3 - 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 Quantities for 2D (Line) Plot Outputs

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

Calculating Quantities for 3D (Volume) Plot Outputs

Calculating Quantities for Animated Outputs

Creating User Defined Named Expressions 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	Scl: {placeholders for numerical results shown in brackets}
Button steps requiring data entry will have entry quantity shown in {brackets}	resulting stack entry/type (*notes follow in italics)*

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. .

$$I = \oint_{l} \vec{H} \cdot \vec{dl}$$

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	SclLin: 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))
∫	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} *(Final complex current result)*

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.

$$V = \int_l \vec{E} \cdot \vec{dl}$$

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) (*user line name may differ from example*)
Tangent	SclLin: 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>Imag	Vec : Imag(<Ex,Ey,Ez>)
Geometry>Line...> {select line}	Lin : Line (line1) (*user line name may differ from example*)
Tangent	ScLin: LineValue(Line(...),Dot(Imag<Ex,Ey,Ez>), LineTangent))
∫	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} (*Final complex voltage result*)

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.

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

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 youmust 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*)
Normal	SclSrf : SurfaceValue(Surface(Facelist1), Dot(Real(Poynting), SurfaceNormal) *(takes the dot product of the vector data with the normal to the surface(s) selected*)

Calculator Operation	Resulting Stack Display (top entry only unless noted)
∫	Scl : Integrate(Surface..)
Eval	Scl : {numerical value} (*final answer in **watts***)

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*)

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Value	SclSrf : SurfaceValue(Surface(plane1), Phase(ScalarX(<Ex, Ey, Ez>)))
∫	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*)
∫	Scl : Integrate(Surface(...
/	Scl : /(Integrate(SurfaceValue(...
Eval	Scl : {numerical value} (*for this example units are in **deg** or **radians***)

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 **E E** within the volume.

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

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>), ... (*note: the dot product of the E with its conjugate should lead to a real quantity, but the calculator still assume as complex*)
Geometry>Volume...> {select volume}	Vol : Volume(box1) (*above is example, user entry may differ*)
∫	Scl : Integrate(Volume(...
Eval	Scl : {numerical quantity}
Constant>Epsi0	Scl : 8.854187817E-012
Number>Scalar> {enter r for volume}	Scl : {numerical quantity}
*	Scl : {numerical quantity} *(stack entry is volume)*
Number>Scalar>0.5	Scl : 0.5
*	Scl : {numerical quantity}
*	Scl : {numerical quantity} *above is electrical energy in **joules***

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:

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

where *s* is skin depth, *tg* is dielectric loss tangent, *n* is the surface normal for the cavity wall faces, and and 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	*(above entry duplicated)*
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) (*above is example; user entry may differ*)
$\int$	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)
	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 r for walls}	Scl : {entered value, unitless}
*	Scl : {numerical result, pi*f*mur}
Constant>Mu0	Scl : 1.25663706143592E-006
*	Scl : {numerical, pi*f*mur*mu0}
Number>Scalar>{enter wall conductivity}	Scl : {entered value, s/meter}
*	Scl : {numerical, pi*f*mur*mu0* }
	Scl : {numerical, sqrt of above}

*	Scl : *(Integrate(Volume(... (*above represents energy lost in dielectric material losses*)
*	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 Q 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*)

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

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>)

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

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.*)
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 **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 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>Rectangl**e (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) *(faces1 used as example)*
Unit Vec>Normal	Vec : SurfaceNormal
Dot	Scl : Dot(AtPhase (<Ex,Ey,Ez>,Phase),…)
Add Type: E_normal OK	E_normal : Dot(AtPhase (<Ex,Ey,Ez>,0),… (*User defined Named Expression E_normal is added*)
Done	{*exit field calculator window*}
Select the surface (or list) **HFSS>Fields>Plot Fields>** **Named Expression** Select E_normal	{Scalar Plot on faces displayed} *(E-field normal component value in **v/m**)*

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... Type: E_x OK	E_x: ScalarX(AtPhase(Smooth<Ex,… (E_x *is used here as an example; you can apply other quantity*)
Done	{*exit field calculator window*}
Highlight the geometry	{*Scalar Plot in the geometry displayed*}
HFSS>Fields>Plot Fields>Named Expression	*(E-field x-component value in **v/m**)*

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Select E_x	
HFSS>Fields>Modify Plot Attribute… Select the plot OK **Scale** tab: Num. Division: *1* Use Limits: *100 (as an example)* **Plots** tab IsoValSurface checked Apply	{*IsoSurface contour is displayed*}

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 dot 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.*)

Calculator Operation	Resulting Stack Display (top entry only unless noted)
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 Type: E_mag0 OK	E_mag0 : Dot(AtPhase(<Ex,Ey,Ez>,0),… (*User defined Named Expression E_mag0 is added*)
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)*
HFSS>Fields>Plot Fields>Named Expression (E_mag0)	*(Plot named expression on YZ plane)*
HFSS>Fields>Plot Animate>New	{*launches Animation Plot Settings*}
Swept variable Normalized Distance OK	{*displays animation*}

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.

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 (*E used as example*)
Scal?>ScalarX	Scl : ScalarX(Vector_E)
Add... Type in Ex OK	
Named Expressions>Vector_E	Vec : Vector_E
Scal?>ScalarY	Scl : ScalarY(Vector_E)
Add... Type in Ey OK	
Named Expressions>Vector_E	Vec : Vector_E
Scal?> ScalarZ	Scl : ScalarZ(Vector_E)
Add... Type in Ez OK	
Save To… Select (**Ex, Ey, Ez**) OK *Type in* **Exyz** (Library Name) Save	(*Exyz used as an library name example*)
Load From… *(Find the pre-defined library)* Open	*(Named Expressions of Ex, Ey and Ez are loaded into Named Expressions)*

Calculator Operation	Resulting Stack Display (top entry only unless noted)
Select (**Ex, Ey, Ez**) OK	