User Manual for reTORT, PFSS and the GEMSIF Computational Framework

E x H, Inc. <support@exhsw.com>

Version 3.3.0

Table of Contents

Introduction

1. What is GEMSIF?

GEMSIF (*GEometry, Modeling, and Simulation InterFace*) is the E x H Computational Framework and is common to all E x H solvers, including reTORT and the solvers comprising PFSS. GEMSIF provides the UI through which you enter data and render your results. It is the focus of the designer's User Experience, allowing you to express yourself in the way best suited to your methods and workflow. GEMSIF is included with any license of an E x H solver and several instances of one or more solvers may be run within a single instance of GEMSIF. GEMSIF is the future of true interoperability among computational electromagnetic solvers.

2. Tutorials

GEMSIF tutorials and example files can be found under Support > Resources at the E x H, Inc. website: https://exhsw.com/exh-support/everything-you-need-for-retort/

Topical Documentation for GEMSIF

1. Application

The Application node is the root node in the GEMSIF user interface. It contains all of the general GEMSIF settings such as docks and user settings, as well as any open Systems. The state of the Application node may be saved using workspaces.

1.1. Children

The Application node children consist of collections of properties and settings, views, and Systems. GEMSIF displays the standard child objects in the hierarchy tree. To view all the child objects, right click the Hierarchy dock and select "Show All".

As the primary child object, Systems can be created or opened by right clicking on the node. Additionally, selecting "File" in the file menu bar will provide the user with the Application node child objects that can be created and edited like the settings and workspace.

GEMSIF settings such as libraries, licenses, and processing settings, as well as keyboard shortcuts are accessed through the file menu File category. Physical dimension properties and workspace descriptions are accessed by right clicking on the Application node and selecting "Properties".

1.2. Objects and Collections

- Global User-specific Settings
- Docks
- Views
- Parameters
- Actions
- Materials
- Design Studies / Optimizers
- Tolerance Analysis

2. System

The system level is the project file which contains the simulations. It can contain multiple simulations and the simulations need not be from the same solver. System nodes can also be used to combine and review data from multiple simulations. The System node serves as the basis for a saved file, allowing the user to save all encapsulated simulations as well as the child collections.

1. To create a new system, either right click the Application node and select "New System" or use the file menu and select "File" → "New System".
2. To save a system, either right click the system node and select "Save System" or use the file menu and select "File" → "Save System". Note that saving a system names the system.
3. To open a previously saved system, either right click the Application node and select "Open System" or use the file menu and select "File" → "Open System".

2.1. Children

GEMSIF Systems contain zero, one, or more Simulation objects. A simulation may be added to an active System either from the System file menu toolbar, by right-clicking a specific System and selecting "New Simulation" from the pop-up menu, or through the scripting interface. A Simulation Template may also be added from these menus.

Views created at the system level can be used to combine data from multiple simulations. Parameters created at this level will be inherited by the system's children (simulations and other parameters).

2.2. Objects and Collections

- Settings
- Parameters
- Actions
- Design Studies
- Views

2.3. See Also

- GEMSIF Simulations

3. Simulation

GEMSIF Simulation objects contain the input data and settings for executing a computational solver, as well as any tools for obtaining and processing the results. GEMSIF does not provide any simulation objects by default; one or more Solver plugins must be licensed and installed in order to make a simulation option available.

For additional simulation documentation, see the solver plugin documentation:

- reTORT Topical Documentation
- PFSS Topical Documentation

4. GEMSIF Objects

GEMSIF Objects consist of all objects that are common to any system or simulation in GEMSIF. Objects that are solver specific can be found in the documentation for the respective solver.

4.1. GEMSIF Object Properties

GEMSIF Object Properties are properties that GEMSIF objects have in common.

In general, properties can support many different types of values, such as text values (strings), arrays, dimensioned values (such as "5 cm"), and dimensionless values (such as "3.14" or "1 + 2i"). Mathematical expressions can also be used in some properties, such as "15 cm / 3". Names of Parameters can also be used, such as in the expression "THICKNESS / 3", assuming there is a parameter called THICKNESS in the simulation. For more information on the types of values supported by properties, see Parameter Values.

4.1.1. name

All objects must be uniquely named within their parent. See the section on object naming restrictions for the rules on what is and is not a valid GEMSIF object name.

Object names are used to identify and reference a particular object from elsewhere in the program. For example, materials defined for a given simulation are referenced by name to assign properties to a geometric object.

The names of some objects are given special importance. For example, Parameters are referenced by their name in the properties of any other data objects to create parameterized geometry and simulations.

4.1.2. color

Object color determines the color of the object when it is visually rendered by a modelview. In the standard Editor, selecting the color box will popup a color selection window. In the Grid Editor, color is specified as an array of three integers between 0 and 255 representing red, green, and blue intensity, respectively.

4.1.3. transparency

Object transparency determines the degree of transparency when the object is rendered in a modelview. Transparency must be a decimal value between 0 and 1, inclusive, where 0 is opaque and 1 is perfectly transparent (invisible in the display).

4.2. Actions

Actions are stored Javascript code that may be executed to automate tasks in the GUI, or to be used as a callback method to provide custom constraints or activities for different data objects.

Actions are not available in the model hierarchy by default.

See the Javascript API for details about the syntax and commands available from the Actions.

4.3. Action Properties

4.3.1. Name

See GEMSIF Object Properties - Name

4.3.2. Content

Stores the commands entered in the action interface.

4.4. Materials

Materials are collections of properties used to define a material's characteristics in GEMSIF. Materials are not solver specific; however, each solver will look for certain properties when evaluating a material.

In addition to creating individual materials, the Material category allows the user to create mixtures of materials by combining two or three materials.

4.4.1. Basic Optical Library Imports

This library contains basic arbitrary optical materials.

Linear Dispersive Material

The linear dispersive material library import provides a material with modifiable properties listed below.

- Refractive Index (n): The refractive index of the material, governing refractive power of a material.
- Abbe Number: The dispersion coefficient governing the reaction of the material to various wavelengths.
- Minimum Wavelength: The minimum wavelength used in the calculation of the Abbe number.
- Midpoint Wavelength: The midpoint wavelength used in the calculation of the Abbe number.
- Maximum Wavelength: The maximum wavelength used in the calculation of the Abbe number.

Index Material

The index material library import provides a material with modifiable properties listed below.

- Refractive Index (n): The refractive index of the material, governing refractive power of a material.
- Loss Factor (k): The loss factor determines the loss at a given refraction.

4.4.2. GEMSIF Import Library

Nondispersive Material

The nondispersive material import is defined on a more traditionally electromagnetic form rather than an optical one. It contains the following properties:

- Permittivity: The electric permittivity or epsilon value for the material.
- Permeability: The magnetic permeability or mu value for the material.
- Conductivity: The conductivity or sigma value for the material.
- Loss Tangent: The loss tangent for the material.

4.4.3. Polynomial GRIN Import Library

This library contains a set of various convenient arbitrary GRIN definitions based on polynomial equations.

Radial Polynomial Spherical GRIN

The Radial Polynomial Spherical GRIN is defined as a polynomial in terms of radial distance (r) from a given center. It has the following properties:

- Polynomial Units: The units applied to the spherical radial coordinate.
- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Linear Term (C1): The linear (first order) contribution from the spherical r coordinate.
- Quadratic Term (C2): The quadratic (second order) contribution from the spherical r coordinate.
- Cubic Term (C3): The cubic (third order) contribution from the spherical r coordinate.
- Quartic Term (C4): The quartic (fourth order) contribution from the spherical r coordinate.
- Quintic Term (C5): The quintic (fifth order) contribution from the spherical r coordinate.
- Sextic Term (C6): The sextic (sixth order) contribution from the spherical r coordinate.
- Septic Term (C7): The septic (seventh order) contribution from the spherical r coordinate.
- Octic Term (C8): The octic (eighth order) contribution from the spherical r coordinate.
- Nonic Term (C9): The nonic (ninth order) contribution from the spherical r coordinate.
- Decic Term (C10): The decic (tenth order) contribution from the spherical r coordinate.

- Axial Relative Origin: The displacement from the origin of the associated surface along the orientation of the object.

Radial Polynomial GRIN

The Radial Polynomial GRIN is defined as a polynomial in terms of cylindrical radial distance (r) from a given center. It has the following properties:

- Polynomial Units: The units applied to the cylindrical radial coordinate.
- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Quadratic Term (C1): The quadratic (second order) contribution from the cylindrical r coordinate.
- Quartic Term (C2): The quartic (fourth order) contribution from the cylindrical r coordinate.
- Sextic Term (C3): The sextic (sixth order) contribution from the cylindrical r coordinate.
- Octic Term (C4): The octic (eighth order) contribution from the cylindrical r coordinate.
- Decic Term (C5): The decic (tenth order) contribution from the cylindrical r coordinate.

Radial Axial Polynomial GRIN

The Radial Axial Polynomial GRIN is defined as a polynomial in terms of radial distance (r) from a given axis and distance (z) along that axis. It has the following properties:

- Polynomial Units: The units applied to the cylindrical radial and axial distance coordinates.
- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Quadratic R Term (C1): The quadratic (second order) contribution from the cylindrical r coordinate.
- Quartic R Term (C2): The quartic (fourth order) contribution from the cylindrical r coordinate.
- Sextic R Term (C3): The sextic (sixth order) contribution from the cylindrical r coordinate.
- Octic R Term (C4): The octic (eighth order) contribution from the cylindrical r coordinate.
- Decic R Term (C5): The decic (tenth order) contribution from the cylindrical r coordinate.
- Linear Z Term (C6): The linear (first order) contribution from the axial z coordinate.
- Quadratic Z Term (C7): The quadratic (second order) contribution from the axial z coordinate.

Linear Spherical GRIN

The Linear Spherical GRIN library import object allows the user to define a spherical material that linearly changes its refraction index over the described range according to a spherical definition. It has the following properties:

- Back Index (n0): The refractive index at the back of the lens.
- Front Index (n1): The refractive index at the front of the lens.
- GRIN Thickness: The Thickness of the lens in question.

- GRIN Radius of Curvature: The radius of curvature of the spherical definition of the GRIN.

Cross Polynomial GRIN

The Cross Polynomial GRIN is defined as a polynomial in terms of cylindrical radial distance (r) from a given axis and distance (z) along that axis. It has the following properties:

- Polynomial Units: The units applied to the cylindrical radial and axial distance coordinates.
- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Quadratic R Term (C1): The quadratic (second order) contribution from the cylindrical r coordinate.
- Quartic R Term (C2): The quartic (fourth order) contribution from the cylindrical r coordinate.
- Sextic R Term (C3): The sextic (sixth order) contribution from the cylindrical r coordinate.
- Octic R Term (C4): The octic (eighth order) contribution from the cylindrical r coordinate.
- Decic R Term (C5): The decic (tenth order) contribution from the cylindrical r coordinate.
- Linear Z Term (C6): The linear (first order) contribution from the axial z coordinate.
- Quadratic Z Term (C7): The quadratic (second order) contribution from the axial z coordinate.
- Quadratic R Linear Z Term (C8): The third order contribution from a linear axial z and quadratic r product.
- Quadratic R and Z Term (C9): The fourth order contribution from a quadratic r and z product.
- Quartic R Linear Z Term (C10): The fifth order contribution from a quartic r and linear z product.
- Quartic R Quadratic Z Term (C11): The sixth order contribution from a quartic r and quadratic z product.

Axial Polynomial GRIN

The Axial Polynomial GRIN is defined as a polynomial in terms of distance along the axis about which the material is oriented. It has the following properties:

- Polynomial Units: The units applied to the axial distance coordinate.
- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Linear Z Term (C1): The linear (first order) contribution from the axial z coordinate.
- Quadratic Z Term (C2): The quadratic (second order) contribution from the axial z coordinate.

4.4.4. Polynomial Binary Mixture GRIN Import Library

This library contains a set of various convenient arbitrary Binary Mixture GRIN definitions based on polynomial equations. This equation governs the mixing proportion between two materials where 0 is 100% the first material and 1 is 100% the second material. All of these include the fields mentioned in Material Mixtures as well.

Radial Polynomial Spherical Binary Mixture

The Radial Polynomial Spherical Binary Mixture is defined as a polynomial in terms of radial distance (r) from a given center. It has the following properties:

- Polynomial Units: The units applied to the spherical radial coordinate.
- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Linear Term (C1): The linear (first order) contribution from the spherical r coordinate.
- Quadratic Term (C2): The quadratic (second order) contribution from the spherical r coordinate.
- Cubic Term (C3): The cubic (third order) contribution from the spherical r coordinate.
- Quartic Term (C4): The quartic (fourth order) contribution from the spherical r coordinate.
- Quintic Term (C5): The quintic (fifth order) contribution from the spherical r coordinate.
- Sextic Term (C6): The sextic (sixth order) contribution from the spherical r coordinate.
- Septic Term (C7): The septic (seventh order) contribution from the spherical r coordinate.
- Octic Term (C8): The octic (eighth order) contribution from the spherical r coordinate.
- Nonic Term (C9): The nonic (ninth order) contribution from the spherical r coordinate.
- Decic Term (C10): The decic (tenth order) contribution from the spherical r coordinate.
- Axial Relative Origin: The displacement from the origin of the associated surface along the orientation of the object.

Radial Polynomial Binary Mixture

The Radial Polynomial Binary Mixture is defined as a polynomial in terms of cylindrical radial distance (r) from a given center. It has the following properties:

- Polynomial Units: The units applied to the cylindrical radial coordinate.
- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Quadratic Term (C1): The quadratic (second order) contribution from the cylindrical r coordinate.
- Quartic Term (C2): The quartic (fourth order) contribution from the cylindrical r coordinate.
- Sextic Term (C3): The sextic (sixth order) contribution from the cylindrical r coordinate.
- Octic Term (C4): The octic (eighth order) contribution from the cylindrical r coordinate.
- Decic Term (C5): The decic (tenth order) contribution from the cylindrical r coordinate.

Radial Axial Polynomial Binary Mixture

The Radial Axial Polynomial Binary Mixture is defined as a polynomial in terms of radial distance (r) from a given axis and distance (z) along that axis. It has the following properties:

- Polynomial Units: The units applied to the cylindrical radial and axial distance coordinates.

- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Quadratic R Term (C1): The quadratic (second order) contribution from the cylindrical r coordinate.
- Quartic R Term (C2): The quartic (fourth order) contribution from the cylindrical r coordinate.
- Sextic R Term (C3): The sextic (sixth order) contribution from the cylindrical r coordinate.
- Octic R Term (C4): The octic (eighth order) contribution from the cylindrical r coordinate.
- Decic R Term (C5): The decic (tenth order) contribution from the cylindrical r coordinate.
- Linear Z Term (C6): The linear (first order) contribution from the axial z coordinate.
- Quadratic Z Term (C7): The quadratic (second order) contribution from the axial z coordinate.

Linear Spherical Binary Mixture

The Linear Spherical Binary Mixture library import object allows the user to define a spherical material that linearly changes its mixing fraction over the described range according to a spherical definition. It has the following properties:

- Back Index (n0): The refractive index at the back of the lens.
- Front Index (n1): The refractive index at the front of the lens.
- GRIN Thickness: The Thickness of the lens in question.
- GRIN Radius of Curvature: The radius of curvature of the spherical definition of the GRIN.

Cross Polynomial Binary Mixture

The Cross Polynomial Binary Mixture is defined as a polynomial in terms of cylindrical radial distance (r) from a given axis and distance (z) along that axis. It has the following properties:

- Polynomial Units: The units applied to the cylindrical radial and axial distance coordinates.
- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Quadratic R Term (C1): The quadratic (second order) contribution from the cylindrical r coordinate.
- Quartic R Term (C2): The quartic (fourth order) contribution from the cylindrical r coordinate.
- Sextic R Term (C3): The sextic (sixth order) contribution from the cylindrical r coordinate.
- Octic R Term (C4): The octic (eighth order) contribution from the cylindrical r coordinate.
- Decic R Term (C5): The decic (tenth order) contribution from the cylindrical r coordinate.
- Linear Z Term (C6): The linear (first order) contribution from the axial z coordinate.
- Quadratic Z Term (C7): The quadratic (second order) contribution from the axial z coordinate.
- Quadratic R Linear Z Term (C8): The third order contribution from a linear axial z and quadratic r product.
- Quadratic R and Z Term (C9): The fourth order contribution from a quadratic r and z product.
- Quartic R Linear Z Term (C10): The fifth order contribution from a quartic r and linear z product.

- Quartic R Quadratic Z Term (C11): The sixth order contribution from a quartic r and quadratic z product.

Axial Polynomial Binary Mixture

The Axial Polynomial Binary Mixture is defined as a polynomial in terms of distance along the axis about which the material is oriented. It has the following properties:

- Polynomial Units: The units applied to the axial distance coordinate.
- Constant Term (C0): The constant contribution to the polynomial (unitless).
- Linear Z Term (C1): The linear (first order) contribution from the axial z coordinate.
- Quadratic Z Term (C2): The quadratic (second order) contribution from the axial z coordinate.

4.4.5. Advanced Material

Material objects define the characteristics of a single material. These characteristics comprise both how GEMSIF will render the material and how GEMSIF will simulate the material. Material properties can be simple or complex so long as it defines any property required by the simulation solver being used.

Material properties are set by selecting the material and editing the properties listed in the Property Editor. Electromagnetic properties are added and defined by adding Material Field Definitions.

4.4.6. Material Properties

Name

See GEMSIF Object Properties - Name

Color

See GEMSIF Object Properties - Color

Transparency

See GEMSIF Object Properties - Transparency

Source

Allows the user to provide a reference for the source of the material properties.

Notes

Notes provide the user with a place to provide additional information on their material. Notes do not impact a materials properties.

Coordinate System

Specifies the coordinate system the material properties are displayed in.

Relative Origin

Allows the user to provide a default relative origin for a material used in reTORT.

Orientation

Specifies the default orientation, or direction, of a material used in reTORT.

4.4.7. Material Field Definitions

Material Fields allow the user to specify the simulated properties of a material. Several different types of Fields are available:

- Debye Dispersion Model - Electric permittivity dispersion model vs. Frequency for the Debye single-resonator dispersion model.
- Drude (metallic) Dispersion Model - Electric permittivity dispersion model vs. Frequency for single-resonator Drude-type metallic or plasma dispersion.
- Lorentz-Drude (lossy dielectric) Dispersion Model - Multi-resonator electric permittivity dispersion model for lossy dielectrics and metals.
- Nondispersive Field - Provides a field value that is independent of other conditions.
- Optical Dispersion Model - Specifies a refractive index dispersion model versus wavelength (in microns) for optical glasses, gasses, and other materials based on one of several analytical forms, including Sellmeier coefficients.
- Polynomial Field - Provides a field value that varies according to a polynomial expression in one or more variable axes.
- Polynomial Spatial Gradient Field - A vector-valued field whose value is computed as the spatial gradient of a dependent polynomial field. Used for specifying gradient-index (GRIN) lenses.
- Table Field - Provides a field value that varies over a specified property and is described by a table of values.

Debye Dispersion Model

The Debye model represents a material as a collection of ideal, non-interacting dipoles and evaluates the effective, collective response to an alternating external electric field. The field value is computed from the solver’s current wavelength, and computes the effective complex permittivity for the specified high and low-frequency permittivity values and relaxation time tau.

$$\epsilon(f) = \epsilon_\infty + \frac{\epsilon_\infty - \epsilon_0}{1 + 2i\pi f\tau}$$

Debye Dispersion Model Properties

- Name - See GEMSIF Object Properties
- Field - See Field Properties
- Data - See Field Properties
- Size - See Field Properties
- Format - See Field Properties
- Minimum Frequency - See Field Properties
- Maximum Frequency - See Field Properties
- High-frequency Permittivity - High-frequency asymptotic permittivity for the Debye model.
- Low-frequency Permittivity Low-frequency asymptotic permittivity for the Debye model. The low frequency and high frequency permittivity coefficients determine the range of permittivity values represented by the model.
- Time Constant - Resonator relaxation time.

Drude (metallic) Dispersion Model

The Drude model of electrical conduction represents the complex permittivity of a conductive material at optical and infrared wavelengths based on a provided plasma frequency and relaxation time constant. The Lorentz-Drude model extends the Drude model to allow modeling lossy dielectrics as well as lossy metals.

Drude Dispersion Model Properties

- Name - See GEMSIF Object Properties
- Field - See Field Properties
- Data - See Field Properties
- Size - See Field Properties
- Format - See Field Properties
- Minimum Frequency - See Field Properties
- Maximum Frequency - See Field Properties
- Plasma Frequency - See Field Properties
- Time Constant - See Field Properties

Lorentz-Drude (lossy dielectric) Dispersion Model

The Lorentz-Drude model allows multiple resonator specification terms to be added to the model, each having their own resonance behavior. The material’s response is the weighted sum of the various resonator responses.

Lorentz-Drude Dispersion Model Properties

- Name - See GEMSIF Object Properties
- Field - See Field Properties
- Data - See Field Properties
- Size - See Field Properties
- Format - See Field Properties
- Minimum Frequency - See Field Properties
- Maximum Frequency - See Field Properties
- High-frequency Permittivity - See Field Properties

Terms

Lorentz-Drude Resonator terms specify both a resonance frequency as well as a plasma frequency along with the relaxation time constant. Appropriate choice of the resonance frequencies will cause the model to represent either a conductive metal or a lossy dielectric. The shape of the dispersion curve for each resonator contribution is a Lorentzian.

- Name - See GEMSIF Object Properties
- Scale Factor - Weighting factor in the summation of all terms within the field. The scale factors of all terms within a Lorentz-Drude field should sum to 1.
- Plasma Frequency - See Field Properties
- Resonance Frequency - The resonance frequency is typically a measured property of a material, but can be related to atomic transition frequencies.
- Time Constant - See Field Properties

Nondispersive Field

Nondispersive fields are assigned a static-fixed value, and are not dependent on any solver parameters or properties. Nondispersive fields are appropriate for representing materials with permittivity, permeability, or loss factors that do not change with frequency or other environmental conditions.

Nondispersive Field Properties

- Name - See GEMSIF Object Properties
- Field - See Field Properties
- Data - See Field Properties
- Size - See Field Properties
- Format - See Field Properties
- Value : Sets the value of a material property.

Optical Dispersion Model

Provides common analytical dispersion models for the refractive index of optical materials with respect to wavelength. These models are used by the material database for importing optical materials. The coefficients for these models are commonly determined numerically from measured data, unlike the Debye, Drude, and Lorentz-Drude models whose coefficients can in some cases be computed analytically based on the underlying physical model. Different models are better suited for representing the dispersion characteristics of different classes of physical materials. For example, Sellmeier coefficients are commonly used to represent the dispersion behavior of optical glasses.

The dispersion model equations specify the refractive index of the material as a function of the wavelength in microns. Each dispersion model is defined using a list of coefficients, where the meaning of each coefficient depends on the equation for the selected model. The length of the coefficient list must also match the requirements of the dispersion model. For instance, a Sellmeier2 equation requires at least one coefficient, and two extra coefficients for every additional term. However, the Retro equation requires exactly four coefficients. For more information, see the Dispersion Model property or the Dispersion Formulas for the RefractiveIndex.INFO database.

Optical Dispersion Properties

- Name - See GEMSIF Object Properties
- Field - See Field Properties
- Data - See Field Properties
- Size - See Field Properties
- Format - See Field Properties
- Dispersion Model - Select the desired optical dispersion model to use for computing the refractive index of the material. The wavelength used by the dispersion formula is given in microns. The available models include:
    - **Sellmeier:** Requires at least one coefficient, and two more coefficients for each additional term.

      $$n^2 - 1 = C_1 + \frac{C_2\lambda^2}{\lambda^2 - {C_3}^2} + \frac{C_4\lambda^2}{\lambda^2 - {C_5}^2} + \frac{C_6\lambda^2}{\lambda^2 - {C_7}^2} + \ldots$$

    - **Sellmeier2:** Requires at least one coefficient, and two more coefficients for each additional term.

      $$n^2 - 1 = C_1 + \frac{C_2\lambda^2}{\lambda^2 - C_3} + \frac{C_4\lambda^2}{\lambda^2 - C_5} + \frac{C_6\lambda^2}{\lambda^2 - C_7} + \ldots$$

    - **Polynomial:** Requires at least one coefficient, and two more coefficients for each additional term. Not to be confused with the Polynomial material field.

      $$n^2 = C_1 + C_2\lambda^{C_3} + C_4\lambda^{C_5} + C_6\lambda^{C_7} + \ldots$$

    - **RefractiveIndexDotInfo:** Requires at least five coefficients for the first two terms, plus four more coefficients for the third term, and two more coefficients for each term after that.

$$n^2 = C_1 + \frac{C_2\lambda^{C_3}}{\lambda^2 - {C_4}^{C_5}} + \frac{C_6\lambda^{C_7}}{\lambda^2 - {C_8}^{C_9}} + C_{10}\lambda^{C_{11}} + C_{12}\lambda^{C_{13}} + C_{14}\lambda^{C_{15}} + \ldots$$

  - **Cauchy:** Requires at least one coefficient, and two more coefficients for each additional term.

    $$n = C_1 + C_2\lambda^{C_3} + C_4\lambda^{C_5} + C_6\lambda^{C_7} + \ldots$$

  - **Gasses:** Requires at least one coefficient, and two more coefficients for each additional term.

    $$n - 1 = C_1 + \frac{C_2}{C_3 - \lambda^{-2}} + \frac{C_4}{C_5 - \lambda^{-2}} + \frac{C_6}{C_7 - \lambda^{-2}} + \ldots$$

  - **Herzberger:** Requires at least three coefficients, and one more coefficient for each additional term.

    $$n = C_1 + C_2\left(\frac{1}{\lambda^2 - 0.028}\right) + C_3\left(\frac{1}{\lambda^2 - 0.028}\right)^2 + C_4\lambda^2 + C_5\lambda^4 + C_6\lambda^6 + \ldots$$

  - **Retro:** Requires exactly four coefficients.

    $$\frac{n^2 - 1}{n^2 + 2} = C_1 + \frac{C_2\lambda^2}{\lambda^2 - C_3} + C_4\lambda^2$$

  - **Exotic:** Requires exactly six coefficients.

    $$n^2 = C_1 + \frac{C_2}{\lambda^2 - C_3} + \frac{C_4(\lambda - C_5)}{(\lambda - C_5)^2 + C_6}$$

- Minimum Wavelength - Lower-wavelength bound for the model's region of validity.
- Maximum Wavelength - Upper wavelength bound for the model's region of validity.
- Coefficients - Array of coefficients for the selected Dispersion Model. The first coefficient in the list corresponds to C_1 in the dispersion formula, the second is C_2, etc. The number of coefficients must be compatible with the selected model.

Polynomial Field

Represent an arbitrary field as a polynomial of one or more variables. The units specified in the axis specification will be used to compute the polynomial, so coefficients should be chosen to account for the provided units.

Polynomial Field Properties

- Name - See GEMSIF Object Properties
- Headers - See Field Properties
- Rows - See Field Properties
- Field - See Field Properties
- Data - See Field Properties

- Size - See Field Properties
- Format - See Field Properties
- Powers - List of exponents used by the polynomial.
- Coefficients - List of coefficients used by the polynomial.
- Minimum - Minimum polynomial value for range calculations.
- Maximum - Maximum polynomial value for range calculations. If the range of the polynomial within the bounded area determined by the axes ranges and the range mode has been set, then the polynomial coefficients will be scaled to restrict the polynomial range to the desired output range.

Polynomial Spatial Gradient Field

When assigned the name of a polynomial field, computes the vector gradient of the field for use when specifying a gradient index (GRIN) optic.

Polynomial Spatial Gradient Field Properties

- Name - See GEMSIF Object Properties
- Field - See Field Properties
- Data - See Field Properties
- Size - See Field Properties
- Format - See Field Properties
- Source - Specifies the Polynomial Field used to compute the gradient.

Table Field

Computes a field value based on solver axes values by interpolating within a (potentially sparse) data table.

Table Field Properties

- Name - See GEMSIF Object Properties
- Headers - See Field Properties
- Rows - See Field Properties
- Field - See Field Properties
- Data - See Field Properties
- Size - See Field Properties
- Format - See Field Properties
- Interpolation - Method of calculating data between points in the table.
- Coordinates - Stored values of the axes.

- Data - Data stored in the table.

Material Axes

Represents a solver variable that will be used to compute the field. Both user-defined and solver-recognized names can be specified. User-specified names will be accessible as properties on the material, and can be bound to parameters to perform sweeps or optimizations.

The axis values used for generating the field values are scaled to the requested units, and checked against the minimum and maximum value range.

Known solver axes include:

- temperature
- frequency
- Cartesian spatial axes: x, y, z
- Cylindrical spatial axes: r, phi, z
- Spherical spatial axes: r, theta, phi

Name

See GEMSIF Object Properties - Name

Title

Title of the axis in question. This specifies the type of axis, such as spherical r or normalized x.

Axis Units

Units for the entered data.

Direct Offset

Offset for the minimum and maximum values.

Default Value

Default value for unspecified axis.

Minimum

Minimum value of the axis.

Maximum

Maximum value of the axis.

Extrapolation

Specifies if data should be extrapolated between points along the axis.

Field Properties

Data

Displays GEMSIF’s evaluation of the material properties.

Field

Title of the material property. This specifies the type of material property, such as eps for permeability.

Format

Specifies the type of field. Changing this property value will change the field type, and may cause any entered data to be lost.

Header

Description of the table column headers.

High-frequency Permittivity

Represents the asymptotic high-frequency permittivity for the Drude and Lorentz-Drude models.

Minimum Frequency

Sets the minimum frequency bound for which the coefficients of the Lorentz-Drude, Drude, and Debye models are accurate. The field will generate an error if frequencies outside the specified range are requested.

Maximum Frequency

Sets the maximum frequency bound for which the coefficients of the Lorentz-Drude, Drude, and Debye models are accurate. The field will generate an error if frequencies outside the specified range are requested

Plasma Frequency

The plasma frequency (in Hz units, not rad/sec units) of the Drude model or Lorentz-Drude resonator term.

Rows

Number of rows in a field’s table

Size

Specifies the material characteristic’s size: isotropic, vector, or anisotropic.

Time Constant

Relaxation time constant for the Debye, Drude, or Lorentz-Drude models. Can also be interpreted as the inverse damping frequency. Commonly determined experimentally, although it can be predicted analytically for some materials.

4.4.8. Material Mixtures

GEMSIF supports binary and ternary material mixtures. The properties of a mixture are determined based on a linear interpolation of each the component material's properties and weighted by the mixing percentages.

Mixture Properties

- Name - See GEMSIF Object Properties
- Color - See GEMSIF Object Properties
- Transparency - See GEMSIF Object Properties
- Source - See Material Properties
- Notes - See Material Properties
- Coordinate System - See Material Properties
- Relative Origin - See Material Properties
- Orientation - See Material Properties
- Material 1 - Reference of the first material to be used in a mixture.
- Material 2 - Reference of the second material to be used in a mixture.
- Material 3 - Reference of the third material to be used in a mixture.

4.5. Parameters

Parameters are user-specified scoped variables that may be used in expressions to set object properties. A Parameter defined in a parent object (Application or System) will be visible to all children (Systems or Simulations). Parameters are referenced by name. Changing the value of a parameter will automatically update all of the referencing properties. Parameters defined at lower levels will override the value of those at higher levels with matching names, allowing simulations to define their own parameters independent of whether or not the parameter has already been defined by the System or Application. Details…

4.5.1. Properties

Name

Parameters are referenced by their name to assign property values or expressions and in the definition of derived Parameters.

A parameter may not be recursively defined or have any circular references to itself; that is, a parameter's value may not reference itself, either directly or indirectly.

For restrictions on Parameter names, see also Object Names.

Editor

Type of editor to provide for the parameter input when an object is imported from a library.

Evaluated

Not user-editable. Automatically updated with the parsed value of the parameter after every change.

Immutable

Boolean-valued - This property is only used when importProperty is true. If true, the parameter will be converted to an immutable property when imported from a library.

This property is only available when editing the properties of a specific parameter, not when working from the Parameter dock.

importProperty

Boolean-valued - when true, this parameter will be converted to an editable property when imported from a library.

This property is only available when editing the properties of a specific parameter, not when working from the Parameter dock.

Position

Defines the order in which the parameters will appear in the editor for an imported object.

Title

String-valued - when importProperty is true, the string value of this property will be used as the title for the imported property.

This property is only available when editing the properties of a specific parameter, not when working from the Parameter dock.

Validation Size

Specify a validation condition on the size of the parameter’s value. Request that the parameter value be validated to take on one of the following enumerated values:

[Empty String]
  No validation is performed.

Any
  No validation is performed

Scalar

The parameter must be a scalar value, not an array of any size.

Array

The parameter must be an array value of an unspecified size.

Vector2

The parameter must be an array value of length 2.

Vector3

The parameter must be an array value of length 3.

Vector4

The parameter must be an array value of length 4.

Vector5

The parameter must be an array value of length 5.

This property is only available when editing the properties of a specific parameter, not when working from the Parameter dock.

valueType

Specify a validation condition on the parsed type of the parameter's value. Request that each element of the parameter value be validated to take on one of the following enumerated values:

[Empty String]

Do not constrain the parameter value.

Any

Do not constrain the parameter value.

Boolean

Constrain parameter value to be convertible to bool (true/false), or an integer that is interpreted as a boolean.

Integer

Constrain the parameter value to be convertible to an integer value.

Double

Constrain the parameter value to be convertible to a real-valued decimal numeric quantity

Complex

Constrain the parameter value to be convertible to a complex numeric quantity.

Angle

Constrain the parameter value to take on units of angle [rad] or [deg].

Time

Constrain the parameter value to take on units of time [s].

Length

Constrain the parameter value to take on units of length [m], [in], etc.

Mass

Constrain the parameter value to take on units of mass [g].

Intensity

Constrain the parameter value to take on units of candela [cd].

Substance

Constrain the parameter value to take on units of moles [mol].

Current

Constrain the parameter value to take on units of current [A].

Temperature

Constrain the parameter value to take on units of temperature [K].

Potential

Constrain the parameter value to take on units of electrical potential [V].

Frequency

Constrain the parameter value to take on units of frequency [Hz].

Impedance

Constrain the parameter value to take on units of electrical impedance [Ohm].

Power

Constrain the parameter value to take on units of power [W].

Energy

Constrain the parameter value to take on units of energy [J].

Capacitance

Constrain the parameter value to take on units of capacitance [F].

Charge

Constrain the parameter value to take on units of charge [C].

Conductance

Constrain the parameter value to take on units of conductance [S].

Inductance

Constrain the parameter value to take on units of inductance [H].

Pressure

Constrain the parameter value to take on units of pressure [P].

Conductivity

Constrain the parameter value to take on units of Conductivity [S/m].

This property is only available when editing the properties of a specific parameter, not when working from the Parameter dock.

Value

Parameters may be one of several data types. The parsed or recognized data type will be displayed in the Parameter editor; the user may also specify the Parameter’s type explicitly, which will generate errors on parsing failures. Depending on the type of the Parameter, different expressions will be valid. For example, strings may be concatenated with the '+' operator, but may not be divided like any of the numerical operators.

A Parameter’s value take on any of the following formats:

Empty

When value is empty or blank, it is treated as a string. Any mathematical operation applied to an empty Parameter will return an error.

Integer

Any integer quantity. By default, all integers are parsed as decimal numbers (10), but hexadecimal (0x10 = 16) and octal (0b10 = 8) values are also allowed.

Real Number

A number with a decimal point (123.4) and/or expressed in scientific notation (1.234e2).

Complex Number

A real number with real and complex components, specified as [real]+[imag]i (eg, 1+2i). See the scripting documentation for additional syntax and functions relating to complex values. For permeability and permittivity, loss must be represented as x-yi.

Dimensioned Number

A dimensioned quantity is the combination of a number and one or more unit names. For example, one meter would be entered as (1 m). Compound dimensions consisting of positive powers of base units are specified by entering several unit names, separated by a space (3.2 m kg). Dimensions with multiple powers can be expressed with an exponent on the unit (eg, 3.2 m**2) Compound dimensions that require quotients of units must be expressed as a quotient of dimensioned numbers (6 m / 1 s), or using negative powers on the units (9.8 m s**-2). Details...

String

Any value that cannot be clearly read as one of the other defined types will be treated as a string. Mathematical operations on a string may have nonintuitive results, and should be avoided. When using a string, place the value in " " for entries that require a string.

Array

A Parameter may contain an array of any other valid values. An array value consists of a comma-separated list of values, enclosed by square braces. (eg, [1,2,3]) Note that arrays must have homogeneous types, and that strings in an array must be quoted. All of the scripting commands for arrays are valid for Array-valued Parameters. There are also sprecialized arrays called vectors for describing 3D locations or orientations. Details...

Expression

Any inline scripting expression may be provided as the value of a Parameter. Such expressions only have access to other Parameter values, not to the Application object or any child data objects. Example: param1 + sin(param2) * exp(1i * cos(param3)) Details...

Javascript function

Custom javascript functions may be defined in a parameter, as well. Such functions should have no saved variables; their output should depend only on their arguments. During function evaluation, the Application object and other data objects are not accessible. Details...

Vector

A Vector is a specialized 3-element array for specifying 3D locations or directions, and has several available formats.

Position Vector

Vectors specifying positions in space can take the following forms:

- An array of length values like **[1 mm, 0 mm, -1 mm]**. Values are specified in the order [x,y,z].
- An axis position such as **zaxis(5 mm)**. Axis positions can be added together, such as **zaxis(-3 cm) + yaxis(1 mm)**.
- Cartesian coordinates such as **cartesian(1 mm, 2 mm, -3 mm)**. The example given would be identical to the array **[1 mm, 2 mm, -3 mm]**.
- Cylindrical coordinates such as **cylindrical(5 mm, 90 deg, -5 mm)**. The values are expressed in order rho (radial distance), phi (azimuthal angle), and z (axial distance or height).
- Spherical coordinates such as **spherical(5 mm, 30 deg, 10 deg)**. The values are expressed in order r (radial distance), theta (polar angle), and phi (azimuthal angle).

Orientation Vector

Vectors specifying orientations or directions in space can take the following forms:

- An array of dimensionless values like **[0,0,1]**. Values are in the order [x,y,z].
- An axis specification such as **zaxis()**. Axis specifications can be added together and weighted, such as **zaxis() + 0.5*yaxis()**.
- A tilt specification such as **tilt(45 deg, 180 deg)**. The first value is the polar angle (theta), the second is the azimuthal angle (phi). If only one value is given, it is taken to be the polar angle, and the azimuthal angle is set to 0 degrees.

4.5.2. Details

Systems, like the Application node, allow the definition of Parameters that are accessible for all of their child objects and simulations. Parameters are edited through the Parameters Dock, which will automatically show the parameters belonging to the most recently selected Simulation, System or Application. The use of parameters at the System, rather than the Simulation level, will be useful when performing simulations of the same or slightly modified geometry in multiple simulations. Defining the geometry parameterization variables in the system rather than the simulation itself allows the settings of all of the simulations to be modified simultaneously by changing the parameter value from the parent System.

Parameters and Libraries

Parameters referenced by imported data objects can become the editable properties of the object, while the original properties of the imported objects are hidden. Parameters used in this way will transfer their default value and validation conditions from the parameter metadata properties (importProperty, importPropertyTitle, importPropertyImmutable, valueType, valueSizeType) of the imported parameter. These properties are typically only useful when creating a custom object library.

4.6. Settings

The built-in GEMSIF settings control and customize the behavior of the GUI.

Installed plugins often provide their own tabs for the settings editor.

4.6.1. Application

Application settings consist of all of the objects with which hotkeys are associated. Users can also specify their own actions and bind them to hotkeys.

Application Properties

Name

See GEMSIF Object Properties - Name

About GEMSIF

Launches the "About GEMSIF" pop up. No default keyboard shortcut.

Action Binding

Action bindings allow the user to write a custom action and then bind the action to a keyboard shortcut.

Action Binding Properties:

- Name - See GEMSIF Object Properties.
- Command - The action script to be hotkeyed.
- Shortcut - The hotkey combination to link to the action.

Cascade Window

Cascades all open views in the viewing area. No default keyboard shortcut.

Close All Systems

Closes all open systems. The user will be prompted to save unsaved systems. No default keyboard shortcut.

Close System

Closes the selected system. The user will be prompted to save unsaved systems. Default keyboard shortcut: CTRL+F4.

Copy

Copies the selected object to the clipboard. Copy does not work for collections (groups of objects). Default keyboard shortcut: CTRL+C

Cut

Moves the selected object to the clipboard. Cut does not work for collections (groups of objects). Default keyboard shortcut: CTRL+X.

Help Contents

Loads the Help documentation window. Default keyboard shortcut: F1.

Help Context

Loads the Documentation Dock. No default keyboard shortcut.

Help Index

Loads the Help documentation window. No default keyboard shortcut.

Help Search

Loads a searchable version of the Help documentation window. No default keyboard shortcut.

Help Tutorials

Loads a dock with a tutorial selected. No default keyboard shortcut.

Interface Language

Allows the user to change the interface language. Default language is English. No default keyboard shortcut.

License Information

Loads the License Information Window. No default keyboard shortcut.

Link to Parameter

Allows the user to specify a parameter and value to be used for a selected property. The parameter is automatically added to the Parameter Dock. Default keyboard shortcut: CTRL+SHIFT+L

Move Backward

Default keyboard shortcut: CTRL+LEFT.

Move Forward

Default keyboard shortcut: CTRL+RIGHT.

New Object

Launches Object Creation Dialog. Default keyboard shortcut: CTRL+M.

New Simulation

Launches dialog box to create a simulation. Default keyboard shortcut: CTRL+SHIFT+N.

Creates a new System Object. Default keyboard shortcut: CTRL+N.

Open Keyboard Shortcut Editor

Launches the keyboard shortcut editor. No default keyboard shortcut.

Open Settings Editor

Launches the Application Settings Window. No default keyboard shortcut.

Open System

Launches a load window for selecting previously saved systems. Default keyboard shortcut: CTRL+O.

Paste

Pastes an object from the clipboard. Default keyboard shortcut: CTRL+V.

Quit GEMSIF

Closes GEMSIF. No default keyboard shortcut.

Redo

Reverts back to a future state after Undo has been used. Default keyboard shortcut: CTRL+Y.

Run Simulation

Runs a selected simulation (all excitations). Default keyboard shortcut: F5.

Save All Systems

Launches the Save Window for each individual system. Default keyboard shortcut: CTRL+SHIFT+S.

Save As

Launches the Save Window. No default keyboard shortcut.

Save System

Saves the currently selected system. Default keyboard shortcut: CTRL+S.

Show Views as Tabs

Switches between showing Views as windows in the viewing area and showing Views as tabbed windows.

Show Welcome Screen at Startup

Allows the user to enable/disable the start splash screen.

Tile Windows

Tiles allow active Views in the viewing area. No default keyboard shortcut.

Undo

Undoes the last action. Default keyboard shortcut: CTRL+Z.

Validate Simulations

Validates a selected simulation (all excitations). Default keyboard shortcut: F4.

Zoom to Hierarchy

Moves the user selection back to the Hierarchy Dock. Default keyboard shortcut: CTRL+SHIFT+H.

Zoom to Parameter

Moves the user selection back to the Parameter Dock. Default keyboard shortcut: CTRL+SHIFT+P.

Zoom to Properties

Moves the user selection back to the Property Dock. Default keyboard shortcut: CTRL+SHIFT+E.

Zoom to Terminal

Moves the user selection back to the Script Console Dock. Default keyboard shortcut: CTRL+SHIFT+T.

Window Geometry

Stores the dimensions of an active View.

Window Maximized

Controls if a view is maximized.

4.6.2. Libraries

Libraries allow the user to import prebuilt objects into their simulations. Users can either create their own custom libraries or use libraries that are provided with GEMSIF.

Library Properties

Library Properties can be edited from the Settings Window or the Property Editor and pertain to all libraries.

Name

See GEMSIF Object Properties - Name

Libraries

List of libraries that are selected for use. These will appear in the library dock and the GEMSIF Hierarchy Dock when "Select All" is active.

Search Paths

Search Paths provide the locations and the order in which GEMSIF will load libraries and their objects.

Shortcuts

List of Object shortcuts available.

4.6.3. License

GEMSIF requires a valid license to operate. A solver license is typically provided as part of the GEMSIF license. License configuration normally occurs when GEMSIF is run for the first time or when a license has expired.

License Properties

Name

See GEMSIF Object Properties - Name

Location

Path to the location of the license file on the user’s computer.

License Type

Specifies the type of license file to use.

4.6.4. Parallel Processing

The Parallel Processing settings allow the user to controls how many cores GEMSIF solvers will have access to.

Parallel Processing Properties

Name

See GEMSIF Object Properties - Name

Enable Parallel Processing

Allows the user to turn parallel processing on or off.

Maximum Cores

Selects the number of threads that GEMSIF will have access to.

4.7. Project Properties

Project properties contain file- and simulation-specific history and editing information. Some of this information is editable by the user, other information is automatically generated.

4.7.1. Name

See GEMSIF Object Properties - Name

4.7.2. Accessed

Displays the date and time of the last time that the file was viewed (not implemented). Not user-editable.

4.7.3. Authors

Contains a list of all authors that have edited and saved the current file or simulation. Automatically-generated, not user-editable.

4.7.4. Company

Displays the company (from the license file) or owner of the original file or simulation. Not user-editable.

4.7.5. Created

Displays the date and time of the initial file creation. Not user-editable.

4.7.6. Description

Free-form text description of the file or simulation. This field is useful primarily when creating a custom object library

4.7.7. Angle

Select the display units of any angle properties in GEMSIF.

4.7.8. Capacitance

Select the display units of any capacitance properties in GEMSIF.

4.7.9. Current

Select the display units of any current properties in GEMSIF.

4.7.10. Frequency

Select the display units of any frequency properties in GEMSIF.

4.7.11. Impedance

Select the display units of any impedance properties in GEMSIF.

4.7.12. Inductance

Select the display units of any inductance properties in GEMSIF.

4.7.13. Length

Select the display units of any length properties in GEMSIF.

4.7.14. Mass

Select the display units of any mass properties in GEMSIF.

4.7.15. Potential

Select the display units of any potential properties in GEMSIF.

4.7.16. Time

Select the display units of any time properties in GEMSIF.

4.7.17. Unit System

Select from a number of preset collections of default units.

Blank

No display units are selected. The units of displayed and computed values in GEMSIF will be determined automatically.

Custom

No preset unit selection is in use.

SI

Convert all dimensioned quantities to their SI base units.

CGS

Convert all dimensioned quantities to the centimeter, gram, second unit system.

RF

Select degrees, m, us, MHz, ohm, uF, mH, V, and A as display units.

Microwave

Select deg, mm, ns, GHz, ohm, nbF, uH, V, and A as display units.

Infrared

Select deg, um, ps, THz, V, and A as display units.

Optical

Select deg, nm, ps, THz, V, and A as display units.

PCB

Set deg, mil, ms, kg, ohm, uF, mH, V, and A as display units.

4.7.18. Editing Time

Displays the cumulative time that the file has been open in the GEMSIF program. Not user-editable.

4.7.19. Image

Saved image data that represents the individual project file.

4.7.20. Last Saved By

Displays the name of the last author to make changes to the file. Not user-editable.

4.7.21. Modified

Displays the date and time of the last time that the file was modified. Not user-editable.

4.7.22. Tags

A list of string values describing the file or project that can be used to sort and filter browsing results.

4.7.23. Template

Displays the template file used to create the current object. Not user-editable.

4.7.24. Title

In addition to the object name, a short project title can be specified. This title will be used when rendering and exporting plots, drawings, and images. This field is useful primarily when creating a custom object library.

4.8. See Also

- GEMSIF Simulations

5. Docks

The GEMSIF GUI makes some functionality available through dockable windows surrounding the main MDI window area. The built-in settings include a default dock layout for GEMSIF, but users may customize the number, size, and position of the docks to suit their workflow by adding, removing, and moving docks around the GUI window. Dock windows may also be undocked to float above the GUI by clicking the arrow icon in the upper-right corner. Details...

5.1. Dock Types

- Documentation - Provides a context sensitive viewer for documentation.
- Editing History - Provides a visual tree of past actions performed.
- Library - Displays selected libraries and their imported objects.
- Model Hierarchy - Displays the model hierarchy for all active systems and simulations.
- Parameters - Displays a hierarchy sensitive list of available parameters.
- Property Editor - Provides an interface for editing object properties.
- Script Console - Provides terminals for using javascript based commands.
- Simulation and Optimization Progress - Displays simulation progress when simulation is running.
- Status Dock - Displays user messages and warnings from GEMSIF.

5.1.1. Details

Docks store their display information under the root Application node, where it is available from the scripting interface. The list of open docks is hidden by default in the Hierarchy display, but this may be changed in the properties of the open Hierarchy dock. Docks may be closed by clicking the 'x' in their upper-right corner or by deleting their corresponding data object from the Application object. Docks can also have their editing modes and references locked by right clicking at the top of the dock. New docks may be added to the interface from the Window→Add Dock menu.

5.2. Documentation Dock

The Documentation Dock allows the user to view the HTML based documentation. It will automatically display the documentation for the last selected object in the interface. In addition, the currently displayed document may be locked by clicking the lock button to prevent future selections or mouse actions from changing the current display.

Navigate the documentation by following the cross-referenced links in the text.

The address field at the top of the dock may be used to directly enter a documentation address, but searching is not yet supported.

More advanced forms of the Documentation viewer can be accessed by selecting items under Help on the top menu bar or by pressing F1.

5.2.1. Documentation Dock Properties

Name

See GEMSIF Object Properties - Name.

Floating

See Dock Properties - Floating.

Home

Specifies location that the home button is addressed to. Default location is the GEMSIF topical documentation.

Location

See Dock Properties - Location.

Lock

Disable or enable context sensitive tracking. True or false.

Next Dock

See Dock Properties - Next Dock.

Pinned

See Dock Properties - Pinned.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

5.3. Editing History

The Editing History dock shows the set of actions that have been performed to the current System since it was created or loaded from a file. An icon shows the last action that was saved to a file. The Undo and Redo commands from the file menu cause the list of actions to be updated to reflect the current state of the system.

The history display shows the Undo history for the currently selected System or Application. The "current" System or Application is updated whenever any object is selected in the GEMSIF interface to be the nearest parent of the selected object.

5.3.1. Editing History Dock Properties

Name

See GEMSIF Object Properties - Name.

Floating

See Dock Properties - Floating.

Location

See Dock Properties - Location.

Next Dock

See Dock Properties - Next Dock.

Pinned

See Dock Properties - Pinned.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

5.4. Library

The Library Dock displays all of active libraries and their child objects. Right clicking on an object provides the user with options on importing the object. Right clicking on the top of the dock allows the user to access the Library Settings Window.

5.4.1. Library Properties

Name

See GEMSIF Object Properties - Name.

Floating

See Dock Properties - Floating.

Location

See Dock Properties - Location.

Next Dock

See Dock Properties - Next Dock.

Pinned

See Dock Properties - Pinned.

Show All

See Dock Properties - Show All.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

5.5. Model Hierarchy Dock

The Hierarchy dock shows all open files and simulations, as well as any objects and properties of those simulations. Selecting items by single-clicking from the tree of entries will open the editor for that object. Right-clicking on entries in the list show the possible actions that may be performed from that element. New systems, simulations, and objects may be created by right-clicking on the desired parent object and selecting the appropriate action from the context menu.

The Application node is the root of the hierarchy.

All GEMSIF objects exist within the hierarchy. To show all of the objects that currently exist in the hierarchy, either use the dock drop down menu or right click on the dock header and select "Show All".

5.5.1. Hierarchy Dock Properties

Name

See GEMSIF Object Properties - Name.

Floating

See Dock Properties - Floating.

Location

See Dock Properties - Location.

Next Dock

See Dock Properties - Next Dock.

Pinned

See Dock Properties - Pinned.

Show All

See Dock Properties - Show All.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

5.6. Parameter Dock

The Parameter Dock is where the user creates, edits, and views parameters. To create a parameter, right click in the blank area of the dock and select "New Parameter". To edit a parameter, double-click the Name or Value columns to edit their values.

The dock will automatically display the parameters of the current node. To prevent this, use the dock menu to select "Lock Display Level". The dock can also display parameters for nodes not currently selected by using the menu and setting the current level.

5.6.1. Parameter Dock Properties

Name

See GEMSIF Object Properties - Name.

Floating

See Dock Properties - Floating.

Edit Inherited

Allows the user to edit the inherited properties without having to navigate to the hierarchy level at which they are stored.

Location

See Dock Properties - Location.

Next Dock

See Dock Properties - Next Dock.

Parameter Column Width

Stores the width setting for columns containing parameters.

Pinned

See Dock Properties - Pinned.

Show Inherited

Shows all parameters that will be inherited at a particular node.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

5.7. Property Editor

The properties of any GEMSIF data object may be modified from this dock. The Property editor provides two tabs: a standard editor, and a grid-based editor. The standard editor provides a user friendly interface and allows the user to display or hide advanced options. It also allows the user to display property values that are derived from parameters using review mode. The Grid editor allows for more advanced inputs and feedback on what GEMSIF is interpreting inputs as, but it is not user friendly as the standard editor. The editor type may be locked by right-clicking the dock's title, and selecting the "Lock" menu button. When locked, the editor may be manually switched from grid to standard mode and vice versa.

5.7.1. Property Editor Dock Properties

Name

See GEMSIF Object Properties - Name.

Column Width

Stores the set width of columns in the Property Editor.

Floating

See Dock Properties - Floating.

Location

See Dock Properties - Location.

Next Dock

See Dock Properties - Next Dock.

Pinned

See Dock Properties - Pinned.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

5.8. Script Console

The scripting console allows interaction with one or more independent scripting sessions. A default session is created at startup, but new sessions can be started by clicking the "plus" button in the upper-right corner of the script dock, and existing sessions may be closed by clicking on the button on each tab.

The Script Console uses a modified version of javascript. Commands and a discuss of the use of the scripting language can be found in the API documentation.

Actions are run through the scripting console. If an action is run, but no script console is open in the GEMSIF window, then a new scripting dock will be added.

5.8.1. Script Console Properties

Name

See GEMSIF Object Properties - Name.

Floating

See Dock Properties - Floating.

Location

See Dock Properties - Location.

Next Dock

See Dock Properties - Next Dock.

Pinned

See Dock Properties - Pinned.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

5.9. Simulation Progress and Optimization

The progress dock shows all pending, running, and finished simulation tasks that have been requested to run through the GEMSIF interface. This dock may be opened from the Window menu, but also by double-clicking the overall completion progress bar that appears in the lower-right hand corner of the GEMSIF window once any simulation or optimization has been started.

5.9.1. Progress Dock Properties

Name

See GEMSIF Object Properties - Name.

Channel Optimization

Advanced Property: Determines if the Progress Dock will report optimization progress.

Channel Other

Advanced Property: Determines if the Progress Dock will report other simulation progress.

Channel Simulation

Advanced Property: Determines if the Progress Dock will report simulation progress.

Floating

See Dock Properties - Floating.

Location

See Dock Properties - Location.

Next Dock

See Dock Properties - Next Dock.

Pinned

See Dock Properties - Pinned.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

5.10. Status

Any errors or messages from the GEMSIF application are displayed in the Status Display Dock. The menu buttons above the list of messages allow the list to be filtered by source or by severity. Error messages are generated whenever a property is edited with an invalid or incorrect value. The error messages are intended to provide sufficient information to resolve the problem. Note that simulations or optimizations will not run as long as any validation errors exist.

Selecting a message from the list expands the display so that the entire message may be read. Double clicking a message will take you to the object that contains the error and the property that is producing the error should be highlighted red.

5.10.1. Status Dock Properties

Name

See GEMSIF Object Properties - Name.

Channel General

Determines if general information will be displayed in the Status Dock.

Channel Progress

Determines if progress information will be displayed in the Status Dock.

Channel Scripting

Determines if scripting information will be displayed in the Status Dock.

Channel Validation

Determines if validation information will be displayed in the Status Dock.

Floating

See Dock Properties - Floating.

Location

See Dock Properties - Location.

Next Dock

See Dock Properties - Next Dock.

Pinned

See Dock Properties - Pinned.

Severity Debug

Determines if debug messages will be displayed in the Status Dock.

Severity Details

Determines if details messages will be displayed in the Status Dock.

Severity Error

Determines if general error messages will be displayed in the Status Dock.

Severity Fatal

Determines if fatal error messages will be displayed in the Status Dock.

Severity Information

Determines if information messages will be displayed in the Status Dock.

Severity Warning

Determines if warning messages will be displayed in the Status Dock.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

5.10.2. Dock Properties

Floating

Determines if a dock is floating or docked in GEMSIF. Options are true or false.

Location

Shows which edge of the GEMSIF window the dock is docked to. Options are N, S, W, E.

Next Dock

Internal dock location property.

Pinned

Determines if a dock is minimized or pinned to the edge of the GEMSIF Window. Options are true or false.

Show All

Displays all of the hidden advanced objects that are normally hidden in the hierarchy.

Size

Internal dock size property.

Tabbed

Determines if the dock is used in tab mode with other docks. Options are true or false.

Title

The visible title of the dock as it is displayed in the interface.

6. Views

Views are subwindows that typically display output data from GEMSIF. Every solver has a collection of views that can be created within its simulations. The list of views is saved with each System file and Application session, and any open windows are reopened when the file is reloaded or the program restarted.

6.1. View Types

- Plot2DView

6.2. Plot2DView

Displays 2D plot results from simulations. Plot2D Views contain trace objects which are used to generate lines on the plot. The user can also add annotations and straight guide lines objects to plots.

6.2.1. Plot2D Properties

Equal Axis Scaling

Scales both axes by the same increments.

Export DPI

Sets the image DPI size for image exports.

Export Size

Sets the physical dimensions of the exported image.

Font

Sets the font type for Plot text.

Geometry

See View Properties - Geometry.

Legend Font

Sets the font style for text in the legend.

Legend Position

Stores the position of the legend on the plot.

Show Legend

Enables or disables the legend on the Plot2D View.

State

See View Properties - State.

Tick Font

Sets the font of the tick markers on the axes.

Tight Auto Axis Limits

Sets the Axes limits to the minimum and maximum data values for both axes.

Title

See View Properties - Title.

Title Font

Sets the font style for the plot title.

Visible

See View Properties - Visible.

Max Sub-Traces

Each Trace will only display this many sub-traces. Additional sub-traces will be truncated. (This is an advanced property that can be shown by clicking the double-gear icon in the Property Editor.)

X Label

Provides a custom X axis label.

X Limits

Provides custom X axis limits.

X-Axis Reversed

Reverses the direction of the X-axis to display decreasing instead of increasing values from left to right.

Y Label

Provides a custom Y axis label.

Y Limits

Provides custom Y axis limits.

Y-Axis Reversed

Reverses the direction of the Y-axis to display increasing instead of decreasing values from top to bottom.

6.2.2. Traces

Traces consist of plotted data of a simulation. Plot2D Views may contain several traces.

Properties

Caption

Sets the description of the trace that will appear in the legend.

Include in Legend

Enable or disable the trace caption displaying in the Plot2D View’s legend.

Data Selection

Contains the path to the source data for a trace.

X Data

Data to be plotted along the X axis.

Y Data

Data to be plotted along the Y axis.

Sweep Coordinate

Determines the order in which data will be plotted. The sweep coordinate typically is the X Data set.

Line Color Map

Determines the color map used for coloring the sub-traces. If set to "Single", then all sub-traces are given the same color, specified by the Line Color property.

Line Color

Determines the color of the trace if the Color Map is set to Single.

Line Transparency

Sets the level of transparency of a given trace. Values between 0 and 1.

Line Color Map Coordinate

Determines which column(s) is used to color sub-traces based on the selected Color Map. By default (the "All" option), each sub-trace is assigned a unique color in the color map. To group the sub-traces by a specific attribute, select the relevant column. For instance, setting this to "Wavelength" will give all sub-traces with the same wavelength the same color. (This is an advanced property that can be shown by clicking the double-gear icon in the Property Editor.)

Line Connect Style

Determines the style of line used to connect between data points.

Line Width

Determines the thickness of a trace.

Line Style

Determines the style of a line. For example, solid or dashed.

Marker Style

Sets the style of the data markers.

Marker Line Color

Sets the color of the data markers.

Marker Transparency

Sets the level of transparency of a given data marker. Values between 0 and 1.

Marker Size

Determines the size of the data markers.

Marker Line Width

Determines the thickness of the data markers.

6.2.3. Annotations

The Annotations Collection objects such as text annotations and guide lines for Plot2DViews.

6.2.4. View Properties

Geometry

Determines the size and location of the View inside the viewing area.

State

Determines if the View is Windowed or not.

Title

Stores the title of the View.

Visible

Stores if the view is visible or hidden (true or false).

7. Javascript Scripting Documentation and API

The GEMSIF scripting language uses an augmented version of the common Javascript language. Javascript is a loosely-typed, dynamic, and flexible language that belongs to the C-like family of programming languages. Although Javascript is most well known for its role as the primary scripting language in web browsers, it can also be implemented in standalone applications to provide automation and customization.

GEMSIF exposes all editing and execution capability of the GUI through the scripting API as well. By issuing scripting commands at the console, or by collecting a series of scripting commands into an Action macro, tasks can be automated or customized to meet a personalized workflow. Custom scripts may be saved into the default session, or stored with a System file or simulation.

GEMSIF extends the Google V8 Javascript virtual machine to support dimensioned quantities as part of the language. This addition allows expressions and calculations like the following:

```
>> 1 m / 2 s
<< 0.5 m s^-1

>> 3 m - 2 in
<< 2.9492000274319854 m
```

Properties and parameters within the GEMSIF application are specified in terms of these dimensioned quantities.

7.1. Scripting Language Syntax

Except for the addition of the dimensioned quantities, the supported scripting language is similar to other common Javascript implementations. As a supplement to the information provided in this documentation, the "Mozilla Developer Network" provides reference documentation on the Javascript language itself, although the examples are focused on web applications.

7.1.1. Object Naming Restrictions

GEMSIF object names may only contain alphanumeric characters (as defined by the current machine locale) and underscores, and a numeric character may not be the first character in an object name. Spaces are not permitted. So, for example, you could name the second ray in a RayBundle: Ray2, RayTwo, _Ray_Two_, second_Ray, etc., but not: 2ndRay, Ray#2, Ray 2, ray(2), ray@position2, Ray-2, etc.

Additionally, the following names are reserved and may not be used for object names:

Javascript Keywords

break, case, catch, continue, debugger, default, delete, do, else, finally, for, function, if, in, instanceof, new, return, switch, this, throw, try, typeof, var, void, while, with

GEMSIF Script Reserved Names

abs, acos, add, all, any, Application, arg, Arithmetic, Arithmetic_Impl, asin, atan, atan2, c0, cartesian, ceil, conj, cos, cylindrical, decenter, Dimensions_Impl, div, dot, e, eps0, eq, eta0, exp, feq, floor, ge, gt, le, ln, log, lt, max, min, mod, mu0, mul, ne, neg, norm, not, pi, pow, prod, Quantity, round, sgn, Simulation, sin, spherical, sqrt, sub, sum, System, tan, test, tilt, Unit, xaxis, yaxis, zaxis

Other Reserved Names

gemsif

Name checks are case-sensitive, so while "delete" would not be allowed, "Delete" or "DELETE" would not cause a validation error. However, to prevent confusion it is best to avoid using such variants of reserved words.

8. Design Studies

Design studies are optimizers that aide in finding designs that fit a certain goal by varying parameters/variables to generated a candidate design. GEMSIF includes local gradient-free, local gradient-based, and global optimization algorithms.

Enable for Run All

If checked, the optimization will run whenever the simulation’s "Run All" action is performed.

Save Optimization Logs

If checked, the optimizer will write a `.restart` file to save its current progress in the event of a crash or unexpected termination. Not all optimizations make use of this. See Restart.

Restart

If checked, the optimizer will continue the optimization at the point saved by the most recent `.restart` file. Not all optimizations make use of this. See Save Optimization Logs.

History Table

Every optimizer records its optimization path in a history table for analysis.

8.1. CMAES Optimizer

CMAES is a global gradient-free optimizer using the Covariance Matrix Adaptation Evolution Strategy algorithm. Each generation of the optimization contains a random sampling of candidate designs. Subsequent generations have a restricted sampling range based on the performance of the previous generation.

Sample Size

The number of candidate designs to evaluate in a generation. If this number is less than two, a sample size will be automatically calculated based on the number of parameters in the optimization.

Cost Convergence Tolerance

The optimization will end if a candidate design cannot be found that is better by a sufficient proportion determined by the cost tolerance.

Function Call Limit

The optimization will stop after evaluating this many candidate designs. The number of evaluated designs will always be a multiple of the sample size, so this limit may be exceeded until the end of the current generation.

Generation Limit

The optimization will end after this many generations.

Random Seed

If this value is zero, the sampling for each generation will be random every run of the optimization. If the value is non-zero, the sampling will be in a fixed random pattern, and assuming the rest of the simulation has not changed, will repeat for every optimization run.

Include Starting Parameters

If checked, the current simulation state will be included as a candidate design in the first generation of the optimization. This will ensure the final optimized design will be at least as good as the current setup.

Seed with Starting Parameters

If checked, the current simulation state will influence the sampling for the first generation.

History Save Interval

When the History Save Interval is set to X, the first X generations will be recorded in the history table. After that, only every X'th generation will be reported in the history table. Generations that discover a new best cost, and the last generation of the optimization, will be reported regardless. If the History Save Interval is set to 1, then every generation is recorded. This can be overridden by the Verbose History property.

Restart Save Interval

The `.restart` file will be written once every this many generations, instead of every generation.

Remain within Initial Bounds

If checked, the sampling of candidate designs will be forced within the ranges specified by the optimization's parameter ranges. This is done by resampling any outlying members of a generation until they are within the bounds. This can negatively influence the optimization if the best design is located at or outside of the parameter ranges.

Verbose History

If checked, every generation will be recorded in the history table, as well as the individual function evaluations that make up the generations. History table entries corresponding to generations will be marked with a 1 in the "GenerationSummary" column, and function evaluations will be marked with a 0.

8.2. Damped Least-Squares Optimizer

The Damped Least-Squares optimizer is a local, gradient-based optimizer using the Damped Least-Squares (DLS) algorithm, aka the Levenberg-Marquardt algorithm. Every step of the optimization is taken based on the cost gradient at the current candidate design. The cost gradient is computed via finite-differencing.

Max Function Evaluations

The optimization will stop after evaluating this many candidate designs. This does not include the candidate designs evaluated during finite-differencing to compute the cost gradient.

Max Iterations

The optimization will end after this many steps. A single iteration may evaluate multiple design studies if line-searching must be performed to ensure the parameters remain within their bounds. If parameter ranges are not enforced, each iteration should contain a single function evaluation. See Remain within Initial Bounds.

Initial Trust Region Radius

Sets the initial size of the DLS trust region, which is used when approximating the shape of the cost function during the optimization.

Function Tolerance

The optimization will end if an optimization step cannot change the cost by at least this much.

Parameter Tolerance

The optimization will end if the magnitude of the last optimization step's change in the candidate design's parameter values was less than or equal to this tolerance.

Gradient Tolerance

The optimization will end if the cost gradient at the current step has a magnitude less than or equal to this tolerance.

Use Nonmonotonic Steps

If checked, the optimization may take steps that increase the cost of the candidate design, rather than decrease it. This can help the optimization pass through local minima to find a better minimum. See Max Consecutive Nonmonotonic Steps.

Max Consecutive Nonmonotonic Steps

If Use Nonmonotonic Steps is checked, the optimization may take up to this many steps that increase the cost of the candidate design, before it must take a step that decreases the cost.

History Save Interval

When the History Save Interval is set to X, the first X function evaluations will be recorded in the history table. After that, only every X'th function evaluation will be reported in the history table. Function evaluations that discover a new best cost, and the last function evaluation of the optimization, will be reported regardless. If the History Save Interval is set to 1, then every function evaluation is recorded.

Remain within Initial Bounds

If checked, the optimization will force candidate designs to be within the ranges specified by the optimization's parameter ranges. This is enforced by line-searching. This can negatively influence the optimization if the best design is located at or outside of the parameter ranges.

8.3. Local Optimizer

The Local Optimizer is a local, gradient-free optimizer that supports several gradient-free algorithms.

Function Call Limit

Specifies the maximum number of steps to take / candidate designs to evaluate in a single optimization run.

Cost Limit

The optimization will end once a candidate design is found with a cost less than or equal to the cost limit.

Cost Convergence Tolerance

The optimization will end when taking a step changes the cost by less than the cost convergence tolerance.

Variable Convergence Tolerance

The optimization will end when taking a step changes the optimized parameters by less than this fraction.

History Save Interval

When the History Save Interval is set to X, the first X function evaluations will be recorded in the history table. After that, only every X'th function evaluation will be reported in the history table. Function evaluations that discover a new best cost, and the last function evaluation of the optimization, will be reported regardless. If the History Save Interval is set to 1, then every function evaluation is recorded.

Mode

Selects which optimization algorithm to use. The algorithms are implemented by the NLopt library.

- COBYLA: Constrained Optimization BY Linear Approximations algorithm, based on the algorithm by M. J. D. Powell.
- BOBYQA: Bound Optimization BY Quadratic Approximation, based on the algorithm by M. J. D. Powell.
- Sbplx: A reimplementation of Tow Rowan's Subplex algorithm.

8.4. Genetic Algorithm

The Genetic Algorithm is an optimization algorithm inspired by evolutionary mutations. This type of optimizer is especially useful for optimizing toggles or binary values.

Each generation of the optimization contains multiple candidate designs, or individuals. A generation is then split into three sections: elites, survivors, and non-survivors. The elite individuals are included in the next generation without any variation. Survivors are included in the next generation, but have a chance of sustaining random mutations. Individuals that do not survive are regenerated based on a combination of "parent" individuals from the generation.

Generation Limit

The optimization will end after this many generations have completed.

Function Call Limit

The optimization will end after this many candidate designs have been evaluated. Because each generation can evaluate multiple designs, this should be higher than the Generation Limit.

Cost Limit

Stop the optimization when the minimum cost has reached this value or below.

Population

Sets the number of function evaluations / individuals to include in each generation.

Elite

This many of the best function evaluations per generation will be selected as the most elite individuals of that generation. These elites will be included in the next generation without any mutations. This can speed convergence when using high mutation rates, but also reduces the rate of adaptation for the best population member.

Bit Mutation Probability

The percentage chance, from 0 to 1, that a bit / toggle will have its value flipped when mutations are applied to an individual.

Survival Fraction

The fraction of each generation whose members will be included in the next generation. Any elites in this fraction will progress to the next generation without mutation, and non-elites will progress with possible mutations. Members of the generation not included in this fraction will be generated from a combination of individuals of the previous generation, their "parents" (see Selection Method).

Selection Method

Determines how parents are selected from the previous generation, to form the basis for generating a new individual in a generation. This selection process is performed for each parent of the new individual. See Survival Fraction.

- Random: A parent is selected at random from the previous generation, with uniform weighting.
- Roulette: A parent is selected randomly from the previous generation, weighted such that higher-cost individuals are more likely to be selected.
- Tournament: A random subset of individuals from the previous generation are selected as potential parents. The best individual from that subset is chosen as a parent. See Tournament Size.

Tournament Size

When tournament selection is used, determines the size of the random subset used for selection. See Selection Method.

Crossover Method

When regenerating an individual based on individuals from the previous generation (their "parents"), this determines how parents are combined to create a new individual. See Survival Fraction.

Crossover Parents

When regenerating an individual based on individuals from the previous generation (their "parents"), this determines how many parents to use when generating the new individual. See Survival Fraction.

Restart Save Interval

The `.restart` file will be written once every this many generations, instead of every generation.

History Save Interval

When the History Save Interval is set to X, the first X generations will be recorded in the history table. After that, only every X’th generation will be reported in the history table. Generations that discover a new best cost, and the last generation of the optimization, will be reported regardless. If the History Save Interval is set to 1, then every generation is recorded.

8.5. Parameter Sweep

A parameter sweep simulates multiple candidate designs, varying the parameters in ways specified by the user. Note that this is not an optimization, but a way to examine multiple design variations at once.

Mode

The sweep mode changes how sweeps behave with multiple parameters added. Outer product simulates all possible combinations of every specified parameter value. Inner product requires that all sweep definitions be the same length.

8.6. Goals

Goals measure the performance of a system, and provide a cost for an optimization or a tolerance analysis. Some goals support multiple input values, and might generate a single cost from multiple values, or produce a cost for each input value, depending on the settings. While the input values may or may not have units, the resulting costs are unitless. Multiple goals can be combined using a Group Goal, whose input values are the costs from each of its subgoals. One of the most useful goals is a Result Goal, which computes its cost from the specified column of a result table.

When processing the input values to compute a list of costs, the following operations are applied:

1. First, the Comparison is applied to each of the input values to generate a cost for each input.
2. Then, the Cost Aggregation Function is applied. Depending on the selected aggregate, this may or may not condense the list of costs into a single cost.
3. Finally, the cost(s) are multiplied by the specified Weight. If a condition specified by the Comparison is satisfied, the cost(s) are also multiplied by the Satisfied Cost Weight.

Comparison

Specifies a desired condition about the goal's input values, resulting in a cost that represents how well the condition is met.

- "<" (less than) - The cost is the difference between the input value and the Target, such that values higher than the target produce a higher cost. If the input value is less than the target value, the comparison is satisfied and the cost is also subject to the Satisfied Cost Weight.
- ">" (greater than) - The cost is the difference between the Target and the input value, such that values lower than the target produce a higher cost. If the input value is greater than the target value, the comparison is satisfied and the cost is also subject to the Satisfied Cost Weight.
- "=" (equal) - The cost is the magnitude of the difference between the input value and the Target, such that values closer to the target produce a lower cost.
- minimize - The cost is the same as the input value, such that higher input values will result in higher costs. This is similar to the "none" comparison.
- maximize - The cost is the negative of the input value, such that lower input values will result in higher costs.
- none - The input value is used as the cost without modification. A Group Goal will normally use this comparison to prevent modification of its subgoal's costs before aggregation.

Target

Specifies a target value to compare against input values when applying the Comparison operation. This property is used with the "<" comparison, ">" comparison, and the "=" comparison. The units for the target value must be compatible with the units of the goal's input values; the target value will also be converted to the input value's units before applying the comparison. For instance, when applying a comparison against spot sizes in a Result Goal, the target must be a length quantity, and the target will be converted to the spot size units (microns) before the comparison. If the spot size is 100um, the comparison is "<" / less than, and the target is 100nm, the final cost will be (100um - 0.1um) = 99.9 (the final cost is unitless).

Cost Aggregation Function

After each of the goal's input values is turned into a cost by the Comparison, the aggregation function is applied. Depending on the selected aggregation, multiple costs may be condensed into a single final cost.

- First - The final cost is the first comparison cost.
- Last - The final cost is the last comparison cost.

- Min - The final cost is the comparison cost with the lowest value.
- Max - The final cost is the comparison cost with the highest value.
- Mean - The final cost is the average of the comparison costs.
- Median - The final cost is the middle comparison cost, when sorted from low to high.
- Sum - The final cost is the sum of the comparison costs.
- RMS - The final cost is the RMS (Root Mean Square) of the comparison costs.
- Norm - The final cost is the magnitude of a vector composed of the comparison costs. This is the same as the RSS (Root Sum Square) of the comparison costs.
- Range - The final cost is the difference between the highest and lowest comparison costs.
- Variance - The final cost is the variance of the comparison costs.
- Count - The final cost is the number of comparison costs / input values.
- None - The comparison costs are not aggregated, and the goal returns a list of all comparison costs, instead of a single final cost.

Weight

After the final cost(s) have been computed by the Cost Aggregation Function, it is multiplied by this weight value. This can be used to prioritize different goals under a Group Goal.

Satisfied Cost Weight

If the condition specified by the "<" comparison or ">" comparison is satisfied, the final cost(s) are multiplied by the satisfied cost weight in addition to the normal Weight. The satisfied cost weight is normally set to zero, so that a satisfied goal can have a cost of zero instead of a negative value. This is treated separately from the normal weight so that any gradients computed via finite-differencing are not also zeroed by the satisfied comparison, but only the cost itself. Allowing non-zero gradients even if the goal has been satisfied provides gradient-based optimizations with additional information about the simulation, resulting in better optimization performance.

8.6.1. Result Goal

A Result Goal is a goal whose input values are taken from a specified column of an optionally-filtered result or data table. Each row of the specified column is taken as an input value for computing the cost(s). For additional information and properties, see Goals.

Data Table Selection

Selects the table used for computing the cost. Once a table has been selected, the rows of that table may optionally be filtered based on values of different columns (for instance, only computing costs for rows with a "Wavelength" column matching a single value). The filtering options provided here are the same as those provided with a Data Table View, which can be opened by right-clicking on a table in the Model Hierarchy and selecting "Show Table."

Data

Selects the column of the filtered table to use as the input values for computing the goal's cost.

Operation

Before using the table values to compute the goal's cost, each row of the filtered table has one of the following operations applied:

- (empty/none) - By default, no operation is applied, and the table column's value is used as-is.
- mag - The magnitude of the value is used instead.
- real - The real value of the table column's value is used. For non-complex table columns, this is the same as the empty operation.
- imag - The imaginary value of the table column's value is used. For non-complex table columns, this results in a value of zero.

8.6.2. Group Goal

Group Goals can contain multiple subgoals and combine their costs. Each cost of each of the subgoals will be taken as an input value when computing the cost(s) of the group goal. For additional information and properties, see Goals.

9. Tolerance Analysis

Tolerance Analysis objects allow tolerancing of a system by perturbing/deviating parameter values based on a set of tolerance ranges, and measuring the change in performance as a result of those changes. Additional "compensating" parameters can be optimized to reduce the change in performance, and bring each perturbed system's performance closer to the nominal performance. These perturbations and compensations simulate the variations in a lens design as a result of manufacturing, and the adjustments made to that system to counteract the manufacturing variations. The exact values of each perturbed system and the computed summary statistics depend on the type of analysis selected.

The performance metric or cost measured during a tolerance analysis is defined using Goals. Only a single cost value may be toleranced, so goals referring to results with multiple rows should filter the table or use an appropriate Cost Aggregation Function to combine the rows into a single metric. Tolerance ranges and other perturbations to be applied to the simulation are defined using Deviated Tolerance Parameters. And values to be optimized during compensation of a perturbed simulation are defined using Compensated Tolerance Parameters.

When running a tolerance analysis, a status message is displayed after computing each perturbed simulation. The status messages contain how long the simulation took to run (including any compensation optimizations), the simulation cost/performance, and the change in that performance from the nominal simulation. If Compensate Performance Changes is turned on, then the number of iterations taken by the compensation optimization is also displayed.

Properties marked with an asterisk (*) are advanced properties made visible through the "Toggle Advanced Properties Visibility" button indicated by the pair of gears (⚙) in the property editor.

Random Seed*

If set to zero, the sampling of tolerances in a Monte Carlo analysis will be random every time the analysis is run. If the seed is non-zero, the sampling will be performed with the given seed, resulting in identical perturbations every time the analysis is run (assuming the simulation and tolerances have not changed between runs). This can be useful for consistently reproducing specific perturbations when troubleshooting.

Analysis Type

Sets the type of tolerance analysis to run:

- Direct Sensitivity - Performs a sensitivity analysis.
- Monte Carlo - Performs a Monte Carlo analysis.
- Sensitivity + Monte Carlo - Performs both a sensitivity analysis and a Monte Carlo analysis.

Number of Simulations

Sets the number of perturbed simulations to generate during a Monte Carlo analysis.

Cost Direction

Determines whether the performance metric defined by the Goals should be maximized or minimized. Primarily used when computing the Monte Carlo Cumulative Probability Table and Estimated Cost.

- Minimize - A lower performance metric cost represents a better system. The cumulative probability result will sort the lower-cost permutations first.
- Maximize - A higher performance metric cost represents a better system. The cumulative probability result will sort the higher-cost permutations first.

Compensate Performance Changes

If checked, each perturbed simulation generated during the analysis will optimize the parameter values specified by any enabled Compensating Parameter objects, in order to minimize the change in performance from the nominal system. Note that this optimization will minimize the change in performance, and not the performance cost itself. Compensators are optimized using the same gradient-free local optimization algorithms as a Local Optimizer.

Compensation Algorithm*

Sets the specific optimization algorithm used for compensation. See the Local Optimizer's Mode for more information. See also Compensate Performance Changes.

Function Call Limit*

Specifies the maximum number of candidate designs to evaluate when compensating each perturbed system. Increasing this limit may increase the amount of time spent optimizing compensated parameters. This is similar to the Local Optimizer’s Function Call Limit. See also Compensate Performance Changes.

Cost Change Limit*

The compensation optimization for a perturbed system will end if the magnitude of the change in the cost/performance metric from the nominal system is less than or equal to the cost change limit. Decreasing this limit may increase the amount of time spent optimizing compensated parameters. This is similar to the Local Optimizer’s Cost Limit. See also Compensate Performance Changes.

Cost Convergence Tolerance*

The compensation optimization will end when an iteration changes the performance metric by less than the cost convergence tolerance. Decreasing this tolerance may increase the amount of time spent optimizing compensated parameters. This is similar to the Local Optimizer’s Cost Convergence Tolerance. See also Compensate Performance Changes.

Parameter Convergence Tolerance*

The compensation optimization will end when an iteration changes the value of the compensated parameters by less than this fraction. Decreasing this tolerance may increase the amount of time spent optimizing compensated parameters. This is similar to the Local Optimizer’s Variable Convergence Tolerance. See also Compensate Performance Changes.

9.1. Direct Sensitivity Tolerance Analysis

A sensitivity analysis computes the change in the performance metric at the extremes of the tolerance ranges set by any enabled Deviated Tolerance Parameters. The performance of the nominal simulation is computed first, followed by perturbing the simulation with the max negative and positive tolerance deviations for each parameter individually. This can help identify the most sensitive tolerances in a simulation, but ignores cross-effects of the perturbations unlike a Monte Carlo analysis.

The number of perturbed simulations generated is $2n$, where n is the number of tolerance parameters. The total number of simulations is $1 + 2n$, accounting for the nominal, unmodified simulation. A history table contains the results of running each simulation, and a summary table contains summary statistics such as RSS cost change and estimated manufactured performance.

9.1.1. Sensitivity History Table

Each row of the history table describes a single simulation. The first row of the table is the unperturbed, nominal system, and each remaining row is a perturbed simulation. If Compensate Performance Changes is turned on, then the rows for the perturbed simulations contain the data for the compensated simulation, which has been optimized to minimize the change in performance. The history table contains the following columns:

Cost

The cost or performance metric computed by the analysis Goals for each simulation.

Cost Change

The change in cost from the nominal simulation (the first row of the table) for each simulation.

For each of the Deviated Tolerance Parameters, the history table contains these additional columns, where *Tolerance* is the name of the tolerance deviation object, and *Parameter* is the name of the parameter specified in its Parameter field:

Change *Tolerance*

The perturbation applied to the *Tolerance* tolerance parameter in each simulation.

Value *Tolerance*

The nominal *Tolerance* tolerance value, plus the perturbation applied to it, in each simulation.

Parameter *Parameter*

The value of the *Parameter* parameter object that corresponds to the tolerance's perturbation. For Simple Deviation tolerances, this value is the same as the "Value *Tolerance*" column. However, specific plugins may have additional tolerance deviation types where the perturbation values and parameter values are different. For instance, reTORT's Radius Tolerance Deviation specifies a tolerance in terms of radius-of-curvature, but uses parameters that define a surface's curvature, so the "Value *Tolerance*" column will show the perturbed radius-of-curvature and the "Parameter *Parameter*" column is the corresponding perturbed curvature value. See the plugin's tolerancing documentation for more details.

For each of the Compensated Tolerance Parameters that was optimized during a compensation, the history table contains these additional columns, where *Compensator* is the name of the tolerance compensator object:

Change *Compensator*

The change in the optimized compensator value from the nominal simulation (first row of the table) for each simulation.

Value *Compensator*

The value of the optimized compensator value in each simulation. Compensators do not need to be converted like some tolerances, so this is also the compensated value of the parameter specified by the Parameter field.

9.1.2. Sensitivity Summary Table

The summary table is computed from the perturbations in the Sensitivity History Table. The cost changes from all negative and positive perturbations are used when computing the results, even if the perturbations are for the same tolerance deviation object. The summary table contains the following data:

Nominal Cost

The cost or performance metric computed for the unperturbed, nominal simulation.

Estimated Cost

An estimation of the as-built performance of the simulation based on the mean and standard deviation performance changes, defined as

$$f_{\text{estimated}} = \begin{cases} f_0 + \mu + 2\sigma, & \text{if cost direction is Minimize} \\ f_0 + \mu - 2\sigma, & \text{if cost direction is Maximize} \end{cases}$$

where f_0 is the Nominal Cost, μ is the Mean Cost Change, and σ is the Std Deviation Cost Change. Whether the estimate is high or low depends on the Cost Direction.

RSS Estimated Cost

An estimation of the as-built performance of the simulation based on the RSS (Root Sum Square) performance change, defined as

$$f_{\text{RSS estimated}} = \begin{cases} f_0 + \sigma_{\text{RSS}}, & \text{if cost direction is Minimize} \\ f_0 - \sigma_{\text{RSS}}, & \text{if cost direction is Maximize} \end{cases}$$

where f_0 is the Nominal Cost, and σ_{RSS} is the RSS Cost Change. Whether the estimate is high or low depends on the Cost Direction.

RSS Cost Change

The RSS (Root Sum Square) of the changes in performance from the nominal simulation for all perturbed simulations, defined as

$$\sigma_{\text{RSS}} = \sqrt{\sum_{i=1}^{N} (\Delta f_i)^2}$$

where N is the number of perturbed simulations, and Δf_i is the Cost Change for the i’th perturbation.

Mean Cost Change

The sample mean of the cost/performance changes from the nominal simulation across all perturbed simulations, defined as

$$\mu = \frac{1}{N-1} \sum_{i=1}^{N} (f_i - f_0) = \frac{1}{N-1} \sum_{i=1}^{N} \Delta f_i$$

where N is the number of perturbed simulations, f_i is the Cost for the i’th perturbation, f_0 is the Nominal Cost, and Δf_i is the Cost Change for the i’th perturbation. Dividing the sum by $N-1$ instead of N is due to applying Bessel’s correction of $\frac{N}{N-1}$.

Variance Cost Change

The sample variance of the cost/performance changes from the nominal simulation across all perturbed simulations, defined as

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} (\Delta f_i - \mu)^2$$

where N is the number of perturbed simulations, Δf_i is the Cost Change for the i'th perturbation, and μ is the Mean Cost Change. Dividing the sum by $N-1$ instead of N is due to applying Bessel's correction of $\frac{N}{N-1}$.

Std Deviation Cost Change

The sample standard deviation of the cost/performance changes from the nominal simulation across all perturbed simulations. This is the square root of the variance, and is defined as

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\Delta f_i - \mu)^2}$$

where N is the number of perturbed simulations, Δf_i is the Cost Change for the i'th perturbation, and μ is the Mean Cost Change. Dividing the sum by $N-1$ instead of N is due to applying Bessel's correction of $\frac{N}{N-1}$.

Avg Deviation Cost Change

The mean absolute deviation of the cost/performance changes from the nominal simulation across all perturbed simulations, defined as

$$\sigma_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N}|\Delta f_i - \mu|$$

where N is the number of perturbed simulations, Δf_i is the Cost Change for the i'th perturbation, and μ is the Mean Cost Change.

Skewness Cost Change

The skewness of the cost/performance changes from the nominal simulation across all perturbed simulations, defined as

$$\tilde{\mu}_3 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\Delta f_i - \mu}{\sigma}\right)^3$$

where N is the number of perturbed simulations, Δf_i is the Cost Change for the i'th perturbation, μ is the Mean Cost Change, and σ is the Std Deviation Cost Change.

Kurtosis Cost Change

The excess kurtosis of the cost/performance changes from the nominal simulation across all perturbed simulations, defined as

$$\kappa = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\Delta f_i - \mu}{\sigma}\right)^4 - 3$$

where N is the number of perturbed simulations, Δf_i is the Cost Change for the i'th perturbation, μ is the Mean Cost Change, and σ is the Std Deviation Cost Change.

For each of the Compensated Tolerance Parameters that was optimized during a compensation, the summary table also contains these additional columns, where *Compensator* is the name of the tolerance compensator object:

Mean Change *Compensator*

The sample mean of the change in the compensated value from the nominal simulation across all perturbed simulations, defined as

$$\mu_y = \frac{1}{N-1}\sum_{i=1}^{N} \Delta y_i$$

where N is the number of perturbed simulations, and Δy_i is the Compensator Change for compensator y in the i'th perturbation. Dividing the sum by $N-1$ instead of N is due to applying Bessel's correction of $\frac{N}{N-1}$.

Std Deviation Change *Compensator*

The sample standard deviation of the change in the compensated value from the nominal simulation across all perturbed simulations, defined as

$$\sigma_y = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} \left(\Delta y_i - \mu_y\right)^2}$$

where N is the number of perturbed simulations, Δy_i is the Compensator Change for compensator y in the i'th perturbation, and μ_y is the Compensator Mean Change for compensator y. Dividing the sum by $N-1$ instead of N is due to applying Bessel's correction of $\frac{N}{N-1}$.

Estimated Change *Compensator*

The estimated amount of change in the compensated value for an as-built simulation. This is estimated as twice the Compensator Std Deviation Change.

9.2. Monte Carlo Tolerance Analysis

Whereas a sensitivity analysis perturbs one tolerance parameter at a time, a Monte Carlo analysis will randomly perturb every tolerance parameter at once between the ranges of their Negative Deviation and Positive Deviation. This makes the tolerance analysis more realistic by accounting for cross-effects from multiple parameters. However, it can be more difficult to identify the most sensitive parameters compared to a sensitivity analysis.

The likelihood of the random values within the deviation ranges is controlled by the Distribution settings. The number of perturbed simulations generated is controlled by the Number of Simulations property. A history table contains the results of running each simulation, and a summary table contains statistics such as the standard deviation cost change, and estimated manufactured performance.

9.2.1. Monte Carlo History Table

The Monte Carlo analysis history table contains the same columns as a sensitivity analysis history table. See the Sensitivity History Table for more information.

9.2.2. Monte Carlo Summary Table

The Monte Carlo analysis summary table contains the same columns as a sensitivity analysis summary table. See the Sensitivity Summary Table for more information.

9.2.3. Cumulative Probability Table

The cumulative probability table provides probability information about the analysis. Each row is a perturbed simulation, where the simulations have been sorted by their cost / performance metric according to the Cost Direction. This information can also be plotted to display a cumulative probability curve. This table contains the following columns:

History Row

The row number of the corresponding perturbed simulation in the Monte Carlo History Table.

Cost

The performance of the perturbed simulation.

Cost Change

The change in performance of the perturbed simulation compared to the nominal cost.

Cumulative Probability

The likelihood of any given perturbation in the Monte Carlo analysis having the same or better performance than this perturbation.

9.3. Deviated Tolerance Parameters

Deviated tolerance parameters define the types of perturbations that can be performed during a tolerance analysis. They specify a Parameter object whose value will be perturbed, and a set of tolerance ranges for the value.

For simple deviations, the tolerance ranges and the parameter being perturbed represent the same kind of value, and any deviations are applied directly to the parameter value. For instance, a thickness tolerance could be represented with a Simple Deviation object using a range of +/-1mm. When the toleranced value is deviated by +1mm, the parameter object containing the thickness value will also be perturbed by +1mm.

However, specific plugins may provide additional tolerance deviation types where the tolerance values and the parameter values are different. In that case, any deviations of the tolerance value will be converted into a corresponding perturbed parameter value. For instance, reTORT’s Radius Tolerance Deviation specifies a tolerance in terms of radius-of-curvature, but refers to parameters that define a surface’s curvature, so any generated deviations in the radius-of-curvature must be converted into the equivalent curvature perturbation before being applied to the simulation. See the plugin’s tolerancing documentation for more details.

Enabled

If unchecked, this tolerance will be ignored during a tolerance analysis run.

Parameter

Selects the Parameter object whose value will be perturbed by this tolerance.

Type

Selects the tolerance deviation type. For simple deviations, the tolerance ranges and the parameter being perturbed represent the same kind of value, and any deviations are applied directly to the parameter value. However, specific plugins may provide additional tolerance deviation types where the tolerance values and the parameter values are different. In that case, any deviations of the tolerance value will be converted into a corresponding perturbed parameter value.

- Simple Deviation - The tolerance range represents changes to the parameter value, and any deviations are applied directly to that value.
- Radius Deviation - The tolerance range is specified in terms of radius-of-curvature. For more information, see reTORT's Radius Tolerance Deviation.
- Power Fringe Deviation - The tolerance range is specified in terms of a fringe-count. For more information, see reTORT's Power-Fringe Tolerance Deviation.

Negative Deviation

Defines the bottom of the tolerance range, or the largest negative deviation that may be applied to this tolerance parameter. This property is a lower limit on the *change* in the parameter value, and not the lower limit of the value itself. The units of this value must be compatible with the type of tolerance it represents. For a Simple Deviation, this means the deviation bounds must have units compatible with the value of the selected Parameter.

Positive Deviation

Defines the top of the tolerance range, or the largest positive deviation that may be applied to this tolerance parameter. This property is an upper limit on the *change* in the parameter value, and not the upper limit of the value itself. The units of this value must be compatible with the type of tolerance it represents. For a Simple Deviation, this means the deviation bounds must have units compatible with the value of the selected Parameter.

Distribution

Selects the type of probability distribution used for this tolerance parameter during a Monte Carlo analysis.

- Uniform - The probability function is evenly distributed between the Negative Deviation and Positive Deviation values, such that any deviation value is equally likely.
- Truncated Gaussian - The probability function is a normal or Gaussian distribution, truncated between the Negative Deviation and Positive Deviation values. The precise shape of the Gaussian distribution is controlled by the Gaussian Settings Type.

- End-Point - Only the most extreme deviation values are possible; either the Negative Deviation or Positive Deviation will be selected at random and are equally likely.

Gaussian Settings Type

Determines how the shape of a Truncated Gaussian distribution is defined.

- Std Deviation Factor - The peak/mean of the Gaussian distribution is centered evenly between the Negative Deviation and the Positive Deviation. The standard deviation is set such that there are a specific number of multiples of the standard deviation between the mean and either end of the truncated distribution, with the number of multiples controlled by the Gaussian Std Deviation Factor.
- Mean and Std Deviation - The peak/mean and standard deviation of the Gaussian distribution are set directly via the Gaussian Mean and Gaussian Std Deviation properties.

Gaussian Std Deviation Factor

Used by the Std Deviation Factor Gaussian settings type to compute the standard deviation for a Gaussian distribution centered evenly between the Negative Deviation and Positive Deviation values. For a factor of N, the standard deviation (σ) is computed such that there are N multiples of the standard deviation between the mean (μ) and either end of the truncated distribution ($\Delta_{\text{max neg}}$ and $\Delta_{\text{max pos}}$, set by the Negative Deviation and Positive Deviation properties). This value does not need to be a whole number.

$$\mu = \frac{\Delta_{\text{max neg}} + \Delta_{\text{max pos}}}{2}$$

$$\sigma = \frac{\Delta_{\text{max pos}} - \Delta_{\text{max neg}}}{2N} = \frac{\Delta_{\text{max pos}} - \mu}{N}$$

Gaussian Mean

Used by the Mean and Std Deviation Gaussian settings type to define the shape of the truncated normal distribution. This value is the location of the peak/mean of the Gaussian curve, and need not necessarily lie within the bounds of the distribution itself. Must have units compatible with the Negative Deviation and Positive Deviation properties.

Gaussian Std Deviation

Used by the Mean and Std Deviation Gaussian settings type to define the shape of the truncated normal distribution. This value is the standard deviation of the Gaussian curve, and must be non-zero. In addition, while negative values are allowed, they will be treated as though they are positive. Must have units compatible with the Negative Deviation and Positive Deviation properties.

9.4. Compensated Tolerance Parameters ⚖

Compensated tolerance parameters allow adjustments to be made to a simulation to counteract any performance changes introduced by perturbations from the Deviated Tolerance Parameters. These compensations are defined by a Parameter object that will be compensated, and the range of min and max changes allowed in that parameter's value. During a tolerance analysis, the parameter values will be optimized within the specified bounds to minimize any changes in performance (not the performance itself). These compensator objects will be ignored unless Compensate Performance Changes is turned on; see that section for additional compensation settings.

Enabled

If unchecked, this compensator will be ignored and unchanged by any compensation optimizations during a tolerance analysis.

Parameter

Selects the Parameter object whose value will be optimized while compensating a perturbed system.

Max Negative Compensation

Defines the largest negative change that may be applied to this compensating parameter. Note that this property is a lower limit of the *change* in the compensator value, and not the lower limit of the value itself. The units of this value must be compatible with the value of the selected Parameter.

Max Positive Compensation

Defines the largest positive change that may be applied to this compensating parameter. Note that this property is an upper limit of the *change* in the compensator value, and not the upper limit of the value itself. The units of this value must be compatible with the value of the selected Parameter.

Topical Documentation for reTORT

1. reTORT Simulation

A reTORT simulation allows the specification of the geometry, material, and incident light of an optical system, and computes the path of light as it propagates through the system.

1.1. Objects and Collections

- Parameters
- Actions
- Properties
- Spatial Domain
- Sources
- Boundaries
- Elements
- Deviated Tolerance Parameters for reTORT
- Results
- Views

1.2. Wizards

- Optimization Wizard
- Tolerancing Wizard
- Spot Diagram Wizard
- Wavefront Error Plot Wizard
- MTF Graph Wizard
- TRA Trace Wizard
- Save Wavefront Profile Wizard
- Clear Wavefront Profiles Wizard

2. Objects and Properties

2.1. Spatial Domain

The spatial domain includes simulation-specific settings and solver configuration. Only one spatial domain is permitted and is included as part of the default template. To create a spatial domain, right click the Spatial Domain folder and select "New Spatial Domain".

2.1.1. Spatial Domain Properties

Name

See GEMSIF Object Properties - Name

Bounding Box Min

Specifies the minimum corner of the bounding box. The value of each corner property is a 3-vector of dimensioned lengths. Example: [xmin,ymin,zmin]. See the position vector documentation for explanations of the different ways the position can be specified.

Bounding Box Max

Specifies the maximum corner of the bounding box. The value of each corner property is a 3-vector of dimensioned lengths. Example: [xmax,ymax,zmax]. See the position vector documentation for explanations of the different ways the position can be specified.

- See Bounding Box Min

Background Material

The background material property configures the material index and loss for rays propagating through the space between lenses.

Max Iterations

Restrict the number of child rays that may result from a single input ray.

Ray Magnitude Threshold

Determines the magnitude threshold used to ignore reflected rays.

Parallelize Solver

The Parallelize Solver toggle enables GEMSIF to utilize multi-core, parallel solving of the simulation.

Maximum Solver Threads

The Maximum Solver Threads setting establishes a limit to the number of threads that the solver can use simultaneously. 0 indicates an automatic selection of a number of threads.

Gradient Solver Tolerance

Determines the accuracy vs. simulation speed trade-off of the numerical solver for gradient-index regions. The default setting is accurate for most purposes, and offers reasonable simulation speed.

Gradient Solver Method

Specifies the solver equation to be used when calculating a GRIN simulation.

Gradient Growth Scaling Factor

The Gradient Growth Scaling factor modifies the growth of the stepsize used in Runge-Kutta when solving the propagation of a ray through a GRIN.

Gradient Decay Scaling Factor

The Gradient Decay Scaling factor modifies the decay of the stepsize used in Runge-Kutta when solving the propagation of a ray through a GRIN.

Gradient Solver Initial Step-Size

The Gradient Solver Initial Step-Size sets an initial step length for Runge-Kutta when solving the propagation of a ray through a GRIN.

Gradient Solver Initial Clamp

The Gradient Solver Initial Clamp value sets the minimum value for the relative numerical error on a step.

GRIN Version

The GRIN Version setting is for internal testing purposes and should not be modified.

Sequential Raytrace

When sequential raytracing is enabled, only Lens Stacks will be traced, and the order in which rays intersect their surfaces will be forced to be the order of the surfaces in the lens stack. Multiple lens stacks can be ordered by their Trace Order property. Rays will end prematurely if they fail to trace through the expected surfaces.

2.2. Sources

Sources represent the incident light into the optical system. Sources can consist of collimated, plane-wave beams, gaussian beams, or arbitrary collections of user-specified rays. All sources may be configured for one or more excitation wavelengths.

2.2.1. Source Types

- Beam Source
- Ray Bundle Source

2.2.2. Beam Source

A beam has a circular profile, with either uniform (top-hat) or gaussian magnitude distributions. The beam source allows the starting location and list of incident angles to be specified.

Beam Source Properties

Name

See GEMSIF Object Properties - Name

Spectral Lines

See Spectrum Characteristics - Spectral Lines

Irradiance Spectrum

See Spectrum Characteristics - Irradiance Spectrum

Origin

See Object Properties - Origin

Optical Axis

See Source Properties - Optical Axis

Incident Elevation Angles

See Source Properties - Incident Elevation Angles

Incident Azimuthal Angle

See Source Properties - Incident Azimuthal Angle

Set Diameter based on Pupil

This toggle sets the Beam Source’s Beam Diameter to match the calculated diameter of the entrance pupil of the system.

Aim Beam at Entrance Pupil

This toggle causes the origin point for off-axis bundles to be set so that the bundle is aimed at the entrance pupil of the system.

Beam Diameter

Diameter of the beam waist.

Focal Length

See Source Properties - Focal Length

Magnitude Distribution

See Source Properties - Magnitude Distribution

Ray Density

See Source Properties - Ray Density

Ray Color Mode

See Source Properties - Ray Color Mode

Color

See Source Properties - Color.

Ray Opacity Mode

See Source Properties - Ray Opacity Mode.

Ray Opacity

See Source Properties - Ray Opacity.

2.2.3. Ray Bundle Source

A ray bundle consists of a group of manually-specified rays. This allows the user to create a custom incident distribution, with custom-specified magnitude, phase, location, and direction for each ray.

Ray Bundle Source Properties

Name

See GEMSIF Object Properties - Name

Spectral Lines

See Spectrum Characteristics - Spectral Lines

Irradiance Spectrum

See Spectrum Characteristics - Irradiance Spectrum

Origin

See Object Properties - Origin

Optical Axis

See Source Properties - Optical Axis

Ray Color Mode

See Source Properties - Ray Color Mode

Color

See Source Properties - Color.

Ray Opacity Mode

See Source Properties - Ray Opacity Mode.

Ray Opacity

See Source Properties - Ray Opacity.

Rays

Collection of properties for a particular ray in a Ray Bundle.

Name

See GEMSIF Object Properties - Name.

Origin

See Object Properties - Origin.

Direction

Direction vector of the ray as a three-dimensional unit vector. See the orientation vector documentation for explanations of the different ways this can be specified.

Relative Magnitude

Specifies the amplitude of the ray relative to the other rays. No particular units are attached to this value.

Phase Offset

Controls the phase offset of the ray relative to the other rays in the aperture.

Polarization

The polarization of a ray is set by a vector that is perpendicular to the direction of propagation, and represents the direction of the electric field vector. The polarization is determined jointly by the E-field orientation vector and the aspect ratio, which determines whether the ray represents a linearly, circularly, or elliptically-polarized ray.

The polarization will affect the magnitude of the output ray distribution if using "Full" analysis mode, since the polarization affects the reflection magnitude of a wave from a surface.

Any value may be provided to this property, and it will be corrected to form a valid polarization vector at runtime.

See the orientation vector documentation for explanations of the different ways this vector can be specified.

Polarization Aspect Ratio

The polarization sense is determined by a double value set as the aspect ratio. A value of zero (0) represents a linearly-polarized incident field, while +1 and -1 represent right-hand and left-hand circular polarization, respectively. All intermediate values indicate elliptical polarization.

Internally, conversions between aspect ratios are performed such that the largest field component is aligned with the pol vector, which constrains the aspect ratio within the range [-1,1]. Aspect ratio represents the difference in field magnitude along the pol vector and along a vector perpendicular to the pol vector.

In isotropic and lossless media, the polarization vector is always normal to the direction of propagation. In anisotropic or lossy media, however, this is not guaranteed. Functionality for ray propagation into anisotropic or lossy media is not presently supported, although such functionality would be desirable.

Ray Lists

Similar to a Ray object, but each property is a list instead of a single value. This allows a RayList to describe multiple rays at once.

All lists in a RayList object should be the same length. However, if all rays in the RayList will be given the same value for a property, then the list for that property may contain only that value. For instance, setting the Origins property of the RayList to **[[0mm,0mm,0mm], [1mm,0mm,0mm]]** and leaving all other properties at their default, would generate two nearly-identical rays that differ only by their starting position. The default "Relative Magnitudes" value of **[1]** would be applied to both rays, without the need to set Relative Magnitudes to **[1, 1]**.

For more information about the RayList properties, see the Ray properties.

2.2.4. Spectrum Characteristics

The evaluation spectrum of the source can be controlled by setting the characteristics of the spectral lines and irradiance spectra. Spectral lines determine the actual incident wavelengths used to excite the system, while the irradiance spectra determine the relative magnitude of the sources at different wavelengths.

Built-in spectra will include standard optical, infrared, and UV evaluation wavelengths, as well as the magnitude spectra of common sources.

Spectral Lines

The collection of spectral lines within a source specify the excitation wavelengths for a model.

Discrete Spectrum

Specify the discrete evaluation wavelengths for the source. Multiple spectra can be added to a single source; all unique wavelengths requested will be modeled.

Discrete Spectrum Properties

- Name - See GEMSIF Object Properties.
- Wavelengths - See Spectrum Properties.

Irradiance Spectra

Black Body Irradiance Spectrum

Allows the user to specify an irradiance spectrum along a temperature-dependent black body curve.

Tabular Irradiance Spectrum

Allows the user to control the magnitude of a source’s spectrum.

Tabular Irradiance Spectra Properties

- Name - See GEMSIF Object Properties.
- Wavelengths - See Spectrum Properties.
- Magnitude - Sets the amplitude of a source’s wave.

Spectrum Properties

Wavelength

Defines the wavelengths in length units (m, nm, etc).

2.2.5. Source Properties

Focal Length

Specifies the distance to the object plane relative to the origin of the beam coordinate system. Can be positive or negative.

Incident Azimuthal Angle

Azimuthal beam angle of the incident beam. Only a single value can be specified.

Incident Elevation Angles

Elevation angle of the beam relative to the boresight direction (orientation) of the incident beam. An array of angle quantities (deg or rad).

Magnitude Distribution

Determines the distribution of rays within the beam. For most systems, a Gaussian Quadrature distribution is most efficient. However, some results (such as spot size) will not be computed accurately for a Gaussian Quadrature distribution if not all rays hit the image plane, or if multiple child rays from the same initial ray hit the image plane. This can happen if there are ABS Boundaries that absorb some rays, or None Boundaries that cause partial reflection and refraction. In such cases, a different beam distribution should be used.

Optical Axis

The optical axis of the source determines the angle of the chief ray and the orientation of the input aperture. See the orientation vector documentation for explanations of the different ways the orientation can be specified.

Ray Density

Configure the number of rays used to model the beam. The number of rays increases with the square of the property value. A setting of 1 traces only chief and marginal rays, and a density of 0 only traces the chief ray.

Ray Color Mode

Allows the user to specify the color of the ray groups as shown in the 3D model.

Color

Sets the base color for the source object and rays in the 3D model view.

Ray Opacity Mode

Sets the mode for determining the opacity of individual rays. Options include Opaque, Manually setting a value between 0 and 1, and Magnitude based where the highest magnitude rays are opaque and other rays are transparent based on their magnitude proportional to the highest magnitude rays.

Ray Opacity

Sets the opacity value for rays generated by this source when the Manual Option is selected for the Ray Opacity Mode.

2.3. Boundaries

The Boundaries defined below allow the user to specify behavior at the surfaces of lenses. This can take the form of idealized coatings as well as simulated meta-surfaces.

2.3.1. Ideal Boundaries

These are all available as pre-made Boundary entries.

- ABS - ABS stands for an absorbing surface that does not reflect or refract rays
- AR - AR stands for an anti-reflective surface that does not reflect rays, but refracts them normally
- None - None stands for a standard surface that will scatter both reflected and refracted rays normally

- PEC - PEC stands for a Perfect Electric Conductor or reflective surface that will not refract rays but reflect them normally.

2.3.2. Inhomogeneous Boundaries

Inhomogeneous boundaries allow the user to manually define a phase contributing surface using an arbitrary definition. This definition can be based in polynomial equations, bezier control points, or table definitions. For this to function as a phase-gradient metasurface, it is required that you also assign the corresponding gradient field. Without this field, the boundary will not be capable of steering a ray bundle.

Types:

- Refractive - Refracts both reflected and refracted rays.
- AntiReflective - Only transmits refracted rays.
- Reflective - Only transmits reflected rays.

Polarization Dependence: This determines the polarization dependent behavior of the boundary. The options include None, SP, XY, and RPhi. When an option other than None is selected, the surface will follow different definitions for each of the two incident ray components listed in the type.

2.3.3. Ideal Metasurfaces Library

This library contains common inhomogeneous metasurfaces.

Spherical Focus Metasurface

This is the spherical focus metasurface library object. The spherical focus metasurface phase response is defined using the following formulae:

$$\Phi = \frac{-2\pi}{\lambda}\left(\sqrt{r^2+f^2}-f\right)$$

$$\nabla\Phi = \frac{-2\pi}{\lambda}\left(\frac{r}{\sqrt{r^2+f^2}}\right)\hat{\mathbf{r}}$$

Where r is the radial coordinate, lambda is the free-space wavelength, and f is the desired focal length.

- Wavelength: The lambda value defined above.
- Focal Length: The f value defined above.

Radial Metasurface

This is the radial metasurface library object. The radial metasurface is defined using a polynomial with respect to the radial coordinate. This radial coordinate is the arc length from the point on the surface to the apex of that surface. If this coordinate is normalized, then the value will be scaled from 0 to 1 where 0 is the center of the surface and 1 is the maximum arc-length along that surface (or the arc-length at the edge).

This pre-made radial metasurface contains the following properties:

- Units Choice: This allows the specification of units the radial coordinate.
- Constant Term (C0): This is the constant contribution to the phase polynomial
- Quadratic Term (C1): This is the quadratic (second order) contribution from the radial coordinate to the polynomial.
- Quartic Term (C2): This is the quartic (fourth order) contribution from the radial coordinate to the polynomial.
- Sextic Term (C3): This is the sextic (sixth order) contribution from the radial coordinate to the polynomial.

Elliptic Metasurface

This is the elliptic metasurface library object. The elliptic metasurface is defined using a polynomial with respect to the x and y Cartesian coordinates. These Cartesian coordinates are the cosine and sine of arc length from the point on the surface to the apex of that surface, then projected into a flat plane. If these coordinates are normalized, then the value will be scaled from -1 to 1 where 0 is the center of the surface.

This pre-made elliptic metasurface contains the following properties:

- Units Choice: This allows the specification of units the x and y coordinates.
- Constant Term (C0): This is the constant contribution to the phase polynomial
- Quadratic X Term (C1): This is the quadratic (second order) contribution from the x coordinate to the polynomial.
- Quartic X Term (C2): This is the quartic (fourth order) contribution from the x coordinate to the polynomial.
- Sextic X Term (C3): This is the sextic (sixth order) contribution from the x coordinate to the polynomial.
- Quadratic Y Term (C4): This is the quadratic (second order) contribution from the y coordinate to the polynomial.
- Quartic Y Term (C5): This is the quartic (fourth order) contribution from the y coordinate to the polynomial.
- Sextic Y Term (C6): This is the sextic (sixth order) contribution from the y coordinate to the polynomial.

2.3.4. Angle Dependent Metasurfaces Library

This library contains common inhomogeneous metasurfaces.

Theta Dependent Radial Metasurface

This is the theta dependent metasurface library object. The theta dependent radial metasurface is functionally the Radial Metasurface but with terms that can be polynomially scaled with incident angle.

The theta dependent radial metasurface is defined using the following terms:

- Units Choice: This allows the specification of units the r coordinates.
- Constant Term (C0): This is the constant contribution to the phase polynomial
- Quadratic r Term (C1): This is the quadratic (second order) contribution from the radial coordinate to the polynomial.
- Quartic r Term (C2): This is the quartic (fourth order) contribution from the radial coordinate to the polynomial.
- Sextic r Term (C3): This is the sextic (sixth order) contribution from the radial coordinate to the polynomial.
- Linear r Theta Term (C4): This is the linear (first order) contribution from the incident angle scaling linearly with the radial coordinate.
- Linear r Quadratic Theta Term (C5): This is the quadratic (second order) contribution from the incident angle scaling linearly with the radial coordinate.

2.4. Elements

Lenses, prisms, mirrors, and all other geometry are modeled as Elements objects. Elements can be specified in several ways, such as by specifying an explicit list of bounding surfaces, importing a pre-configured optic from a library, or requesting a Paraxial lens or other ideal or nonphysical element. Apertures and stops are also specified as elements. Users familiar with spreadsheet-style surface entry will find the Lens Stack elements to be the most intuitive form of geometry entry; this is the recommended method for modeling lens geometry.

2.4.1. Lens Stacks

Lens stacks are a form of sequential surface entry, where surfaces and the materials between them are listed in order in a table. Any sections with a non-background material (see Spatial Domain Background Material) will be enclosed to form a lens, with the surfaces on either side forming the front and back surfaces of the lens.

Lens Stack Editor

The Surfaces table in the lens stack property editor displays the basic properties of each surface, with a row for each surface in the stack. To add a new surface to the lens stack, click the "Add" button below the Surfaces table. New surfaces will be added to the end of the lens stack, but before any Image Planes. To remove or reposition an existing surface, click on the row corresponding to the surface and click "Delete", "Move Up", or "Move Down." To view the complete list of properties for a surface, click on the row for the surface and click the "Edit" button.

If a surface's property is set to a parameter, editing that value in the Surfaces table will edit the parameter's value, rather than changing the surface property directly. For instance, if a surface radius is set to the parameter "myRadius", editing that surface in the table will change the "myRadius" parameter and leave the surface still set to *myRadius*. This will also occur if the surface's property is set to a simple supported expression, such as *convertCurvature(myCurvature)*. In this case, the Surfaces table will set the "myCurvature" parameter to the corresponding curvature for the entered radius, instead of directly changing the surface radius to the entered value. But this will not happen for more complex or multi-parameter expressions like *myRadius + (factorValue * 2mm)*.

This feature is designed to make it easier to edit lens dimensions after running the Optimization Wizard or Tolerancing Wizard, which parameterize the lens system when configuring the optimization/analysis. However, it can lead to unexpected edits if custom parameters are used in the lens stack. Care should be taken when using custom parameters that the lens stack Surfaces table is editing the correct values. If this behavior is interfering with your use of parameters, you can bypass it by editing the surface properties directly with the "Edit" button.

Lens Stack Properties

Override Color

By default, lenses in the lens stack will be rendered in the 3D modelview using the colors of their assigned materials. If this property is checked, then Color and Transparency are used instead to render all lenses in the lens stack.

Color

The color used to render lenses in the 3D modelview, if Override Color is checked.

Transparency

The transparency used to render lenses in the 3D modelview, if Override Color is checked.

Trace Order

When Sequential Raytracing is enabled, this property determines the order in which separate lens stacks will be traced. This ordering is also used when finding the aperture stop and when solving lens diameters.

Origin

Defines the origin point of the lens stack. This location will be the origin point of the first surface in the stack. See the position vector documentation for explanations of the different ways the location can be specified.

Optical Axis

The direction in which surfaces will be added to the lens. The "back" of the lens stack is in this direction. See the orientation vector documentation for explanations of the different ways the orientation can be specified.

Max Surface Diameter

The maximum surface diameter that will be used while solving for diameters (see Lens Stack Surface Diameter).

2.4.2. Lens Stack Surfaces

Surfaces added to a lens stack define the front and back boundaries of the lenses in the stack. The most basic lens stack surface is the Surface type, which is a simple spherical cap defined by a radius of curvature.

Hide Dummy Surface

If a surface is in mid-air (surrounded by the Spatial Domain’s Background Material), this property determines whether or not the surface is shown in the modelview.

Type

The type of surface to be inserted in the lens stack.

- Surface: A spherical cap with a radius of curvature. This is the most basic type of surface.
- Stop: A surface with a stop centered around the apex of curvature.
- Image Plane: A surface that can be used to determine the location of the focal plane, if the Image Plane focus method is used.
- Asphere: An aspheric surface.

Material

The material following this surface in the lens stack. If this material is the same as the Spatial Domain’s Background Material, then there will be no lens after this surface. The last surface in a lens stack must be assigned the background material.

Boundary

The boundary condition or metasurface coating applied to this surface in the lens stack.

Thickness

The distance between the apex points of this surface and the next. If this surface is assigned a non-background material (see Material), this thickness will be the thickness of the lens component following this surface.

Surface Radius

The radius of curvature of this surface. A positive radius curves the edges of the surface towards the back of the lens stack (in the same direction as the Lens Stack Optical Axis), and a negative radius curves the edges towards the front.

Diameter

The clear aperture diameter of this surface. Lens diameters are the largest of the diameters of the front and back surfaces. For Stops, this also determines the diameter of the stop. If the diameter is zero, then the diameter will be automatically determined based on the point at which rays intersect the surface. To see the diameter of a lens stack surface, including any solved diameters, add a "Lens Stack Metrics" output to the Ray Result's Outputs folder, and open its LensStackData table.

Clear Aperture Margin

Provides a buffer zone at the edge of a lens. The actual lens diameter will be the sum of the Diameter and the Clear Aperture Margin.

Component Decenter

Moves a component from its default position in the lens stack by the specified position vector. The surface for which this is specified must be either the first surface in the lens stack or be preceded by the Spatial Domain's Background Material. The position of subsequent surfaces will be determined based on the decentered position of this surface, until after the next surface which is followed by the background material. This value is applied before the Component Tilt.

Component Tilt

Tilts a component away from the lens stack's Optical Axis by the specified orientation vector, with the tilt being applied at the vertex of the surface. Note that tilt is applied not in global coordinates but relative to the lens stack orientation, so (0,0,1) would be no tilt at all regardless of the orientation of the optical axis. Subsequent surfaces will follow the same tilted orientation, until after the next surface which is followed by the background material. This value is applied after the Component Decenter and has the same restrictions as to what surfaces it can be specified for.

Surface Decenter

Moves a surface from its default position in the current lens by the specified position vector. Any applicable Component Tilt and Component Decenter is applied first. The position of subsequent surfaces is unaffected. This value is applied before the Surface Tilt. Note that the boundary of the surface is not moved, only the vertex from which surface curvature is calculated.

Surface Tilt

Tilts a surface away from the Optical Axis by the specified orientation vector of the current lens, with the tilt being applied at the vertex of the surface. Note that tilt is applied not in global coordinates but relative to the lens orientation, so (0,0,1) would be no tilt at all regardless of the orientation of the optical axis. Subsequent surfaces are not affected. This value is applied after the Component Decenter and any Component Tilt and Component Decenter.

Lens Stack Stops

Stop surfaces are much like normal surfaces, except that in addition to the surface, a stop is centered around the apex of curvature, which can be used to define the simulation aperture.

Aperture Stop

> If checked, this stop's diameter is used to define the simulation's aperture diameter, which can also influence the diameter of the source in some cases (see Beam Source properties). If no stops have this property checked, the first one in the lens stack is used to define the aperture.

Lens Stack Image Plane

Image Plane surfaces are much like normal surfaces, but they can be used to determine the location of the focal plane if the Image Plane focus method is used. If this method is used, there should be only one Image Plane surface in the simulation.

Lens Stack Aspheres

Aspheres in the lens stack are much like normal surfaces, except instead of being defined by a radius of curvature alone, the sag of an aspheric surface is defined by one of the following equations (a fuller description of the primary forms and the polynomial portion can be found in the article Description of aspheric surfaces, Volume 8 Issue 3-4 of the journal Advanced Optical Technologies):

Conic Asphere Equation

The conic asphere equation is used for symmetric and asymmetric aspheres.

$$z(r) = \frac{r^2}{R\left(1+\sqrt{1-(1+\kappa)\frac{r^2}{R^2}}\right)} + [\text{Polynomial Terms}]$$

In the equation, R is the effective spherical radius, kappa is the conic constant, r is the cylindrical radial coordinate and z is the cylindrical axial coordinate. The local cylindrical coordinate system is defined by the origin and orientation of the surface; the orientation vector defines the z-axis and uniquely defines the r-phi plane. The polynomial terms are defined by the specific Asphere Equation.

Biconic Asphere Equation

$$z = \frac{\frac{x^2}{R_x} + \frac{y^2}{R_y}}{1+\sqrt{1-(1+\kappa_x)\frac{r^2}{R_x^2}-(1+\kappa_y)\frac{r^2}{R_y^2}}} + [\text{Polynomial Terms}]$$

In the equation, R_x and R_y are the effective spherical radii in the X and Y directions respectively, κ_x and κ_y are the conic constants in the X and Y directions, and x, y, and z are the Cartesian coordinates of the point on the surface, with r defined by $r^2 = x^2 + y^2$. The local coordinate system is defined by the origin, the orientation of the surface (which defines the z-axis), and the Asymmetry X Axis of the surface. The polynomial terms are defined by the specific Asphere Equation.

Asphere Equation

Defines what polynomial terms are used in the asphere equation for this surface, and whether it is conic (symmetric/asymmetric) or biconic. Options are:

- Even Order Symmetric Asphere: Conic Asphere Equation with polynomial terms $\alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \ldots$
- Even and Odd Order Symmetric Asphere: Conic Asphere Equation with polynomial terms $\alpha_1 r + \alpha_2 r^2 + \alpha_3 r^3 + \ldots$
- Even Order Asymmetric Asphere: Conic Asphere Equation with polynomial terms $\alpha_1 x^2 + \alpha_2 y^2 + \alpha_3 x^4 + \alpha_4 x^2 y^2 + \alpha_5 y^4 + \alpha_6 x^6 + \alpha_7 x^4 y^2 + \alpha_8 x^2 y^4 + \alpha_9 y^6 + \ldots$
- Even and Odd Order Asymmetric Asphere: Conic Asphere Equation with polynomial terms $\alpha_1 x + \alpha_2 y + \alpha_3 x^2 + \alpha_4 xy + \alpha_5 y^2 + \alpha_6 x^3 + \alpha_7 x^2 y + \alpha_8 xy^2 + \alpha_9 y^3 + \ldots$
- Even Order Biconic Asphere: Biconic Asphere Equation with polynomial terms $\alpha_1 x^2 + \alpha_2 y^2 + \alpha_3 x^4 + \alpha_4 x^2 y^2 + \alpha_5 y^4 + \alpha_6 x^6 + \alpha_7 x^4 y^2 + \alpha_8 x^2 y^4 + \alpha_9 y^6 + \ldots$
- Even and Odd Order Biconic Asphere: Biconic Asphere Equation with polynomial terms $\alpha_1 x + \alpha_2 y + \alpha_3 x^2 + \alpha_4 xy + \alpha_5 y^2 + \alpha_6 x^3 + \alpha_7 x^2 y + \alpha_8 xy^2 + \alpha_9 y^3 + \ldots$
- Symmetric Zernike Surface: Conic Asphere Equation with only symmetric Zernike polynomial terms (see Zernike Index for details).
- Conic Zernike Surface: Conic Asphere Equation with Zernike polynomial terms ordered according to the selected Zernike Index.
- Biconic Zernike Surface: Biconic Asphere Equation with Zernike polynomial terms ordered according to the selected Zernike Index.
- Symmetric Zernike Asphere: Conic Asphere Equation with two sets of polynomial terms - aspheric terms identical to those for the Even Order Symmetric Asphere, and Zernike polynomial terms ordered according to the Symmetric-only Zernike Index.
- Conic Zernike Asphere: Conic Asphere Equation with two sets of polynomial terms - aspheric terms identical to those for the Even Order Symmetric Asphere, and Zernike polynomial terms ordered according to the selected Zernike Index.
- Biconic Zernike Asphere: Biconic Asphere Equation with two sets of polynomial terms - aspheric terms identical to those for the Even Order Asymmetric Asphere, and Zernike polynomial terms ordered according to the selected Zernike Index.

Conic Constant Kappa

The conic constant term in the asphere surface equation (κ). For biconic aspheres, this represents the conic constant in the defined X direction (κ_x).

Conic Constant Kappa (Y)

The conic constant term in the Y direction (κ_y) for biconic aspheres. The regular Conic Constant Kappa in that case becomes the conic constant term along the Asymmetry X Axis. For symmetric and asymmetric aspheres, this term is unused.

Asymmetry X Axis

The axis (vector) which defines 'X' for asymmetric and biconic aspheres. Note that the axis is defined not in global coordinates but relative to the lens orientation. Since it must be perpendicular to the lens orientation, any 'Z' component will be ignored; if the vector chosen is purely Z or -Z, the default 'X' axis from the lens' orientation will be used.

Surface Radius (Y)

The radius of curvature of the surface along the asymmetry Y axis for biconic aspheres. The regular Surface Radius in that case becomes the radius of curvature along the asymmetry X axis. This property is not used for symmetric or asymmetric aspheres.

Polynomial Coefficients

The list of alpha coefficients in the asphere surface equation. Which polynomial term each coefficient is associated with is defined by the Asphere Equation.

Polynomial Units

The unit of length used for the radius-coordinate (or x/y coordinates) in the asphere surface equation. Most tools use millimeters.

Zernike Coefficients

The list of Zernike term coefficients in the asphere surface equation. The terms are ordered according to the selected Zernike Index. The labels indicate the index position and the n and l values. As an example, the second term in the Wyant index is $Z_1^{-1}(\rho, \phi)$, so its coefficient C_1^{-1} would be labelled $Z2(1, -1)$ in the editor.

Normalization Radius

The distance from the center of the surface at which the normalized radial distance ρ equals one. If this is set to zero, the radius used will be half of the surface diameter.

Zernike Index

The index selected determines the order of the Zernike terms displayed in the coefficient editor. The OSA/ANSI, Noll, Wyant, and Fringe/UA indices are described concisely in the Wikipedia article on Zernike polynomials. The Symmetric Only index includes only those terms $Z_n^l(\rho, \phi)$ where $l = 0$ and $n = 0, 2, 4, 6, \ldots$.

Extrapolate

Zernike surfaces are normally only defined up through a normalized radial distance (ρ) of one. If this box is checked, the Zernike equations will be evaluated for values of ρ greater than one where applicable. If this box is not checked, then if the surface extends beyond $\rho = 1$ the outer area will be a flat disk.

Normalize Terms

If checked, the Zernike terms will be normalized. See the Notes on Zernike Polynomials for a fuller description of normalization.

Notes on Zernike Polynomials

Zernike polynomials in reTORT can be expressed as:

$$\sum C_n^l\, Z_n^l(\rho, \phi)$$

Where C_n^l is the user-entered coefficient for the corresponding Zernike polynomial term, ρ is the normalized radius (the radial distance from the surface center to the point being evaluated, divided by the Normalization Radius), and ϕ is the azimuthal angle (measured counter-clockwise from the Asymmetry X Axis to the point being evaluated). The calculated value is considered to have the units of length defined by the Polynomial Units. The Zernike terms themselves are defined as:

$$Z_n^{-m}(\rho, \phi) = N_n^m R_n^m(\rho) sin(m\phi)$$

$$Z_n^m(\rho, \phi) = N_n^m R_n^m(\rho) cos(m\phi)$$

$$\text{Where: } m = |l|, \quad n \geq 0, \quad \text{and } (n - m)\%2 = 0$$

NOTE: References are inconsistent as to whether the definition of $Z_n^{-m}(\rho, \phi)$ should have a negative sign. We have followed what seems to be the majority usage.

The normalizing constant N_n^m is defined as the value for which:

$$\int_{\phi\, =\, 0}^{2\pi} \int_{\rho\, =\, 0}^{1} Z^2\, \rho\, d\rho\, d\phi = \pi$$

which results in the equation:

$$N_n^m = \sqrt{\frac{2(n+1)}{1+\delta_{m0}}}$$

where $\delta_{m0} = 1$ if $m = 0$, otherwise $\delta_{m0} = 0$. However, if the Normalize Terms flag is not checked, N_n^m is treated as having a value of 1.

A fuller description of the radial polynomial term $R_n^m(\rho)$ as well as the various indices used to order the Zernike terms can be found at the Wikipedia article on Zernike polynomials.

2.4.3. Bounded Elements

Bounded Elements are defined by an explicit list of bounding surfaces, such as spherical caps, aspheres, cylinders, and circles.

A ray enters an element when it collides with a surface belonging to that element, and exits the element when it collides with a second element. By this definition, a bounded element does not require a strict closed surface. However, modeling a geometry with open boundaries may result in rays that miss the second surface of an element and are not accurately refracted. This result is especially common when tracing GRIN regions, which means that closed boundaries must be specified when tracing a GRIN optic.

Bounded Element Properties

Name

See GEMSIF Object Properties - Name

Override Color

Allows the user to override a material’s color and specify a lens specific color.

Surfaces

See Elements - Surfaces.

Dimensions

See Element Properties.

Material

Material inside the element. If the referenced material is dispersive, then the resulting optic will demonstrate different behavior for different wavelengths. If the referenced material has a spatial (x,y,z or r,phi,z) dependence, then rays within the element will be traced using the GRIN engine.

Origin

See Object Properties - Origin

Normal Vector

See Element Properties.

Component Temperature

Temperature of the lens. Used when computing refractive index data for materials that incorporate temperature information.

2.4.4. Bounded Element Surfaces

Two surfaces may not be coincident; a bonded doublet or triplet lens must either use a "reference" surface to use a surface belonging to an existing lens, or a small air gap must be introduced between the two sublenses.

Surfaces in reTORT belong to exactly two elements. The second element is either the implicit background material, or a element with an explicit "reference" surface defined.

Individual surfaces are defined with their geometric properties as well as optical surface properties (perfect Anti-reflective (AR), perfectly absorbing (ABS), perfectly reflective (PEC), or normal (NONE)). Additional boundary conditions (including real AR and partially-reflective coatings and proper handling of surface roughness) will be added in the future.

Annular Ring

An Annular Ring is the planar surface between an inner and outer circle, defined by an inner and outer diameter. Rays do not interact with this surface inside the inner circle, or outside the outer circle.

Annular rings with an absorbing (ABS) boundary condition are used to model circular stops, and slots. In absorbing mode, rays contacting the surface are terminated, and do not generate reflected or refracted rays.

Annular Ring Properties

- Name - See GEMSIF Object Properties.
- Base Position - See Surface Properties.
- Normal Vector - See Surface Properties.
- Diameter - See Surface Properties.
- Inner Diameter - See Surface Properties.
- Boundary - See Surface Properties.

Asphere

Aspheric profiles are used to correct spherical and higher-order aberration. reTORT bounded element aspheres are surfaces extending from an inner to an outer diameter according to the selected Asphere Equation.

It is computationally inefficient to use an asphere with extremely large radius of curvature to model a flat surface; use a circle (or even a sphere with a large radius of curvature) instead.

Asphere Properties

- Name - See GEMSIF Object Properties.
- Base Position - See Surface Properties.
- Normal Vector - See Surface Properties.
- Diameter - See Surface Properties.
- Inner Diameter - See Surface Properties.
- Surface Radius - See Surface Properties.
- Asphere Equation - Specifies the asphere equation for this surface, which determines the base form (conic or biconic) as well as what polynomial terms are used. See the section on Lens Stack Asphere Equations for a complete description of the available equations. While both rotationally symmetric and rotationally asymmetric equations may be used, if a solid lens with an asymmetric surface is desired, the lens stack is a better choice since it can automatically adjust the shape of the connecting cylinder between the lens surfaces.
- Conic Constant Kappa - The conic constant term in the asphere surface equation (κ).

The kappa terms specify the base shape of the aspheric surface.

kappa	Conic Section
kappa < -1	hyperbola
kappa = -1	parabola
-1 < kappa < 0	ellipse (prolate spheroid)
kappa = 0	sphere
kappa > 0	ellipse (oblate spheroid)

- Polynomial Coefficients - The list of alpha coefficients in the asphere surface equation. Which polynomial term each coefficient is associated with is defined by the Asphere Equation.
- Polynomial Units - The unit of length used for the radius-coordinate (or x/y coordinates) in the asphere surface equation. Most tools use millimeters.
- Boundary - See Surface Properties.
- Zernike Coefficients:: The list of Zernike term coefficients in the asphere surface equation. The terms are ordered according to the selected Zernike Index. The labels indicate the index position and the n and l values. As an example, the second term in the Wyant index is $Z_1^{-1}(\rho, \phi)$, so its coefficient C_1^{-1} would be labelled $Z2(1, -1)$ in the editor.
- Normalization Radius:: The distance from the center of the surface at which the normalized radial distance ρ equals one. If this is set to zero, the radius used will be half of the surface diameter. If the normalization radius is set smaller than half the surface diameter, the Zernike equation will still be evaluated for values of [\large \rho] greater than one.

Circle

Planar circle defined by a radius, position, and orientation vector. Used for plano-convex lenses

Circle Properties

- Name - See GEMSIF Object Properties.
- Base Position - See Surface Properties.
- Normal Vector - See Surface Properties.
- Diameter - See Surface Properties.
- Boundary - See Surface Properties.

Cylinder

Cylindrical shell that is open at the two planar ends; only the side walls interact with rays. Cylinders are used to connect other surfaces.

Cylinder Properties

- Name - See GEMSIF Object Properties.
- Base Position - See Surface Properties.
- Normal Vector - See Surface Properties.
- Diameter - See Surface Properties.
- Height - Cylinder height.
- Boundary - See Surface Properties.

Group

Represents a collection of surfaces grouped together. A group of surfaces can be used as an import object to allow loading multiple surfaces at a time from a library.

Group Properties

- Name - See GEMSIF Object Properties.
- Surfaces - See Bounded Elements.
- Origin - See Object Properties.
- Normal Vector - See Surface Properties.

Reference

Surfaces may not be coincident, so any two elements that would naturally make contact over a finite area must be defined such that one references a surface defined by the other. Bonded doublet or triplet lenses will require this approach.

Reference Properties

- Name - See GEMSIF Object Properties.
- Element - The name of the Element object that is being referenced
- Surface - The name of the surface within the parent element that is being referenced.

Sector

A spherical sector is the spherical analog to the annular ring. It is a spherical cap with a region removed corresponding to a requested diameter about the origin. The position of the sector is still defined as the apex of the sphere, rather than the maximum point of the sector itself.

Sector Properties

- Name - See GEMSIF Object Properties.
- Base Position - See Surface Properties.

- Normal Vector - See Surface Properties.
- Diameter - See Surface Properties.
- Inner Diameter - See Surface Properties.
- Surface Radius - See Surface Properties.
- Boundary - See Surface Properties.

Sphere

Spherical cap surface, defined by a radius of curvature and a diameter. The origin of the sphere is defined as the apex of the spherical cap. The cap is defined as the intersection of the spherical hemisphere along the positive z-axis (as defined by the surface orientation) and a cylinder of the requested radius, centered at the origin. The diameter of a spherical cap cannot be larger than twice the radius of curvature.

The offset for a sphere’s base position can be calculated using the following equation:

$$\text{offset} = \text{Surface Radius} - \sqrt{(\text{Surface Radius})^2 - \left(\frac{1}{2} \cdot \text{Diameter}\right)^2}.$$

Sphere Properties

- Name - See GEMSIF Object Properties.
- Base Position - See Surface Properties.
- Normal Vector - See Surface Properties.
- Diameter - See Surface Properties.
- Surface Radius - See Surface Properties.
- Boundary - See Surface Properties.

Common Bounded Element Surface Properties

Base Position

Position of the surface relative to the owning element’s local coordinate system. See the position vector documentation for explanations of the different ways the position can be specified.

Normal Vector

Orientation of the surface relative to the owning element’s local coordinate system. See the orientation vector documentation for explanations of the different ways the orientation can be specified.

Diameter

Diameter of the surface normal to the local z-axis.

Inner Diameter

Inner diameter of a ring, section, or asphere normal to the local z-axis.

Surface Radius

Defines the radius of curvature for Spheres, Sectors, and Aspheres.

Boundary

Defines whether a surface is treated as Anti-Reflective (AR), Perfectly absorbing (ABS), Perfectly reflecting (PEC), or normal (NONE).

2.4.5. Paraxial Lens

The paraxial lens represents an ideal collimating element, specified by an infinitesimally thin circular surface and a focal length. By definition, all rays incident on the surface with the same incident angle are refracted to a point on the focal plane, and rays passing through the geometric center of the paraxial lens (specified by the origin property) are not refracted.

Paraxial Lens Properties

Name

See GEMSIF Object Properties - Name.

Origin

See Object Properties - Origin.

Normal Vector

See Element Properties.

Diameter

Diameter of the paraxial lens.

Focal Length

Focal Length of the paraxial lens. A collimated beam incident to one side of the lens will be focused to a point on the plane placed a focal length away from the paraxial lens along the lens's orientation vector.

2.4.6. Stops

Stop elements can be used to define apertures in a simulation.

Stop Properties

Name

See GEMSIF Object Properties - Name.

Dimensions

See Element Properties.

Color

See GEMSIF Properties

Transparency

See GEMSIF Properties

Origin

See Object Properties - Origin

Normal Vector

See Element Properties.

Diameter

Sets the diameter of the stop.

Aperture

If checked, this stop is used to define the aperture size of the simulation.

2.4.7. Common Element Properties

Dimensions

Stores the dimension lines used on the 2D diagram.

Normal Vector

Orientation of the lens relative to the global coordinate system.

The local coordinate system is defined by rotating the z-axis of the global Cartesian coordinate system to be aligned with the orientation vector. The same rotation is applied to the y- and x-axes to obtain the local rotated coordinate system. No control is provided at present over the local orientation or offset of the y- and x-axes.

See the orientation vector documentation for explanations of the different ways this vector can be specified.

2.5. Deviated Tolerance Parameters for reTORT ±

reTORT provides additional tolerance deviations that are specific to tolerancing a lens design. For more information, see Tolerance Analysis or Deviated Tolerance Parameters.

2.5.1. Radius Deviation

A radius tolerance deviation allows the specification of radius-of-curvature tolerances while using parameters that define surface curvature. Deviations in the radius based on the tolerance ranges are converted into the equivalent curvature when the simulation is perturbed during analysis.

The Negative Deviation and Positive Deviation must be length quantities, and represent the min and max radius-of-curvature tolerances. A positive radius deviation will flatten a surface with a positive curvature, but will sharpen the curve of a surface with a negative curvature. Meanwhile, the Parameter property must refer to a parameter that has units of inverse length, representing a surface curvature. For additional details and property information, see Deviated Tolerance Parameters.

2.5.2. Power-Fringe Deviation

Power or fringe tolerance deviations allow the specification of tolerances in terms of fringes while using parameters that define surface curvature. A fringe count (number of fringes observed when the manufactured lens is compared with a test surface) is converted into an equivalent curvature when the simulation is perturbed during analysis. These fringe deviations can perform better than curvature and radius-of-curvature deviations when tolerancing plano or near-plano surfaces.

The Negative Deviation and Positive Deviation must be unitless quantities, and represent the fringe count tolerances. Meanwhile, the Parameter property must refer to a parameter that has units of inverse length, representing a surface curvature. Additional details must also be provided, such as the Test Wavelength and Test Diameter used when comparing the manufactured lens to a test surface. For additional details and property information, see Deviated Tolerance Parameters.

One fringe is equivalent to a $\frac{\lambda}{2}$ change in the sag of the surface, where λ is the Test Wavelength and the sag is computed at the Test Diameter. A positive fringe deviation moves the sag in the direction of the lens stack’s orientation, toward the back of the lens. Thus, a positive fringe deviation will make a positive curvature more extreme, but will flatten a surface with a negative curvature.

Test Wavelength

Wavelength of light used when comparing the manufactured lens to a test surface.

Test Diameter

Diameter of the test surface used to measure the manufactured lens.

2.6. Results

Result objects specify the calculated outputs of the simulation engine. Multiple results may be added to request outputs computed with different inputs or parameters.

2.6.1. Ray Result

Specifies the fundamental reTORT ray data output. Additional outputs may be requested by adding "Outputs" children to the ray result.

RayResult Properties

Name

See GEMSIF Object Properties - Name

Enabled

When enabled, this result is re-evaluated when "Run All Results" is invoked on the simulation.

Generate Ray Diagram

Request generation of the ray diagram output from the ray tracer. If enabled, the rays will be displayed on the 3D model display.

Disable All Overlap Detection

Disables all checks for overlapping elements in the model and overlapping surfaces within lens stacks. Other validation checks are unaffected.

2.7. Ray Result Outputs

2.7.1. Optical Output

Compute outputs for imaging metrics at a requested focal plane.

Note that most results are computed relative to a focal point for each raybundle (that is, for each wavelength and incident angle), where the focal point is determined by the Focus Calculation Method. Most focus calculation methods determine the lateral focal point position using the intersection of the chief ray with the focal plane. The chief ray is the ray in the center of each raybundle. If the chief ray does not intersect the focal plane, then the first ray to intersect the focal plane will be used instead (the ray with the lowest Ray ID).

- Name - See GEMSIF Object Properties.
- Focus Calculation Method - This option determines how the focal plane location is determined.
    - Paraxial - The focal point is located at the intersection point of the paraxial rays in each bundle.
    - Marginal - The focal point is located at the intersection point of the marginal rays in each bundle.
    - Minimum RMS - The focal plane is located at the minimum RMS spot for all rays in each bundle. The lateral focal point position is determined by the intersection of the chief ray.

    - Minimum Peak - The focal plane is located at the minimum peak spot for all rays in each bundle. The lateral focal point position is determined by the intersection of the chief ray.
    - Image Plane - The focal plane is located at the Image Plane surface. The lateral focal point position is determined by the intersection of the chief ray.
    - Lens Stack Surface - The focal plane is located at the surface specified by the Lens Stack and Lens Stack Index properties. The lateral focal point position is determined by the intersection of the chief ray.
    - Manual - The focal plane is located at the point specified by the Focal Plane property. The lateral focal point position is determined by the intersection of the chief ray.
    - BFL - The focal plane is located a certain distance away from the back surface of the system along the Optical Axis, where the distance is specified by the Target Focal Length property. This can be used to target a certain back focal length during an optimization. The lateral focal point position is determined by the intersection of the chief ray.
    - EFL - The focal plane is located a certain distance away from the back principal point of the system along the Optical Axis, where the distance is specified by the Target Focal Length property. This can be used to target a certain effective focal length during an optimization. The lateral focal point position is determined by the intersection of the chief ray.
- Use Only Exit Rays* - For manually-specified Focus Calculation Methods, such as Image Plane, Manual, Lens Stack Surface, BFL, and EFL, all rays that pass through the plane will be included in the results if this option is not checked. This means that rays passing through the plane multiple times will be included in the results each time. If "Use Only Exit Rays" is selected, only rays that exit the lens system will be included in the results. This means that any rays passing through the focal plane will be ignored, and only rays that have exited the lens system will be counted. If the rays that exit the lens system do not cross this focal plane, they will be traced backwards if necessary to compute the results. For instance, if the focal plane is placed in front of a focusing lens, so that the focusing exit rays are traveling away from the plane, the spot size will be much larger, as if the exit rays had been traced backwards to meet the plane. If computing results about rays that actually pass through the focal plane is important, such as computing intermediate lens results at a dummy surface using the Lens Stack Surface method, then this property should be unchecked. This property is disabled if the Focus Calculation Method is set to Paraxial, Marginal, Minimum RMS, or Minimum Peak, because those methods are always based on only rays that exit the system.
- Focal Plane - Manually-specified location of the focal plane, when the Focus Calculation Method is set to Manual. If set to Image Plane or Lens Stack Surface, this property will be the location of the specified surface. Otherwise, this property will be the focal point of the first raybundle. See the position vector documentation for explanations of the different ways the location can be specified.
- Target Focal Length - Sets the desired back focal length or effective focal length at which measurements will be performed. This is used when the Focus Calculation Method is set to BFL or EFL.
- Focal Plane Diameter - Sets the effective diameter of the focal plane for the purpose of determining which rays intersect it. This is used when the Focus Calculation Method is anything other than Image Plane or Lens Stack Surface.

- Lens Stack - If the Focus Calculation Method is set to Lens Stack Surface, this property is used to specify which lens stack contains the surface in question.
- Lens Stack Index - If the Focus Calculation Method is set to Lens Stack Surface, this property specifies the index of the surface in question.
- Optical Axis - Sets the direction of the optical axis for the system, along which calculations are based. See the orientation vector documentation for explanations of the different ways the axis direction can be specified.
- Independent Group Focal Planes - This option forces each focal plane to be evaluated independently for each ray group.

Focal Plane Metrics

- Compute Spot Metrics - Request calculation of the spot metrics. This includes information such as focal point location.
- Generate Spot Diagram - Request calculation of the planar spot diagram.
- Use Centroids For Spot Size Calculation - If this is checked, the centroid of each ray bundle will be treated as the average of each ray's intersection with the focal plane, instead of where the chief ray intersects the plane. This will affect the RadialTRA result used for DLS spot size optimization, but will not affect the focal points or spot sizes given in the FocalPoint table.
- Enable Lateral Color Correction - This is only available if "Use Centroids For Spot Size Calculation" is selected. Lateral color correction means that all ray bundles at a given incident angle - regardless of wavelength - will be combined to determine a "joint" centroid.

Imaging Metrics

- Compute Pupil Function - Request generation of the Pupil Function.
- Compute Wavefront Error - Request generation of the OPD or Wavefront Error.
- Compute OTF - Request generation of the complex Optical Transfer Function.
- Compute MTF - Request generation of the Modulation Transfer Function image.
- Compute MTF Trace - Request generation of an MTF trace for plotting.
- Compute MTF Value - Evaluates a linear interpolation of the MTF trace at the spatial frequency specified by MTF Spatial Frequency.
- MTF Spatial Frequency - Sets the spatial frequency used to compute the MTF Value result.
- Compute Zernike Coefficients* - Request Zernike Polynomial Decomposition of the Wavefront Error.
- Zernike Polynomial Max Order* - Set the maximum order of Zernike Polynomial used in the decomposition. A hard limit has been emplaced at tenth order.
- Generate Zernike Recomposition* - Request generation of a recomposed wavefront error based on the Zernike Decomposition of the wavefront. Generally used to smooth over noisy data.

- Use Zernike Recomposition* - Request that the Zernike Recomposed wavefront error be used in place of the normally generated OPD.
- Compute Seidel Coefficients* - Request generation of the Seidel Aberration Coefficients for the given Wavefront Error. These coefficients are approximated through combination of appropriate Zernike Polynomial Coefficients.
- Compute RMS Wavefront Error - Request Calculation of the RMS value of the Wavefront Error.
- Compute Point Spread Function - Request generation of the Point Spread Function.

Ray Aberration Curves

- Compute Transverse Ray Aberration - Request a 2D plot of the Transverse Ray Aberration across the pupil.
- Compute Meridonal TRA - Request a trace of the Transverse Ray Aberration across the Meridonal Plane of the pupil.
- Compute Sagittal TRA - Request a trace of the Transverse Ray Aberration across the Sagittal Plane of the pupil.
- Compute Longitudinal Ray Aberration - Request a 2D plot of the Longitudinal Ray Aberration across the pupil.
- Compute Meridonal LRA - Request a trace of the Longitudinal Ray Aberration across the Meridonal Plane of the pupil.
- Compute Sagittal LRA - Request a trace of the Longitudinal Ray Aberration across the Sagittal Plane of the pupil.

Results with an asterisk (*) are advanced properties made visible through the "Toggle Advanced Properties Visibility" button indicated by the pair of gears (⚙) in the property editor.

2.7.2. Paraxial Output

This output contains paraxial information about the system including EFL, BFL, Entrance and Exit Pupil information and similar information. For more detail see the Paraxial System Data Table documentation.

Paraxial System Data Result Table

This output contains the following information on a Per Wavelength basis:

- System Back: The distance from the source along the optical axis to the physical back point of the system.
- Front Principal Point: The distance from the source along the optical axis to the front principal point or nodal point of the system.
- Back Principal Point: The distance from the source along the optical axis to the back principal point or nodal point of the system.

- Front Focal Point: The distance from the source along the optical axis to the front paraxial focal point.
- Back Focal Point: The distance from the source along the optical axis to the back paraxial focal point.
- Effective Focal Length: The distance from the Back Principal Point along the optical axis to the Back Focal Point.
- Back Focal Length: The distance from the System Back along the optical axis to the Back Focal Point.
- Entrance Pupil Location: The distance from the source along the optical axis to the center of the Entrance Pupil.
- Entrance Pupil Size: The diameter of the Entrance pupil.
- Exit Pupil Location: The distance from the source along the optical axis to the center of the Exit Pupil.
- Exit Pupil Size: The diameter of the Exit pupil.

2.8. Common reTORT Object Properties

2.8.1. Origin

Specify the location of the object relative to the parent or global coordinate system. See the position vector documentation for explanations of the different ways the location can be specified.

2.8.2. Orientation

Specify the orientation of the object in terms of the global coordinate system. See the orientation vector documentation for explanations of the different ways the location can be specified.

3. Views

See also GEMSIF Views.

3.1. Model View

Display a 3D rendered model of the optical system geometry and some of the output data.

3.1.1. Model View Properties

Axes Visible

Hide or show the axes in the model view.

Boundary Box Visible

Hide or show the boundary box in the model view.

4. Wizards

Several reTORT wizards are provided for your convenience. These wizards can assist in setting up optimizations and tolerance analyses, and in generating common plots such as spot diagrams.

4.1. Optimization Wizard

This is a wizard for setting up and running optimizations. It is divided into three tabs for selecting what properties to optimize, what goal to optimize for, and what optimizers to use.

4.1.1. Optimizer Selection

The first time the optimization wizard is run in a given simulation, a dialog appears with a dropdown to select what types of optimizers should be initialized. Either a global CMAES optimization, a local Damped Least-Squares optimization, or both, can be initialized.

4.1.2. Optimization Parameters Tab

This tab lists commonly-optimized properties of the current simulation, and allows control over which properties are optimized. Radii-of-curvature properties will be optimized as curvature.

Optimized Checkbox

If checked, the parameter for this row will be optimized.

Parameter Name

The name of the parameter for this row. Material and boundary coefficients are listed as the material/boundary name followed by the property name. Lens stack properties are listed by the name of the lens stack, followed by the number of the surface in the stack, followed by the property name.

Current Value

Current value of the parameter. For radii-of-curvature properties, this value is the current curvature.

Min Value

Minimum value of the parameter during the optimization. For radii-of-curvature properties, this is the minimum curvature.

Max Value

Maximum value of the parameter during the optimization. For radii-of-curvature properties, this is the maximum curvature.

Enforce Global Bounds

If checked, the global CMAES optimization will not allow properties to stray beyond their specified Min Value and Max Value. See CMAES - Respect Bounds.

Enforce Local Bounds

If checked, the local DLS optimization will not allow properties to stray beyond the specified Min Value and Max Value. See DLS - Respect Bounds.

4.1.3. Advanced Parameter Settings

This dialog lists more advanced parameter settings, such as parameter pickups.

Type

Defines the parameter type:

- Fixed - The parameter is set directly by the user, and will not be optimized.
- Optimized - The parameter is set directly by the user, and will be optimized when either of the Run Optimization buttons are pressed.
- Pickup - The parameter is defined by another parameter’s value, and is not set directly. The parameter’s value is defined as $(\text{Reference} \cdot \text{Factor}) + \text{Offset}$, using the Reference, Factor, and Offset properties.

Pickup Reference

For Pickup parameters, sets the source parameter whose value will be copied. Only parameters of the same category as the current parameter will be listed (eg, curvatures, thicknesses, material coefficients).

Pickup Factor

For Pickup parameters, sets the factor by which the source parameter’s value will be multiplied.

Pickup Offset

For Pickup parameters, sets an offset that is added to the source parameter’s scaled value.

4.1.4. Optimization Goals Tab

This tab configures the metric used for the optimization goal, as well as the focal plane at which that metric is computed.

Focal Method

Determines the method for computing the location of the focal plane.

- Image Plane - The focal plane is positioned at the Image Plane surface.

- Back Focal Length - The Focal Length property is used to set the distance of the focal plane from the back of the lens system.
- Effective Focal Length - The Focal Length property is used to set the distance of the focal plane from the second principal plane.

Focal Length

If the Focal Method is set to either Back Focal Length or Effective Focal Length, this determines the distance from the focal plane to the corresponding reference point.

Use Centroid For Spot Size Calculation

If this is checked, the centroid of each ray bundle will be treated as the average of each ray’s intersection with the focal plane, instead of where the chief ray intersects the plane.

Enable Lateral Color Correction

This is only available if "Use Centroid For Spot Size Calculation" is selected. Lateral color correction means that all ray bundles at a given incident angle - regardless of wavelength - will be combined to determine a "joint" centroid.

Primary Goal

Sets the metric used for optimization. Optimizations can minimize either spot size or RMS wavefront error.

Secondary Goals

Lens Center Thickness

Minimum and maximum allowable values for the center thickness of individual lenses in the lens stack. Optimizations may consider either, both, or neither.

Lens Edge Thickness

Minimum and maximum allowable values for the edge thickness of individual lenses in the lens stack. Optimizations may consider either, both, or neither.

Airgap Center Thickness

Minimum and maximum allowable values for the center thickness of individual airgaps in the lens stack. Optimizations may consider either, both, or neither.

4.1.5. Optimization Settings Tab

This tab defines which optimizers are configured by the wizard, and the optimizer-specific settings used for each type.

Global Optimizer Settings

Global Optimizer Enabled

If checked, the wizard will configure a global CMAES optimization.

Global Function Limit

The maximum number of simulation runs for a single optimization run. See CMAES - Function Call Limit.

Global Generation Limit

The maximum number of CMAES generations for a single optimization run. See CMAES - Generation Limit.

Global Enforce Bounds

If checked, the optimization will not allow properties to stray beyond their Min Value and Max Value specified in the Optimization Parameters Tab. See CMAES - Respect Bounds.

Local Optimizer Settings

Local Optimizer Enabled

If checked, the wizard will configure a local Damped Least-Squares optimization.

Local Function Limit

The maximum number of simulation runs for a single optimization run. See DLS - Function Call Limit.

Local Iteration Limit

The maximum number of CMAES iterations for a single optimization run. See DLS - Iteration Limit.

Local Enforce Bounds

If checked, the optimization will not allow properties to stray beyond their Min Value and Max Value specified in the Optimization Parameters Tab. See DLS - Respect Bounds.

4.2. Tolerancing Wizard

This is a wizard for setting up and running tolerance analyses. It is divided into two tabs for selecting which parameters to tolerance/compensate, and configuring the analysis settings. The run button at the bottom of the wizard will start the analysis, or stop an analysis that is already running.

Using this wizard will generate a tolerance analysis object called "AutoToleranceAnalysis", and an output called "AutoOpticalOutput". Several parameters beginning with "Auto" will also be generated as needed and applied to the relevant lens properties. The generated optical output and parameters may be shared with the Optimization Wizard.

4.2.1. Tolerancing Parameters Tab

This tab lists properties of the current simulation that are commonly used during a tolerance analysis, and allows control over which properties are toleranced or compensated. Most properties have their tolerance range and maximum compensation changes defined directly as a change in the parameter value. Surface curvature properties can specify their tolerance range in terms of curvature, radius-of-curvature, or power/fringe deviation. However, a surface curvature's maximum compensation changes must be in terms of curvature.

Tolerance Status

Determines whether this parameter is toleranced, compensated, or ignored during the tolerance analysis:

- Empty - Parameter will not included in the tolerance analysis.
- "TOL" - Parameter will be toleranced/perturbed during the tolerance analysis, in order to measure its effect on performance. At least one parameter must be toleranced to perform a tolerance analysis. If the parameter is a surface curvature, the tolerance range is also defined via the Tolerance Type column. Toleranced parameters also have access to the Advanced Distribution Settings dialog. For more information on deviated parameters in a tolerance analysis, see Deviated Tolerance Parameters.
- "CMP" - Parameter will be compensated during the tolerance analysis. The parameter's value will be optimized in order to minimize performance changes as a result of any toleranced parameters. For more information on compensated parameters in a tolerance analysis, see Compensated Tolerance Parameters.

Parameter Name

The name of the parameter for this row. Material and boundary coefficients are listed as the material/boundary name followed by the property name. Lens stack properties are listed by the name of the lens stack, followed by the number of the surface in the stack, followed by the property name.

Tolerance Type

Controls how the tolerance range (Negative Change and Positive Change) is defined for a parameter when tolerancing or compensating its value. Tolerances for surface curvature parameters can be specified in terms of curvature, radius-of-curvature, or a fringe count. Switching between the different tolerance types will automatically convert the tolerance range into the new tolerance type. Other toleranced parameters, such as lens thickness or material coefficients, can only have their tolerance ranges defined directly as changes in the parameter value. Compensated parameters are also restricted to direct changes in the parameter value.

- Direct - The negative/positive changes represent direct changes to the parameter's value. Units for the tolerance range must be compatible with the parameter's units. This corresponds to a Simple Tolerance Deviation or Compensated Tolerance Parameter, depending on if the parameter is toleranced or compensated.
- Curvature - The negative/positive changes represent changes in the curvature of a surface, and must have units of inverse-length. This corresponds to a Simple Tolerance Deviation or Compensated Tolerance Parameter, depending on if the parameter is toleranced or compensated, because surface curvatures are parameterized in terms of curvature.
- Radius - The negative/positive changes represent changes in the radius-of-curvature of a surface,

and must be lengths. This corresponds to a Radius Tolerance Deviation and is only available for toleranced surface curvatures.

- Power - The negative/positive changes represent changes in the fringe-count when the perturbed surface is compared to a test surface. The test surface is defined by the nominal/unperturbed surface curvature and the specified Test Diameter, and is compared to the perturbed surface at the Test Wavelength. The negative/positive changes must be unitless. This corresponds to a Power-Fringe Deviation and is only available for toleranced surface curvatures.

Negative Change

The maximum negative change or perturbation that can be applied to a parameter during tolerancing or compensation. For toleranced parameters, this represents the lower bound of the parameter’s tolerance range. For compensated parameters, this represents the lower bound of the parameter’s optimization range. The change value must be less than or equal to zero, and must have units compatible with the selected Tolerance Type. See Tolerance Deviation’s Negative Deviation property and Tolerance Compensator’s Max Negative Compensation property for more information.

Positive Change

The maximum positive change or perturbation that can be applied to a parameter during tolerancing or compensation. For toleranced parameters, this represents the upper bound of the parameter’s tolerance range. For compensated parameters, this represents the upper bound of the parameter’s optimization range. The change value must be greater than or equal to zero, and must have units compatible with the selected Tolerance Type. See Tolerance Deviation’s Positive Deviation property and Tolerance Compensator’s Max Positive Compensation property for more information.

4.2.2. Advanced Distribution Settings

This dialog lists probability distribution settings for a parameter. A Monte Carlo analysis will use these distribution settings to control how random perturbations are generated. These settings can be accessed via the gear button on the Tolerancing Parameters Tab (⚙), and are only available for toleranced parameters.

Distribution

Selects the type of probability distribution used for this tolerance parameter during a Monte Carlo analysis. For more information, see Tolerance Deviation’s Distribution property.

- Default - The probability distribution is defined by the Default Distribution.
- Uniform - The probability function is evenly distributed between the Negative Change and Positive Change values, such that any deviation value is equally likely.
- Truncated Gaussian - The probability function is a normal or Gaussian distribution, truncated between the Negative Change and Positive Change values. The precise shape of the Gaussian distribution is controlled by the Gaussian Settings Type.

- End-Point - Only the most extreme deviation values are possible; either the Negative Change or Positive Change will be selected at random and are equally likely.

Gaussian Settings Type

Determines how the shape of a Truncated Gaussian distribution is defined. For more information, see Tolerance Deviation’s Gaussian Settings Type property.

- Std Deviation Factor - The peak/mean of the Gaussian distribution is centered evenly between the Negative Change and the Positive Change. The standard deviation is set such that there are a specific number of multiples of the standard deviation between the mean and either end of the truncated distribution, with the number of multiples controlled by the Std Deviation Factor.
- Mean and Std Deviation - The peak/mean and standard deviation of the Gaussian distribution are set directly via the Gaussian Mean and Gaussian Std Deviation properties.

Std Deviation Factor

Used by the Std Deviation Factor Gaussian settings type to compute the standard deviation for a Gaussian distribution centered evenly between the Negative Change and Positive Change values. For a factor of *N*, the standard deviation is computed such that there are *N* multiples of the standard deviation between the mean and the negative/positive change values. This value must be unitless and greater than zero. For more information, see Tolerance Deviation’s Gaussian Std Deviation Factor property.

Gaussian Mean

Used by the Mean and Std Deviation Gaussian settings type to define the shape of the truncated normal distribution. This value is the location of the peak/mean of the Gaussian curve, and need not necessarily lie within the bounds of the distribution itself. Must have units compatible with the Negative Change and Positive Change properties, and will be automatically converted to match the current Tolerance Type. For more information, see Tolerance Deviation’s Gaussian Mean property.

Gaussian Std Deviation

Used by the Mean and Std Deviation Gaussian settings type to define the shape of the truncated normal distribution. This value is the standard deviation of the Gaussian curve, and must be nonzero. In addition, while negative values are allowed, they will be treated as though they are positive. Must have units compatible with the Negative Change and Positive Change properties, and will be automatically converted to match the current Tolerance Type. For more information, see Tolerance Deviation’s Gaussian Std Deviation property.

4.2.3. Tolerancing Settings Tab

This tab configures a variety of tolerance analysis settings, such as the metric used for the tolerance analysis, the focal plane at which that metric is computed, and default probability distributions.

The focal plane settings are shared with the Optimization Wizard.

Focal Method

The method for determining the location of the focal plane where the metric is calculated:

- Image Plane - The focal plane is positioned at the Image Plane surface.
- Back Focal Length - The Focal Length property is used to set the distance of the focal plane from the back of the lens system.
- Effective Focal Length - The Focal Length property is used to set the distance of the focal plane from the second principal plane.

Focal Length

If the Focal Method is set to either Back Focal Length or Effective Focal Length, this determines the distance from the focal plane to the corresponding reference point. For more information, see Optical Output’s Target Focal Length property.

Analysis Type

Sets the type of tolerance analysis to run. See Tolerance Analysis' Analysis Type property for more information.

- Direct Sensitivity - Performs a Direct Sensitivity analysis.
- Monte Carlo - Performs a Monte Carlo analysis.
- Sensitivity + Monte Carlo - Performs both a Direct Sensitivity analysis and a Monte Carlo analysis.

Metric

Sets the metric used to measure lens performance during the tolerance analysis. Each metric is summed over all ray bundles, where a ray bundle is the rays for a single wavelength and incident angle emitted from a source.

- RMS Spot Diameter - Measures the sum of the RMS spot diameter across all ray bundles.
- RMS Wavefront Error - Measures the sum of the RMS wavefront error across all ray bundles.
- MTF - Measures the sum of the MTF at the specified MTF Spatial Frequency across all ray bundles.

MTF Spatial Frequency

Sets the spatial frequency used to compute the MTF metric. Must have units of inverse-length. See Optical Output’s MTF Spatial Frequency property and Optical Output’s MTF Value result for more information.

Compensate Performance Changes

If checked, each perturbed simulation generated during the analysis will optimize the values of any compensated parameters, in order to minimize the change in performance from the nominal system. Note that this optimization will minimize the change in performance, and not the performance metric itself. For more information, see Tolerance Analysis' Compensate Performance

Changes property.

Monte Carlo Settings

Number of Simulations

Sets the number of perturbed simulations to generate during a Monte Carlo analysis. See Tolerance Analysis' Number of Simulations property for more information.

Default Distribution

Selects the default probability distribution used for tolerance parameters during a Monte Carlo analysis. If a Truncated Gaussian distribution is selected, the distribution is further defined by the Default Std Deviation Factor. See the per-parameter Distribution setting and Tolerance Deviation’s Distribution property for more information.

Default Std Deviation Factor

If the Default Distribution is set to Truncated Gaussian distribution, this controls the spread of the default probability distribution for toleranced parameters during a Monte Carlo analysis. For a factor of *N*, the standard deviation is computed such that there are *N* multiples of the standard deviation between the mean and the negative/positive change values. This value must be unitless and greater than zero. For more information, see the per-parameter Std Deviation Factor setting and Tolerance Deviation’s Gaussian Std Deviation Factor property.

Power Tolerance Settings

Test Diameter

When surface curvature parameters are toleranced using the Power tolerance type, the tolerance range represents the number of fringes observed when comparing a test lens to the manufactured/perturbed lens. This setting controls the diameter of the test surface that is compared against the manufactured lens. See the Power-Fringe Deviation object for more information on fringes. See also Power-Fringe Deviation’s Test Diameter property.

Test Wavelength

When surface curvature parameters are toleranced using the Power tolerance type, the tolerance range represents the number of fringes observed when comparing a test lens to the manufactured/perturbed lens. This setting controls the wavelength of light used when testing the manufactured lens against the test surface. See the Power-Fringe Deviation object for more information on fringes. See also Power-Fringe Deviation’s Test Wavelength property.

4.3. Plotting Wizards

Several plotting wizards are provided to automatically configure common plots, graphs, and diagrams of a selected focal plane, source, wavelength, and incident angle. Clicking 'OK' will create a plot with the selected settings. Running a plotting wizard multiple times will overwrite previously-generated diagrams from that wizard.

Optical Output

Selects the optical output whose focal plane will be used to compute the diagram/plot. If "Create New Output" is selected, then a new optical output will be created with the specified focal plane settings. See Optical Output.

Focus Method

Sets the Focal Calculation Method for the selected optical output.

Focal Plane

Sets the Focal Plane location for the selected optical output. This property is only displayed if the Focus Method is set to Manual. See the position vector documentation for explanations of the different ways the location can be specified.

Focal Length

Sets the Target Focal Length of the selected optical output. This property is only displayed if the Focus Method is set to BFL or EFL.

Lens Stack

Selects the Lens Stack to be used by the selected optical output. This property is only displayed if the Focus Method is set to Lens Stack Surface.

Lens Stack Index

Sets the Lens Stack Index to be used by the selected optical output. This property is only displayed if the Focus Method is set to Lens Stack Surface.

Source

Selects the source whose rays will be used to generate the diagram/plot. If "All Sources" is selected, then rays from all sources will be included in the plot. See Sources.

Wavelength

Selects the wavelength to include in the diagram/plot. If "All Wavelengths" is selected, then rays of any wavelength will be included in the plot. See Spectral Lines.

Incident Elevation Angle

Selects the incident elevation angle to include in the diagram/plot. If "All Angles" is selected, then rays of any incident elevation angle will be included in the plot. See Incident Elevation Angle.

4.3.1. Spot Diagram Wizard

The spot diagram wizard configures a 2D spot diagram illustrating where each ray intersects the selected focal plane. See Plotting Wizards for more information.

4.3.2. Wavefront Error Plot Wizard

The wavefront error plot wizard configures a contour plot illustrating the wavefront error at the selected focal plane. See Plotting Wizards for more information.

4.3.3. MTF Graph Wizard

The MTF graph wizard configures a graph of the MTF trace at the selected focal plane. See Plotting Wizards for more information.

Trace Axis

Selects whether the meridonal MTF, sagittal MTF, or both are plotted.

4.3.4. TRA Trace Wizard

The TRA trace wizard configures a graph of the TRA trace at the selected focal plane. See Plotting Wizards for more information.

Trace Axis

Selects whether the meridonal TRA, sagittal TRA, or both are plotted.

4.4. Save Wavefront Profile Wizard

The Save Wavefront Profile wizard saves the current wavefront profile at the selected focal plane for use in later wavefront-matching optimizations. See Plotting Wizards for more information.

See also Clear Wavefront Profiles Wizard.

Name

The name under which to save the wavefront profile.

4.5. Clear Wavefront Profiles Wizard

This wizard deletes any wavefront profiles that have been saved by the Save Wavefront Profile Wizard, and disables any wavefront-matching results that used those profiles.

Topical Documentation for PFSS

1. PFSS Simulation

PFSS models the electromagnetic response of a periodic tiling of rectangular multilayered metallo-dielectric unit cells to an incident plane wave. The PFSS Simulation plugin provides the additional capabilities to GEMSIF to design, build, and simulate periodic tiling of unit cells. Two solvers are contained in the plugin, PFEBI and PMM. Details...

1.1. Objects and Collections

- Actions
- Parameters
- Boundary Conditions
- Materials
- Spatial Domain
- Excitations
- Layers
- Patterns
- Results
- Views
- Layer Dock
- Stackup View

1.2. Details

A simulation is defined and constructed by the objects contained within it. The objects can be broken into three categories: general simulation characteristics, component properties, and layers. General simulation properties contain the physical dimensions of the simulation and consist of a Spatial Domain object and one or more excitation objects. Component properties define the building blocks of the simulation. These consist of Boundary Conditions, Materials, and Pattern objects. Finally, Layers consist of the Layer object which reference the component property objects to build the simulation. A simulation must have at least one material or boundary condition layer.

1.3. See Also

- Introduction to PFSS

- PFSS Simulation Model
- PFSS Solvers

2. Objects and Properties

2.1. Spatial Domain

Only one Spatial Domain object is permitted for each PFSS simulation. The spatial domain objects specifies the physical dimensions of the unit cell and mesh properties used by the solver.

2.1.1. Spatial Domain Properties

Name

See GEMSIF Object Properties - Name

Bottom Deembed Distance

Specifies the phase deembedding distance to be applied to the bottom surface of the unit cell. The bottom deembedding distance affects both the reflection and transmission phase values.

Property: Length, default value "0 cm".

NOTE	See the deembed explanation in Top Deembed Distance for an explanation of the deembed property.

Top Deembed Distance

Specify the phase deembedding distance to be applied to the top surface of the unit cell. The top deembedding distance affects both the reflection and transmission phase values.

Property: Length, default value "0 cm".

When designing for a specific reflection or transmission phase profile, often the designer will have a specific phase reference plane that may not correspond to the top layer of the simulation domain. This factor is relevant in the design of high-impedance surfaces (Sievenpiper surfaces, or Artificial Magnetic Conductors) or reflectarray unit cells. Setting the deembed distance moves the phase reference plane inwards into the topmost (bottommost) layer of the simulation. Deembedding will only consider the material parameters of the topmost (bottommost) layer even if the deembeding thickness is greater than the top (bottom) layer thickness.

Height

Specify the size of the unit cell along the y-axis.

Property: Length, default value "1 m".

NOTE Please see width for constraints and considerations on the unit cell and pixel dimensions.

Max Voxel Aspect Ratio

Specify the maximum aspect ratio (layer thickness / max (pixel width, pixel height)) of the voxels in finite-thickness material layers. Set to zero to disable automatic layer subdivision.

Property: Positive double quantity.

NOTE This property is used only by the PFEBI solver and ignored by the PMM solver.

When modeling layers of finite-thickness dielectric in PFEBI, the layer as specified in the simulation configuration must be split into thinner layers for use by the solver. The solver prefers the voxels (whose dimensions are determined by the layer thickness and the domain properties width, height, nX, and nY) to be nearly cubical (aspect ratio of 1.) However, for very thick layers, this constraint can result in extremely long simulation times. Increasing the maximum aspect ratio generates fewer simulation layers, reducing the simulation complexity and computational time. Using excessively large voxel aspect ratios may reduce the accuracy of the PFEBI solver.

IMPORTANT If the Max Voxel Aspect Ratio is set to zero, then material layers will not be automatically remeshed into thin layers by the solver, and the calculated results may be inaccurate.

Pixels, X

Specify the number of pixels in the unit cell along the x-axis.

Property: Positive, nonzero integer quantity.

The number of pixels in the simulation is used by the solvers as the simulation mesh. The effective number of pixels in the simulation The number of pixels in the domain should be specified according to the geometrical or fabrication constraints of the design, with increased meshing for simulations at high frequencies specified by an oversampling factor in the relevant Excitation objects.

NOTE Please see width for constraints and considerations on the pixel count and dimensions within the simulation.

Pixels, Y

Specify the number of pixels in the unit cell along the y-axis.

Property: Positive, nonzero integer quantity.

NOTE Please see width for constraints and considerations on the pixel count and dimensions within the simulation.

Constraint

See Object Properties - Pattern Constraint

Width

Specify the size of the unit cell along the x-axis.

Property: Length, default value "1 m".

The PFSS solvers are sensitive to the physical size of the unit cell. The formulations of the solvers were derived with the assumption that the unit cell lateral dimensions are each less than a wavelength in the dielectric material; therefore, the highest accuracy and simulation speed is achieved when modeling periodic structures whose unit cells are smaller than a wavelength. In practice, the constraint is often reduced to require that the unit cell dimensions be less than one wavelength in free space. Violating this soft constraint will increase the number of solver iterations required to converge to a solution, and may decrease solver accuracy, especially the PMM solver.

The unit cell width (or height) and the number of pixels along the x axis (or y axis) should be selected such that the maximum pixel width (width/nX) is less than λ/10 at the highest simulation frequency within the material with the highest index in the simulation (see the PMM and PFEBI solver limitations for more details). Alternatively, nX may be set to a logical size to satisfy the geometric constraints on the design problem and use the oversample property of the excitation object to a sufficiently large value to satisfy the maximum pixel size constraints.

2.2. Excitations

Excitations specify the frequency and angular sweep settings of the incident field used by the solver to excite the periodic geometry defined by the layers and patterns in the simulation. One or more excitations must be defined to perform a simulation. All excitations may be run at once, or the individual excitations may be selected and run to obtain only a single set of results at a time.

2.2.1. Excitation Types

- Excitation with Automatic Solver : Selects the best solver based on the simulation geometry.
- PFEBI Excitation : Creates an excitation object that uses the PFEBI Solver (flexible). Primarily used for modeling geometries with nonuniform dielectric layers.
- PMM Excitation : Creates an excitation object that uses the PMM Solver (faster). Primarily used for modeling geometries with homogeneous dielectric layers.

2.2.2. Frequency Considerations

When specifying the simulation frequency range, the electrical size of the unit cell at the upper and lower frequency points must be considered. The solvers may have stability or accuracy problems when the electrical size of the unit cell increases past one wavelength, and the pixel dimensions should remain smaller than λ/10 in both dimensions (see the PMM and PFEBI solver limitations for more details). In simulations over a wide frequency range, it can be worthwhile to break the simulation range into multiple excitations so that the oversample factor may be increased for the high frequency simulations, providing accuracy without increasing simulation time at the low frequencies.

Depending on the required simulation frequency range, the user should either specify an automatic oversampling factor or a high enough factor to satisfy the pixel size constraints.

2.2.3. Auto Solver

PFSS will automatically attempt to identify which solver to use. The Auto Solver is biased towards PMM for improved simulation speed unless unsupported features (such as patterned material layers, or pixel dimensions that are not powers of 2) are requested in the simulation.

The Auto Solver excitation also lacks some of the more advanced simulation constraints that can be set using a particular solver's excitation.

Auto Solver Properties

Name

See GEMSIF Object Properties - Name

Frequency Count

See Excitation Properties - Frequency Count

Frequency Start

See Excitation Properties - Frequency Start

Frequency Step

See Excitation Properties - Frequency Step

Frequency Step Type

See Excitation Properties - Frequency Step Type

Frequency Stop

See Excitation Properties - Frequency Stop

Near Fields

See Excitation Properties - Near Fields

Oversample

See Excitation Properties - Oversample

Phi Start

See Excitation Properties - Phi Start

Phi Step

See Excitation Properties - Phi Step

Phi Stop

See Excitation Properties - Phi Stop

Polarization

See Excitation Properties - Polarization

Scattered Fields

See Excitation Properties - Scattered Fields

Theta Start

See Excitation Properties - Theta Start

Theta Step

See Excitation Properties - Theta Step

Theta Stop

See Excitation Properties - Theta Stop

2.2.4. PFEBI Solver

Creates an excitation object that uses the PFEBI Solver (flexible). Primarily used for modeling geometries with nonuniform dielectric layers.

See the Solver Criteria Table for a quick reference to the best solver to use.

PFEBI Solver Properties

Name

See GEMSIF Object Properties - Name

Frequency Count

See Excitation Properties - Frequency Count

Frequency Start

See Excitation Properties - Frequency Start

Frequency Step

See Excitation Properties - Frequency Step

Frequency Step Type

See Excitation Properties - Frequency Step Type

Frequency Stop

See Excitation Properties - Frequency Stop

Maximum Iterations

See Excitation Properties - Maximum Iterations

Minimum Iterations

See Excitation Properties - Minimum Iterations

Near Fields

See Excitation Properties - Near Fields

Oversample

See Excitation Properties - Oversample

Phi Start

See Excitation Properties - Phi Start

Phi Step

See Excitation Properties - Phi Step

Phi Stop

See Excitation Properties - Phi Stop

Polarization

See Excitation Properties - Polarization

Scattered Fields

See Excitation Properties - Scattered Fields

Theta Start

See Excitation Properties - Theta Start

Theta Step

See Excitation Properties - Theta Step

Theta Stop

See Excitation Properties - Theta Stop

Tolerance

See Excitation Properties - Tolerance

2.2.5. PMM Solver

Creates an excitation object that uses the PMM Solver (faster). Primarily used for modeling geometries with homogeneous dielectric layers.

See the Solver Criteria Table for a quick reference to the best solver to use.

PMM Solver Properties

Name

See GEMSIF Object Properties - Name

Frequency Count

See Excitation Properties - Frequency Count

Frequency Start

See Excitation Properties - Frequency Start

Frequency Step

See Excitation Properties - Frequency Step

Frequency Step Type

See Excitation Properties - Frequency Step Type

Frequency Stop

See Excitation Properties - Frequency Stop

Maximum Iterations

See Excitation Properties - Maximum Iterations

Minimum Iterations

See Excitation Properties - Minimum Iterations

Near Fields

See Excitation Properties - Near Fields

Oversample

See Excitation Properties - Oversample

Phi Start

See Excitation Properties - Phi Start

Phi Step

See Excitation Properties - Phi Step

Phi Stop

See Excitation Properties - Phi Stop

Polarization

See Excitation Properties - Polarization

Scattered Fields

See Excitation Properties - Scattered Fields

Solver Method

Specifies the calculation method of the solver. Default specification is "LUD"

Theta Start

See Excitation Properties - Theta Start

Theta Step

See Excitation Properties - Theta Step

Theta Stop

See Excitation Properties - Theta Stop

Tolerance

See Excitation Properties - Tolerance

2.2.6. Solver Criteria

	PMM	PFEBI
Speed	Fast	Slow
Dielectric	Uniform	Nonuniform
Material Layer	Homogeneous	Patterned
Pixel Dimensions	Values must be a power of 2	Any
Boundary Layers Patterns	Multiple adjacent	Single

2.2.7. Excitation Properties

Frequency Step Count

Displays the number of frequency steps in a frequency sweep.

Property: Positive, non-zero integer.

NOTE frequencyStop is used when frequencyStepType is set to either Count or Logarithmic.

Frequency Start

Lower frequency limit for the frequency sweep. If Frequency Stop-Frequency Start < Frequency Step, then Frequency Start is the only frequency simulated.

Property value: Frequency, default value "1 GHz"

NOTE see Frequency Considerations for more information on the impact of the frequency sweep range.

Frequency Step

Step size for the frequency sweep. Either Frequency Step or Frequency Step Count are required, depending on the value of Frequency Step Type.

Property value: Frequency, default value "1 GHz"

NOTE Frequency Step is used only when Frequency Step Type is set to Linear.

Frequency Step Type

Frequency Step Type is an enumerated field that may only take on one of the following values:

Linear

Specify the start frequency, step size, and maximum allowed frequency. Note that, unless the step size is an exact factor of the frequency band specified by Frequency Start - Frequency Stop, then Frequency Stop will not be included as a simulation point.

Count

Specify an exact number of frequency points to be simulated in Frequency Count in even steps between Frequency Start and Frequency Stop.

Logarithmic

Specify a sweep with logarithmic spacing between frequencies Frequency Start and Frequency Stop with exactly Frequency Count frequency points in the sweep.

The default value is Linear, which requires the properties Frequency Start, Frequency Stop, and Frequency Step to be specified. Count and Logarithmic require Frequency Start, Frequency Stop, and Frequency Count to be specified.

Frequency Stop

Upper frequency limit for the frequency sweep. If Frequency Step is specified, then the highest frequency to be simulated is $f_{max} = (\text{Frequency Start}) + n \cdot (\text{Frequency Step})$ where n is the largest integer such that $f_{max} \leq$ Frequency Stop.

Property value: Frequency, default value "1 GHz"

NOTE see Frequency Considerations for more information on the impact of the frequency sweep range.

Maximum Iterations

The maximum number of iterations allowed before convergence is terminated in the BCG solver. If the solver requires more than the specified number of iterations to converge, then the convergence will be terminated early and the current results returned, with possible loss of accuracy. Reducing the maximum iteration count can decrease the worst-case simulation time, but potentially increase the probability of simulation inaccuracies.

Positive integer, default value of 4000.

NOTE This property is only used if solverMethod is "BCG", and ignored otherwise.

Minimum Iterations

The minimum number of iterations allowed before convergence in the BCG solver. Positive integer, with a default of 3. Increasing the minimum iteration count can help to prevent preconvergence in some problems.

NOTE This property is only used if solverMethod is "BCG", and ignored otherwise.

Near Fields

Currently not used.

Oversample

Specify the oversampling factor for the unit cell pixel counts nX and nY from the simulation domain. If oversample is set to zero, the program will automatically select an oversample value based on the excitation wavelength and the unit cell and pixel dimensions specified in the domain object. A non-unity oversample value replaces each pixel in the simulation by a group of n x n pixels for the solver. For example, a simulation of an 8x8 periodic geometry with an oversample property of 3 will be modeled with a 24x24 pixel geometry by the solver. The unit cell dimensions do not change when modifying oversample. Increasing the oversample factor increases the simulation mesh size in the solver and increases the required solution time.

Both PFSS solvers use the pixel count as specified as the domain to form their simulation mesh. The solvers converge with higher accuracy when the pixel dimensions are small with respect to the wavelength (maximum pixel dimension should be less than lambda/10 within the dielectric material at each simulation frequency); when this constraint is not met, the iterative solvers may produce accurate results, inaccurate results, or may not converge at all.

The oversample property of the Excitation object allows the geometry of a simulation to be specified at a 1-1 scaling with the fabrication requirements, even if such pixels would be very large with respect to a wavelength. Specifying a nonunity oversample factor to increase the grid resolution when simulating at high frequencies will help to ensure simulation accuracy.

Property value: Positive integer, default value of 1

NOTE Oversample value of 0 corresponds to "Auto" and a value of 1 corresponds to "None".

Phi Start

Specify the lower bound of the phi sweep of the incident plane wave. In general, phi may take on values between 0 and 360 degrees (0 and 2π rad), PMM restricts Phi Start to 1. Zero degrees is parallel with the positive X-axis, ninety degrees is parallel to the positive Y-axis.

The phi angle of the incident wave determines the orientation of the electric field vector with respect to the X-axis, which will be different between the TE and TM incident polarizations.

Property: Angle dimensioned quantity, default value 0 or 1 deg.

Phi Step

Specify the step size of the phi sweep of the incident plane wave.

Property: Angle dimensioned quantity, default value 1 deg.

Phi Stop

Specify the upper bound of the phi sweep of the incident plane wave. If Phi Stop is less than or equal to Phi Step, then only a single incident phi angle is simulated. Phi Stop is generally limited to 360 degrees, PMM restricts Phi Stop to 359 degrees.

Property: Angle dimensioned quantity, default value 0 or 1 deg.

Polarization

Specify the polarization of the incident field. Polarization is an enumerated value that must take on one of the following values:

TE

Transverse-electric field polarization; the incident plane wave is oriented such that the electric field is perpendicular to the normal vector of the unit cell surface.

TM

Transverse-magnetic field polarization; the incident plane wave is oriented such that the magnetic field is perpendicular to the normal vector of the unit cell surface.

At normal incidence, TE and TM polarizations are identical.

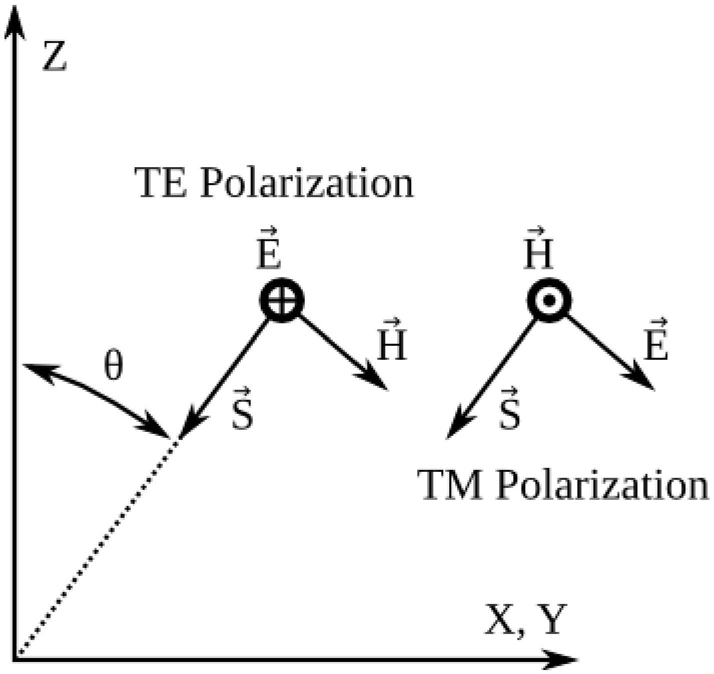


Figure 1. Polarization diagram showing the difference between TE and TM polarizations.

When designing for a specific polarization behavior, different selections for the pattern symmetry may be helpful. For example, 8-fold symmetry is generally beneficial for producing a polarization-independent response, while rotationally-symmetric or asymmetric designs will promote chiral behavior in the resulting devices.

Scattered Fields

Currently not used.

Solver Method

Specify the linear equation solver. solverMethod is an enumerated quantity that may take on one of the following values:

LU

(default) Use the direct LU-decomposition solver to find the solution to the linear MoM matrix equation.

BCG

Use the FFT-accelerated iterative Bi-conjugate Gradient (BCG) solver to solve the MoM matrix equation.

For large FSS problems, the BCG iterative solver accelerated by the FFT is faster than the direct LU solver. But, for small FSS problems, the direct LU solver is faster than the BCG iterative solver via the

FFT acceleration technique. Changing this property may be able to improve performance for some problems.

Theta Start

Specify the lower bound of the theta sweep of the incident plane wave. Theta may take on values between 0 and 90 degrees (π/180 and π/2 rad), PMM restricts Theta Start to 1. Zero degrees indicates normal incidence, ninety degrees is grazing incidence.

Property: Angle dimensioned quantity, default value 0 or 1 deg.

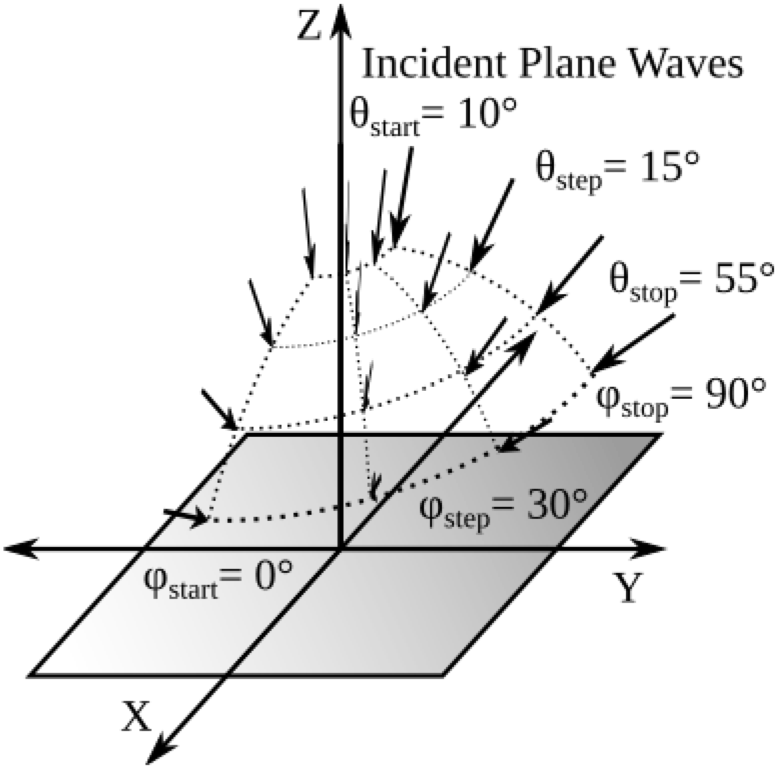


Figure 2. Illustration of theta and phi angle ranges within the simulation geometry.

Theta Step

Specify the step size of the theta sweep of the incident plane wave.

Property: Angle dimensioned quantity, default value 1 deg.

Theta Stop

Specify the upper bound of the theta sweep of the incident plane wave. If thetaStop is less than or equal to thetaStop, then only a single incident theta angle is simulated. Theta Stop is generally limited to 90 degrees, PMM restricts Theta Stop to 89 degrees.

Property: Angle dimensioned quantity, default value 0 or 1 deg.

Tolerance

The convergence tolerance setting for the bi-conjugate gradient (BCG) iterative linear equation solver. A positive quantity less than one, 0.001 by default. Decrease the value to increase the difficulty of the convergence condition.

NOTE This property is only used if solverMethod is "BCG", and ignored otherwise.

2.3. Boundary Conditions

Boundary conditions represent infinitely thin layers used to specify the electromagnetic behavior of surfaces. Boundary conditions are used to model metallic surfaces (lossy or ideal) for RF designs, as well as lumped elements. At optical and infrared wavelengths, it is typically more accurate to model metals and conductors as lossy dielectrics, rather than as a boundary layer.

2.3.1. Boundary Condition Types

- Impedance Boundary Condition : Specify a complex surface impedance value as the surface property.
- Material Boundary Condition : Specify the surface impedance based on the complex and (potentially) dispersive permittivity and permeability of an isotropic material.
- Lumped RLC Element Boundary Condition : Specify the surface impedance to be equivalent to a parallel or series combination of lumped resistance, capacitance, and inductance.
- Simple Boundary : Transparent (no boundary condition) or PEC.

2.3.2. Impedance Boundary

The impedance boundary condition specifies a nondispersive complex surface impedance. The surface impedance is specified in units of ohms, and may be lossy, lossless, inductive, capacitive, or any combination. If the impedance is small (near zero), then the impedance boundary condition should be replaced with a PEC boundary condition.

Impedance Boundary Properties

Name

See GEMSIF Object Properties - Name

Color

See GEMSIF Object Properties - Color

Impedance

Complex surface impedance, specified as a complex number in units of ohms. If the magnitude of the impedance is too small, the solver will prompt the user to replace the impedance boundary condition with a PEC boundary condition.

Property: Impedance, default value 0 ohms.

Transparency

See GEMSIF Object Properties - Transparency

2.3.3. Material Boundary

The material boundary condition specifies the complex surface impedance using a potentially dispersive and lossy isotropic material. The effective surface impedance will be computed from the complex material permittivity and permeability.

The material boundary condition may be used to simulate a thin layer of lossy copper in an RF simulation, for example. At higher frequencies, however (infrared and optical), metals and conductors should be specified with volumetric layers instead of a boundary condition for higher simulation accuracy.

The specified material must not be an active (gain) medium. Any active materials specified in a simulation will generate an error.

NOTE At present, only nondispersive material parameters are supported, and must be entered directly into the boundary condition editor. Library integration with the GEMSIF material library will be added in the future.

Material Boundary Properties

Name

See GEMSIF Object Properties - Name

Color

See GEMSIF Object Properties - Color

Epsilon

Complex relative electric permittivity of the effective material. Unitless quantity, default to 1 (free space).

Mu

Complex relative magnetic permeability of the effective material. Unitless quantity, default to 1 (free space).

Sigma

Material conductivity in units of Siemens/meter. Default value is "0 S / 1m".

Transparency

See GEMSIF Object Properties - Transparency

2.3.4. RLC Boundary

The RLC boundary condition allows the specification of the frequency-dispersive surface impedance as a combination of lumped element values (resistance, capacitance, inductance) in either a parallel or series configuration. Each pixel of the simulation that is marked with an RLC boundary condition, regardless of the physical size of the pixel or model, will have an assigned surface impedance that produces the effect of the requested lumped element values.

The orientation of the lumped elements must be specified as either the X- or Y-axes. A single simulation may only have a single x-directed and a single y-directed RLC boundary condition for series connections, and one of each direction for the parallel specification as well, although the same boundary condition may be applied to multiple pixels in the model and on multiple layers.

RLC Boundary Properties

Name

See GEMSIF Object Properties - Name

R

The resistance must be specified as a dimensioned quantity, with units of ohm, Mohm, etc. A negative value for resistance will disable the resistor from the parallel or series combination.

L

The inductance must be specified as a dimensioned quantity, with units of H, mH, uH, etc. A negative value for inductance will disable the inductor from the parallel or series combination.

C

The capacitance must be specified as a dimensioned quantity, with units of F, uF, nF, etc. A negative value for capacitance will disable the capacitor from the parallel or series combination.

Color

See GEMSIF Object Properties - Color

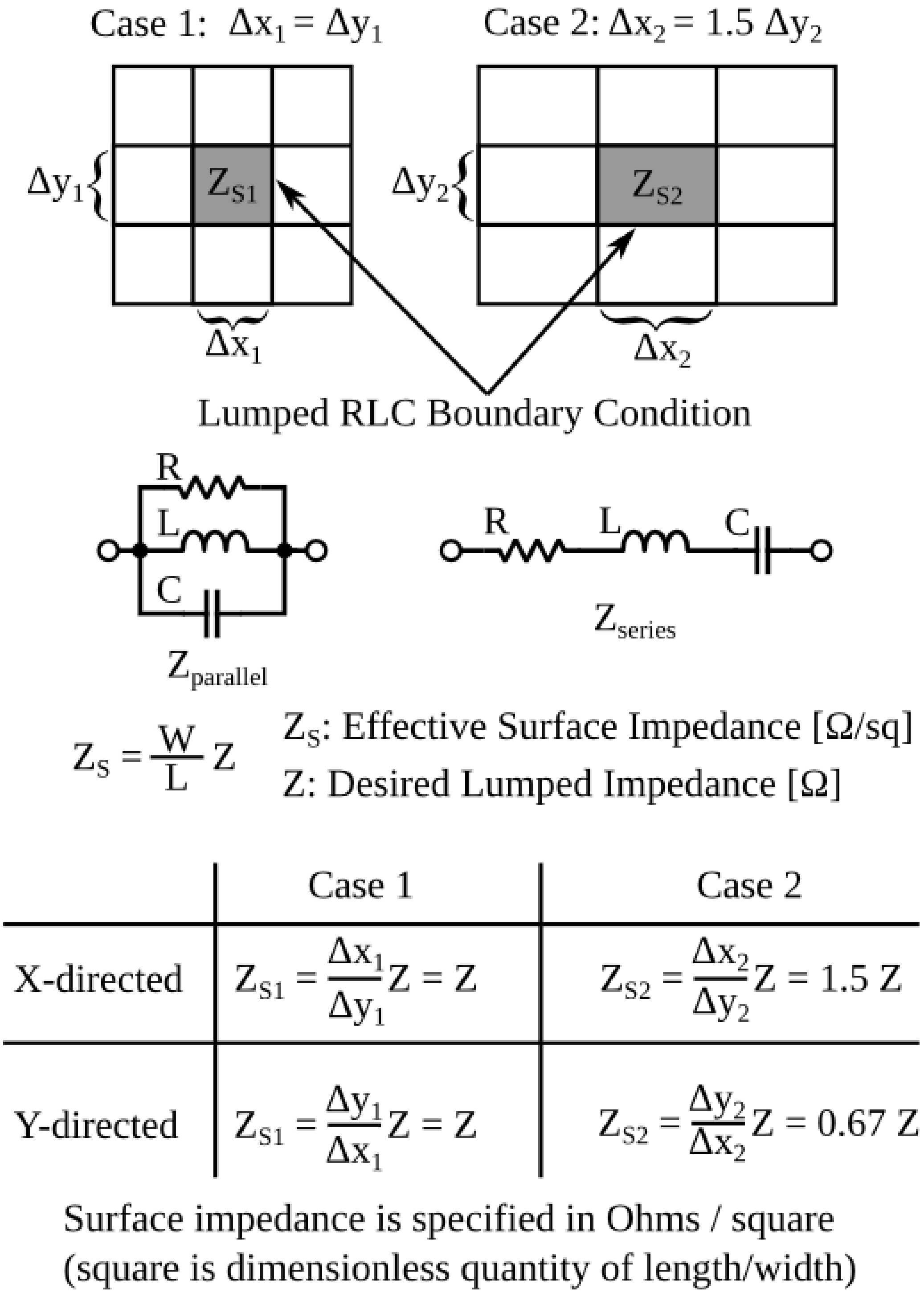
Direction

The specified direction of the lumped boundary condition does not affect the operation of element in different directions, but it does affect the calculation of the effective surface impedance of the pixel in the case of non-square pixels. For rectangular pixels, the desired lumped element value is only correct for current propagating in the specified direction. In square geometries, the direction parameter has no effect on the results.

Direction is an enumerated quantity that must take on one of the following values:

- x : The lumped element will connect the front and back faces of the pixel through the lumped impedance.
- y : The lumped element will connect the left and right faces of the pixel through the lumped

impedance.


Case 1: $\Delta x_1 = \Delta y_1$
Case 2: $\Delta x_2 = 1.5\ \Delta y_2$
Δy_1
Z_{S1}
Δx_1
Δy_2
Z_{S2}
Δx_2
Lumped RLC Boundary Condition
R
L
C
$Z_{parallel}$
R
L
C
Z_{series}
$Z_S = \frac{W}{L} Z$
Z_S: Effective Surface Impedance [Ω/sq]
Z: Desired Lumped Impedance [Ω]

	Case 1	Case 2
X-directed	$Z_{S1} = \frac{\Delta x_1}{\Delta y_1} Z = Z$	$Z_{S2} = \frac{\Delta x_2}{\Delta y_2} Z = 1.5\ Z$
Y-directed	$Z_{S1} = \frac{\Delta y_1}{\Delta x_1} Z = Z$	$Z_{S2} = \frac{\Delta y_2}{\Delta x_2} Z = 0.67\ Z$

Surface impedance is specified in Ohms / square
(square is dimensionless quantity of length/width)

Figure 3. Calculation of effective surface impedance for lumped element boundary condition, and importance of the direction parametter for non-square pixels.

CAUTION

Within a single simulation, only a single x-oriented and a single y-oriented RLC boundary condition for either the series or parallel configuration may be specified due to a limitation within the solver.

Transparency

See GEMSIF Object Properties - Transparency

Type

Type is an enumerated quantity that must take on one of the following values:

- Series : The effective lumped impedance will be computed with a series connection of the constituent elements.
- Parallel : The effective lumped impedance will be computed with a parallel combination of the constituent elements.

2.3.5. Simple Boundary

The simple boundary specifies either a null boundary condition or a PEC boundary condition, based on the value of the type property.

Simple Boundary Properties

Name

See GEMSIF Object Properties - Name

Color

See GEMSIF Object Properties - Color

Transparency

See GEMSIF Object Properties - Transparency

Type

Type is an enumerated quantity that must take on one of the following values:

- transparent : Specifies a transparent, or empty, boundary condition.
- pec : Specifies a perfectly-electrically conducting boundary condition.

2.4. Pattern

A pattern is a rectangular array or image of boolean values that may be either true or false. Each value is associated with a pixel. When associated with a material or boundary condition in a pattern-layer, the pattern applies the specified material/boundary condition to its assigned layer at each pixel. The pixel dimensions of the pattern may be any factor of of the simulation size specified in Domain. Patterns may be assigned a symmetry constrain that forces reflective and/or rotational symmetry along one or both axes (x and/or y).

Patterns may be specified as a bit string that is mapped directly to the list of pixels, as a rectangular region specified as a start and end point, or as a real-valued function of the pixel coordinates over which a threshold is applied to determine the boolean state of each pixel.

2.4.1. Binary Pattern

A binary pattern uses an array of bits to represent a pattern’s pixels. The standard Editor provides the user with the ability to draw their desired pattern by clicking on each pixel individually. Right clicking on a pixel will erase the pixel.

Binary Pattern Properties

Name

See GEMSIF Object Properties - Name

Bits

Binary patterns are stored as an array of bits equal to the number of pixels in a pattern (nX times nY) divided by the number of pixels per select from the symmetry choice.

nX

See Pattern Properties - nX

nY

See Pattern Properties - nY

Pattern Constraint

See Object Properties - Pattern Constraint

Symmetry

See Pattern Properties - Symmetry

2.4.2. Function Pattern

Javascript defined pattern. Not currently implemented.

Function Pattern Properties

Name

See GEMSIF Object Properties - Name

Function

Function used to describe the pattern.

nX

See Pattern Properties - nX

nY

See Pattern Properties - nY

Pattern Constraint

See Object Properties - Pattern Constraint

Symmetry

See Pattern Properties - Symmetry

2.4.3. Rectangle Pattern

Describes a rectangular pattern.

Rectangular Pattern Properties

Name

See GEMSIF Object Properties - Name

Corner1

Defines the bottom left corner of the rectangle.

Corner2

Defines the top right corner of the rectangle.

nX

See Pattern Properties - nX

nY

See Pattern Properties - nY

Pattern Constraint

See Object Properties - Pattern Constraint

Polarity

Defines the rectangle as either a sheet or a cutout.

Symmetry

See Pattern Properties - Symmetry

2.4.4. Pattern Properties

nX

The number of pixels present in the X direction. It must be less than or equal to the Pixels X value in the Spatial Domain.

nY

The number of pixels present in the Y direction. It must be less than or equal to the Pixels Y value in the Spatial Domain.

symmetry

- None - No symmetry. Single pixel per select.
- Ref180x - Reflection symmetry about the middle X row. Two pixels per select.
- Ref180y - Reflection symmetry about the middle Y column. Two pixels per select.
- Rot180 - Rotational symmetry. Two pixels per select.
- Ref90 - Reflection symmetry about the X and Y axis. Four pixels per select.
- Rot90 - Rotational symmetry about the X and Y axis. Four pixels per select.
- RefRot90 - Reflection and Rotational symmetry about the X and Y axis. Eight pixels per select.

2.5. Layers

The collection of layers within a simulation constructs the geometry of the unit cell. Layers may specify a dielectric material of finite thickness or an infinitely thin boundary condition layer. Two boundary layers may not be adjacent and must be separated by a material layer.

NOTE PMM cannot model inhomogeneous material layers; only PFEBI allows this capability. Only PMM supports the modeling of inhomogeneous Boundary layers.

2.5.1. Layer Types

- Material Layer : Patterned or unpatterned dielectric layer of finite thickness.
- Boundary Layer : Patterned or unpatterned boundary condition layer (PEC, Impedance, lumped element).

2.5.2. Boundary Layer

A boundary layer represents an infinitely-thin boundary condition between two material regions.

Boundary Layer Properties

Name

See GEMSIF Object Properties - Name

Boundary

Set the background boundary condition of the layer to any named boundary condition defined in the simulation. Initially, all pixels across the layer are set to the specified boundary condition. This setting may be overridden by any included patternLayers in the patterns property.

Pattern Constraint

See Object Properties - Pattern Constraint

Patterns

See Layer Properties - Patterns.

Position

See Layer Properties - Position.

2.5.3. Material Layer

Material layers represent homogeneous and inhomogeneous dielectric slabs. Multiple patterns assigned to multiple material may be applied to a single layer. The enabled pixels in the topmost pattern will take precedence over the bottom patterns.

Material Layer Properties

Name

See GEMSIF Object Properties - Name

Material

Specify the name of the default material of the layer. Any included patterns will be applied over top of the background material.

Max Voxel Aspect

Override the value of Domain.maxVoxelAspect for this layer only. Setting to zero (default value) uses the global setting from Domain.

NOTE The maxVoxelAspect is only used by the PFEBI solver.

See Also Domain.maxVoxelAspect

Pattern Constraint

See Object Properties - Pattern Constraint

Patterns

See Layer Properties - Patterns.

Position

See Layer Properties - Position.

Thickness

Specify the layer thickness. Thick layers (with respect to the lateral pixel dimensions) may be subdivided into thinner layers for the PFEBI solver based on the maxVoxelAspect setting.

Property: Length, default value "1 cm".

2.5.4. Pattern Layer

A pattern-layer is the direct child of the layer objects. A pattern-layer specifies a pattered material or boundary condition to be imposed on another material or boundary condition layer.

Material Pattern-Layer

Material pattern-layers may only be added to Material Layer objects.

Properties

- Name - See GEMSIF Object Properties.
- Index - See Pattern Layer Properties.
- Material - Specify the name of the material to be used with this pattern-layer. The material will be applied to each pixel of the layer where the pattern pixel is set.
- Pattern - See Pattern Layer Properties.

Boundary Pattern-Layer

Boundary pattern-layers may only be added to BoundaryLayer objects.

Properties

- Name - See GEMSIF Object Properties.
- Boundary : The specified boundary condition will be applied to each pixel of the layer where the pattern pixel is set.
- Face : The face property specifies to which face of the material voxel the boundary condition should be applied. For this to apply, the boundary layer is assumed to lie between two material layers. The Top face is the shared face between the two material layers; the bottom face is the bottom face of the lower layer.

  NOTE The face property is only used by PFEBI. PMM will not execute for any value other than "Top".

  The face property may hold any one of the following values. The default value is "Top".

  - Top
  - Bottom
  - Left
  - Right

    - Front
    - Back
- index : See Pattern Layer Properties
- pattern : See Pattern Layer Properties

Pattern Layer Properties

Index

Specify the order of the patternlayer. No two pattern-layers that are children of the same layer may share the same index value. PatternLayers are applied to the underlying layer in increasing order starting at index zero.

When edited through the GUI interface, changing the index of one pattern-layer will update any other pattern-layers that would share the new index value to prevent conflicts.

Pattern

Name of the pattern to be included in the layer.

2.5.5. Layer Properties

Patterns

A layer includes zero or more pattern objects, each associated with a material or boundary condition. The patterns property contains an array of PatternLayer names.

NOTE The list of patterns must be empty for Material Layers when using the PMM solver.

Position

The position property specifies the layer’s position within the multilayer stack of the unit cell. The layer position is a zero-based index, where low indices are at the bottom of the stack. No two layers may share the same index; when edited in the GUI, positions are automatically updated to ensure that all layers are provided with a unique index. Direct editing of the property will move other layer positions in order to accommodate changes made to the select layer.

2.6. Results

A results file is created for every excitation and is linked to a HDF5 (.hdf) file when a simulation is first saved. The HDF5 file contains the results of a simulation after it has been run and will appear in the same directory that the simulation is saved in. When a simulation is moved, the HDF5 file will need to be moved as well and relinked in order for the plotted result to display. Otherwise, the simulation will have to be run again.

An excitation can be validated and run from the results file by right clicking on it. In addition, once a simulation is run, results can be plotted by right clicking the desired results file and selecting "Create plot". If a results file is deleted, it can be recreated by validating or running the corresponding excitation again.

2.6.1. Results Properties

Name

See GEMSIF Object Properties - Name

Enable for Run All

Allows the user to exclude the path excitation from "Run all" commands.

Property: String, default value "true".

Path

Path to results file.

Excitation

Name of the connected excitation.

2.7. General Object Properties

2.7.1. Pattern Constraint

Specify the name of a user-defined Action as a constraint filter on the allowable patterns that may be applied within the current simulation.

WARNING	This property is not yet implemented.

The named action will be called during binary pattern generation with a trial pattern. The action must return a modified pattern that satisfies any design constraints

3. Docks

Currently, PFSS contains a single dock, the Layer Editor Dock.

3.1. Layer Editor

The Layer Editor Dock provides the user with a 2D side view of the layer stack that constructs the unit cell. The dock can be used to select a layer, add a layer, or remove a layer. The dock can also be used to export an image of the layer stack to the clipboard.

3.1.1. Layer Editor Properties

Name

See GEMSIF Object Properties - Name.

Floating

See Dock Properties - Floating.

Location

See Dock Properties - Location.

Next Dock

See Dock Properties - Next Dock.

Pinned

See Dock Properties - Pinned.

Size

See Dock Properties - Size.

Tabbed

See Dock Properties - Tabbed.

Title

See Dock Properties - Title.

4. Views

PFSS utilizes both the Action View and the Plot2D view from GEMSIF. For more information on these views, see GEMSIF Views.

In addition to the GEMSIF views, PFSS has a Layer Stackup View that displays a 3D model of the layer stack.

4.1. Layer Stackup View

The Layer Stackup View provides a 3D rendering of the layer stack. Each layer is rendered based on the color and transparency of the material or boundary condition.

The user can freely rotate the 3D model, add axes, and dimensions. The 3D model can be saved and printed from this view as well.

Selecting a layer will open the layer’s object if the Property Editor Dock is open, allowing the user to easily edit the layer. The user can turn off either type of layer, allowing for easier selection or viewing of a particular type of layer.

4.1.1. Stackup View Properties

Name

See GEMSIF Object Properties - Name.

Window Geometry

See GEMSIF View Properties - Window Geometry.

Window State

See GEMSIF View Properties - Window State.

Window Visible

See GEMSIF View Properties - Window Visible.

Axes Visible

Enables visible axes on the 3D model. Value is true or false.

Boundaries Visible

Turns visible Boundary Condition Layers on or off. Value is true or false.

Dimensions Visible

Enable dimensions on the 3D model. Value is true or false.

Expansion Factor

Configure the layers within the simulation to be separated by a percentage factor relative to the total unit cell thickness. The expansion factor is 0 by default, which corresponds to no expansion. With no expansion, all layers are displayed as simulated. The view can be expanded for better visualization by increasing the expansion factor to 100 (interpreted as percent) or more.

Materials Visible

Turns visible Material Layers on or off. Value true or false.

rulerVisible

Displays a ruler if enabled. True or false.

Title

See GEMSIF View Properties - Title

User Manual for PFSS

1. Introduction to PFSS

The PFSS tool is used to rapidly and efficiently solve the problem of electromagnetic scattering from periodic, 2.5D structures for any incidence angle or polarization. The software is suitable for designing devices for RF, terahertz, infrared, and optical applications. As a specialized simulation tool, the solvers within PFSS can rapidly compute the scattering properties from periodic metallodielectric frequency selective surfaces, metamaterials, and antennas. The efficient nature of the solvers allows the use of GEMSIF's advanced stochastic optimizers to select the design properties.

1.1. Features

The simulation capabilities of the PFSS tool are summarized in the list below. Details on the capabilities of the individual solvers PMM and PFEBI are also available.

- Doubly-periodic rectangular multilayered unit cells
- Fast, efficient calculation of scattering parameters from complex, inhomogeneous layered media
- Tight integration with optimization algorithms
- Plane wave incidence angular sweeps in theta and phi
- Frequency sweeps
- Dual-polarization excitation with cross-polarized reflection and transmission analysis
- Absorptivity and emissivity analysis
- Homogeneous and inhomogeneous material and boundary condition layers
- Isotropic, anisotropic, and bianisotropic dielectric materials
- Isotropic magnetic materials
- Lossy and lossless dielectric and magnetic materials
- Dispersive and nondispersive materials
- Integration with GEMSIF material database
- Parallel and series RLC lumped elements
- Surface impedance boundary conditions across any voxel face
- Thoroughly tested and validated simulation kernels

1.2. Simulation Model

Each simulation model is represented by a single, rectangular cuboid unit cell consisting of one or more volumetric layers of dielectric media that are (optionally) separated by planar layers of a specific boundary condition. Available boundary conditions include perfect electric conductors, surface impedance, lumped elements, and surface impedance calculated for specific permittivity and permeability. PFSS supports isotropic, anisotropic, and bianisotropic materials with optional frequency dispersion. The individual layers are discretized into collections of rectangular pixels, where the material properties or boundary condition of each pixel may be independently specified. Layers may be homogeneous or inhomogeneous depending on the solver, and boundary conditions may also be specified on any faces of a volumetric material voxel, which allows for the modeling of vias and other 2.5D structures.

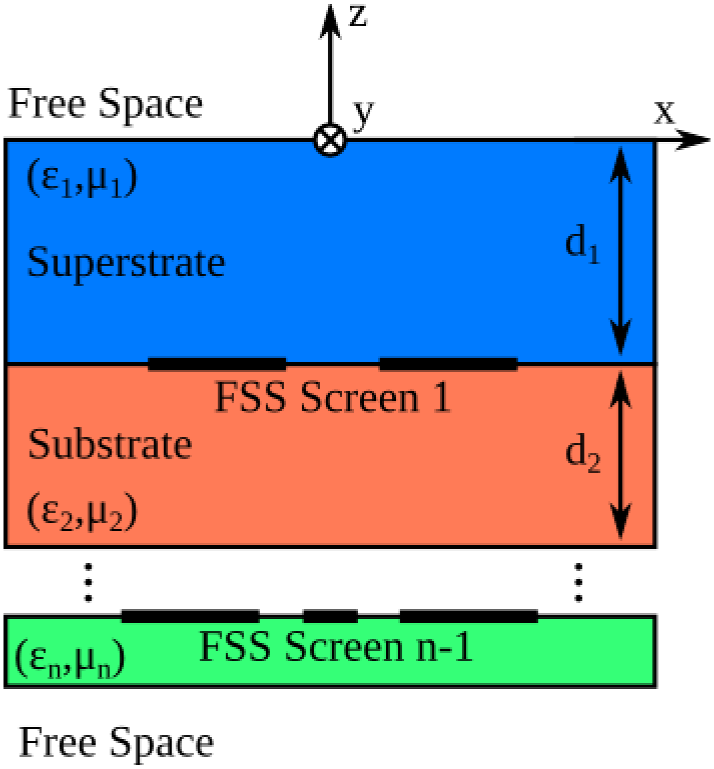


Figure 4. Geometry of multilayer metallodielectric geometry.

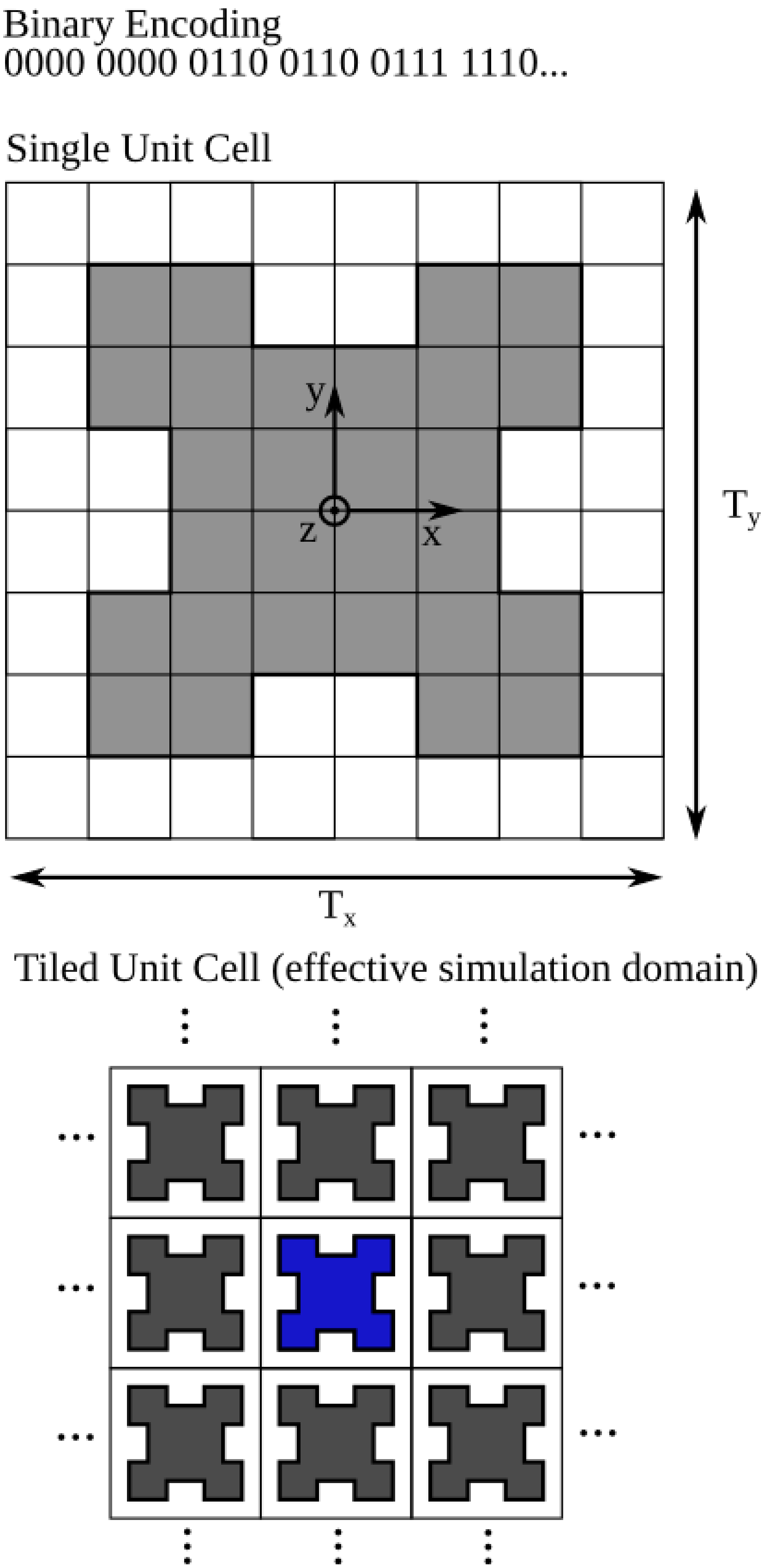


Figure 5. Patterned unit cell dimensions and geometry for an 8x8 pattern.

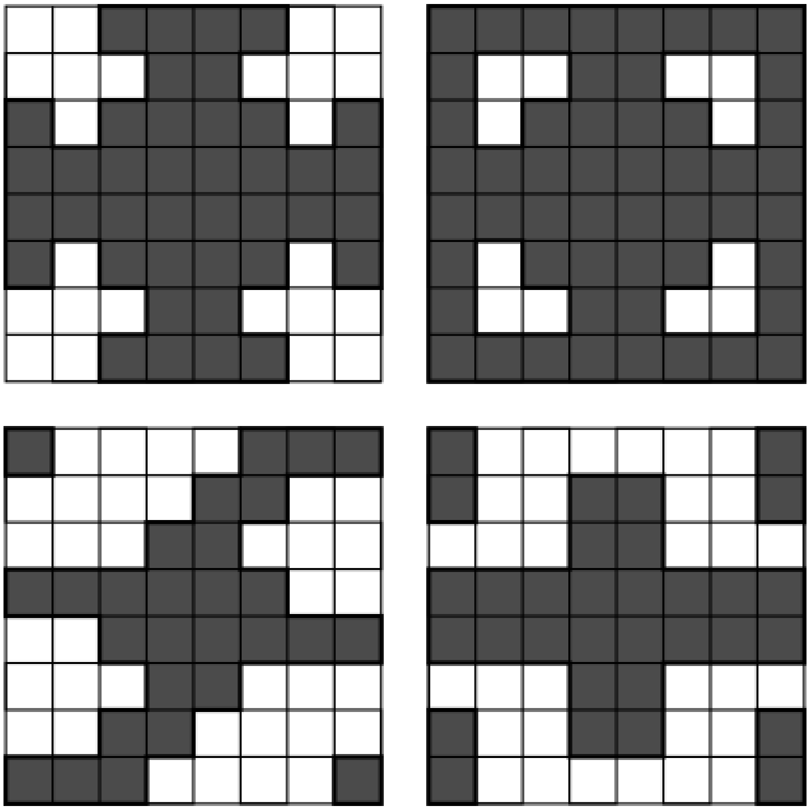


Figure 6. Demonstration pixel patterns.

Periodic boundary conditions are used for the x- and y-faces of the simulation domain, which allows a simulation of a single unit cell to accurately predict the electromagnetic response of an infinite array of unit cells. The response of a finite array of unit cells, as the array size becomes large, converges to the response of the infinite tiling. Commonly, 10-50 unit cells can be considered to have a bulk behavior where the periodic simulation results are an accurate predictor of the finite array response. The periodic unit cell response is most accurate for the unit cells in the center of the array, while it is less valid for elements on the boundaries of the array. For a large array, however, the response of the edge elements is generally unimportant for the behavior of the array as a whole, and the bulk, effective response of the periodic unit cell may be accurately applied.

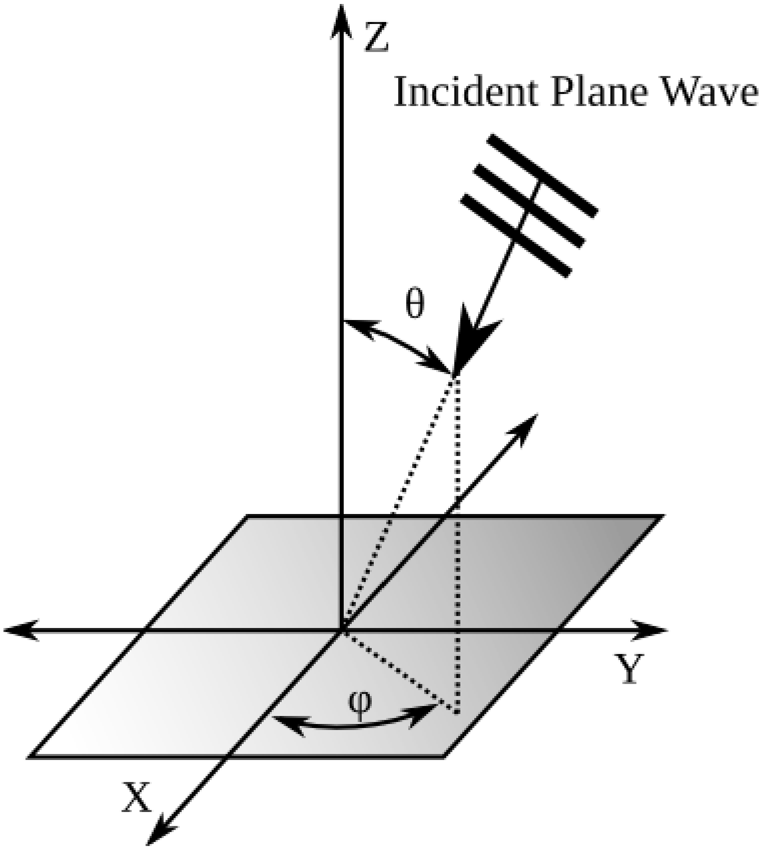


Figure 7. Illustration of theta and phi angle ranges within the simulation geometry.

2. Design and Usage Example

This user manual provides a detailed walk through of setting up, running, and plotting a PFSS simulation. The example will utilize a simple half meter, half wavelength dipole antenna to keep the results simple. For more complex examples, please see the additional tutorials.

2.1. See Also

- GEMSIF Topical Documentation
- PFSS Topical Documentation

2.2. Creating a PFSS Simulation

Every simulation is a child object of a System object. System objects can contain any number of simulations and are the objects that are saved.

1. The first step in creating a simulation is to create a System.

Once the System object is created, a PFSS simulation object can be created. Simulation objects can be created by either creating a new simulation or loading a template, typically one created by the user. This tutorial will use the blank simulation template to avoid the default boundary condition and material objects that exist in a default simulation.

2. Create a simulation using the "Empty" template.

2.3. Saving and Loading a System and Simulations

- Saving a System will give the System a name. Save the system as "Half Wavelength Dipole".
- To save a simulation as a template, right click on the simulation and select "Save as Template". Once saved, the simulation and settings can be loaded as a new template.

NOTE GEMSIF will autosave the model at regular intervals even if it has not been saved before.

2.4. Building a Model

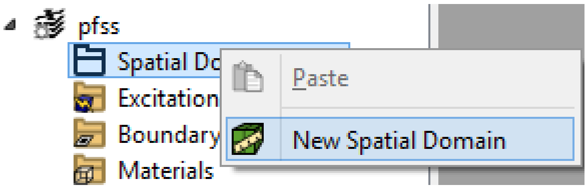
Creating a simulation model requires several objects to be defined. The objects can be broken into three categories: general simulation characteristics, component properties, and layers. General simulation properties contain the physical dimensions of the simulation and consist of a Domain object and one or more excitation objects. Component properties are used to define the electromagnetic properties of the materials to be used in the unit cell. These consist of Boundary Conditions, Materials, and Pattern objects. Finally, Layers consist of the Layer object which reference the component property objects and constructs the unit cell that will be simulated.

For this model, our half wavelength dipole will be 0.5m long so that the peak radiation will be at 300MHz.

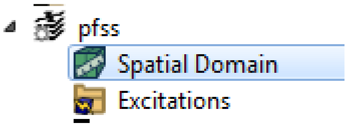
2.4.1. Spatial Domain

The spatial domain objects specifies the global data required for the simulation, including the mesh properties and unit cell dimensions.

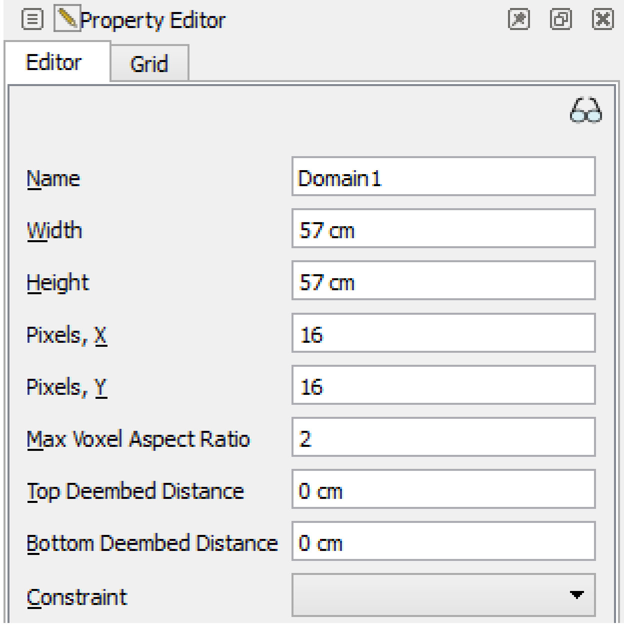
1. Create a new Spatial Domain object by right clicking on Spatial Domain and selecting "New Spatial Domain".



2. Select the "Spatial Domain" object.



3. In the "Properties" dock to the right of GEMSIF, set the following properties:
   a. "Width" - 57cm — This will set the physical width of the unit cell.
   b. "Height" - 57 cm — This will set the physical width of the unit cell.
   c. "Pixels, X" - 16 — Specifies the number of pixels in the unit cell along the x-axis.
   d. "Pixels, Y" - 16 -- Specifies the number of pixels in the unit cell along the y-axis.



Name	Domain1
Width	57 cm
Height	57 cm
Pixels, X	16
Pixels, Y	16
Max Voxel Aspect Ratio	2
Top Deembed Distance	0 cm
Bottom Deembed Distance	0 cm
Constraint	

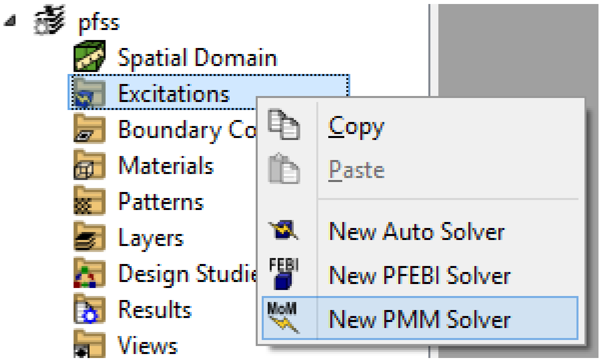
NOTE For this example, we are using the PMM solver, which requires the pixel dimensions to be powers of 2 (i.e. 8, 16, 32, 64).

For a complete list of spatial domain properties, see the Spatial Domain Object in the PFSS Topical Documentation.

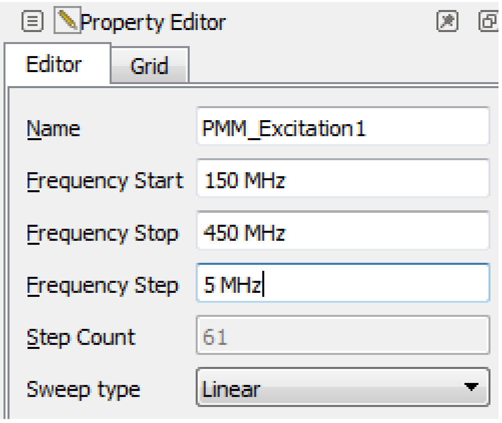
2.4.2. Excitation

Excitations specify the frequency and angular sweep settings of the incident field used by the solver to excite the periodic geometry defined by the layers and patterns in the simulation. There are three solvers to choose from.

- Auto Solver - PFSS will automatically attempt to identify which solver to use. Currently the Auto Solver is biased towards PMM.
- PFEBI Solver - Creates an excitation object that uses the PFEBI Solver (flexible). Primarily used for modeling geometries with nonuniform dielectric layers.
- PMM Solver - Creates an excitation object that uses the PMM Solver (faster). Primarily used for modeling geometries with homogeneous dielectric layers.
  1. Right click on "Excitations" and select "New PMM Solver".



2. Expand "Excitations" and select "PMM_Exication1" and set the following properties:
   a. "Frequency Start" - 150 MHz — Start frequency of the simulation.
   b. "Frequency Stop" - 450 MHz — Stop frequency of the simulation.
   c. "Frequency Step" - 5 MHz — Frequency interval for simulation sweep.



Name: PMM_Excitation1
Frequency Start: 150 MHz
Frequency Stop: 450 MHz
Frequency Step: 5 MHz
Step Count: 61
Sweep type: Linear

For a complete list of excitation properties, see the Excitation Object in the PFSS Topical Documentation.

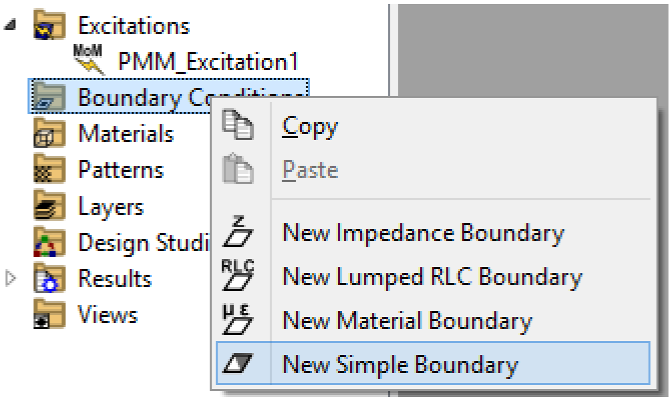
2.4.3. Boundary Conditions

Boundary conditions represent infinitely thin layers used to specify the electromagnetic behavior of surfaces. PFSS provides four types of boundary conditions.

- Impedance Boundary Condition : Specify a complex surface impedance value as as the surface property.
- Material Boundary Condition : Specify the surface impedance based on the complex and (potentially) dispersive permittivity and permeability of an isotropic material.
- Lumped RLC Element Boundary Condition : Specify the surface impedance to be equivalent to a parallel or series combination of lumped resistance, capacitance, and inductance.
- Simple Boundary : Transparent (no boundary condition) or PEC.

The goal of this tutorial is to build a simple approximation model. To do this, only simple boundary conditions will be used.

1. Right Click on Boundary Conditions and select "New Simple Boundary".



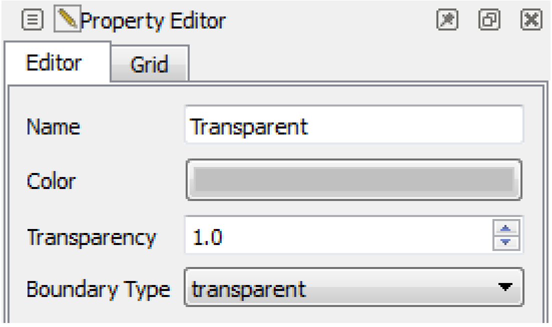
2. Select the new boundary condition SimpleBoundary1 and set the following properties:

   a. "Name" - Transparent

   b. "Color" - Leave as default. — Pops up a color selection menu.

   c. "Transparency" - 1.0 — Level of transparency of the boundary condition when it is visually rendered. Solid: 0, Clear: 1

   d. "Boundary Type" - transparent — transparent provides a transparent boundary condition on which to impose a patterned boundary condition.



Name Transparent

Color

Transparency 1.0

Boundary Type transparent

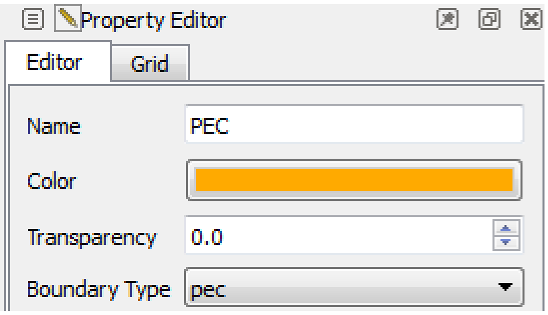
3. Create a second simple boundary condition and set the following properties:

   a. "Name" - PEC

   b. "Color" - Orange, RGB: [255,170,0]

   c. "Transparency" - 0.0

   d. "Boundary Type" - pec — Ideal PEC model.



Name PEC

Color

Transparency 0.0

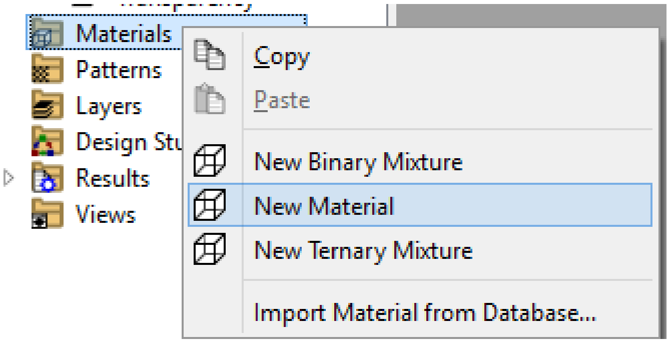
Boundary Type pec

For a complete list of boundary condition properties, see the Boundary Condition Object in the PFSS Topical Documentation.

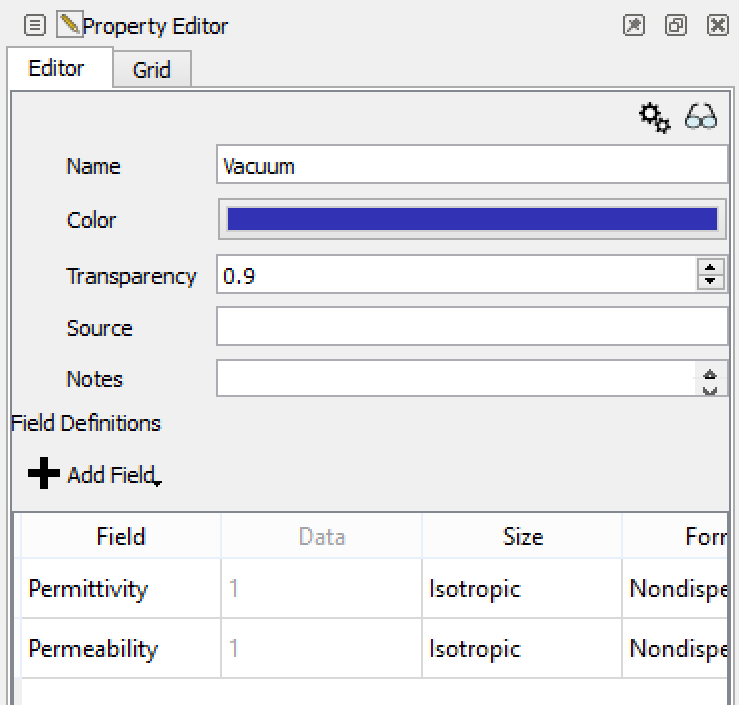
2.4.4. Material

Material objects represent the properties of dielectric or magnetic materials for the PFSS solvers.

1. Right click on Materials and select "New Material"



2. Select the new material Material1 and set the following properties:
   a. "Name" - Vacuum
   b. "Color" - Leave as default.
   c. "Transparency" - Leave as default.
   d. Add the following Field Definitions:
      i. "Isotropic Permittivity" - 1
      ii. "Isotropic Permeability" - 1



Name: Vacuum

Color:

Transparency: 0.9

Source:

Notes:

Field Definitions

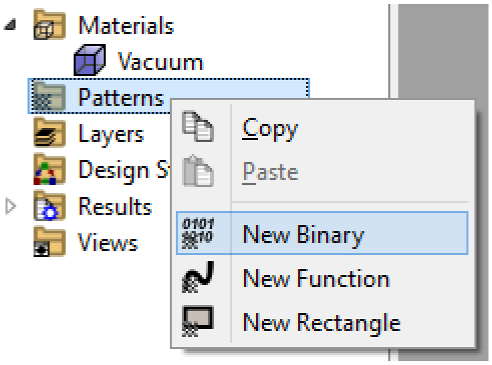
Field	Data	Size	Forr
Permittivity	1	Isotropic	Nondispe
Permeability	1	Isotropic	Nondispe

For a complete list of material properties, see the Material Object in the GEMSIF Topical Documentation.

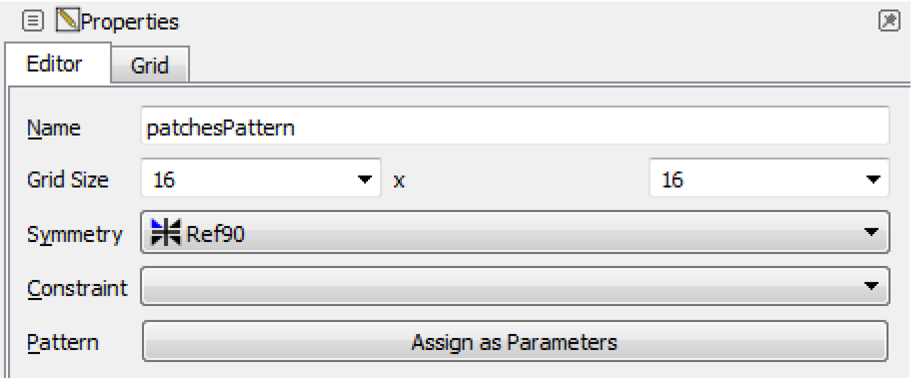
2.4.5. Pattern

A pattern is a rectangular array or image of boolean values that may be either true or false. Each value is associated with a pixel. The pattern determines how a boundary condition or material will be rendered in the model.

1. Right-click "Patterns" and select "New Binary".

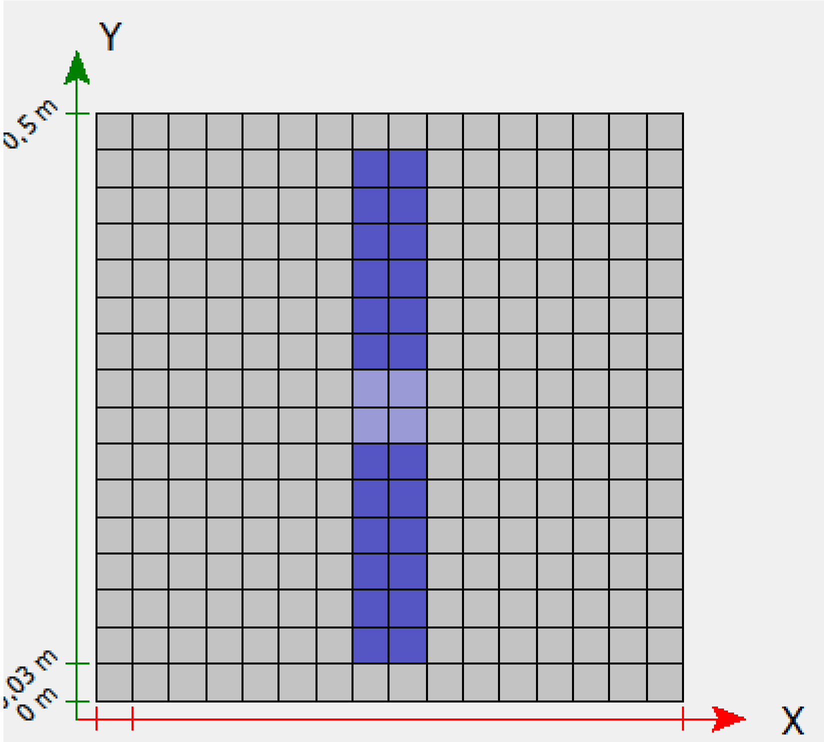


2. Expand "Patterns" and then select the "BinaryPattern1" object to select it.
3. The setting and a pattern editor should appear in the Properties Dock. Set the following properties:
   a. "Name" - patchesPattern
   b. Set the two "Grid Size" values to 16
   c. "Symmetry" - Ref90



Name: patchesPattern
Grid Size: 16 x 16
Symmetry: Ref90
Constraint:
Pattern: Assign as Parameters

4. Click on the pixels (gray squares) in the pattern to form a vertical dipole about the center, leaving the top and bottom row of pixels empty



NOTE

Use the buttons just above and to the right of the pattern editor ▣ ◪ ⬚ to switch between setting pixels, toggling pixels, and clearing pixels. In the default setting pixel mode, left clicking selects a pixel, right clicking deselects a pixel.

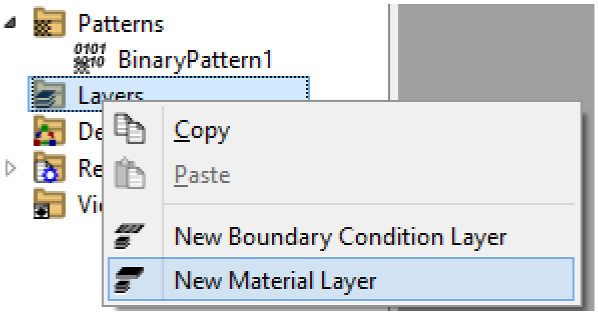
For a complete list of pattern properties, see the Pattern Object in the PFSS Topical Documentation.

2.4.6. Layers

Layers combine general properties and component properties to build a unit cell to simulate.

- Material Layer : Patterned or unpatterned dielectric layer of finite thickness.
- Boundary Layer : Patterned or unpatterned boundary condition (PEC, Impedance, lumped element).

1. Right-click "Layers" and click "New Material Layer." Expand "Layers" and select the new "MaterialLayer1".



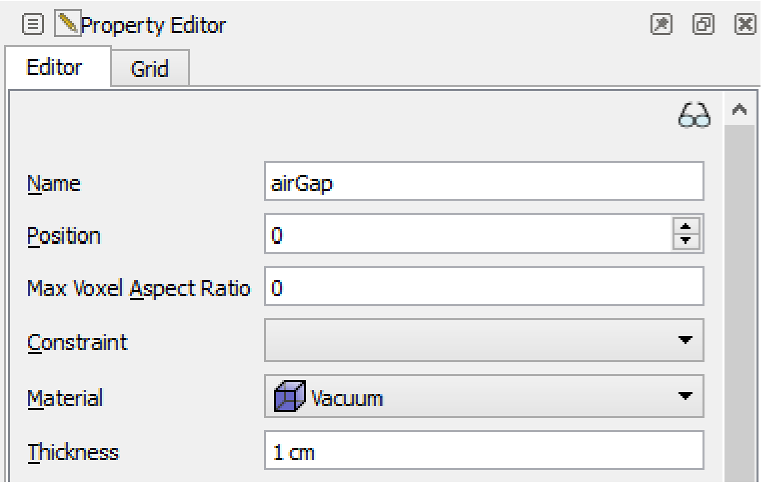
2. Set the following properties:

a. "Name" - airGap

b. "Position" - 0

c. "Material" - Vacuum

d. "Thickness" - 1 cm



Name: airGap
Position: 0
Max Voxel Aspect Ratio: 0
Constraint:
Material: Vacuum
Thickness: 1 cm

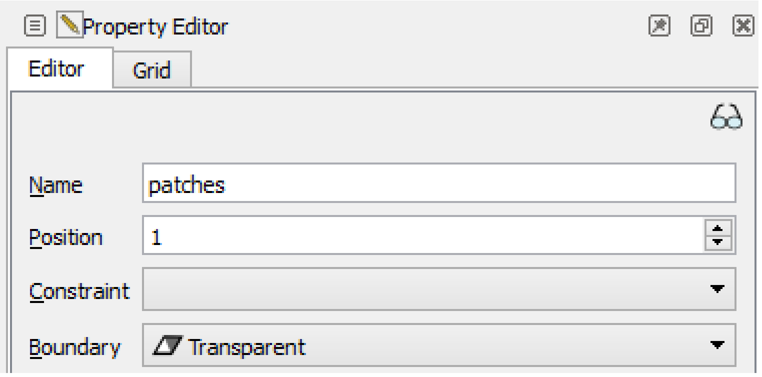
3. Use the above steps to create a "New Boundary Condition Layer".

4. Select "BoundaryLayer1" and set the following properties:

   a. "Name" - patches

   b. "Position" - 1

   c. "Boundary" - None



Name: patches
Position: 1
Constraint:
Boundary: Transparent

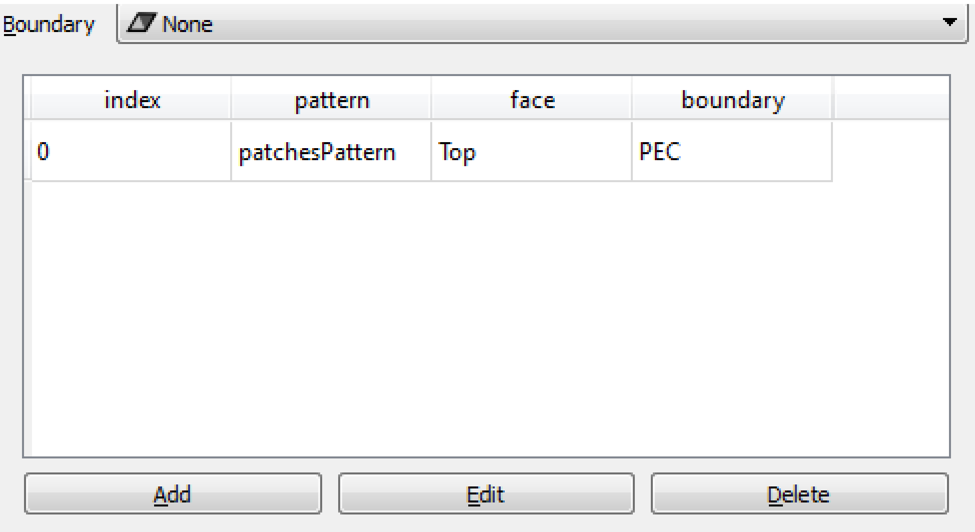
5. Toward the bottom of the "Editor", click "Add."

6. In the new entry in the list, set the following columns:

   a. "index" - 0

   b. "pattern" - patchesPattern

   c. "boundary" - pec



Boundary: None

index	pattern	face	boundary
0	patchesPattern	Top	PEC

NOTE

In some cases adding a pattern to a boundary layer will cause the layer's initial boundary to reset to PEC. Make sure it is still set to None.

NOTE

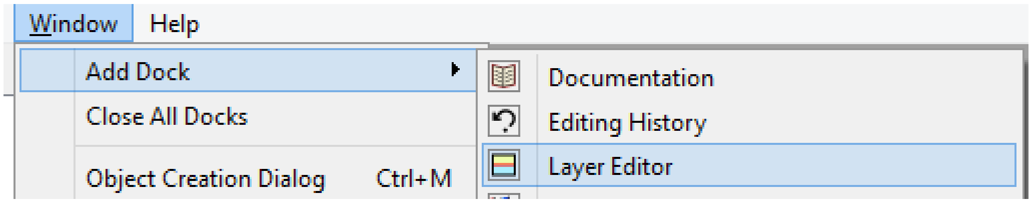
GEMSIF PFSS automatically adds a vacuum layer if the top layer is a boundary condition. The point at which GEMSIF calculates the frequency response can be controlled by changing the deembed distance under "Domain."

For a complete list of layer properties, see the Layer Object in the PFSS Topical Documentation.

2.5. Layer Editor Dock

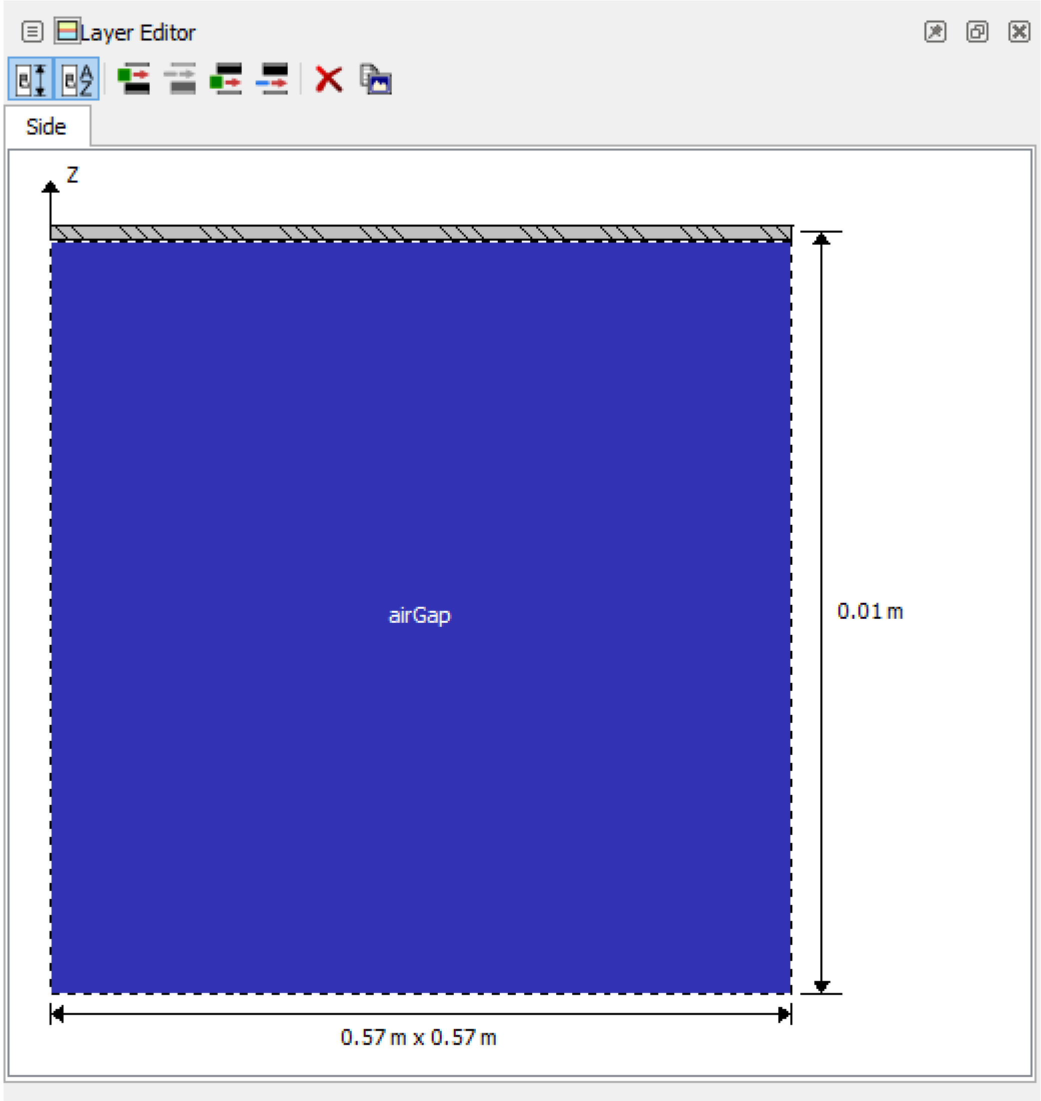
The Layer Editor provides a 2D side view of the unit cell. This allows the user to see the location of each layer in the cell and add or remove layers. Patterns are not shown in this view and must be added and edited from each layer object.

1. To add the the Layer Editor, select "Window" from the file menu and then "Add Dock"→"Layer Editor"



NOTE

The Layer Editor will be appear at the bottom right corner of GEMSIF. The dock can be moved and resized to fit the user's need.



2.5.1. Layer Editor Controls

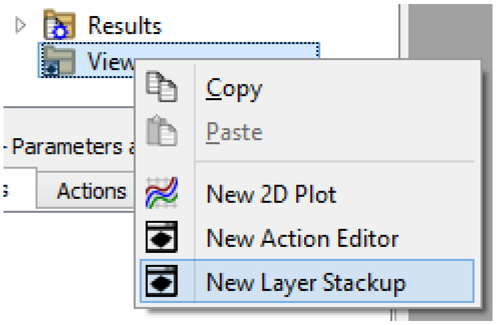
- : Toggle dimension lines.
- : Toggle in-layer text.
- : Adds a material layer above the selected layer.
- : Adds a boundary layer above the selected layer.
- : Adds a material layer below the selected layer.
- : Adds a boundary layer below the selected layer.
- : Deletes the selected layer.
- : Saves a snapshot of the editor to the clipboard.

NOTE Right clicking on a layer will also provide the same options as the controls.

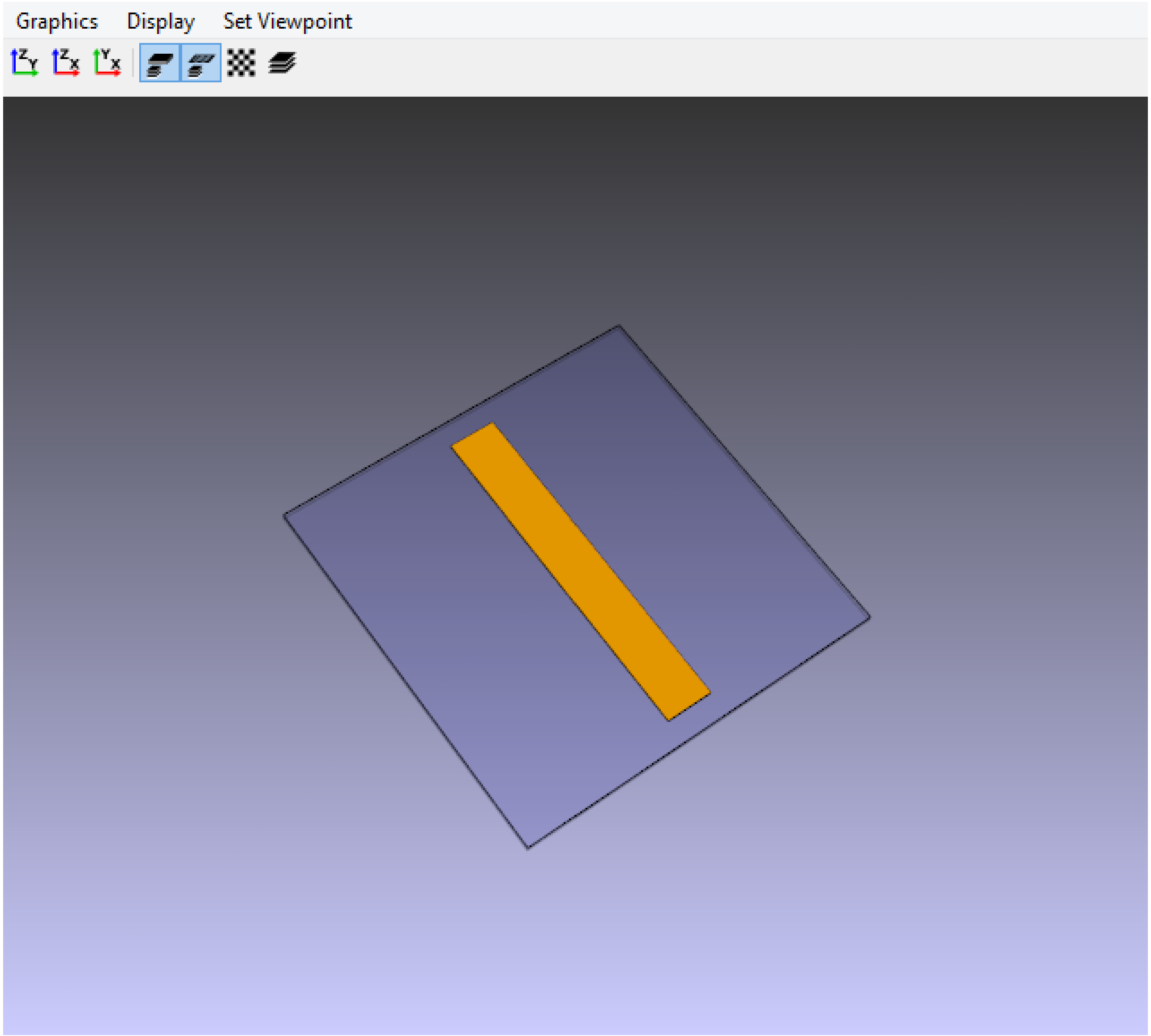
2.6. Layer Stackup View

PFSS provides a Layer Stackup View to view a 3D rendering of the unit cell.

1. Right click on "Views" at the bottom of the Hierarchy Tree and select New Layer Stackup.



2. Clicking on the view and holding will allow you to rotate the model.

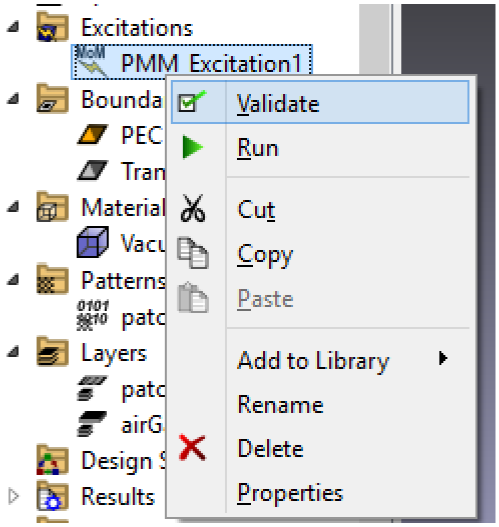


Selecting a layer in the Stackup View will load the editor for the layer in the Property Editor Dock for quick editing.

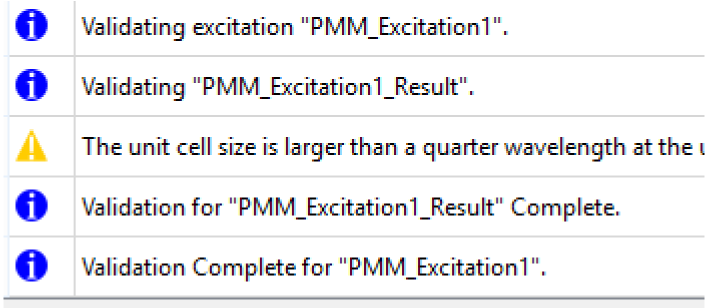
Using the menu options, the view can be saved as an image or exported to the clipboard. Axes and dimensions can also be added.

2.6.1. Validating and Running

1. After building a simulation, it needs to be saved before it can be run. This is required to link the excitation object to the results file. For future changes, saving is not required, but it is recommended.
2. Next, the simulation should be verified. This is a useful check to ensure that the simulation is free from errors. While not required, validating is recommended.
   a. Right click on "PMM_Excitation1" and select "Validate". If there are multiple excitations, they can be validated at the same time by right clicking on the simulation object (the name) and selecting "Validate all Excitations."



   b. A properly validated simulation will provide no errors in the "Status" dock.



Validating excitation "PMM_Excitation1".

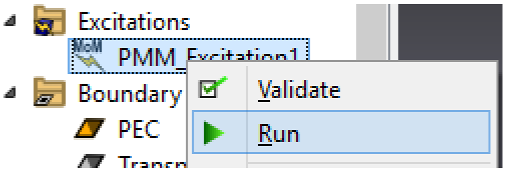
Validating "PMM_Excitation1_Result".

The unit cell size is larger than a quarter wavelength at the

Validation for "PMM_Excitation1_Result" Complete.

Validation Complete for "PMM_Excitation1".

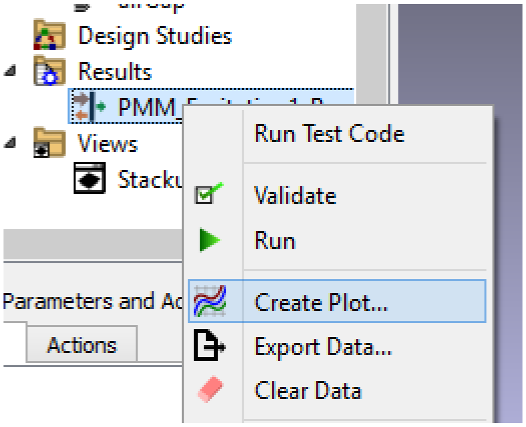
3. Now that the simulation has been validated, it can now be run. For later changes to the simulation, it is not necessary to validate the simulation unless you want to check for errors.
   a. Right click on "PMM_Excitation1" and select "Run."



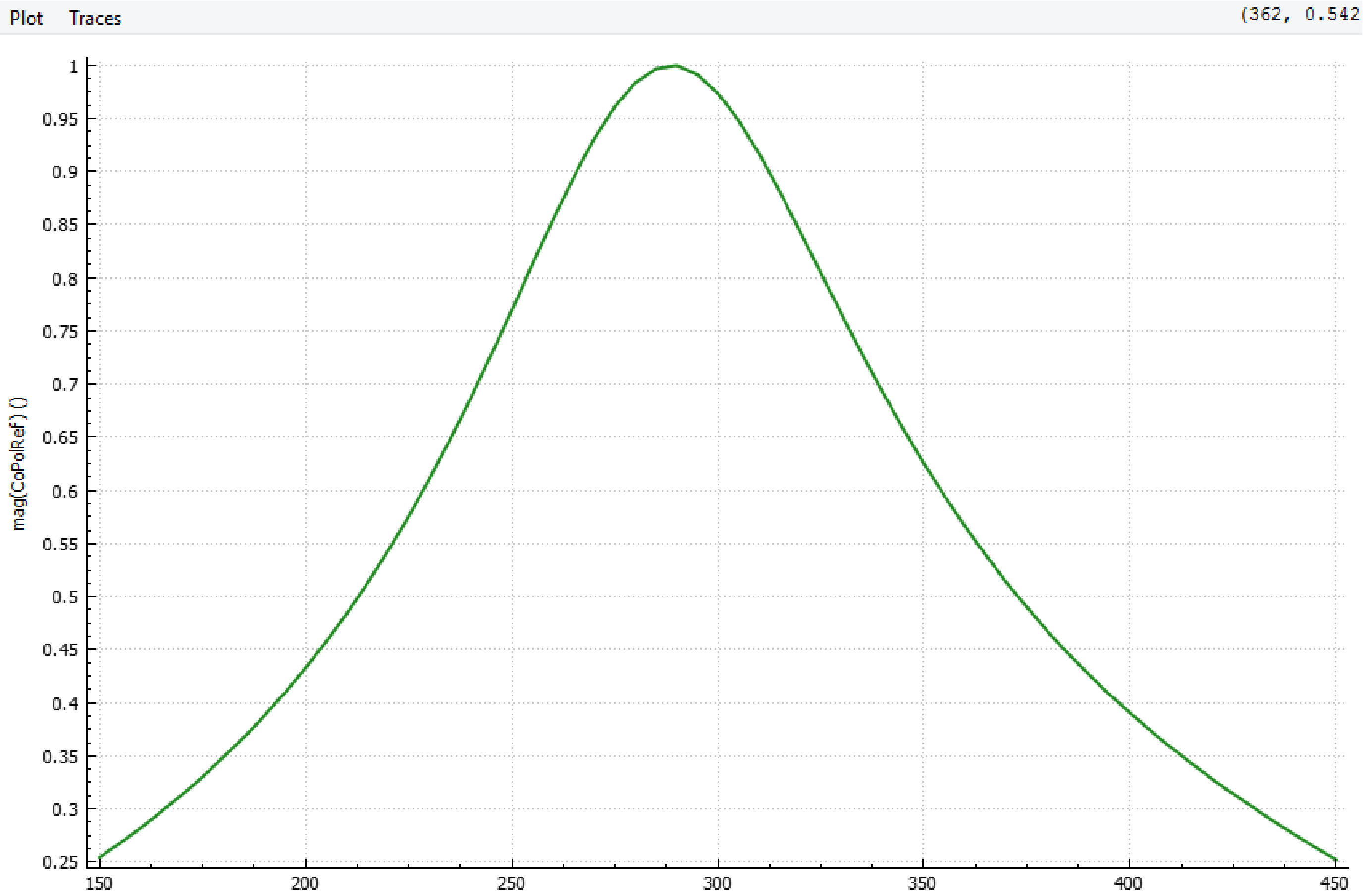
2.7. Plotting

After running the simulation, the simulation results will be available to the user.

1. Select "Results" and expand it.
2. Right Click on "PMM_Exicitation1_Result" and select "Create Plot."



3. Select the desired data and GEMSIF will provide a plot view to display the results.



3. Solvers

Two solvers are included in the PFSS package, each specialized for a particular design problem. PMM is based on the method of moments formulation, and so has limited support for complex, inhomogeneous material parameters but is extremely rapid to execute. In comparison, the PFEBI solver based on a hybridized finite element-boundary integral solver has fewer geometrical limitations and supports nearly arbitrary designs specified on the voxel grid, but is generally slower than the PMM tool. The PFSS tool will by default automatically select the best solver for a given model depending on the specified geometry and other model requirements, but the user can request a specific solver, as well.

3.1. PMM

The Periodic Method of Moments (PMM) is an FFT-accelerated method of moments solver specialized to periodic, multilayered dielectric structures separated by patterned layers of metallic or surface-impedance boundary conditions.

PMM uses the spectral-domain Method of Moments accelerated with the fast Fourier transform (FFT) to analyze the problem of scattering from a frequency selective surface (FSS). The solver computes the electric current distribution on a single unit cell of a doubly-periodic structure in order to determine the resulting scattered fields. The FSS is printed on a dielectric substrate and covered by an additional superstrate layer. This configuration can be optionally backed by a PEC ground plane. In addition, multiple such screens may be cascaded to form a multilayer metallo-dielectric stack. The superstrate and substrate can be either lossy or lossless materials. The metallic FSS screen may be modeled as a perfectly-electrically conducting screen, or as possessing a specified surface impedance. The reflection and transmission coefficients may be calculated for incident fields for both polarizations (TE and TM cases).

3.1.1. Solver Implementation and Limitations

The solver and its outputs have been thoroughly verified and validated against analytical solutions and the outputs of other simulation tools; please see the papers cited in the PMM References. A description of the mathematical underpinnings and limitations of the solver are listed below.

- Only a single unit cell of the periodic structure must be discretized.
- The number of pixels in the x- and y-directions must be integer powers of 2 (8, 16, 32, etc.)
- The discretization of the unit cell of the FSS should approximately satisfy the following requirements: $\max(\Delta x, \Delta y) < \lambda/10$.
- Isolated pixels or pixels that are only connected via a diagonal are not allowed in PMM. The MoM basis functions extend over two adjacent pixels with the same boundary conditions, so an isolated pixel (or an isolated hole) will not produce accurate results. Note that the PFEBI solver is not subject to this limitation. Also, setting the oversample factor of the excitation to any value greater than unity will also resolve the issue.

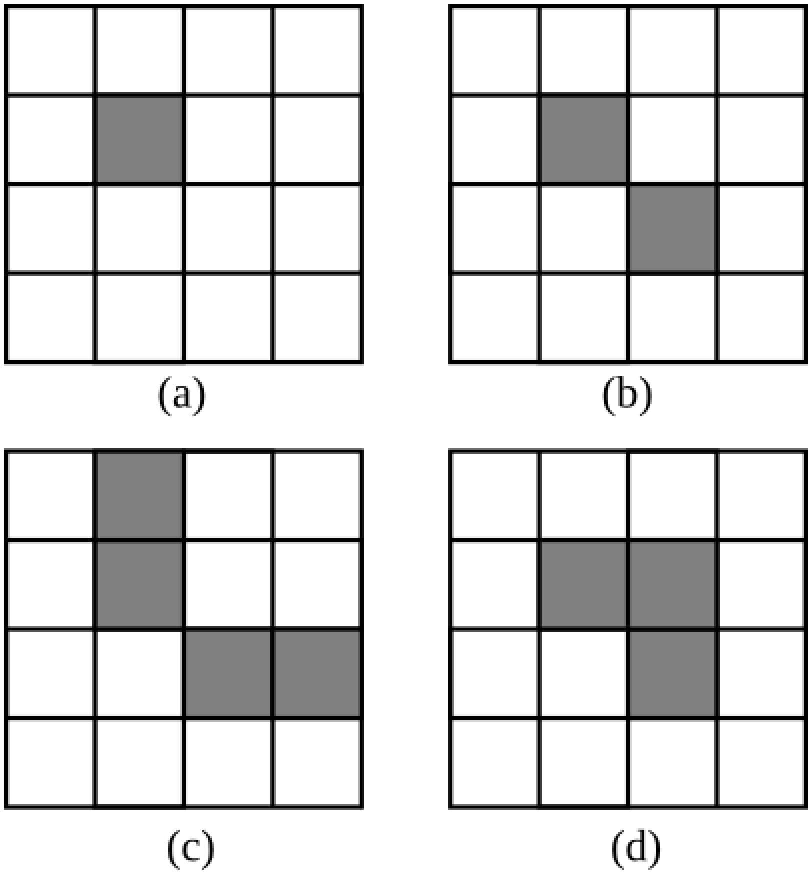


Figure 8. The pixels in (a) and (b) are isolated, and will cause computation problems. The pixels in (c) and (d), although there are diagonal connections, are connected, and so will generate accurate results.

- The minimum allowable incident wave theta angle is 1 degree. Angles less than 1 degree create numerical convergence and tolerance problems in the solver.
- A detailed discussion of the spectral-domain PMM can be found in [1, 2] of PMM References for the analysis of doubly periodic FSS structures (e.g., an FSS with both a dielectric substrate and a superstrate or an FSS with a dielectric substrate only.
- By expanding the surface electric current within a unit cell in terms of roof-top basis functions and then employing Galerkin's procedure to the appropriate boundary integral equation (here, the electric field integral equation is used), the following linear matrix equation is obtained:

$$V_i = C_0 - C_3 \frac{C_1 \cos(\theta_i + C_3)}{C_4 + C_1 \cos(\theta_i + C_2)}$$

- The first FFT-based method is applied here for fast computation of the impedance matrix elements for the uniform discretization of the FSS unit cell [3].
- In order to obtain the electric current distribution on the FFS screens, two solvers are available to solve the matrix equation (1) for calculation of the scattering parameters. (a) The first solver is the direct LU decomposition method. (b) The second method is the BCG (Bi-conjugate gradient) iterative solver combined with an O(N log N) FFT-based acceleration technique [4]. The second FFT accelerated technique was included to speed-up the computation of matrix-vector products within the iterative solver. This is possible due to the asymmetric multilevel block-Toeplitz structure of the impedance-matrix that results from the uniform discretization of the FSS unit cell.

- Upon solving the matrix equation (1), the reflection and transmission coefficients are calculated based on the zeroth order scattered harmonic.
- This program is developed based on the code written by Joseph D. Vacchione.

3.1.2. References

The papers referenced below include background material [1-4] on the algorithms used by the PMM tool, as well as references on the development, testing, validation, and usage for example designs of the PMM solver [5-10].

1. L. Li, D. H. Werner, J. A. Bossard, and T. S. Mayer, “A Model-based parameter estimation technique for wideband interpolation of periodic moment method impedance matrices with application to genetic algorithm optimization of frequency selective surfaces,” IEEE Trans. Antennas Propagat., vol.54, no.3, pp.908–924, 2006.
2. C. H. Chan, “Analysis of frequency selective surfaces,” in Frequency Selective Surface and Grid Array, T. K. Wu, Eds., Chap. 2, New York: Wiley, 1995.
3. J. D. Vacchione, “Techniques for analyzing planar, periodic, frequency selective surface systems,” Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 1990.
4. B. E. Barrowes, F. L. Teixeira, and J. A. Kong, “Fast algorithm for matrix-vector multiply of asymmetric multilevel block-Toeplitz matrices in 3-D scattering,” Microwave Opt. Technol. Lett., vol. 31, no. 1, pp. 28–32, 2001.
5. Xiande Wang and Douglas H. Werner, “Improved model-based parameter estimation approach for accelerated periodic method of moments solutions with application to the analysis of convoluted frequency selected surfaces and metamaterials,” IEEE Transactions on Antennas and Propagation, Vol. 58, No. 1, pp. 122-131, January 2010.
6. Z. Bayraktar, M. D. Gregory, X. Wang, and D. H. Werner, "Matched Impedance Thin Planar Composite Magneto-dielectric Metasurfaces," IEEE Transactions on Antennas and Propagation, Vol. 60, No. 4, pp. 1910 - 1920, April 2012
7. Z. Bayraktar, X. Wang, and D. H. Werner, "Thin Composite Matched Impedance Magneto-Dielectric Metamaterial Absorbers," Proceedings of the 2010 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, Toronto, Canada, July 11-17, 2010.
8. X. Wang and D. H. Werner, "Improved Model-Based Parameter Estimation Approach for Accelerated Periodic Method of Moments Solutions with Application to the Analysis of Convoluted Frequency Selective Surfaces and Metamaterials," IEEE Transactions on Antennas and Propagation, Vol. 58, Issue 1, pp. 122-131, January 2010.
9. X. Wang and D. H. Werner, "Multiband Ultra-Thin Electromagnetic Band-Gap and Double-Sided Wideband Absorbers Based on Resistive Frequency Selective Surfaces," Proceedings of the 2009 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, Charleston, SC, USA, June 1-5, 2009.
10. X. Wang and D. H. Werner, "Accelerated PMM Solutions via Model-Based Parameter Estimation and Fast Fourier Transform Techniques," Proceedings of the 2008 IEEE AP-S International Symposium

and USNC/URSI National Radio Science Meeting in San Diego, CA, USA, July 5-11, 2008

3.2. PFEBI

The Periodic Finite-Element Boundary-Integral (PFEBI) solver is more general than the PMM solver, as it allows inhomogeneous material layers and boundary conditions to be applied to any face of a voxel. The PFEBI solver is made available under license from the Penn State Research Foundation (PSRF).

The PFEBI solver analyses the scattering or radiation from a double-periodic structure. This periodic structure may be a Frequency Selective Surface (FSS), metamaterial, metasurface, or infinite antenna array. In both cases, the structure is composed of (1) one or more homogeneous dielectric slabs (isotropic, anisotropic, or bi-anisotropic), (2) one or more inhomogeneous dielectric slabs, and (3) one or more metal or surface impedance boundary condition screens. Dielectric media may be lossy, lossless, and/or dispersive. Active (gain) materials are not allowed. Isotropic metallic materials may also be modelled. Metal screens may be modelled as PEC or as a potentially lossy surface impedance.

Scattering parameters are computed for FSS problems and the input impedance and radiation (Gain) pattern are calculated for antenna problems.

3.2.1. Solver Implementation and Limitations

The PFEBI solver uses the hybrid finite-element boundary-integral (FEBI) method to numerically compute the electric field inside and on the surface of the structure. The solver and its outputs have been thoroughly verified and validated against analytical solutions and the outputs of other simulation tools; please see the papers cited in the PFEBI References. A description of the mathematical underpinnings and limitations of the solver are listed below.

- Only a single unit cell of the doubly-periodic structure is modelled.
- The number of pixels in the x and y directions may be any positive integer, including 1, and may be different. Geometries that are periodic along one axis, but infinite along the other may be specified by only a single pixel in the infinite direction.
- The discretization of the unit cell should approximately satisfy the following requirements: a) max(Δx, Δy, Δz) < λε/10, and b) Δx/ Δy/ Δz ~= 1.0 That is, each voxel dimension should be smaller than a tenth of a wavelength in the highest-index material in the simulation, and the voxels should be approximately cubic.
- PFEBI uses brick elements for finite-element volumetric discretization and uses roof-top function for top and bottom surface discretization. Galerkin’s method is used for testing in the boundary-integral calculation. In order to increase computation efficiency, symmetries in calculating the entries of the impedance matrix in boundary-integral part (only for normal incidenec case) and in finite-element part are fully exploited in the programming. This requires that all the elements are of the same dimensions.
- The BCG-FFT (Fast Fourier Transform) algorithm has been included in the FEBI-code to speed-up the computation of matrix-vector for BI part due to a BI sub-matrix with a Toeplitz-Block Toeplitz

structure.

- For the boundary-integral part, the conventional free-space Green's function becomes an infinite double summation (periodic Green's function), and the Ewald transformation technique is used to accelerate the convergence of the double summation. The Ewald transformation converts the original periodic Green's functions into two parts, one in spatial domain and one in spectral domain, both of which converges at a much faster speed.
- PFEBI uses the Mixed Potential Integral Equation (MPIE) and solves for the total electric field inside the structure and the tangential electric on the surface of the structure. For Isotropic materials, the pertinent MPIE formulation is given by: $V_i = C_0 - C_3 \frac{C_1 \cos(\theta_i + C_3)}{C_4 + C_1 \cos(\theta_i + C_2)}$

For the general case of inhomogeneous and lossy bi-anisotropic materials characterized by the following constitutive relations:

The pertinent MPIE formulation is given as follows:

$$V_i = C_0 - C_3 \frac{C_1 \cos(\theta_i + C_3)}{C_4 + C_1 \cos(\theta_i + C_2)}$$

- For FSS scattering problem, the reflection and transmission coefficients are calculated by performing the FFT of the tangential electric field on the top and bottom surface of the unit cell. Generally speaking, the FFT computes the scattering matrix of the periodic structure, with the center entry being the reflection/transmission coefficients (for oblique incidence, phase compensation is required before performing the FFT). Although the FFT is most efficient when the number of pixels in the x and y directions are integer powers of 2, the PFEBI solver supports any pixel dimensions, including 1 x 1.

3.2.2. References

The papers referenced below include background material [1-4] on the algorithms used by the PFEBI tool, as well as references on the development, testing, validation, and usage for example designs of the PFEBI solver [5-27].

1. J. L. Volakis, A. Chatterjee, and L. C. Kempel, Finite Element Method for Electromagnetics, IEEE Press, 1998.
2. T. F. Eibert, J. L. Volakis, D. R. Wilton, and D. R. Jackson, "Hybrid FE/BI modeling of 3-D doubly periodic structures utilizing triangular prismatic elements and an MPIE formulation accelerated by the Ewald transformation", IEEE Trans. Antennas and Propag., vol. 47, no. 5, pp. 843-850, May 1999.
3. J. Jin, The Finite Element Method in Electromagnetics, John Wiley, 2002.
4. F. Bilotti, L. Vegni, and A. Toscano, "Radiation and scattering feature of patch antennas with bianisotropic substrates,"
5. Xiande Wang, Douglas H. Werner and Jeremiah P. Turpin, "Application of AIM and MBPE techniques to accelerate modeling of 3-D doubly periodic structures with non-orthogonal lattices composed of bianisotropic media," Accepted for publication in IEEE Transactions on Antennas and Propagation, Vol. 62, No. 8, Aug. 2014. (triangular-based)

6. J. Ashbach, X. Wang, and D. H. Werner, "The Finite Element Boundary Integral Method Accelerated Using a Graphics Processing Unit," Proceedings of the 2013 IEEE International Symposium on Antennas & Propagation and USNC/URSI National Radio Science Meeting, July 7-13, 2013, Orlando, FL, USA. (triangular-based).
7. J. Ashbach, J. A. Bossard, X. Wang, and D. H. Werner, "Metamaterial Absorber for the Near-IR with Curvilinear Geometry based on Beziér Surfaces," Proceedings of the 2013 IEEE International Symposium on Antennas & Propagation and USNC/URSI National Radio Science Meeting, July 7-13, 2013, Orlando, FL, USA. (triangular-based)
8. Q. Wu, C. P. Scarborough, B. G. Martin, R. K. Shaw, D. H. Werner, E. Lier, and X. Wang, "A Ku-band Dual Polarization Hybrid-Mode Horn Antenna Enabled by Printed-Circuit-Board Metasurfaces," IEEE Transactions on Antennas and Propagation, Vol. 61, No. 3, pp. 1089-1098, Mar. 2013.
9. Q. Wu, C. P. Scarborough, D. H. Werner, E. Lier, and X. Wang, "Design Synthesis of Metasurfaces for Broadband Hybrid-mode Horn Antennas with Enhanced Radiation Pattern and Polarization Characteristics," IEEE Transactions on Antennas and Propagation, Vol. 60, No. 8, pp. 3594-3604, Aug. 2012.
10. Z. Bayraktar, J. A. Bossard, X. Wang, and D. H. Werner, "A Real-valued Parallel Clonal Selection Algorithm and Its Application to the Design Optimization of Multi-layered Frequency Selective Surfaces," IEEE Transactions on Antennas and Propagation, Vol. 60, No. 4, pp. 1831 - 1843, April 2012.
11. Z. Bayraktar, M. D. Gregory, X. Wang, and D. H. Werner, "A Versatile Design Strategy for Thin Composite Planar Double-Sided High-Impedance Surfaces," IEEE Transactions on Antennas and Propagation, Vol. 60, No. 6, pp. 2770-2780, June 2012.
12. Q. Hao, Y. Zeng, B. K. Juluri, X. Wang, B. Kiraly, I-K. Chiang, L. Jensen, D. H. Werner, V. H. Crespi and T. J. Huang, "Metallic Membranes with Subwavelength Complementary Patterns: Distinct Substrates for Surface-Enhanced Raman Scattering," ACS Nano, Vol. 5, No. 7, pp. 5472-5477, July 2011.
13. M. Gregory, X. Wang, and D. H. Werner, "Flexible Design of Doubly Periodic Frequency Selective Surfaces with a Prismatic Mesh Based FEBI Simulation Tool and CMA-ES," Proceedings of the 2011 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, Spokane, WA, USA, July 3-8, 2011.
14. Y. Zeng, X. Wang, D. H. Werner, Q. Hao, and T. Huang, "Linear Optical and Surface-enhanced Raman Scattering Study on Metallic Membranes with Subwavelength Complementary Patterns," Proceedings of the 2011 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, Spokane, WA, USA, July 3-8, 2011.
15. Z. Jiang, J. A. Bossard, X. Wang, and D. H. Werner, "Synthesizing Metamaterials with Angularly Independent Effective Medium Properties Based on an Anisotropic Parameter Retrieval Technique Coupled with a Genetic Algorithm," Journal of Applied Physics, Vol. 109, pp. 013515/1-11, 2011.
16. Q. Hao, Y. Zeng, X. Wang, Y. Zhao, B. Wang, D. H. Werner, V. H. Crespi, and T. J. Huang, "Characterization of Complementary Patterned Metallic Membranes Produced Simultaneously by a Dual Fabrication Process," Applied Physics Letters, Vol. 97, 193101, 2010.
17. D. H. Werner, Z. H.. Jiang, F. Toor, S. Yun, Z. Bayraktar, X. Wang, T. Mayer, and P. L. Werner,

"Metamateria Absorbers with Wide-Angle Polarization-Insensitive Multiband Properties for RF through Mid-Infrared Applications," the Fourth International Congress on Advanced Electromagnetic Materials in Microwaves and Optics, Karlsruhe, Germany, September 12-16, 2010.

18. X. Wang and D. H. Werner, "Fast Analysis of 3-D Doubly Periodic Structures with Complex Geometry and Anisotropic Materials using the Adaptive Integral Method," Proceedings of the 2010 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, Toronto, Canada, July 11-17, 2010.
19. Z. H. Jiang, Q. Wu, X. Wang, and D. H. Werner, "Flexible Wide-Angle Polarization-Insensitive Mid-Infrared Metamaterial Absorbers," Proceedings of the 2010 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, Toronto, Canada, July 11-17, 2010.
20. Z. H. Jiang, J. A. Bossard, X. Wang, and D. H. Werner, "Genetic Algorithm Synthesis of Impedance-Matched Infrared ZIMs with Wide FOV Using a Generalized Inversion Algorithm," Proceedings of the 2010 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, Toronto, Canada, July 11-17, 2010
21. Z. Bayraktar, J. A. Bossard, X. Wang, and D. H. Werner, "Real-Valued Parallel Clonal Selection Algorithm for Design Optimization in Electromagnetics," Proceedings of the 2010 IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, Toronto, Canada, July 11-17, 2010.
22. X. Wang, D. H. Werner, and D. -H. Kwon, "Acceleration of Periodic FEBI Simulations for General Bi-anisotropic Media Using a Model-Based Parameter Estimation Technique," Proceedings of the 2008 IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting in San Diego, CA, USA, July 5-11, 2008.
23. X. Wang, D. -H. Kwon, D. H. Werner, and I. -C. Khoo, "Anisotropic Liquid Crystals for Tunable Optical Negative-Index Metamaterials," Proceedings of the 2008 IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting in San Diego, CA, USA, July 5-11, 2008.
24. D.-H. Kwon, X. Wang, Z. Bayraktar, B. Weiner, and D. H. Werner, "Reconfigurable transmissive/Reflective Metamaterial Slab in the Near IR," Proceedings of the 2008 IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting in San Diego, CA, USA, July 5-11, 2008.
25. D.-H. Kwon, X. Wang, Z. Bayraktar, B. Weiner, and D. H. Werner, "Near-Infrafed Metamaterial Films with Reconfigurable Transmissive/Reflective Properties," Optics Letters, Vol. 33, No. 6, pp. 545-547, March 15, 2008.
26. X. Wang, D.H. Kwon, D. H. Werner, I.-C. Khoo, Alexander V. Kildishev and Vladimir M. Shalaev, "Tunable Optical Negative-Index Metamaterials Employing Anisotropic Liquid Crystals," Applied Physics Letters, 91, 143122, 2007.
27. M. G. Bray, X. Wang, and D. H. Werner, "Dispersive Chiral Frequency Selective Surfaces Simulated via FEBI and FDTD," Proceedings of the 2007 IEEE Antennas and Propagation Society International Symposium, Honolulu, Hawaii, USA, pp. 2538-2541, June 10-15, 2007.

License for GEMSIF

1. Software Licensing

1.1. E x H Software

GEMSIF is provided to users under the terms of the license agreement signed or agreed to when the software was purchased or downloaded. The license agreement with E x H pertains to the software, documentation, images of the software, and E x H-provided solver plugins. The license does not apply to any outputs of the program, and saved files, or simulation results, which are owned solely by the user of the software. GEMSIF makes use of third-party libraries, each of which is provided under the terms of its own license. These licenses do not affect the use of GEMSIF tool or its outputs, but solely affect the rights of E x H to use and distribute the third-party libraries as part of a proprietary software product.

1.2. Third-Party Libraries and Acknowledgements

GEMSIF makes use of a number of third-party open-source libraries and tools during development, testing, and distribution. These libraries are listed below. This acknowledgement is considered to be the notification of license rights of users of the GEMSIF software due to the inclusion of these open-source tools. E x H, Inc. does not claim ownership or rights other than those granted by the provided licenses to any of these third-party software tools. No warranty or guarantee of usefulness or correctness is given by the owners of any of these licensed tools.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED 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 OWNERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Open-Source Third-party Libraries and Dependencies

These libraries are built into the GEMSIF binaries or included as standalone shared libraries. The source code for any of the following third-party library whose license requires the source code to be freely distributed (i.e., LGPL) may be obtained on request to E x H or from the specific project websites. The use of each of these libraries are governed by, and the libraries themselves distributed according to the terms of their individual licensing agreements, as indicated in the following table.

Software Library	License	Description	Website	Ownership or Copyright Information
Armadillo 5.100	Mozilla Public Library 2.0	C++ Linear Algebra Library	website	Copyright © NICTA and Contributors
BLAS 3.5.0	Modified BSD	BLAS Basic Linear Algebra Subprograms	website	Copyright © 1992-2015 Contributors
Boost 1.55.0	Boost Software License	Peer-reviewed portable C++ source libraries	website	
Ceres Solver	New BSD	A fast non-linear least squares minimizer.	website	Copyright 2016 Google Inc. All rights reserved.
CMAES	LGPLv3 License	CMA-ES for non-linear function minimization	website	Copyright (c) 1996, 2003, 2007, 2011 Nikolaus Hansen, Alexander Fabisch
CXSparse	LGPLv2.1	Sparse matrix library, used by the Ceres library.	website	Copyright (c) 2006-2012, Timothy A. Davis.
Date	MIT	Extensions to the standard C++ datetime/chrono utilities.	website	Copyright (c) 2015-2018, Howard Hinnant and Contributors
Eigen	Various licenses, primarily MPL2, some features LGPLv2.1, BSD, or BSD-like	Linear algebra library, used by the Ceres library.	website	Copyright © Benoît Jacob, Gaël Guennebaud, and Contributors
FastDelegate 1.5	The Code Project Open License	C++ Type-safe Delegate Library	website	Copyright © 2005, Don Clugston
Google Breakpad	BSD	Cross-platform crash recovery library	website	Copyright © 2006, Google Inc. All rights reserved.

Software Library	License	Description	Website	Ownership or Copyright Information
Google gflags	BSD-like	Commandline flag processing library, used by the Ceres library.	website	Copyright (c) 2006, Google Inc. All rights reserved.
Google glog	BSD-like	Logging library, used by the Ceres library.	website	Copyright (c) 2008, Google Inc. All rights reserved.
Google V8	BSD	High-performance C++ Javascript engine.	website	Copyright © 2008, Google Inc. All rights reserved.
HDF5	BSD-like HDF5 License	Hierarchical Document Format for Scientific Datasets	website	HDF5 (Hierarchical Data Format 5) Software Library and Utilities Copyright 2006-2013 by The HDF Group. NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities Copyright 1998-2006 by the Board of Trustees of the University of Illinois.
Netlib Hull	BSD-like License	Convex Hull Calculations	website	Copyright (c) 1995 AT&T
Iconic	Creative Commons Attribution-ShareAlike 3.0 Unported License	Open-source Icon Library	website	

Software Library	License	Description	Website	Ownership or Copyright Information
JSEdit	BSD	Qt Javascript Editor Widget	website Copyright © 2011 Ariya Hidayat	
JSONCpp	MIT	JSON parsing and generation library in C++	website	
LAPACK 3.5.0	Modified BSD	LAPACK linear algebra library	website	Copyright © 1992-2015 Contributors
libb64 1.2	Public Domain	ANSI C Base-64 Encoding and Decoding	website	Dedicated to Public Domain by Chris Venter
libExecStream	BSD	Lightweight cross-platform process execution library	website	
libsigc++	LGPL v3	Type-safe callback system for C++	website	Copyright 2002, The libsigc++ Development Team
matplotlib colormaps	Creative Commons CC0	Perceptually-uniform colormaps for more "honest" plotting.	website	Copyright waived 2015, Nathaniel J. Smith, Stefan van der Walt, and Eric Firing
Nullsoft Scriptable Install System	zlib/libpng license and others	Generator for Windows installers	website	Copyright © 1999-2014 Contributors
NLOpt 2.4.2	MIT and LGPL	The NLOpt nonlinear-optimization package	website	Copyright © 2007-2010 Massachusetts Institute of Technology and Contributors

Software Library	**License**	**Description**	**Website**	**Ownership or Copyright Information**
OpenBLAS 0.2.14	BSD	Optimized BLAS Basic Linear Algebra Subprograms	website	Copyright © Zhang Xianyi and contributors
Open CASCADE Technology (OCCT)	LGPL v2.1 with exception	Constructive geometry library	website	Copyright © OPEN CASCADE SAS
OpenSSL	OpenSSL License (BSD-like)	Encryption and Digital Signature Library	website	Copyright © 1995-1998 Eric Young, 1998-2011 The OpenSSL Project
optionparser 1.3	MIT License	The Lean Mean C++ Option Parser	website	Copyright © 2012 Matthias S. Benkmann
polypartition	MIT License	Tiny Polygon Partitioning and Triangulation Library	website	Copyright © 2011 Ivan Fratric
QCustomPlot	Commercial License	Qt C++ Plotting Widget	website	Copyright © Emanuel Eichhammer
Qhull	Qhull License	Convex hull and Delauney triangulation program	http://www.qhull.org/	Copyright © 1993-2015 C.B. Barber and The National Science and Technology Research Center for Computation and Visualization of Geometric Structures (The Geometry Center) University of Minnesota

Software Library	License	Description	Website	Ownership or Copyright Information
Qt 5.2 - Community Version	LGPL v2.1, LGPL v3, and Qt LGPL Exception	GUI Framework	website	The Qt Toolkit is Copyright © 2014 Digia Plc and/or its subsidiary(-ies). Contact: http://www.qt-project.org/legal
Shark 3.0.0	LGPL	Optimization and machine learning framework	website	Copyright © Shark development team
Threading Building Blocks (TBB)	Apache v2.0	Intel multithreading library.	website	Copyright © 2005-2017 Intel Corporation. All Rights reserved.
TRUNCATED_NORMAL	LGPL	Random sampling of truncated normal distributions.	website	Copyright © John Burkardt
TryCatch	MIT	Try-catch macro implementation for C	website	Copyright © 2009-2015 Francesco Nidito
VTK	BSD	Visualization Toolkit for graphs, visualization, and rendering	website	Copyright (c) 1993-2008 Ken Martin, Will Schroeder, Bill Lorensen. All rights reserved.
YAML-CPP	YAML-CPP License	C++ YAML Parser	website	Copyright (c) 2008-2015 Jesse Beder
YARD	MIT	C++ Recursive-Descent Parser Generator	website	Authored by Christopher Diggins, December 2004

Software Library	License	Description	Website	Ownership or Copyright Information
ZeroMQ	LGPLv3	Messaging framework, used for communicating with other languages and tools via the scripting console	website	Copyright (c) The ZeroMQ authors

LGPL Compliance

The LGPL-licensed third-party shared-object (*.so, *.dll) libraries that are installed with GEMSIF may be replaced by new versions of the library as required by the LGPL license without recompiling or relinking the GEMSIF tool. However, GEMSIF has only been tested with the included versions of the libraries, and using different versions of these libraries with the GEMSIF tool is not supported by E x H. GEMSIF links to shared-object versions of the LGPL libraries using limited material from the header files that is restricted to data structure layouts, accessors, small macros, and small inline functions; GEMSIF thus forms a derivative work according to the LGPL, but is not subject to the source-code distribution requirements of derived works due to the specific limitation in section 5 of the LGPL. Source code for the LGPL binaries is freely available from the listed websites or on request to E x H.

GPLv2 Compliance

The GPLv2-licensed third-party programs that are packaged with GEMSIF are executed as separate processes, and are not linked with GEMSIF. Therefore GEMSIF is a separate program and is not a derived work of those programs. Thus GEMSIF is not subject to the source-code distribution requirements of derived works which are listed in section 2 of the GPLv2.

SHARK Optimization Library

The authors of the LGPL-licenced SHARK optimization library (which provides the Multi-objective CMA-ES optimizer to GEMSIF) make the following statement:

Any application of the SHARK code toward military research and use is expressly against the wishes of the SHARK development team.

Citations

This software makes use of third-party libraries that have been released under various open-source licenses. The original work behind the libraries referenced above are cited below. We gratefully acknowledge the contributions to scientific computing of these authors and of their generosity in sharing their work and code with the community.

- Steven G. Johnson, The NLopt nonlinear-optimization package, http://ab-initio.mit.edu/nlopt.
- S. Gudmundsson, "Parallel Global Optimization," M.Sc. Thesis, IMM, Technical University of

Denmark, 1998.

- K. Madsen, S. Zertchaninov, and A. Zilinskas, "Global Optimization using Branch-and-Bound," unpublished (1998).
- S. Zertchaninov and K. Madsen, "A C++ Programme for Global Optimization," IMM-REP-1998-04, Department of Mathematical Modelling, Technical University of Denmark, DK-2800 Lyngby, Denmark, 1998.
- M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in Advances in Optimization and Numerical Analysis, eds. S. Gomez and J.-P. Hennart (Kluwer Academic: Dordrecht, 1994), p. 51-67.
- M. J. D. Powell, "Direct search algorithms for optimization calculations," Acta Numerica 7, 287-336 (1998).
- M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," Department of Applied Mathematics and Theoretical Physics, Cambridge England, technical report NA2009/06 (2009), link:http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf.
- T. Rowan, "Functional Stability Analysis of Numerical Algorithms", Ph.D. thesis, Department of Computer Sciences, University of Texas at Austin, 1990.
- Conrad Sanderson. Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments. Technical Report, NICTA, 2010.
- This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)
- M. N. Polyanskiy. Refractive index database. Available at http://refractiveindex.info
- Christian Igel, Verena Heidrich-Meisner, and Tobias Glasmachers. Shark. Journal of Machine Learning Research 9, pp. 993-996, 2008.
- K. L.Clarkson, K.Mehlhorn, and R.Seidel. Four results on randomized incremental constructions. Comp. Geom.: Theory and Applications, pages 185—121, 1993. Preliminary version in Proc. Symp. Theor. Aspects of Comp. Sci., 1992.
- K. L.Clarkson. Safe and effective determinant evaluation. In Proc. 31st IEEE Symposium on Foundations of Computer Science, pages 387—395, Pittsburgh, PA, October 1992.
- Barber, C.B., Dobkin, D.P., and Huhdanpaa, H.T., "The Quickhull algorithm for convex hulls," ACM Trans. on Mathematical Software, 22(4):469-483, Dec 1996, http://www.qhull.org

Tools Used during GEMSIF Development

These open-source tools have been freely released by the development community. These tools have proved useful for our internal development process, and we acknowledge our use of these tools here. These tools are not redistributed with the GEMSIF software. Source code for these tools is available from the respective websites.

Software Library	License	Description	Website
Artistic Style 2.04	LGPL3	Artistic Style is a source code indenter, formatter, and beautifier for the C, C++, C# and Java programming languages.	website
Asciidoctor	MIT	Generation of help files and documentation	website
asciidoctor-mathematical	MIT	Processing equations in help files and documentation	website
Asciidoctor PDF	MIT	Generation of PDF documentation files	website
BeautifulSoup	MIT	Processing and generating some intermediate help files	website
CairoSVG	LGPLv3	Processing some SVG images in help files and documentation	website
cmake	BSD-Like	Cross-platform build system generator	website
doxygen	GPL	Automatic source-code reference manual formatting tool	website
Google Test	New BSD License	C++ Unit Test Framework	website
pegjs	MIT	Parser Generator for Javascript	website
Qt Creator	GPL3	C++ IDE	website

1. Individual Notices and Acknowledgements

2. Netlib Hull Ken Clarkson wrote this. Copyright (c) 1995 by AT&T.. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE