) sensors gather data on a 5cm by 5cm arbitrarily-oriented rectangle in freespace. They are used to determine if a wireless device (such as a cellphone) will generate electrical and magnetic fields large enough to interfere with a hearing aid. In these cases, they are useful for evaluating the wearer's ability to adjust the position of the phone to a better location.

The HAC sensor is centered at the origin of the coordinate system described in the

Geometry tab.

This sensor collects steady-state E and H fields at grid points near the HAC plane at each frequency of interest. These values can be then interpolated onto the plane at a user-defined spatial resolution.

HAC Sensor properties dialog



Far Zone Sensors

Far Zone Sensors are located at theoretical infinite distance from the simulation geometry. They are only available in the absence of PMC or periodic outer boundary conditions, or when more than one PEC boundary is used.

To create a Far Zone Sensor, under the *Geometry* tab, choose its coordinate system:

- THETA/PHI
- ALPHA/EPSILON
- ELEVATION/AZIMUTH

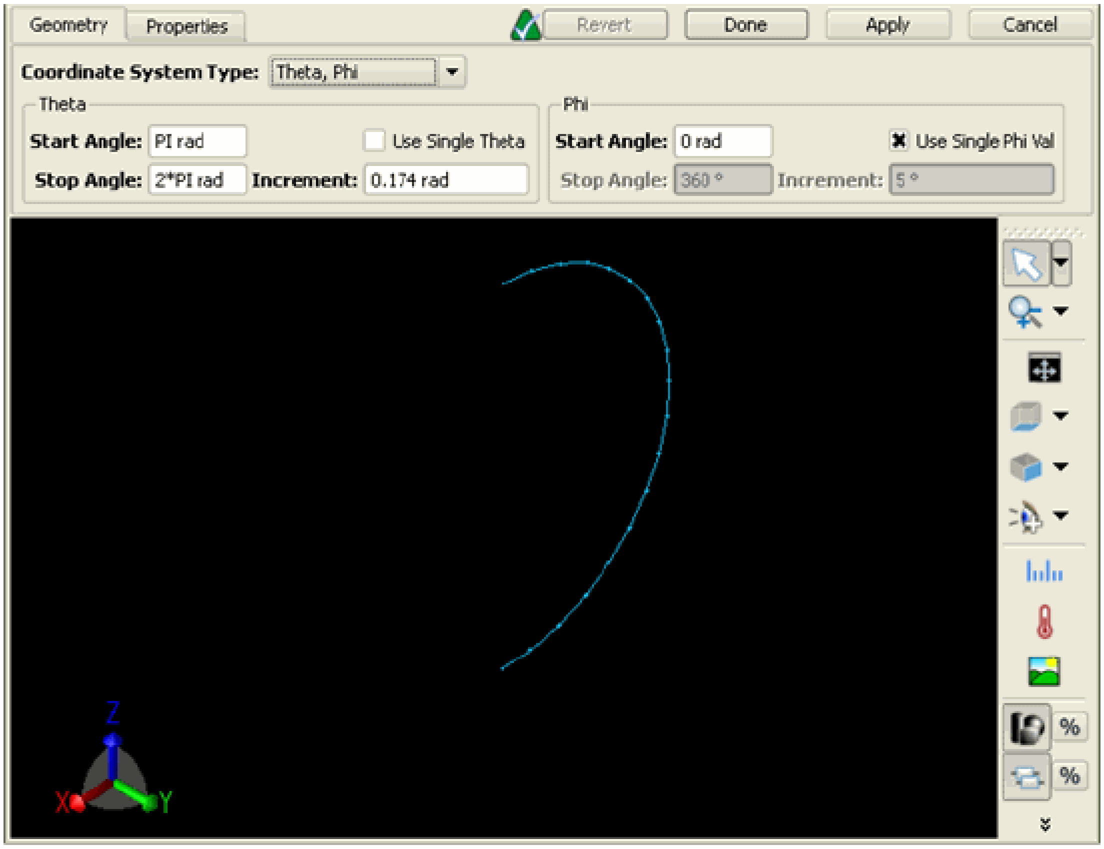
Far zone sensors can be created with one of the following geometries:

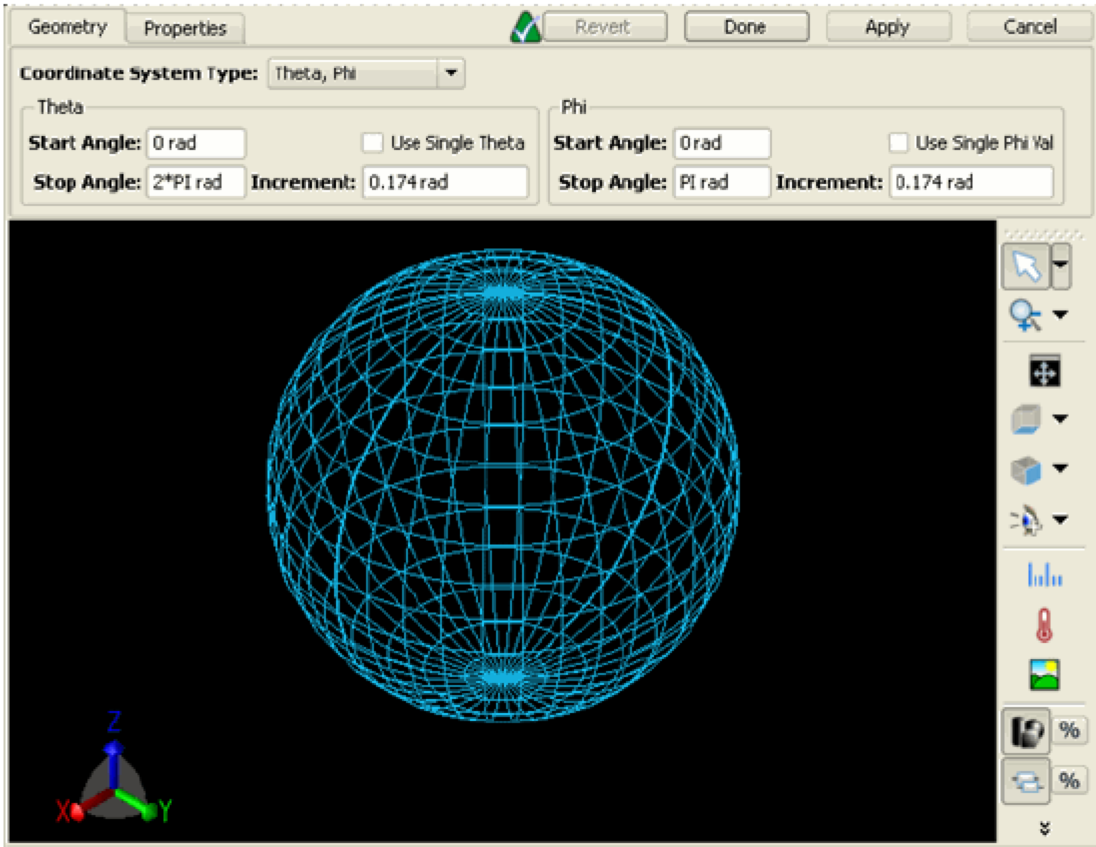
- A range of theta/alpha/elevation over a constant (single) phi/epsilon/azimuth
- A range of theta/alpha/elevation over a range of phi/epsilon/azimuth

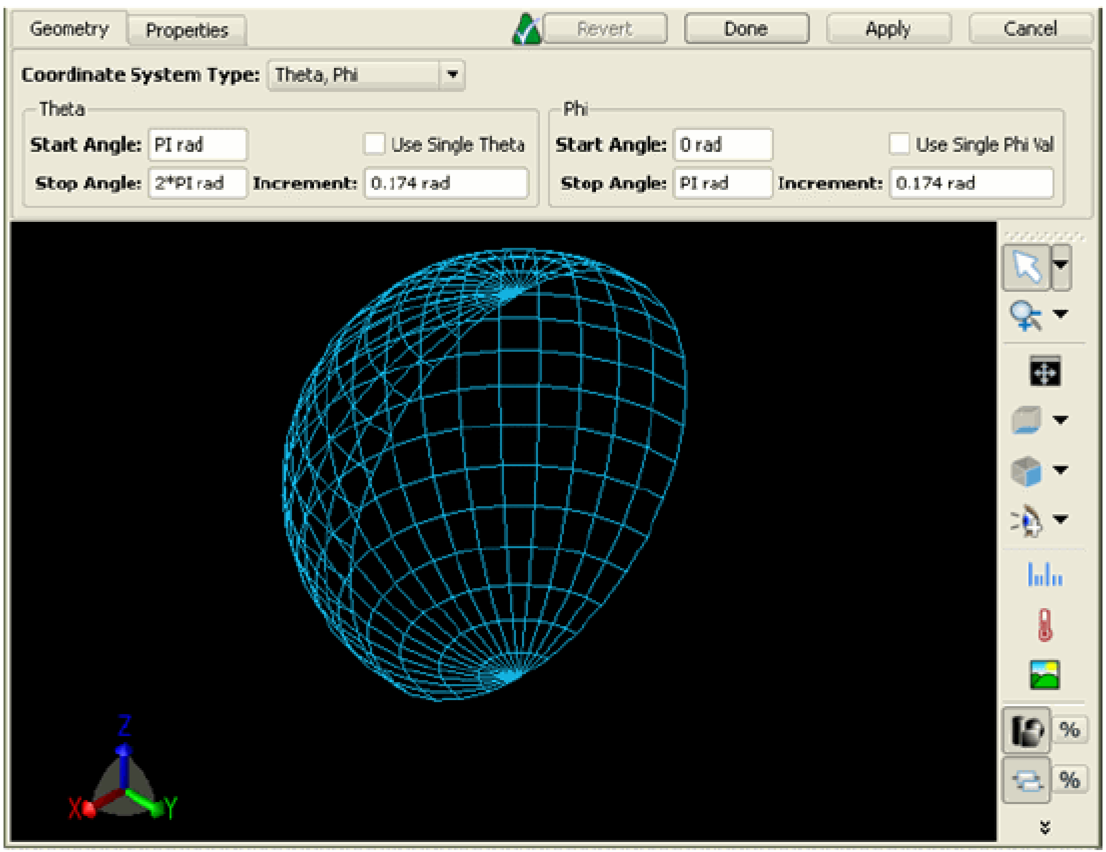
- A range of phi/epsilon/azimuth over a constant (single) theta/alpha/elevation

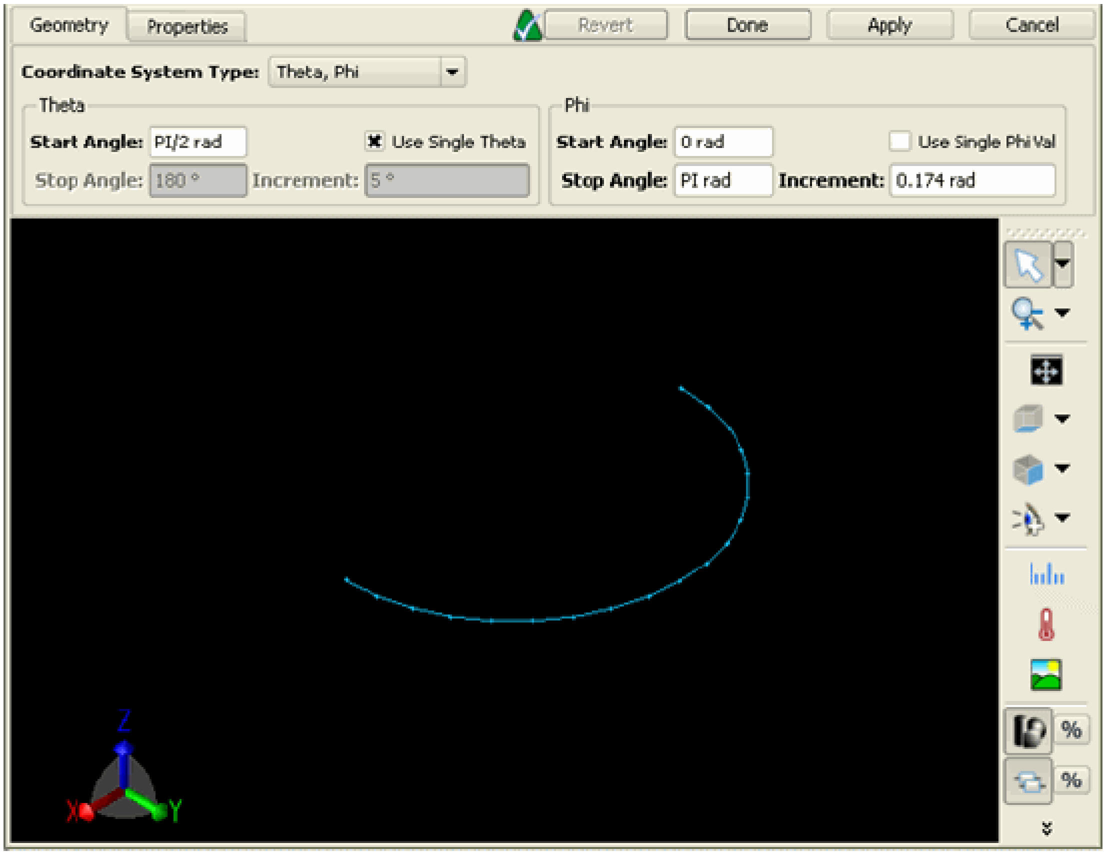
The following figures demonstrate the transformation of the far zone sensor based on the defined geometry in the *Theta/Phi* coordinate system.

Several Far Zone sensor geometries




Geometry
Properties
Revert
Done
Apply
Cancel
Coordinate System Type: Theta, Phi
Theta
Start Angle: 0 rad
Use Single Theta
Stop Angle: 2*PI rad
Increment: 0.174 rad
Phi
Start Angle: 0 rad
Use Single Phi Val
Stop Angle: PI rad
Increment: 0.174 rad
Z
X
Y


Geometry
Properties
Revert
Done
Apply
Cancel
Coordinate System Type: Theta, Phi
Theta
Start Angle: PI rad
Use Single Theta
Stop Angle: 2*PI rad
Increment: 0.174 rad
Phi
Start Angle: 0 rad
Use Single Phi Val
Stop Angle: PI rad
Increment: 0.174 rad
Z
X
Y



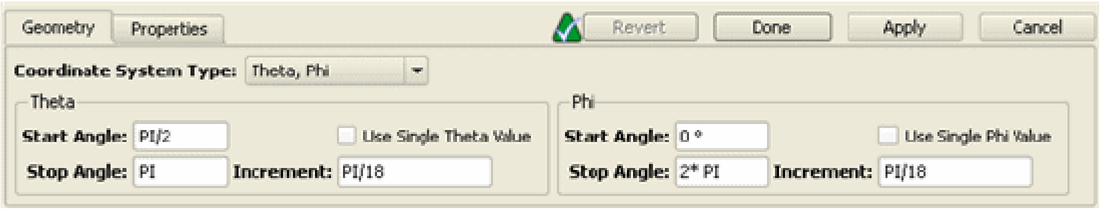
As seen under the *Properties* tab, EMPro has the ability to compute both a steady-state far-zone (near-to-far) and a transient far-zone transform. These two options are described below.

Steady-state Far-zone Transformations

Steady-state transformations are particularly advantageous because the calculation overhead is minimal. They do not require the definition of specific far-zone angles before the FDTD computation, since all patterns are computed in post-processing using data that is automatically stored by the EMPro calculation engine. Instead, the calculation saves the tangential electric and magnetic fields on the far-zone transformation surface at two timesteps, near the end of the calculation when the system should be in steady state. This sampling determines the complex tangential fields on the far-zone surface at the excitation frequency. These fields are then used in post-processing to provide radiation gain or bistatic scattering in any far-zone direction at any pattern increment. This saves considerable computer time and memory if many far-zone directions are required.

Additionally, the selection of a steady-state far-zone transformation computes the single frequency input impedance, total input power, radiated power, and antenna efficiency. All values computed require that the calculation has reached steady state.

Defining a Far Zone sensor



The far-zone gain is displayed in units of dBi. This is the number of decibels of gain of an antenna referenced to the zero dB gain of a free-space isotropic radiator. This value is calculated based on the net power available at the source voltage output. Directivity is not available.

Transient Far-Zone Transformations

The transient far-zone calculation should be used when broadband results are desired, since the steady-state transform is only performed for a single frequency. The broad frequency range, therefore, can be determined at a few points in space. An additional feature of the transient far-zone transformation is that the time-domain far-zone electric fields are also generated and may be plotted.

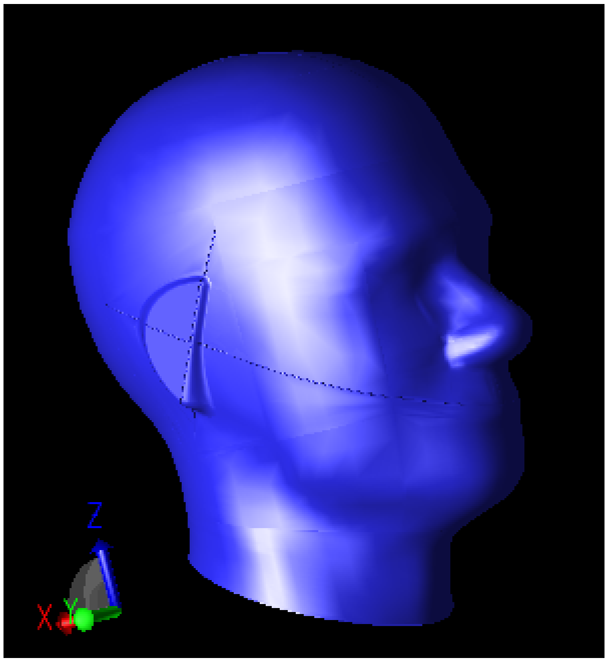
The transient far-zone calculation requires extra calculation time for each far-zone angle specified, and unlike steady-state far-zone transformations, all far-zone angles must be defined before running the calculation engine. The transient far-zone calculation is intended for use in cases where the far-zone results at a few points are desired since it is computationally intensive. This calculation may be desired in instances when far-zone time-domain fields are needed.

> If detailed gain patterns versus angle are necessary, you may reduce calculation time by enabling only steady-state data collection for your sensor and specifying a DFT frequency for your simulation at each frequency you are interested in.

SAR Sensors

EMPro includes several features that fall under the category of biological applications. For compliance with regulations on field absorption in human tissue, the Specific Absorption Rates (SARs) can be computed and averaged. Detailed human body meshes are available for simulations related to effects on realistic heterogeneous models of the body. For some wireless applications, the Specific Anthropomorphic Mannequin (SAM) head is used in addition to the heterogeneous human head (see the figure below).

The Specific Anthropomorphic Mannequin (SAM) head



The Specific Absorption Rate, or SAR, is the unit of measure commonly used to determine the interaction of electromagnetic fields with human tissue. Most regulations involving devices producing electromagnetic fields must not exceed some exposure limits, typically defined in terms of the SAR averaged over a cubical volume of tissue.

> **Note**
> As an example, the IEEE sets exposure levels in terms of 1 g averaging volumes for most of the body, with a 10 g averaging volume applying to extremities such as the ears and fingers.

SAR is defined in terms of the room mean square (RMS) of the electric field magnitude by the relation

$$SAR = \frac{2\sigma|E|}{\rho}$$

Where:

σ is the electrical conductivity in $\frac{S}{m}$, and

ρ is the material density (defined in $\frac{kg}{m^3}$ in EMPro)

Since the FDTD grid defines the electric fields at the edges of the cells, a single SAR value is formed by summing and averaging the contributions of the 12 electric fields on the edges of the cells. The SAR is then referenced to the center of the FDTD cell.

In EMPro, the SAR is measured with the SAR sensor and may only be computed in normal dielectric materials. Frequency-dependent materials have a loss term formed by the imaginary part of the permittivity rather than simply by the conductivity, and are not supported for SAR calculations.

The SAR values are saved only in complete voxels (closed FDTD cells) where all 12 edges of the cell are lossy dielectric material (non-zero conductivity) with a non-zero density, therefore steady-state values for SAR and conduction currents will not exist in all planes. To exclude certain materials from a SAR calculation, simply leave the material density as zero. Saving the SAR in a plane of free-space will not produce any useful output as all values will be zero.

Note
For more information on voxels, refer to the section on Voxels.

Un-averaged SAR Calculation

Un-averaged SAR is measured in EMPro using the *SAR Sensor*. Note that most specifications which involve SAR limits are defined in terms of constant-mass regions, so they will require averaged SAR.

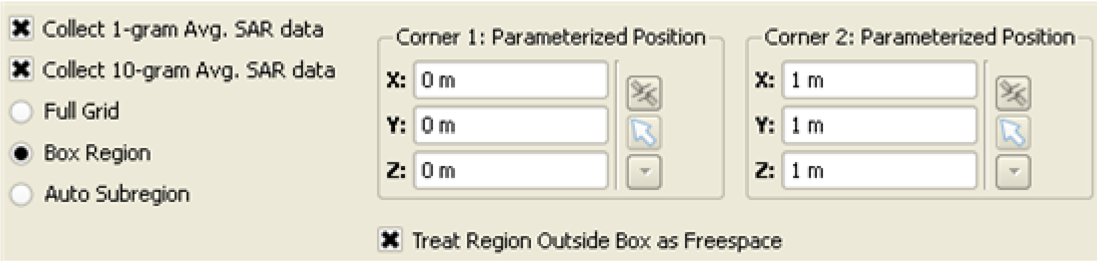
Averaged SAR Calculation

The averaged SAR calculation is more meaningful under most circumstances. This calculation is defined by regulations from organizations such as the IEEE and various government bodies. It is computed over cubical volumes of voxels where no face of the averaging volume is external to the body (and thus full of air or other non-tissue material). In certain cases, particularly at the surface of the body, the cubical volume rule can not be satisfied. In those situations, special rules exist for setting the SAR value in a given voxel. Refer to the IEEE published standards for regulating SAR calculations and setting SAR values in the Bibliography.

Note
Only one SAR averaging region can be defined per calculation run. Additional averaging can be performed as a post-processing step, given that sufficient un-averaged SAR was collected for the region of interest.

Averaged SAR is measured in EMPro using the *SAR Averaging Sensor*. There are several ways to compute average SAR values in EMPro, as shown in the *SAR Averaging* tab. One way is to save 1 gram or 10 gram average SAR regions over the *Full Grid*. During the calculation the averaged SAR values will be computed for all appropriate voxels. This process is time consuming, and since the 10 gram SAR is only applicable to the extremity tissues, it is not necessary to compute it for the entire geometry.

As an alternative to computing values over the entire grid, the EMPro interface also has a tool for computing the average SAR over a *Box Region*. If a subregion of the whole object is defined, then an option is available to allow all data outside of that subregion to be considered as free space. The figure below displays the *Box Region* dialog.



Another alternative is to select the *Auto Subregion* option. In this case, the *Max/Min SAR Ratio* is defined in decibels so that the requested 1 gram or 10 gram average is performed only where applicable, thus saving a great deal of calculation time. This quantity must be entered as a unitless ratio (amplitude) or in dBp (a decibel unit with suffix to indicate an absolute unit of electric power). For example, in a typical application, the extremity tissues would be identified by different material types from the body tissues, so indicating this value in the *Max/Min SAR Ratio* would isolate the calculation to that specific region. The figure below displays this dialog.

Requesting averaged SAR statistics in an automatic subregion

EMPro also offers tissue selection control under the *Tissue Materials* tab. You can compute averaged SAR for *All Tissue Materials*, or for *Selected Extremity Tissue Materials*. In choosing the latter option, a dialog box will appear with a list of available pre-defined materials to include in the calculation.

Running Calculations

In this section, you will learn:

- How to create and run a simulation
- How to start the EMPro calculation engine
- The main factors to consider before beginning a calculation

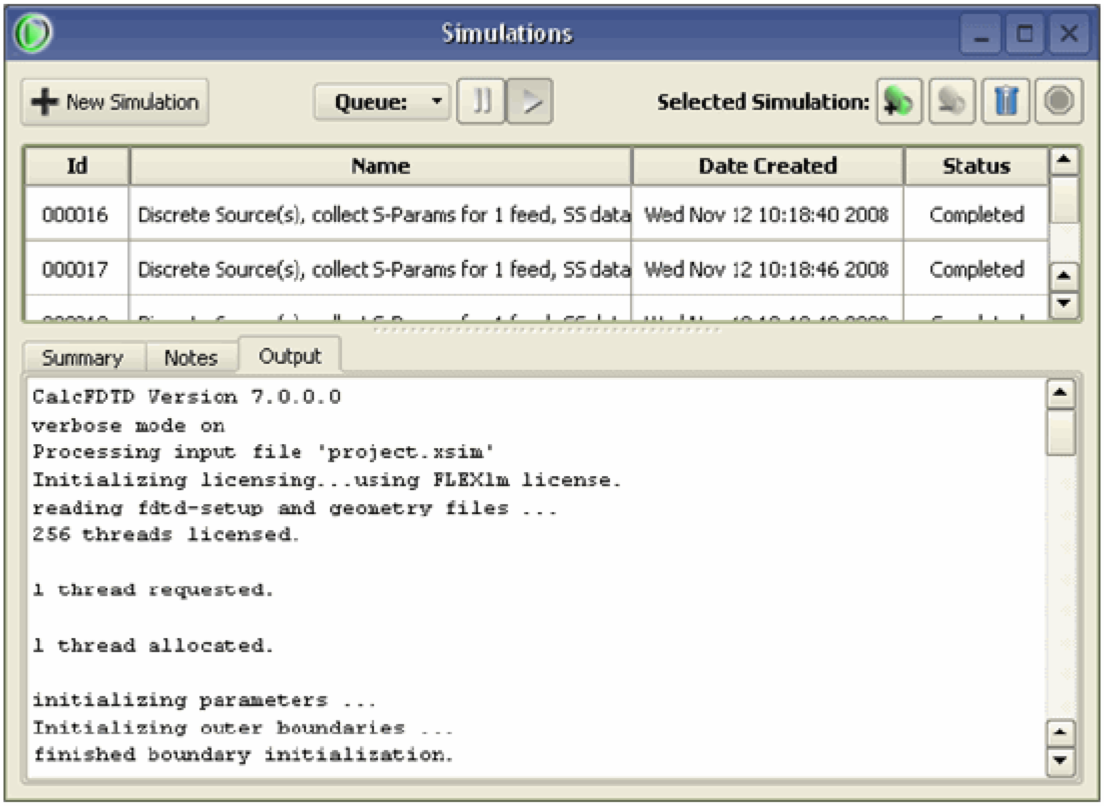
After the EMPro project setup is complete, it is time to run calculations on the geometry. The *Simulations* workspace window stores the project simulation(s). From this window, the user creates, queues and runs the simulations.

The actual electromagnetic calculations are not made by the EMPro GUI. Rather, the electromagnetic calculations are run by a separate program called EMPro FDTD once the finished project file has been saved. Usually this process is run from the EMPro GUI, which calls EMPro FDTD as needed. However, the user may also run EMPro FDTD directly from the command line, or from a remote computer. Once the simulations have been run and the desired calculations have completed, the results can be viewed within the EMPro GUI.

Simulations Workspace Window

The *Simulations* workspace window provides the interface to queue projects to be run with EMPro FDTD, the calculation engine.

The Simulations workspace window



The *Simulations* workspace window lists the name of every simulation that has been created for the project. Its *Status* column shows whether the simulation has been created, queued, or completed.

This window also provides the user with the ability to choose how to run the simulation on their computer. Under the *Queue* drop-down list, specify whether to run the simulation on the CPU or on the hardware card with Acceleware hardware acceleration.

New simulations are created by pressing the *New Simulation* button in the upper-left corner of this window. The associated definitions are described below.

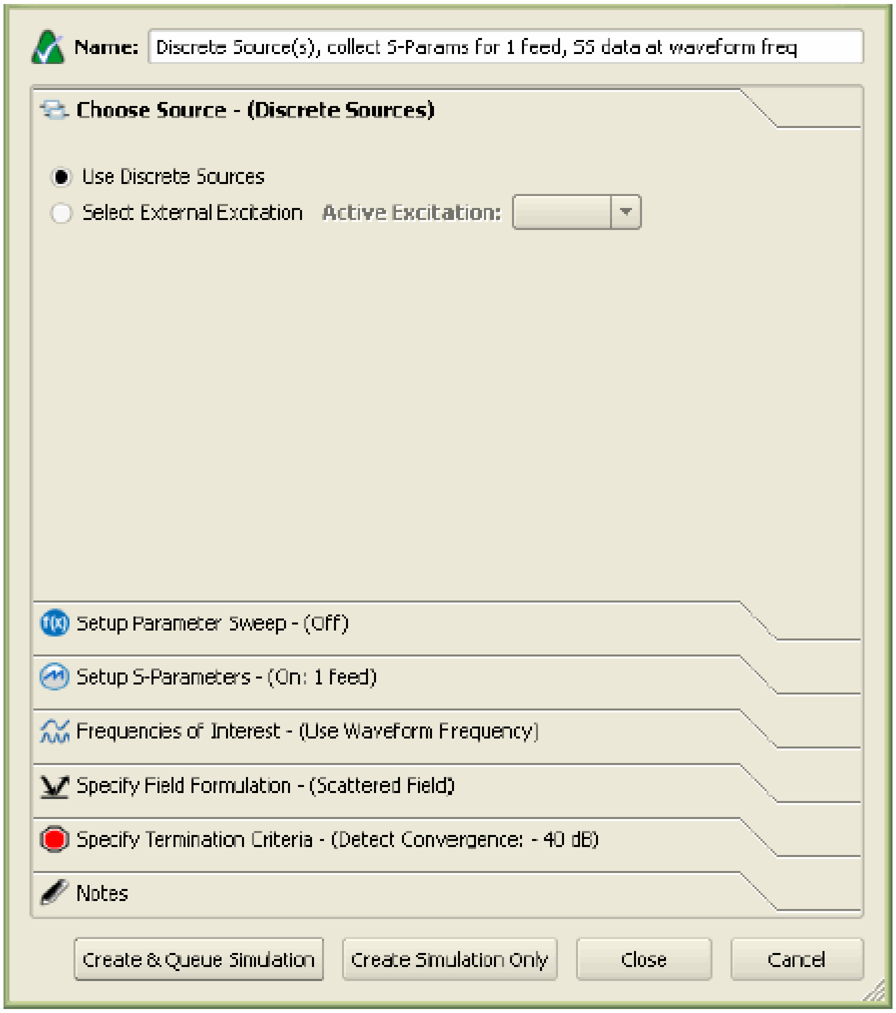
Creating a New Simulation

Choosing a Source

There are two main options for choosing a source in in EMPro, as seen in the figure below. When a voltage or current source is used as an input, check *Use Discrete Sources*. Alternatively, the user may also *Select External Excitation* to use as the source. *Use No Sources* should be selected in special cases with the Static Solver. Discrete sources and external excitation sources are briefly described below.

> **Note**
> For more information on discrete sources, refer to Circuit Component Definition Editor. For more information on external excitations, refer to External Excitation Editor.

Choosing a source for a simulation



Discrete Source

A discrete source is a cell edge on which the electric field is modified by the addition of some type of input waveform. The cell edge can be modified to behave like a voltage or current source. All calculations with discrete source input are performed in total field. Antenna and microwave circuit computations are examples of calculations that may be performed using discrete sources.

Plane Wave

An incident *Plane Wave* source is assumed to be infinitely far away so that the constant field surfaces are planar and normal to the direction of propagation. All calculations with this plane wave source are performed in scattered-field. Total field values may be saved and are typically more desirable than scattered-field plane wave. Calculations of radar cross section or scattering may be performed using this input.

> **Note**
> For important considerations when choosing between scattered and total-field plane wave sources, refer to Plane Wave Excitations.

Gaussian Beam

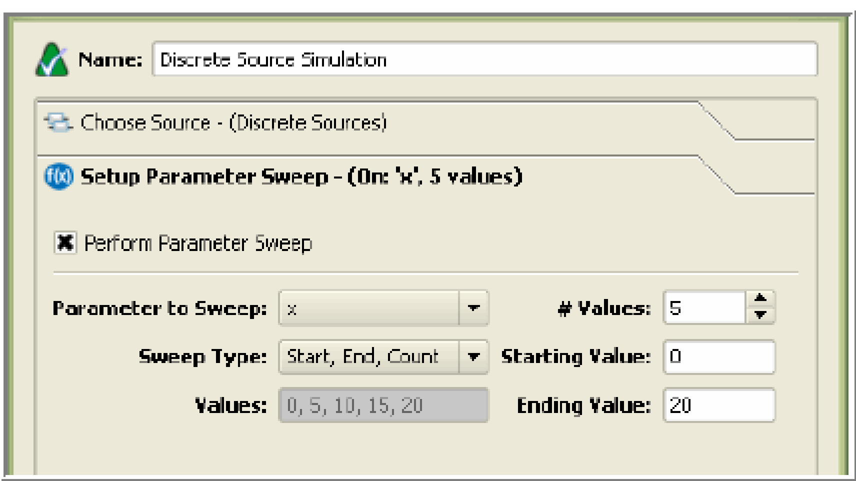
This choice allows for a focused Gaussian beam source in which the incident electric field has a two-dimensional, radially-symmetric Gaussian distribution in planes normal to the incident direction and converges to maximum intensity at the focus point. As with the plane wave source, all calculations with a Gaussian beam source are performed in scattered field, though total field values may also be saved and displayed also. Unlike the plane wave and discrete sources, the Gaussian beam source requires that the source waveform be sinusoidal. Examples where this type of source is useful include structures used at optical frequencies and situations where it is desired to illuminate only a portion of the geometry.

Parameter Sweep Setup

A parameter sweep can be set up so that a script will loop through a particular parameter multiple times and run a calculation at each iteration. For example, if the parameter is antenna length, a script can loop through various antenna lengths, and run a calculation at each length. The *Sweep Type* is defined by the values it will collect (*Start, Incr., Count* values, or *Start, End, Count* values), in order to generate the *Ending Value* or *Increment* values, respectively. A third option is to select *Explicit Values* for the parameter sweep, which do not necessarily have to be at evenly spaced intervals.

> **Note**
> For more information on scripting and to see a sample script for a simple parameter sweep, refer to Scripting Workspace Window.

Setting up a parameter sweep



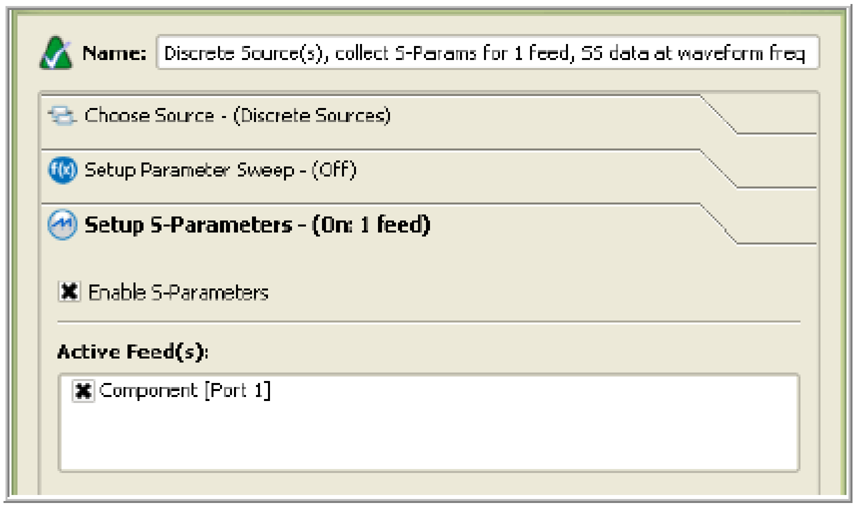
S-parameters Simulation Setup

To calculate a S-parameter for a project with multiple ports defined, the simulation will consider one port at a time to be the active port. The computation with that source active will provide data for a column of the S-parameter matrix. For example, if port one is excited in a three-port circuit, EMPro will determine $\mathbf{S_{11}}$, $\mathbf{S_{21}}$ and $\mathbf{S_{31}}$. If different ports are to be excited, a separate calculation must be performed with each port active. For example, if the full S-parameter matrix for a two-port problem is desired, two calculations must be performed with a different port active in each. EMPro will save the S-parameters for each run in separate files, differentiated by the active port number.

> **Note**
> For more information about the data saved with port sensors, refer to Sensor Tools.

If a parameter sweep is specified in the *Setup Parameter Sweep* tab for multiple parameters in addition to specifying multiple ports within the *Setup S-Parameters* tab, the parameter sweep will be performed for each individual port.

Setting up S-parameters



Frequencies of Interest

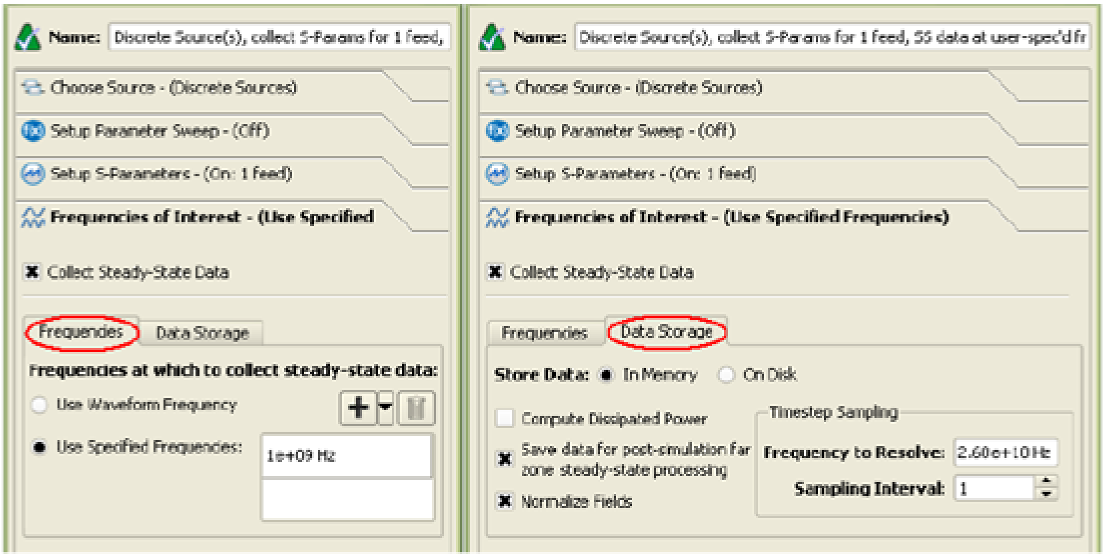
Within this section, the *Frequencies* tab specifies whether the simulation is a broadband (transient) or steady-state calculation. For broadband calculations, uncheck the *Collect Steady-State Data* box at the top of this tab. For steady-state calculations, check this box and choose whether the calculation is to only *Use Waveform Frequency* or to *Use Specified Frequencies* of interest. By specifying more than one frequency of interest, the calculation engine will essentially run a separate calculation at each discrete frequency by running DFT, saving each as its own run. This will therefore increase the calculation time in comparison to using only the waveform frequency.

In the *Data Storage* tab, the user has the ability to specify whether to save temporary data *In Memory* or *On Disk*. Saving the data in memory will speed up the calculation because there is no file saving or loading from disk, but it increases the memory requirements.

Also use this tab to designate which data to save for steady-state far zone post-processing. For a broadband excitation, the user has several options. Checking *Normalize Fields* will match calculated values to a sinusoidal run. Checking *Compute Dissipated Power* will calculate dissipated power based on electric field and magnetic field samplings. It is recommended to leave this box unchecked unless there is specific interest in dissipated power, because it can increase run time significantly due to sampling data over the entire geometry. *Sampling Interval* specifies how often to sample a data type. A sampling interval of one provides the most accurate results because it reduces the effects of aliasing.

Note
For more information on viewing far zone post-processing results, refer to Post Processing.

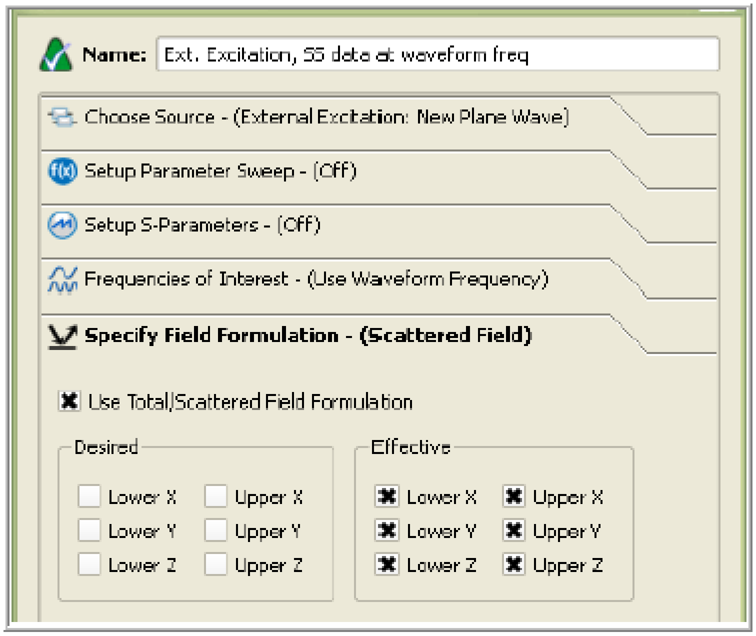
Setting up Frequencies of Interest under the Frequencies and Data Storage tabs



Specifying Field Formulation

The *Specify Total/Scattered Field Interfaces* definitions are only applicable with *Plane Wave* sources and are dependent on boundary condition specifications. They regulate how to perform the calculation in certain regions, specifically where a region of total-field is surrounded by a region of scattered-field. The interface between the two regions must be free space. Scattered fields cannot be sampled inside the total-field region and vice versa. Calculations of radar cross section or scattering are based on fields inside the scattered-field region. For non-periodic boundaries, the six sides of the total-field region are defined as eight cells into the FDTD mesh. For these conditions, there is no option to turn the interface off.

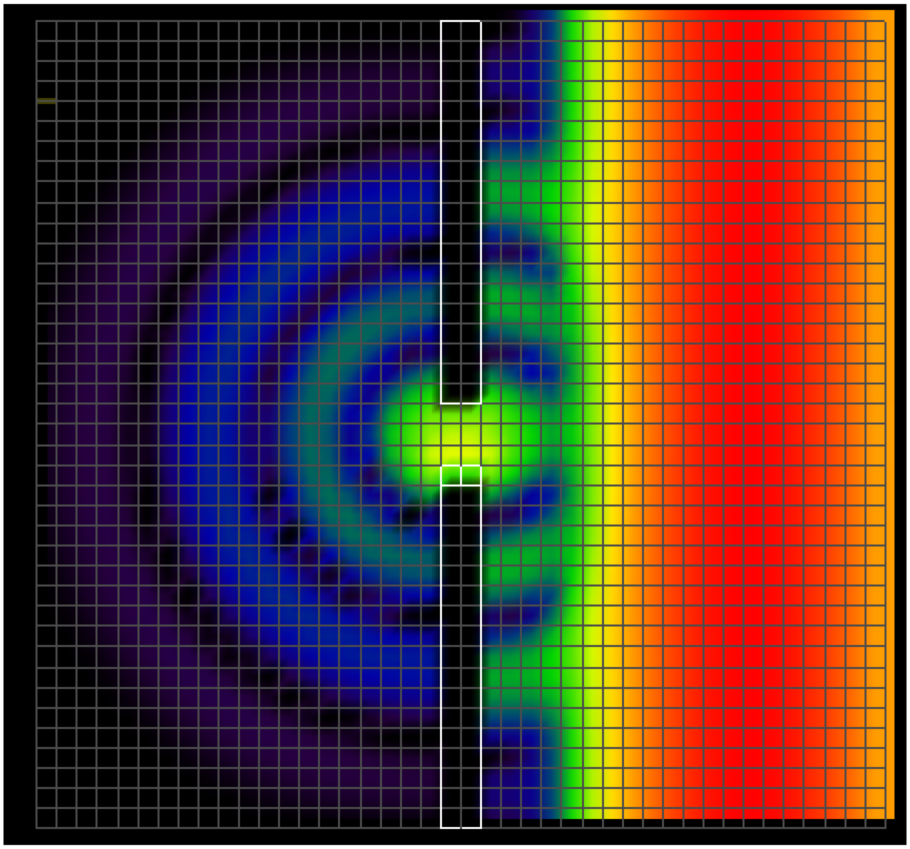
Setting up Total/Scattered Field Formulation



There are two sets of ***X***, ***Y***, and ***Z*** boundaries listed within this dialog. The checkboxes listed in the *Desired* dialog are available for users to indicate which interfaces will be turned on. Depending on the boundary conditions, however, the selected *Desired* definitions may not be applicable or may have to be applied in conjunction with other definitions. Thus, the *Effective* dialog displays the actual definitions that will be applied during the calculation.

When *Periodic* boundaries are specified, certain sides for the interface may be turned off and the total-field region may extend to the boundary using this definition. Periodic boundaries may be useful for applications such as optics where small geometries are repeated over and over again. The figure below illustrates an example in which the outer boundaries have been set to *Periodic* in the the ***Y*** and ***Z*** directions, and the total/scattered field interface has been turned off (unchecked) in the lower ***X*** , upper and lower ***Y*** , and upper and lower ***Z*** directions.

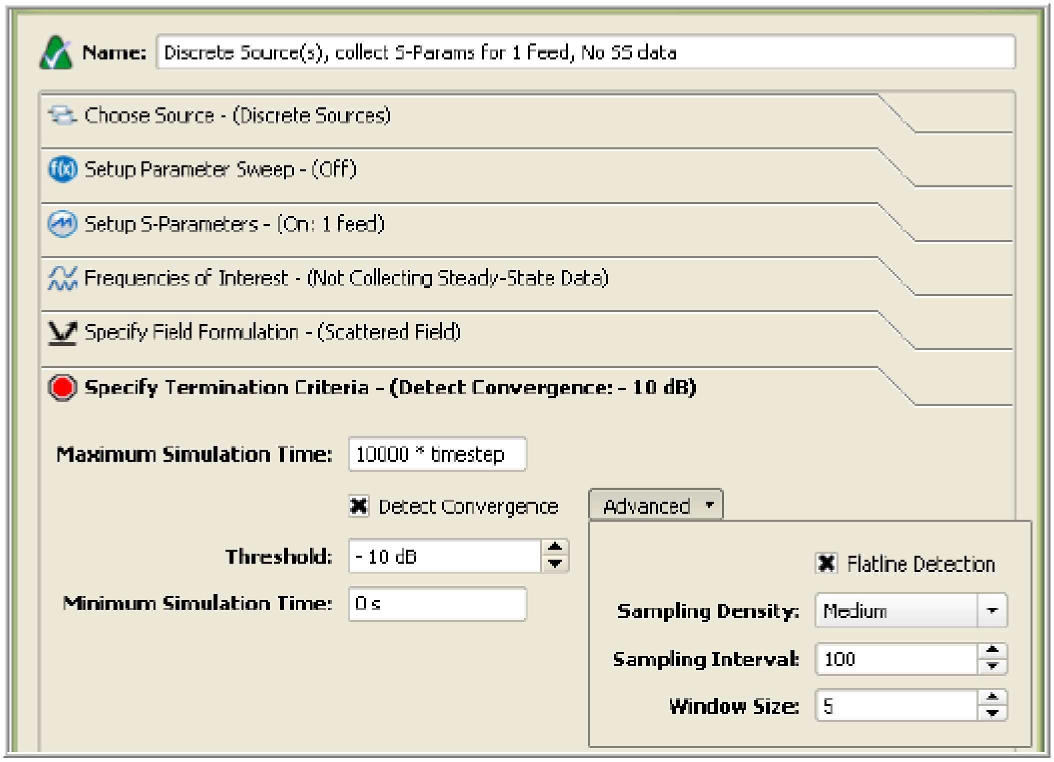
The interface may be turned off for problems that use periodic boundary conditions



Specifying Termination Criteria

Convergence and stability are essential in determining whether a calculation will yield usable results. Convergence in a broadband calculation is met when all electromagnetic energy has dissipated to essentially zero. There are several options in EMPro to define termination criteria to ensure proper convergence had been reached.

Selecting termination criteria within the Simulations workspace window



The most basic method of ending a calculation is defining a value in the *Maximum Timesteps* definition. Once the defined number of timesteps has completed, the calculation will stop. It is important to note that the calculation will terminate regardless of whether or not convergence has been met, so setting this definition to a proper value is important. If it is too low, results will be of no use.

Selecting the *Detect Convergence* check-box will automatically stop the calculation if slow convergence is detected, regardless of whether the number of maximum timesteps have completed. Due to numerical noise in the calculation, there may be a trivial amount of electromagnetic energy, even after the calculation has converged. The value defined in the *Threshold* dialog dictates when the calculation has reached an acceptable value to assume convergence.

Note
A general rule of thumb is that the values should have diminished by at least 30 dB or 1/1000th from the peak values.

For sinusoidally-excited problems, typical values for this setting range from -55 dB to -25 dB depending on the level of accuracy versus runtime desired. For instance, if high accuracy S-parameters are the goal, then the convergence threshold should be set to lower than -30 dB. If however, the user wants to view antenna patterns, -30 dB to -35 dB is suitable. The trade-off here is run time for accuracy. In general, for sinusoidally excited problems, using automatic convergence with a threshold of about -35 dB will produce very accurate results and will run in the shortest possible time to reach this level.

If a calculation is finished but convergence has not been reached, the output from most calculations will be meaningless. The only option is to decrease the convergence *Threshold* or increase the number of *Maximum Timesteps* in the calculation and run it again. If a resonance is occurring at a frequency beyond the range of interest, and a broadband input is used, the input waveform can be modified to limit the frequency content and eliminate the resonance. If the resonance is in band, or a sinusoidal input is used, then this is not applicable and more timesteps must be run or a lower convergence threshold must be chosen.

Advanced Options

There are several options available in the *Advanced* button of this window that allow for more specific termination criteria.

Selecting the *Flatline Detection* check-box will stop the calculation if a slow convergence is detected. This may occur if the user sets the convergence threshold to a very low value (e.g. < -50 dB). In this case the calculation may converge but to a level higher than specified.

> **Note**
> To prevent false convergence, "slow" convergence can only be detected once the convergence level has reached at least -40 dB.

For a steady-state calculation, convergence is reached when near-zone data shows a constant amplitude sine wave - when all transients have died down and the only variation left is sinusoidal. In this case "convergence" is tested on the average electric field in the space for its deviation from a pure sine wave. If *Detect Convergence* is turned on, EMPro automatically places points throughout the space for this purpose. It is particularly important to monitor the results inside high permittivity dielectrics since the field propagation in these materials is much slower than in free space.

To ensure that steady-state calculations converge, EMPro will enable you to control the *Sampling Interval* and *Sampling Density* of the sample points. The temporal *Sampling Interval* definition is used to control the interval for which convergence is tested during calculations with broadband (pulse) excitations.

> **Note**
> Setting this value to 100 or 200 timesteps is typical. Setting this value to much less than that increases the computational overhead a small amount.

Sampling Density is used to control how many spatial samples are used to determine convergence. The sample points are equally spaced in all three dimensions of the grid. *Low* density samples every 4th point in each dimension, while *High* density samples every point in each dimension. For very low frequency problems or where the number of timesteps per RF cycle is greater than 200 (e.g., very small cells with a low frequency excitation), this should be set to LOW. For moderate frequencies or where the number of timesteps per RF cycle is less than 200 but greater than 100 this should be set to *Medium*. For high frequency problems where the number of timesteps per RF cycle is less than 100, a *High* setting gives the best accuracy. This setting is for both broadband as well as sinusoidal excitations.

Notes

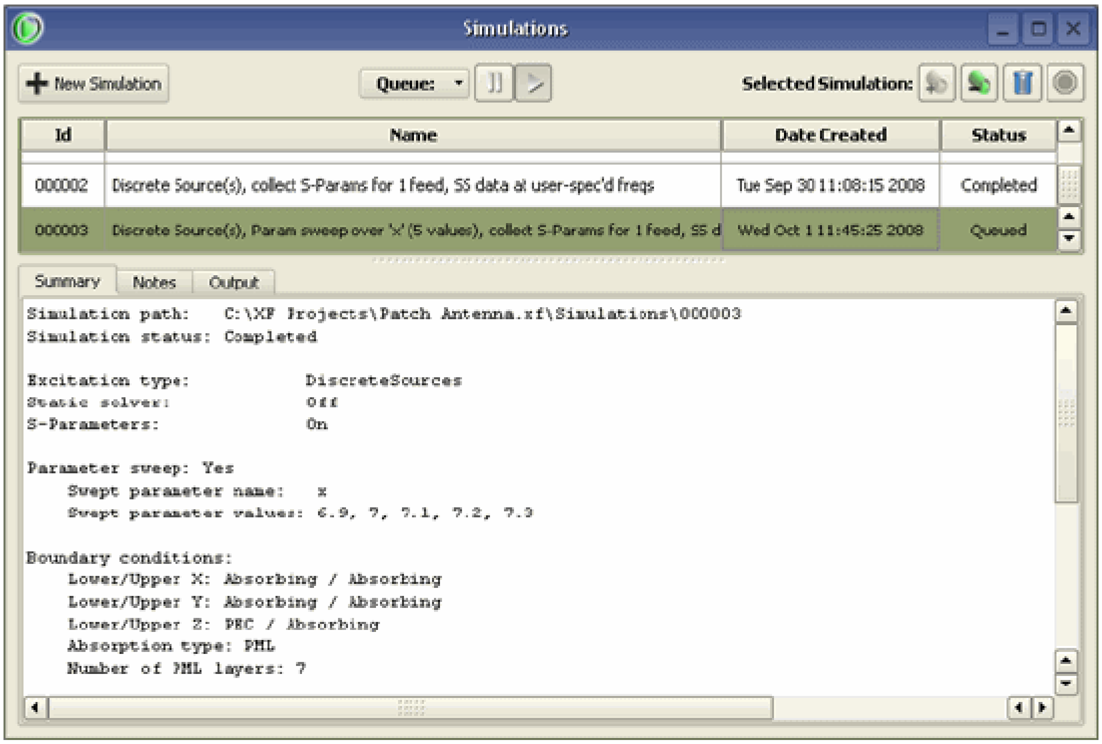
The *Notes* section is simply a tool for users to add any notes to be attached to a project.

The notes will be available in the *Simulations* workspace window after the simulation is created.

Queuing and Running Simulations

After defining the necessary components of the simulation in the *New Simulation* dialog described above, select *Create And Queue Simulation* and the main window will show all of the queued and completed simulations as seen in the figure below. Only one simulation can be run at one time, so as soon as a calculation is complete, another queued simulation will begin.

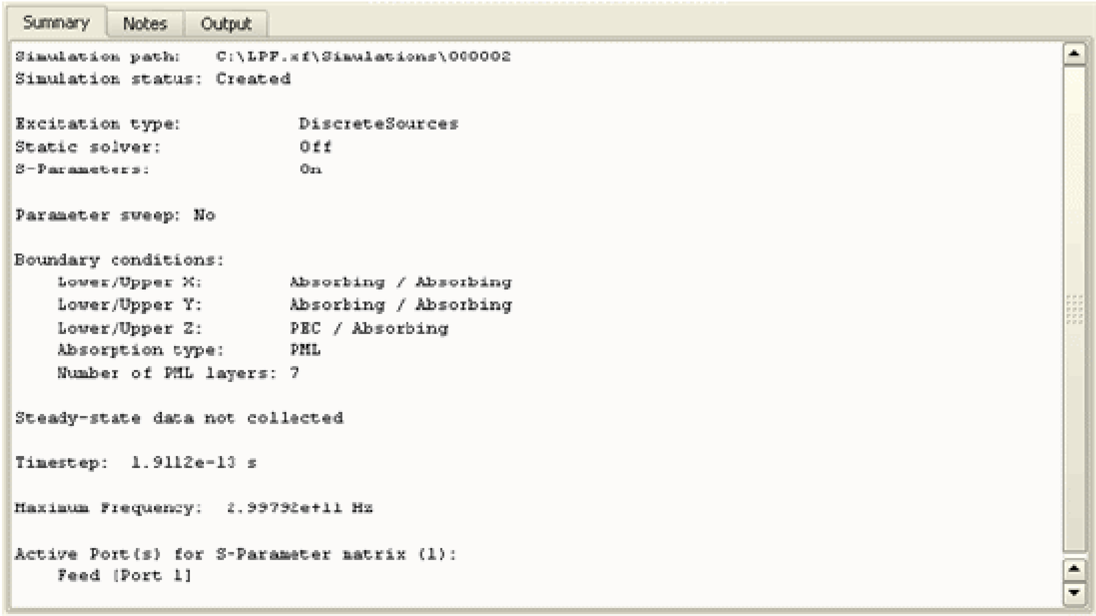
Running a calculation



Below this dialog are three tabs:

- **Summary**, where a basic summary of the calculation is provided, as seen in below.
- **Notes**, which simply documents any notes that were added by the user in the *New Simulation* dialog.
- **Output**, which provides the output generated by EMPrFDTD. Statistics such as the percentage of completion, current timestep, convergence, time elapsed, etc, are listed for every simulation in this tab (including output information for every parameter swept).

The calculation Summary tab



Starting the Calculation Engine

There are two ways to start the EMPrFDTD engine. The first way is to launch it from EMPro in the *Simulations* workspace window. The second way is to start the calculation from a command line. This is preferable for calculations that require a large amount of memory in order to free memory used by the interface.

Running EMPrFDTD Within the EMPro Interface

Calculations launched from EMPro are sent to the EMPrFDTD engine from the *Simulations* workspace window. A list of every simulation that has been created within the *New Simulation* dialog is listed in the main window. When a simulation is ready to run, send it to EMPrFDTD by selecting the simulation and clicking the *Add To Queue* button, located to the right of *Selected Simulation* at the top of the window. Although multiple simulations can be queued at once, only one simulation is run in EMPrFDTD at a time. Once the *Play* button is pressed, each simulation will run one at a time until all of the queued simulations have terminated.

> **Note**
> For more on creating a new simulation, refer to Creating a New Simulation.

Running EMPrFDTD From the Command Line

In some cases, it is useful to run EMPrFDTDfrom a command line. This will prevent EMPro from overloading since it will not be allocating memory for EMPrFDTD in addition to its normal memory requirements. This is not an issue for most calculations, but it may cause problems when the memory required to run a calculation approaches the limitations of computer memory. In this case, closing EMPro and running EMPrFDTD from the command line will free up any memory used by EMPro.

To run EMPrFDTD, navigate to the appropriate project folder and run the command:

Windows: `emprfdtd.exe` [options]

Mac OS X/Linux: `emprfdtd` [options]

For normal operation no [options] need be specified, but the table below lists several options that may be of interest.

Command Line Options

option	**Function**
`v`	Verbose mode, prints progress of calculation.
`proc N`	Sets the number of processors to use for a calculation with the multi-processor module (optional).
`loadstatice`	Loads fields saved by the static voltage solver.
`savestaticeonly`	Saves fields by the static voltage solver and terminates before time-stepping.
`savestatice`	Saves fields by the static voltage solver and continues timestepping.
`hardware`	Uses Acceleware hardware acceleration; in the case that an error occurs initializing Acceleware, the engine uses the software to run calculation.
`hardwareonly`	Uses Acceleware hardware acceleration; in the case that an error occurs initializing Acceleware, the calculation is terminated.
`forcetemporarydatatomemory`	Forces steady-state data to be stored in memory rather than in an external file.
`saronly`	Calculation skips time-stepping and only performs post-processing of SAR data.

Running the Calculation Remotely

Running a remote calculation is an alternative way to potentially speed up calculation time and free local computer resources for other operations. This section will explain how to copy files and launch the calc engine on a remote computer.

> **Note**
> If your EMPro project is stored on a file system which is remotely accessible, you can avoid the process of copying files back and forth from the remote machine. Simply log into the remote machine, navigate to the simulation directory and launch the calc engine.

Running Simulations Remotely

1. **Create the simulation.** After creating the simulation from the *Simulations* workspace window, click the *Create Simulation Only* button to save the simulation to the *Simulations* folder where your EMPro project is stored.

   You can find the path to this directory in the first line of text under the *Summary* tab

   > **Note**
   > For more information about setting up a simulation in EMPro, refer to Simulations Workspace Window.

2. **Copy files to the remote machine.** For simplicity, you can copy the whole

Simulations directory (you'll need to copy it recursively, since it contains subdirectories). However, only a few files in this directory are required by the calculation engine:

project.xsim
Run*/geometry.input
Run*/mesh.input
Run*/project.input

> **Note**
> The * in the file name represents the simulation number, (e.g., Run0001/geometry.input, Run0002/geometry.input, etc).

For Windows users, it may be easiest to zip each file individually and transfer them to the remote machine over an ftp connection.
For Mac OS X users, you can save these files in a compressed archive with the following command:

```
tar -czf inputFilesForCalcEngine.tar.gz project.xsim Run*/geometry.input
Run*/mesh.input Run*/project.input
```

Then copy the *inputFilesForCalcEngine.tar.gz* file to the remote machine and extract it using:

```
tar -xzf inputFilesForCalcEngine.tar.gz
```

3. **Run the simulation** Log in to the remote machine, change directories to the *Simulations* folder, and run EMPrFDTD for the project.

> **Note**
> For more information on running the calc engine from the command line, refer to Running EMPrFDTD From the Command Line.

4. **Copy files back to local machine.** In order to view your results from the EMPro GUI, you'll need to copy all of the *Simulations* files from the remote machine to your local machine, *with the exception of the files listed above*. That is, after running the simulation on the remote machine, all *new* files should be copied back into the *Simulations* folder on your local machine. The `tar` utility makes this easy. On the remote machine, change directories to the *Simulations* directory and run

```
tar -czf outputFilesFromCalcEngine.tar.gz -exclude=project.xsim -
exclude=Run*/geometry.input -exclude=Run*/mesh.input -
exclude=Run*/project.input *
```

For Windows users, zip the *outputFilesFromCalcEngine.tar.gz* file and send it back to your local machine over the ftp connection. You can then extract the files using Winzip. Overwrite any old files with newer versions of the same file.
For Mac OS X users, copy the outputFilesFromCalcEngine.tar.gz file to your local machine, change directories and extract it using:

```
tar -xzf outputFilesFromCalcEngine.tar.gz
```

5. **Refresh results.** To make your results available from the EMPro GUI, click *Refresh* in the *Results* workspace window.

Remote SAR Post-processing

1. **Setting up post-processing.** After requesting post-processing from the RESULTS workspace window, in the subsequent dialog box, tell the application to not run the calculation right away.
2. **Copy raw SAR data, calc engine input files, and SAR request files.** The following files are required for post-processing in the *Simulations* folder on the remote machine:

   project.xsim
   Run*/geometry.input
   Run*/mesh.input
   Run*/project.input
   Run*/SteadyStateOutput/**/SAR_Raw_Sensor.**.sar.gz
   Run*/SteadyStateOutput/*/SAR_Raw_Sensor.stats
   Run*/request.sar

   The * in the file name represents the simulation number, (e.g., Run0001/geometry.input, Run0002/geometry.input, etc).

   For Windows users, it may be easiest to zip each file individually and transfer them to the remote machine over an ftp connection.

   For Mac OS X users, change the directory to *Simulations* and run the following command on your local machine:

   ```
   tar -czf sarRawData.tar.gz project.xsim Run*/geometry.input Run*/mesh.input
   Run*/project.input Run*/SteadyStateOutput//.sar.gz Run*/request.sar
   ```

   Then copy the *sarRawData.tar.gz* file to the remote machine and extract it using:

   ```
   tar -zxf sarRawData.tar.gz
   ```
3. **Run SAR averaging on the remote machine.** On the remote machine, change directories to the *Simulations* directory, and execute the calc engine application using the `-saronly` command line flag.

   > **Note**
   > For more information on running the calc engine from the command line, refer to Running EMPrFDTD From the Command Line.

4. **Copy SAR results back to the local project.** If you've already copied your other results back to the local project, you'll need to package up your SAR results and copy them back. From the remote machine, copy the following files to the *Simulations* directory on the local machine:

   Run*/status
   Run*/SteadyStateOutput/*/*gsar.gz
   Run*/SteadyStateOutput/**/**.infosar1g

Run*/SteadyStateOutput//.infosar10g
Run*/SteadyStateOutput//.infoseq
Run*/SteadyStateOutput//.*gssq
Run*/SteadyStateOutput//.stats

Then run the following command:

```
tar -czf sarResults.tar.gzRun*/status Run*/SteadyStateOutput/*/*gsar.gz
Run*/SteadyStateOutput//.infosar1g
Run*/SteadyStateOutput//.infosar10g Run*/SteadyStateOutput//.infoseq
Run*/SteadyStateOutput//.gssq Run/SteadyStateOutput//.stats
```

> **Note**
> Notice that this command sequence retrieves all of the SAR averaging results, including those done with your initial calc engine run.

For Windows users, zip the *sarResults.tar.gz* file and send it back to your local machine over the ftp connection. You can then extract the files using Winzip. Overwrite any old files with newer versions of the same file.

For Mac OS X users, copy the *sarResults.tar.gz* file to your local machine, change directories and extract it using:

```
tar -xzf sarResults.tar.gz
```

5. **Refresh results.** To make your results available from the EMPro GUI, click **Refresh** in the *Results* workspace window.

Calculation Considerations

The calculation portion of EMPro may be quite lengthy depending on the application. A few guidelines are provided here for estimating computer resources, monitoring the progress of the calculation, and avoiding calculation instability.

> **Note**
> For information about defining proper termination criteria to ensure that the calculation has finished, refer to Creating a New Simulation .

Computer Resources Estimation

EMPrFDTD will give a time estimate while the calculation is running. This is recalculated every time EMPrFDTD updates its status based on how much time passed since the last update and the remaining number of timesteps. It is not a completely accurate estimation since it does not consider data such as near-field samplings which may only be saved during certain portions of the calculation. Also, the estimate does not include any post-processing which may occur. Of special note are the SAR averages since they are computed in post-processing and may require a significant amount of calculation time.

Note
A quick way to estimate the amount of memory that EMPrFDTD will need for a given problem is to multiply the number of cells in the geometry by 27. If magnetic materials are included in the geometry (any material with non-free space permeability), multiply by 30, rather than 27. The resulting number is the approximate number of bytes needed to calculate the project.

Multi-Processing Modules

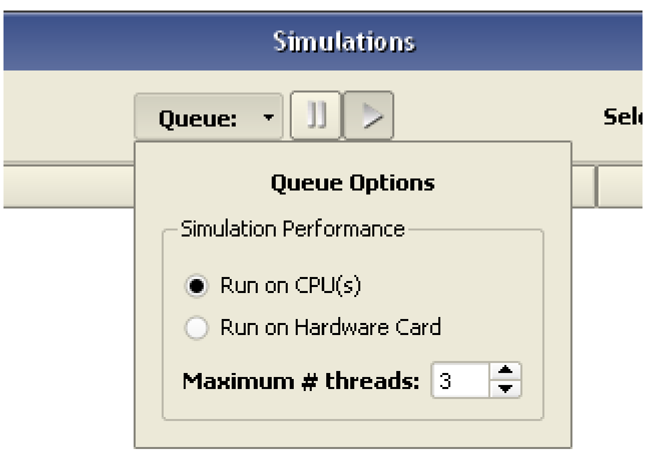
EMPrFDTD has the ability to do both threaded and Message Passing Interface (MPI) calculations. Both of these capabilities are optional features which may be added to the calculation engine.

Multi-threaded calculations use shared memory and are intended for computers with multiple processors and/or cores on a single motherboard. The overhead for the multi-threading routines can cause very small calculations to run slower when more than one thread is selected.

Note
In general, the multi-threaded option should not be used when the number of FDTD cells in the geometry is less than one million.

The number of cells is computed simply from the ***X*** , ***Y*** , and ***Z*** dimensions (in cells) of the geometry space. The number of threads used for a calculation is defined in the *Queue* drop-down box of the *Simulations* workspace window, as seen below.

Specifying calculation engine threads



Specify the number of processors that are to be used for the calculation under *Maximum # Of Threads*.

To specify the number of threads from the command line, the "`-proc N`" option should be used, where `N` is the number of threads.

Note
For a summary of command-line options, refer to Running EMPrFDTD From the Command Line.

MPI calculations can be executed on cluster of computers that are connected by a network. At present, only Linux computers are supported and each computer must have the same MPI libraries installed. To run an MPI calculation, see the documentation for the MPI library that is installed on all machines in the cluster. This documentation will give instructions on how to start an application using the MPI tools.

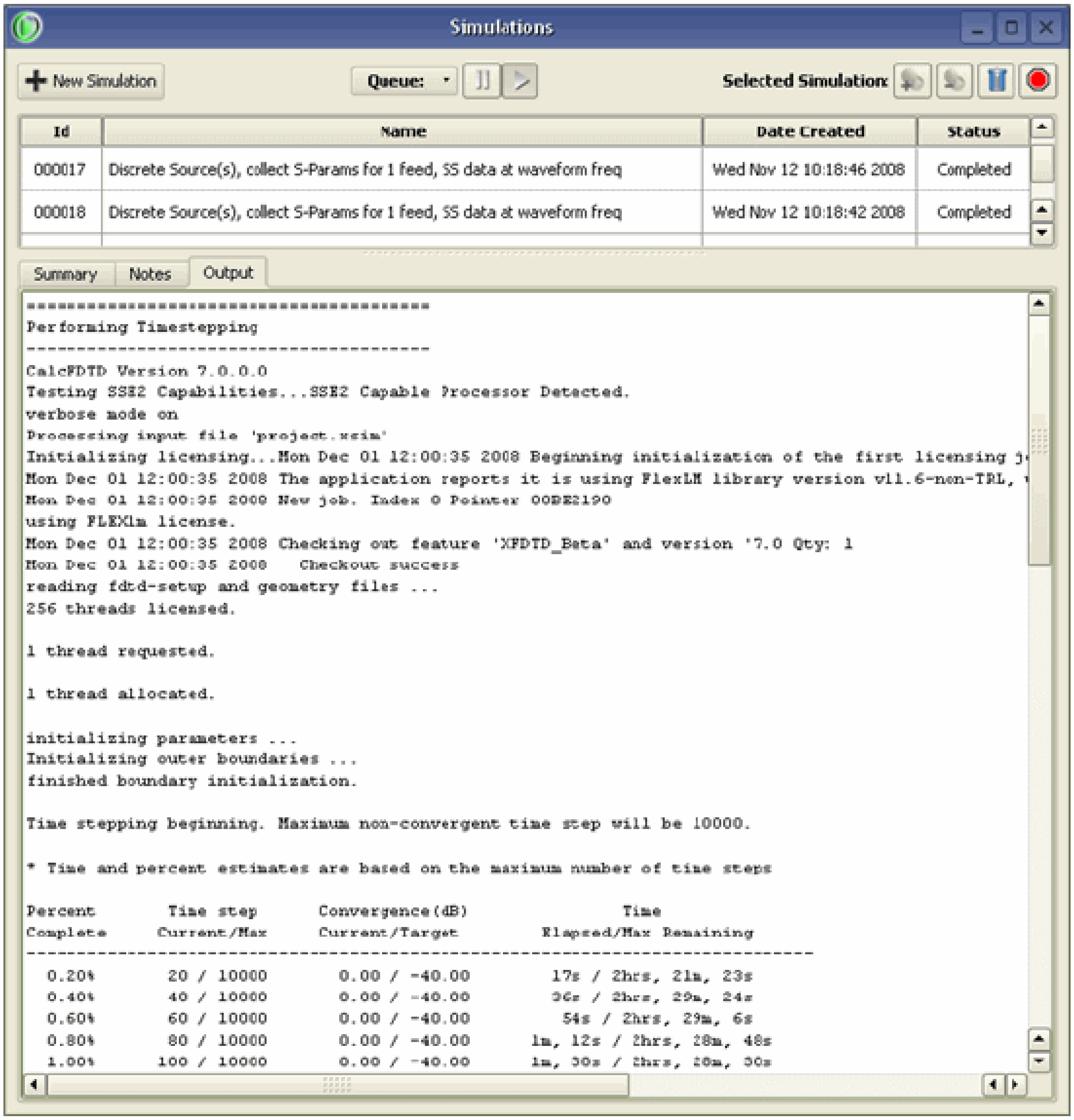
Acceleware Hardware Acceleration

EMPro has optional Acceleware hardware acceleration. The hardware is available in the form of graphics cards, which can replace existing graphics cards or act from a stand-alone computer. When this hardware is available, it may be enabled within the *Simulations* workspace window under the *Queue* drop-down box. To enable it from the command line, the "`-hardware`" option should be used.

Monitoring Calculation Progress

While EMProFDTD is running, its progress will be updated periodically. When launched from the EMPro interface, the progress of the calculation will be printed in the *Output* tab located within the *Simulations* workspace window, as seen in the figure below. When running from a command line, the progress will be printed to the window that was used to start the calculation if the "`-v`" option is used.

Monitoring calculation progress in the Output tab



Calculation Stability

Improper application of the outer boundaries can lead to unstable calculations. Typically the stability of a calculation depends on a few simple guidelines involving boundaries and source placement. Because an absorbing boundary condition like LIAO tries to simulate free space, it requires that a certain amount of continuity be present in the cells leading to the boundary. The cross section at a boundary must be the same for at least 10 cells in from the boundary.

Note
For many problems, a free-space border of 10-20 cells is the best way to ensure stability and accurate performance of the outer boundary.

Another rule for stability is that no source can be placed within 10 cells of an absorbing boundary. An unstable calculation is easily determined by viewing the line plots of time domain data or by viewing the field snapshots. When automatic convergence is enabled, EMProFDTD can automatically detect an unstable simulation and terminate it.

In some cases instability can be introduced by a frequency-dependent material. If such materials are used in the calculation and an instability results, it may be necessary to change the material parameters or reduce the calculation timestep.

Viewing Output

In this section, you will learn about:

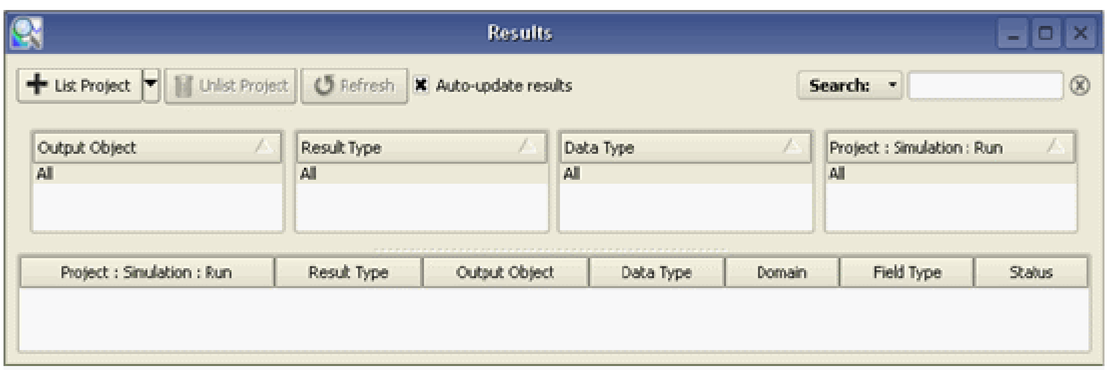
- How to read and filter your calculation results
- The available tools to analyze 2-D (plotted) and 3-D (field display) results

Following an EMPro calculation, the results may be reviewed in the *Results* workspace window. The results that are available in this window depend on the characteristics of the project such as discrete sources, sensors, external excitations, and other project criteria specified in the *Simulations* workspace window.

This section details the review of results available in this window. Some results will be in the form of numerical values. These are typically single-frequency results performed with a near-zone source. Other results will be displayed in the form of plots. There are several types of plots available to view results based on whether they are time-dependent, frequency-dependent, or angle-dependent. Finally, some results will be available to review as colored field displays. Broadband results collected by *Surface* sensors and *Solid* sensors are viewed as individual **"Field Snapshots"** or **"Field Sequences"** (strings of snapshots). Three-dimensional far-zone fields may also be available for view depending on simulation criteria.

Results Workspace Window

The Results workspace window



The *Results* workspace window stores all of the data collected by the *Sensors* during the calculation. Once the simulation is queued for calculation, the project will be listed in this window so that the results can be viewed. Additionally, the results of any other saved project can be loaded by pressing the *List Project* button in the upper-left of the dialog. This makes it possible to view and compare the results of multiple projects without having to load several projects individually. Similarly, the results of any project may be closed by selecting the project and clicking the *Unlist Project* button.

While the calculation progresses, several options may be selected to control how data is refreshed in the *Results* workspace window as more and more results become available. The list of results in the workspace window may be manually updated by selecting the *Refresh* button. The list will be automatically updated if *Auto-Update Results* check box is selected. This check box will also automatically update plotted data when the specific plot window is opened as the calculation progresses.

Filtering Data Results

It is possible to filter the data within the *Results* workspace window by searching or by categorizing. The *Search* box in the top right corner enables you to to search for the name of any field or data visible within the window. The results will automatically appear in the list below.

The user is also able to customize the four columns at the top of the window, which filter data according to the specified categories. The column headings are controlled by right-clicking on any of the current column headings and selecting one of the available categories. Each category is described in the table below.

Data Filtering Options

Data Filter	***Description***
Project Id	Displays the EMPro project ID, which references the location of the loaded project in the file directory.
Project Name	Displays the project name indicated by the user.
Simulation Id	Displays the simulation ID that refers to the simulation's location in the file directory.
Simulation Number	Displays the simulation number that is automatically generated based on how many simulations have been created in a specific project.
Simulation Name	Displays the simulation name specified in the *Simulations* workspace window.
Run Id	Displays the run ID which references the location of the run in the file directory. Multiple runs are created in the case that a simulation collects data for more than one variable or location (i.e., during parameter sweeps, multiple ports, etc.).
Run Number	Displays the run number.
Result Type	Filters results by type. Any of the following results may be viewed depending on the project, simulation criteria, and the type of data that was requested: Electric Field (E) Magnetic Field (H) Conduction Current (Jc) Magnetic Flux Density (B) Poynting Vector (S) Voltage (V) Current (I) Impedance Available Power Input Power Instantaneous Power Feed Loss Reflection Coefficient S-Parameters SAR (Specific Absorption Rate) Maximum SAR Value Average SAR in Exposed Object Average Power Net Input Power

	Net Feed Loss Net Available Power System Efficiency Radiation Efficiency Dissipated Power Dissipated Power in Tissue Dissipated Power in Non-Tissue Dissipated Power per Material Radiated Power Dissipated Power per Electric Material Component Dissipated Power per Magnetic Material Component Dimension (time, position, etc.) Axial Ratio Radar Cross Section Gain Voltage Standing Wave Ratio (VSWR)
OUTPUT OBJECT	Filters results according to its *Output Object* , which refers to a specific sensor by name.
DATA TYPE	Filters results according to the type of sensor that collected the data. The sensor is referenced by its general type, rather than by its user-defined name (Refer to *Output Objects* for filtering data by sensor name). The following is a comprehensive list of the sensors that may be listed within this filter: **Point Sensor** - Retrieves data from any *Point* sensor in the project. **Surface Sensor** - Retrieves data from any *Surface* sensor in the project. **Solid Sensor** - Retrieves data from any *Solid* sensor in the project. **Far Zone Sensor** - Retrieves data from any FAR ZONE sensor in the project. **Raw SAR Sensor** - Retrieves raw SAR data. **Averaged SAR Sensor** - Retrieves averaged SAR data. **Circuit Component** - Retrieves data from a *Ciruit Component* **HAC Sensor** - Retrieves hearing aid compatibility data from HAC sensors. **System** - Retrieves ambient result data (not associated with a sensor object). **External Excitation** - Retrieves data on the *External Excitation* waveform. **Raw Steady-State Far-Zone Data** - Contains information which can be used to generate new steady-state far zone patterns after a simulation is run.
Project: Simulation : Run	Displays the Project, Simulation, and Run name in one column.
Domain	Filters data according to Time, Frequency or Discrete Frequencies domains.
Field Type	Filters data according to total-field or scattered-field.
Status	Displays whether a result is complete or still being calculated while the simulation is running. This status can be refreshed manually by pressing the *Refresh* button or automatically by selecting *Auto-Update Results* .
Misc	Displays query-specific information. For example, for the circuit component voltage when collecting S-Parameters, it could contain the active port. For S-parameters, it could contain the **S** number or simply the active port. For System numbers, it could contain the number (such as efficiency) or net input power.

Viewing Numerical Results

When a single-frequency calculation has been performed with a near-zone source (voltage or current), parameters such as input impedance, S-parameters, VSWR and the reflection coefficient are displayed as numerical values rather than as line plots since the data only is relevant for the input frequency. Other numerical values are collected by means of a system sensor, which is automatically present in every EMPro project.

Numerical Data Collected by EMpro

Results Displayed as Numerical Data

Sensor Type	Time Dependence	Result Type
Port Sensor	Single-Frequency	Available Power
		Current
		Impedence
		Input Power
		Reflection Coefficient
		S-Parameters
		Voltage
		VSWR
System Sensor	Single-Frequency	Dissipated Power
		Dissipated Power in Non-Tissue
		Dissipated Power in Tissue
		Dissipated Power Per Electric Material Component
		Dissipated Power Per Magnetic Material Component
		Dissipated Power Per Material
		Net Available Power
		Net Feed Loss
		Net Input Power
		Radiated Power
		Radiation Efficiency
		System Efficiency

Impedance

All input impedances are calculated by the ratio of the complex V over complex I for the FDTD mesh edge at the port location. The sign convention is positive for power flow into the antenna or other structure. Thus for a port which delivers power to the antenna, the impedance will have a positive real part, and for a port which absorbs power from the antenna, the impedance will have a negative real part.

When only one active port is present, the feed point impedance at that port is the self-impedance at that port. If more than one port is active, the port impedance values listed represent the ratio of the complex voltage and current at each port including the effects of the sources at all active ports. Thus these impedance values are not the self-impedances at each port, but rather terms of an impedance matrix.

S-Parameter Calculations

When an S-parameter calculation is made with the steady-state far-zone transformation selected, then both the steady-state antenna data and the S-parameter data will be displayed. This may be useful, for example, when making calculations for a microstrip antenna when both $\mathbf{S_{11}}$ and input impedance are of interest.

Net Available Power

The available power is calculated as the total power delivered by the active port or ports into a matched load.

Radiated Power

The radiated power is computed as the difference between the total net input power delivered by the active ports and the dissipative losses from conductive materials and resistive loads in inactive ports.

System Efficiency

The system efficiency is calculated as the ratio of radiated power to available power. Thus, it includes radiation efficiency and mismatch efficiency.

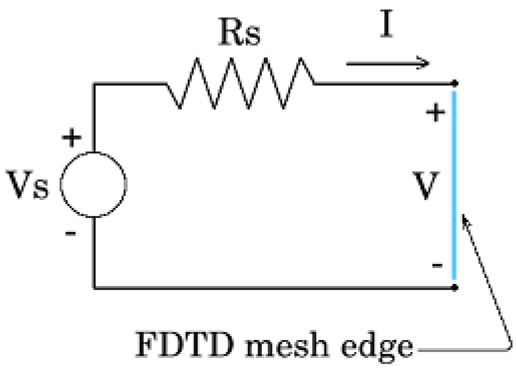
Power Scaling Factor

When viewing single-frequency results collected by the system sensor, the user can specify a scaling factor of the available power to the calculation and determine the scaled result. The absolute input power can be adjusted as desired or an overall scaling can be applied. For example, if the user wanted to know what the output power would have been with an input of one mW, they can enter this value in the available power (will be subject to mismatch loss) or in the net power (after mismatch loss) to see the overall effect. Similarly, any of the output results may be scaled to determine the effect on input and output. Clicking reset will return the values to the un-scaled state.

Input power, Input Impedance, and Loss

When making calculations that include the input voltage, current and/or power in the calculation formulas, such as antenna gain or input impedance, the input voltage, current and/or power will be that provided at the terminals of the mesh edge. Referring to the following figure, the impedance at the port would be the (complex) mesh edge voltage V divided by the (complex) mesh edge current I. The complex values would be obtained from an FFT for a broadband calculation or from two samples of the voltage and current (electric and magnetic fields at the port mesh edge) for a sine wave excitation. If the port is delivering power, then the impedance at that port will have a positive real part. If the port is absorbing power, then the impedance at that port will have a negative real part and the input power will be negative. This is determined by the direction of current flow and voltage polarity for the FDTD mesh edge.

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



Active ports (those with active voltage or current sources) are treated differently than inactive ports for some calculations. For an antenna calculation, the input power to the antenna is the algebraic sum of all powers delivered by active ports. Power absorbed by active ports will not reduce the antenna efficiency; however, power absorbed by inactive ports will reduce antenna efficiency.

To clarify this, consider two different situations. An antenna composed entirely of perfect conductor includes 2 ports, the first containing a 1-V source and a resistor, the second a passive port containing only the resistor. Assuming that some current flows in the passive resistor, the antenna efficiency will be less than 100%. If we repeat the same calculation, but with a 0.00001-V source added to the formerly passive port, and with both sources set active, the antenna efficiency will now be calculated as 100% even though the 0.00001-V source will have negligible effect on the antenna currents and radiation. This discrimination is done so that active ports may utilize lumped circuit elements to match to an antenna without changing the antenna efficiency, impedance, and gain, while passive elements may be added to an antenna with their effects included in the antenna efficiency and gain results.

Similarly, for input impedance calculations, the source resistance, capacitance and inductance values will not be included in the input impedance. For example, the input impedance of an antenna as calculated using EMPro should not change regardless of any changes in the active port components (source/R/L/C). This is as it should be, since the antenna impedance is a function of the antenna geometry/materials and not of how the antenna is fed.

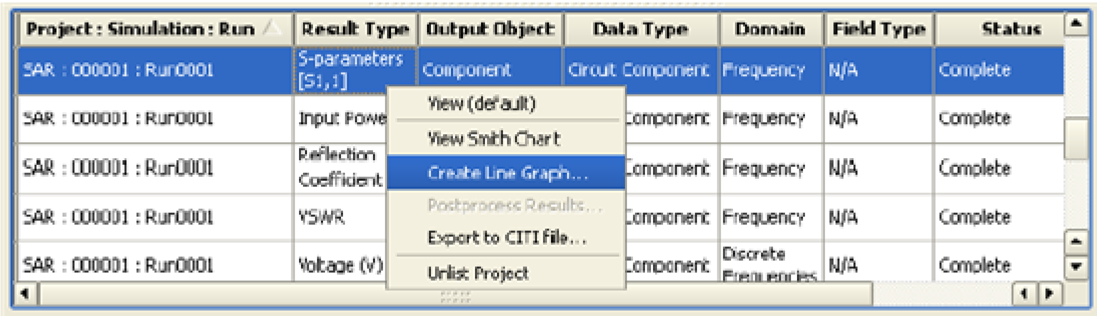
Viewing 2-D Plotted Results

Some results are viewed as 2-D plots. There are three basic categories of plots in EMPro, depending on the ***X*** -axis (abscissa) type:

- **XY PLOTS** - visualizes data in a Cartesian coordinate system.
- **POLAR PLOTS** - represents data with an angle-independent axis.
- **SMITH CHARTS** - displays complex data vs. frequency, such as S-parameters and reflection coefficient.

The 2-D plots are viewed by right-clicking on the desired result type in the *Results* workspace window and selecting *Create Line Graph*, as seen in the following figure.

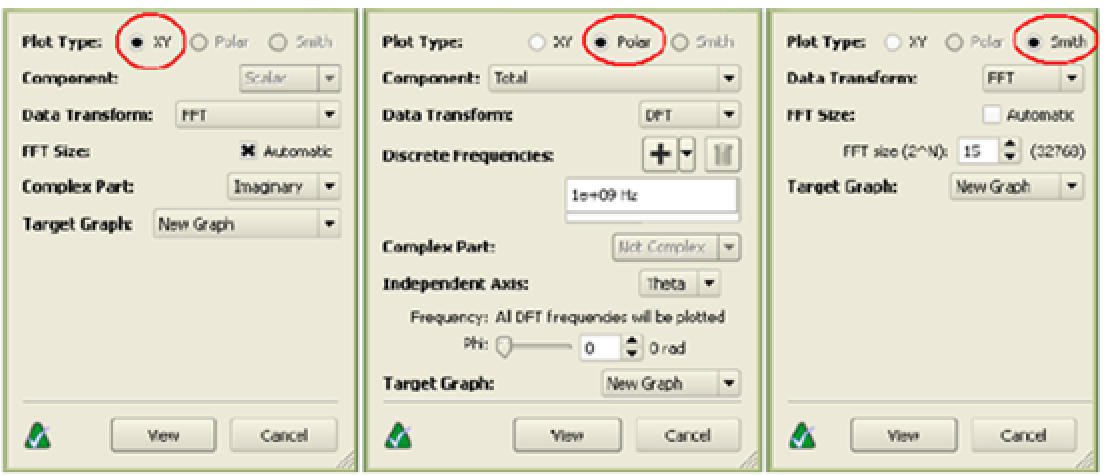
Results right-click menu



Before the graph is displayed, the user has the option of adjusting the graph properties such as the component, data transform, and complex part. The Target Graph option enables you to view and edit plots that were previously created (from the same data selected in the Results window).

The following figure shows samples of the create graph dialog for each plot.

Create graph dialog for XY, Polar, and Smith plots



For all calculations, the most important quantities are the time-domain plots of the fields in the problem space. Always perform a quick review of these values to ensure that a calculation has converged. Without convergence, most other results will be meaningless, particularly any plots converted to the frequency domain such as S-parameters or impedance.

Note
Note that input impedance and S-parameters may be plotted in rectangular form vs. frequency or as a Smith chart.

Results Displayed as Plotted Data

Plotted Data Collected by EMPro

Sensor Type	Time Dependence	Result Type
Port Sensor	Broadband	Current
		Impendence
		Input Power
		Instantaneous Power
		Reflection Coefficient
		S-Parameters
		VSWR
Point Sensor/Surface/Volume	Broadband	E-Field (E)
		H-Field (H)
		B-field (B)
		Conduction Current (Jc)
		Poynting Vector (S)
System Sensor	Broadband	Net Feed Loss
		Net Input Power
Far Zone Sensor	Broadband	Radar Cross Section
		Gain
		E-Field (E)
Far Zone Sensor	Single-Frequency	Radar Cross Section
		Axial Ratio
		E Theta, E Phi
		Circular Polarization
		Ludwig-2 Az, El
		Ludwig-2 Al, Ep
		Ludwig-3
Far Zone Post Processor	Single-Frequency	E-Field (E)

Customizing Plots

There are several ways to customize plotted data after it is opened from the *Results* workspace window. The following sections detail the various tools that are available to modify plotted results.

> **Note**
> Keep in mind that the tools available within the plot window depend on the type of graph you are viewing (i.e., *XY Plots*, *Polar Plots*, or *Smith Charts*).

Export Data

Select this option to export graphical data point values to a text file in a specified directory.

Export Image Tool

Select this icon to save an image of the current plot to a specified directory.

Pan Tool

Select this tool, and drag the mouse within the plot to pan to the desired view of the plot. Press the |**Ctrl**| key and drag to pan along the independent axis. Press the |**Shift**| key and drag to pan along the dependent axis.

Zoom Tool

Select this tool to zoom-in and to zoom-out of the plot. The mouse wheel as well as the right and left mouse buttons, can be used to perform zoom operations as described below.

Using mouse-wheel:

- Roll the center wheel of the mouse forward to zoom-out of both axes simultaneously.
- Press |**Ctrl**| and roll the center wheel forward to zoom-out of the independent axis.
- Press |**Shift**| and roll the center wheel forward to zoom-out of the dependent axis.
- Roll the center wheel of the mouse backward to zoom-in to both axes simultaneously.
- Press |**Ctrl**| and roll the center wheel backward to zoom-in to the independent axis.
- Press |**Shift**| and roll the center wheel backward to zoom-in to the dependent axis.

Using mouse buttons:

- Right-click and drag the the mouse anywhere in the plot window to zoom-out of both axes simultaneously.
- Press |**Ctrl**| and right-click/drag to zoom-out of the independent axis.
- Press |**Shift**| and and right-click/drag to zoom-out of the dependent axis.
- Left\click and drag to define a rectangular view-window in the plot window to zoom-in to both axes.
- Press |**Ctrl**| and left-click/drag to zoom-in to the selected domain of independent axis values.
- Press |**Shift**| and and left-click/drag to zoom-in to the selected range of dependent axis values.

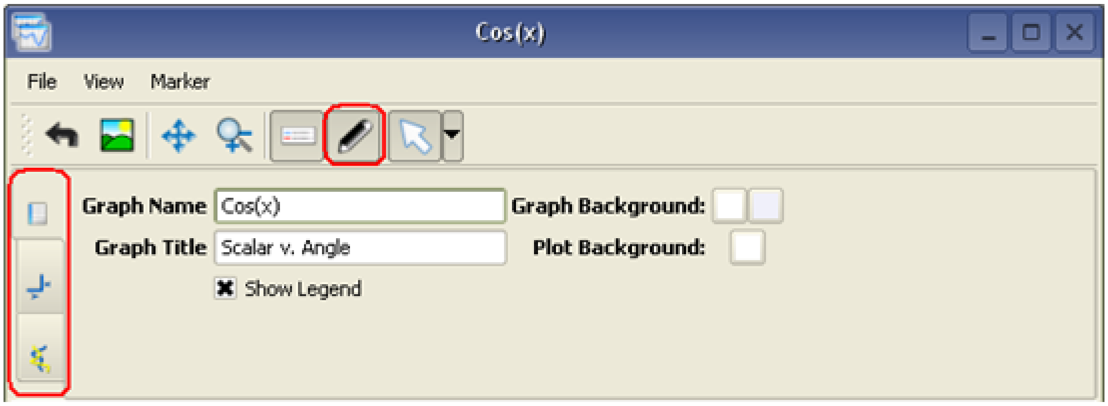
Legend Visible

This button toggles the display of the legend within the graph.

Graph Properties Tool

The following illustration highlights the three tabs that are available for editing the properties of any graph in EMPro. Each tab is detailed below.

Tabs available for editing graph properties



Title properties

Define the graph name and title as well as the background color of the graph in this tab. A checkbox also toggles the legend display on and off. This tab is displayed in the figure above.

Axes properties

Define the title of the axes and the limits of the axes in this tab. The *Auto* checkbox may be selected to auto-select these limits.

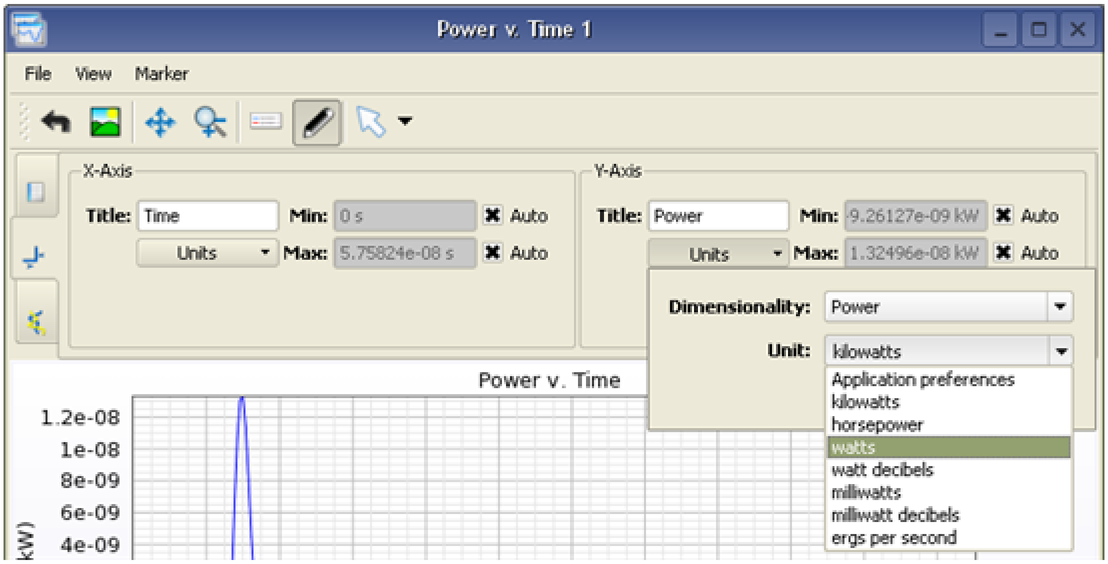
> **Note**
> If a graph only contains continuous data, the ***X*** -axis must be specified (this will not happen automatically).

The *Units* drop-down menu is used to specify the units and to apply a log scale if necessary.

For certain Smith plots, you will also have the option of modifying the *Reference Impedance*.

The figure below shows the axes properties editor for a 2-D XY graph.

Editing the axes of graphs



Plot properties

Define the characteristics of the plotted lines in this tab. A list of every dependent variable is listed with a customizable line color, line width, and line style. Any unwanted variables can also be deleted in this tab.

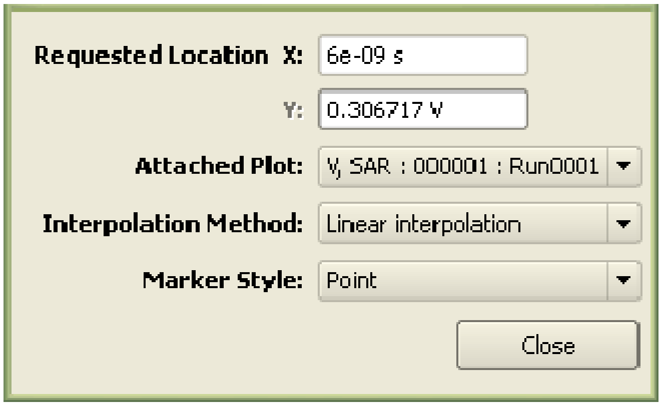
Selection tool

Select this tool to move, delete, or edit a marker's properties. To move a marker, click on the marker with the selection tool (once selected, it will turn yellow) and roll the mouse-wheel forward or backward to move an attached marker along its plot. Pressing |**Ctrl**| and rolling the mouse-wheel will speed up the movement of the marker.

To delete any marker, simply select the unwanted marker, right-click and press *Delete Marker* . All markers may be deleted at one time by right-clicking anywhere in the plot area and choosing *Delete All Markers* .

To edit a marker's properties, click on the desired marker so that it turns yellow, then right-click and select *Marker Properties* . A window will appear with several different editing options. The location coordinates of the marker can be adjusted by manually typing in the desired values in the *Requested Location* section of the dialog box.

Marker properties dialog



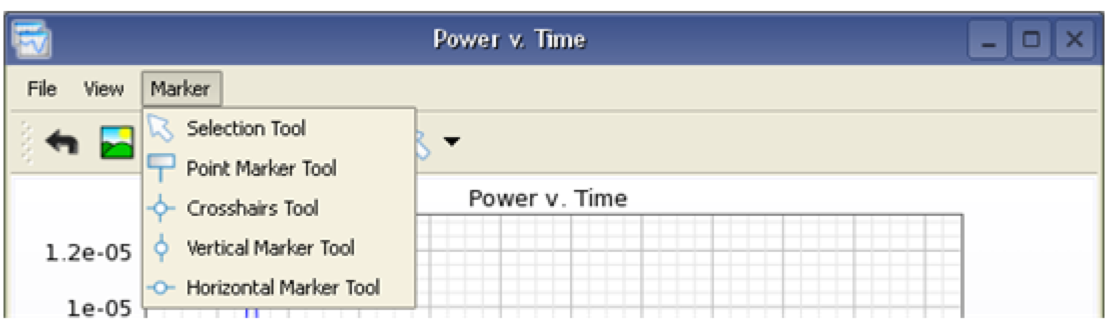
Note
If the coordinate box you desire to edit is disabled, select the appropriate option in the *Attached plot* drop-down menu.

The marker may also be attached to a particular plot by selecting its name in the *Attached Plot* drop-down menu and selecting an *Interpolation Method* for EMPro to use in order to place the point. Depending on this definition, the marker will be shifted to the nearest point on the selected function or linearly interpolated based on the independent axis that is entered by the user in the *Requested Location* dialog box. Finally, the type of marker can also be redefined in this editor window in the) *Type* drop-down list.

Additionally, this tool can be used to move or close the legend in the graphical space. (Any tool, however, can be used to perform this function.)

Select this tool by clicking on its icon or select it from the *Markers* menu, as shown in below.

Marker drop-down menu



Point Marker Tool

Select this tool to mark any point on the plot by clicking on the desired marker location. A marker with the location coordinates will appear above the point, depending on the type of plot:

- **XY Plot:** (***X*** -location, ***Y*** -location)
- **Polar Plot:** (radius, angle)
- **Smith Plot:** (real part of location, imaginary part of location, frequency)

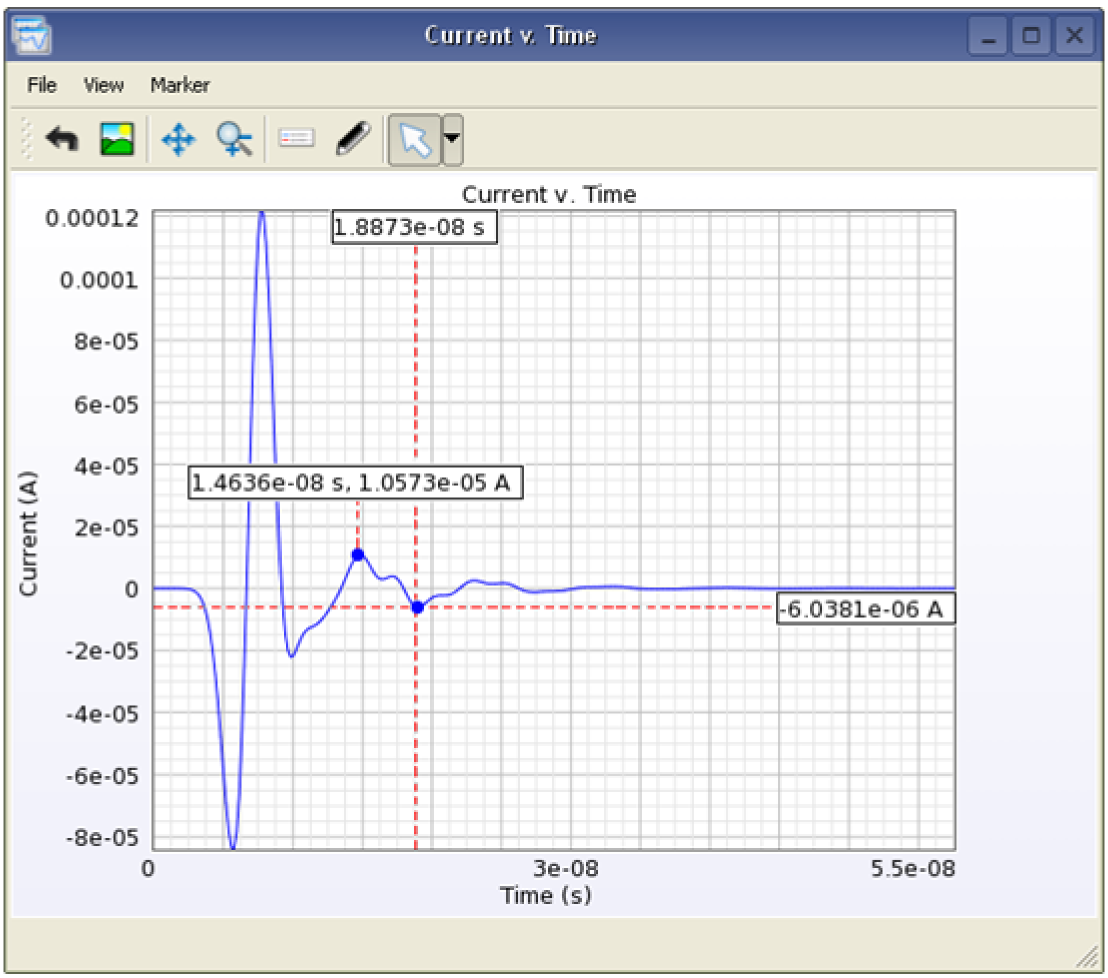
When the mouse moves close to the plotted curve, it is snapped to the closest location on the interpolated line or sampled point. Holding the |**Ctrl**| key will disable the snapping action, allowing a point to be placed anywhere. Holding the |**Shift**| key will snap the marker to sampled points only. Note that markers placed on sampled points are blue, and markers placed on interpolated points are black.

Crosshair Marker Tool

Select this tool to mark the location of a single point by two intersecting cross-hairs. The marker is placed at the right edge of the plot. (Snapping actions are the same as the Point/Tracker Marker described above.)

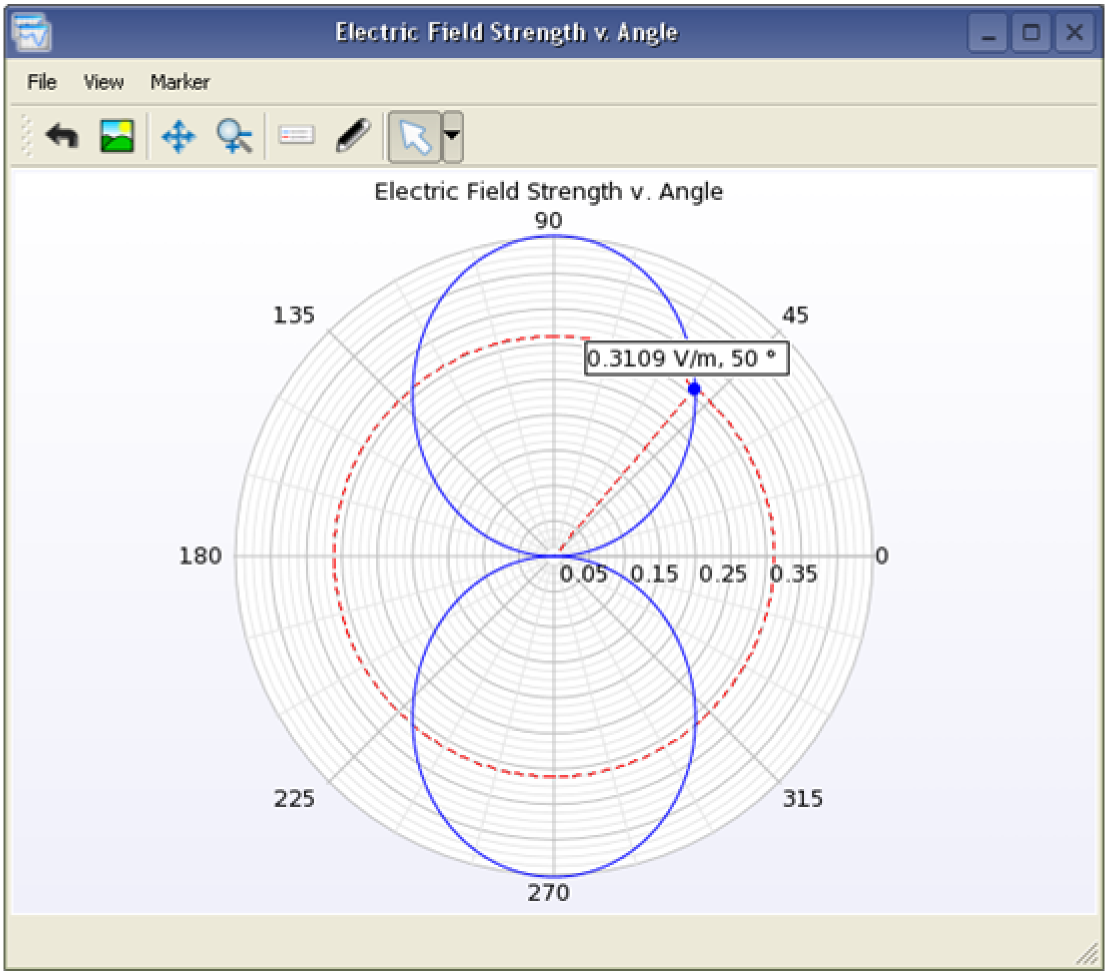
The following figure shows an *XY Plot* with a Crosshair Marker (1.8873e-08 s, -6.0381e-06 A) and Point Marker (1.4636e-08 s, 1.0573e-05 A).

2-D XY graph with Crosshair and Point markers (highlighted in red)



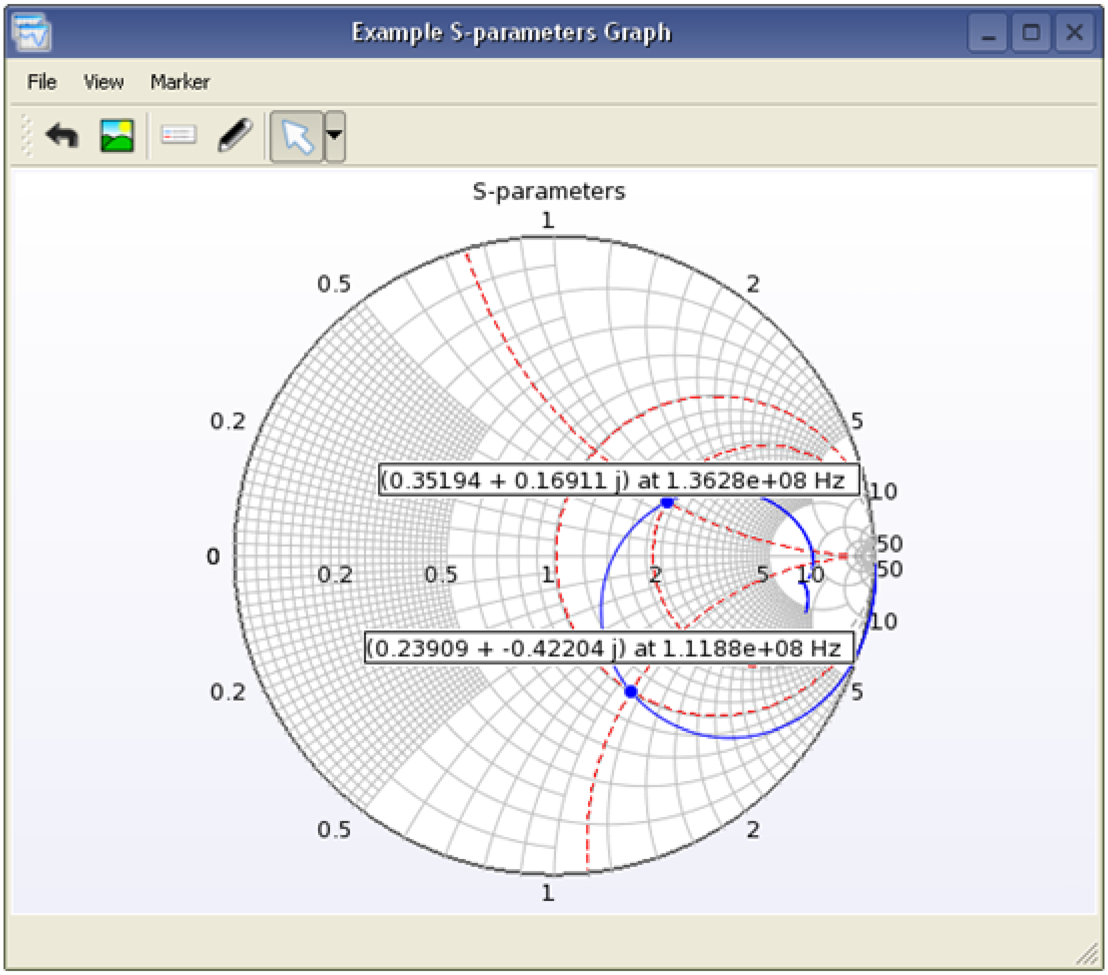
The figure below shows a polar graph with a Crosshair Marker along a radius.

Polar graph with Crosshair marker (highlighted in red)



The following figure shows a smith graph with a *Crosshair Marker* along a radius.

Smith graph with Crosshair markers (highlighted in red)



Vertical Marker Tool

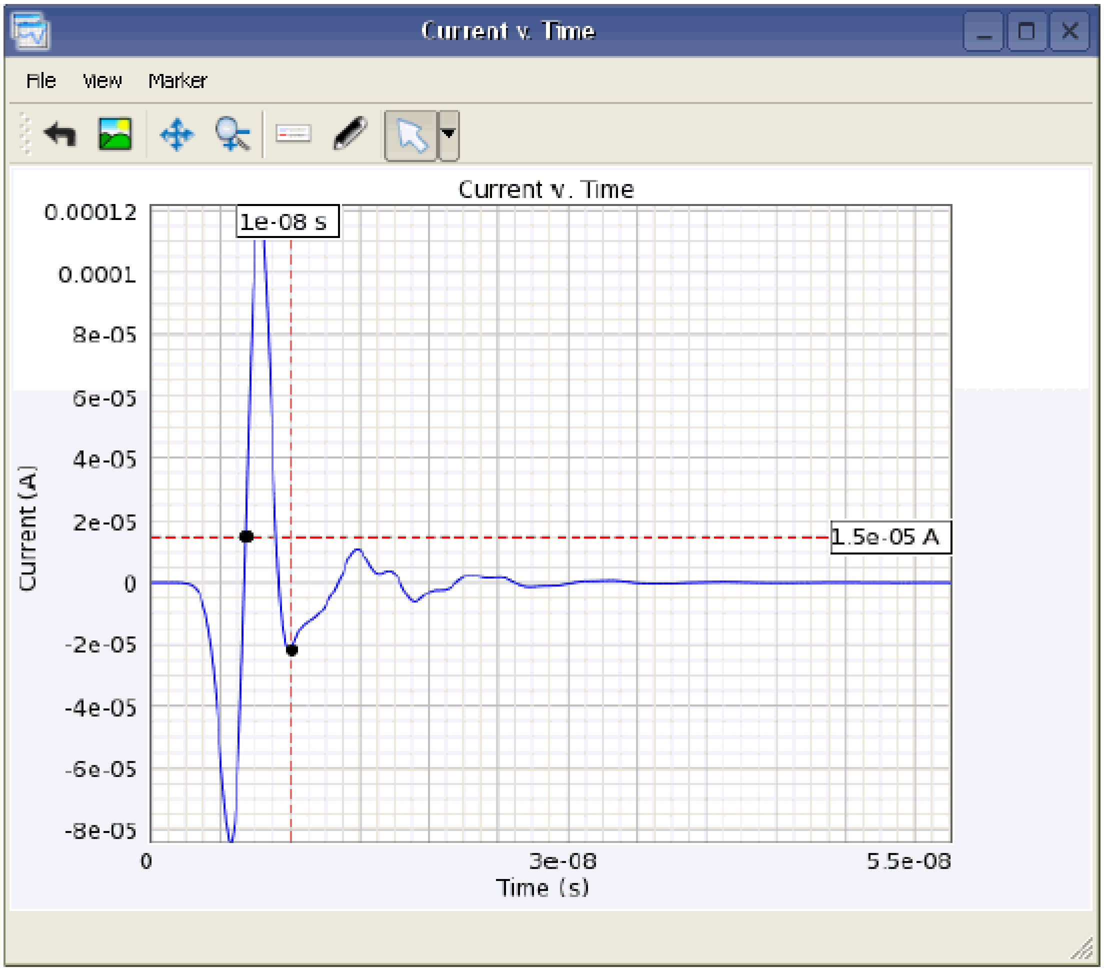
Select this tool to place a vertical line that intersects with the ***X*** -axis. The marker (***Y*** = constant) will be placed along the top-edge of the plot area. (Snapping actions are the same as the Point/Tracker Marker described above.)

Horizontal Marker Tool

Select this tool to place a horizontal line that intersects with the ***X*** -axis. The marker (***X*** = constant) will be placed along the right-edge of the plot area. (Snapping actions are the same as the Point/Tracker Marker described above.)

The figure below shows a 2-D XY Graph with a Vertical Marker (at ***X*** = 1e-08 s) as well as a Horizontal Marker (at ***Y*** = 1.5e-05 A).

2-D XY graph with Vertical and Horizontal markers (highlighted in red)



3-D Field Displays

Colored 3-D field displays may be viewed in slices of the geometry by saving either broadband or single-frequency field quantities. The fields may be viewed in with the geometry (solid or meshed), or by themselves.

Three types of fields may be viewed with the geometry:

- **Time Domain Snapshots** - available for any calculation as they are simply snap shots of the near-zone fields at specific steps in time.
- **Complex Fields or Derived Quantities** (such as SAR) - available at specific frequencies
- **Three Dimensional Far-Zone Fields** - can be requested either before the calculation with a far-zone sensor, or after a calculation through post-processing.

Data Collected With Near-Field Sensors

Field snapshots are listed in the field control panel as either single slices or as field sequences. The field sequences are movies of the individual slices as the fields progress with time in a particular slice of the geometry. Depending on what was saved, each field snapshot may have electric and magnetic fields, current densities, Poynting vectors stored for each direction (***X*** , ***Y*** and ***Z***) and a display of the combined magnitude. Field snapshots are collected by surface sensors and solid volume sensors for broadband data.

> **Note**
> In order to view single-frequency data, the *Collect Steady-State Data* check box must be checked in the *Simulations* workspace window (under the *Frequencies Of Interest* tab).

Results Plots Available For Each Sensor Type

Sensor Type	Result Type	Time Domain	Discrete-Frequency	Broadband
Point Sensor	E-Field (E)	X		X
	H-Field (H)	X		X
	B-field (B)	X		X
	Poynting Vector (S)	X		X
	Conduction Current (Jc)	X		X
	Scattered E	X		X
	Scattered H	X		X
	Scattered B	X		X
	Average Power			X
Surface/Solid Sensor	E-Field (E)	X	X	
	H-Field (H)	X	X	
	B-field (B)	X	X	
	Conduction Current (Jc)	X	X	
	Scattered E	X	X	
	Scattered H	X	X	
	Scattered B	X	X	
	Poynting Vector (S)	X		
	Average Power		X	
SAR Sensor	SAR (Specific Absorption Rate)		X	
HAC Sensor	E-Field (E)		X	
	H-Field (H)		X	
	HAC max E-Field (E)		X	
	HAC max H-Field (H)		X	

(Scattered) Electric Fields (E): magnitude, normal, or ***X*** , ***Y*** or ***Z*** components of the (scattered) electric field data at each cell edge.

(Scattered) Magnetic Fields (H): magnitude, normal, or ***X*** , ***Y*** or ***Z*** components of the (scattered) magnetic field data at each cell edge.

(Scattered) Magnetic Flux Density (B): magnitude, normal, or ***X*** , ***Y*** or ***Z*** components of the (scattered) B-field computed from the magnetic fields at each cell edge and the associated permeability for that cell edge.

Average Power Density (SAV): magnitudes of the average power density computed from the electric and magnetic fields at each cell edge.

Conduction Current Magnitude (Jc): conduction current at an electric field cell edge.

SAR (Specific Absorption Rate): computed for each complete cell containing a lossy dielectric with a non-zero material density.

Data Collected with Three Dimensional Far-zone Fields

The *Results* workspace window also displays the data collected within 3-D far-zone fields. Any 3-D far-zone request generated using EMPro's post-processing engine (run separately from the calculation engine) is automatically added once the post-processing is complete. If steady-state far-zone data is enabled for the sensor, the 3-D results will include the E-field and axial ratio in the discrete-frequency domain. Gain or discrete-frequency radar cross section (RCS) will also be available when using a feed or external excitation, respectively.

	Note Note that the polarization (Theta/Phi, Ludwig-2, etc.) is selected through the *Setup* tab of the *Field Editing Toolbar*.

Viewing 3-D Field Displays

EMPro displays 3-D field data in the *Geometry* workspace window. The following subsections discuss how to configure and analyze the display, using the *Scale Bar*, the *Field Reader Tool*, rescaling, and the field editing toolbar.

The Scale Bar

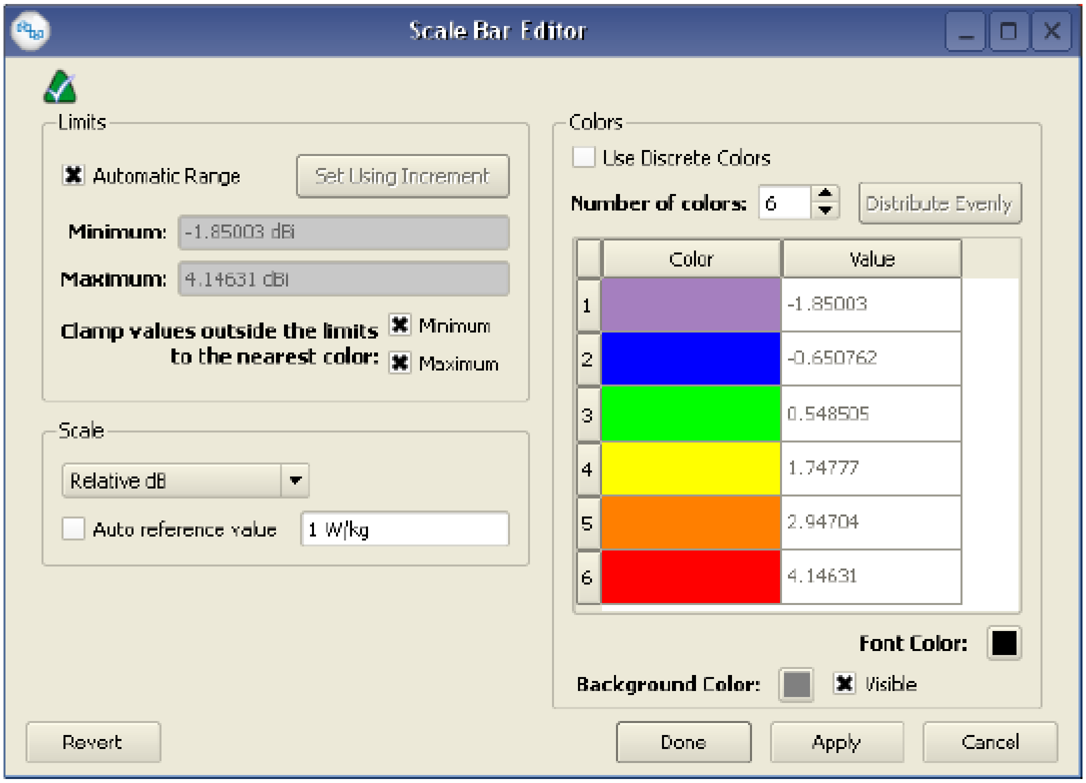
The *Scale Bar*, located at the top of the *Geometry* workspace window, "paints" the view with a range of colors which correspond to the range of values displayed. By default, the *Scale Bar* color palette is shown in continuous mode within a default range of values. You can adjust these properties by right-clicking on the scale bar and selecting *Discrete Mode*, to change the palette to discrete colors, or *Automatic Range*, to change the range of values to that which is actually present.

There is also a *Properties* option under the right-click menu. Clicking this will bring up the Scale Bar Editor. Using this editor, you can manually set the *Scale Bar* limits, units and colors to your preference. Take note of several settings that may not be intuitive for the first-time user:

- Under the *Limits* section, it is possible to "**clamp**" values outside the defined scale bar limits to the nearest color. This makes it possible to view outliers that otherwise would not be colored with the *Scale Bar*.
- Under the *Scale* section, when *Relative dB* is selected, you can define its reference value or use the value automatically selected by EMPro.
- Checking *Use Discrete Colors* makes it easier to view contours in the 3-D results.
- *Font Color* and *Background Color* change the colors of the *Scale Bar* itself, not the coloration of the results.

The following two figures show the *Scale Bar Editor*, and the Scale Bar_ with the *Field Reader* tool.

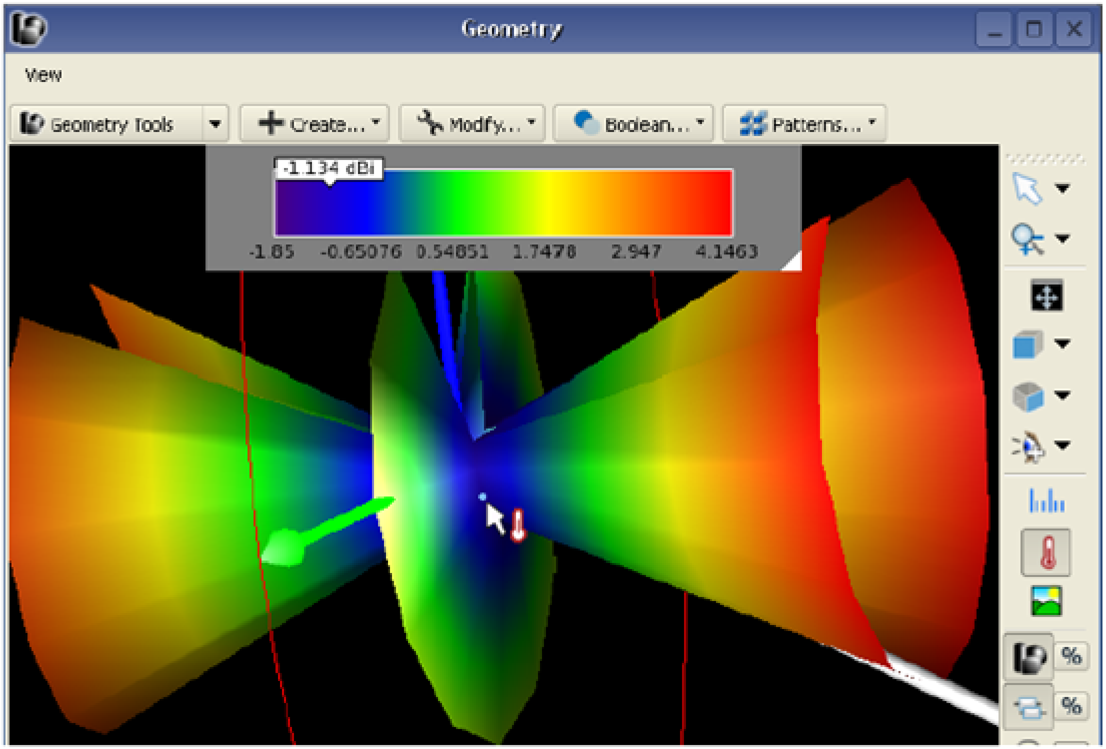
The Scale Bar Editor



The Field Reader Tool

The *Field Reader Tool* is located in the toolbar to the right of the *Geometry* workspace simulation space. After selecting this icon, wheel the mouse over the geometry object to identify its field values. A marker in the Scale Bar will display the nearest known field value to the location of the mouse. This location is represented by a small dot on the screen.

The Field Reader tool



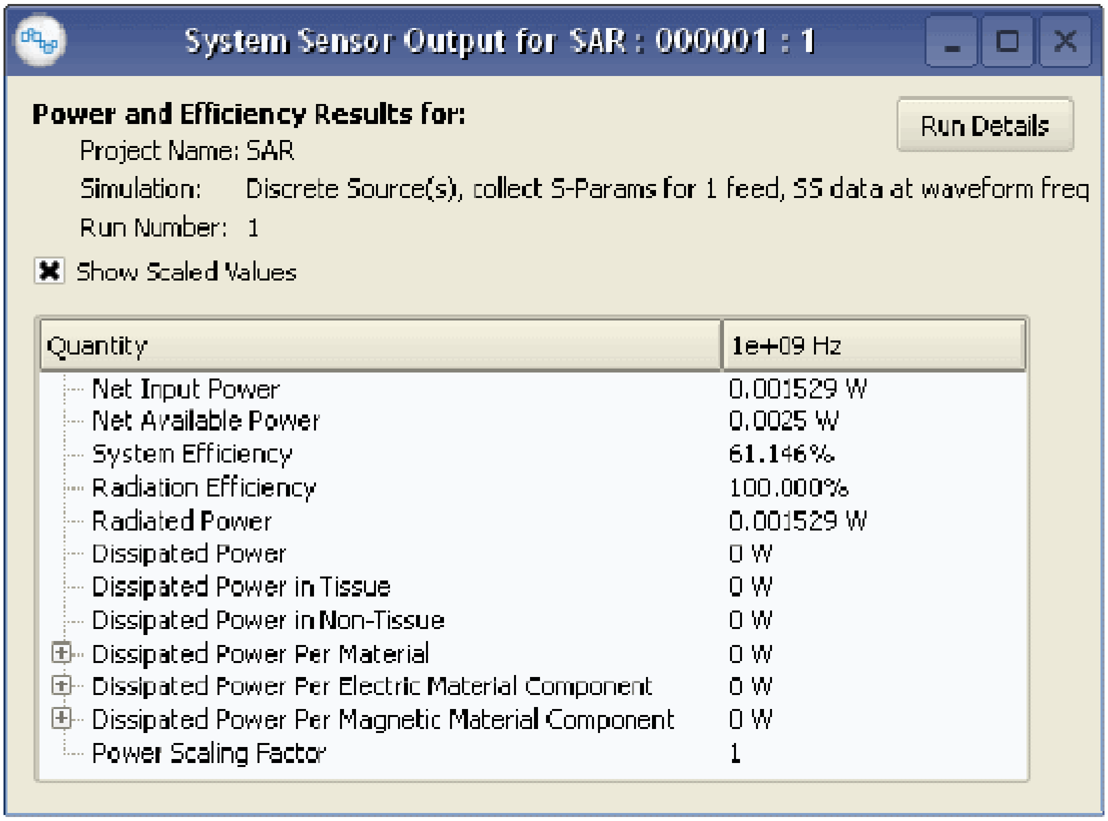
Rescaling

In the *Results* workspace window, filter the results by the output object System. Double-clicking on results such as *Dissipated Power* or *System Efficiency* will bring up the *System Sensor Output* dialog. Changing any value in this table will rescale the other values shown.

> **Note**
> The *Show Scaled Values* box must be checked to enable editing in the *System Sensor Output* dialog.

- Scaling only affects results which are in the discrete frequencies domain.
- When you change the scaling by editing values in the *System Sensor Output* dialog, you only scale the results for that particular calculation engine run at that particular DFT frequency. Any other tables of data and plots associated with this run and frequency will also be affected.

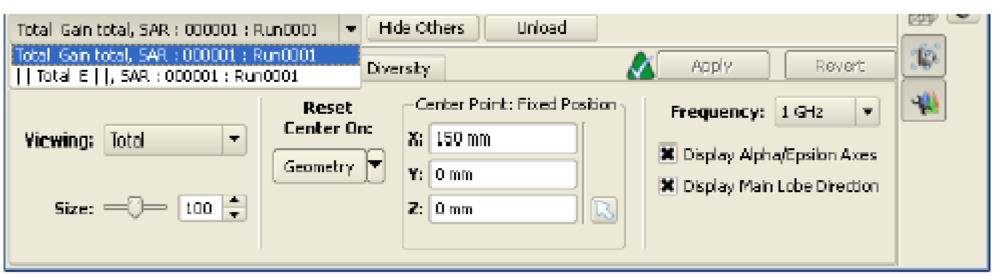
System Sensor Output dialog



The Field Editing Toolbar

Located at the bottom of the *Geometry* workspace window, the field editing toolbar is used to configure the properties of the view. In the upper-left corner of this toolbar, a drop-down list will display any view that you have opened from the Results window. You can use the *Hide Others* and *Unload* buttons to single out certain view(s) if necessary. The following figure shows a drop-down list of such results in the field editing toolbar.

The results drop-down list of the field editing toolbar



The tabs and configuration options below will change depending on the active view. A comprehensive list of the available options are described below.

The Setup Tab

There are two main configurations for the *Setup tab*, depending on the type of sensor you are evaluating.

- Keep in mind that not all fields listed will be available for every view.

Surface, Solid SAR Sensor Configuration

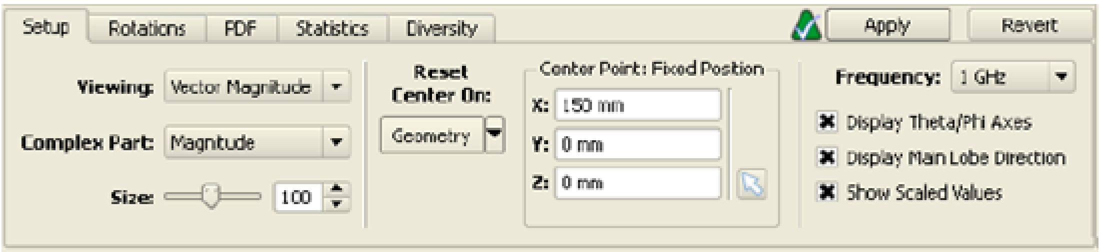
- **Sequence Axis** - Select the axis along which to view the sequence (from the *Sequence* tab).
- **Display Mode** - The available display modes are *Vector Field*, *Point Cloud*, *Flat*, and *3D*.
- **Size Factor** - Used to scale the size for point clouds and vector display modes
- **Displayed Field** - Lists the available components of a vector field that can be shown in a plot.
- **Decimation** - Used to sample a subset of points for solid sensors
- **Surface Resolution** - Used to sample a subset of points for Surface sensors. This control is only available if the Surface sensor sampling method is configured as *Field Interpolation*.
- **Complex Part** - Used to single out real/imaginary/magnitude values for single-frequency data
- **Axis Ranges** - Set the independent axis during the sequence
    - **Axis** - defines the axis
    - **Full** - runs along the full range of the axis
    - **Min And Max** - subsets non-sequence axes
- **Enable Scaling** - enable the scaling factor

The Setup tab for a solid sensor configuration

Far-Zone Sensor Configuration

- In the first section, set up your viewing preferences.
    - **Viewing** - choose among the available result types
    - **Complex Part** - view the *Magnitude*, *Real*, or *Imaginary* part of the results.
    - **Size** - enter the size of the maximum radius.
- In the second section, choose the orientation of the *Center* point.
    - Enter the (***X***, ***Y***, ***Z***) coordinates of the center, or
    - Choose the center using the *Pick* tool.
    - The *Reset Center On* drop-down enables you to center the plot at the center of the
      geometry or at the location of any of the ports in your active project.
- Select the following viewing options:
    - *Display Axes Theta/Phi*
    - *Display Main Lobe Direction* (the direction of maximum gain)
    - *Show Scaled Values*

The Setup tab for a far-zone sensor configuration

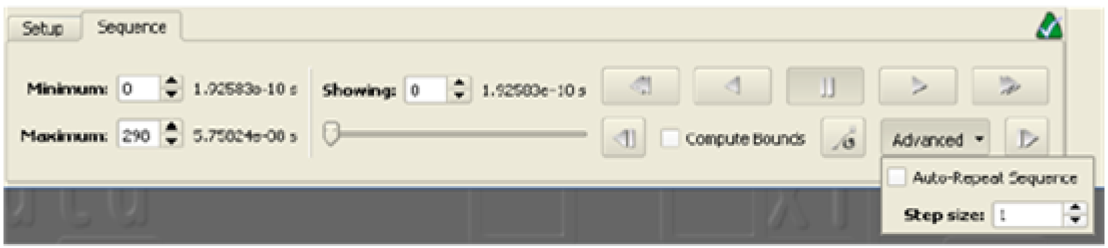


The Sequence Tab

This tab is available when viewing an SAR, Solid, or Volume sensor. Configure the settings here to "play through" a simulation.

- **Minimum/Maximum** - values entered here set limits on the range of the sequence axis.
- **Showing** - shows the currently plotted value of the sequence axis.
- The scroll bar at the bottom of the window can be used to view any point along the sequence.
- The simulation buttons enable you to move through the sequence:
  - The *forward* and *back* buttons play the sequence forwards and backwards, respectively.
  - The *pause* button pauses a playing sequence.
  - The *fast forward* and *rewind* buttons fast forward and rewind the sequence, respectively.
  - The *jump forward* and *jump backward* buttons jump to the end or beginning of the sequence, respectively.
- **Compute Bounds** - when this box is checked, EMPro will compute the bounds during the sequence.
- - performs the **Compute Bounds** calculation
- The **Advanced** tab
  - Checking the **Auto-Repeat Sequence** box will automatically replay the sequence continuously.
  - The **Step Size** determines the number of indices to change for the simulation buttons.

The Sequence tab of the field editing toolbar



The Rotations Tab

The Rotations tab, shown in Figure 13.20, oers several operations to adjust the orientation of the far-zone results by setting the Up Vector of the view. Changing the Up Vector (which is the Z-axis by default), will set the reference point for the spherical coordinate

systems (e.g., theta/phi, alpha/epsilon, etc.) This affects field plots of single polarization components, partial power effciency and power computations, mean eective gain computations, and antenna diversity. This is useful in
cases where the geometry is aligned with an axis of the computation grid, but the real-world position of the geometry is not axis-aligned.

- The *Up Vector Presets* list enables you to select a pre-defined *Up Vector* in the ***X***-, ***Y***-, or ***Z*** -direction.
- The *Pick Up Vector* tool enables you to click and drag within the simulation space to create your own *Up Vector*.
- The *Undo* button will undo an operation.
- The *Redo* button will redo an operation.
- The *Reset* button will reset the *Up Vector*.

The Rotations tab of the field editing toolbar

The PDF Tab

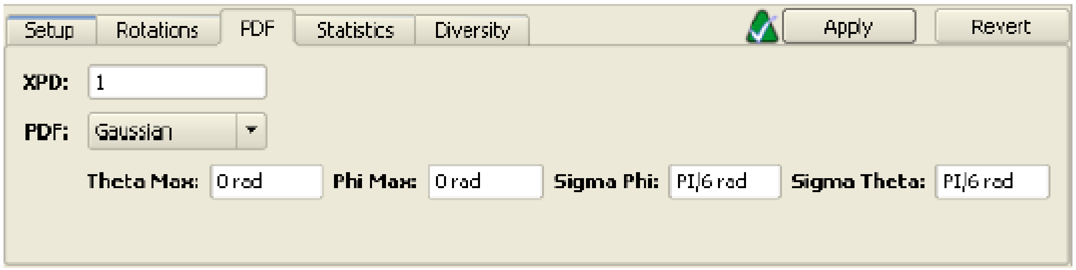
The PDF tab controls the settings used in the computation of mean eective gain and antenna diversity far zone results. The following illustration shows the PDF tab.

- **XPD** is cross-polarization discrimination of the incident multipath field. It can be expressed by the ratio of time-averaged vertical power to time-averaged horizontal power in the fading environment. Users can input a linear value or a decibel value.
- **PDF**, or Probability Density Function, is used to model communication channels. This function "**weights**" different directions in the far zone sphere such that certain far zone direction are taken into account more than others in the mean effective gain and antenna diversity calculations. It has a maximum value of one and a minimum value of zero. EMPro provides the following choices for PDF's:
  - **Uniform Value:** in this case, PDF is simply one.
  - **Gaussian:** this is most commonly used as a model of channels. *Theta Max* and *Phi Max* are the angles of the maximum incoming field. *Sigma Phi* and *Sigma Theta* are the standard deviations for the ϕ and θ directions, respectively.
  - **User-Defined:** this enables you to import a PDF from a text file. The file must contain a grid of theta/phi angles and the corresponding probability density at each angle. For example, to define the PDF at ten theta angles with twenty phi angles, the text file data would look something like the following:

```
theta0radians phi0radians pdfAtTheta0Phi0
theta0radians phi1radians pdfAtTheta0Phi1
```

```
...
theta0radians phi19radians pdfAtTheta0Phi19
theta1radians phi0radians pdfAtTheta1Phi0
...
theta9radians phi19radians pdfAtTheta9Phi19
```

The PDF tab in Gaussian mode



The Statistics Tab

There are two main configurations for the Statistics tab, depending on the type of sensor you are evaluating.

Note
Keep in mind that not all fields listed will be available for every view.

Far-Zone Sensor Statistics

The next figure shows the *Statistics* tab for a far-zone sensor configuration.

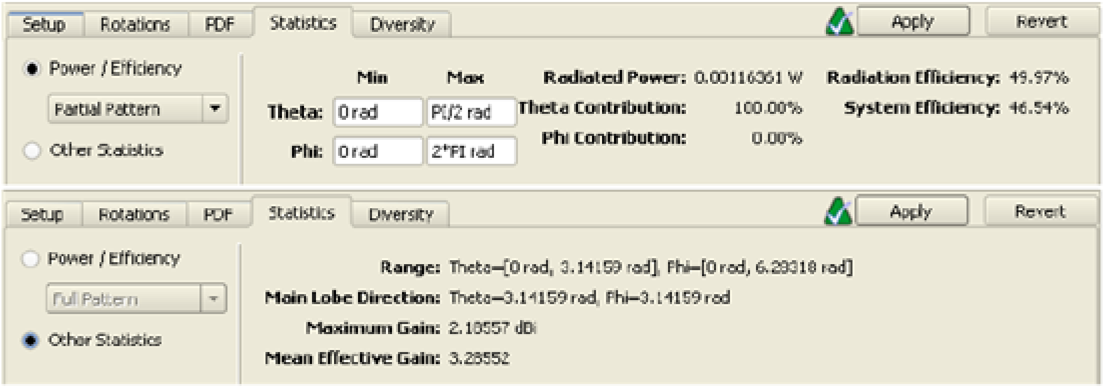
When selected, *Power/Efficiency* gives the user the option of choosing from among the following radiation patterns:

- Full Pattern - specifies the use of a full far zone sphere.
- Upper Hemisphere - specifies the use of theta = [0, 90°] and phi = [0, 360°].
- Open Sky - specifies the use of theta = [0, 80°] and phi = [0, 360°].
- Partial Pattern - enables you to specify an arbitrary solid angle. These angles are defined relative to the coordinate system defined under *Rotations*.

Each pattern will display its statistics to the right. In the case of *Partial Pattern*, the user can redefine the min and max values of *Phi* and *Theta* by typing them in to the corresponding boxes. The statistics will update accordingly.

The *Other Statistics* option presents a list of additional statistics.

The Statistics tab for a Far-Zone sensor, showing Power/Efficiency and Other



statistics

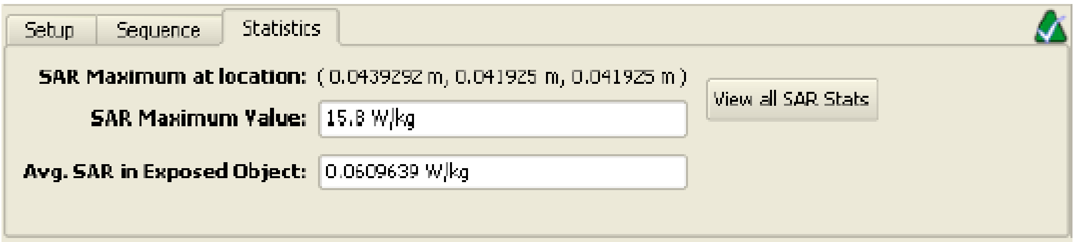
SAR Sensor Statistics

The next illustration shows the *Statistics* tab for an SAR sensor configuration.
This tab is available when looking at SAR data for both the *SAR Sensor* and the *SAR Averaging Sensor*. The following figure displays the dialog that appears when the *View All SAR Stats* button is pressed.

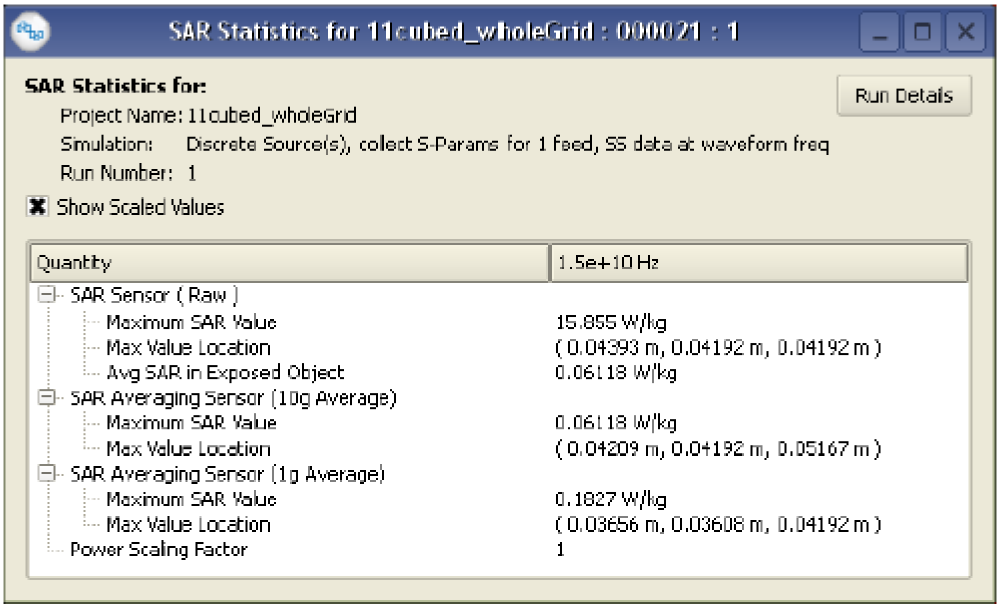
- The *SAR Maximum Value* displays a max SAR value dependent on the properties of the SAR Sensor, (i.e., the maximum 1-gram Averaged SAR value, the maximum 10-gram Averaged SAR value, or the maximum raw SAR value). A value entered here affects the power scaling for this run and frequency, in the same manner as adjusting the *Power Scaling Factor* in the *System Sensor Output* dialog.
- *SAR Maximum at Location* displays the global coordinates of the location where the SAR maximum value is found.
- The *Avg. SAR in Exposed Object* field is available when looking at raw SAR sensor data only. It displays the whole body average SAR value. An adjustment entered here aects the power scaling in the same way as adjusting the *SAR Maximum Value*.
- The *View Slice With Max Value* button takes you to the planar slice of SAR data which contains the maximum value.
- The *View All SAR Stats* button opens a separate window displaying the statistics for the Raw SAR, 1g Averaged SAR, and 10g Averaged SAR together in a table. Through this window, the scaling factor can be adjusted, which affects all SAR values in this window, including data displayed in the SAR Sensor Statistics tab. The scaling factor and SAR statistics values will also automatically update when the user manually changes (rescales) any other statistics available through the *Results* workspace window, such as power and efficiency values.

Note For more information on rescaling values in the *System Sensor Output* dialog, refer to Rescaling.

The Statistics tab for an SAR sensor



The complete list of SAR statistics



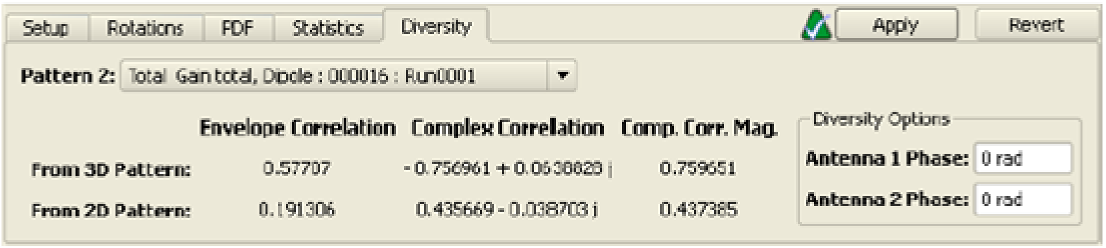
The Diversity Tab

The *Diversity* tab is used to compute antenna diversity metrics between two far zone patterns. To perform the diversity computations, you must load the data for both far zone patterns in the *Results* workspace. Then go to the diversity tab for one of the two patterns. The following figure shows the *Diversity* tab.

- The *Pattern 2* drop-down list enables you to select the far zone result which your current far zone result is compared against. The only choices listed here are results which have been loaded into the *Results* workspace window.
- The statistics next to *From 3D Pattern* show the diversity computation for the full far zone sphere. The statistics next to *From 2D Pattern* show the diversity computation at theta = 90^o.
- Under *Diversity Options*, you can set the *Antenna 1 Phase* and *Antenna 2 Phase* controls to specify the phase difference between the two far zone patterns.

> **Note**
> Note that the *XPD* and *PDF* settings under the *PDF* tab impact this computation, as well as the settings under the *Rotations* tab.

The Diversity tab of the Field editing toolbar



The Hearing Aid Compatibility Tab

This tab, as the name suggests, is only available when viewing (HAC) sensors.

- **Frequency** - Select the frequency of interest for the hearing aid.
- **Display Mode** - The available display modes are *Point Cloud* and *Flat*.
- **Show Scaled Values** - When checked, the displayed results are scaled by the scaling factor entered by the user on the system sensor output table.
- *Surface Resolution* - This option appears whenever there is a HAC or interpolated surface sensor. It controls the resolution at which field data is shown for the sensor. The higher the resolution setting, the more data points are shown on screen (and the closer they are together).
- *M-Rating* - Displays the M-rating of the sensor, or the suitability of the hearing aid used with a wireless device. The rating range is 1-4, with 4 being the best rating.
- *Threshold* - This is the maximum allowed field value given the current M-rating.
- *Band* - Displays the band type. Under the *Edit* button, you can configure the properties of the band, or define a pre-set list of properties using the *Manage Presets* button. An editor dialog will appear where you can add a new preset or import presets from a text file. Each line in the file must follow this format:
    - Lines in the file starting with **#** or **!** are comments and are ignored.
    - Otherwise, the line must contain six semicolon-delimited items in this order: name; waveform modulation factor; E-field probe modulation factor; H-field probe modulation factor; articulation weighting factor, in dB (must be either 0 or -5); and a boolean value which determines whether RMS conversion should be used (1 to enable, 0 to disable).

  An example of this format is as follows:

  ```
  ! This is a comment
  Band 1 Name;0.85;0.9;0.8;-5; 0
  Band 2 Name; 1.0; 0.92; 0.33; 0; 1
  ```

- The 3x3 Grid Region - This shows the maximum field values in each of the 9 squares of the HAC grid. The double-line on the top-level of the readout helps you orient in reference to the white grid on the displayed field (it also has a double-line in the top-left corner). Three of the cells in the 3x3 grid will have a dotted red border around them. These are the three cells chosen to be excluded from the HAC rating computation as allowed by the IEEE HAC standard. Also, any cells with red text are cells which exceed the field value threshold amount for the current M-rating.

The Hearing Aid Compatibility tab of the field editing toolbar

Post-Processing

Post-processing gives the user the option of running additional computations after the calculation engine run is finished to save certain results to disk. Two types of post-processing are available.

Far Zone Post-Processing

Far-zone post-processing is enabled by checking the *Save Data for Post-simulation Far Zone Steady-state Processing* box under the *Frequencies of Interest: Data Storage* tab of the *Simulations* workspace window. When this configuration is set, there will be an entry for *Raw Steady-State Far Zone Data* in the *Results* workspace window. Double-clicking this entry will enable you define a new far zone geometry at which the steady-state far zone pattern will be computed. Edit the geometry exactly as you would edit a far zone sensor. When you click **Done** in the far zone sensor editor, the computation of the far zone pattern will begin. Once complete, you can find a new result in the *Results* workspace window which corresponds to the newly defined far zone pattern.

SAR Post-Processing

Right-clicking on any Raw SAR Sensor entry in the *Results* workspace window will show an option in the context menu to *Postprocess Results*. This enables you to perform SAR averaging. Edit the geometry and averaging parameters exactly as you would in the *SAR Averaging Sensor* editor. Once you click **Done** in the editor, you will be given the option to automatically run the post-processor. If you say yes, the post-processing operation is queued as if you had created a new simulation. Otherwise, you will have to do the post-processing run manually. Once the SAR averaging is complete, new SAR average data will appear in the *Results* workspace window.

Additional Tools for Customizing and Organizing Projects

In this section, you will learn how to:

- Use scripts to streamline tasks in EMPro
- Define your own universal parameters
- Store object definitions externally to reuse them in multiple projects
- Organize objects in your EMPro project for more convenient access

EMPro provides several tools to facilitate the creation and organization of projects.

- Parameterization is a powerful tool that makes it easy to define variables and functions in one convenient workspace window, which can be referenced anywhere within the EMPro interface. Additionally, Parameterization can be used in conjunction with scripting to sweep through a series of parameters (i.e., multiple antenna lengths) to run a calculation at every swept point. In EMPro, parameters are defined in the *Parameters* workspace window.

- The *Scripting* workspace window makes it convenient to write scripts to accomplish tasks that are specific to an EMPro project. It provides users with the ability to create fully-customizable functionality within the EMPro interface that is specific to their own tasks. Scripting may be useful for quickly performing repetitive tasks, referencing external files, employing a series of modeling operations at once, or virtually any task that is tedious with the standard EMPro tools.

- Libraries provide useful means of storing definitions and any types of objects created within an EMPro project. They are saved in the *Libraries* workspace window as files that are not attached to a specific project so that they can be referenced again and again. They are very useful for creating new projects that reuse definitions and objects from past projects. Rather than having to rebuild a project from scratch, pertinent libraries can be imported so that time is not wasted redefining similar objects and properties.

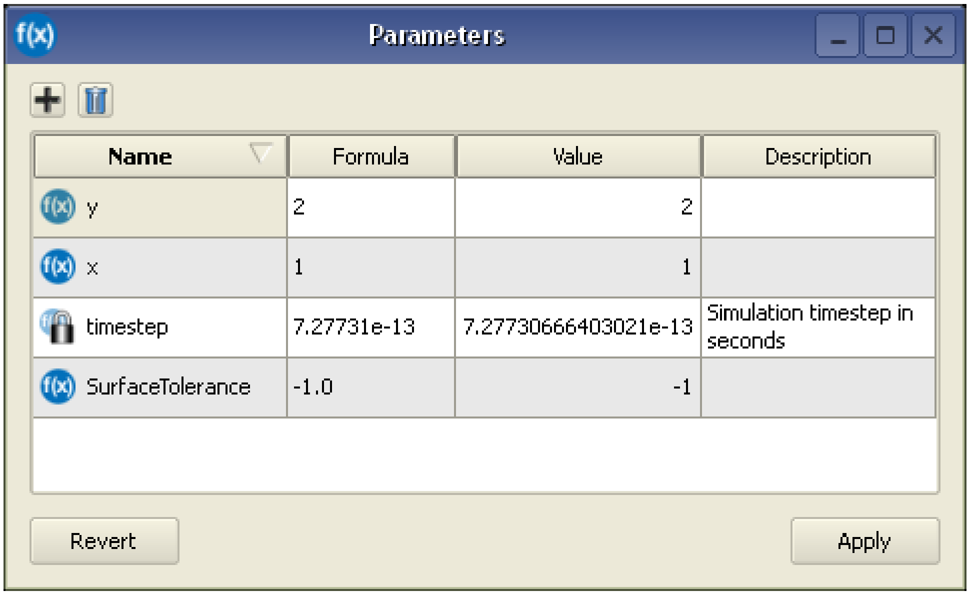
- EMPro also provides various features that help in grouping and organizing definitions and objects. The *Groups* branch of the *Project Tree* functions to store **shortcut groups**, which are groups of objects that are added and organized by the user. Similarly, *Assemblies* are user-defined groups of geometric objects that are added to the *Parts* branch of the Project Tree. They are convenient, especially for projects that contain a large number of parts, so that objects in the tree remain organized and easy to access.

Parameters Workspace Window

The *Parameters* workspace window enables you to define parameters that can be used in other places in the project to parameterize their project. A parameter can be referenced in any dialog box in EMPro. It can have a simple numeric value or a mathematical expression, and can reference other parameters.

The *Parameters* workspace window can be accessed at any time by right-clicking in the tabbed workspace or selecting the window in the View menu of the Application Menu Bar.

The Parameters workspace window



Defining Parameters

The *Parameters* workspace window contains four fields: *Name, Formula, Value* and *Description*. Note that a value named 'timestep' is already present upon opening this window.

A new parameter is added by clicking the button above the table of parameters. A new line with default values will be added to the list of parameters. These values can edited by double-clicking on any value. The |**Tab**| key will scroll through the columns of the table and the |**Escape**| key will cancel any changes that have been made to a parameter entry.

The *Formula* column is where a user will input a mathematical formula or a simple numeric value that will define the value of a given parameter. This formula can reference other parameters that have already been defined.

The *Value* column is a read-only column that displays the evaluated value of the parameter. If an invalid formula is entered, an error message will appear within this field with a description of the invalidity. Simply hold the mouse over the error message to view this description.

Similarly, a parameter is deleted by selecting the unwanted parameter and clicking the button above the table. If a parameter is deleted that is referenced within another parameter's definition, an error message will appear since the parameter that is referenced is no longer defined.

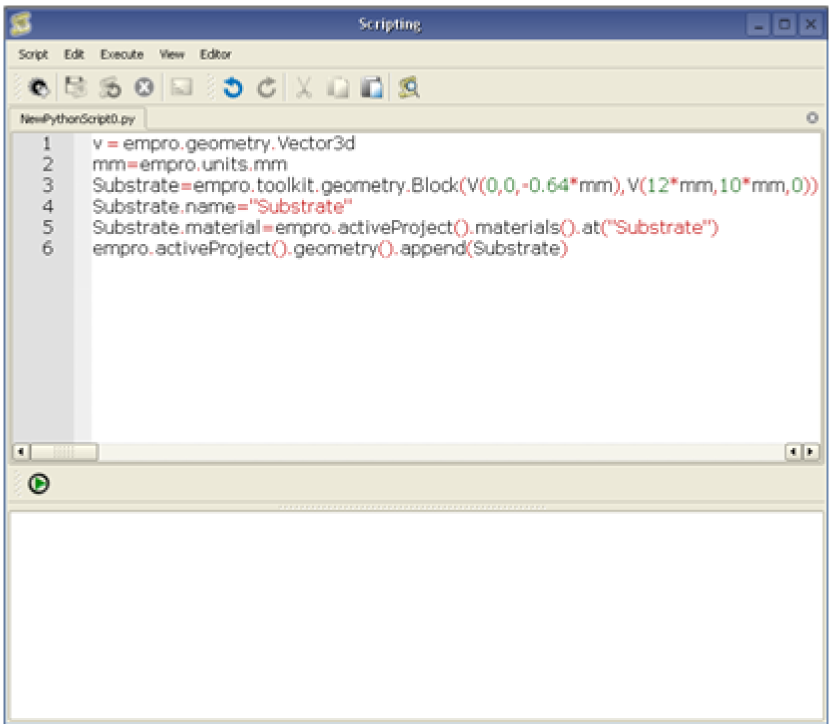
Each parameter is referenced by its assigned name defined in the *Name* column of this window. For instance, a parameter named "*length*" can be referenced at any time by typing "*length*" into any dialog box within EMPro, and it will assume this defined value.

> **Note**
> A sweep over a parameter may be set up in the *Simulations* workspace window. For more information, refer to Simulations Workspace Window.

Scripting Workspace Window

The *Scripting* workspace window enables you to create, edit, manage, and execute user-defined scripts, which are capable of gathering and reporting information from the EMPro project or making changes to the project. Scripts are blocks of Python scripts that use EMPro as module. They are typically used to automate repetitive or tedious tasks (that could otherwise be done through the EMPro GUI) with greater speed and precision.

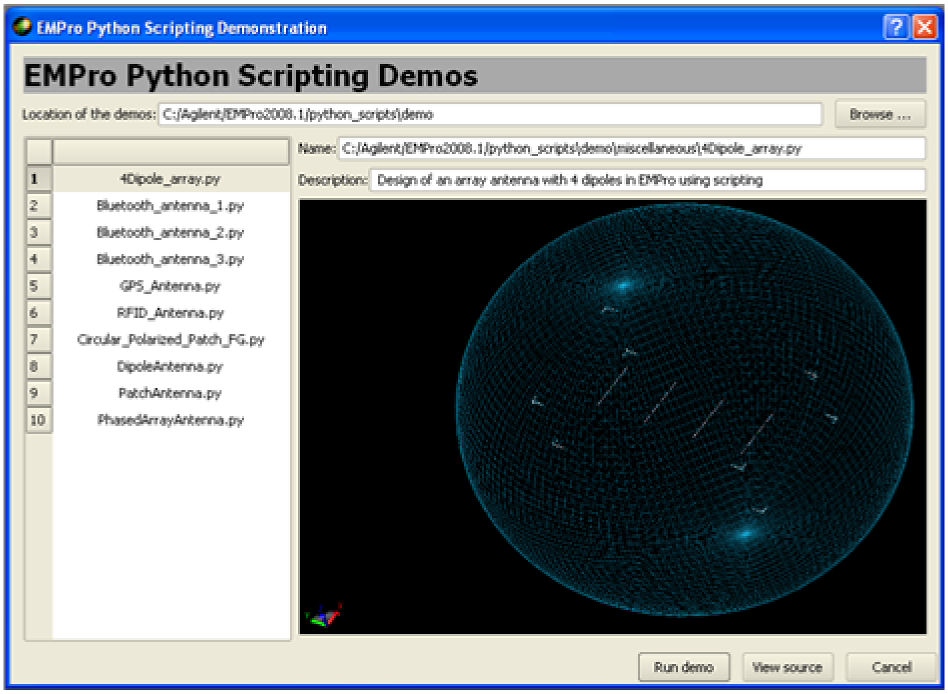
The Scripting workspace window



Parameters that have been defined in the active parameter list of the *Parameters* workspace window may be referenced within any script. A consequence of this ability is that changing a parameter in the parameter list may change the behavior of user-defined functions.

There are a dozen of example scripts included. You can launch the demo by using the menu item Help > Scripting Demo.

Demo Scripting Window



EMPro Scripting Language

All scripts in EMPro are written in the Python scripting language.

> **Note**
> For more information on Python, visit the Python Software Foundation homepage at http://www.python.org or use the Help functionality within EMPro.

New Python Script

Select this icon to bring up a *New Python Script* tab where a new *Script* can be defined. The script is not executed until the *Execute Python* icon is pressed.

Commit Script

Select this icon to commit a change to a script after an edit.

Revert

Select this icon to "revert" or abandon an edit. Any changes that have been made in the editor will be lost, and the editor will revert back to the script that was stored in the EMPro project or application.

Clear Output Window

Select this icon to clear the text in the output window where error messages and script output are written.

Search and Replace

Select this icon to search for text within scripts. By selecting the *Replace With* check-box, text can be located and replaced with the desired text.

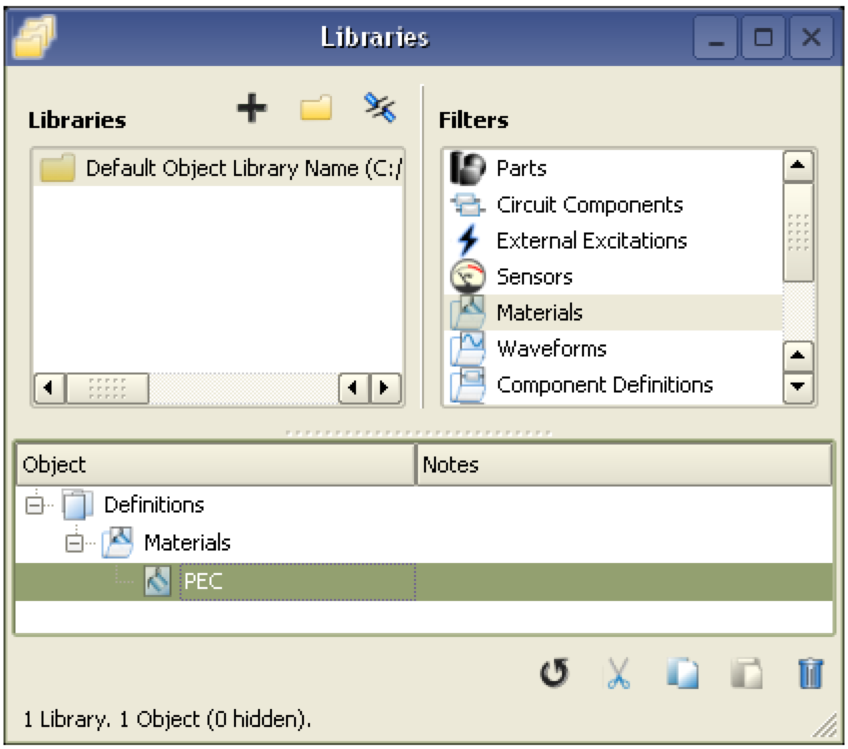
Execute Script

Select this icon to execute a macro from one of the user-defined scripts.

Libraries Workspace Window

The *Libraries* workspace window enables you to create libraries or collections of objects grouped by category so that they can be easily referenced during a project and can be accessed in subsequent projects. This makes it very easy to access commonly used objects and definitions so that they do not have to be recreated during every project.

The Libraries workspace window



Creating a New Library

To create a new library directory or subdirectory, click on the button above the *Libraries* space in the workspace window. Specify the name of the location to store the new library file.

Accessing Existing Libraries

To access an existing library, click on the button, navigate to the appropriate directory, and select the desired library to load into the project.

Adding Objects and/or Definitions to a Library

To add objects of definitions to a library, simply drag the desired object from the *Project Tree* into lower workspace entitled *Object/Notes*. The object will be placed in a library grouping that corresponds to its original position in the Project Tree. For example, a *Material* dragged from the Project Tree will be placed in the *Materials* folder of the Library. Additionally, filters can be applied to library objects to control the visibility of each group. They are controlled in the *Filters* section of the *Libraries* workspace window.

Appendix of Geometric Modeling

In this appendix, you will learn how to use the geometric modeling tools available within EMPro.

Editing Cross-Sections for 2-D and 3-D Models

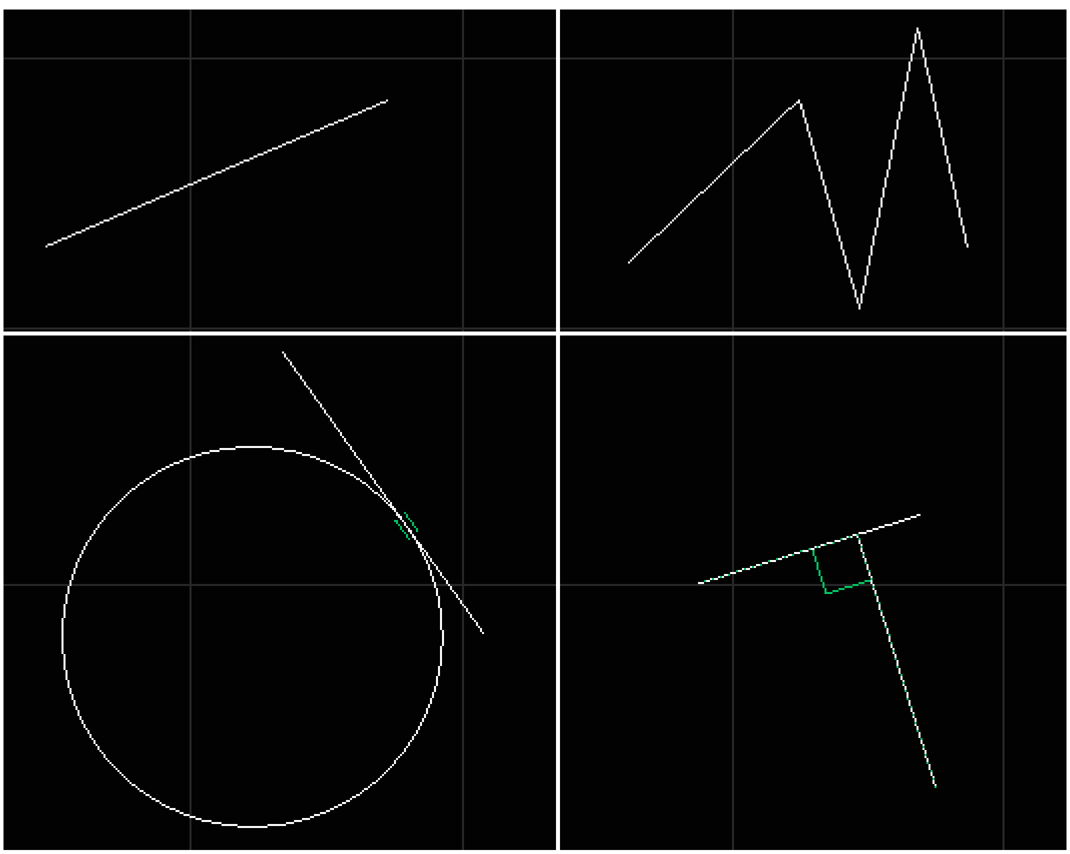
Shapes

Edge tools

Edge tools are used to create lines of various shapes within the EMPro interface. The following figure displays the Edge Tools including the *Straight Edge* tool (upper left), *Polyline Edge* tool (upper right), *Tangent Line* tool (lower left) and *Perpendicular Edge* tool (lower right).

> **Note**
> Pressing **|Tab|** while using these tools will bring up the *Specify Position* dialog, which is used to enter relevant properties to the tool being used.

The Edge Tools



Straight Edge

Creates a simple straight edge. To use this tool, click the *Straight Edge* button and click two points in the sketching plane where the endpoints should be located.

Polyline Edge

The *Polyline Edge* is similar to the *Straight Edge* tool except it allows multiple points to create a series of connected straight edges. Click a starting point in the sketching plane and continue clicking on the locations of subsequent endpoints to create desired polyline edge. Click on the first vertex or press |**Return**| to finish.

Perpendicular Edge

Creates a straight edge perpendicular to an existing edge. To use, select the *Perpendicular Edge* button and click on the existing edge that will define the perpendicular direction. This can be a straight or curved edge. Then click on the location of the first and second endpoints of the perpendicular straight edge.

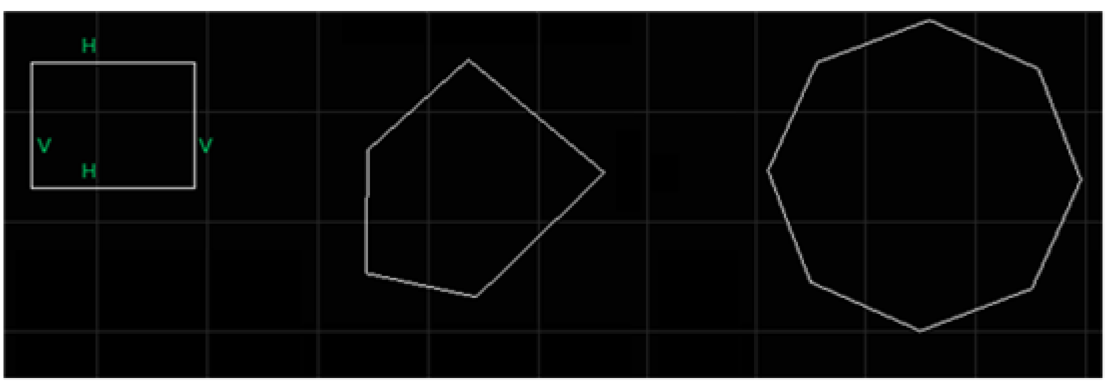
Tangent Line

Similar to the Perpendicular Edge tool, but instead draws a line *tangent* to a pre-existing, non-linear edge. To use, select the *Tangent Line* tool, and click on the existing curve that will define the tangential direction. Then click on the location of the first and second endpoints of the tangential straight edge.

Closed Polygon Tools

The following illustration displays the *Closed Polygon* tools including the *Rectangle, Polygon* and *N-Sided Polygon* tools.

The Closed Polygon tools



Rectangle

Creates a simple rectangle. Click the desired location of the first vertex of the rectangle and drag the mouse to the location of the second vertex.

Polygon

Creates a polygon specified by the user. (For regular polygons, see *N-Sided Polygon*). It functions like the *Polyline Edge* tool. Click the starting point and all subsequent points, then press |**Return**| to close the polygon. This will draw a line from the last selected endpoint to the first endpoint.

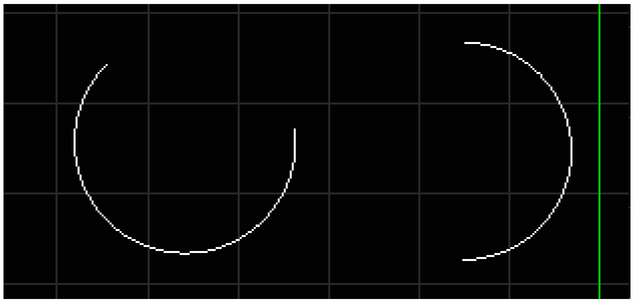
N-Sided Polygon

Creates a regular, N-Sided Polygon of a user-specified number of sides. Click the location of the center of the polygon. Then press the left-bracket key '[', or the right-bracket key ']' to decrease or increase the number of sides, respectively. Once the correct number of sides is selected, drag the mouse until the desired size and orientation around the center point is achieved and click again to finish the N-sided polygon.

Arc Tools

The figure below displays two of the arc tools: the *3-Point Arc* and *2-Point Arc* tools.

The Arc tools



3-Point Arc Tool

Creates an open arc from three points. Click on the location of the first endpoint. Click a second location to specify a point between the two endpoints (which helps determine size), and a third location to specify the other endpoint.

2-point Arc Tool

Creates a semi-circle from two points. Click on the first endpoint location and drag the mouse until the desired semi-circle size and orientation is achieved. Click this second end point location to finish.

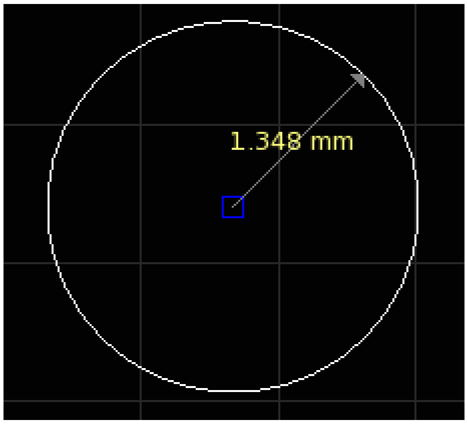
Arc center, 2 points Tool

Creates an open arc from three points. First, click on the location of the center of the arc. Secondly, click a point to specify the radius of the arc. Finally, click the location of the endpoint to specify the length of the arc.

Circle and Ellipse Tools

The following figure displays an example of a circle drawn with the *Circle Center, Radius* tool.

A circle drawn with Circle Center, Radius tool



Circle Center, Radius

Creates a circle defined by its center point and radius. Click the location of the circle's center point, then select another point to define the radius and finish the circle.

3-point Circle

Creates a circle based on three user-specified points, similar to the *3-Point Arc* tool. Click the first two points to set the location of the circle and the third to specify its size.

2-point Circle

Creates a circle based on the distance between two points. After selecting the first point, choose the second to define the diameter and finish the circle.

Ellipse

Draws an ellipse from three points: the center and two perpendicular radii. Click the center point of the ellipse, then select the desired location of the first radii. Finally, select the desired length of the second radii, perpendicular to the first.

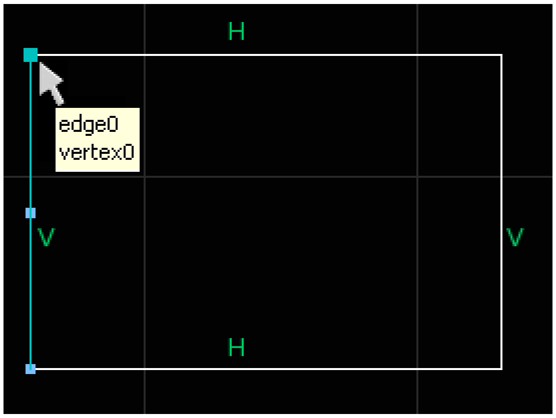
Tools

Select/Manipulate

Selects anything within the sketch. This is the default tool when no other tool is selected. It can be used to:

- Move an object, edge, or vertex to a new position, by clicking-and-dragging
- Select a vertex or edge and lock or edit its position, by right-clicking and selecting *Lock Position* or *Edit Position*.
- Edit the value of an angle or distance constraint, by right-clicking and selecting the edit option.
- Delete an edge or constraint, by right-clicking and selecting the delete option.

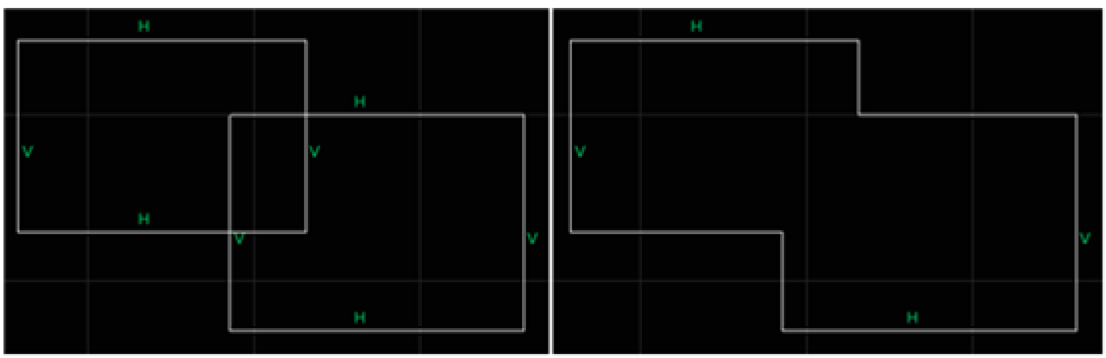
Select/Manipulate tool



Trim Curves

Deletes segments of curves until they intersect with other curves. To use this tool, click on the section of the curve that is to be deleted.

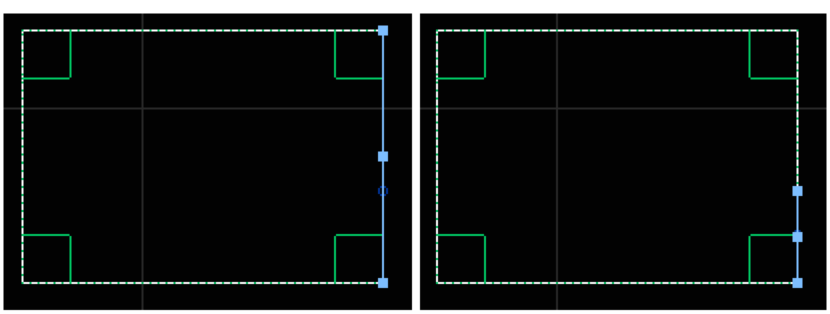
Trim Curves tool



Insert Vertex

Inserts a vertex onto an already existing edge. Click the desired location of the new vertex on the existing edge.

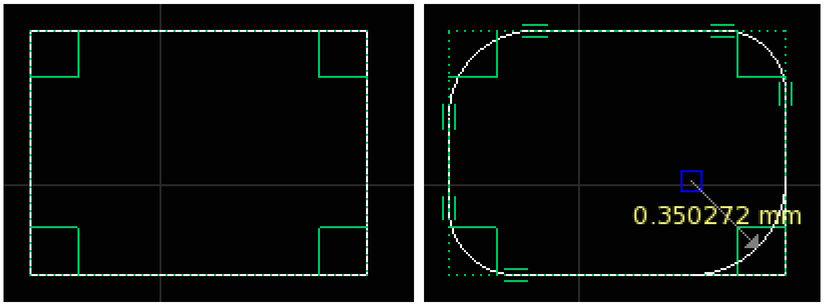
Insert Vertex tool



Fillet Vertex

Converts a sharp corner into a rounded corner between two curves. Click on any sharp corner and drag until the desired fillet radius is achieved and click to finalize fillet.

Fillet Vertex tool



Constraints

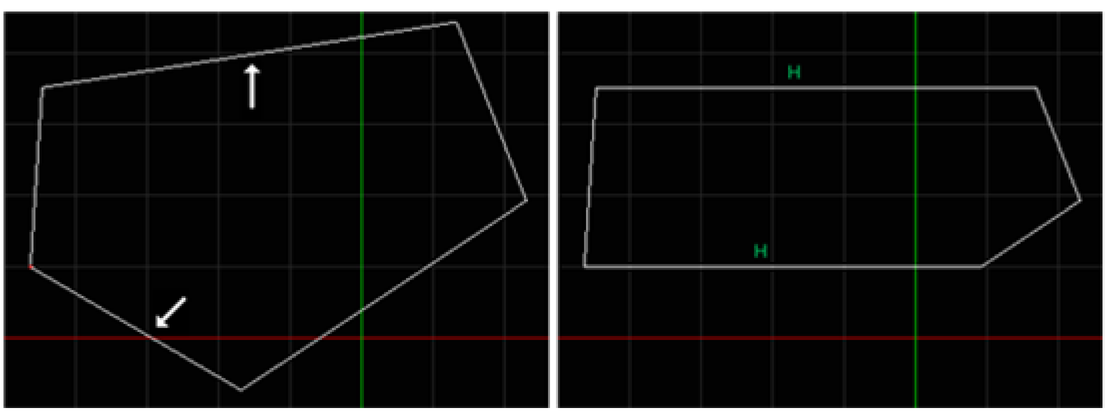
The geometry Constraints tools are used to modify pre-drawn shapes to the desired specifications.

> **Note**
> Some of the "before" images below have been marked with white arrows to show which edges are constrained in the "after" image on the right.

Horizontal Constraint

Constrains a segment to the horizontal direction.

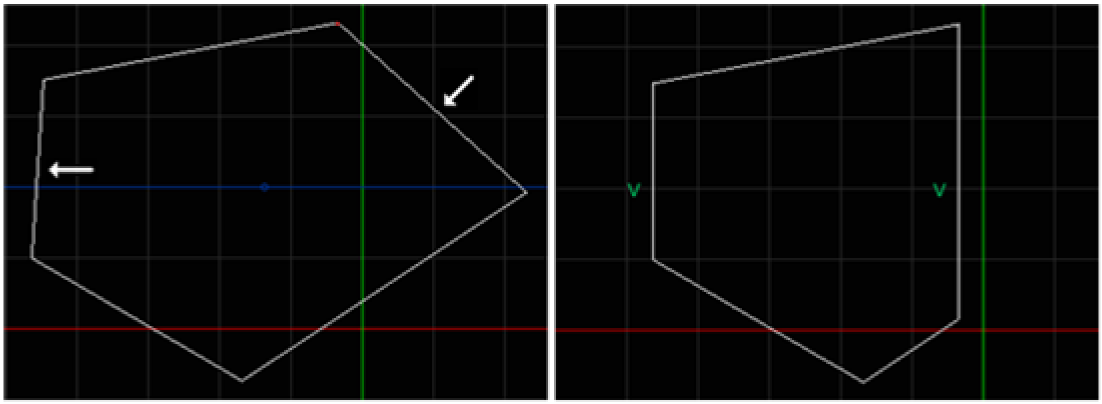
Polygon before (left) and after (right) two sides are constrained horizontally



Vertical Constraint

Constrains a segment to the vertical direction.

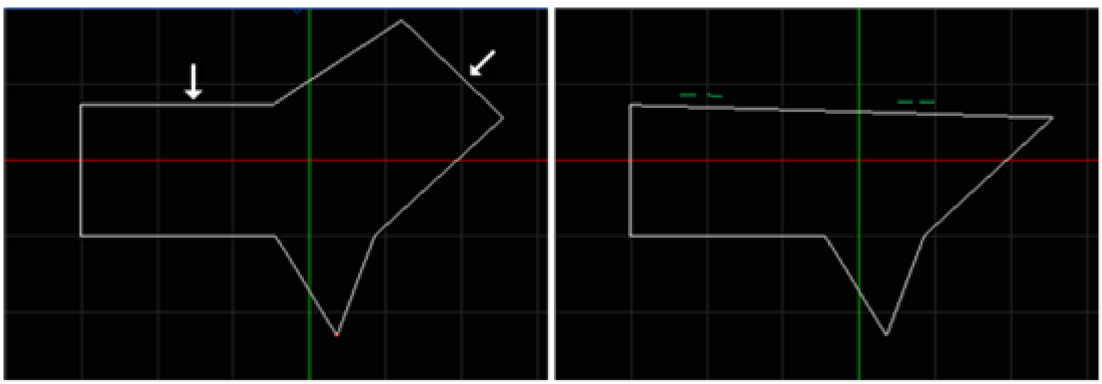
Polygon before (left) and after (right) two sides are vertically constrained



Collinear Constraint

Constrains two straight segments so that they are in line with each other.

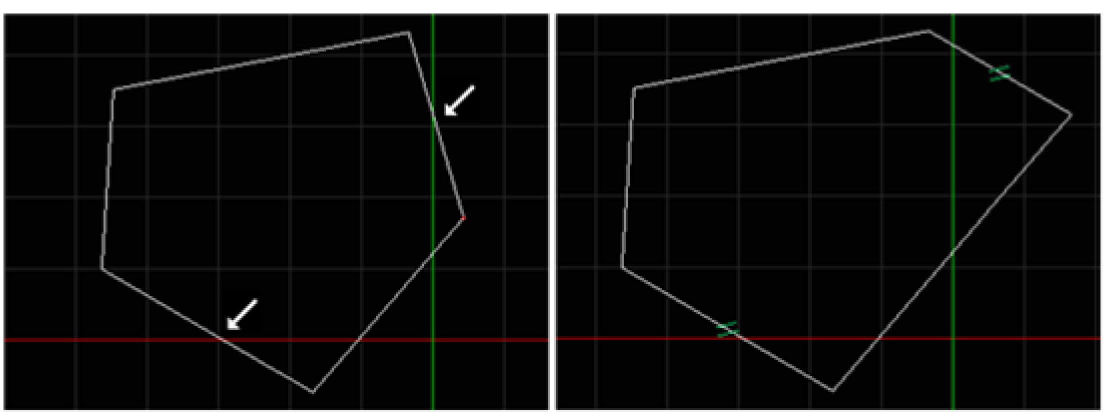
Polygon before (left) and after (right) after two sides are constrained to be collinear



Parallel Constraint

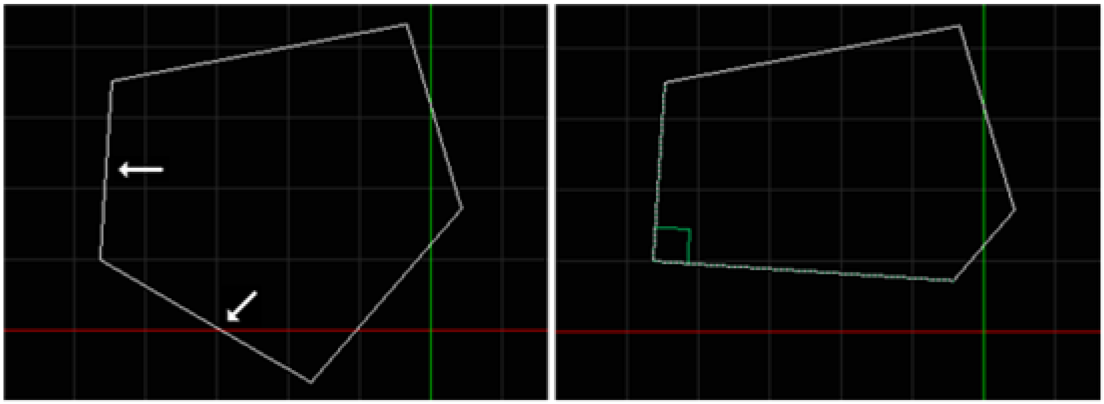
Constrains two straight segments so that they are parallel to each other.

Polygon before (left) and after (right) two sides are constrained in parallel



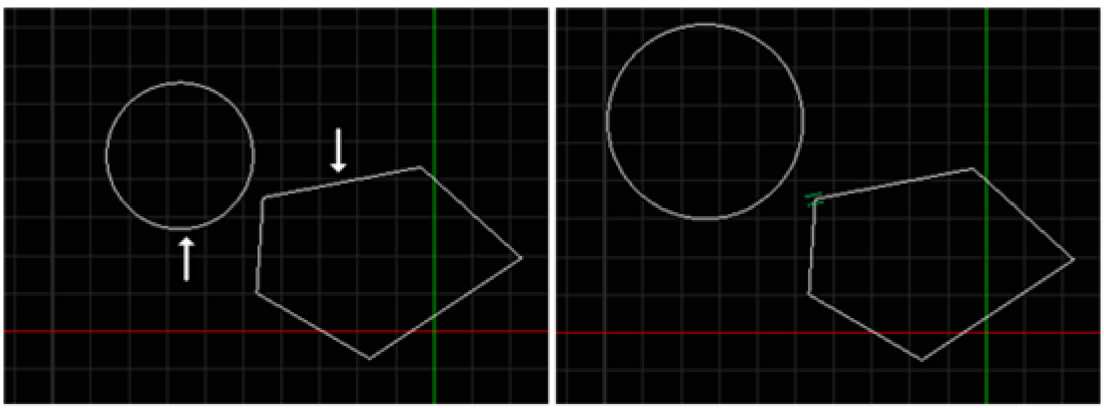
Perpendicular Constraint

Constrains two straight segments so that they are perpendicular to each other. The following figure displays a polygon before (left) and after (right) two sides are perpendicularly constrained



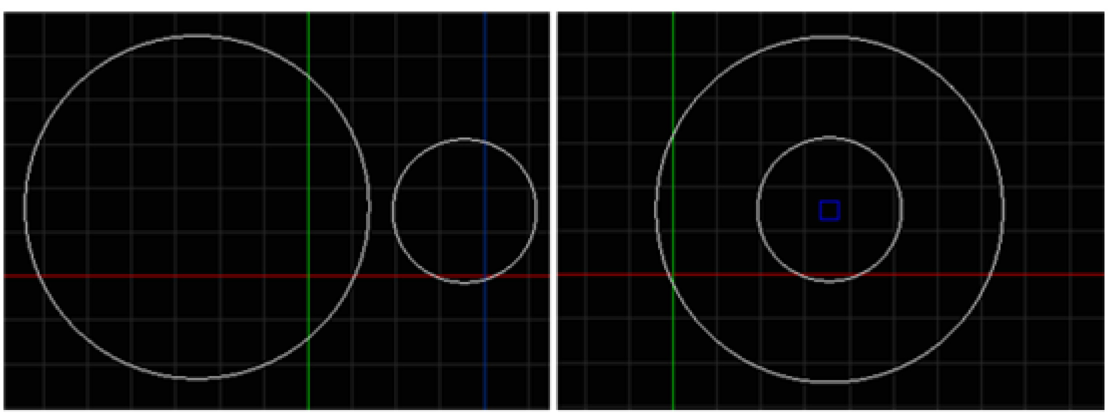
Tangent Constraint

Constrains a straight segment so that it is tangent to a circular segment at a point. In the following figure, Circle and polygon before (left) and after (right) a side of the polygon is constrained tangentially with reference to the circle:



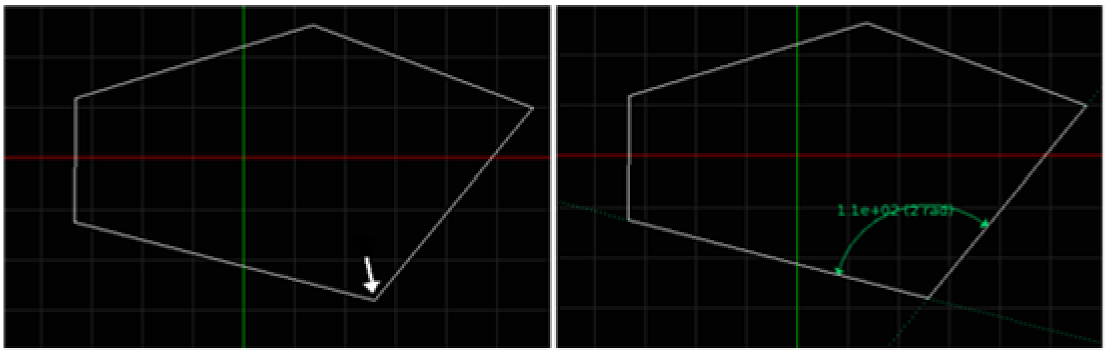
Concentric Constraint

Constrains two circular segments so that they are centered upon the same point. In the following figure, two circles before (left) and after (right) are made concentric:



Angle Constraint

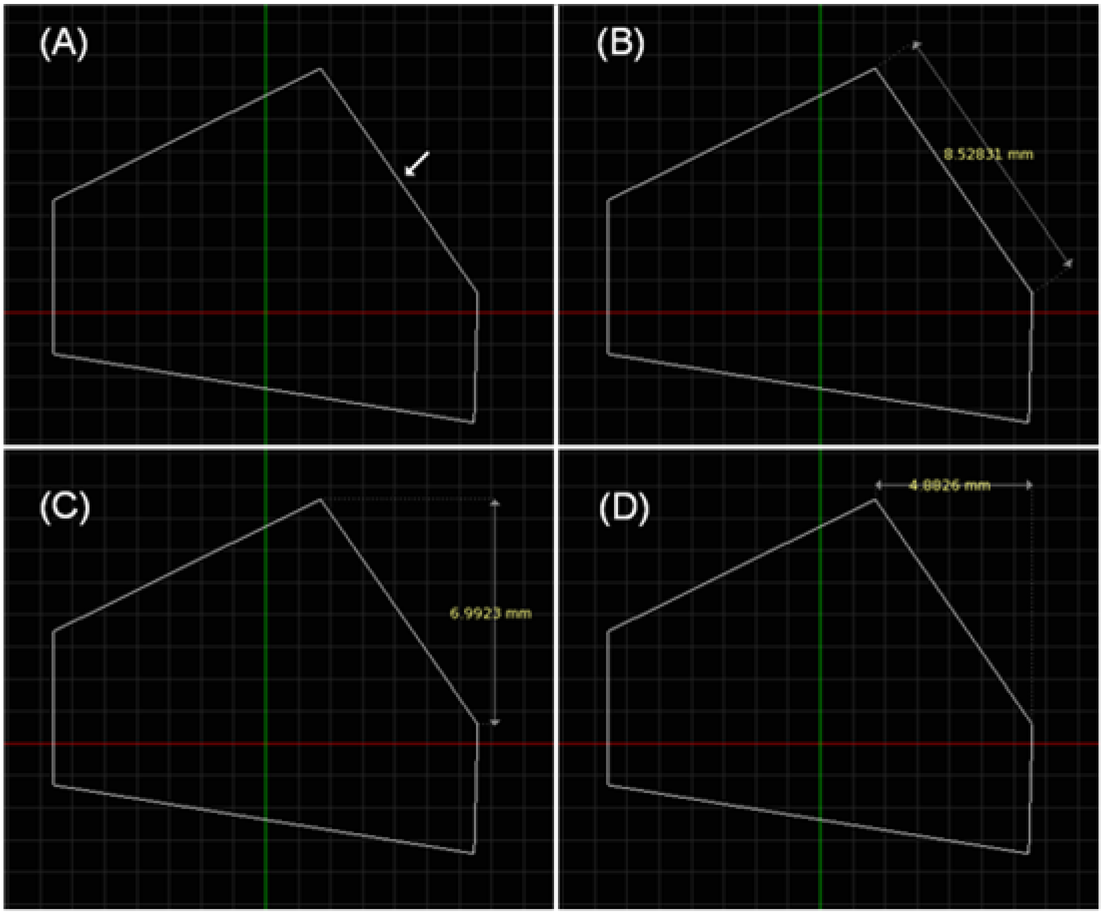
Constrains an angle to a user-specified value between two straight lines. Click once to select angle, then click a second time to place label and enter the angle size. In the figure below, the polygon before (left) and after (right) an angle has been constrained to a user-defined value.



Distance Constraint

Constrains the distance between two points, the distance between a point and a line, or the length of a line to a user-specified value. After selecting the object(s) to constrain, click a final time to place label and enter distance.

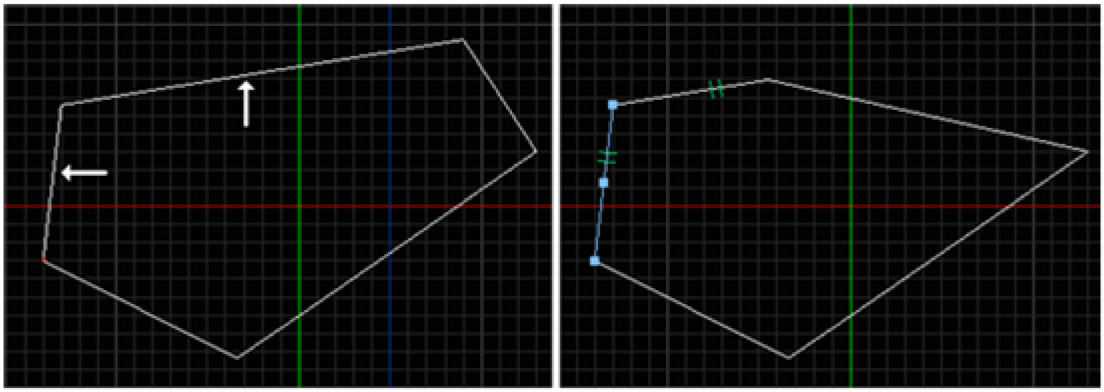
As shown in the figure below, there are three different constraint **"modes"**: parallel, vertical and horizontal. The mode is determined by the location of the mouse cursor when you click to specify where the constraint should be drawn.



The polygon (A) before line has been constrained, (B) with a parallel distance constraint, (C) with a vertical distance constraint and (D) with a horizontal distance constraint.

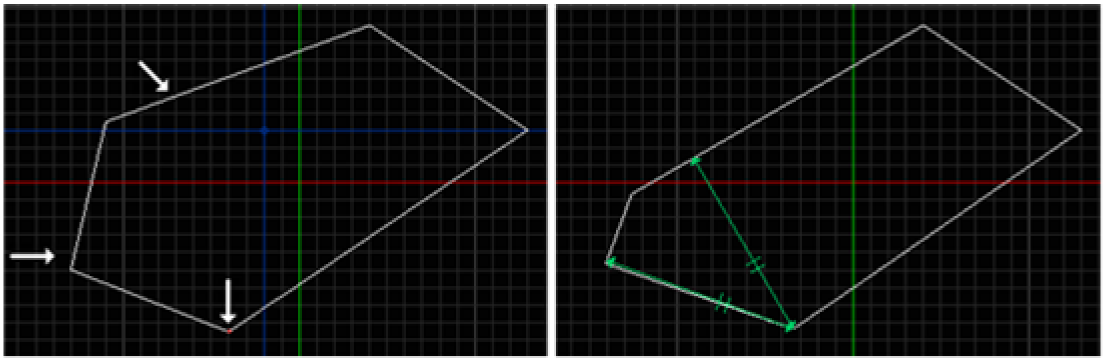
Equal Length Constraint

Constrains selected segments to an equal length (assumes the length of the segment selected second). Polygon before (left) and after (right) two sides are made equal length to one another.



Equal Distance Constraint

Constrains two pairs of points so that each pair assumes a distance from each other equal to the distance between the original pair.In the following figure, polygon before (left) and after (right) two sides are made equal distance from each other:

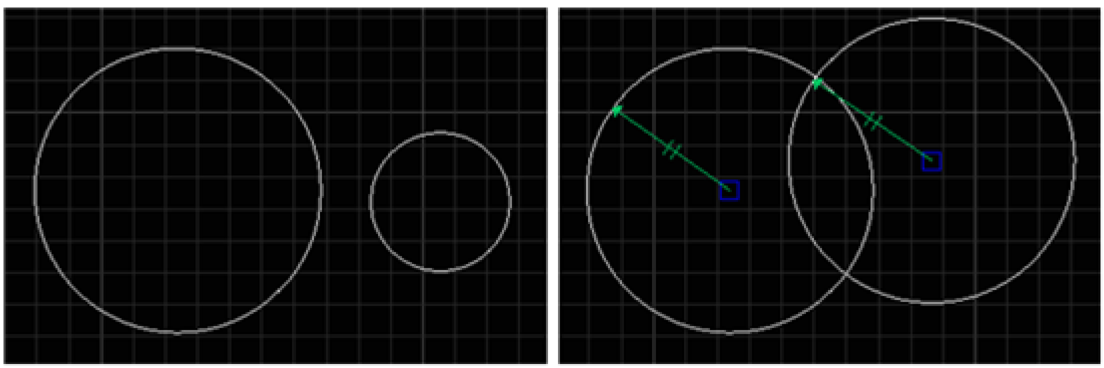


Radius Constraint

Constrains the radius to a user-specified value.

Equal Radius Constraint

Constrains selected radii to an equal length. In the following figure, two Circles before (left) and after (right) their radii are made equal:



Snapping

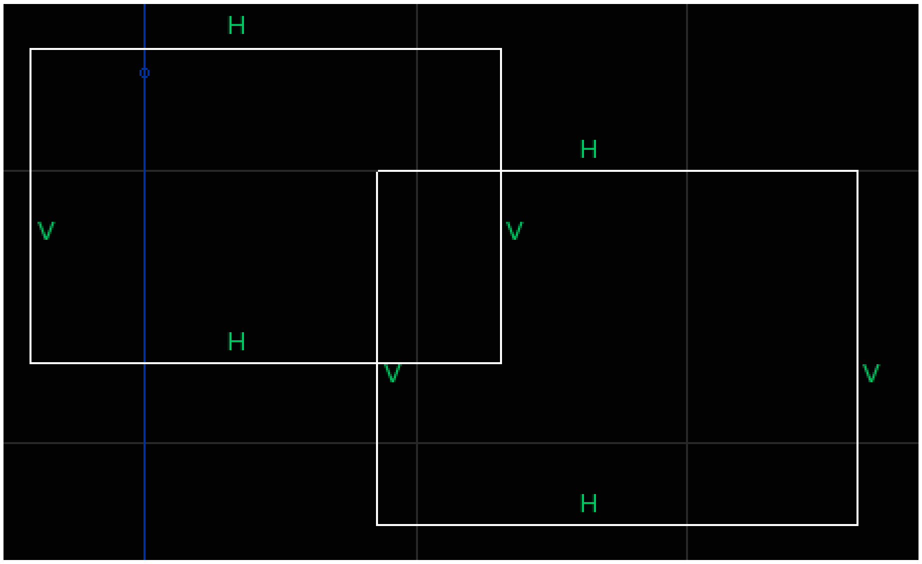
Snapping tools are used to snap the mouse to a specific point or edge in the EMPro geometry.

> **Note**
> The blue lines in the images below highlight the "snap-to" landmarks.

Snap to Grid Line

Mouse is snapped to the nearest point on the nearest grid line.

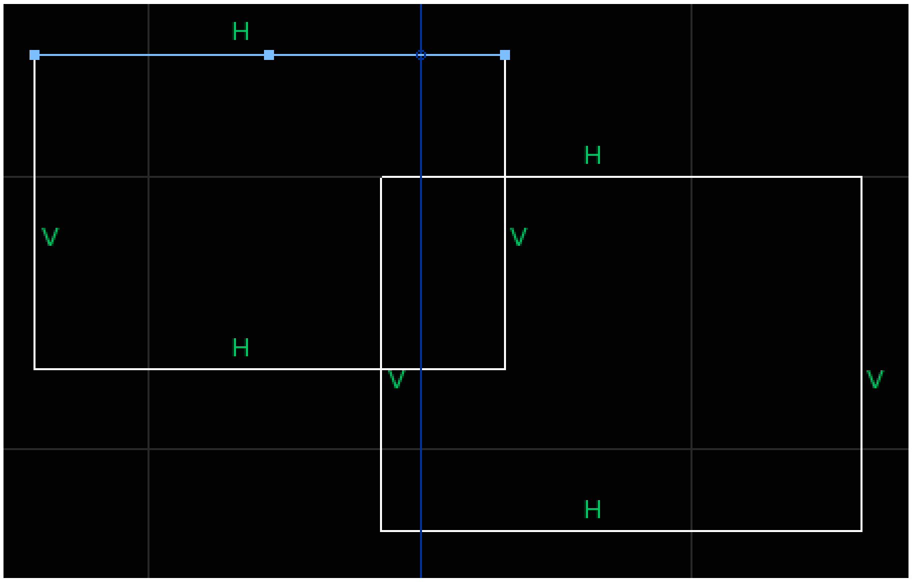
Snap to Grid Line Tool



Snap to Grid/Edge Intersections

Mouse is snapped to the nearest intersection between the grid and the sketch edge.

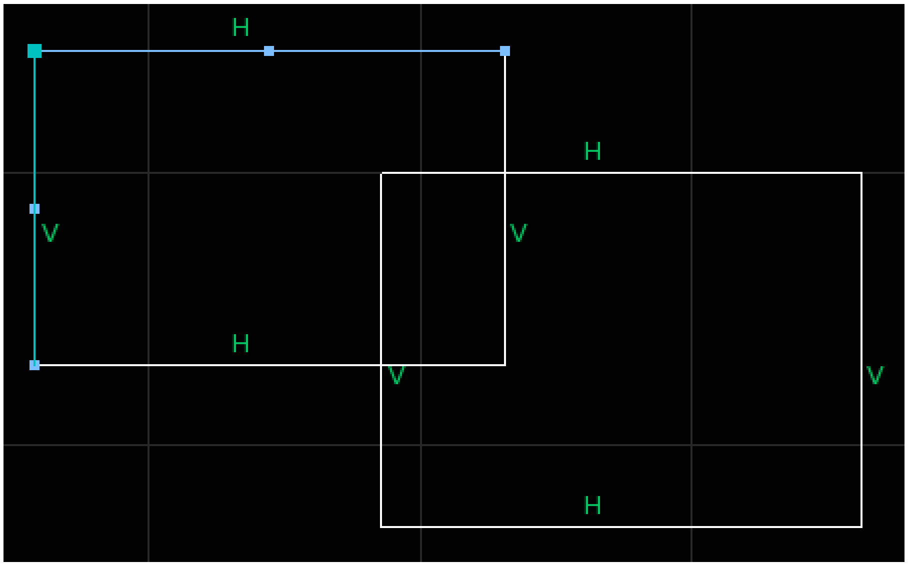
Snap to Grid/Edge Intersections Tool



Snap to Vertices

Mouse is snapped to the nearest vertex of the sketch or edge mid-point within range.

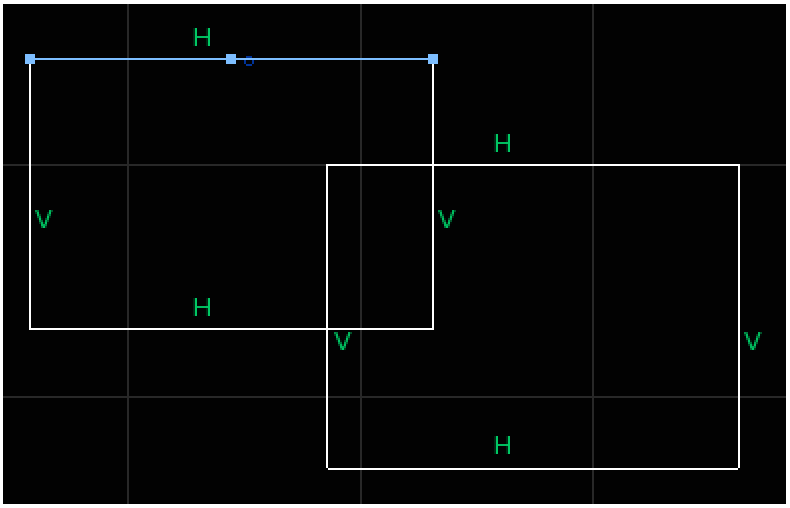
Snap to Vertices Tool



Snap to Edges

Mouse is snapped to the edges of a pre-defined object.

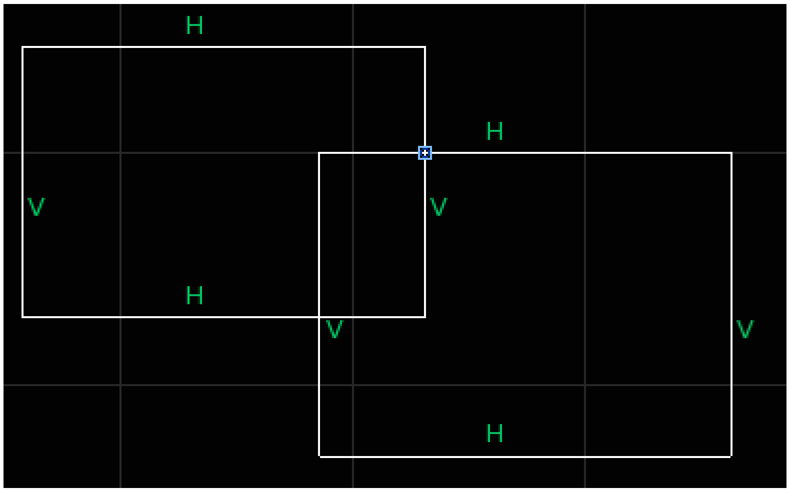
Snap to Edges Tool



Snap to Edge/Edge Intersections

Mouse is snapped to the vertices of intersecting edges.

Snap to Edge/Edge Intersection Tool



2-D Modeling Options

The 2-D Modeling tools are used to outline or fill-in a simple geometry object.

Wire Body

The *Wire Body* tool is the simplest geometry object. Any of the Shape tools can be used to create the desired wire geometry.

Sheet Body

The *Sheet Body* tool is similar to the *Wire Body* tool except its interior is filled with a material.

> **Note**
> It is also possible to create a sheet body using advanced options with 3-D modeling operations.

Sheet Body from Faces

The *Sheet Body from Faces* tool enables you to create a *Sheet Body* from the face of a pre-existing geometry object. The interface will prompt the user to select the desired object face.

3-D Solid Modeling Options

The 3-D Modeling tools are used to create simple solid geometry objects from 2-D forms.

> **Note**
> For solid body creation, the 2-D sketch must be closed so that there are no lingering endpoints.

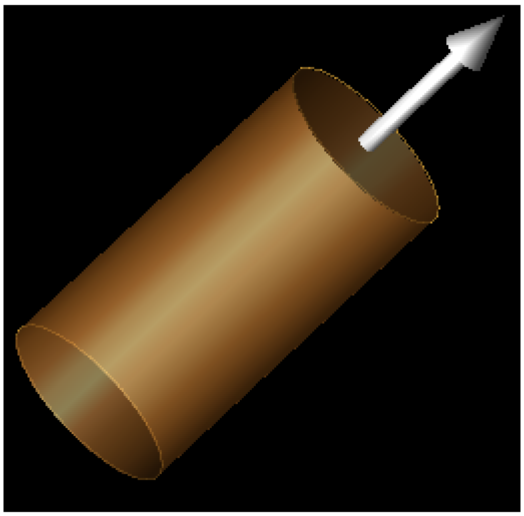
Extrude

Extrude is used to sweep a face in the normal direction from its center. Once a 2-D form is made in the *Edit Cross Section* tab, select the *Extrude* tab to its right to perform an extrusion. For a default extrusion, define the distance in the *Extrude Distance* dialog box by typing in a numerical value, parameter name (See: Section Defining Parameters), or equation.

> **Note**
> If units are not entered next to the numerical value, the default units are assumed.
>
> For more information about defining distances with parameter names, refer to Defining Parameters.

Additionally, the *Direction* dialog box specifies the axis along which the extrusion will occur. Clicking done after the desired geometry is created will add the object to the project. It can now be seen in the *Project Tree*.

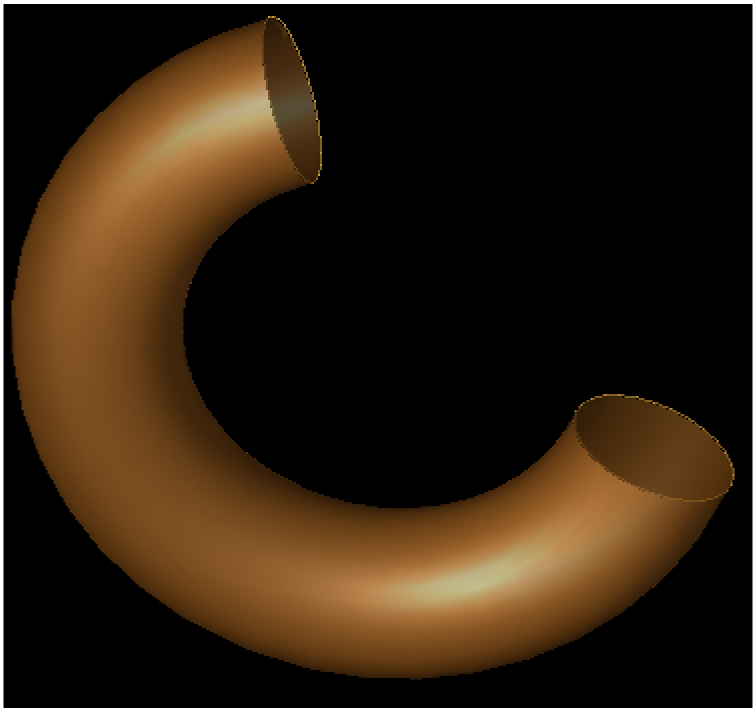
Extrusion Tool



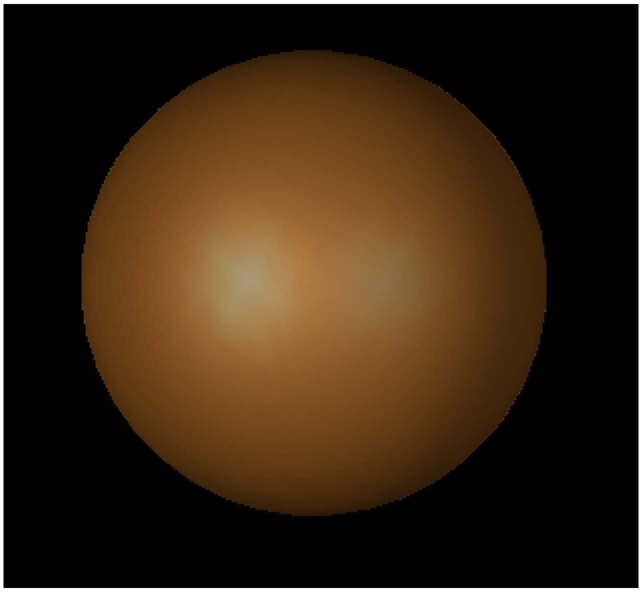
Revolve

Revolve is used to sweep a face in a circular path. Once a 2-D form in made in the *Edit Cross Section* tab, select the *Revolve* tab to perform a revolution. For a default revolution, define the angle in the *Angle* dialog box by typing in a numerical value, parameter name, or equation. The *Axis Root Position* dialog specifies the location of the root of the axis around which the shape will revolve. The *Axis Direction* box specifies the direction along which the revolution will occur. Clicking DONE after the desired geometry is created will add the object to the project. It can now be seen in the *Project Tree*.

Revolution Tool



Creating a sphere with the Revolution Tool



Advanced 3-D Solid Modeling Operations

The Advanced 3-D Modeling tools are used to modify a pre-defined 3-D geometry object. They are available within the *Extrude* and *Revolve* operations.

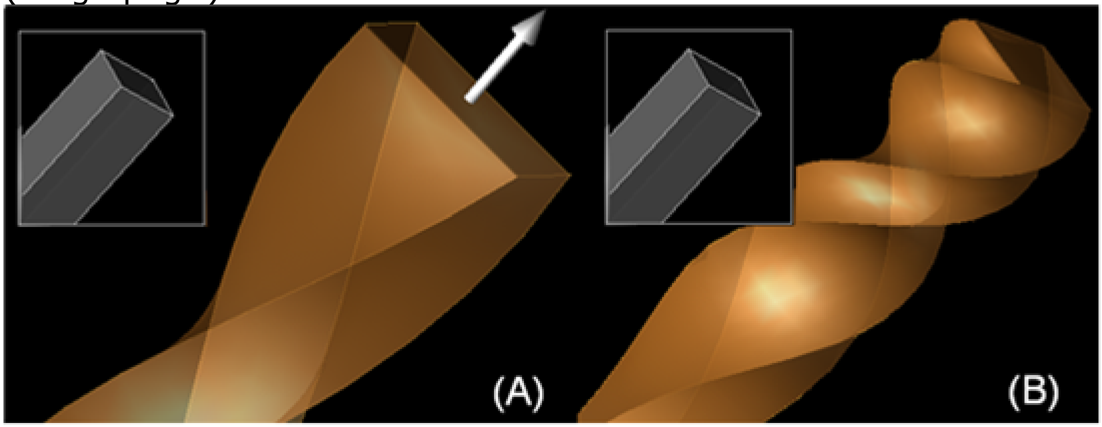
Twist

Twist options control how much the face is twisted as it is swept. They can be specified by angle or law.

By Angle: Specify the total number of degrees that the face will twist while it is swept.

By Law: Specify a mathematical expression to control the rate of twist as a function of the variable ***X*** .

In the following figure, the Twist Tool defined by A) Angle (90 degrees) and B) Law (!img4.png!)

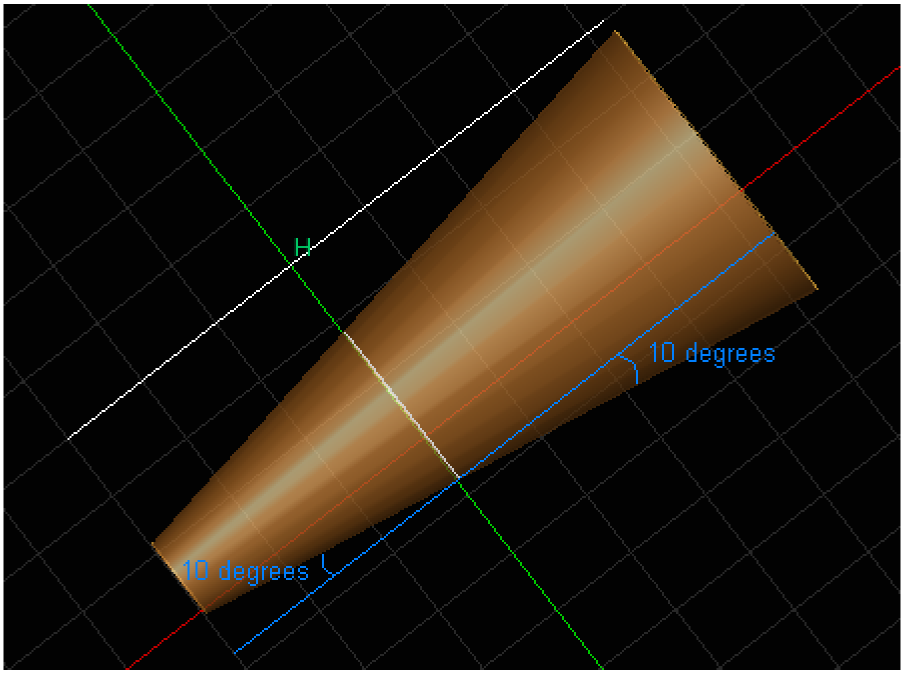


Draft Type

Draft Type options control the expansion or contraction of the edges of the face as it is swept from its initial position.

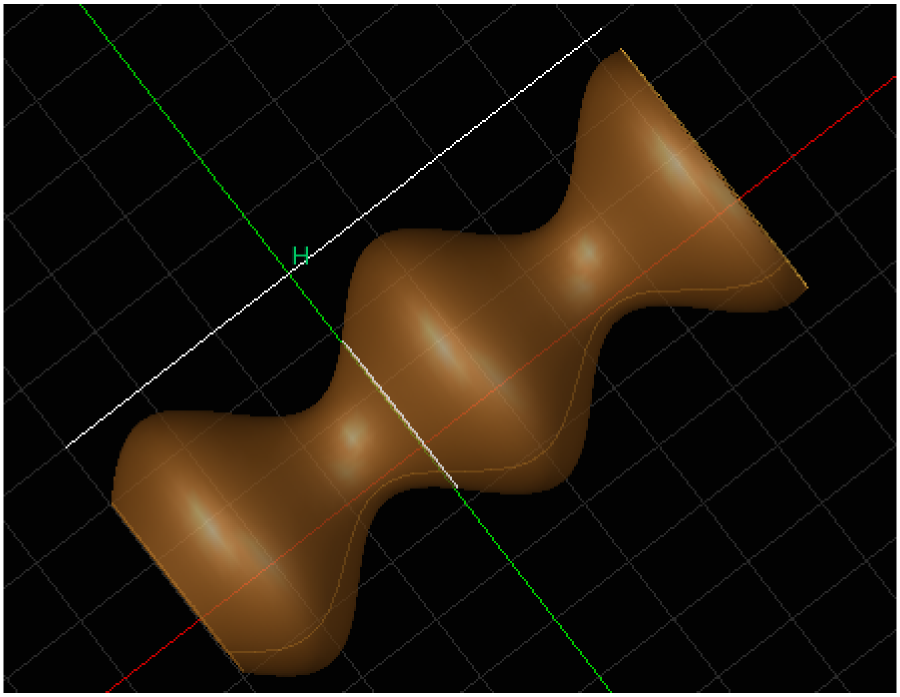
No Draft: No expansion or contraction of edges during sweep.
Draft Angle: Specify the expansion or contraction angle from initial position.

A cylinder sweep with Draft By Angle (10 degrees)



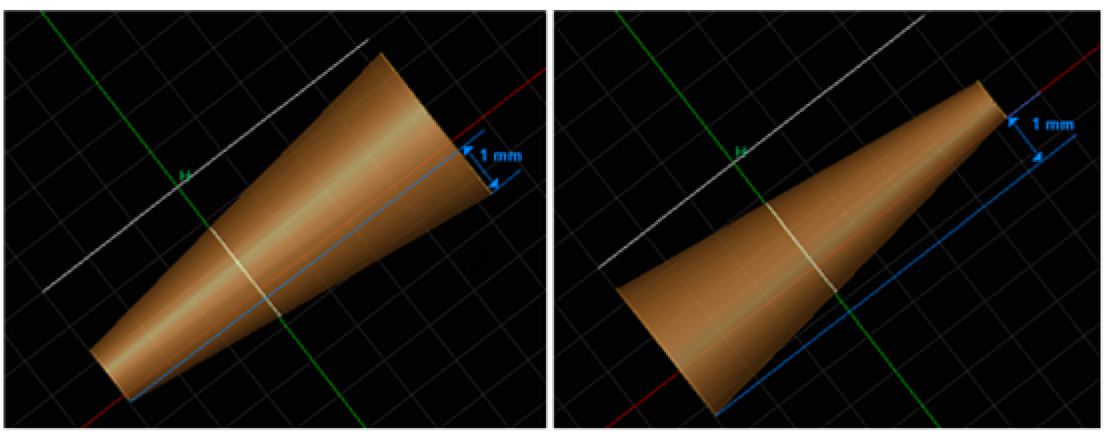
Draft Law: Specify a mathematical law to control the shape of the sides as the face is swept from initial position as a function of the variable ***X*** .

A cylinder sweep with Draft By Law (.5sin(2x))



End Distance/Start Distance: Specify the offset distance in the plane where the sweep ends/begins.

A cylinder sweep with Draft By End Distance (1 mm) and Start Distance (1 mm)



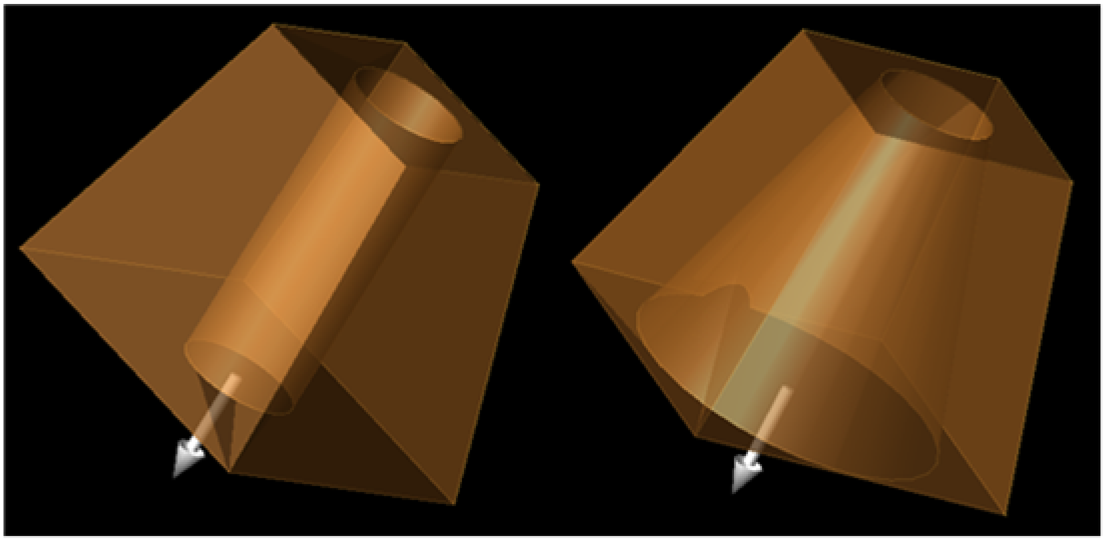
Hole Draft Type

Hole Draft Type options control the expansion and contraction of a hole. They are therefore only valid during sweeping operations applied to a faces that contain holes. *Hole Draft Type* can be defined based on the values assigned to the edges in Draft Type options, or by angle.

No Draft: No expansion or contraction is applied to the hole, even if the face has a *Draft Type* applied to it.

Draft Angle: Specify the expansion or contraction angle from initial position.

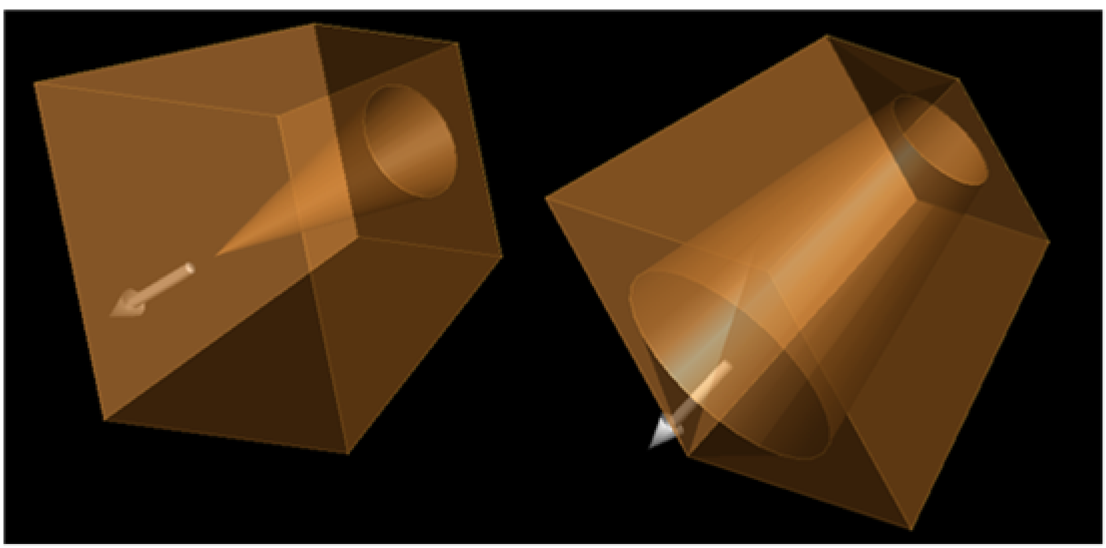
Hole with no Draft (left) and a defined Draft Angle (right)



With Periphery: The expansion or contraction of the hole will be the same as the outside edges of the face as specified in *Draft Type*.

Against Periphery: The expansion or contraction of the hole will be the opposite to the outside edges of the face as specified in *Draft Type*. (i.e., the hole will contract as the face expands and expand when the face contracts.)

Hole with Draft Angle against (left) and with the Periphery (right)



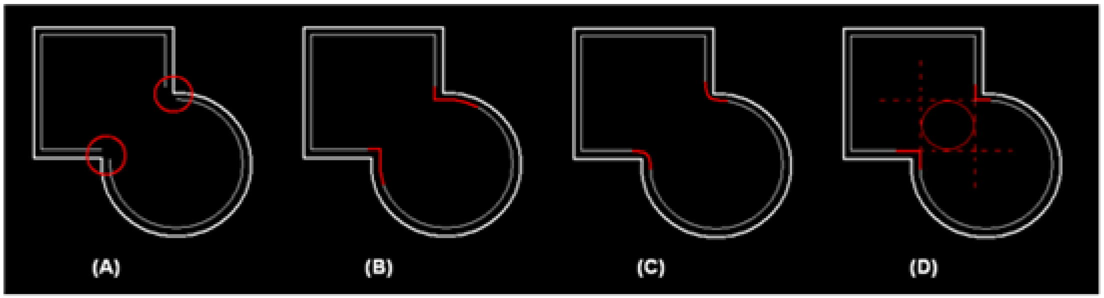
Gap Type Modeling Operations

The *Gap Type* specifies how to close the gap created by an offset. The default gap type is *Natural*, but the following options are available for filling gaps in the geometry.

Natural: Extends the two shapes along their natural curves until they intersect.

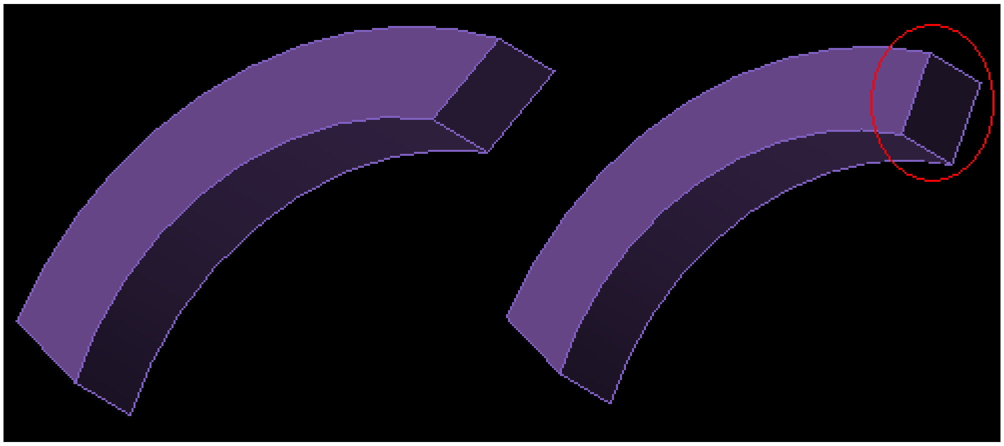
Rounded: Creates a rounded corner between the two shapes.
Extended: Draws two straight tangent lines from the ends of each shape until they intersect.

. Illustration of gap types, showing A) the original gap, B) Natural, C) Rounded and D) Extended.



Cut Off End

Controls the orientation of a face that does not follow its normal during a straight sweeping operation. Select this option to chop the end of the swept 3-D object so that the normal of the end face is aligned with the line used for sweeping. Original Model (Left) and Model After Cut Off End (Right)



Make Solid

This option makes the model entirely solid. If this option is not selected, the model will be hollow.

Modifying Existing Geometry

Specify Orientation

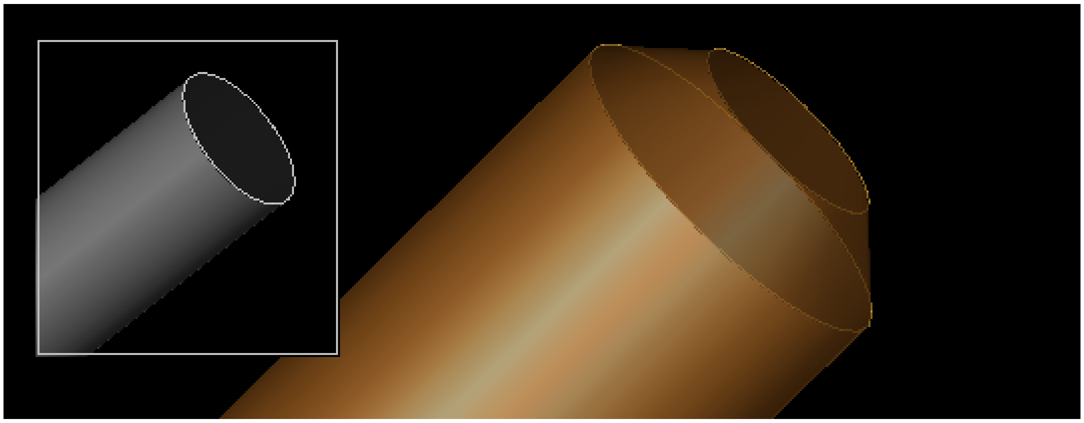
The *Specify Orientation* button is used to position the selected geometry in the simulation space. Clicking this icon will bring up the *Specify Orientation* tab.

Note
For more information on using the *Specify Orientation* tab, refer to *Specify Orientation Tab* (quickstart).
For descriptions of the tools used to rotate, translate and zoom into the simulation space View Tools.

Chamfer Edges

Chamfer Edges operation creates a beveled edge between two surfaces. After selecting the edge, it will be trimmed at a 45 ° angle if *Constant Distance* is selected in the *Specify Distance* tab. Otherwise, the user enters the chamfer distance for the surfaces on the left and right sides of the edge.

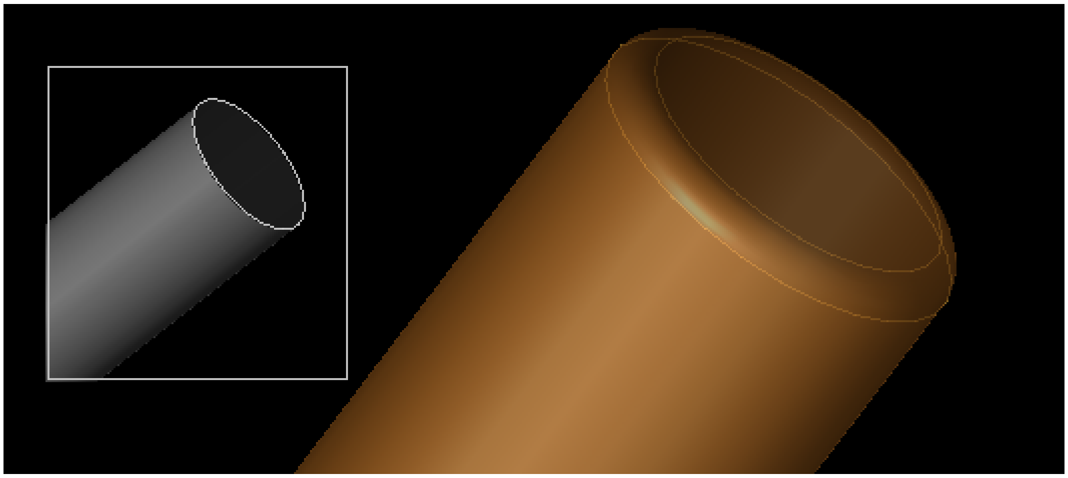
A Chamfer operation applied to a cylinder edge



Blend Edges

The *Blend Edges* operation rounds the selected edge of the geometry. Under the *Specify Radius_tab, the user can enter the _Blend Radius* to adjust the rounding factor.

A Blend operation applied to a cylinder edge

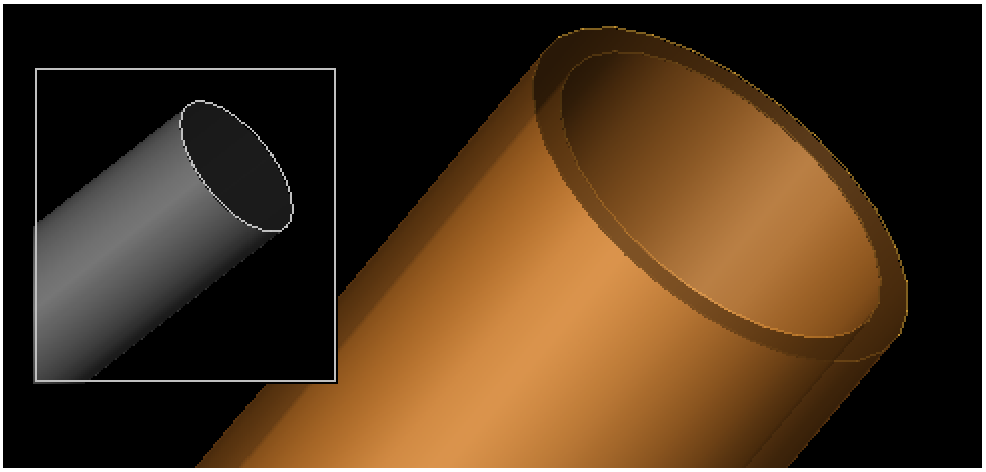


Shell Faces

The *Shell Faces* operation creates a shell from existing geometry. After selecting the faces to keep open, the user can enter the *Shell Thickness* under the *Specify Thickness* tab.

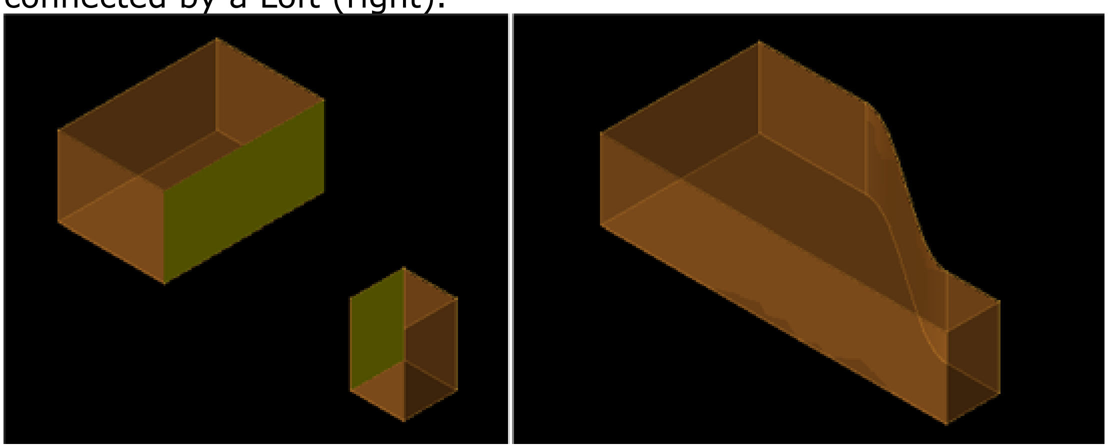
Note
By definition, the shell operation is used on geometry which is intended to have volume. This operation is not for use an object such as a *Sheet Body*, whose volume is insignificant in the EMPro calculation.

A Shell operation applied to a cylinder



Loft Faces

The *Loft Faces* operation connects two parts of an existing geometry. Under the *Specify Loft* tab, the user can adjust the *Smoothness Factor* to create the desired shape. In the following figure, two objects within a geometry with faces selected (left) and later connected by a Loft (right).



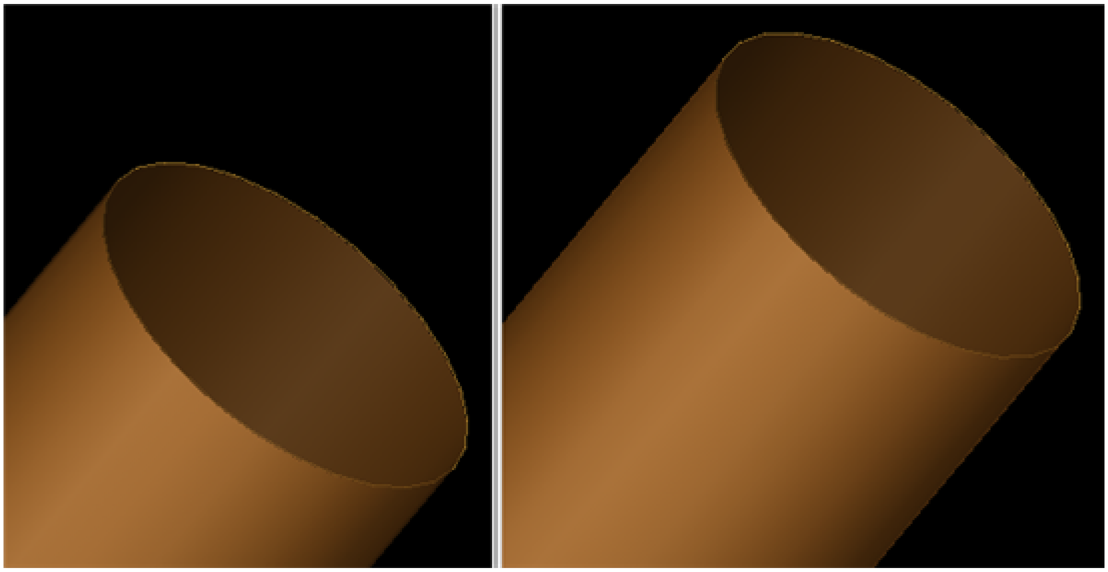
Remove Faces

The *Remove Faces* operation removes a blend or chamfer that was previously applied to a geometry edge. This operation must be applied before the user can offset the length of an object.

Note
This operation is useful for modifying objects that have been imported from CAD files.

Offset Faces

With *Offset Faces*, the user enters a positive or negative offset distance to increase or decrease the length of the selected model, respectively. The following figure displays a cylinder with an applied negative offset (left) and positive offset (right).



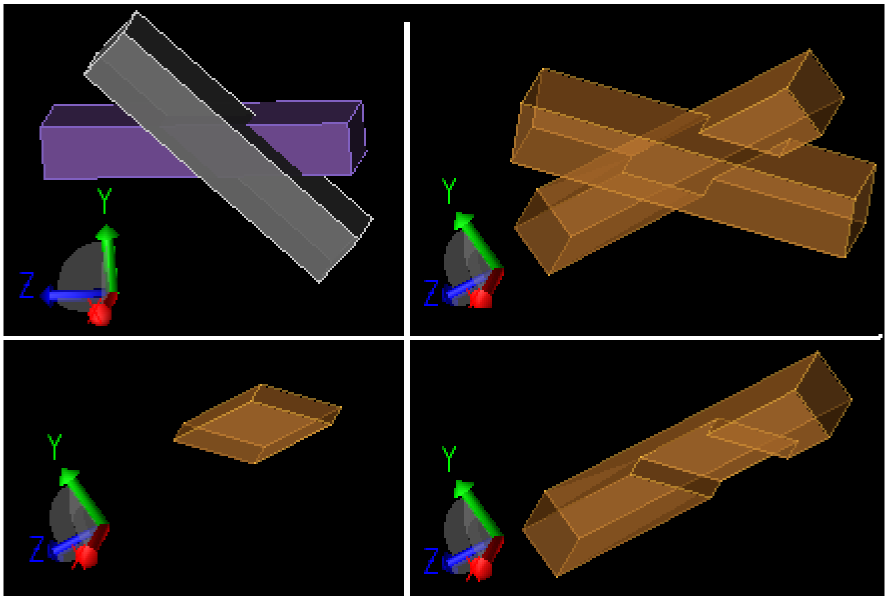
Boolean Operations

Two Parts Boolean Operation

The *Two Parts* Boolean options perform operations on two existing geometry parts. In each case, one object is identified as the *Tool* (the part used to perform the modification), and the other as the *Blank* (the part that is modified). There are three types of operations:

- Subtract
- Intersect
- Union

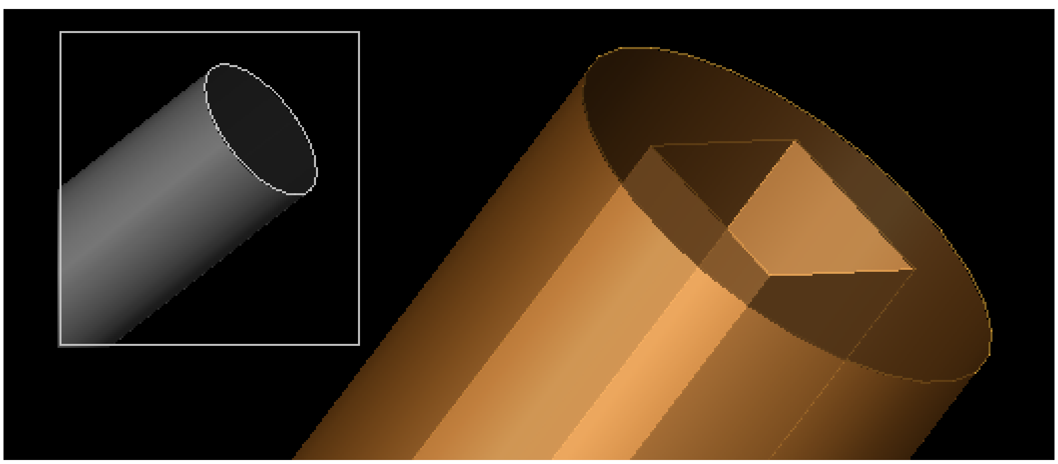
In a *Subtract* operation, the *Tool* is subtracted from the *Blank*. In the *Intersect* and *Union* operations, the part selected first is inconsequential. The following figure displays the original two objects (Upper Left), objects after Boolean Union (Upper Right), objects after Boolean Intersection (Lower Left) and objects after Boolean Subtraction (Lower Right).



Extrude Boolean Operation

The *Extrude* Boolean option performs an operation on an existing geometry part. In this case, the user chooses the *Blank*, and then creates the object to use as the TOOL. The user then specifies the orientation of the extrusion and the nature of the operation (*Subtract, Intersect*, or *Union*). In essence, this operation is a shortcut for the *Two Parts* Boolean operation.

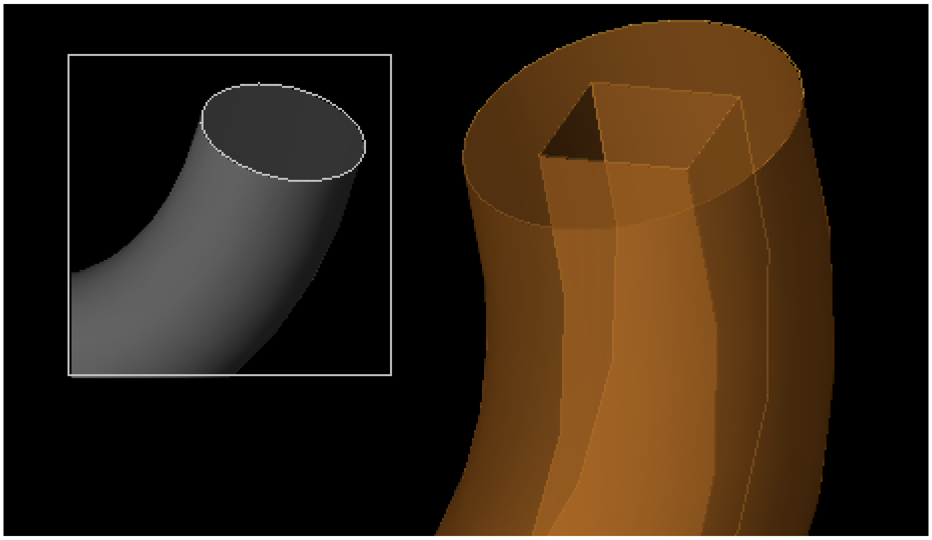
A boolean extrude operation



Revolve Boolean Operation

The *Revolve* Boolean option performs an operation on an existing geometry part. The user chooses the *Blank*, and then creates the object to use as the *Tool*. The user then specifies the orientation of the revolution and the nature of the operation (*Subtract, Intersect*, or *Union*).

A boolean revolve operation



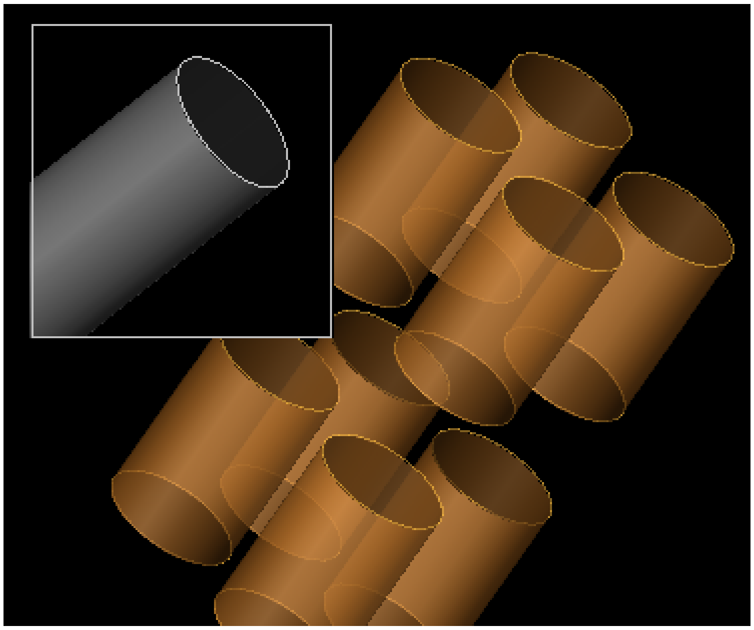
3-D Patterns

Linear/Rectangular Pattern

The *Linear/Rectangular* Pattern option enables you to select a part in the geometry and replicate it in a linear pattern. After selecting the part to modify, define the *Spacing* and *Number Of Instances* in the **U'**, **V'** and **W'** directions. Spacing refers to the distance between objects in the specified direction, and Number of Instances refers to the number of objects in the specified direction. For example, if three cylinders are to be spaced at 2-mm intervals in the **U'**-direction, the Spacing in the **U'**-direction is 2 mm and the Number of Instances in the **U'**-direction is 3. Additionally, the *Stagger* check-boxes apply a stagger in the specified direction at every other instance in that direction.

> **Note**
> Spacing refers to the distance between each object's center point in the specified object. So, for example, if the spacing between two cylinders does not exceed the distance of the cylinder's diameter, the cylinders will overlap.

A linear pattern applied to a cylinder

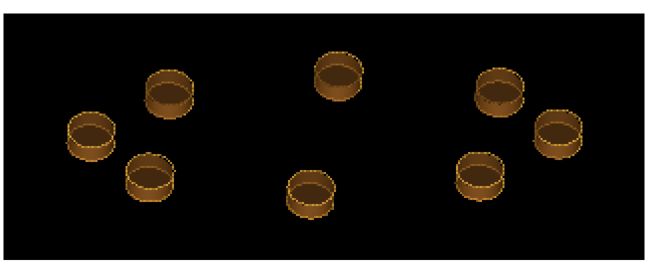


Circular/Elliptical Pattern

The *Circular/Elliptical* Pattern option enables you to select a part in the geometry and replicate it in a circular or elliptical pattern. After selecting the part to modify, navigate to the *Specify Circular/Elliptical Pattern* tab and define the following fields:

- **Axis Point** - specifies the position of the axis
- **Axis Normal** - specifies values to define the direction of the pattern
- **Root Position** (available in *Elliptical Mode*) - specifies a point (usually the center of a part) to use as the reference to replicate in the elliptical pattern.
- **Major Axis** (available in *Elliptical Mode*) - specifies the direction of the major axis.
- **Pattern Options**
  - **Instances** - specifies the number of objects in the pattern
  - **Angle** - specifies the angle across which the objects are patterned (i.e., 180!img463.png! means that objects are patterned across half of the ellipse)
  - **Ratio** (available in *Elliptical Mode*) - specifies the ratio of the minor axis to the major axis

An elliptical pattern applied to a small cylinder



Grid Appendix

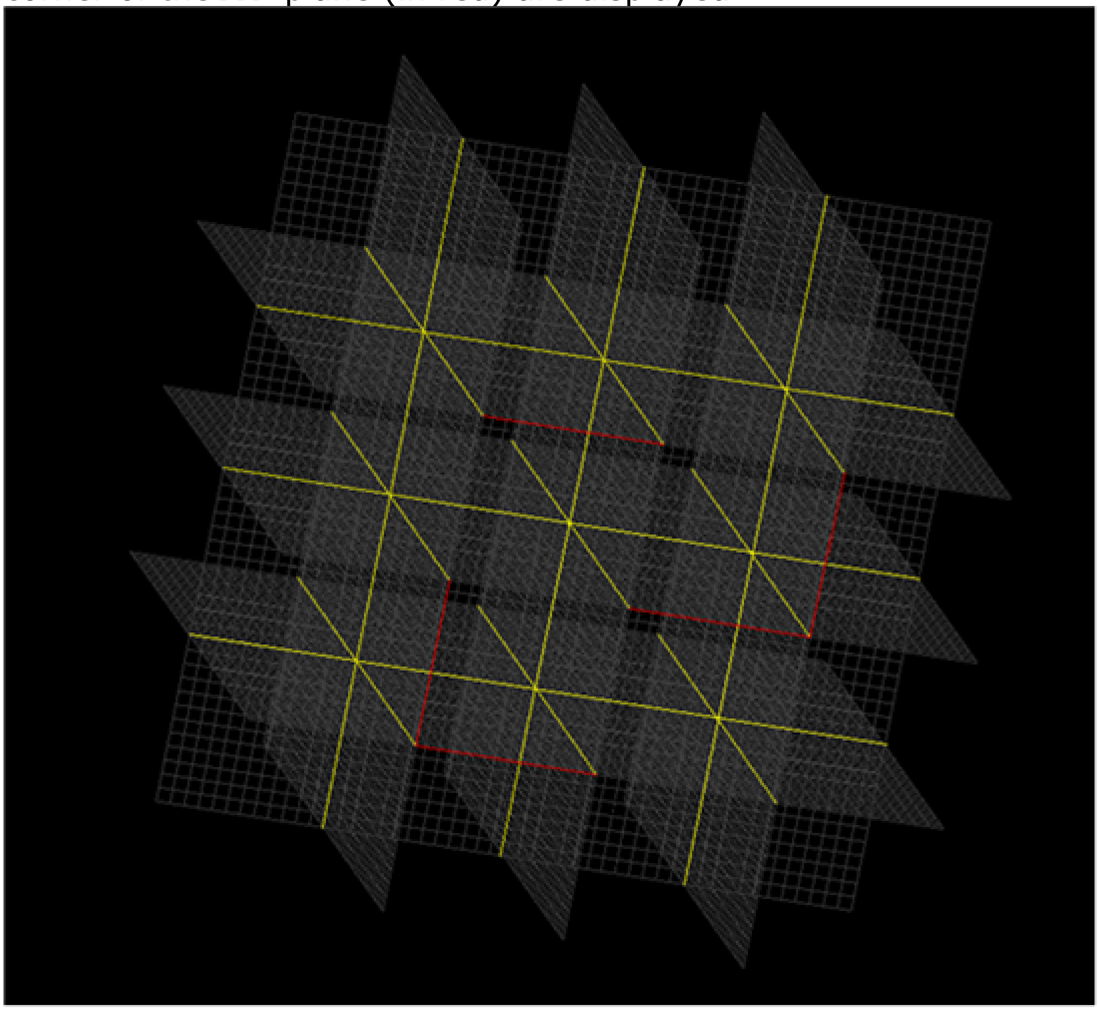
In this section, you will learn about:

- Concept of creating a grid in EMPro.
- How to choose a cell size to optimize your EMPro project calculation.
- Methods for varying the EMPro grid.
- How to debug the EMPro grid.

Grid Concepts Overview

The grid consists of three sets of points, one set for each axis: ***X***, ***Y*** and ***Z***. At each point is a plane. For example, consider the point at ***X*** = 3, which defines a plane in ***Y*** and ***Z***. The point ***Y*** = 4 defines a plane in ***X*** and ***Z***. The point ***Z*** = 5 defines a plane in ***X*** and ***Y***. Where two planes intersect is a line. Planes from the remaining axis cut that line into edges. These edges are called "**cell edges**". Building the grid consists of defining the appropriate set of plane-defining points for each axis. (Meshing, which occurs after gridding, is the act of assigning materials to each cell edge.)

The figure below displays intersecting planes in 3-D with lines and cell edges highlighted. The intersecting planes with lines (in yellow) and sample cell edges in the upper-left corner of the ***XY***-plane (in red) are displayed.



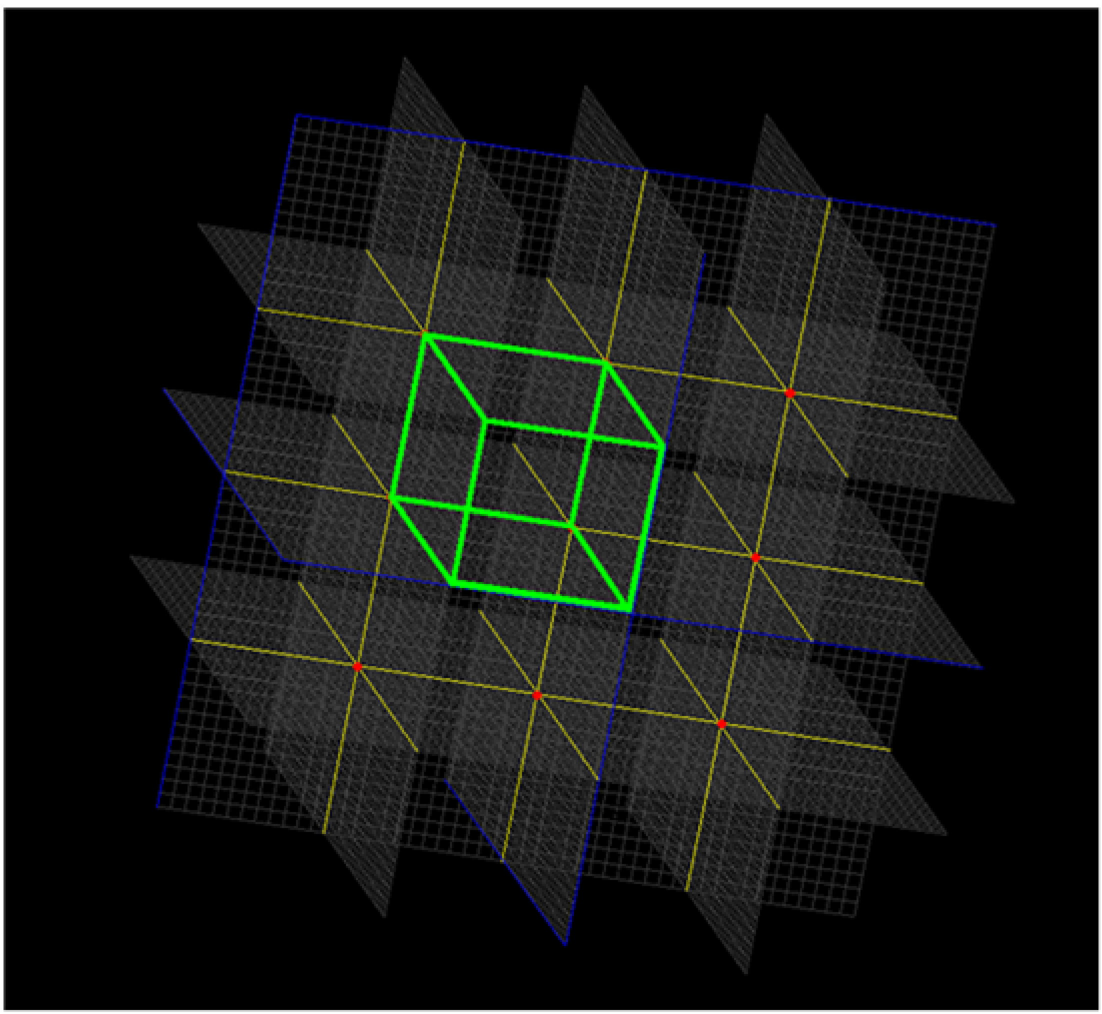
Except for some special features associated with components, which are discussed below, the grid is made up of the following elements:

- Fixed points
- Grid regions
- Target cell sizes
- Automatic fixed point merge distances

A fixed point is a point on an axis at which a plane, in the other two axes, exists. Edge-on, that plane is seen as a line. That is what is meant by a "**grid line**". A grid region is a bounded part of the grid. Grid lines are placed at the grid region boundaries. Within a grid region the target cell size and the fixed point merge distance can be different from the project's default target cell size and/or merge distance. One fixed point is placed at the beginning of each grid region and another fixed point is placed at the end of each region.

The figure below displays intersecting planes in 3-D with fixed points, grid lines and a grid region highlighted.

Intersecting planes with fixed points (in red), grid lines (in blue) and a grid region (in green)



Manual fixed points and manual grid regions are specified in the *Grid Tools* dialog. Other grid regions and fixed points, associated with individual parts, may be specified using the *Gridding Properties Editor*. Grid regions associated with a part are called part grid regions. Fixed points associated with parts are called automatic fixed points, because they are automatically extracted from the geometry of the part.

Taking these elements together, the grid is made up of manual fixed points, automatic fixed points, manual grid regions and part grid regions. Each kind of grid region contains a target cell size and an automatic fixed point merge distance. Those values are used when creating the grid.

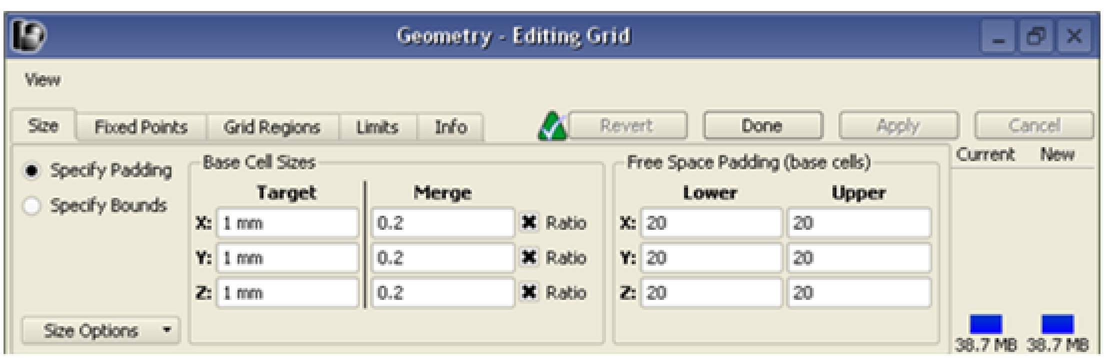
> **Note**
> It is important when building a grid to control the size of the smallest cell in the grid. The project's timestep is derived from the smallest grid cell's edge length. Smaller timesteps result in longer runs and larger timesteps result in faster runs. For that reason it is important to prevent the grid from having a cell edge smaller than what is necessary to get the desired results.

Automatic fixed points and part grid regions are extracted from geometry and may result in points so close that the timestep is smaller than desired. The fixed point merge distance is used to merge automatic fixed points in order to provide control of the timestep. Automatic fixed points are merged so that they are no closer than the merge distance. If a grid region start or end boundary is too close to another grid point, the grid region may be expanded to prevent the too-small timestep. The grid region is always expanded, never contracted, in these situations.

The grid is created in the following steps. Each axis, ***X***, ***Y*** and ***Z***, is considered separately. Note that when considering only a single axis, a cell size is really just an edge size.

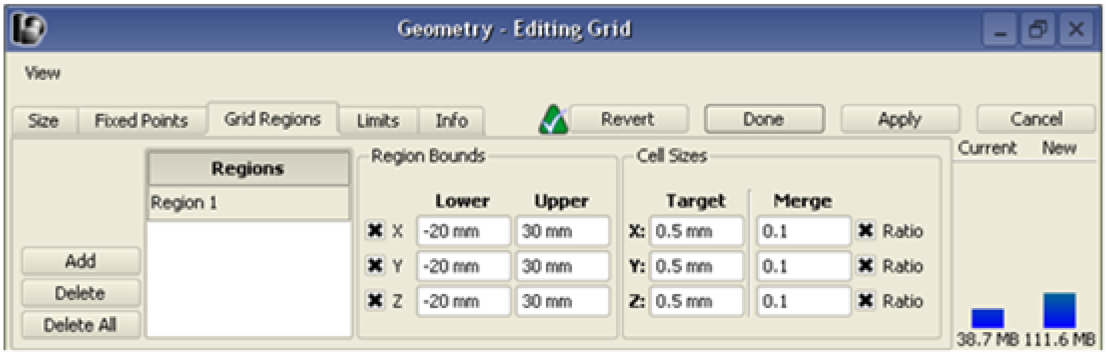
- Create a set containing manual and automatic fixed points and the fixed points from the borders of manual and part grid regions. If the *Specify Padding* option is chosen, then fixed points for overall bounds of the geometry are added to the set.

Editing the grid with the Specify Padding option chosen



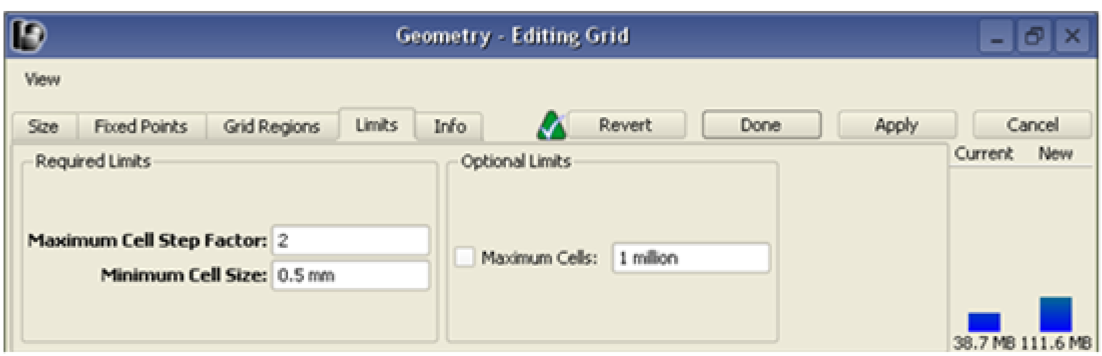
- Automatic fixed points are merged according to the merge distance. The fixed point merge distance can have different values at different points on the grid. There is a " **main grid**" fixed point merge distance that is specified on the *Size* tab (see the figure above). Each grid region, including both manual and part grid regions, has its own fixed point merge distance. The smallest fixed point merge distance for a given point on the grid is chosen from all manual and/or part grid regions covering the given point. If no grid region covers the given point, the main grid fixed point merge distance is chosen. Note that this allows grid regions to specify a merge distance greater than the main grid fixed point merge distance. Notice that in the figure below a smaller merge distance is specified in the grid region than that of the main grid.

Adding a grid region with a smaller cell size than the main grid



- The movable points, for example automatic fixed points and grid region start/end boundaries, are moved away from unmovable points. Unmovable points include manual fixed points and some entries associated with components, as described below. Remember that grid regions never shrink in order to ensure that regions needing improved accuracy get it.

- The set of grid points is examined. Transition regions are added to prevent adjacent cell size ratios from violating the maximum cell step factor specified on the *Limits* tab. For example, a transition region is generated if the maximum cell step factor is 2, and 2 adjacent cells have sizes 5 mm and 1 mm. A transition region contains the fewest number of cells required to reach the desired cell size. Each cell in the transition region has a progressively larger size. In the following figure, the Maximum Cell Step Factor shown is never exceeded by adjacent cell size ratios.



- The set of grid points, including the transition regions, is examined. Gaps greater than the target cell size at the given point are filled evenly with the fewest number of cells required such that the cell size is less than or equal to the target. For example, consider a gap of 9.7 mm with a target cell size of 1 mm. In this case, 8 new points must be added to the 2 points surrounding the gap. The distance is $9.7\ mm - 1\ mm = 8.7\ mm$, bridged by 9 cell edges. Each cell edge will be

$$\frac{8.7mm}{9edges} = 0.9667\ mm$$

per edge.

That completes the calculation of grid points. There are some key facts to note:

No matter what the fixed point merge distance may be, it is always possible for there to be two fixed points a distance of $(mergedistance + \epsilon)$ apart after the merging has been completed. If the target cell size at the given point is less than $(mergedistance + \epsilon)$, then this distance must be subdivided into two or more cell edges.

Consider a 0.7 mm merge distance with a .2 mm target size. Given that

$$\frac{0.7mm}{0.2mm} = 3$$

remainder !img220.png! , there must be 3 grid lines in the 0.7 mm gap. Those 3 grid lines create 4 spaces in the gap. The gaps are

$$\frac{0.7mm}{4} = 0.175\ mm$$

each. The formula is: $\frac{mergedistance}{\lceil \frac{mergedistance}{target} \rceil}$

The target cell size may be greater than the automatic fixed point merge distance. Consider a merge distance of 0.7 mm and a target distance of 1 mm. If the automatic fixed points are dense then every gap is exactly 0.7 mm. No gaps are greater than 1 mm, so the work is done. If there was at least one region of sparse automatic fixed points then there may be a gap greater than 1 mm. That gap must be bridged using one or more extra grid lines. The smallest such gap is $\frac{target}{2}$, or 0.5 mm in our example.

The smallest cell in the grid will be the smaller of $\frac{mergedistance}{\lceil \frac{mergedistance}{target} \rceil}$ and $\frac{target}{2}$. Any given grid may not encounter one or both of those situations, so that grid's smallest cell size may be larger.

Choosing an Appropriate Cell Size

Since smaller cells require longer calculation time, it may be advantageous to also define the lower limit of the cell size in the *Size* tab of the *Gridding Properties Editor* window. The minimum cell size can be defined as a *Merge* distance (i.e., a specific distance with units), or as a ratio of the *Merge* value to the *Target* base cell size (i.e., a *Merge* value of 0.8 would restrict the minimum cell size from dropping to a value below 80% of the *Target* base cell size.)

> **Note**
> To learn about the factors that affect the smallest cell size the equations at the end of the previous section, refer to Grid Concepts Overview.

When defining the *Target* base cell size, ensure that the cell size is much less than the smallest wavelength for which accurate results are desired. A commonly applied constraint is *ten cells per wavelength* , meaning that the side of each cell should be less than one-tenth of the wavelength of the highest frequency (shortest wavelength) of interest. If the cell size is much larger than this, the Nyquist sampling limit is approached too closely for reasonable results to be obtained. Significant aliasing is possible for signal components above the Nyquist limit.

Choosing a cell size of one-tenth of a wavelength is a good starting point, but other factors may require a smaller cell size to be chosen, such as small geometry features and material characteristics.

Grid definitions can be customized for specific objects in the *Gridding Properties Editor* window, so that smaller features are considered, without having to apply smaller, memory-intensive cells to the whole grid.

> **Note**
> For more information about assigning grid definitions to specific object *Gridding Properties Editor* (using).

Material characteristics will also influence cell characteristics since EMPro 2010 is a volumetric computational method. If some portion of the computational space is filled with penetrable material, the wavelength in the material must be used to determine the maximum cell size. Geometries containing electrically dense materials require smaller cells than geometries that contain only free space and perfect conductors. For this reason, a material definition must be applied to *Parts* objects to generate a valid mesh. An error message will appear in the case that a material is not assigned to an object.

> **Note**
> For information on applying material definitions to objects *Material Editor* (using).

Grid Regions vs. Fixed Points

There are two primary means of varying the grid in EMPro 2010. A grid region is a region within the grid that is assigned its own target cell size, which is different from the default grid size defined in the main *Size* tab of the *Grid Tools* button. A fixed point is a point on an axis where a grid line will be placed. Cell sizes are adjusted to flow smoothly between fixed points, never exceeding the *Target* cell size.

The target cell size can vary within different grid regions along a given axis. The main grid's target cell size applies everywhere except within grid regions. A grid region has start and end boundaries on an axis. Grid regions can have target cell sizes and automatic fixed point merge distances that differ from the main grid and from other grid regions. Grid regions can overlap. When they do, the smallest target cell size and the smallest automatic fixed point merge distance are chosen from all of the overlapping grid regions at the given point.

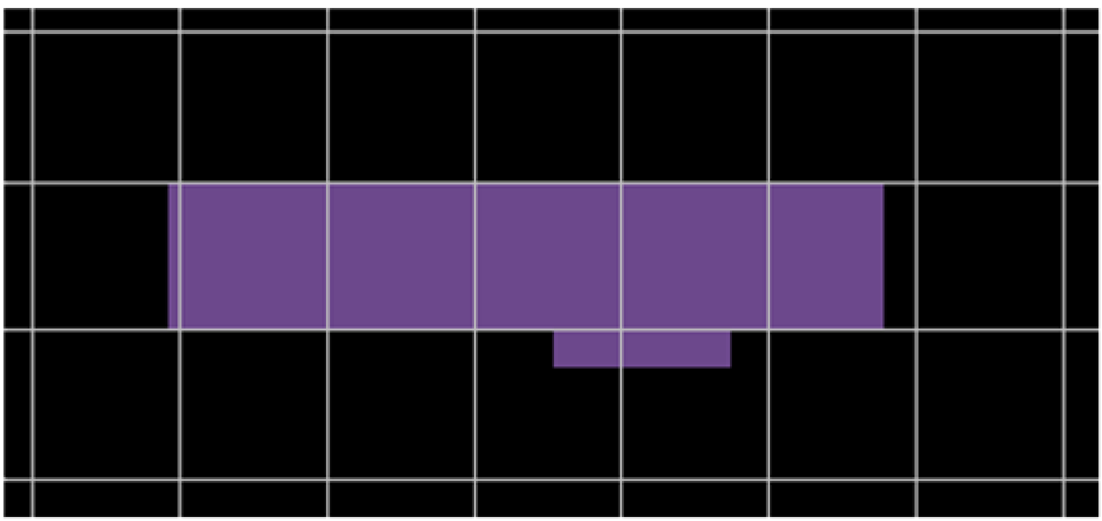
Grid regions, like fixed points, can be manual or automatic. Manual grid regions are defined on tabs associated with the main grid editor. Automatic grid regions are associated with parts and so are also called part grid regions. Part grid regions are defined by right-clicking on a part in the tree and choosing Gridding Properties.

Any grid region includes fixed points for the grid region boundaries. Between the bounds, the target cell size and automatic fixed point merge distances can be different than their values outside the grid region bounds.

The grid flows as evenly as possible between fixed points, using at each point the appropriate target cell size and automatic fixed point merge distance for that point.

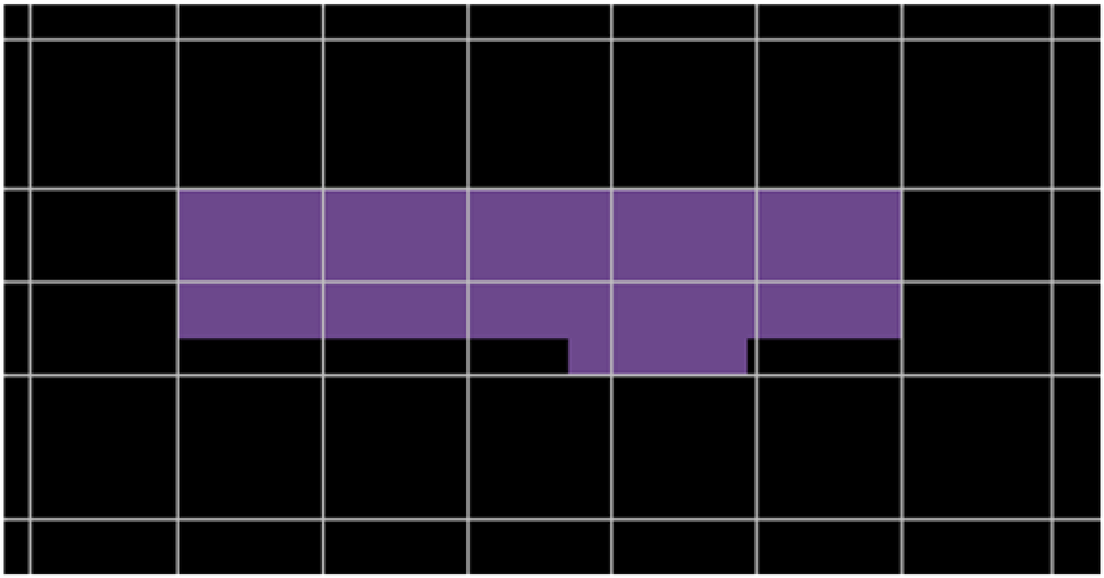
The figure below shows a simple shape that has a uniform grid size of 1 m.

Simple extrusion with uniform grid



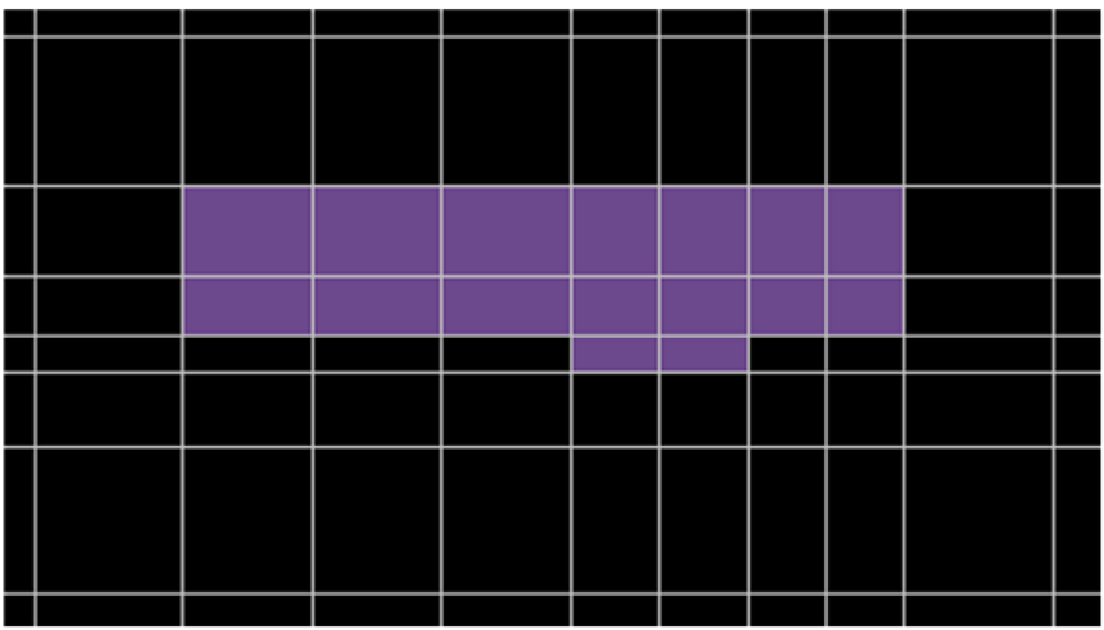
The following figure applies an automatic grid region to this simple shape. Note that the edges of the rectangle are now aligned with the edges of the grid. Also note that, because the height of the rectangle was not evenly divided by 1 m, the main grid spacing adjusted slightly to accommodate for this.

Simple extrusion with applied grid regions



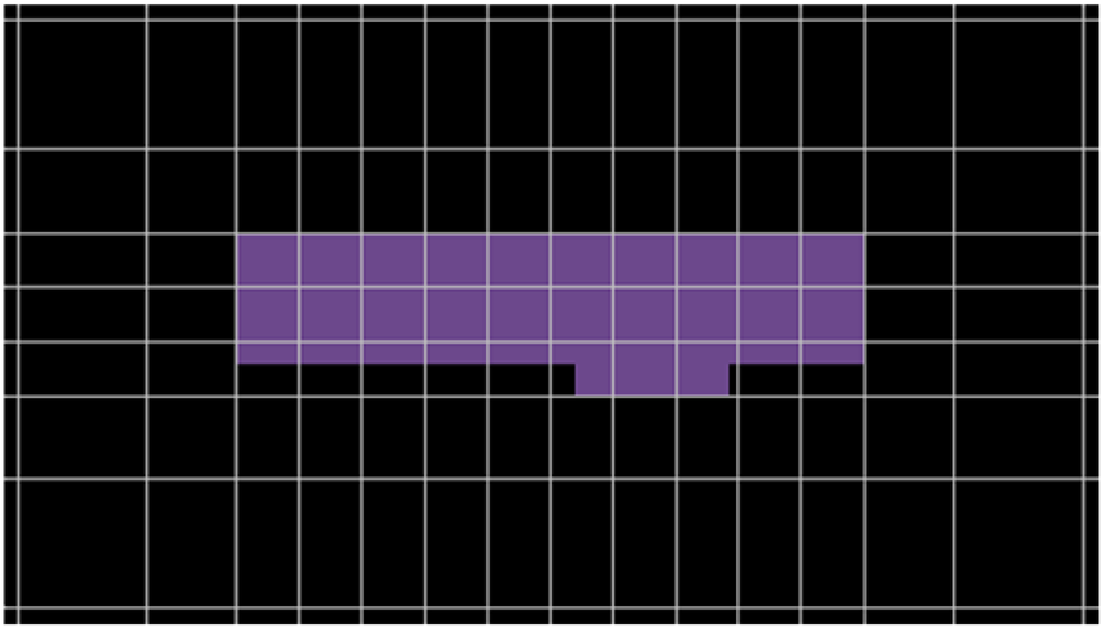
The following figure applies automatic fixed points to the shape (all default settings in the *Gridding Properties Editor* are applied). Note that the edges, like in the case of the applied grid regions, are aligned with the rectangle's edges. The cells within the shape, however, are auto-generated, and therefore vary from the default target cell size as little as possible.

Simple extrusion with applied fixed points



The next illustration applies a user-defined grid region to cover the area of the shape, with cell sizes significantly smaller than the main grid. Note the transition region of cells surrounding the shape. This region contains cells of non-uniform size, which vary from the *Cell Size* defined for the specific grid region to the *Cell Size* defined for the entire grid, at a rate which is limited by the *Maximum Cell Size Step Factor* defined in the *Limit* tab of the *Edit Grid* dialog in *Grid Tools*.

Simple extrusion with a manual grid region defined in Grid Tools



Debugging the Grid

There may be times when one does not understand how certain EMPro grid features in a given grid came to exist. The individual grid features may be turned off so that the grid becomes, in essence, a blank slate. Individual grid features can then be turned back on one at a time. Seeing how the grid changes in this step-by-step fashion makes it possible to locate the cause of any given grid feature.

The following steps outline how to turn off individual grid features.

- Be sure to save a copy of the project and open the copy for use in the debugging process.
- View the mesh and use the MEASURE tool to measure the geometry's largest extent.
- Open the GRID TOOLS dialog:
  - Under the SIZE tab, set the X, Y, and Z TARGET sizes to something just larger than the geometry's largest extent.
  - Set the MERGE sizes for each axis to e.g. 1e-12 mm (not a ratio).
  - Set the *Free Space Padding* cell values to zero or possibly one.
  - Delete all entries under *Fixed Points* and *Grid Regions*.
- If your project includes any *Circuit Components*, ensure that the box labeled *Evenly Spaced In Orthogonal Directions* is unchecked in the *Circuit Component Properties: Properties* tab.

  Be sure to turn this property back on (where applicable) later in the debugging process, as accurate results in many cases depend on even component spacing.

- Right-click on a part under *Project Tree: Parts* and choose *View Flat Parts List*
  - Click on the GR (Grid Region) column heading to bring check-marked parts to the top of the list.
- Right-click on the part in the flat parts list and select *Gridding Properties*. Turn off the part's *Automatic Grid Regions* and/or *Fixed Points*.

    - Click the FP (Fixed Points) column heading to bring check-marked parts to the top of the list.
        - As was done for parts with grid regions above, turn off any remaining parts with automatic fixed points chosen.
- Under the *Grid Tools: Limits* tab, set the *Maximum Cell Step Factor* to something large, e.g. 20000. Set the *Minimum Cell Size* to something very small, (e.g., 1e-12 mm).

After performing those steps the grid should be relatively bare, consisting only of the geometry bounds and padding cells, if any were retained.

Now begin turning on grid features one at a time.

- Choose an important part and turn on its fixed points (in its *Gridding Properties* editor). Turn off all *Automatic Discovery Options* except one, and click *Apply*. Turn off that extraction type and turn on the next one, examining each type in turn.
    - As you examine the fixed points for the part, experiment with its automatic grid region. Turn it on and initially set its *Target* value to the same value as the main grid's target (under *Grid Tools*). Experiment with different *Merge* distances, clicking apply each time. At first, leave the part grid region's target cell size the same as the main grid's target cell size.
    - Choose the best combination of fixed point extraction types and merge distances for the part.
- Go to another important part, in turn, until all parts for which automatic fixed points have been examined. Ensure that automatic fixed points are turned on for those parts and that an appropriate merge distance was chosen for each.
- Review each important part. Experiment with the target cell sizes for each part, setting the target cell size to the value desired for the final grid.

At this point the grid probably contains adjacent cells whose widths vary by more than is allowed by FDTD theory. Go to the *Grid Tools: Limits* tab and change the *Maximum Cell Step Factor* to 2.0, being sure to tab out of the field to make the new setting take effect. Transition regions will appear. You may wish to experiment with values lower than 2.0, although the value must be greater than 1.0. If the *Maximum Cell Step Factor* is too low, the grid will report that a grid could not be created. Choose a higher step factor that is not greater than 2.0.

The grid probably will contain large gaps even after the transition regions have been added. Change the main grid's target cell size to its final desired value to see the remaining gaps filled with grid lines spaced appropriately.

Bibliography

1. K. S. Kunz and R. J. Luebbers, "The Finite Difference Time Domain Method for Electromagnetics". Upper Saddle River: CRC Press, 1993.
2. A. Taflove and S. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method, Third Edition. New York: Artech House Publishers, 2005.
3. C95.3-2002, "Recommended practice for measurements and computations of radio frequency electromagnetic fields with respect to human exposure to such fields, 100khz to 300ghz," IEEE Standards and Coordinating Committee 28 on Non-Ionizing Radiation Hazards, pp. i-126, April 2002.
4. K. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," IEEE Transactions on Antennas and Propagation 14, pp. 302-307, 1966.
5. C. Balanis, Advanced Engineering Electromagnetics. New York: Wiley, 1989. Section 2.8.3.
6. J. R. W. Simon Ramo and T. V. Duzer, Fields and Waves in Communication Electronics. New York: Wiley, 1994. Section 13.12.
7. R. E. Collin, Foundations for Microwave Engineering. New York: Wiley-IEEE P, 2000. Section 6.7.
8. B. Lax and K. J. Button, Microwave Ferrites and Ferrimagnetics. McGraw-Hill, 1962. Sections 4.1, 4.2.
9. F. Kung and H. T. Chuah, "A finite-difference time-domain (fdtd) software for simulation of printed circuit board (pcb) assembly," Progress in Electromagnetic Research, PIER 50, vol. Elsevier book series, pp. 299-335, 2005.

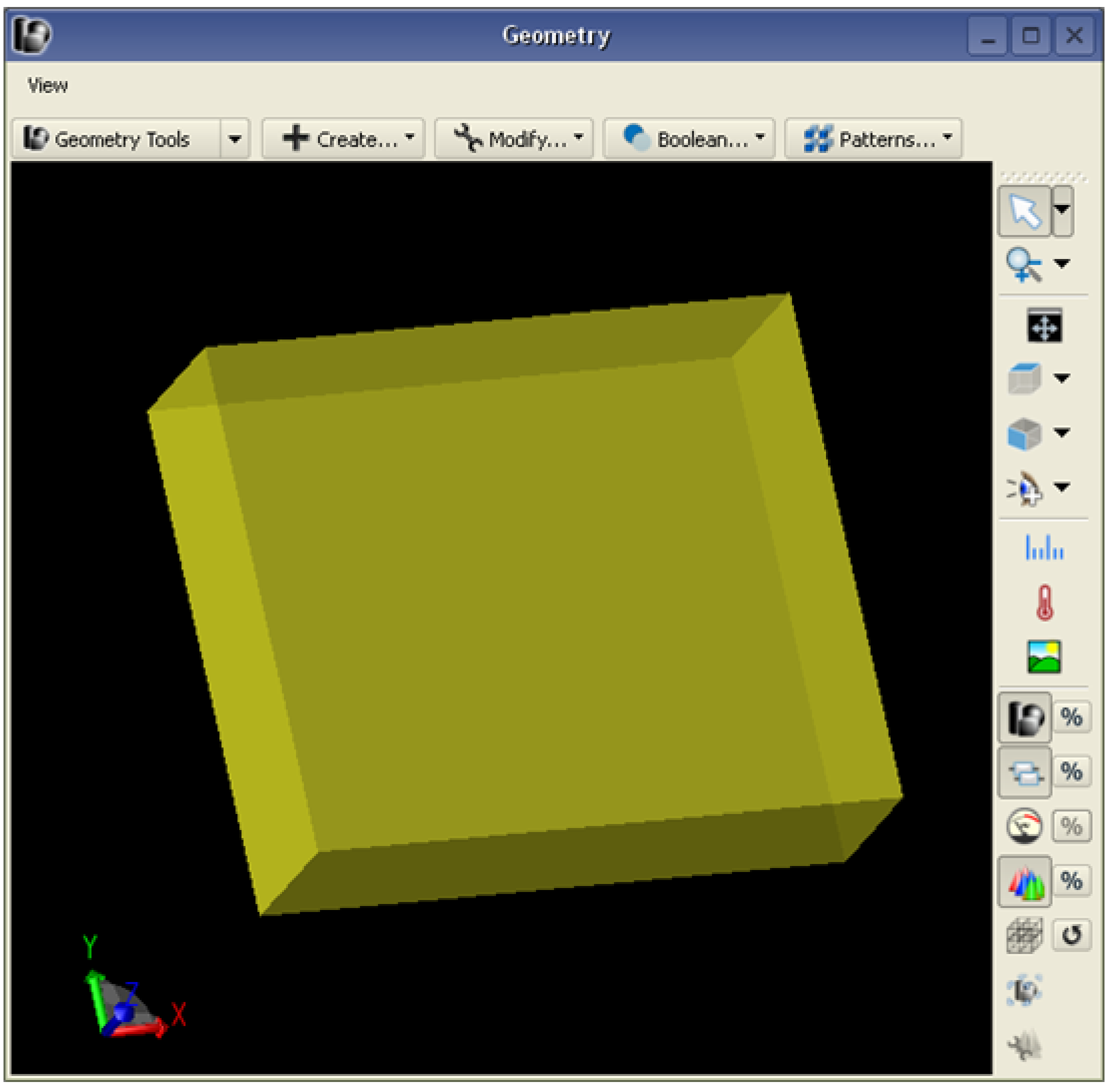
Using the Geometry Workspace

The *Geometry* workspace window provides the graphical interface of the EMPro project. Within the window, there are a series of *View Tools* along the right of the viewing area that can be used to manipulate the view of the simulation space at any time. Along the top of the viewing area, the window contains *Geometry Tools* used to create and edit various aspects of the project geometry, in preparation for the final calculation.

The first drop-down menu in the upper-left part of the *Geometry* workspace window contains four different tools:

- Geometry Tools
- Component Tools
- FDTD Grid Tools
- Sensor Tools

The following figure displays a Geometry workspace window:



This section focuses on geometric modeling within the EMPro interface. It begins by describing the basic functions available within the *Geometry* workspace window, where the project geometry is created.

> **Note**
> For more about creating and editing discrete components with *Component Tools*, refer to *Defining Circuit Components and Excitations* (using).
>
> For more about controlling the characteristics of the grid and meshing parameters with *Grid Tools*, refer to *Defining the Grid and Creating a Mesh* (using).
>
> For more about collecting data in EMPro with *Sensor Tools*, refer to *Saving Output Data with Sensors* (using).

The *View Tools* option which is available for orienting the perspective of the simulation space, is detailed in the following section.

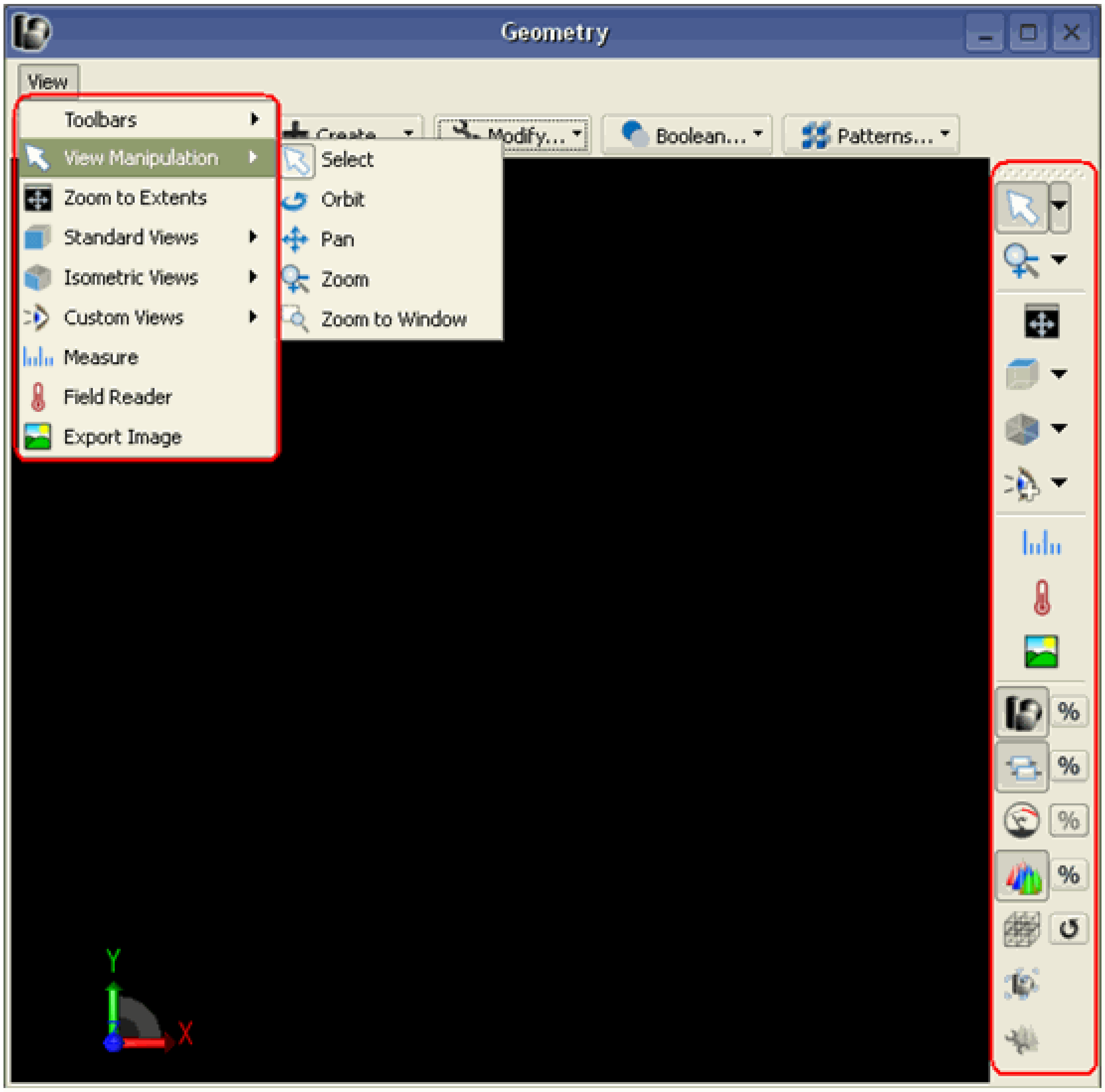
Under *Geometry Tools*, you will be introduced to the 2-D and 3-D modeling tools, modification and boolean operations, and patterned arrays that are available within this dialog. In addition to creating geometry from scratch within Geometry Tools, external files, such as CAD and voxel files, can also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

After the geometry is created or imported into a project, it is often necessary to adjust an object's orientation. EMPro has the capability of orienting not only geometric parts, but also other physical parts, such as components and sensors.

The final section details the *Specify Orientation* tab that is available during any editing session involving a physical part.

View Tools

The *View Tools* are used to alter the perspective of the viewing window by manual rotation, translation, and zoom, as well as automatic orientations to achieve the desired perspective. The View Tools are visible on the right-hand side of the *Geometry* workspace window. They can also be found in the top left-hand corner under the *View* drop-down menu, as shown in the following figure:



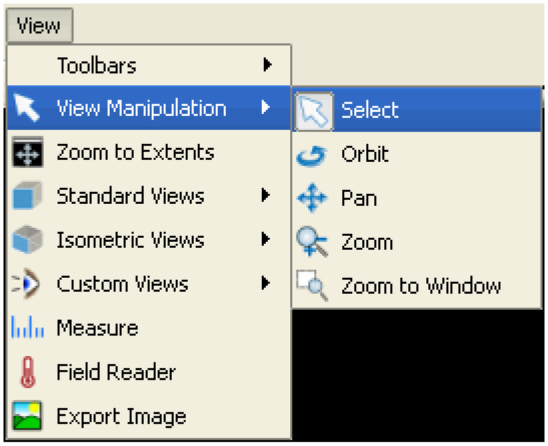
The *View Tools* may be hidden by right-clicking the toolbar and deselecting the toolbar check-box. The toolbar can be unhidden at anytime using the *View* drop-down menu.

Each tool is detailed below.

Using View Manipulation Options

The *View Manipulation* tool provides the following options:

- Select
- Orbit
- Pan
- Zoom
- Zoom to Window



Select

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

- Rotation about a fixed point:
  - Left-click and drag.
  - Click the mouse wheel and drag.
- Translation (panning):
  - Right-click and drag.
  - Hold **Shift**, left- or right-click and drag.
- Zooming:
  - Roll the mouse wheel backwards or forwards (to zoom-in or zoom-out, respectively).
  - Hold **Ctrl**, left-click and drag the mouse up or down (to zoom-in or zoom-out, respectively).

Orbit

The *Orbit* tool is selected to perform rotation of the simulation space through left-clicking-and-dragging.

Pan

The *Pan Tool* tool is selected to perform translation of the simulation space through left-clicking-and-dragging.

Zoom

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

Zoom to Window

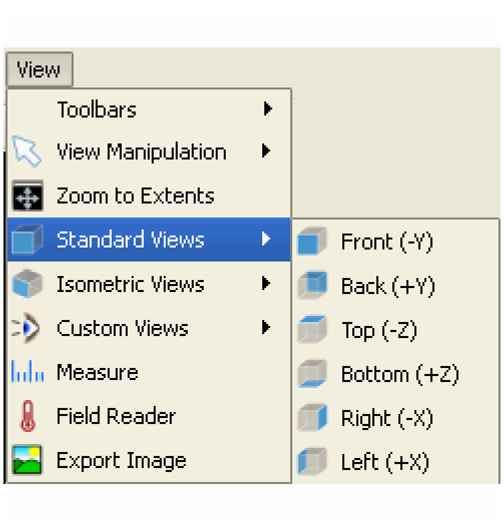
Zoom into a rectangular shaped area of the geometry as specified by the user. To use, select the tool, then left-click and drag the mouse to designate the rectangular zoom area.

Zoom to Extents

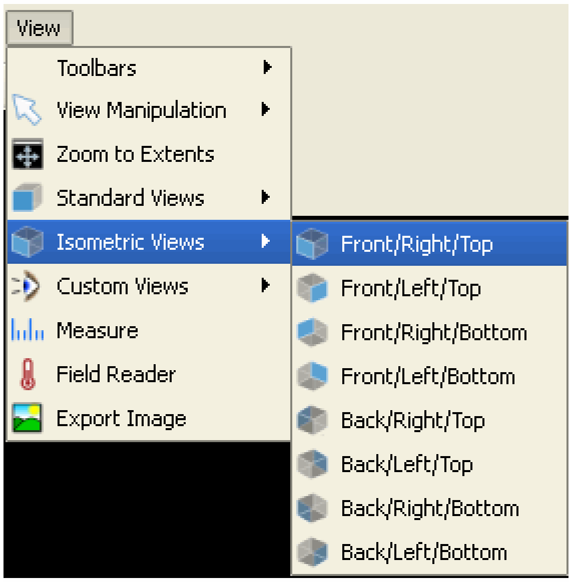
Select this tool to automatically zoom so that the entire geometry can be viewed in the simulation space.

Standard View, Isometric View, and Custom View

The *Standard Views* and *Isometric Views* buttons function to automatically change the perspective of the objects in the *Geometry* workspace window.



Isometric Views



The Standard View changes the view to the following orientations:

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

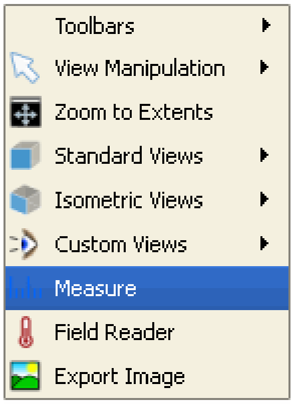
The Isometric View changes the perspective to any combination of these views:

- Front/Right/Top
- Front/Left/Top
- Front/Right/Bottom
- Front/Left/Bottom
- Back/Right/Top
- Back/Left/Top
- Back/Right/Bottom
- Back/Left/Bottom

If these buttons do not achieve the desired perspective, use the *Select*, *Orbit* or *Pan* tools to customize the orientation, and save the desired view by clicking the **Custom Views > Add View** button.

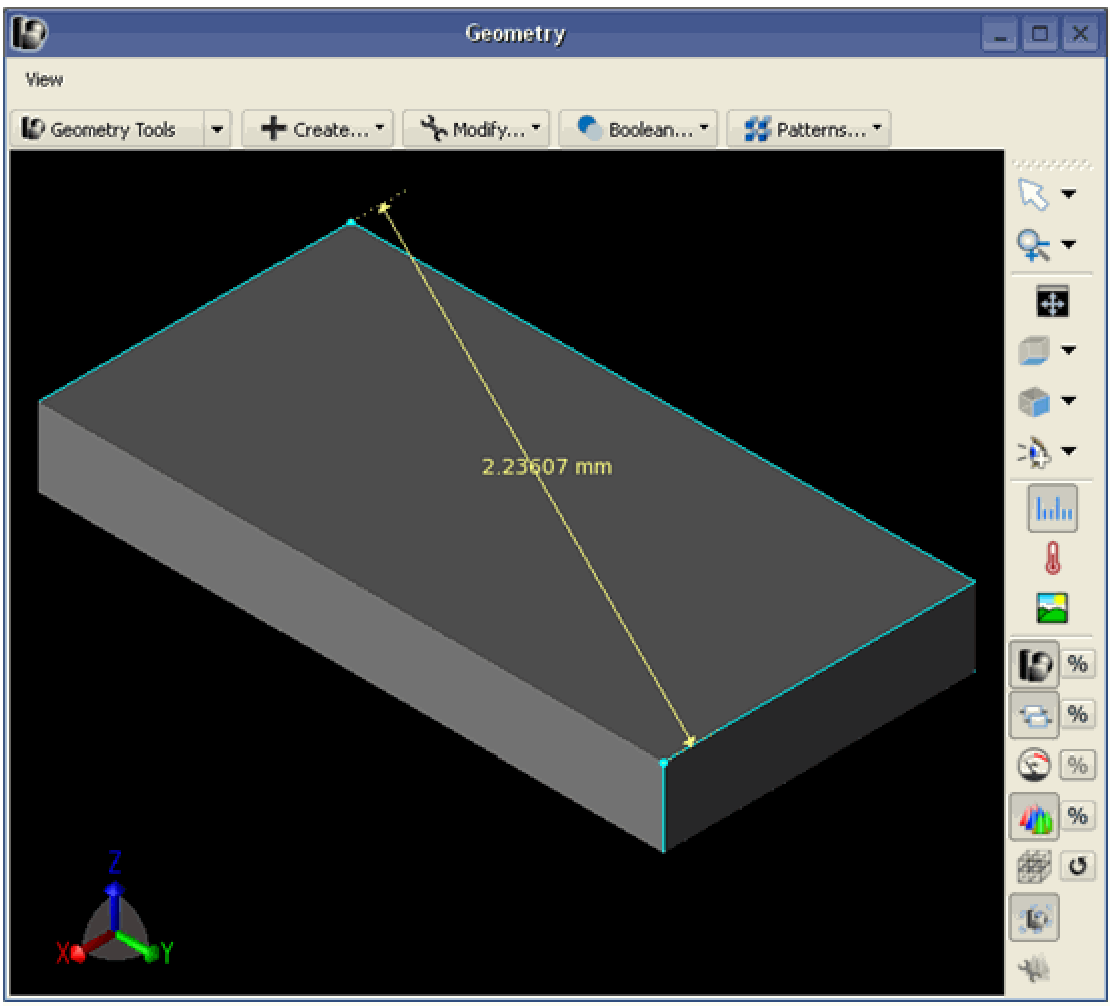
Measure Tool

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



The following illustration shows the Measure Tool calculating the distance between the corners of a rectangle.

Using the Measure Tool



Field Reader Tool

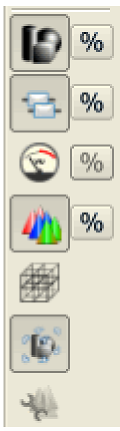
The *Field Reader* tool measures field values at the location where the mouse hovers over the geometry. For more information on the field reader tool, refer to *Viewing Output* (using).

Export Image Tool

The *Export Image* tool takes a screen shot of the geometry as it is currently shown in the *Geometry* workspace window, and saves it to a specified location.

Opacity and Visibility Tools

The *Visibility* buttons control the view of parts of the project.



Clicking any of these buttons will hide its corresponding objects. They include:

- **Parts View** - Toggles the geometric parts on and off.
- **Circuit Components View** - Toggles the circuit components on and off.
- **Sensors View** - Toggles the sensors on and off.
- **Result Fields View** - Toggles the result fields on and off.

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

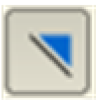
Note
There are several ways EMPro can render this translucency. For more information on how to adjust these settings the notes on *Transparency Algorithm*, refer to *Application Preferences* (using).

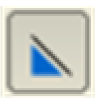
Mesh View

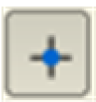
This button toggles between *Mesh View* and Normal View. Alternatively, double-clicking the *FDTD: Mesh* branch of the *Project Tree* will enable *Mesh View*.

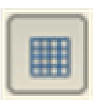
When in Mesh View, there are two main viewing modes, *Mesh Cutplanes* and *3D Mesh*, that are controlled by radial buttons along the bottom of the *Geometry* workspace window. A valid mesh must be generated to use these viewing options. For more information on generating a mesh with the *Meshing Properties Editor*, refer to the *Meshing Properties Editor* (using) in "Defining the Grid and Creating a Mesh".

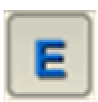
The first mode, *Mesh Cutplanes*, creates cutplanes of the mesh in any or all of the three primary planes. Toggle any of these cutplanes on or off by checking or unchecking their respective boxes. The sliders associated with each of these planes are enabled when its respective plane is turned on. The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

 - Cuts Away solid geometry on the -Z side of the mesh slice.

 - Cuts Away solid geometry on the +Z side of the mesh slice.

 - Toggles viewing edges normal to the slide.

 - Toggles viewing the grid.



- Toggles rendering of all the electric components.

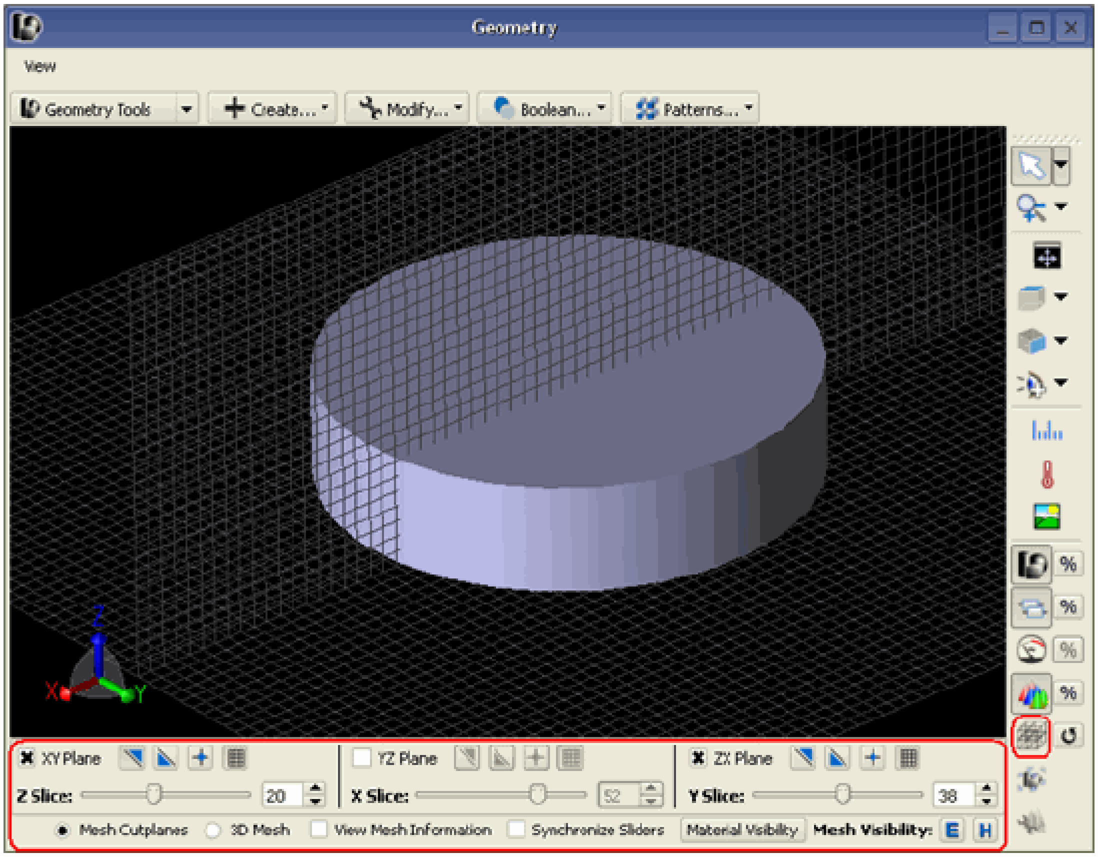
- Toggles rendering of all the magnetic components.

There are also two check boxes available:

- **View Mesh Information** - Displays a dialog box with information about the mesh at the location of the mouse.
- **Synchronize Sliders** - Moves the cutplane simultaneously with a slider adjustment while the mouse button is still pressed.

The following illustration shows this first mode that is displayed when the *Mesh View* icon is selected. Note that this is only a preview of the mesh when it is shown while editing the grid within the *Grid Tools* dialog. Any other time, it is a representation of the most recently generated mesh.

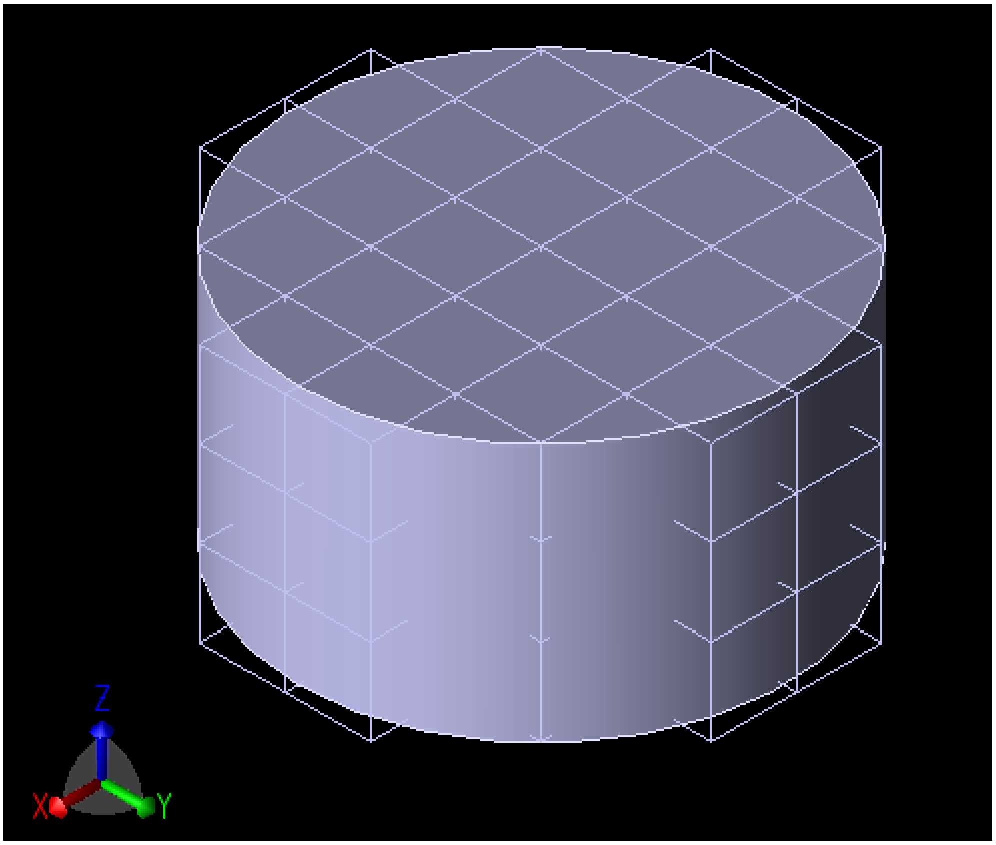
Viewing the mesh



The second mode, *3D Mesh*, provides several different options to view the mesh. Select any of the radial buttons within this option to create a customized three-dimensional view of the mesh. The *Mesh Visibility* icons are available in this mode as well so that *Electrical* and *Magnetic* components of the mesh may be toggled on and off. Additionally, the *Material Visibility* button controls which specific materials are visible in the view.

The figure below shows an object in *3D Mesh* mode with *All Edges* displayed.

Viewing a 3-D mesh



Meshing Options

The *Automatic Remeshing* feature is located in the *View Tools* toolbar. When this feature is enabled, remeshing is performed any time a change is made to the geometry. If this feature is not enabled, remeshing must be performed manually. Automatic Remeshing is not desirable when large meshes are imported because of their memory and performance demands.

The Automatic Remeshing dialog

Toggle Bounding Box Visibility

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

Toggle Output Viewing Controls

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

Geometry Tools

EMPro provides *Feature Based Modeling* that allows the creation of geometric objects as a set of repeatable actions rather than one stringent primitive object. This provides more flexibility in customizing an object and allows any unwanted step to be easily undone by use of the *Undo* button without using excess memory that was formally required to rebuild an entire object. It also tracks every step in the modeling sequence as a separate object in the tree to facilitate even simpler additions, deletions and modifications to the modeling sequence.

This section describes the *Geometry Tools* interface, through which geometric modeling in EMPro is performed. This interface enables the user to create new geometry, modify existing geometry, perform boolean operations such as unions, subtraction, and intersections, and create patterns. To begin using Geometry Tools, open the *Geometry* workspace window and select Geometry Tools from the drop-down menu. A more comprehensive discussion of each Geometry Tool is available in the "*Appendix of Geometric Modeling* (using)".

Creating New Geometry: After selecting *Geometry Tools* in the drop down list of the *Geometry* workspace window, click **Create** to prompt a drop-down menu to appear. This menu includes the modeling operation options. For more information, refer *Creating a New Geometry* (using).

Using Feature-Based Modeling

Parts are created in a step-by-step sequence (such as extrude, revolve, and boolean) that propagate through during creation as the part becomes more complicated. Each step can be re-entered and edited separately.
It facilitates undo/redo operations, parameterization, and constraints

Integrating 2D and 3D Editing

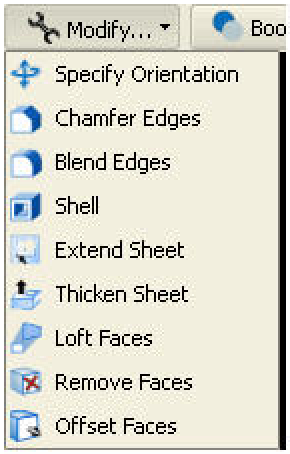
3D objects are created by operations on 2D cross-sections. The global (X,Y, Z) and local (U,V,W) coordinate systems allow easy rotation and translation of objects (connector example).

For more information, refer to *Orienting Objects in the Simulation Space* (quickstart).

Modifying Existing Geometry

You can use the **Modify** menu in the Geometry browser window to modify the geometry of existing objects in the project. This menu provides the following options:

- Create a 3D object
- Translate the object
- Rotate the object
- Chamfer Blend any corner
- Shell Offset Faces from a 3D object



Performing Specific Rotation

Using the Modify menu, you can move and rotate the object with reference to a point:

1. Create an object
2. Select specific rotation menu
3. Move and rotate object

Chamfer edges

For chamfering the edge, perform the followings steps:

1. Create an object.
2. Select Chamfer edges.
3. Perform operation by selecting one edge.

Shell Faces

For shelling the object, perform the followings steps:

1. Create an object.
2. Select Shell Faces.
3. Perform operation by selecting one face.

Performing Boolean Operations

The following Boolean operations need to be tested:

- **Two Parts**: The Two Parts tool provides several boolean operations to subtract, intersect, or unite two objects. To test these feature create two objects, one object must be selected to be the BLANK, and the other the TOOL which acts on the blank.

- **Extrude**: Using the Extrude tool, you can perform an operation on an existing geometry part. In this case, the user chooses the Blank, and then creates the object to use as the TOOL. The user then specifies the orientation of the extrusion and the nature of the operation (Subtract, Intersect, or Union). In essence, this operation is a shortcut for the Two Parts Boolean operation.

- **Revolve**: Using the Revolve tool, you can perform an operation on an existing geometry part. The user chooses the Blank, and then creates the object to use as the Tool. The user then specifies the orientation of the revolution and the nature of the operation (Subtract, Intersect, or Union).

Holes may also be extruded, revolved, or swept through any part any with its respective tool in this menu.

Creating Patterns

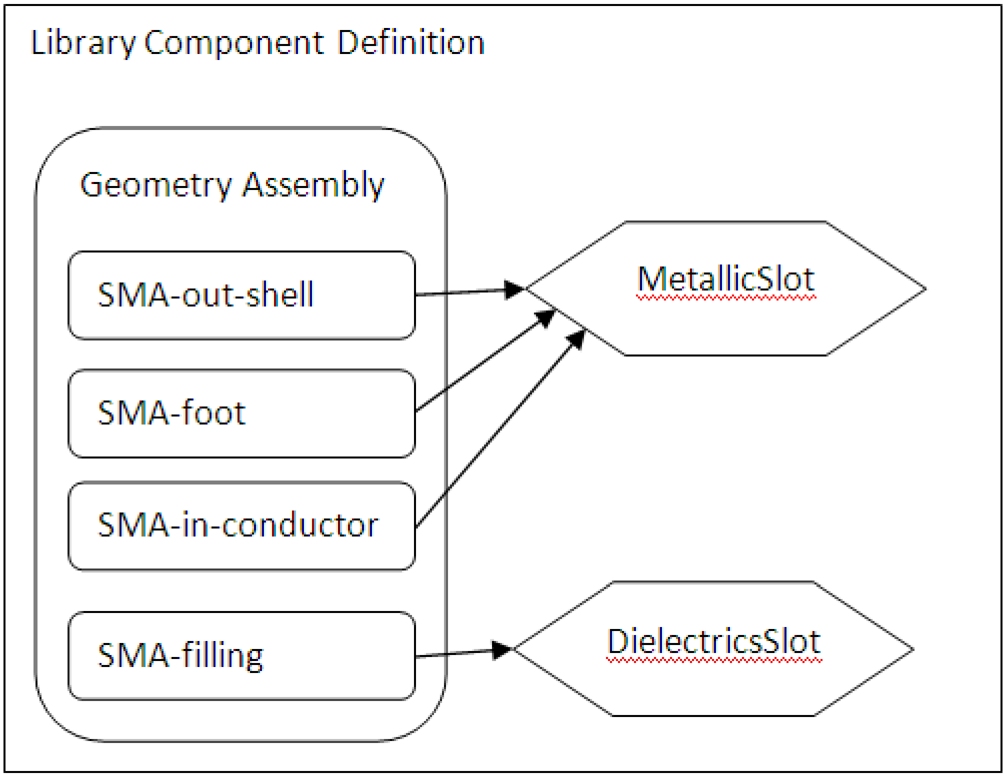
Patterns are created by replicating a single selected object multiple times in one of the organized arrangements listed below:

- Linear pattern
- Cylindrical pattern
- Hex-cylindrical pattern
- Spherical pattern
- Elliptical pattern hex pattern
- Radial pattern
- Polar grid pattern

Using Library Components

A library component is a user-defined object that can be used multiple times in a project. The process of using a library component is called instancing a library component definition. A library component definition is created from a geometry and material slots.

In this section, an example is used to build an SMA library component. The following figure illustrates an SMA library component, where the geometry consists of four geometry elements:

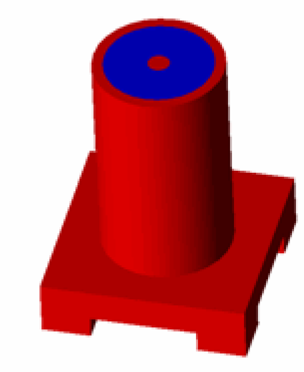


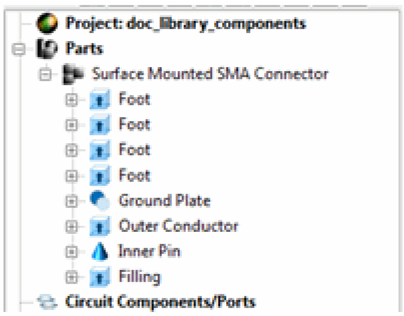
In the SMA library component, three geometry elements share the same material slot called MetallicSlot while the filling of the coax of the SMA has a different material slot called DielectricsSlot. Material slots are used to define a component without fixing the material of it. While instancing a library component, you can decide the material to assign into the slot.

In this example, the predefined surface mounted SMA of the toolkit is used. To insert this into the project, execute the following code in the Script Editor:

```
import empro.toolkit.geometry.connectors.sma
empro.activeProject.geometry().append( empro.toolkit.geometry.connectors.sma.surfaceMountedSMA() )
```

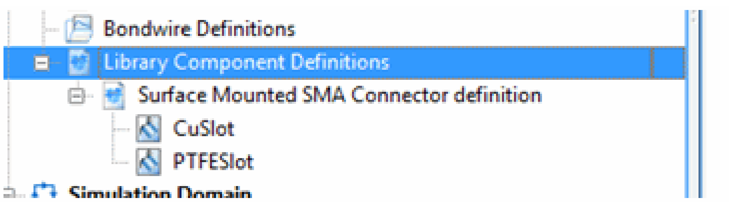
SMA is included in the geometry window, as shown in the following figure:





Creating a Library Component

To create a library component from SMA, select **Surface Mounted SMA Connector** > **Create Definition**. The definition is added to the list of Library Component Definitions, as shown in the following figure:

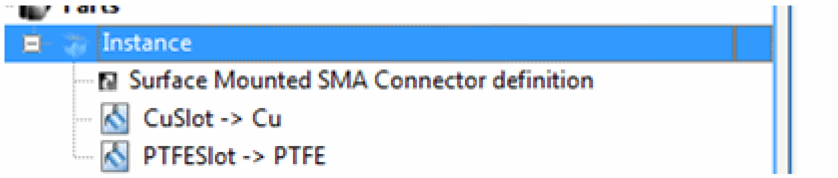


> **Note**
> An alternative way to create a library component is to drag the assembly on the **Library Component Definitions** tree item.

During the creation of the definition, two slots are identified as CuSlot and PTFESlot. These slots are derived from the fact that the original SMA has two materials assigned to it.

Instancing a Library Component

To use the library component definition, select the SMA component definition that was created earlier and drag it to the **Parts** item in the project tree. In the project tree, a new item is created called **Instance**, which consists of a link to the instance definition. The lines below that represent the assignments in the material slots, as shown in the following figure:



Assigning a new material into a slot is done by dragging a material to the slot. When an instance is first created, a default fill-in of the slots is made. This instance searches for materials that have the same name of the slot without the **Slot** suffix and use them. If they are not available, the assignments will be left empty. Unassigned slots will invalidate the instance.

To move an instance around, use the **Specify Orientation** context menu on an instance. You can use multiple number of instances of a definition in a project.

Library Components also serve as the basis for 3D Components. A 3D Component is a library component where the original materials defining the material slot are used. For more information about 3D components, refer to *Creating Parameterized 3D Components* (export).