



Electromagnetics

HFSS™ 15.0

Fields Calculator Cookbook

A BRIEF PRIMER AND COLLECTION OF STEP-BY-STEP CALCULATOR RECIPIES FOR USE IN HFSS FIELDS POST-PROCESSING

June 2012

The information contained in this document is subject to change without notice. Ansoft makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Ansoft shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

© 2012 SAS IP Inc., All rights reserved.

ANSYS

275 Technology Drive

Canonsburg, PA 15317

Tel: (+1) 724-746-3304

Fax: (+1) 724-514-9494

General Information: AnsoftInfo@ansys.com

Technical Support: AnsoftTechSupport@ansys.com

HFSS, HFSS-IE and Optimetrics are registered trademarks or trademarks of SAS IP, Inc. Microsoft and Visual Basic are registered trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

Concerning the use by HFSS 14 of LBNL double-double precision, quad-double precision and arbitrary precision (also termed "multiprecision" or "multiple precision") software:

Copyright (c) 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy)

All rights reserved.

THE LBNL SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUD-

ING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Third-party open source software notice:

Copyright (c) 2000-2001, Aaron D. Gifford

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTOR(S) "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

New editions of this manual will incorporate all material updated since the previous edition. The manual printing date, which indicates the manual's current edition, changes when a new edition is printed. Minor corrections and updates which are incorporated at reprint do not cause the date to change.

Update packages may be issued between editions and contain additional and/or replacement pages to be merged into the manual by the user. Note that pages which are rearranged due to changes on a previous page are not considered to be revised.

Edition	Date	Software Version
1	June 2010	12.0
2	June 2011	14.0

Getting Help

Ansoft Technical Support

To contact Ansoft technical support staff in your geographical area, please log on to the Ansoft corporate website, <http://www.ansoft.com>, click the **Contact** button, and then click **Support**. Your Ansoft sales engineer may also be contacted in order to obtain this information.

E-mail can work well for technical support. All Ansoft software files are ASCII text and can be sent conveniently by e-mail. When reporting difficulties, it is extremely helpful to include very specific information about what steps were taken or what stages the simulation reached. This allows more rapid and effective debugging.

Context-Sensitive Help

To access online help from the HFSS user interface, do one of the following:

- To open a help topic about an HFSS menu command, press **Shift+F1**, and then click the command or toolbar icon.
- To open a help topic about an HFSS dialog box, open the dialog box, and then press **F1**.

Table of Contents

1. Fields Calculator Cookbook

Cautionary Notes	1-2
Field Convergence and Accuracy	1-2
Fast Sweep and Dispersive Models	1-2
Inputs/Excitations	1-3
Units	1-3
Eigensolutions	1-3
Scripting Support	1-3
Online Help	1-3

2. Calculator Interface Basics

Named Expressions:	1-3
Solution Context	1-4
Data Stack	1-4
Stack Commands	1-4
Input Column	1-5
General Column:	1-5
Scalar Column:	1-5
Vector Column:	1-6
Output Column:	1-6
Calculator Stack Quantities	1-6
ScI:	1-6
CSc:	1-6
Vec:	1-7
CVc:	1-7
Geometric Data:	1-7

List	1-7
Calculator Data Extraction (Output Column)	1-8
Output for Post-Processing with the HFSS Report Editor and Fields	
Overlay (Named Expression)	1-8
Access the Value of a Field Quantity (Value)	1-12
Single Numerical Value Output (Evaluate)	1-12
Outputs for Calculation in Future Field Calculator Sessions (Write) ...	1-13
Outputs for Post-Processing outside of HFSS (Export)	1-13

3. **Fields Calculator Recipes**

Calculating the Current along a Wire or Trace	1-3
Calculating the Voltage Drop along a Line	1-5
Calculating Net Power Flow through a Surface	1-7
Calculating the Average of a Field Quantity on a Surface	1-9
Calculating the Peak Electrical Energy in a Volume	1-11
Calculating the Q of a Resonant Cavity	1-13
Calculating the Max Value of a Field Quantity in a Volume	1-16
Calculating Dielectric (or Conduction) Loss in a Volume	1-17
Plotting Wave Impedance along a Line	1-18
Plotting the Phase of E Tangential to a Line/Curve	1-20
Plotting the Maximum Magnitude of E Tangential to a Line/Curve	1-21
Plotting the E-Field Magnitude Normal to a Surface	1-23
Generating an Iso-Surface Contour for a Given Field Value	1-24
Generating an Animation on Multiple Planes with a Positional Variable	1-26
Generating Cartesian/Cylindrical/Spherical-Coordinate Field-Components Library	1-28

Fields Calculator Cookbook

This document contains calculator routines, or “recipes”, for use within the Field Calculator feature of Ansoft HFSS. The field calculator is a very powerful but frequently misunderstood and underutilized tool within the 3D “Fields” Post-Processor.

These routines represent only a small set of the complete capabilities of the calculator. Starting from field data obtained by performing an HFSS solution, the calculator can generate thermal information, voltages and currents, or any other quantity that can be viewed in a 3D environment upon the modeled geometry. This document is intended to give you a head start in using the calculator by codifying some frequently used calculations into easy-to-follow steps. In many cases the steps identified in this document are not the only sequence of operations which can obtain the same results. However, an attempt has been made to identify the routines that require the least number of button clicks and stack manipulations to obtain the desired answer.

[Cautionary Notes](#)

[Calculator Interface Basics](#)

[Fields Calculator Recipes](#)

Cautionary Notes

The following text provides some brief cautionary notes regarding use of the Post-Processor Field Calculator in Ansoft HFSS. Most of the statements below are fairly generalized, and may not apply to all HFSS projects. When in doubt about the applicability of a particular warning for a particular project, please feel free to contact your local HF Applications Engineer for further assistance.

[Field Convergence and Accuracy](#)

[Fast Sweep and Dispersive Models](#)

[Inputs/Excitations](#)

[Units](#)

[Eigensolutions](#)

[Scripting Support](#)

[Online Help](#)

Field Convergence and Accuracy

Ansoft HFSS is a finite element method (FEM) field solver, which arrives upon its solution via adaptive meshing convergence. There are different algorithms available for determining where in each given model mesh adaptation is performed, but convergence is always evaluated by comparison of S-parameters (for driven solutions), changes in overall scattering energy (for incident wave problems) or resonant frequencies (for eigenmode solutions) from pass to pass. Since these quantities represent the results of the model as a whole, they tend to converge more rapidly than the field values. Each point in the modeled space can be said to have converged to some value. As a result, specific field quantities at each mesh point are likely to be less accurate than the overall S-parameter or Eigen frequency result of a project solution.

In order to obtain high accuracy results from calculations on field data, it is advised that you take extra precautions to assure that the model's field data is dependable. These extra precautions might include:

- Running the project to a tighter than usual convergence value.
- Seeding or manually refining the mesh in the areas to be used for calculations .
- Running parametric variations to isolate sensitivity to modeling parameters such as adaptation frequency or circular cross-section facetization.
- Specifying expressions for output convergence.

As long as the accuracy of specific field data points to be used has been assured, the results of the HFSS Field Calculator operations should provide valuable information for your electromagnetic design tasks.

Fast Sweep and Dispersive Models

If an HFSS solution has been performed to include an ALPS Fast Frequency Sweep, the Fields Post-Processor can be tuned to display field data at any point in the frequency band swept. The specific frequency selected for viewing need not even be a precise data point at which the S-parameters were calculated. While field calculator operations may be performed at any frequency to which

the Fields Post-Processor is set, fast sweep solution field data (away from the center frequency of the sweep) may not be as accurate for lossy and dispersive media, within the interior of solid-meshed finite conductors, etc. For higher accuracy under these conditions, field calculator operations should be performed on a full matrix solution completed at the desired frequency.

Since materials assigned as part of a “Perfectly Matched Layer” (PML) model termination are anisotropic and highly lossy, performing field calculations on the surface of or interior to objects designated as PMLs is not recommended.

Inputs/Excitations

Remember to set the field excitation using **HFSS>Fields>Edit Sources** appropriate to the calculation to be performed. In some cases (e.g. FSS calculations) picking the right field solution set (incident, scattered, or total) is also paramount to obtaining the intended result.

Any field calculation which has not yet been completed (such that the calculator stack still shows some form of “text” string rather than a simple numerical value) is merely a placeholder. Altering the field data loaded in the Post-Processor (by altering port excitations, changing frequency, or picking a different solution set using **HFSS>Fields>Edit Sources**) will result in subsequent evaluation of the placeholder to the newly loaded data. To preserve a placeholder’s association to an existing data set before altering the excitation to a different data set, the register stack should be exported using the “Write” button. The correctly associated quantity can be brought back into the stack using “Read” after the field data set selection has been altered.

Units

All units in Driven HFSS field solutions are expressed in the MKS system, regardless of drawing units. Therefore E-mag is always in V/m, H-mag in A/m, etc. The exception is that when plotting along a geometry (e.g. along a line) the dimension along the X axis of the graph shows the position along the line in the drawing units, while the vertical (field quantity) axis will be in the MKS system.

Eigensolutions

Field values in eigensolutions are normalized to a peak value of 1.0, since there is no real excitation to which to scale the internal field results. If desired, the peak value can be scaled to a user-selected number using the **HFSS>Fields>Edit Sources** menu.

Scripting Support

All calculator operations are fully scriptable. A user can save the commands used in a field calculator session by first clicking the **Tools>Record Script to File** menu and replay the same commands by clicking the **Tools>Run Script** menu in a later session.

Online Help

HFSS online help contains more useful information about the field calculator, including detailed descriptions of all the buttons and some examples for basic operations that can be performed with the calculator. We highly recommend users go through the HFSS online help section Post Processing and Generating Reports, Using the Fields Calculator before reading the rest of this document and applying the recipes presented herewithin.

Calculator Interface Basics

Most engineers who use HFSS find that the standard post processing features are sufficient for their work. The scattering parameters, Y or Z matrix, animated field plots and far field patterns cover most of what one needs from such a simulation tool. For those few cases where these are not sufficient the post processor within HFSS includes a Field Calculator. Using this calculator one can perform mathematical operations on all saved field data in the modeled geometry at a single frequency. The resulting quantities can be plotted, tabulated, or exported in ways that are similar to the pre-defined quantities.

To access the Field Calculator: click **HFSS>Fields>Calculator**

The Field Calculator (hereafter referred to as the “calculator”) interface is shown as Figure 1, below.

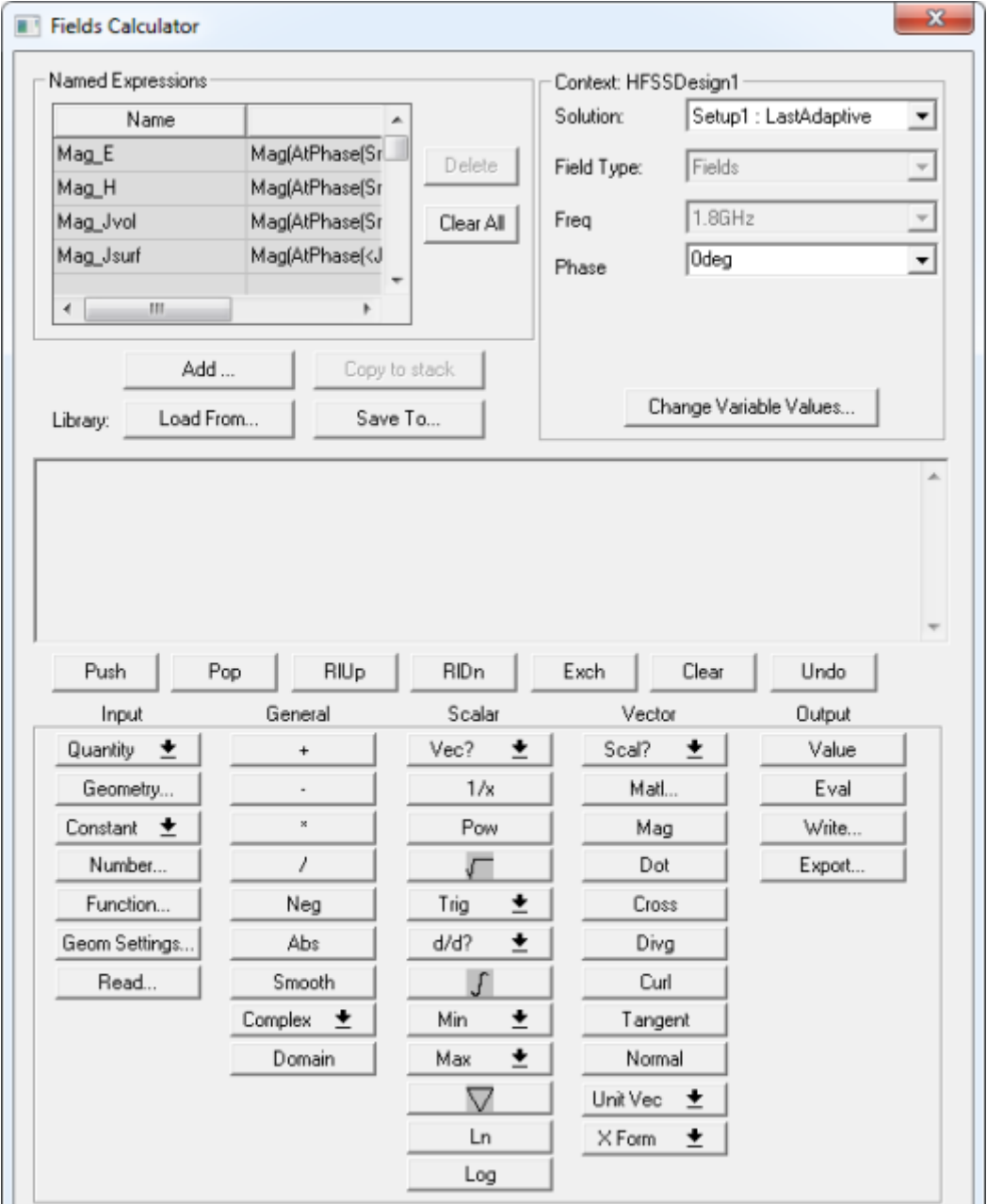


Figure 1: Field Calculator Interface.

At the top-left corner of the calculator is a list of [Named Expressions](#), which are standard or user defined field quantities that are accessible from outside of the calculator. They can be added, copied to stack, saved to, or loaded from a library file using the buttons right beneath the list. At the top-right corner of the calculator is the [Solution Context](#) section, in which users can select the desired solutions, field types, frequency and phase, for the current session. The middle of the calculator contains the [Data Stack](#), in which calculator entries are held in stack registers. The data type in the Data Stack is denoted by its prefix abbreviation. Immediately beneath the stack is the row of [Stack Command](#) buttons that define some basic operations on the data in the Data Stack. The bottom half of the calculator holds the columns containing the actual calculator buttons, organized into columns, classifying them by the type of operation and the type of data upon which the operation can be performed. These columns are headed [Input](#), [General](#), [Scalar](#), [Vector](#), and [Output](#). Each will be discussed in further detail below. At the very bottom of the calculator is the button to exit, **Done**.

Most buttons are single function buttons and their functions are self-explanatory by their names, as shown in Figure 2. However, some calculator buttons are expandable and contains further options as dropdown menus or dialogs. For example, some calculator buttons with a down-arrow symbol on the right side are actually dropdown menus, containing multiple selectable options. Some calculator buttons with an ellipsis [...] symbol will open a dialog that allows users input more detailed or complicated information.

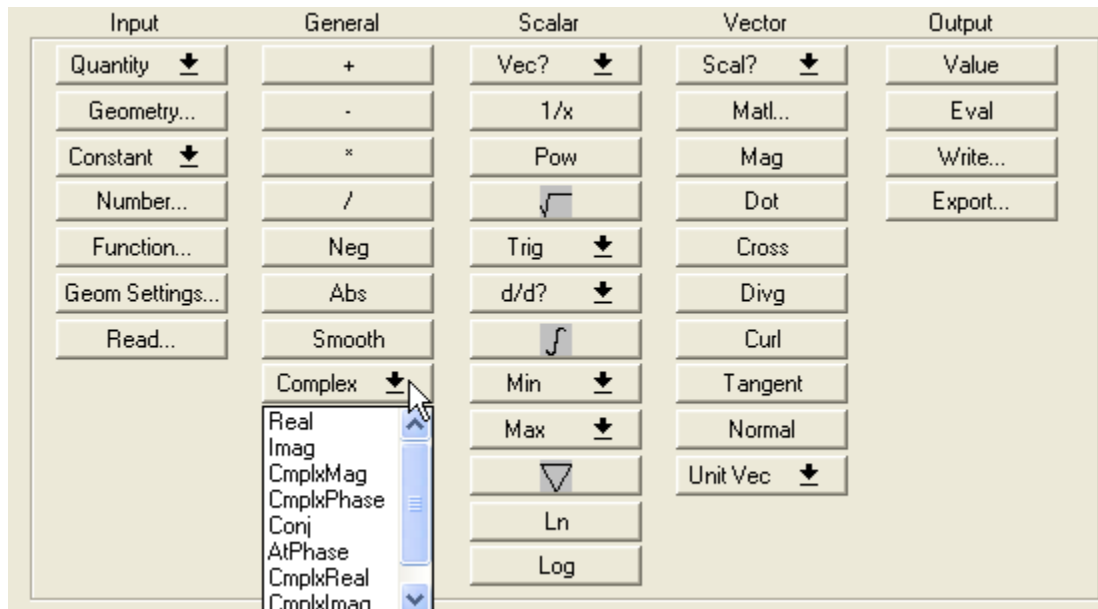
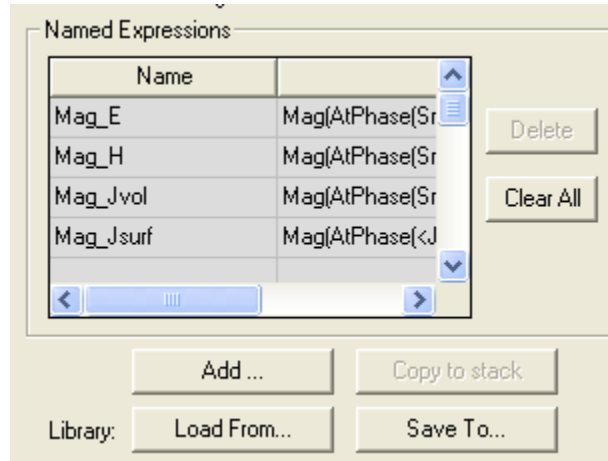


Figure 2: Dropdown Menu “Selection” Example.

Named Expressions:

Named expressions lists defined expressions for use in the field calculator or HFSS’s post-processing. Some standard field quantities have been pre-defined and are always accessible. These field

quantities have a gray background color and cannot be deleted. You can assign a name to the top entry of Data Stack and add it in the Named Expression list. In HFSS V11, the named expression to be added must be a scalar or a vector quantity; while in HFSS V12 and above, the named expression to be added could be a scalar, a vector, or even a complex vector quantity. It will be displayed on a white background color, and user can delete this new added named express any time. Frequently used expressions can be included in a user library that can be loaded in to each project through the **Save to...** and **Load From...** buttons.



Once these new expressions are added/loaded, they are available through HFSS's standard post-process capabilities. For example, they can be copied back to the Data Stack in calculator by "**Copy to Stack**" for further operations, or be plotted using Report Editor and Field Overlay.

Solution Context

Solution Context allows for HFSS solution, frequency, and phase selection. Specially, it permits specification of solution type if outputs to be generated directly from calculator. However, it should be noted that this solution selection context does not replace the function of editing the excitation settings specified in the **HFSS>Fields>Edit Sources** dialog.

Data Stack

Calculator data stack is the buffer for results and operations in progress. It contains current and saved entries in a scrolling stack. General use progresses is from left to right: the input quantity or quantities are left, perform operations are in middle, and defined desired output type at right. The calculator stack register adds to the stack display above preceding entries. Therefore, the entry at the top of the stack represents the last register filled. This convention is opposite to that which many users may be familiar with from the use of hand-held multi-line calculators, which often build their stacks from the bottom up.

Stack Commands

Stack Commands are those commands which influence the entries in the calculator stack and their position. These commands are for manipulating stack contents only, such as copy/paste, delete or

rearrange. The names for these commands match standard stack manipulation conventions. Many are self explanatory, and some can be used in pairs.



For example, **Push** will duplicate the last entry from the stack, while **Pop** deletes only the last entry from the stack. **RIDn** represents “roll down”, which moves the top entry to the bottom, while **RIUp** represents “roll up”, which moves the bottom entry to the top. **Exch** will swap the top two stack entries. **Clear** will empty the stack of all contents, and **Undo** will take back the last operation between stack entries but it may not work for all data type. A full description of all the calculator stack commands can be found in the online help.

Input Column

The Input Column contains all the calculator functions which place new values into the stack, such as field data, constant, geometry data, coordinate system selection, or user-entered vector or complex numbers. Field data (e.g. E-field, H-field, and Poynting vector) for the current project solution is input from the Quantity dropdown menu selection. Other Input dropdown selection should be self-explanatory.

The Quantities specifically available from the calculator are the E-field, H-field, J-vol (volume current) and Poynting vector. All quantities are Peak Phasors, and not RMS quantities, with the phase information captured in the real and imaginary components and the field orientation captured in the vector components. Although the Poynting vector is automatically calculated by the interface as $0.5 (E \times H^*)$, it will appear in the calculator stack as a Complex Vector quantity. The imaginary portion should however be zero or ignored. [See CVC in the section of this document regarding [Calculator Stack Quantities](#), below.]

General Column:

The General Column contains calculator operations which can be performed on many different data types (e.g. vector, scalar, complex, etc.). With the exception of the Complex menu, all are distinct functions. Most are self-explanatory, with the exception of Smooth which performs some data “smoothing” or statistical averaging on the top stack entry.

Scalar Column:

The Scalar Column contains calculator operations which can only be performed on scalar stack entries. Dropdown menus in this column include Vec? (convert scalar to vector), Trig (trigonometric, containing sin, cos, etc. functions), d/d? (derivative with respect to...), Max and Min (self-explanatory). Note that the calculator’s Integrate function is located in the Scalar column. The implication is that integration can only be performed on scalar quantities. To perform integration upon complex quantities, the integration must be performed separately on real and imaginary sub-components.

Vector Column:

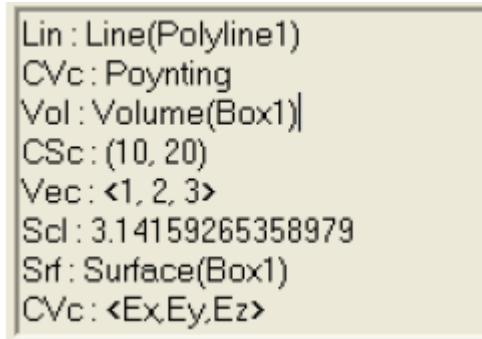
The Vector Column contains calculator operations which can only be performed on vector stack entries. Dropdown menus in this column are Scal? (convert vector to scalar) and Unit Vec (create unit vector). Standard vector algebra operations (Dot, Cross, etc.) are also present.

Output Column:

The Output Column contains those calculator operations that result in final data outputs from calculations. The **Eval** button obtains final numerical results from the last stack placeholder (such as integrations)

Calculator Stack Quantities

The calculator is capable of performing operations on a number of different data types. In many instances, a calculation requires certain type(s) of data to be present in the correct order in the stack register. Many operations result in a different data type than the inputs. In order to show you the type of data contained in each stack entry, calculator denotes its data type by a prefix indicator as shown in Figure 3, below. The following list will describe the definition of each indicator, and provide guidance regarding operations which can convert data from one type to another.



```

Lin : Line(Polyline1)
CVc : Poynting
Vol : Volume(Box1)
CSc : (10, 20)
Vec : <1, 2, 3>
Scl : 3.14159265358979
Srf : Surface(Box1)
CVc : <Ex,Ey,Ez>

```

Figure 3: Stack Contents showing Data Type Indicators (at left)

Scl:

Scl denotes a Scalar quantity. This is a simple numerical value. To convert a scalar to a vector quantity, use the Vec? dropdown menu in the Scalar column. The choices VecX, VecY, and VecZ convert the scalar data to vector data aligned with the X, Y, or Z unit vectors, respectively. You can also multiply the scalar quantity by a desired vector direction entered manually (Num dropdown in the Input column) or obtained using the Unit Vec button from the Vector column. To convert a scalar to a complex quantity, use either CmplxR (assign the scalar value as the real component of a complex quantity) or CmplxI (assign the scalar value as the imaginary component of a complex quantity), both found under the Cmplx dropdown in the General Column.

CSc:

CSc denotes a Complex Scalar quantity. This is a numerical value with real and imaginary components. Convert to a vector quantity using the same techniques described for Scl, above. Convert to a scalar using Real (take the real component), Imag (take the imaginary component), CmplxMag,

(take the magnitude of the complex number) or CmplxPhase (take the phase of the complex number), all within the Cmplx dropdown in the General column.

Vec:

Vec denotes a Vector (non-complex) quantity. Vectors are always evaluated in the coordinate system of the model. To convert a vector quantity to a scalar, use the Scal? dropdown menu from the Vector column. Suboptions ScalarX, ScalarY, and ScalarZ will take the appropriate scalar component of the vector data. Optionally, you can also Dot the vector with another vector to obtain the appropriate scalar result, or use the Tangent (return the tangential scalar component of) or Normal (return the normal scalar component of) operations to relate the vector quantity to a geometric data (Lin, Srf) stack entry. Convert to a Complex quantity using the CmplxR and CmplxI operations described in Scl, above.

CVc:

CVc denotes a Complex Vector quantity. This is a quantity with real and imaginary components for each vector component. In normal calculator usage, the complex nature of the vector components represent the magnitude and phase data of a field quantity, while the vector components themselves represent the orientation of the field quantity in space. Convert to a non-complex Vector as described in CSc, above. Convert the vector to a scalar quantity as described in Vec, above.

Geometric Data:

Geometric data is indicated in the calculator stack by the headers Lin (line), Srf (surface), and Vol (volume). Lines may be straight, curved, or “polylines” in three dimensional space. Lines may also be open (have two endpoints) or closed (ending vertex same as starting vertex). Surfaces need not be planar, and may actually comprise a list of object faces (faces list) as well as planar slices through the entire model space (cutplanes). Volumes may include sets of discontinuous object volumes created as an Object

List

These indicators may exist alone, representing geometric data only, or in combination with one of the categories above, indicating a type of data applied to the geometric entity in question. For example, the notation SclSrf identifies a stack entry containing Scalar data on a Surface geometry set. To select only the portion of a given data entry which exists along, on, or within a given geometry quantity, use the Value button in the Output column of the calculator. Other operations (e.g. integration, or the Normal button) operate when a data quantity is in the second stack register and a geometric quantity is in the top stack register. Full descriptions of the register requirements for each individual command is available in the on-line help.

Note Users must be cautious on what type of data is manipulating and whether or not it is compatible with the desired operation. For example, the integral in calculator is a very commonly misused operation. Note that the integral sign is in the scalar column, implying that to integrate complex number/quantities; user will have to integrate the real and imaginary components separately. In the other words, performing integration on complex number/quantities must be achieved by parts.

Calculator Data Extraction (Output Column)

Data can be extracted from the calculator stack register via a number of different operations. Proper operations should be chosen based on how the extracted data will be used in the future:

1. To plot/tabulate the extracted data using create report or create field overlays, a named expression should be used.
2. To gain access to the value of a field quantity on certain geometry, use the **Value** command.
3. To obtain a single scalar, vector, or complex numerical value, use the **Evaluate** command.
4. To save the content of the stack register for future re-use in a later Field Calculator session, use the **Write** command.
5. To save the extracted data for use in a third-party post-processor outside HFSS, use the **Export** command.

Output for Post-Processing with the HFSS Report Editor and Fields Overlay (Named Expression)

Named expressions are expressions that can be included in the calculator registers by their names. They can be saved to and loaded from a library file, making it possible to reuse the same expression across different HFSS projects. Please note that, starting from HFSS 12.0, named expressions can be defined for expressions that evaluate to not only real scalars and real vectors, but also **complex** scalars and vectors.

However, complex scalar named expressions can only be used in the calculator, not in create reports or create field overlays. Named expressions that evaluate to single-number outputs (per Design Instance) or are in linear graph outputs can be plotted/tabulated using the Report Editor, in a similar way as plotting Matrix Results. The steps for plotting values of named expressions are outlined as follows:

1. Define a field quantity and add it to the list of named expressions in the Field Calculator.
2. Click **HFSS>Results>Create Fields Report>Rectangular Plot**

The **Report Editor** dialog pops up.

The screenshot shows the ANSOFT Report Editor dialog box. The 'Context' tab is active, displaying the following settings:

- Solution:** Setup1 : LastAdaptive
- Geometry:** None
- Points:** 1

Below the 'Context' tab, there is an 'Update Report' section with a checked 'Real time' checkbox and an 'Update' button.

The 'Families' tab is also visible, showing the following settings:

- Trace:** Families Families Display
- Primary Sweep:** Freq All
- X:** ☒ Default Freq
- Y:** Range Function..

The 'Quantity' list shows the following items:

- Variables
- Output Variables
- Calculator Expressions (highlighted)
- Calculator Complex Expre
- Design

The 'Function' list is currently empty.

3. Select a valid line geometry in the "Context" group to activate the Calculator Expressions category. **Or**, select None if the result of the named expression does not depend on the location on a specific geometry.
4. Select "Calculator Expressions" as the "Category" and pick the named expression from the "Quantity" list.
5. Select the appropriate function that will be applied to each value of the numerical expression from the "Function" list
6. Select the appropriate sweeps and variations inside the "Families" tab if necessary
7. Click the "New Report" to create a trace for values of the named expression

As an example, Figure 4 shows what the report editor looks like while trying to create a plot along a line for the named expression "Zwave" that has been defined in the Field Calculator following the recipe shown in Plotting Wave Impedance Along a Line.

Just like any other quantities that can be plotted with the Reporter, the values of named expressions can be exported to a data file using data exporting functions of the Reporter. The number of data points in the file will be equal to the number of Points if a geometric line has selected.

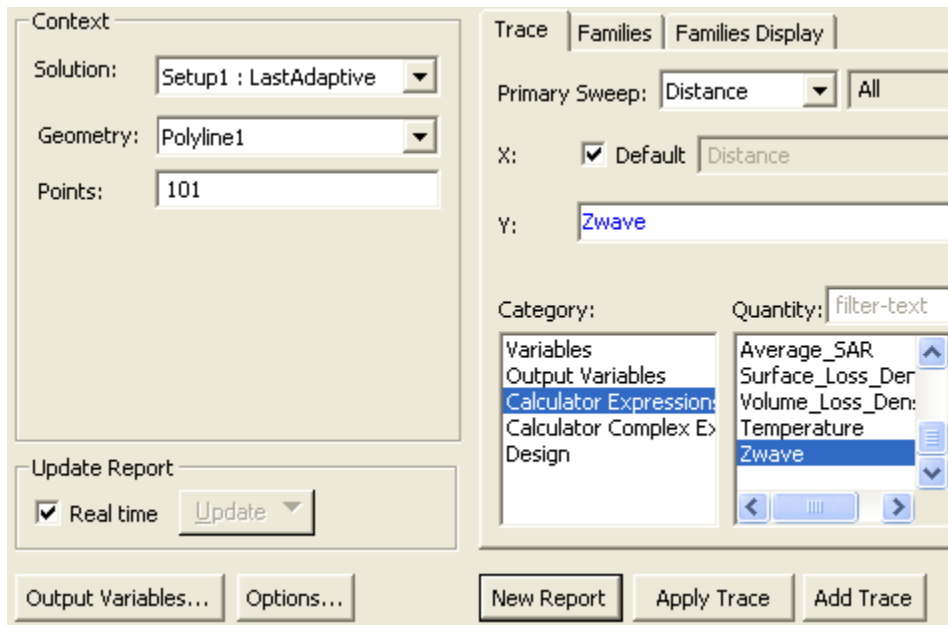
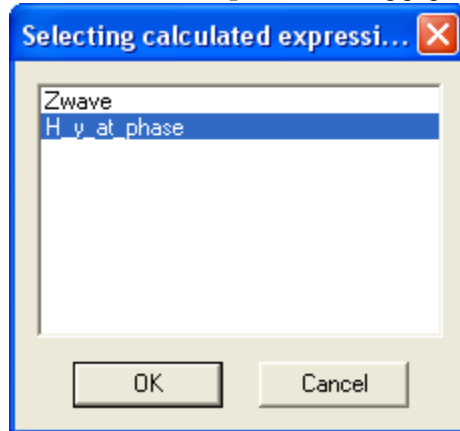


Figure 4 Creating a plot for the named expression "Zwave"

Named expressions can also be used to create Field Overlays in the same manner as with other standard field quantities. The steps for creating a Field Overlay with named expression are as follows,

1. Define a field quantity and add it to the list of named expressions in the Field Calculator
2. Select a geometry in the 3D modeler window
3. Use right mouse button click to bring up the context menu, and choose **Plot Fields>Named Expressions**. Or, menu click from the main program window **HFSS>Fields>Plot Fields->Named Expressions**.

The **Select Named Expression** dialog pops up.



4. Select the named expression to be plotted

The **Create Field Report** dialog pops up.

Figure 5 is the screen capture of the "Create Field Plot" dialog while trying to create a Fields Overlay for the named expression of "H_y_at_phase", which is a scalar representing the value of the H field y component at a given phase.

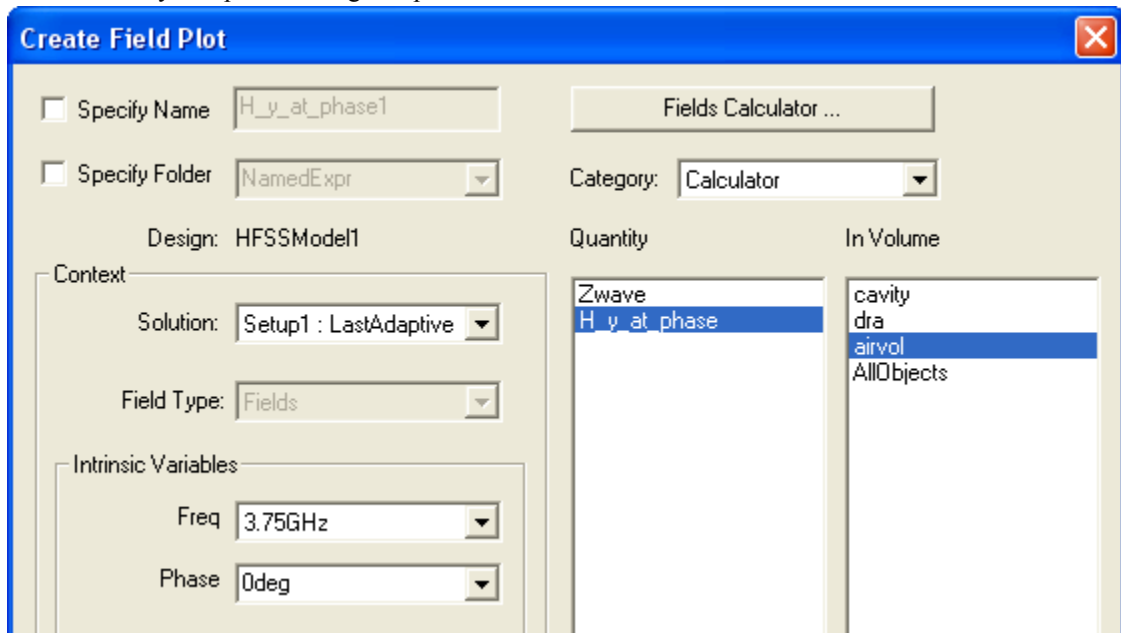


Figure 5 Create Field Plot dialog

5. Select the desired frequency (Freq) and phase in the **Create Field Plot** dialog
6. Click **Done** to create the Fields Overlay plot.

Access the Value of a Field Quantity (Value)

The **Value** command applies the value of the next-to-top stack entry (a field quantity) on the geometry entry at the top stack entry. The field quantity entry may be scalars, real or complex vectors. The geometry entry may be of any type (point, surface, line, or volume).

For example, if the top stack entry in the Field Calculator is a surface and the second stack register is for the real part of the E Vector ("Real(<Ex, Ey, Ez>)"), the resulting quantity will be an **expression** for the real part of the E vector on the surface (a VecSrf quantity). Again, the result of this command is still an expression, i.e., a placeholder, not the final numerical values.

Strictly speaking, the **Value** command provides access to the field quantity, not data outputting. It simply provides a handle to the (numerical) values of a field quantity on a given geometry, and thus the outcome of the command is not numerical values themselves. If the numerical value of a field quantity is desired, you will need to follow it up with one of the three other output commands, or define a named expression for the quantity and plot it with the Report Editor.

Single Numerical Value Output (Evaluate)

The **Evaluate** command finalizes computations and converts the text-string indicating the computation being performed to an actual numerical value. For this command to work properly, the stack entry must be reduced to a single-value entry which can be a scalar, complex, or vector (**real or complex starting from v12.0**). Units of the numerical value are not provided.

For a calculator expression that evaluates to a single value, the **Evaluate** command offers a quicker way to obtaining the final numerical result than defining a named expression and then trying to plot/tabulate it in the reporter. But for many expressions that don't evaluate to single values, such as the wave impedance along a line mentioned earlier, Evaluate command does not apply. As such, use of named expression and the Report Editor will be the only option.

Figure 6 shows the definition of an expression for the voltage between two conductors by integrating the electric field along a line using the recipe given for Calculating the Current along a Wire or Trace. Then, by clicking the **Evaluate** command in the Output column, we obtain the numerical

value for the voltage as shown in the top of the stack. It can then be either written down or copied/pasted to other programs.

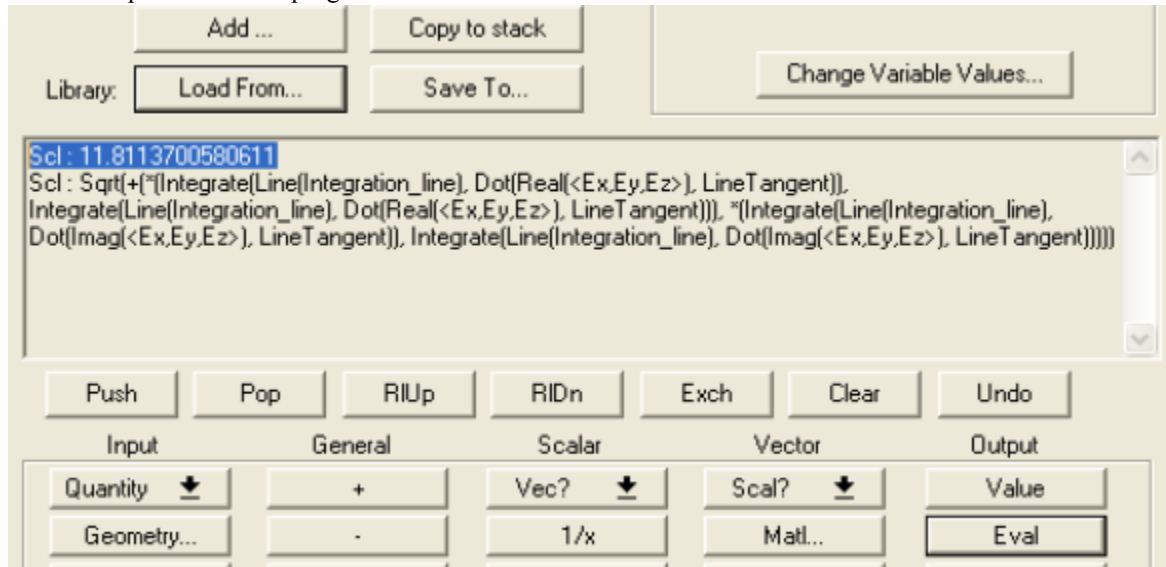


Figure 6 Evaluate the expression for voltage along a line

Outputs for Calculation in Future Field Calculator Sessions (Write)

If the calculator operations performed have obtained a stack entry that is intended for use in still other calculator operations, the stack entry can be saved for this purpose by using the **Write** button in the calculator Output column. Note that in this case no name will be requested for the expression; only a save filename location will be required. This function will not work for field values derived upon a specific geometric quantity (those containing either Lin, Srf, or Vol in the stack data type indicator) as the calculator cannot know that these geometric quantities exist in identical forms in other post-processing sessions.

When you try to read a file written with the **Write** command by clicking the **Read** button in the Input column, you must ensure that the mesh used for generating the current solution is the same as the one with which the saved field data is generated. Since the two results must be generated on the same mesh, not only the geometries of the designs on which the **Write/Read** commands were applied must be the same, they must also have gone through the same adaptive meshing process. In practice, this limits the use of **Write/Read** command to solutions for different frequency sweeps of the same analysis (adaptive) setup of the same design.

Outputs for Post-Processing outside of HFSS (Export)

To output a field quantity or calculation result for use by some third-party post-processors, use the **Export** button in the calculator Output column. You can map the field quantity to either a customized grid of points specified by a points file or a three-dimensional Cartesian, Cylindrical, or Spherical grid specified interactively through a dialog box. In the latter case, you must specify the

dimensions and spacing of the grid in the coordinate system with units. For Cylindrical and Spherical coordinate systems, you can also specify an offset from the origin.

Export Solution

Output file name:

Grid points on which to export ☒ Include points in output file

☒ Input grid points from file

☐ Calculate grid points

Coordinate System: ☒ Cartesian ☐ Cylindrical ☐ Spherical

Offset: 0 mm 0 mm 0 mm

	Minimum	Maximum	Spacing
X	<input type="text"/> mm	<input type="text"/> mm	<input type="text"/> mm
Y	<input type="text"/> mm	<input type="text"/> mm	<input type="text"/> mm
Z	<input type="text"/> mm	<input type="text"/> mm	<input type="text"/> mm

OK Cancel

The format for the points in a customized grid is simply X, Y, and Z coordinate location values of a point, one per row, in space-delimited form with no header or row numbering required. The unit of the coordinate values should be specified in the first line of the file as "Unit=xx", where xx is length unit, e.g., mm. The points are not required to be in a particular order, but it is recommended that neighboring points are written next to each other as doing so will aid the point searching algorithm and make the process run faster. The content of a sample input file copied from the online help is shown below.

```
Unit = mm
-5.5 -5.5 -5.21475
-5.5 -5.5 -5.14425
-5.5 -5.5 -5.07375
-5.5 -5.5 -5.021
```

Fields Calculator Recipes

The following pages contain calculator recipes for deriving a number of commonly used output parameters from solved HFSS projects.

Calculating Numerical Quantities

- Calculating the Current along a Wire or Trace
- Calculating the Voltage Drop along a Line
- Calculating the Net Power Flow through a Surface
- Calculating the Average of a Field Quantity on a Surface
- Calculating the Peak Electrical Energy in a Volume
- Calculating the Q of a Resonant Cavity
- Calculating the Max Value (or Position) of a Field Quantity in a Volume (or Surface)
- Calculating Dielectric (or Conduction) Loss in a Volume

Calculating Quantities for 2D (Line) Plot Outputs

- Plotting the Wave Impedance along a Line
- Plotting the Phase of E Tangential to a Line/Curve
- Plotting the Maximum Magnitude of E Tangential to a Line

Calculating Quantities for 3D (Surface or Vector) Plot Outputs

- Plotting the E-Field Magnitude Normal to a Surface

Calculating Quantities for 3D (Volume) Plot Outputs

- Generating an Iso-Surface Contour for a Given Field Value

Calculating Quantities for Animated Outputs

- Generating an Animation on Multiple Planes with a Positional Variable

Creating User Defined Named Expressions Library

[Generating Cartesian/Cylindrical/Spherical Coordinates Field Components Library](#)

Recipe Format

Each calculator recipe will be provided in the format shown below:

EXAMPLE: Title of Current Calculation

Description:

The first paragraph will give a brief description of the calculation's intent.

Usage Example(s):

The second paragraph will give an example of a project type on which the calculation might be useful. It may also comment upon the reasons such a calculation might be of interest.

Prerequisites:

The third (optional) paragraph will indicate what must be present before doing the calculator operations, e.g. if certain geometry (lines, faces lists, etc.) need to be generated to use in calculations.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Each button click shown as a step	resulting stack entry/type
Dropdown Menu>Submenu pick also shown as a single step	Sci: {placeholders for numerical results shown in brackets}
Button steps requiring data entry will have entry quantity shown in {brackets}	resulting stack entry/type (<i>notes follow in italics</i>)

Calculating the Current along a Wire or Trace

Description:

Obtains the full complex current in a wire or trace conductor (e.g. microstrip, stripline) at a specific location by integrating the magnetic field along a closed path encircling the conductor. .

$$I = \oint_l \vec{H} \cdot d\vec{l}$$

where l is a closed path, which could be a circled line object

Usage Example(s):

To find the current distribution along a wire (dipole, monopole, etc.) antenna, this calculation could be repeated at periodic positions along the length of the antenna.

Prerequisites:

You must create a closed line for the integration path using **Draw>Line** before beginning calculator operations. The line must be orthogonal to the direction of current flow, should not intersect the wire/trace, and should not be too much bigger than the wire/trace.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>H	CVc : <Hx,Hy,Hz>
Complex>Real	Vec : Real(<Hx,Hy,Hz>)
Geometry>Line...>{select line}	Lin : Line (line1) (user line name may differ from example)
Tangent	ScLin: LineValue(Line(...),Dot(Real<Hx,Hy,Hz>), LineTangent))
∫	Scl : Integrate(Line(...
Complex>CmplxReal	CSc : CmplxR(Integrate(Line(Line1),Dot(...)))
Quantity>H	CVc : <Hx,Hy,Hz>
Complex>Imag	Vec : Imag(<Hx,Hy,Hz>)
Geometry>Line...>{select line}	Lin : Line (line1) (user line name may differ from example)
Tangent	ScLin: LineValue(Line(...),Dot(Imag<Hx,Hy,Hz>), LineTangent))

Calculator Operation	Resulting Stack Display (top entry only unless noted)
\int	Scl : Integrate(Line(...
Complex>CmplxImag	CSc : CmplxI(Integrate(Line(Line1,Dot(...)))
+	CSc: (CmplxR(Integrate(Line(Line1),Dot(...))), CmplxI(Integrate(Line(Line1),Dot(...))))
Eval	CSc : {complex numerical value} <i>(Final complex current result)</i>

Calculating the Voltage Drop along a Line

Description:

Provide the complex voltage drop, in volts between two points by integrating the E-field along a line.

$$V = \int_l \vec{E} \cdot d\vec{l}$$

where l is a path between two points on which voltage difference are measured. Usually it is a straight line object.

Usage Example(s):

To find the voltage excited across the width of a slot antenna element; to test whether a voltage exceeds breakdown in a particular dielectric media.

Prerequisites:

You must create the line along which the E-field is to be integrated using **Draw>Line** before you can complete the calculator routine.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>E	CVc : <Ex,Ey,Ez>
Complex>Real	Vec : Real(<Ex,Ey,Ez>)
Geometry>Line...>{select line}	Lin : Line (line1) <i>(user line name may differ from example)</i>
Tangent	ScLin: LineValue(Line(...),Dot(Real<Ex,Ey,Ez>), LineTangent))
∫	Scl : Integrate(Line(...
Complex>CmplxReal	CSc : CmplxR(Integrate(Line(Line1),Dot(...)))
Quantity>E	CVc : <Ex,Ey,Ez>
Complex>mag	Vec : Imag(<Ex,Ey,Ez>)
Geometry>Line...>{select line}	Lin : Line (line1) <i>(user line name may differ from example)</i>
Tangent	ScLin: LineValue(Line(...),Dot(Imag<Ex,Ey,Ez>), LineTangent))
∫	Scl : Integrate(Line(...

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Complex>CmplxImag	CSc : CmplxI(Integrate(Line(Line1),Dot(...)))
+	CSc: (CmplxR(Integrate(Line(Line1),Dot(...))),CmplxI(Integrate(Line(Line1),Dot(...))))
Eval	CSc : {complex numerical value} (<i>Final complex voltage result</i>)

Calculating Net Power Flow through a Surface

Description:

This recipe allows calculation of power flow through an open or closed surface by integrating the Poynting vector normal to that surface.

$$W = \int_S \text{Re}(\vec{P}) \cdot \vec{n} dS$$

where S is the surface that is used to calculate the power, and \vec{n} is the normal vector to the surface S .

Usage Example(s):

This calculation could be used on scattered field data resulting from an incident wave excited HFSS project to evaluate reflection from a radome filter or FSS (frequency selective surface). It might also be used on the closed exterior surface of a solid volume to determine power dissipation within the volume (due to conservation of energy, what goes in a closed surface must come out, unless there is a loss or storage [e.g. standing wave or resonance] mechanism).

Prerequisites:

The surface on which the integration is to be performed must exist prior you can complete the calculation. If the surface is the exterior of a solid object, no customer geometry creation is necessary. If the surface is only a subset of an object's faces, or a slice through the entire plane of the model not already defined by a separate 2D entity, then you must create a Faces List and/or Cutplane to represent the integration location.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>Poynting	CVc : Poynting
Complex>Real	Vec : Real(Poynting) (discards the unneeded imaginary component)
Geometry>Surface...>{select surface}	Srf : Surface(Facelist1) (above is example; user surface shown may vary)

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Normal	SclSrf : SurfaceValue(Surface(Facelist1), Dot(Real(Poynting), SurfaceNormal) <i>(takes the dot product of the vector data with the normal to the surface(s) selected)</i>
\int	Scl : Integrate(Surface...)
Eval	Scl : {numerical value} <i>(final answer in watts)</i>

Calculating the Average of a Field Quantity on a Surface

Description:

This recipe permits you to calculate the average of a field quantity on a Surface geometry, by dividing the Integration of the field value on the surface by the surface area.

Usage Example(s):

This calculation could be used to determine the average phase of the E-field at a given cutplane through a project, to find the average current on a trace surface, or to calculate the average H-field tangential to a 2D object used as an aperture. The specific example steps below will be for the first usage example mentioned (average phase of an E-field on a surface), but the format for integration on a surface and for finding the area of the surface is identical for the other applications as well.

Prerequisites:

The surface on which the integration is to be performed must exist. If the surface is the exterior of a solid object, no customer geometry creation is necessary. If the surface is only a subset of an object's faces, or a slice through the entire plane of the model not already defined by a separate 2D entity, then you must create a Faces List and/or Cutplane to represent the integration location.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>{select field quantity}	CVc : <Ex, Ey, Ez> (E-field used as example)
{Derive desired scalar field data For example: Scal?>ScalarX Complex>CmplxPhase}	CSc : ScalarX(<Ex, Ey, Ez>) (first operation result) Scl : Phase(ScalarX(<Ex, Ey, Ez>)) (second operation result)
Geometry>Surface...>{select surface}	Srf: Surface(plane1) (user surface shown may vary)
Value	SclSrf : SurfaceValue(Surface(plane1), Phase(ScalarX(<Ex, Ey, Ez>)))
\int	Scl : Integrate(...)
Geometry>Surface...>{select surface}	Srf : Surface(plane1)
Unit Vec>Normal	Vec : SurfaceNormal
Geometry>Surface...>{select surface}	Srf : Surface(plane1)
Normal	SclSrf: SurfaceValue(Surface(plane1)... (takes the dot product of the surface with its own normal)

Calculator Operation	Resulting Stack Display (top entry only unless noted)
\int	Scl : Integrate(Surface(...
/	Scl : /(Integrate(SurfaceValue(...
Eval	Scl : {numerical value} (for this example units are in <i>deg</i> or <i>radians</i>)

Calculating the Peak Electrical Energy in a Volume

Description:

This recipe permits you to calculate the peak electrical energy in a volume object. The solution is achieved by integrating $\mathbf{E} \cdot \mathbf{E}^*$ within the volume.

$$W = \int_V \frac{\epsilon_r \epsilon_0 (\vec{E} \cdot \vec{E}^*)}{2} dV$$

where V is the volume.

Usage Example(s):

This calculation could be used to determine the average total energy with respect to time in a terminating resonant cavity. (In a sealed, one-port structure at resonance, energy is converted back and forth between the electrical and magnetic fields, but maintains the same total quantity; therefore the peak electrical energy is equal to the average total energy.)

Prerequisites:

The volume object which the integration is to be performed must exist before the computation can be completed. If the volume for integration consists of the volume of several drawing objects, you must create a single list entry representing their combined volumes using **Modeler>Lis>Create>Object List**.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>E	CVc : <Ex, Ey, Ez>
Complex>Conj	CVc : Conj(<Ex, Ey, Ez>)
Quantity>E	CVc : <Ex, Ey, Ez>
Dot	CSc : Dot(Conj(<Ex, Ey, Ez>), <Ex...
Complex>Real	Scl : Real(Dot(Conj(<Ex, Ey, Ez>), ... <i>(note: the dot product of the E with its conjugate should lead to a real quantity, but the calculator still assume as complex)</i>
Geometry>Volume...>{select volume}	Vol : Volume(box1) <i>(above is example, user entry may differ)</i>
∫	Scl : Integrate(Volume(...
Eval	Scl : {numerical quantity}

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Constant>Epsi0	Scl : 8.854187817E-012
Number>Scalar>{enter ϵ_r for volume}	Scl : {numerical quantity}
*	Scl : {numerical quantity} <i>(stack entry is volume ϵ)</i>
Number>Scalar>0.5	Scl : 0.5
*	Scl : {numerical quantity}
*	Scl : {numerical quantity} <i>above is electrical energy in joules</i>

Calculating the Q of a Resonant Cavity

Description:

This recipe permits you to calculate the Q in a homogeneous dielectric-filled cavity with uniform wall losses, using the equation:

$$Q_u = \frac{\int_{\Omega} |H|^2 d\Omega}{\frac{s}{2} \oint_{\Gamma} |n \times H|^2 d\Gamma + tg\delta \left(\int_{\Omega} |H|^2 d\Omega \right)}$$

where s is skin depth, $tg\delta$ is dielectric loss tangent, n is the surface normal for the cavity wall faces, and Γ and Ω represent wall surface area and cavity volume, respectively.

Usage Example(s):

To calculate the Q of an air- or solid-dielectric filled cavity, fed with a below-cutoff port aperture, or obtained via an eigen solution.

Prerequisites:

The Object (or Object List) representing the cavity total volume must already exist, as must the Face List corresponding to the total wall surface area of the cavity. You can create both via the Modeler menu. The solution should be tuned to the desired resonant frequency for evaluation.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>H	CVc : <Hx, Hy, Hz>
Push	<i>(above entry duplicated)</i>
Complex>Conj	CVc : Conj(<Hx, Hy, Hz>)
Dot	CSc : Dot(<Hx, Hy, Hz>, Conj(...
Complex>Real	Scl : Real(Dot(<Hx, Hy, ...
Geometry>Volume>{select cavity volume}	Vol : Volume(cav_total) <i>(above is example; user entry may differ)</i>

Calculator Operation	Resulting Stack Display (top entry only unless noted)
\int	Scl : Integrate(Volume(cav... (above represents energy stored in cavity)
Push	(above entry duplicated)
Number>Scalar>{enter loss tan for volume}	Scl : {numerical value} (loss tangent for dielectric fill in cavity)
The above equation is only valid for cavities filled with one dielectric material across the entire volume. For cavities with different dielectric fills (e.g. a dielectric resonator within a larger metal cavity), dielectric loss must be evaluated using integration by parts for each dielectric material volume. The equation also assumes the same conductivity for all walls, and no nonreciprocal (e.g. ferrite) property to either walls or fill.	
*	Scl : *(Integrate(Volume(... (above represents energy lost in dielectric material losses)
Quantity>H	CVc : <Hx, Hy, Hz>
Geometry>Surface>{select cavity surfaces}	Srf : Surface(cav_tot_faces) (above is example; user entry may differ)
Unit Vec>Normal	Vec : NormalSurfaceNormal
Cross :	Cross(<Hx, Hy, Hz>, SurfaceNormal)
Push	(above entry duplicated)
Complex>Conj	CVc : Conj(Cross(<Hx, Hy, Hz>, ...
Dot	CSc : Dot(Cross(<Hx, Hy, Hz>, ...
Complex>Real	Scl : Real(Dot(Cross(<Hx, ...
Geometry>Surface>{select cavity surfaces}	Srf : Surface(cav_tot_faces)
\int	Scl : Integrate(SurFace(...
Number>Scalar>2	Scl : 2
Constant>Pi	Scl : 3.14159265358979
Function>Scalar>Freq	Scl : {current freq, in Hz}
*	Scl : {numerical result, pi*f}
Number>Scalar>{enter μ_r for walls}	Scl : {entered value, unitless}
*	Scl : {numerical result, pi*f*mur}
Constant>Mu0	Scl : 1.25663706143592E-006

*	Scl : *(Integrate(Volume(... (above represents energy lost in dielectric material losses)
*	Scl : {numerical, pi*f*mur*mu0}
Number>Scalar>{enter wall conductivity}	Scl : {entered value, s/meter}
*	Scl : {numerical, pi*f*mur*mu0*σ}
√	Scl : {numerical, sqrt of above}
*	Scl : {numerical result, 2*above}
1/x	Scl : {numerical result} (above is skin depth/2)
*	Scl : *(Integrate(... (above is energy lost in walls)
+	Scl : +(*(Integrate(...
/	Scl : /(+(*(Integrate(...
Eval	Scl : {numerical result} (above is <i>Q</i> of homogeneous fill and wall conductivity cavity, unitless)

Calculating the Max Value of a Field Quantity in a Volume

Description:

This recipe permits you to calculate the Max electrical field (value or position) in a volume object.

Usage Example(s):

This calculation could be used to determine the Max (or Min) value (or position) of electrical field in a resonant cavity or dielectric object. Find out the hot (or quiet) spot value and location.

Prerequisites:

The volume (or surface) object within which the Max function is to be performed must already exist before the computation can be completed. If the volume (or surface) for Max function consists of the volume of several drawing objects, you first create a single list entry representing their combined volumes using **Modeler>List>Create>Object (or Face) List**.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Named Expressions>Mag_E>Copy to stack	Scl : Mag_E (Mag_E is used as an example)
Geometry>Volume..>{select volume}	Vol : Volume(box1) (above is example, user entry may differ)
Max>Value	Scl:Maximum(Volume(box1),Mag_E) (user can enter Max→Position to find out Max E location. Also Min→Value (Position) leads to Min E value and location)
Eval	Scl : {numerical quantity}

Calculating Dielectric (or Conduction) Loss in a Volume

Description:

This recipe shows you to calculate the dielectric/conduction loss in a volume object.

Usage Example(s):

This calculation could be used to determine the loss in a dielectric or metal object. For loss dielectric or finite conductivity metal object, its loss may have significant impact on user's design performance. Find out the key loss factors is critical for a successful design.

Prerequisites:

The volume object within which the loss calculation is to be performed must already exist before the computation can be completed. If the volume for loss calculation consists of the volume of several drawing objects, you must create a single list entry representing their combined volumes using **Modeler>List>Create>Object List**.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Named Expressions>Volume_Loss_Density> Copy to stack	Scl : Volume_Loss_Density <i>(Volume Loss Density is used as an example)</i>
Geometry>Volume...>{select volume}	Vol : Volume(box1) <i>(above is example, user entry may differ)</i>
∫	Scl: Integrate(Volume(box1), Volume_Loss_...)
Eval	Scl : {numerical quantity} <i>(above is loss from box1, either dielectric or conductional, Watt)</i>

Plotting Wave Impedance along a Line

Description:

This recipe generates a 2D plot of wave impedance in ohms vs. length for a line geometry. Wave impedance is obtained directly by taking the ratio of the transverse components of the electric field to the ratio of the transverse components of the magnetic field.

$$Z = \frac{E_o^-(x)}{H_o^-(x)}$$

Where $E_o^-(x)$ is the transverse component of the electric field, and $H_o^-(x)$ is transverse component of the magnetic field.

Usage Example(s):

This calculation could be used to display wave impedance vs. position along a length of waveguide with a changing cross-section. It could also be used to display the changes in wave impedance in free space at some boundary (i.e. a frequency selective surface or radome) when performed on an incident wave problem.

Prerequisites:

The line along which the impedance is to be plotted should be defined before performing this calculation. You can generate a line using **Modeler>Draw>Line**.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>E	CVc : <Ex, Ey, Ez>
Smooth	CVc : Smooth(<Ex, Ey, Ez>)
Complex>CmplxMag	Vec : CmplxMag(Smooth(<Ex, Ey, ...
Number>Vector>{enter unit vector in direction of propagation}	Vec : <0, 0, 1> (Z-directed unit vector used for example)
Cross	Vec : Cross(CmplxMag(Smooth(<...
Mag	Scl : Mag(Cross(CmplxMag(Smooth...
Quantity>H	CVc : <Hx, Hy, Hz>
Smooth	CVc : Smooth(<Hx, Hy, Hz>)
Complex>CmplxMag	Vec : CmplxMag(Smooth(<Hx, Hy,...
Number>Vector>{enter unit vector in direction of propagation}	Vec : <0, 0, 1> (Z-directed unit vector used for example)
Cross	Vec : Cross(CmplxMag(Smooth(<...

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Mag	Scl : Mag(Cross(CmplxMag(Smooth...
/	Scl : /(Mag(Cross(CmplxMag(Sm...
Add Type: Zwave OK	Zwave : /(Mag(Cross(CmplxMag(... (User defined Named Expression Zwave is added)
Done	{exit field calculator window}
HFSS>Results> Create Fields Report> Rectangular Plot> Geometry: Line1 Category: Calculator Expressions Quantity: Zwave New Report	{2D graph displayed} (y axis is wave impedance in <i>ohms</i> and x axis is position along line in drawing units)

Plotting the Phase of E Tangential to a Line/Curve

Description:

This recipe generates a 2D plot of the phase of an E-field whose vector component is tangential to a line. The line may also be a curve (faceted polyline).

Usage Example(s):

This calculation could be used to display the change in phase of the E field tangential to a circular path within a cylindrical dielectric resonator, when used on either a driven or eigensolution problem. Identifying the phase change along this curved path is often necessary to determine the mode index (e.g. Mode 10 δ) which a particular eigensolution or S-parameter resonance represents.

Prerequisites:

You must define the line along which the phase is to be plotted before performing this calculation. Use the **Modeler>Draw>Line** command.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>E	CVc : <Ex, Ey, Ez>
Geometr>Line>{select desired line}	Lin : Line(line1) (above line is example; user's may vary)
Unit Vec>Tangent	Vec : LineTangent
Complex>CmplxReal	CVc : CmplxR(LineTangent) (converts unit vector to complex vector)
Dot	CSc : Dot(<Ex,Ey,Ez>),CmplxR(...
Complex>CmplxPhase	Scl : Phase(Dot(<Ex,Ey,Ez>), ...
Add	Ephase : Phase(Dot(<Ex,Ey,Ez>),...
Type: Ephase	(User defined Named Expression Ephase is added)
OK	
Done	{exit field calculator window}
HFSS>Results> Create Fields Report> Rectangular Plot	{2D graph displayed} (y axis is E field phase in deg and x axis is position along line in drawing units)
Geometry: Line1	
Category: Calculator Expressions	
Quantity: Ephase	
New Report	

Plotting the Maximum Magnitude of E Tangential to a Line/Curve

Description:

This recipe generates a 2D plot of the maximum magnitude of an E-field tangential to a line. The line may also be a faceted curve. The maximum magnitude is not necessarily tied to the same input phase value along the length of the line.

Usage Example(s):

This calculation could be used to display the maximum magnitude of an E-field at all points along a line or curve in a transmission line structure, where it is the maximum magnitude and not the magnitude along the line corresponding to a single „snapshot in time“ (single port excitation phase) that is of interest. Such data could be used to determine whether the present design might exceed dielectric breakdown voltage in a particular location.

Prerequisites:

You should define the line along which the field data is to be plotted before performing this calculation. Use the **Modeler>Draw>Line** command.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>E	CVc : <Ex, Ey, Ez>
Geometry>Line>{select desired line}	Lin : Line(line1) (above line is example; user's may vary)
Unit Vec>Tangent	Vec : LineTangent
Complex>CmplxReal	CVc : CmplxR(LineTangent) (converts unit vector to complex vector)
Dot	CSc : Dot(<Ex,Ey,Ez>),CmplxR(...
Complex>CmplxPeak	Scl : CmplxMag(Dot(<Ex,Ey,Ex ... (above quantity is the maximum magnitude of the E-field tang. to the line. To obtain the mag. associated with a particular port phase excitation, enter a number into the stack and use the Complex>AtPhase operation instead.)

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Add Type: Et_max OK	Et_max : Phase(Dot(<Ex,Ey,Ez>),... (User defined Named Expression Et_max is added)
Done	{exit field calculator window}
HFSS>Results>Create Fields Report>Rectangular Plot Geometry: Line1 Category: Calculator Expressions Quantity: Et_max New Report	{2D graph displayed} (y axis is E field mag in $\mathbf{v/m}$ and x axis is position along line in drawing units)

Plotting the E-Field Magnitude Normal to a Surface

Description:

This recipe generates a scalar intensity plot of the E-field magnitude normal to a particular surface (or group of object surfaces, list of object faces), relative to a given input phase excitation. where S is the surface geometry, and \vec{n} is the normal vector to the surface S .

Usage Example(s):

This calculation could be used instead of the automatic Plot→Fields→MagE upon surface, when only the magnitude of the E-field with a particular vector orientation is desired. For example, to evaluate the field available for coupling to a probe structure with a particular orientation.

Prerequisites:

You should create the plane to which the desired field component should be normal before you begin the following steps. Use **Modeler>Draw>Plane**, or **Modeler>List>Create>Faces List**, or **Modeler>Draw>Rectangle** (or other 2D sheet).

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>E	CVc : <Ex, Ey, Ez>
Function>Phase	Scl : Phase
Complex>AtPhase	Vec : AtPhase(<Ex,Ey,Ez>, Phase)
Geometry>Surface>{select desired cutplane, faces list, or surface list}	Srf : Surface(faces1) (<i>faces1 used as example</i>)
Unit Vec>Normal	Vec : SurfaceNormal
Dot	Scl : Dot(AtPhase(<Ex,Ey,Ez>,Phase),...)
Add Type: E_normal	E_normal : Dot(AtPhase(<Ex,Ey,Ez>,0),... (<i>User defined Named Expression E_normal is added</i>)
OK	
Done	{exit field calculator window}
Select the surface (or list) HFSS>Fields>Plot Fields> Named Expression Select E_normal	{Scalar Plot on faces displayed} (<i>E-field normal component value in v/m</i>)

Generating an Iso-Surface Contour for a Given Field Value

Description:

This recipe generates a geometry entry called an IsoSurface which represents the surface upon which a selected scalar field quantity has a single value. This surface can be displayed, or used in later operations (to plot other quantities upon, etc.).

Usage Example(s):

This calculation could be used to locate regions of excessive field magnitudes for voltage breakdown or ohmic heating analysis. It could also be used to generate a desired isosurface to be used as an integration surface for another quantity.

Prerequisites:

You should plot of the field quantity of interest to determine the isovalue to use. Isovalues should be entered in MKS units (e.g. V/m, A/m) unless the problem is an eigen solution, in which case all field values are normalized to a peak of 1.0.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>E	CVc : <Ex, Ey, Ez> (IsoSurfaces for other quantities can also be created; E used as example.)
Smooth	CVc : Smooth(<Ex,Ey,Ez>) (as this routine generates a surface geometry object, data smoothing is recommended)
Function>Phase	Scl : Phase
Complex>AtPhase	Vec : AtPhase(Smooth(<Ex,Ey,Ez>),Phase)
Scal>ScalarX	Scl : ScalarX(AtPhase(Smooth(<Ex,...
Add...	E_x: ScalarX(AtPhase(Smooth<Ex,...
Type: E_x	(E_x is used here as an example; you can apply other
OK	quantity)
Done	{exit field calculator window}

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Highlight the geometry	<i>{Scalar Plot in the geometry displayed}</i>
HFSS>Fields>Plot Fields>Named Expression	<i>(E-field x-component value in v/m)</i>
Select E_x	
HFSS>Fields>Modify Plot Attribute...	<i>{IsoSurface contour is displayed}</i>
Select the plot	
OK	
Scale tab: Num. Division: <i>1</i>	
Use Limits: <i>100 (as an example)</i>	
Plots tab	
IsoValSurface checked	
Apply	

Generating an Animation on Multiple Planes with a Positional Variable

Description:

This recipe generates animated field output in which each frame is a snapshot of the fields on a different plane of the modeled volume. Any derived field quantity could be plotted in this manner, but this example will simply use the E-field magnitude at zero degrees input excitation.

Usage Example(s):

This calculation permits you to generate animated output results in addition to those automatically available from the post-processor. For example, peak E field ($E \cdot E$ conjugate) could be plotted at multiple planes in sequence.

Prerequisites:

This operation will only work in the global coordinate system if you are using X, Y, or Z positions as the animation variable.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Quantity>E	CVc : <Ex, Ey, Ez> (Animations for other quantities can also be created; E used as example.)
Smooth	CVc : Smooth(<Ex,Ey,Ez>)
Number>Scalar>"0"	Scl : 0
Complex>AtPhase	Vec : AtPhase(Smooth(<Ex,Ey,Ez>,...
Mag	Scl : Mag(AtPhase(Smooth(<Ex,...
Add	E_mag0 : Dot(AtPhase(<Ex,Ey,Ez>,0),...
Type: E_mag0	(User defined Named Expression E_mag0 is added)
OK	
Done	{exit field calculator window}
Planes>Global:YZ (user can choose other planes Under modeler tree/Planes)	(YZ plane that can vary with X position)

Calculator Operation	Resulting Stack Display (top entry only unless noted)
HFSS>Fields>Plot Fields>Named Expression (E_mag0)	<i>(Plot named expression on YZ plane)</i>
HFSS>Fields>Plot Animate>New	<i>{launches Animation Plot Settings}</i>
Swept variable Normalized Distance OK	<i>{displays animation}</i>

Generating Cartesian/Cylindrical/Spherical-Coordinate Field-Components Library

Description:

This recipe demonstrates the steps to export user-defined named expressions into a library which can be loaded into and reused in other designs or projects.

Usage Example(s):

This calculation allows you to generate named expressions in addition to those automatically available ones in the field calculator, and save them as a user library, and reload into other designs/projects for use. For example, Cartesian components of E field are used for demonstration. If you are interested in Cylindrical/Spherical components of E field, please contact Ansoft support to obtain such libraries.

Prerequisites:

Since this recipe is intended to generate generalized user-defined named-expressions, this operation should not be geometry-related.

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Named Expressions>Vector_E	Vec : Vector_E
	<i>(E used as example)</i>
Scal?>ScalarX	Sc1 : ScalarX(Vector_E)
Add...	
Type in Ex	
OK	
Named Expressions>Vector_E	Vec : Vector_E
Scal?>ScalarY	Sc1 : ScalarY(Vector_E)
Add...	
Type in Ey	
OK	
Named Expressions>Vector_E	Vec : Vector_E
Scal?> ScalarZ	Sc1 : ScalarZ(Vector_E)

Calculator Operation

Resulting Stack Display (top entry only unless noted)

Add...

Type in Ez

OK

Save To...

(Exyz used as an library name example)

Select (**Ex**, **Ey**, **Ez**)

OK

Type in **Exyz** (Library Name)

Save

Load From...

*(Named Expressions of Ex, Ey and Ez are loaded into
Named Expressions)*

(Find the pre-defined library)

Open

Select (**Ex**, **Ey**, **Ez**)

OK

