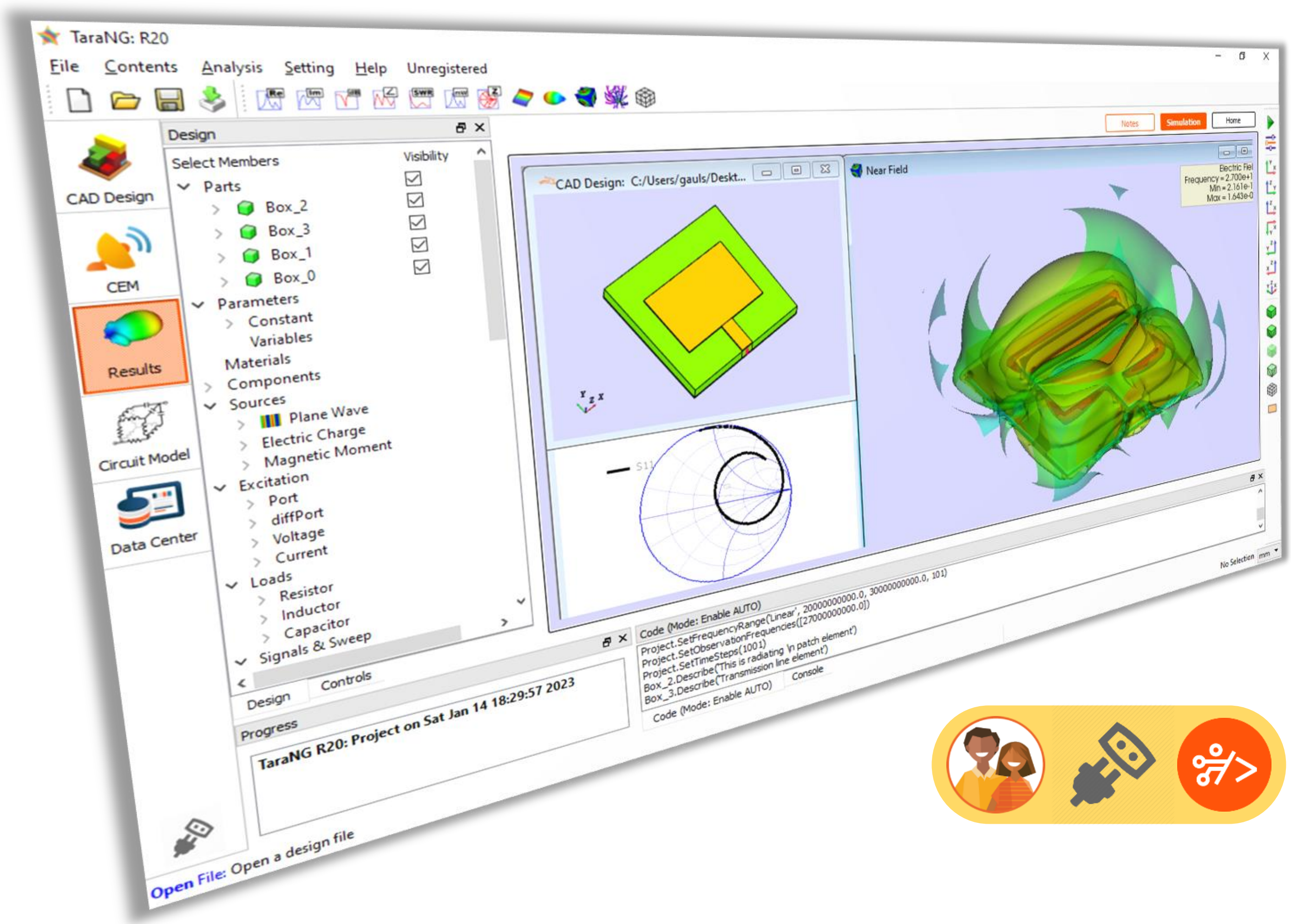
NUMEREGION's
TaraNG

TaraNG – 24.4

High Frequency Simulations



User Manual

TaraNG is indigenously developed simulation software tool in the domain of electromagnetics. TaraNG offers simulation based hands-on experience which is best suitable for interactive pedagogical learning

NUMEREGION is a DIPP recognized start-up working in field of simulation and mathematical modelling. The start-up got initial fund support incubation from Department of Science and Technology (DST) Government of India.

For more details visit our website: **www.numeregion.com**

Contents

Chapter 1: An Overview

TaraNG Simulation Tools

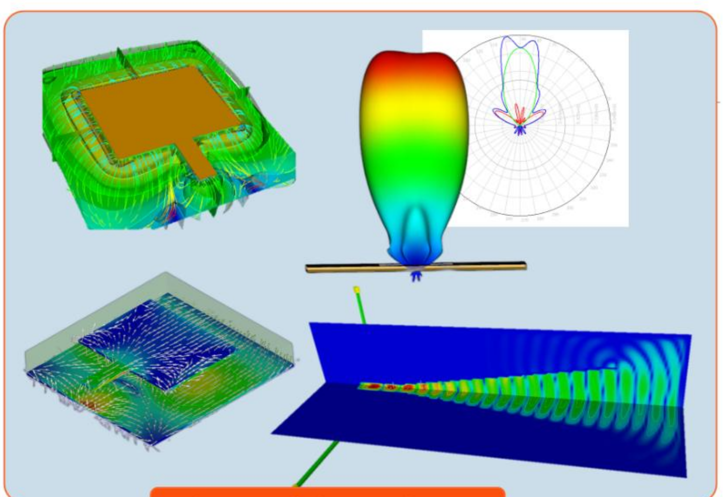
The name 'TaraNG' (in Hindi) refers to wave. TaraNG Simulation software Tool is a comprehensive electromagnetic analysis tool that provides solution to a variety of EM scenarios through simulation driven techniques. TaraNG has 3D CAD Design environment and analysis tool to compute the performance and present it with interactive visualization. TaraNG has capabilities to solve the problems ranging from low frequency (LF) to high frequency (HF) electronics both in time domain and frequency domain.

TaraNG - Applications

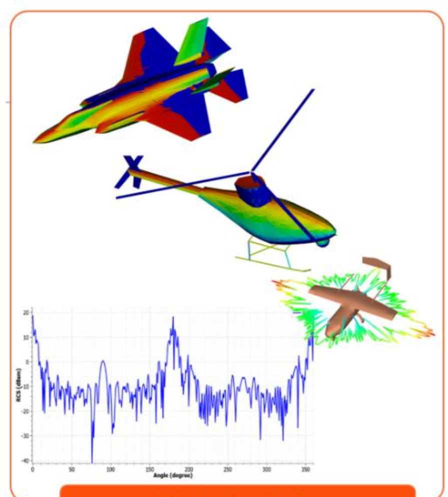
TaraNG has capabilities to solve the problems ranging from low frequency (LF) to high frequency (HF) electronics both in time domain and frequency domain.

The applications of TaraNG - High Frequency Module includes scenarios such as

Antennas & Antenna Arrays Design: Various types of antennas including Wire antennas, Planar Antennas, Deformed Ground Structures, Dielectric Resonating Antennas, Implantable antennas, Reconfigurable antennas, MIMO Antennas, 5G so on.



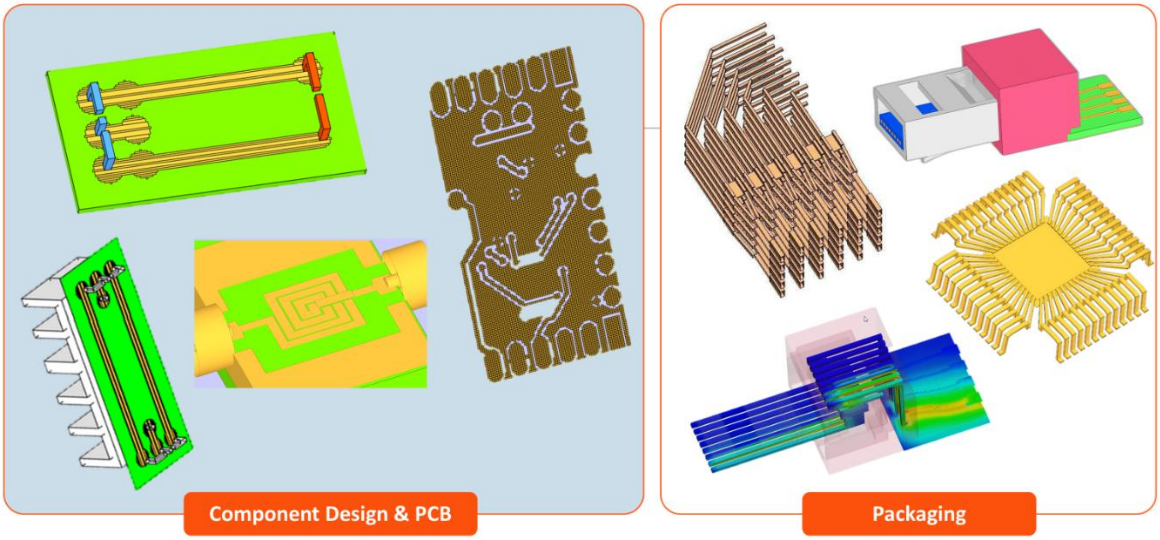
Antennas & Antenna Arrays



Radar Analysis

Microwave Circuits and Networks (RFIC/MMIC): It includes high frequency circuit design when the circuit size approaches to wavelength of signal a special, high frequency effects needs to be considered. Such designs include Power divider, Ring Resonators, Filters, Rat-race couplers so on.

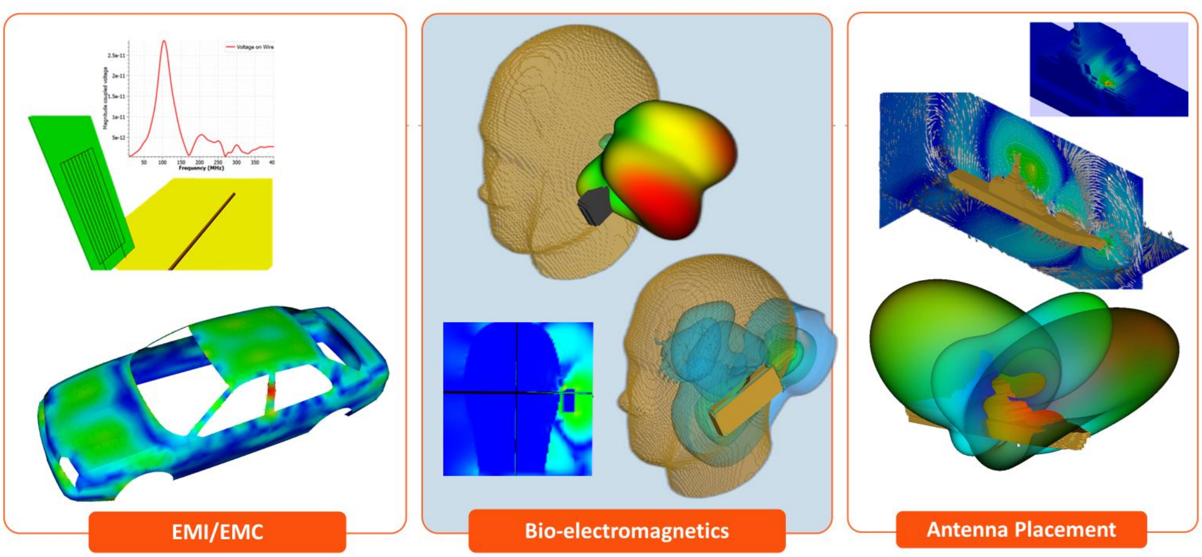
Metamaterial & Frequency Selective Surfaces: Analysing the designs with infinite repetations using periodic boundary condition, simulation of unit cell split ring resonators etc.



Bio-electromagnetics and SAR Analysis: Study of electromagnetic interactions and effects of radiations on human body, calculation of SAR and energy levels at different locations on body phantom.

Electromagnetic Interference & Combability (EMI/EMC issues): When group of conducting bars are kept together then filed coupling occurs which may result in induced field on nearby bar. This module allows on-chip interconnect wires to be considered in circuit analysis for high precision analysis.

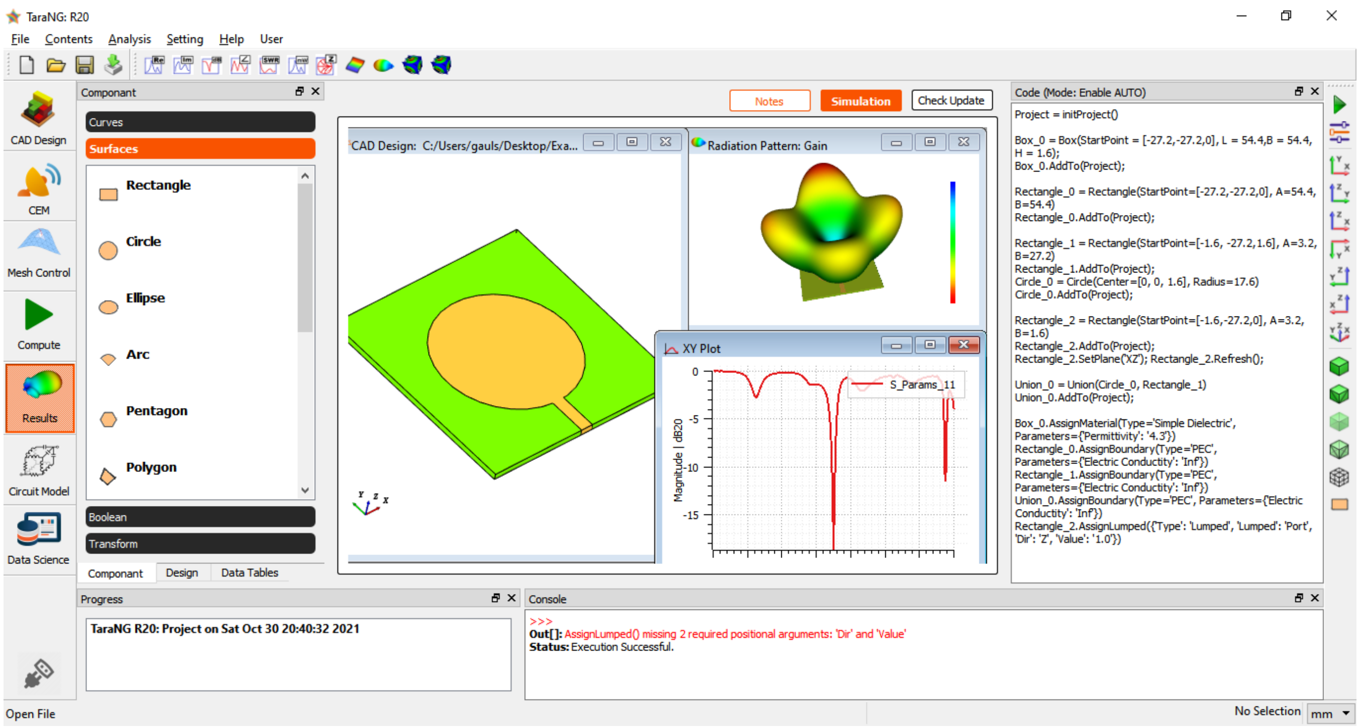
Radar Signature Analysis (Static RCS): At this higher frequency full wave techniques are not suitable as it requires more computational power. TaraNG's RCS Analysis module makes use of asymptotic methods to solve such large body problems without requiring computing resources. This module is capable of solving Bistatic & Monostatic RCS of objects with multilayer dielectric coating (stealth technology).



Ground Penetrating Radar (GPR): Time domain reflection & transmission (TDR/TDT)

Dynamic RCS Analysis: RCS when the target is moving over a particular trajectory. GIS Integration of the RCS result is also possible which enables data to be plotted on map.

TaraNG User Interface Layout



- <u>3D Design Capabilities:</u> This feature offers a full-fledged 3D CAD environment which allows user to create different shapes i.e. CAD primitives like circle, box cylinder etc and perform different Boolean and transformation operations like Union, Subtract, Intersect, Sweep, Revolve, Taper, Array, Scale so on. The software also supports import and export facilities to include other well-known file formats and designs.
- <u>User Customization:</u> TaraNG software allows users to add their own contents/codes that results in modifying the software and adding the features as per teacher's requirement.
- <u>Result Visualization and interaction:</u> The Software has facilities to generate all necessary 2D and 3D results like Port parameters, Radiation patterns, 3D planes, contours and streamlines with necessary controls and animations.
- <u>The Data Center for thing beyond simulation:</u> The software's data centre provides access to several important data sets and tables generated during simulation procedure. These data sets can be easily integrated with inbuilt algorithms for further optimizations and trainings

Circuit Schematic Mode

Circuit schematic design allows user to design and simulate circuits with target high frequency simulation components. Both active and passive electronics systems. Using this module one can design and solve Filters, power dividers, couplers, splitter etc.

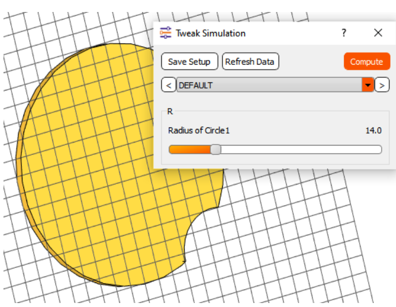
General Design Process In TaraNG

- ✓ Create CAD geometry using curve or surface structures and perform boolean and transformation operations on these primitives
- ✓ Import and modify CAD models and meshed geometries
- ✓ Set the definition of certain design parameters and add them as widget
- ✓ Set material parameters (e.g. dielectric constant, coating, conductivity)
- ✓ Set solution parameters (e.g. frequency, loads)
- ✓ Set excitations (e.g. frequency, loads)
- ✓ Create voxels from CAD geometries
- ✓ Assign boundary conditions
- ✓ Set calculation parameters (e.g. far-fields, near-fields, S-parameters, SAR analysis).
- ✓ Select computational method for analysis & set CEM parameters
- ✓ Generate equivalent circuit
- ✓ 3D far-fields and near-fields, iso-surfaces and ortho-slices

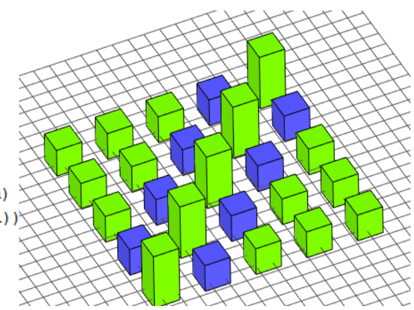
Chapter 2: Creating Models in TaraNG

TaraNG CAD Modeler is used for drawing different shapes into the 3D environment which are then useful for further analysis. TaraNG offers certain unique user interface features that makes easy to use yet powerful tool, these unique features are:

- ✓ **Parametric design with connected objects:** User can define variables and constraints of variables also called as parameter. When a particular parameter is updated, entire design with all connected objects updates itself.
- ✓ **User defined widgets:** Those parameters can be treated as UI widgets such as Sliders etc. Users can create their own tunning interface to control and set various parameters.
- ✓ **Auto code creation:** No coding experience is required for newbie user; The software generates the code automatically which can be further modified and updated easily.
- ✓ **Coding environment:** A full-fledged coding environment gives access of intermediate data (workspace) in a tabular format which can be exported in well-known formats. This makes designing most complicated problems through programming.
  It supports conditional loops such as if-else condition




```
Project = initProject()
Bricks = [[]]*5*5;
i = 0;
for y in range(0,10,2):
    for x in range(0,10,2):
        if x==y: h=2
        else: h=1
        Bricks[i] = Box(StartPoint = [x, y, 0],
                                L=1, B=1, H=h)
        Bricks[i].AddTo(Project, 'myBlock__'+str(i))
        if abs(x-y)==2:
            Bricks[i].SetColor('#5555ff');
            Bricks[i].Refresh();
        i += 1;
```



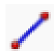
This chapter is intended to make user familiar with building different CAD models by drawing different shapes and performing various operations or transformations on them. The topics discussed in this tutorial are:

- Drawing curves (1D shapes: Lines, Polyline, Polygon, Rectangle, Circle, Ellipse, Arc, Parabola etc.)
- Drawing Surfaces (2D shapes: Polygon, Rectangle, Circle, Ellipse, Arc,)
- Drawing Volumes (3D shapes: Box, Cylinder, Sphere, Cone)
- Perform Boolean Operations (Union, Subtract, Intersect, Group)
- Perform Transformation (Sweep, Extrude, Revolve, Taper, Twist)
- Importing and Exporting Designs (STL, OBJ, NAS, PLY file formats)
- Auto code generation and creating own design macros instantly
- Python scripting and PLG (Programming Language Grammer)

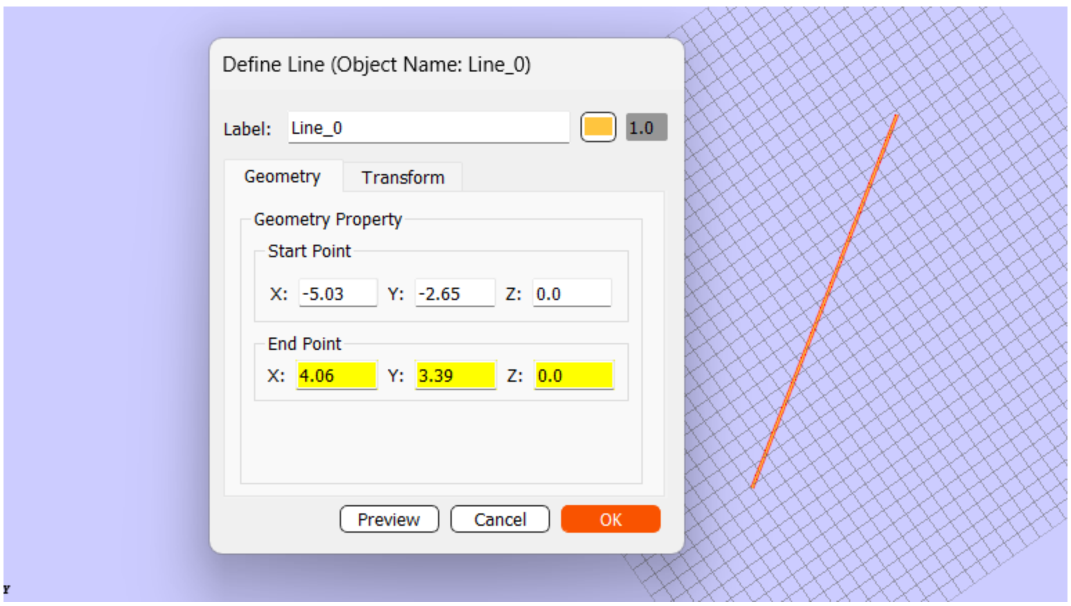
Drawing Shapes

We can draw different Curves, Surfaces and shapes within the design window just by making mouse clicks on the grid and later by doble clicking on them we can change the dimensions to enter accurate values.

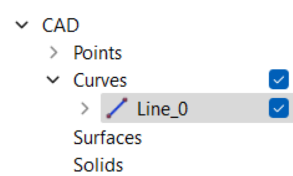
1. Drawing Line

To draw a line, click on Line  tool

- Click on the two points (*Start Point* and *End Point*) on the design grid where we want to draw the line. Press Esc key on the keyboard.
- The clicks may not be precise, in order to enter the dimensions precisely doble click on the line drawn in the design window.



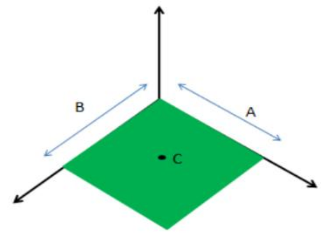
- Enter the desired values of *Start Point* and *End Point* in the window
- Click on the *OK* button and the window will be closed
- The line will be updated with desired coordinate points



- After successfully adding the Line into design it will be reflected under *Design Tree under CAD> Curves*
- The below script will be automatically generated after adding line to a *Project*

```
Line_0 = Linle(StartPoint=[-5.3,-2.65,0.0], EndPoint=[4.06,3.39,0.0])
Line_0.AddTo(Project,'Line_0')
```

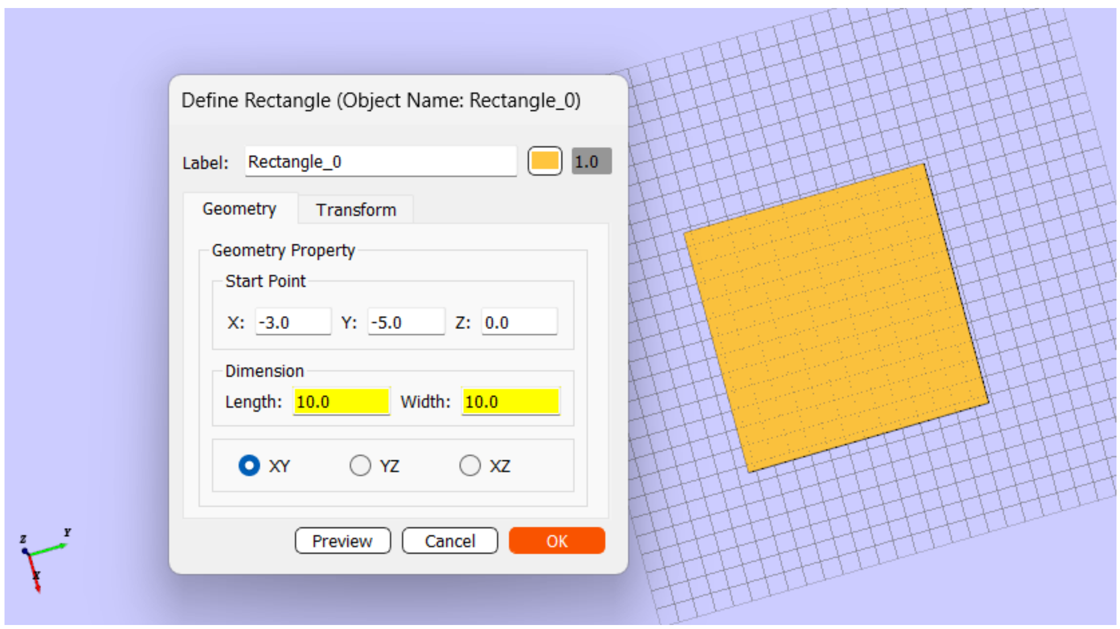
2. Drawing Rectangle



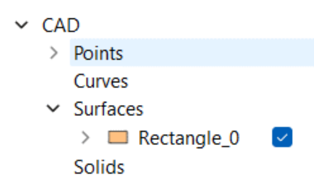
Start Point: The start points of the rectangle [X, Y, Z]
Width (A): The width of the rectangle.
Length (B): The length of the rectangle.

To draw a rectangle, click on Rectangle tool  under *Shapes* toolbar.

- Click on the two points (*Start Point* and *End Point*) on the design grid where we want to draw the rectangle

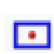


- Enter the *Start Point* of rectangle to define the position of rectangle in *Start Point* box under *Geometry Property*.
- Enter the Length(A) and width(B) of rectangle in *Length* and *Width* box under *Dimension* of Geometry Property which defines dimensions of rectangle.
- Set the Plane of Rectangle as XY which is by-default. User can change the plane and apply Transform to any given orientation of the object. *Transform* option is explained in later section



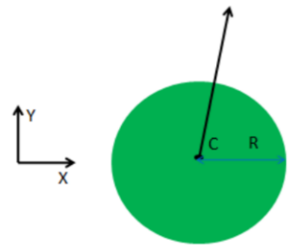
- After creating the Rectangle into design it will be reflected under *Design Tree: CAD>Surfaces*
- The below script will be automatically generated after adding rectangle to a *Project*

```
Rectangle_0 = Rectangle(StartPoint=[-3.0,-5.0,0.0], A=10.0, B=10.0)
Rectangle_0.AddTo(Project,'Rectangle_0')
```

Drawing Rectangle as a Curve: To draw a rectangular curve, click on *Rectangle*  tool and repeat the above process.

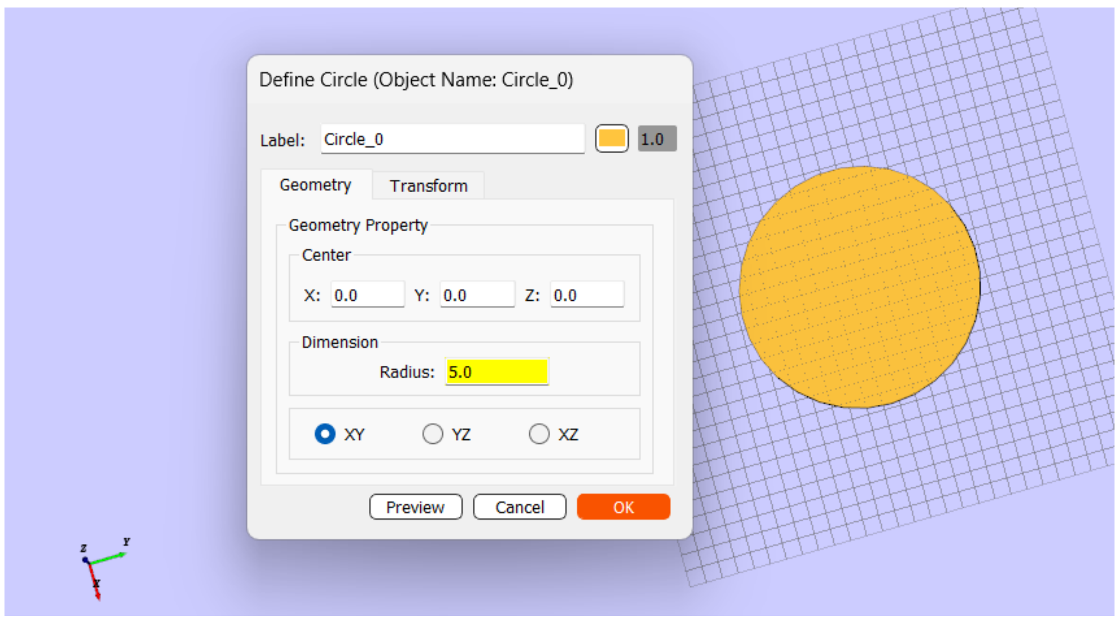
3. Drawing Circle

Center: The center points of the circle [X, Y, Z]
Radius (A): The radius of the circle.



To draw a circle, click on Circle tool under *Shapes* toolbar.

- Click on the two points (*Center Point* and *a point to get Radius*) on the design grid where we want to draw the circle.

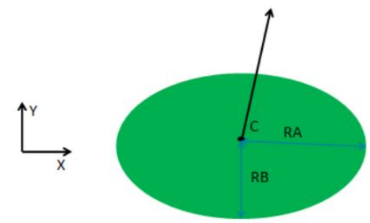


- Enter the *Center Point* of circle to define the position of circle in *Center Point* box under *Geometry Property*.
- Enter the Radius of circle in *Radius* box under *Dimension* of *Geometry Property* which defines dimensions of circle.
- Set the Plane of Circle as XY which is by-default. User can change the plane and apply *Transform* to any given orientation of the object this option is explained in later section.
- After successfully adding the Circle to design, it will be reflected under *Surfaces of Design Tree*
- The below script will be automatically generated after adding circle to a *Project*

```
Circle_0 = Circle(Center=[0.0,0.0,0.0], Radius=5.0)
Circle_0.AddTo(Project,'Circle_0')
```

Drawing Circle as a Curve: To draw a rectangular curve, click on *Circle* tool and repeat the above process.

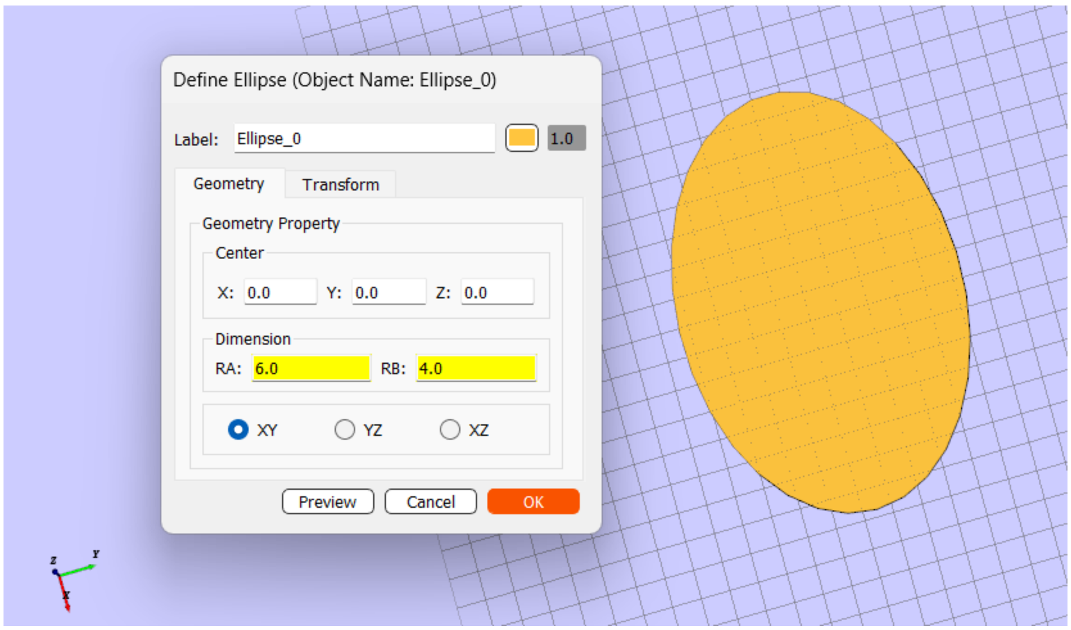
4. Drawing Ellipse



Center: The center points of the ellipse [X, Y, Z]
Radius (RA): The radius of the ellipse in one direction.
Radius (RB): The radius of the ellipse in other direction.

To draw a circle, click on Circle tool  under *Shapes* toolbar.

- Click on the two points (*Center Point* and *a point to get Radius*) on the design grid where we want to draw the ellipse.

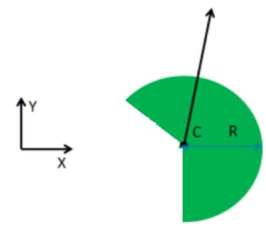


- Enter the *Center Point* of ellipse to define the position of circle in *Center Point* box under *Geometry Property*.
- Enter the Radius of ellipse in *RA* and *RB* box under *Dimension* of *Geometry Property* which defines radius of ellipse in the two directions.
- Set the Plane of Ellipse as XY which is by-default. User can change the plane and apply Transform to any given orientation of the object. This option is explained in later section.
- After successfully adding the Ellipse, it will be reflected under *Surfaces of Design Tree*
- The below script will be automatically generated after adding ellipse to a *Project*

```
Ellipse_0 = Ellipse(Center=[0.0,0.0,0.0], RA=6.0, RB=4.0)
Ellipse_0.AddTo(Project,'Ellipse_0')
```

Drawing Ellipse as a Curve: To draw a elliptical curve, click on *Ellipse*  tool and repeat the above process.

5. Drawing Arc



Center: The center points of the arc [X, Y, Z]
Radius: The radius of the arc
Angle: The angle made by an arc.

To draw an arc, click on Arc tool  under *Shapes* toolbar.

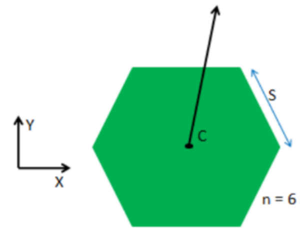
- Click on the two points (*Center Point* and *a point to get Radius* and *Angle*) on the design grid where we want to draw the arc.

- Enter the *Center Point* of arc to define the position of arc in *Center Point* box under *Geometry Property*.
- Enter the Radius of an arc in *Radius* and an arc angle *Angle* box under *Dimension* of *Geometry Property* which defines radius of an arc and arc angle in degrees.
- Set the Plane of Arc as XY which is by-default. User can change the plane and apply Transform to any given orientation of the object. This option is explained in later section.
- After successfully adding an Arc, it will be reflected under *Surfaces of Design Tree*
- The below script will be automatically generated after adding an arc to a *Project*

```
Arc_0 = Ellipse(Center=[1.26,0.37,0.0], Radius=7.5, Angle=145.0)
Arc_0.AddTo(Project,'Arc_0')
```

Drawing Arc as a Curve: To draw a arc curve, click on *Arc*  tool and repeat the above process.

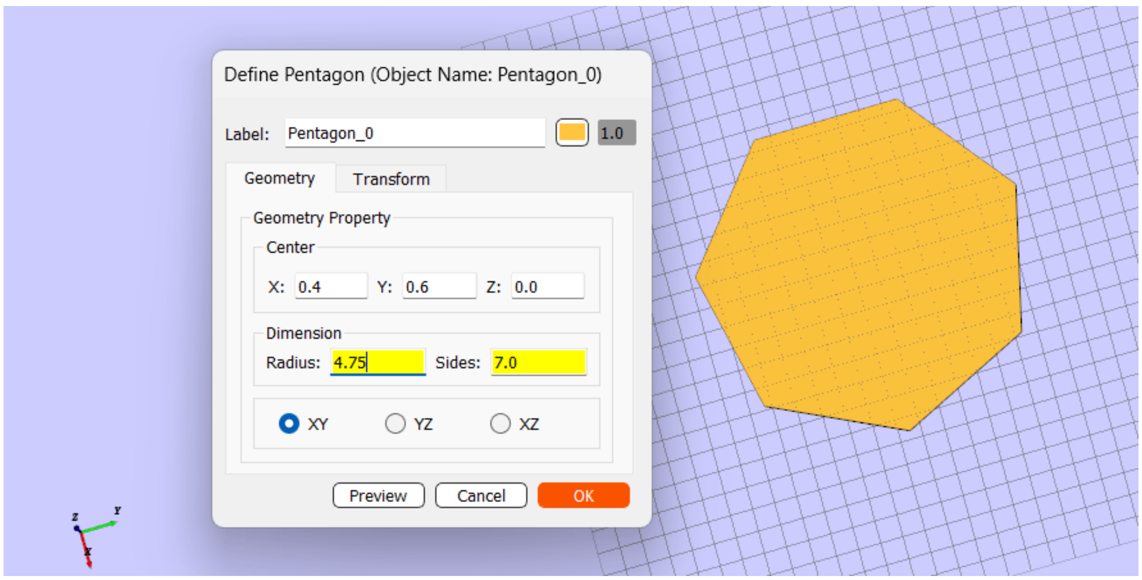
6. Drawing Regular Polygon (Pentagon)



Center: The center points of the regular polygon [X, Y, Z]
Radius: The radius of the regular polygon
Sides: Number of sides for a regular polygon.

To draw a regular polygon, click on tool under *Shapes* toolbar.

- Click on the two points (*Center Point* and *a point to get Radius*) on the design grid where we want to draw the regular polygon.



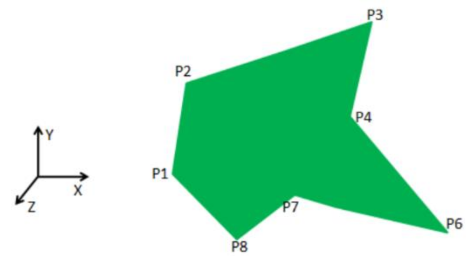
- Enter the *Center Point* of regular polygon to define the position of regular polygon in *Center Point* box under *Geometry Property*.
- Enter the Radius of a regular polygon in *Radius* and number of sides in *Sides* box under *Dimension* of *Geometry Property*
- Set the Plane of a regular polygon as XY which is by-default. User can change the plane and apply Transform to any given orientation of the object. This is explained in later section.
- After successfully adding regular polygon, it will be reflected under *Surfaces of Design Tree*
- The below script will be automatically generated after adding a regular polygon to a *Project*

```
Pentagon_0 = Pentagon(Center=[0.4,0.6,0.0], Radius=4.75, Sides=7)
Pentagon_0.AddTo(Project,'Pentagon _0')
```

Drawing Regular Polygon as a Curve: To draw a regular polygon curve, click on *Pentagon* tool and repeat the above process.

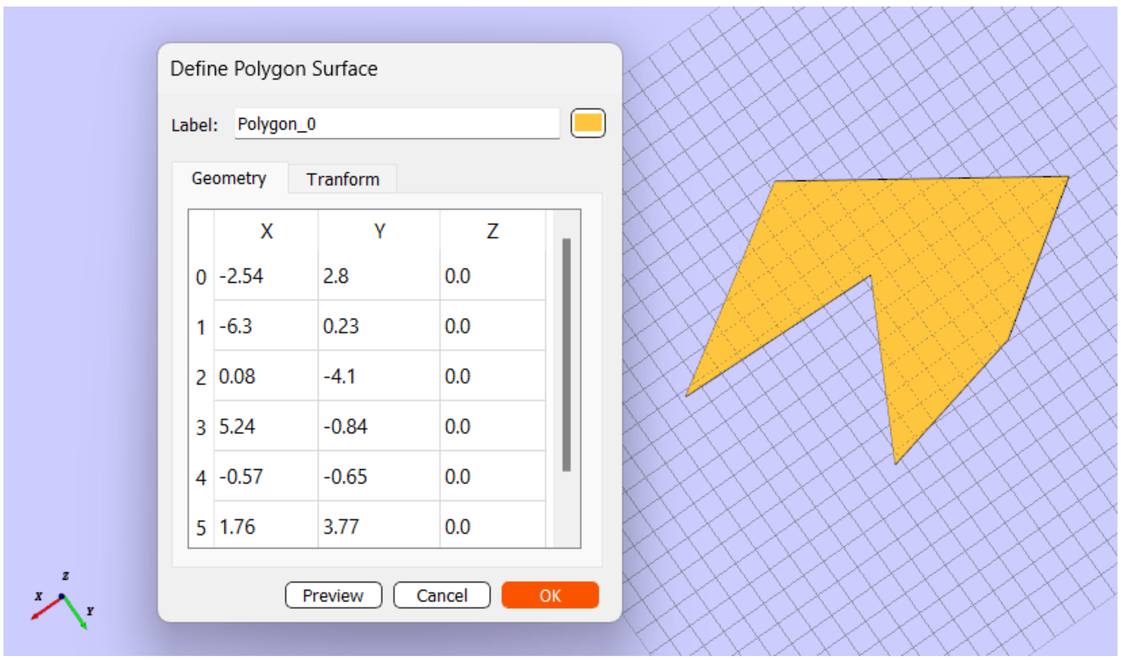
7. Drawing Polygon

Points: List of point coordinates to draw the polygon [X, Y, Z]



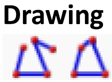
To draw polygon, click on *Polygon* tool  under *Shapes* toolbar.

- Click on the multiple points on the design grid where we want to draw the polygon after adding sufficient points to the polygon right click the mouse.



- To edit the drawn polygon, doble click on the polygon, it will open a window consisting of points list.
- Enter the points coordinates precisely under *X, Y, Z* of *Geometry Property*
- User can change the plane and apply Transform to any given orientation of the object. This is explained in later section.
- After successfully adding polygon, it will be reflected under *Surfaces of Design Tree*
- The below script will be automatically generated after adding a polygon to a *Project*

```
Polygon_0 = Polygon(Points=[[-2.54, 2.8, 0.0], [-6.3, 0.23, 0.0], [0.08, -4.1, 0.0], [5.24, -0.84, 0.0],
[-0.57, -0.65, 0.0], [1.76, 3.77, 0.0]])
Polygon_0. AddTo (Project, 'Polygon_0')
```

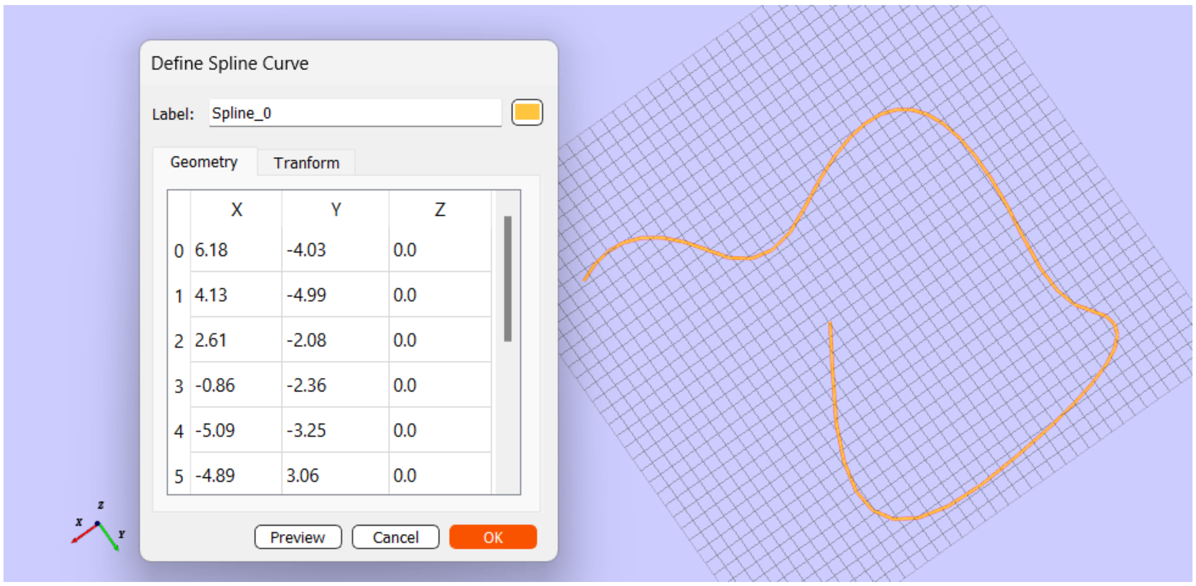
Drawing Polygon and Polyline as a Curve: To draw a polygonal curve, click on *Polyline and Polygon*  tool and repeat the above process.

8. Drawing Spline

Points: List of point coordinates to draw the polygon [X, Y, Z]

To draw polygon, click on *Polygon* tool  under *Curves* toolbar.

- Click on the multiple points on the design grid where we want to draw the spline after adding sufficient points to the screen right click the mouse. To edit the drawn polygon, doble click on the polygon, it will open a window consisting of points list.



- Enter the points coordinates precisely under *X, Y, Z* of *Geometry Property*
- User can change the plane and apply Transform to any given orientation of the object. This is explained in later section.
- After successfully adding polygon, it will be reflected under *Surfaces of Design Tree*
- The below script will be automatically generated after adding a polygon to a *Project*

```
Spline_0 = Spline(Points=[[6.18, -4.03, 0.0], [4.13, -4.99, 0.0], [2.61, -2.08, 0.0], [-0.86, -2.36, 0.0],
[-5.09, -3.25, 0.0], [-4.89, 3.06, 0.0], [-7.1, 5.0, 0.0], [-0.4, 8.2, 0.0], [0.56, 2.08, 0.0], [1.2, 1.6, 0.0]])
Spline_0.AddTo (Project, 'Spline_0')
```

9. Drawing Parabolas and Hyperbolas:

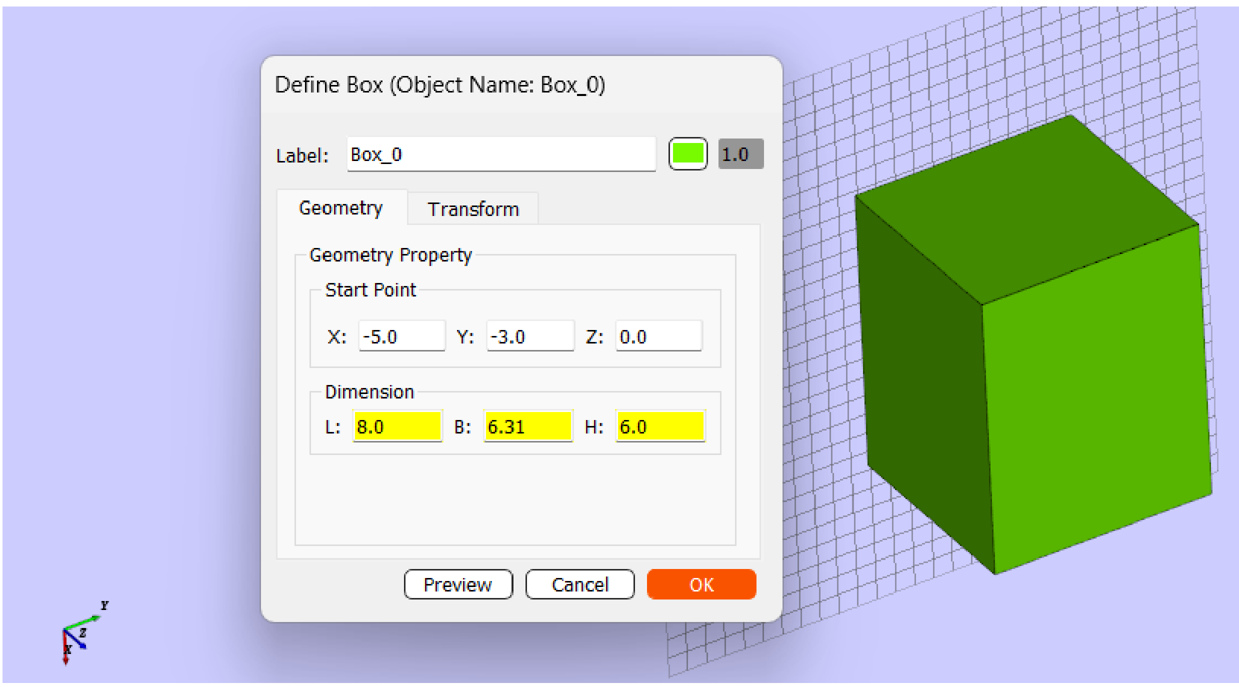
In CAD Modeller of TaraNG it is possible to draw Parabolas, Hyperbolas or any equation-based curve through readily available macros as well as scripting environment. The scripting envirnment is detailed in the later section.

10. Drawing Box

Start Point: The start points of the box [X, Y, Z]
Length (L): The dimension of the box in one direction.
Breadth (B): The dimension of the box in second direction.
Width (W): The dimension of the box in third direction.

To draw a box, click on Box tool  under *Shapes* toolbar

- Drawing Box requires three clicks, click on the start point on the design grid where we want to draw the box then click on two more positions to define Length, Width and Breadth of the Box respectively. To edit the drawn box, doble click on the box, it will open a window consisting of dimensions



- Click on OK after entering the desired *Geometry Property* and *Dimension*
- After successfully adding Box, it will be reflected under *Volumes of Design Tree > CAD*
- We can also change the colour and opacity of the box by clicking on the colour button and entering the value of opacity between *0.0-1.0*
- The below script will be automatically generated after adding a box to a *Project*

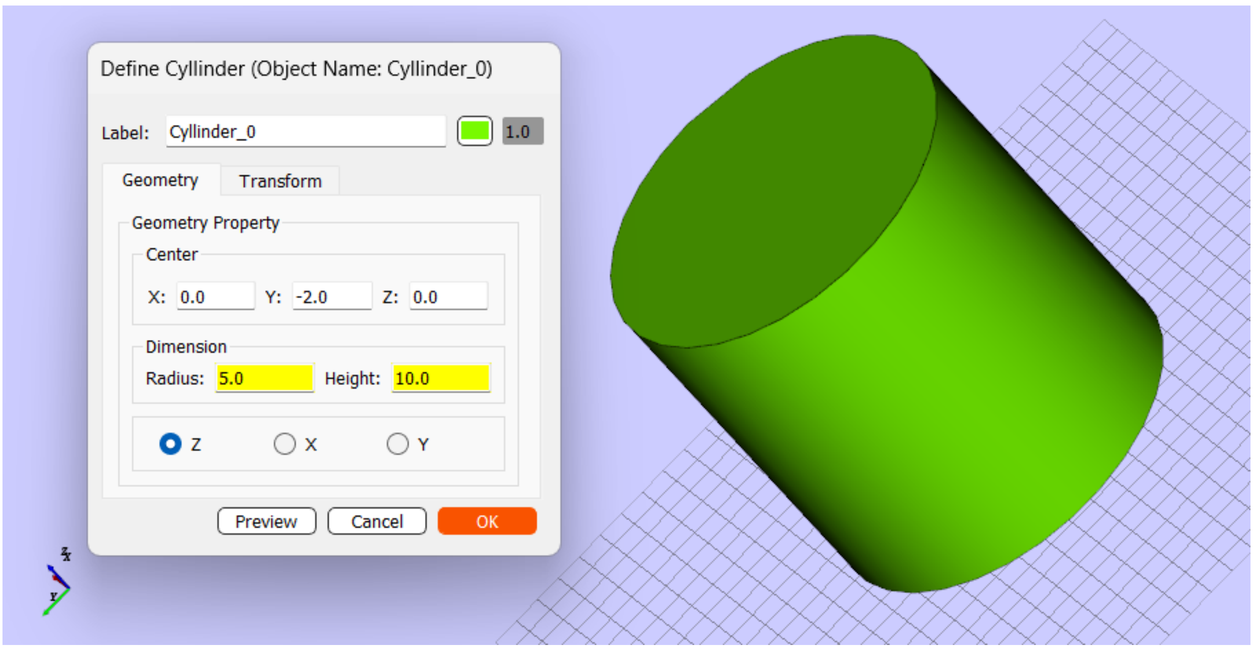
```
Box_0 = Box(StartPoint=[-5.0,-3.0,0.0], L=8.0, B=6.31, H=6.0)
Box_0.AddTo (Project, 'Box_0')
```

11. Drawing Cylinder

Center Point: The center points of the cylinder [X, Y, Z]
Radius: The radius of the cylinder
Height: The height of the cylinder in z direction.

To draw a cylinder, click on Box tool  under *Shapes* toolbar

- Drawing cylinder requires three clicks, click on the center point on the design grid where we want to draw the cylinder then click on two more positions to define Radius and Height of the cylinder respectively. To edit the drawn cylinder, doble click on the cylinder, it will open a window consisting of dimensions



- Click on OK after entering the desired *Geometry Property* and *Dimension*
- After successfully adding Cylinder, it will be reflected under *Design Tree: CAD > Volumes*
- We can also change the colour and opacity by clicking on the colour button and entering the value of opacity between *0.0-1.0*
- The below script will be automatically generated after adding a cylinder to a *Project*

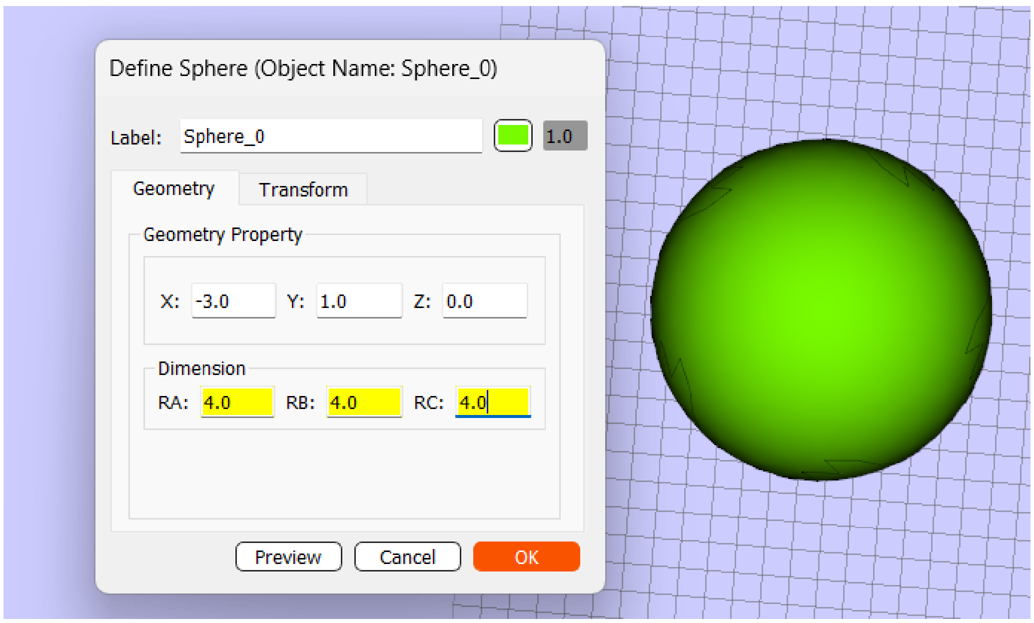
```
Cyllinder_0 = Cyllinder (Center=[0.0,-2.0,0.0], Radius=5.0, Height=10.0)
Cyllinder_0.AddTo (Project, ' Cyllinder_0')
```

12. Drawing Sphere

Center Point: The center points of the sphere [X, Y, Z]
Radius: The radius of the sphere

To draw a sphere, click on Sphere tool  under *Shapes* toolbar

- Drawing sphere requires two clicks, click on the center point on the design grid where we want to draw the sphere then click on positions to define Radius of the sphere. To edit the drawn sphere, doble click on the sphere, it will open a window consisting of dimensions



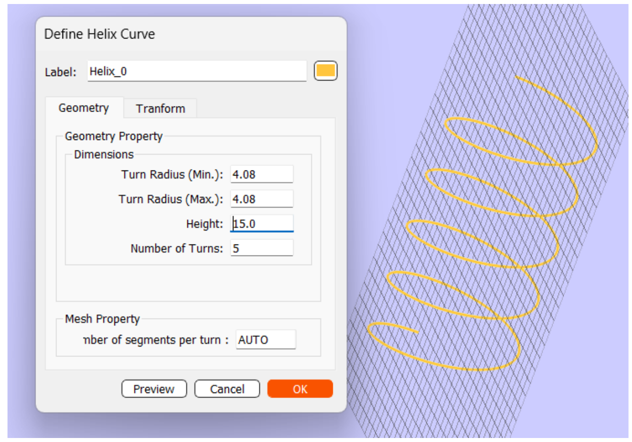
- Click on OK after entering the desired *Geometry Property* and *Dimension*
- After successfully adding Sphere, it will be reflected under *Design Tree: CAD > Volumes*
- We can also change the colour and opacity by clicking on the colour button and entering the value of opacity between *0.0-1.0*
- The below script will be automatically generated after adding a sphere to a *Project*

```
Sphere_0 = Sphere (Center=[-3.0,1.0,0.0], RA=4.0, RB=4.0, RC=4.0)
Sphere_0.AddTo (Project, ' Sphere_0')
```

13. Drawing Helix

To draw a Helix, click on Helix tool  under *Curves* toolbar

- Drawing helix requires three clicks, click on the center point on the design grid where we want to draw the helix then click on two more positions to define Radius and Height of the helix. To edit the drawn helix, doble click on the helix, it will open a window consisting of dimensions
- Click on OK after entering the desired *Geometry Property* and *Dimension i.e Minimum and Maximum Turn Radius, Height and Number of Turns*

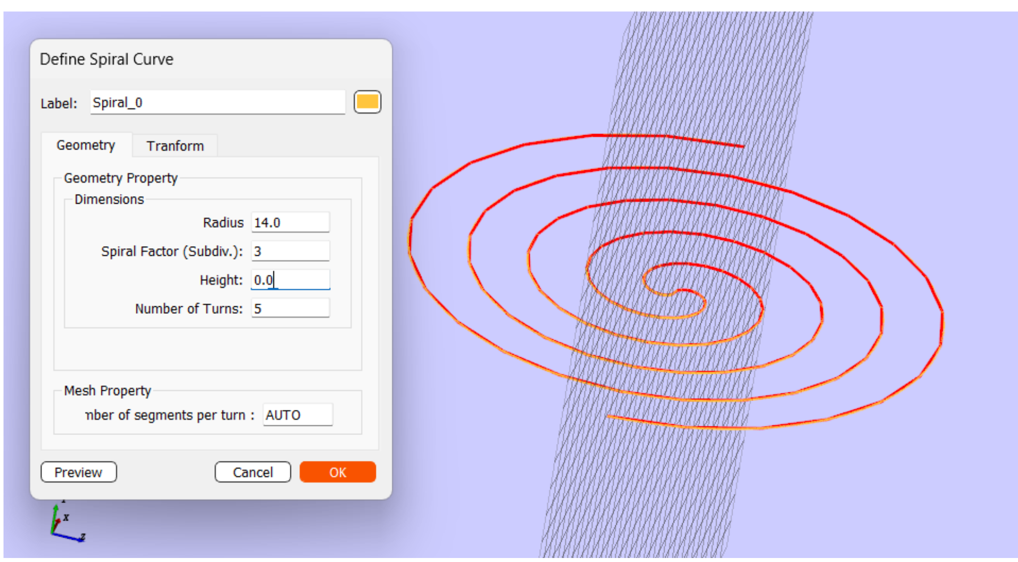



```
Helix_0 = Helix (Center=[-3.0,4.0,0.0], R1=4.08, R2=4.08, Height=15.0, NTurns=5)
Helix_0.AddTo (Project, 'Helix_0')
```

14. Drawing Spiral

To draw a Spiral, click on Spiral tool  under *Shapes* toolbar

- Drawing spiral requires three clicks, click on the center point on the design grid where we want to draw the spiral then click on positions to define Radius and Height of the spiral. To edit the drawn spiral, doble click on the spiral, it will open a window consisting of dimensions



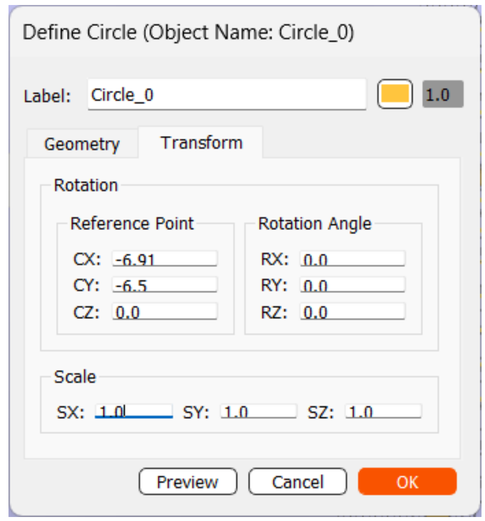
- Click on OK after entering the desired *Geometry Property* and *Dimension i.e Radius, Height and Number of Turns*

```
Spiral_0 = Spiral (Radius=14.0, SF=3, Height=0.0, NTurns=5)
Spiral_0.AddTo (Project, 'Spiral_0')
```

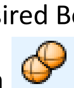
Transforming the object

The transform setting is applied on the object to Translate, Scale or Rotate the object with respect to a given reference point

- Select the object and doble click on the object
- Choose the **Transform** Tab in the window
- Enter the *Reference Point [CX,CY,CZ]*, *Rotation Angle {RX,RY,RZ} and the Scaling [SX,SY,SZ]*
  In the window as shown
- Click on the OK button once desired operation is applied.



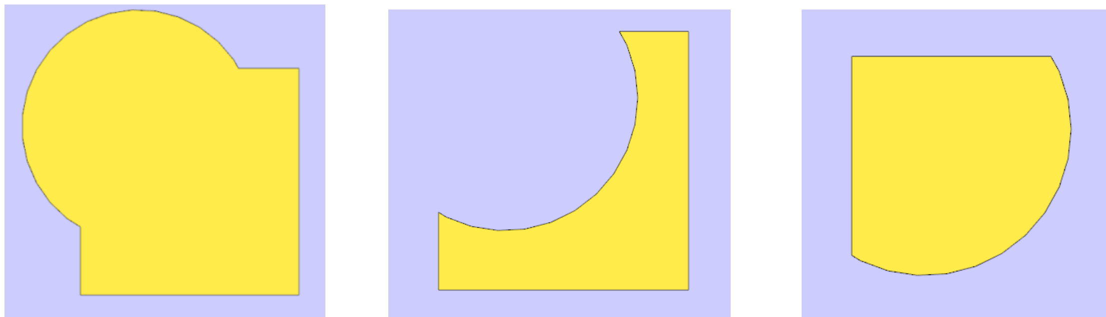
Performing Boolean Operation

- To perform the Boolean operation, we need to select two objects on which the Boolean operation need to be applied.
- Choose the desired Boolean operation  Union, Subtract or Intersect
- **Union:** Click on  tool for applying union of two drawn objects. The examples of Union is as below.
  Union_0 = Union(Rectangle_0, Circle_0)

- **Subtract:** Click on  tool for applying subtract of two drawn objects. The examples of Subtract is as below.

  Subtract_0 = Subtract(Rectangle_0, Circle_0)

- **Intersect:** Click on  tool for applying intersect of two drawn objects. The examples of Intersect is as below.
  Intersect_0 = Intersect(Rectangle_0, Circle_0)

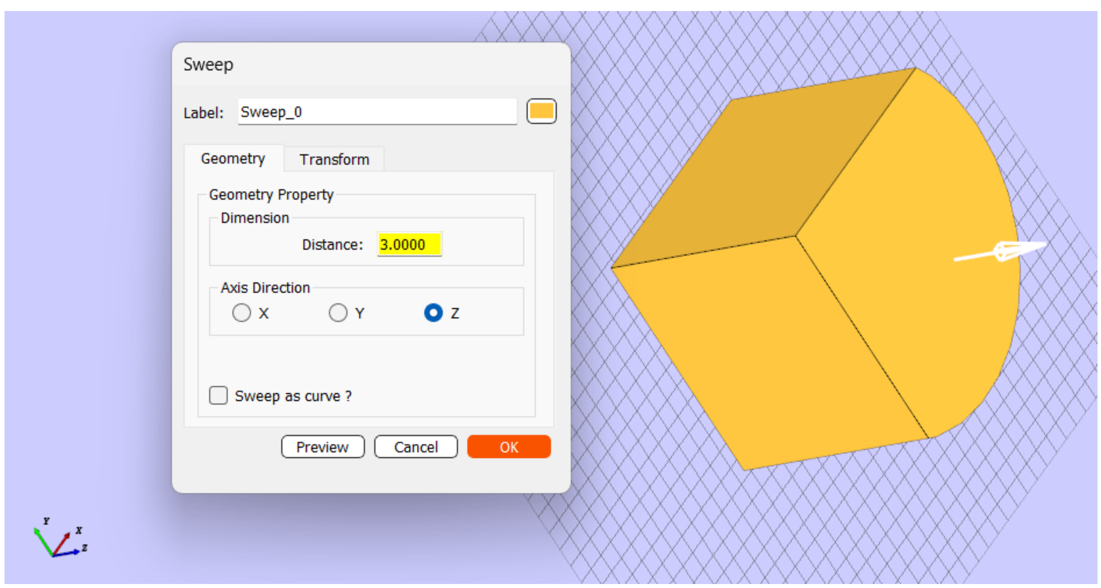


Performing Transformation Operation

To perform the Transformation operation, we need to select one objects on which the Transformation operation need to be applied. We can choose the desired Transformation operation  Sweep, Extrude, Revolve, Taper or Twist the object

1. Sweep the object

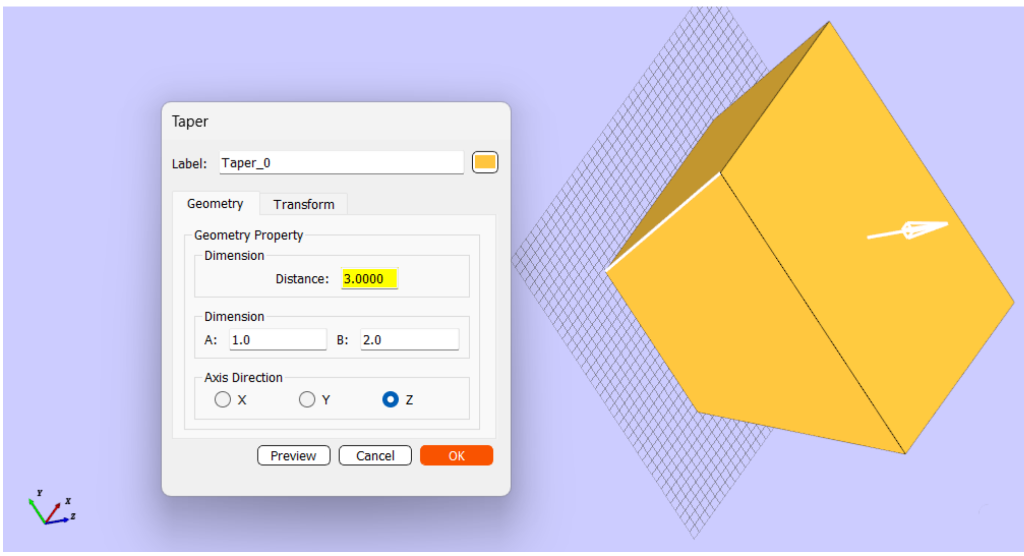
- Select the object on which the Sweep operation to be performed
- Click on the Sweep tool  from the *Transformation* toolbar



- Choose the *Axis Direction* of the Sweep and enter the *Distance* for the Sweep in the *Geometry Property* section. Click on *OK* button.
- We can perform this operation on a surface as well as on a curve

```
Sweep_0 = Sweep (Intersect_0, Distance=3.0, Dir='Z')
Sweep_0.AddTo (Project, 'Sweep_0')
```

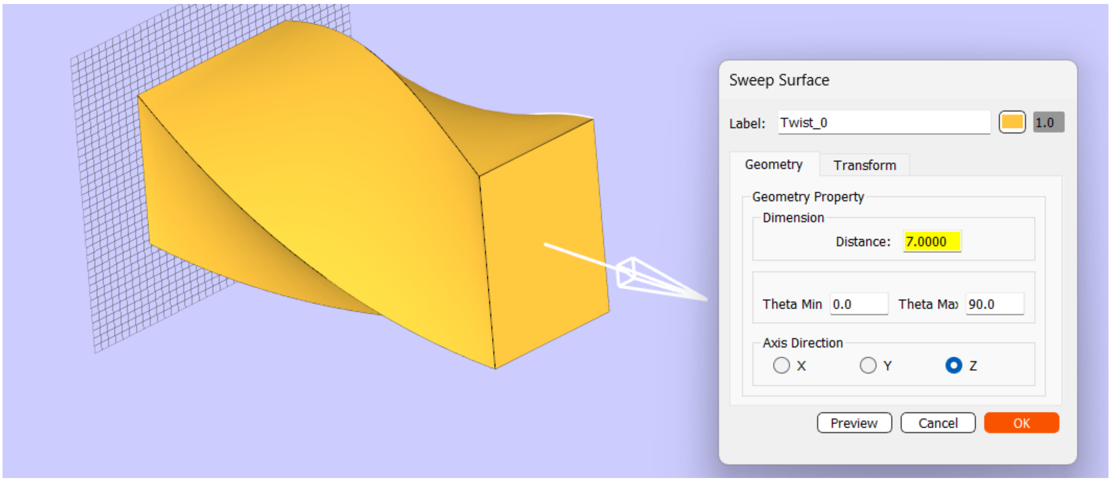
2. **Taper the object**
    - Select the object on which the Taper operation to be performed
    - Click on the Sweep tool from the *Transformation* toolbar
    - Choose the *Axis Direction* of the Taper and enter the *Distance* for the Taper in the *Geometry Property* section.
    - Enter the amount of scaling in the two directions *A, B* in the *Dimensions*
    - Click on *OK* button.



- We can perform this operation on a surface as well as on a curve
- The auto-generated code for this operation is as below

```
Taper_0 = Taper (Rectangle_0, Distance=3.0, Dir='Z')
Taper_0.AddTo (Project, 'Taper_0')
```

3. **Twist the object**
    - Select the object on which the Twist operation to be performed
    - Click on the Twist tool from the *Transformation* toolbar
    - Choose the *Axis Direction* of the Twist and enter the *Distance* for the Twist in the *Geometry Property* section.
    - Enter the angle of twist by the value of *Theta Min* and *Theta Max* in the *Dimensions*
    - Click on *OK* button.

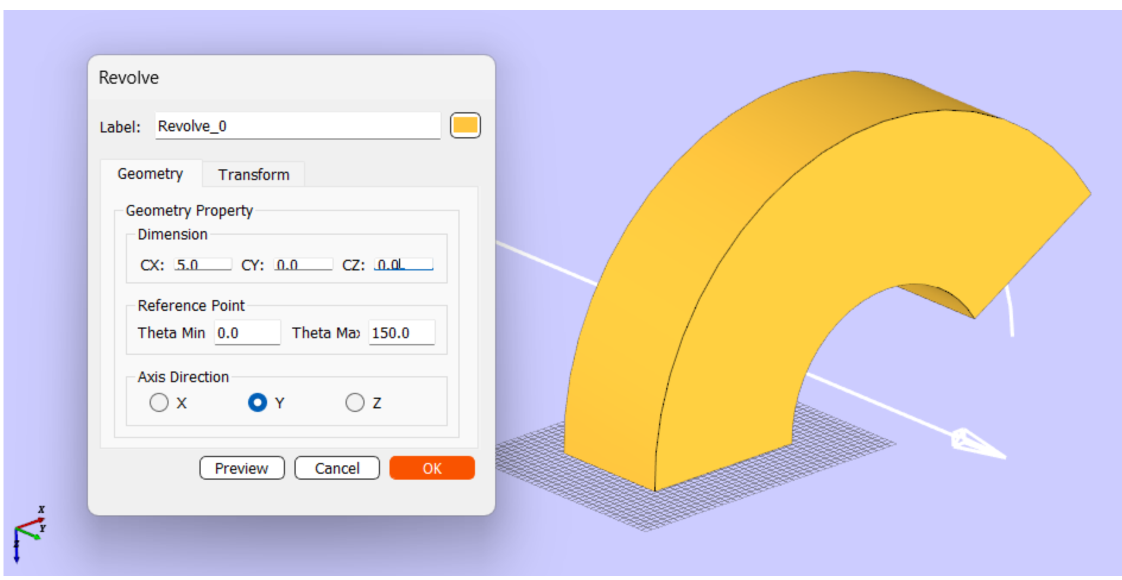


- We can perform this operation on a surface as well as on a curve
- The auto-generated code for this operation is as below

```
Twist_0 = Twist (Rectangle_0, Distance=7.0, Angle = [0.0,90.0], Dir='Z')
Twist_0.AddTo (Project, 'Twist_0')
```

4. Revolve the object

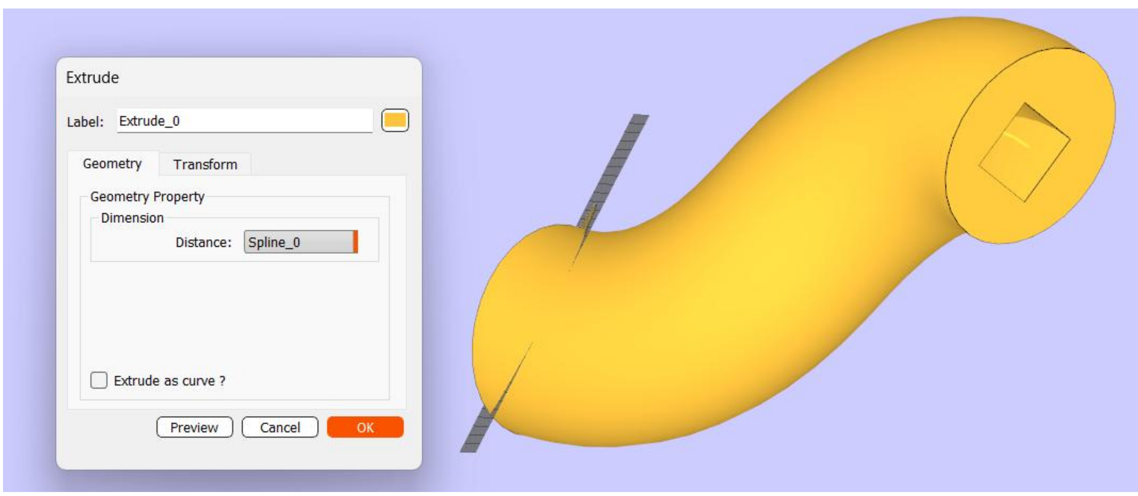
- Select the object on which the Revolve operation to be performed
- Click on the Revolve tool  from the *Transformation* toolbar
- Choose the *Axis Direction* of the Revolve and enter the *Reference Point [CX, CY, CZ]* for the Revolve in the *Geometry Property* section.
- Enter the angle for revolving by the value of *Theta Min* and *Theta Max* in the *Dimensions*
- Click on *OK* button.



- We can perform this operation on a surface as well as on a curve
- The auto-generated code for this operation is as below

```
Revolve_0 = Revolve (Rectangle_0, Center = [5.0,0.0,0.0], Angle = [0.0,150.0], Dir='Y')
Revolve_0.AddTo (Project, 'Revolve_0')
```

5. **Extrude the object**
   - Select the object on which the Extrusion to be performed
   - Click on the Extrude tool from the *Transformation* toolbar
   - Choose the *Curve as a path* of the Extrusion for the Revolve in the *Geometry Property* section.
   - Click on *OK* button.



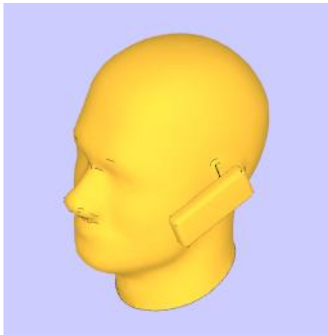
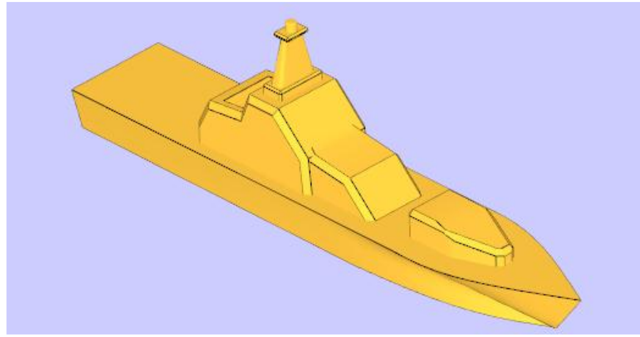
- We can perform this operation on a surface as well as on a curve
- The auto-generated code for this operation is as below

```
Extrude_0 = Extrude (Difference_0, Curve = Spline_0)
Extrude_0.AddTo (Project, 'Extrude_0')
```

Importing CAD Models

TaraNG offers Import functionality to use the CAD models designed in other third-party software.

- To import the CAD model click on button from *File Menu*
- *The File Open Window will appear*
- *Browse to the desired file*
- *Open the file which has file formats as: STL, OBJ,PLY*

Designing CAD Models through Scripting

TaraNG offers drawing most complicated shapes through Python compatible scripting. This scripting supports

- Defining classes and functions
- Defining loops such as for loop, while loop
- Checking and defining if, else conditions
- Using datasets arrays, lists etc.
- Using existing existing python libraries such as numpy, scipy, pandas, ternsorflow etc.
- It includes mostly all the important python features.

1. Drawing Cookies in TaraNG

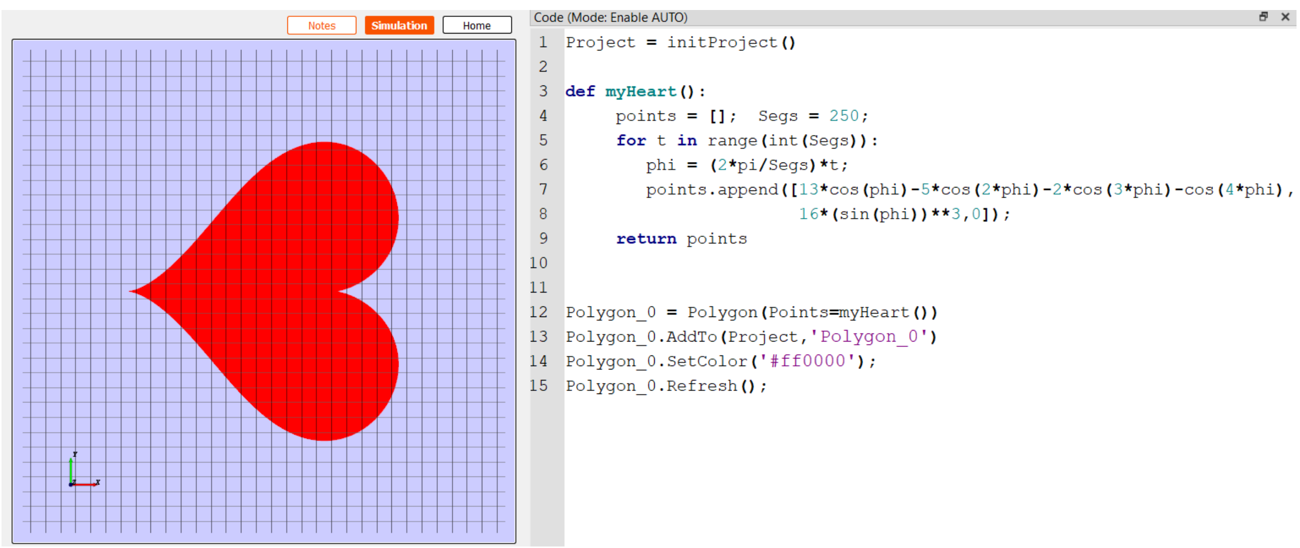
For drawing cookies, we can define a function to find coordinates for the desired shape of cookies and call this function to set the coordinates of polygon

The function used for making the cookie shape is as below,

```
def myCookie(r0 = 10, r1 = 2, N = 10):
        points = [];  Segs = 250;
        for t in range(int(Segs)):
                phi = (2*pi/Segs)*t;
                r=r0+r1*cos(N*phi);
                points.append([r*cos(phi), r*sin(phi),0]);
        return points

Polygon_0 = Polygon(Points=myCookie(r0 = 10, r1 = 1, N = 10))
Polygon_0.AddTo(Project,'Polygon_0')
```

2. Drawing Heart shape in TaraNG

For drawing cookies, we can define a function to find coordinates for the desired shape of cookies and call this function to set the coordinates of polygon



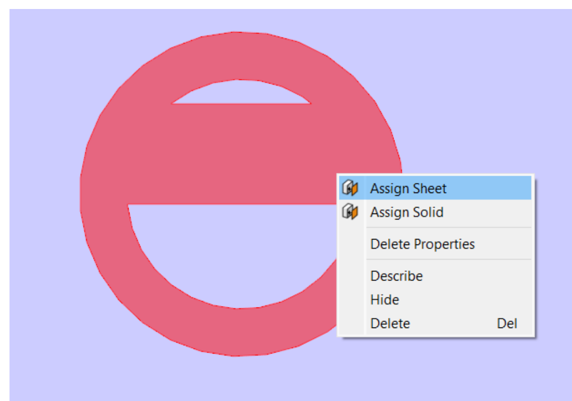
Chapter 3: Electromagnetic Properties

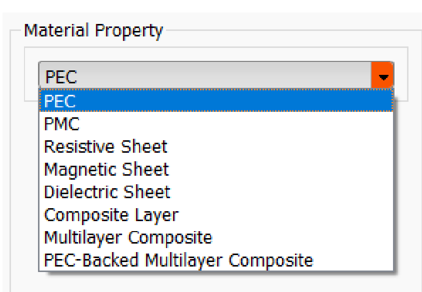
TaraNG Supports most of all useful electromagnetic properties like materials, boundary conditions, sources and excitations, lumped or distributed loading etc.

- **Sheet Material Supported:** PEC, PMC, Impedance Boundary Condition (IBC), Resistive Sheet, Dielectric Sheet, Magnetic Sheet, Multi-layer composite,
- **3D Materials Supported:** Simple Dielectric, Complex Dielectric (Loss Tangent), In-homogeneous dielectric, Magnetic Material, Conducive Materials, Directional Material Properties, Human Tissues, Soil and Nonlinear Material Models (Debye's & more)
- **Boundary Conditions:** PEC, PMC, Absorbing Boundary Conditions, Perfectly Matched Layer (PML/CPML), Periodic Boundary Condition, Symmetric Boundary Condition, Infinite boundary, Array Boundary Condition*
- **Lumped and Distributed Loads:** Resistor, Inductor Capacitor and Lumped Diode
- **Sources & Excitation:** Lumped Port, Delta Gap, Voltage Sources, Current sources, Waveguide port, Plane wave source, Point Source (E & H Field), Radiation pattern as source

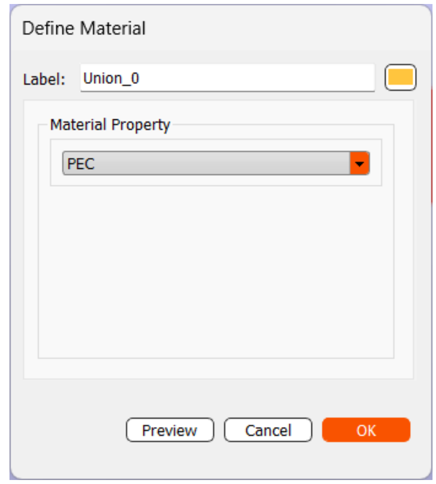
Assigning Surface Sheet Properties

- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose desired material definition



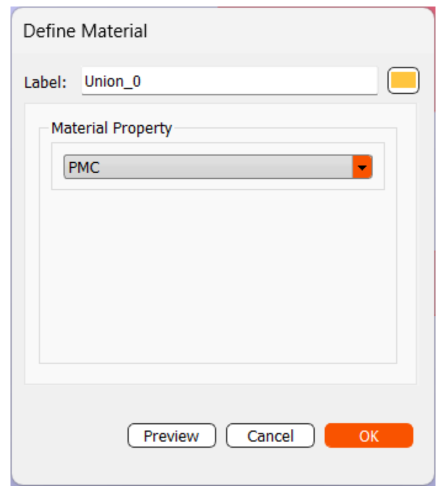


Assigning Perfect Electric Conductor (PEC):



- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose *PEC* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignBoundary(Type = 'PEC' )
```

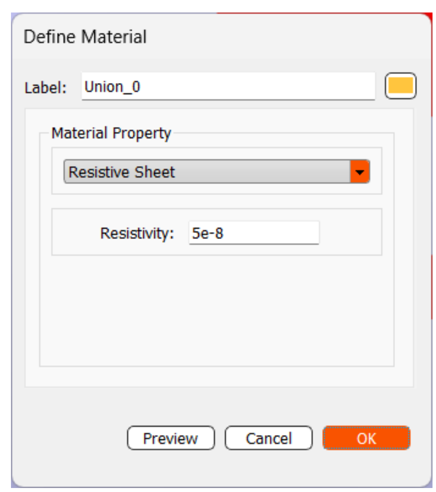


Assigning Perfect Magnetic Conductor (PMC):

- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose *PMC* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignBoundary(Type = 'PMC' )
```

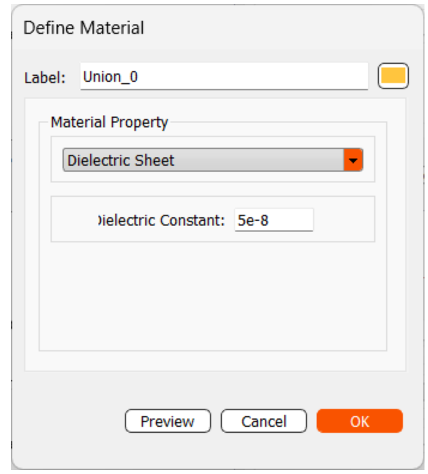
Assigning Resistive Sheet (RS):



- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose *PMC* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignBoundary(Type = 'Resistive Sheet', Parameters={'Resistivity': '5e-8'})
```

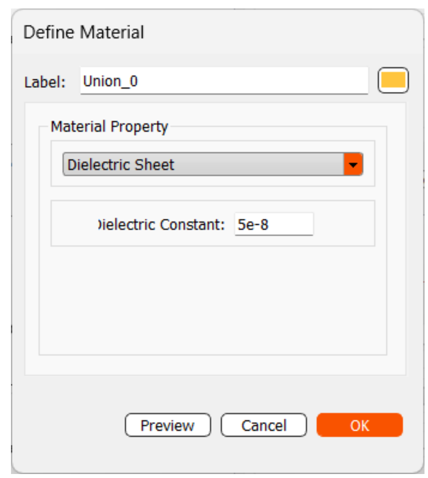
Assigning Dielectric Sheet (RS):



- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose *PMC* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignBoundary(Type = 'Dielectric Sheet',
Parameters={'Permittivity': '5e-8'})
```

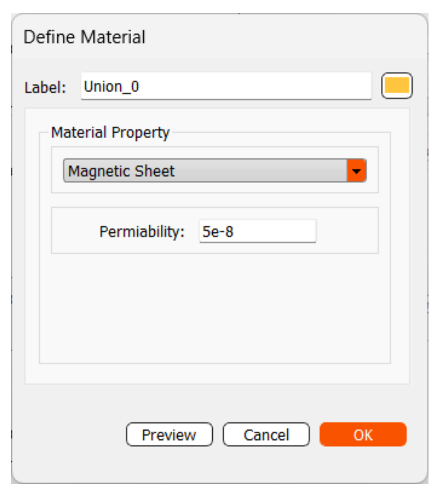
Assigning Dielectric Sheet:



- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose *Dielectric Sheet* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignBoundary(Type = 'Dielectric Sheet',
Parameters={'Permittivity': '5e-8'})
```

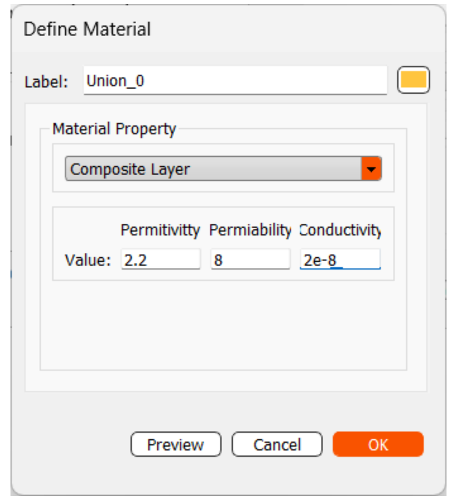
Assigning Magnetic Sheet:



- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose *Magnetic Sheet* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignBoundary(Type = 'Magnetic Sheet',
Parameters={'Permeability': '5e-8'})
```

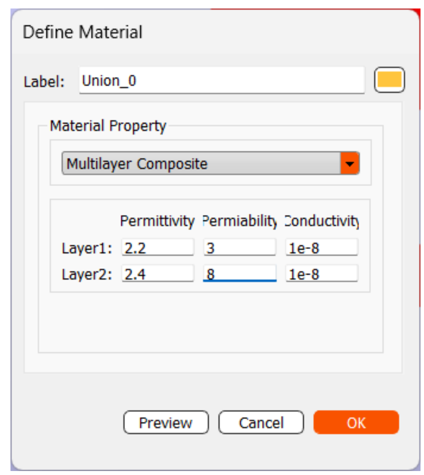
Assigning Composite Layer:



- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose *Composite Layer* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignBoundary(Type = 'Composite Layer,
Parameters={'Permittivity': '2.2', 'Permeability': '8',
'Conductivity': '2e-8'})
```

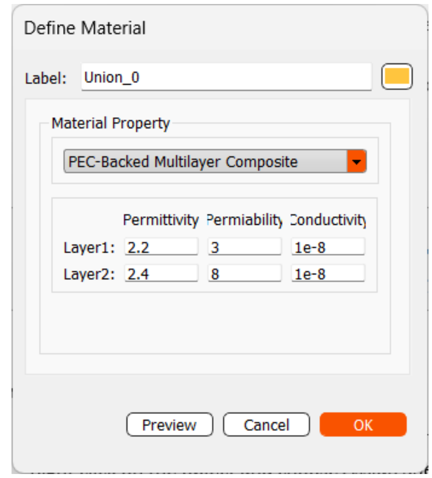
Assigning Multi-Layer Composite:



- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose *Multilayer Composite* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignBoundary(Type = 'Multilayer
Composite', Parameters={'Permittivity': ['2.2', '2.4'],
'Permeability': ['3', '8'], 'Conductivity': ['1e-8', '1e-8']})
```

Assigning Multi-Layer Composite backed with PEC:

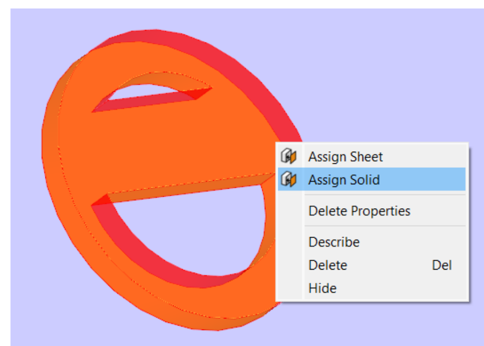


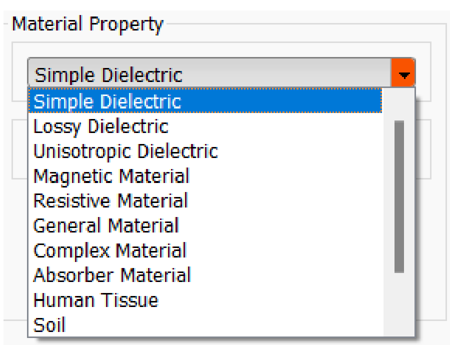
- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose PEC - Backed *Multilayer Composite* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignBoundary(Type = 'PEC – Backed
Multilayer Composite', Parameters={'Permittivity':
['2.2', '2.4'], 'Permeability': ['3', '8'], 'Conductivity':
```

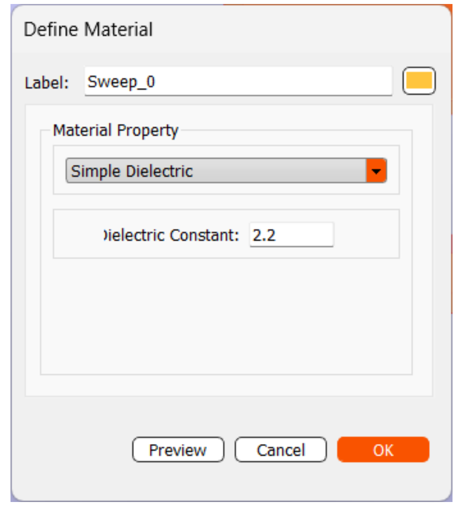
Assigning Solid Material Properties

- Select the object to which the material properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose desired material definition





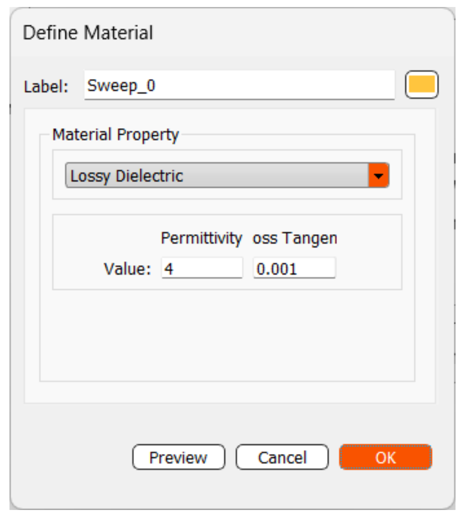
Assigning Simple Dielectric:



- Select the object to which the dielectric properties to be assigned.
- Right click on the object and choose *Assign Solid* from the context menu
- Choose *Simple Dielectric* from the drop-down menu in the *Material Property group box*
- Enter the value of *Dielectric Constant*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignMaterial(Type = 'Simple Dielectric',
Parameters={'Permittivity': '2.2'})
```

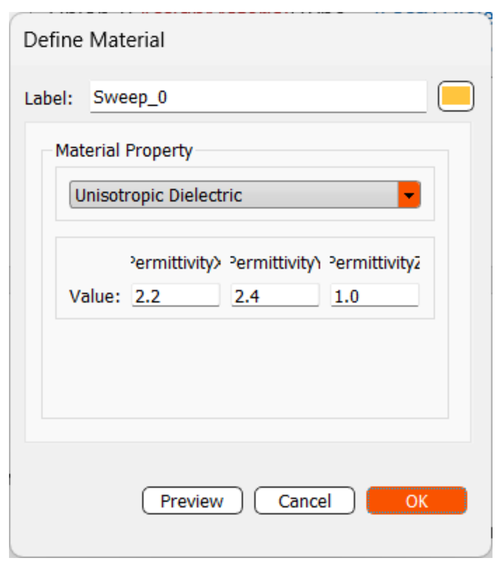
Assigning Lossy Dielectric:



- Select the object to which the dielectric properties to be assigned.
- Right click on the object and choose *Assign Solid* from the context menu
- Choose *Lossy Dielectric* from the drop-down menu in the *Material Property group box*
- Enter the value of *Dielectric Constant* and *Loss Tangent*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignMaterial(Type = 'Lossy Dielectric',
Parameters={'Permittivity': '2.2', Loss Tangent: '0.001' })
```

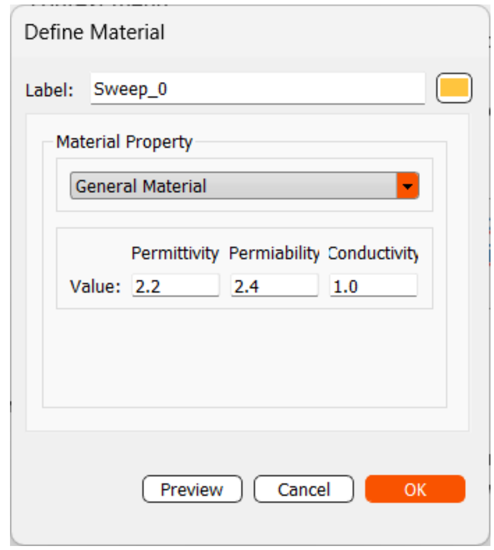
Assigning Unisotropic Dielectric:



- Select the object to which the dielectric properties to be assigned.
- Right click on the object and choose *Assign Solid* from the context menu
- Choose *Unisotropic Dielectric* from the drop-down menu in the *Material Property group box*
- Enter the value of *Dielectric Constant* in three directions
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignMaterial(Type = 'Unisotropic Dielectric',
Parameters={'PermittivityX': '2.2', 'PermittivityY': '2.4',
'PermittivityZ': '1.0', })
```

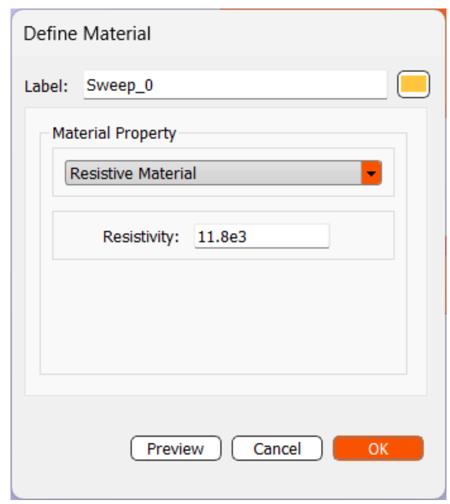
Assigning General Material



- Select the object to which the dielectric properties to be assigned.
- Right click on the object choose *Assign Solid* from context menu
- Choose *General Material* from the drop-down menu in the *Material Property group box*
- Enter the value of *Permittivity, Permeability and Conductivity*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignMaterial(Type = 'General Material,
Parameters={'Permittivity': '2.2', 'Permeability': '2.4',
'Conductivity': '1.0', })
```

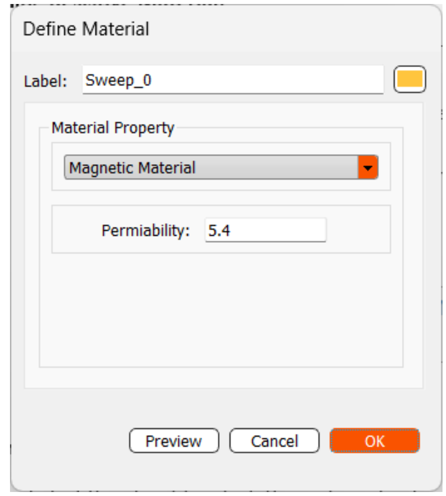
Assigning Resistive Material:



- Select the object to which the surface sheet properties to be assigned.
- Right click on the object, choose *Assign Sheet* from context menu
- Choose *Resistive Material* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignMaterial(Type = 'Resistive Material',
Parameters={'Resistivity': '11.8e3'})
```

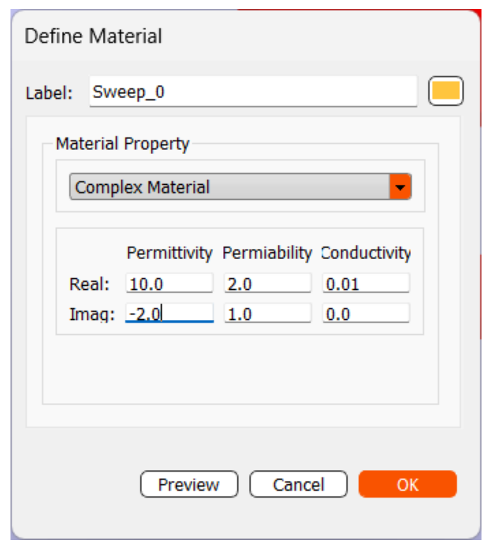
Assigning Magnetic Material:



- Select the object to which the surface sheet properties to be assigned.
- Right click on the object and choose *Assign Sheet* from the context menu
- Choose *Magnetic Material* from the drop-down menu in the *Material Property group box*
- *Click on OK*
- Below script will be auto-generated

```
Union_0.AssignMaterial(Type = 'Magnetic Material', Parameters={'Permeability': '5.4''})
```

Assigning Complex Material



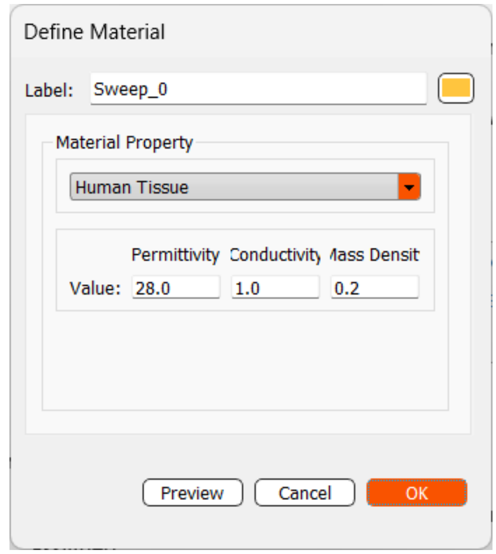
- Select the object to which the dielectric properties to be assigned.
- Right click on the object and choose *Assign Solid* from the context menu
- Choose *Complex Material* from the drop-down menu in the *Material Property group box*
- Enter the value of *Real and Imaginary of Permittivity, Permeability and Conductivity*
- Below script will be auto-generated

```
Union_0.AssignMaterial(Type = 'Complex Material, Parameters={'Permittivity': ['10.0', '-2.0'], 'Permeability': ['2.0', '1.0'], 'Conductivity': ['0.01', '0.0']}})
```

Assigning Human Tissue

- Select the object to which the dielectric properties to be assigned.
- Right click on the object and choose *Assign Solid* from the context menu
- Choose *Human Tissue* from the drop-down menu in the *Material Property group box*
- Enter the value of *Permittivity, Conductivity and Mass Density*
- *Click on OK*
- Below script will be auto-generated

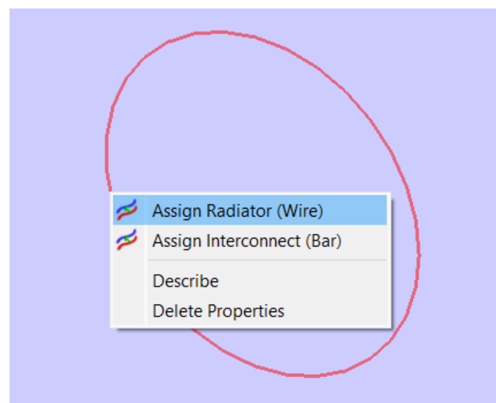
```
Union_0.AssignMaterial(Type = 'Human Tissue',
Parameters={'Permittivity': '28.0', 'Conductivity': '1.0',
'Mass Density': 0.2})
```

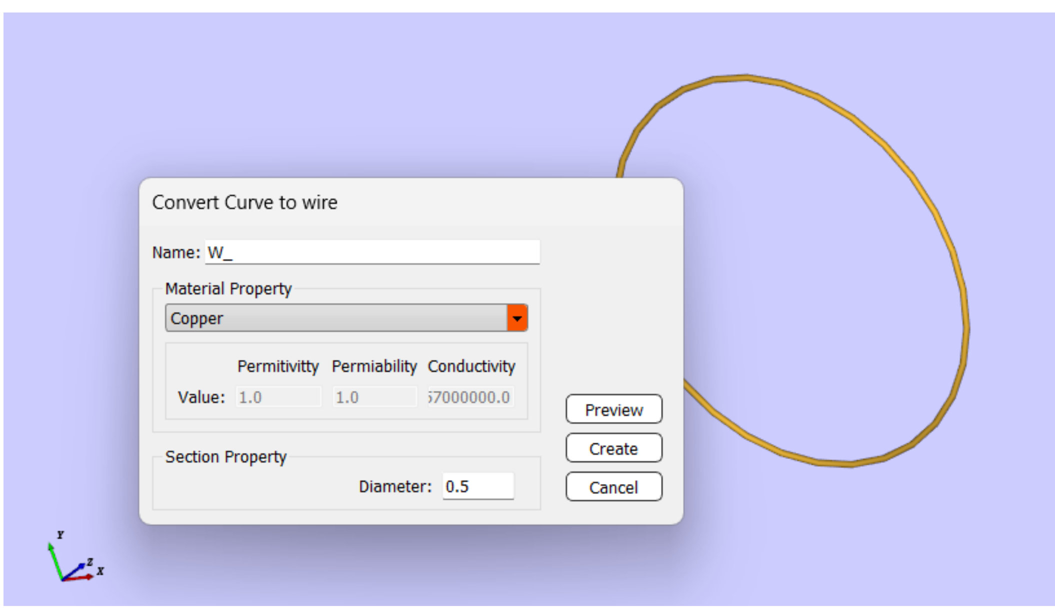


Assigning Wire Elements

The Curve entities such as Line, Spline, Polyline, Helix, Spiral, Circle etc. can be assigned as wires in TaraNG.

- To assign the wire properties right click on the curve entity and select *Assign Radiator (Wire)* from the context menu.
- The Wire window will appear
- Select the material from the drop-down menu. By default, the Copper material will be assigned
- Enter the diameter of the wire and click on the *Create* button

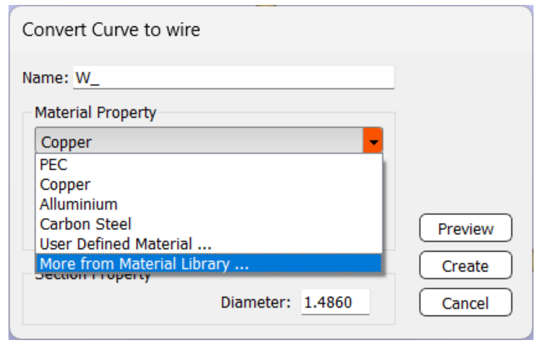




- The code will be automatically generated with the command as below

```
Circle_0.AssignWire(Mu=1.0, Eps=1.0, Sigma=57000000.0, Diameter=0.5)
```

Assign Material from Material Library



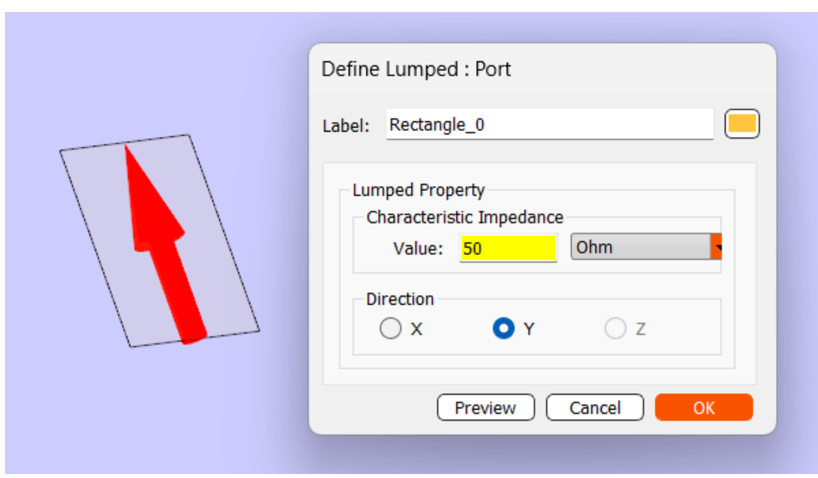
It is not always the case that user shall remember the values of the material parameters, hence TaraNG allows picking up the material from the datasheet table and assigning the properties accordingly.

User can choose the option *More from Material Library* and then choose the desired material from the Table

Assigning Excitation

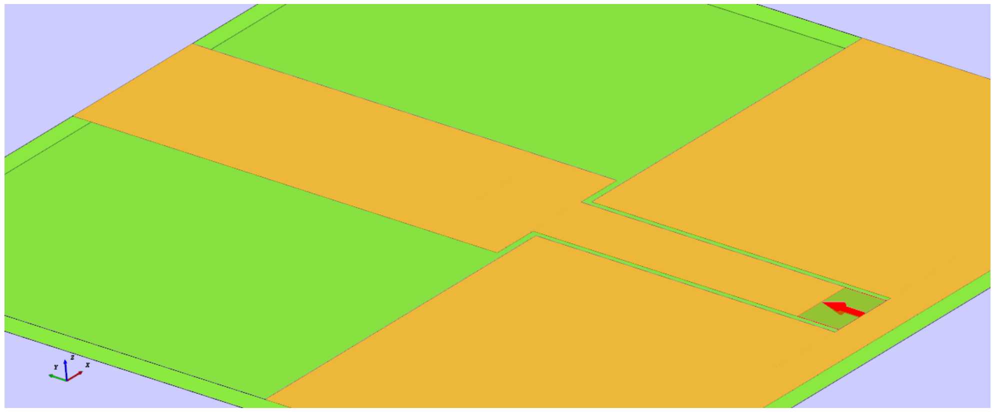
Please note that in TaraNG we can assign Lumped elements on a point of wire element or over a rectangular surface.

- To assign the Port excitation click on *Port*  tool in *Electromagnetics Toolbar*
- Click on two points of the rectangle in order to draw the integration line
- This integration line will be denoted by an arrow which depicts the direction of excitation
- After applying the excitation doble click on the rectangle on which the excitation is applied



- Now the Lumped Port window will appear and we can change the value of Characteristic Impedance default value is 50 ohm and the *Direction* of excitation

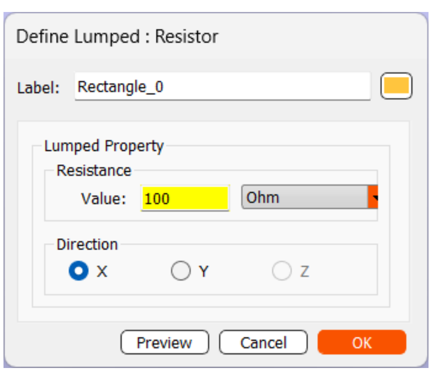
```
Rectangle_0.AssignLumped(Type='Port', Dir='Y', Value=50)
```

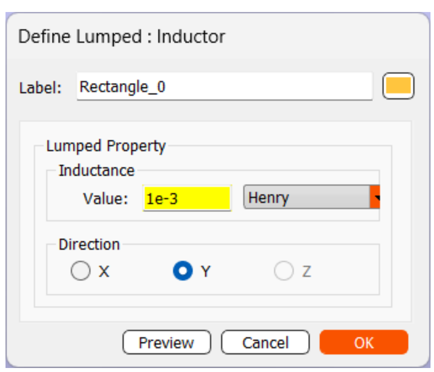


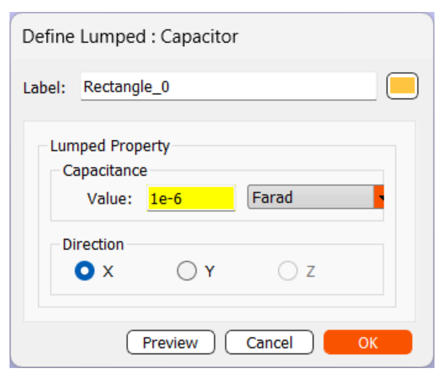
Assigning Loads

The process of assigning the loads is exactly similar to the process of assigning *Port* we can assign Lumped elements by clicking on the respective load icon in the *Electromagnetics Toolbar*

- To assign the Resistor or Inductor or Capacitor click on  tool respectively in *Electromagnetics Toolbar*
- Click on two points of the rectangle in order to draw the integration line
- This integration line will be denoted by an arrow that depicts the load connection
- After applying the load doble click on the rectangle on which the load is applied
- Doble click on the Load to change its properties







- The code generated for the lumped load is as below

```
Rectangle_0.AssignLumped(Type='Resistor', Dir='X', Value=100)
```

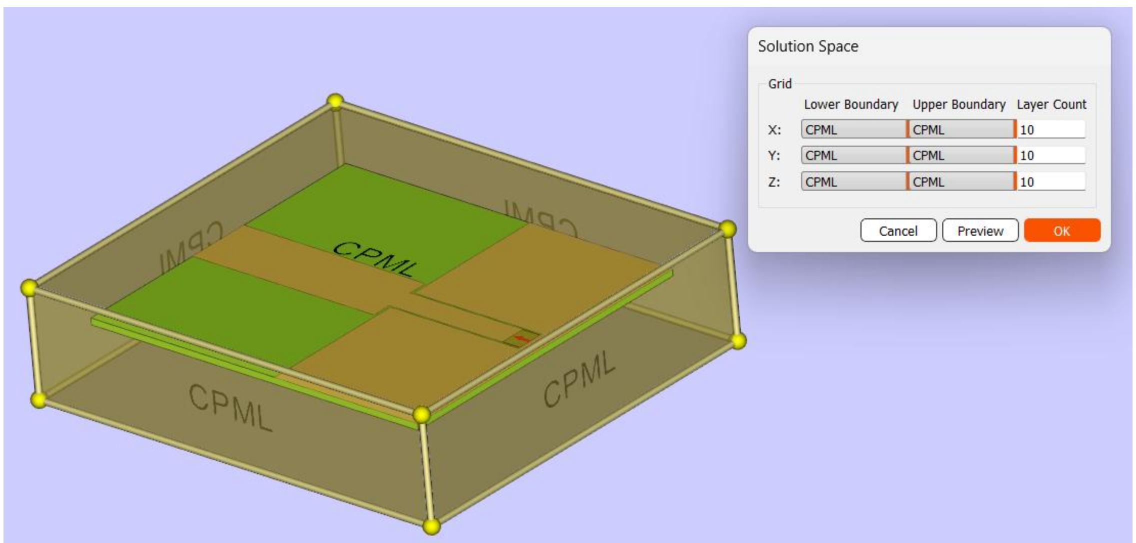
```
Rectangle_0.AssignLumped(Type='Inductor', Dir='Y', Value=1e-3)
```

```
Rectangle_0.AssignLumped(Type='Capacitor', Dir='X', Value=1e-6)
```

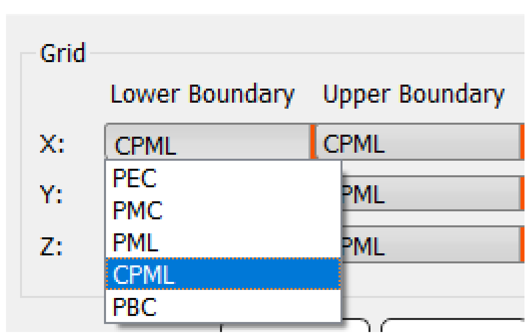
Assigning Boundary Conditions

TaraNG supports different types of boundary conditions that truncates the simulation domain, each wall of the simulation box can be assigned with different type of boundary condition

- To assign the *Boundary Condition* click on tool in *Electromagnetics Toolbar*
- By default, all the sidewalls are assigned as CPML Boundary with 10 layers



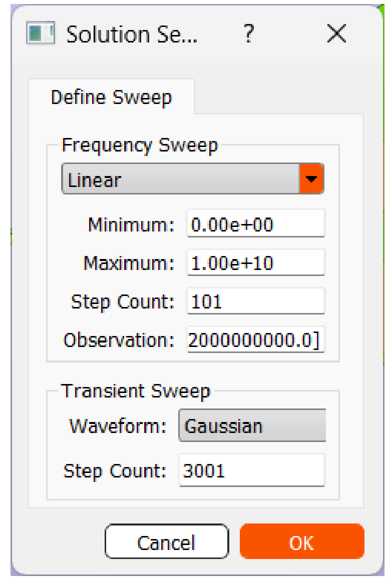
- User can change the layer count for CPML or PML boundary
- User can also set different boundary conditions such as PEC, PMC, PML, CPML and Periodic Boundary Condition (PBC) as required



Defining Solution Setup

For simulating a particular problem, we need to first define the frequency sweep i.e the frequencies of interest to solve a problem. To define Solution Setup,

click on  icon in the *Electromagnetics Toolbar*



- **Assigning Frequency Sweep:** Choose the frequency sweep as *Linear* or *Logarithmic*
- Enter *Minimum* and *Maximum* frequency and the *Step Count*
- Define the observation frequency for obtaining near and far field, radiation patterns
- **Assigning Transient Sweep:** Choose the transient waveform as *Gaussian, Modulated Gaussian, Square wave etc.*
- Enter the number of time steps *Step Count*
- Click on *OK* button

After defining the Solution Space next step will be to performing meshing and running the simulation using appropriate solver. The voxelization and meshing is discussed in the next chapter.

Chapter 4: Performing Voxelization

TaraNG has built in voxelization and meshing algorithms that can generate the high-quality uniform and non-uniform meshes using adaptive mesh elements. However, user will also be given a choice to control the mesh parameters

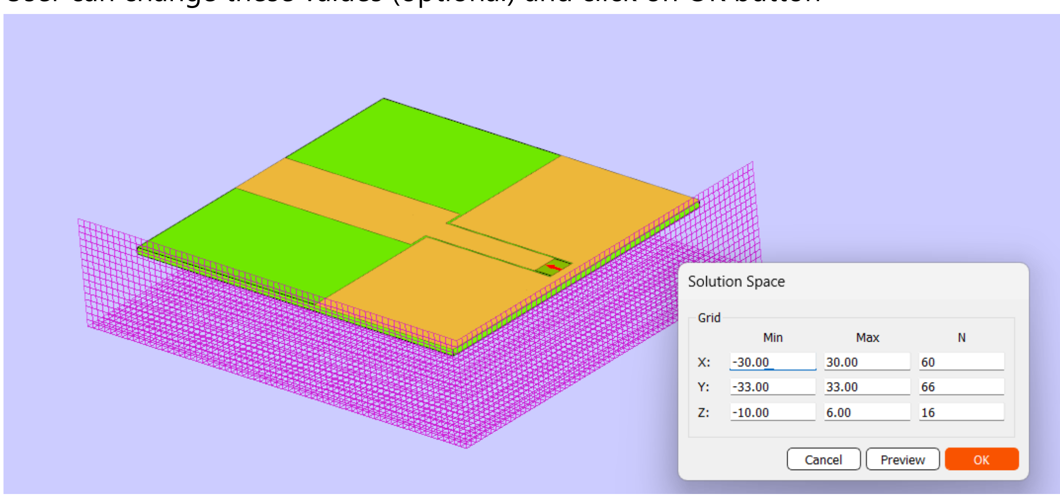
The Yee cells type meshing is used, these meshes allow for the interleaving of unknown nodal values of electric and magnetic field components, resulting in an explicit algorithm that is second-order accurate in both space and time.

The general guideline for meshing is that the cell size should be 20 times lesser than the minimum wavelength of the signal and the airgap towards the boundary needs to be around 10 cells for the case of CPML boundary.

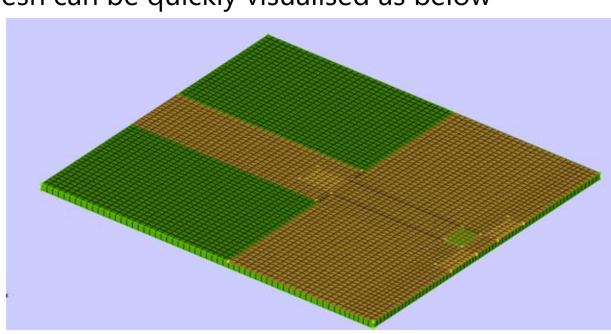
Performing Voxelization

TaraNG automatically sets the rules for meshes and suggests the lower and upper boundary and the number of cells in each direction.

- To perform meshing click on  tool from the meshing toolbar
- The Solution Space box will appear which will include pre-computed values
- User can change these values (optional) and click on OK button



- The generated mesh can be quickly visualised as below



Chapter 5: Performing Computation

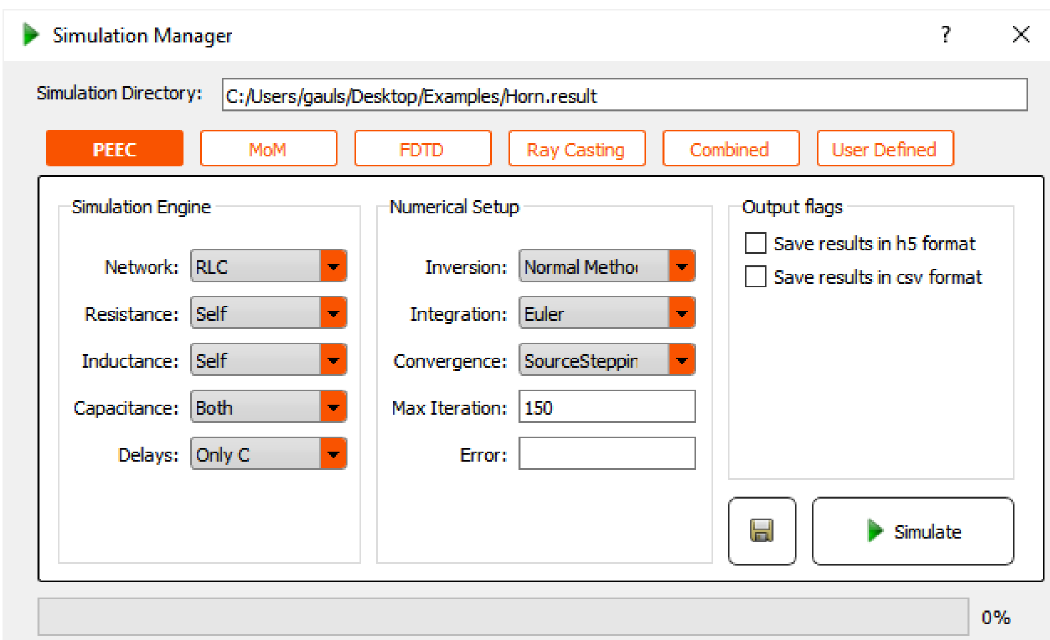
Good specialist computational electromagnetics (CEM) software should mostly have similar User Interface (UI) and CAD capabilities. It is mainly the CEM techniques on which these codes are based which differentiate them. This differentiation is then also less on which is "better" but more on which is "more applicable for a certain problem type".

TaraNG includes time domain and frequency domain solvers which are suitable for different types of EM problems, those method includes PEEC, MoM, FDTD and Physical Optics.

Simulation Manager for PEEC

Simulation Manager enables analysis of problem using different computational methods. Here we will choose PEEC as our simulation Engine.

- Before simulating the problem make sure you have saved the design.
- To solve the design using PEEC method user can make desired topology setup and click on Simulate button.
- The solver will generate the equivalent circuit and then solve it for different frequencies as specified in the solution setup.

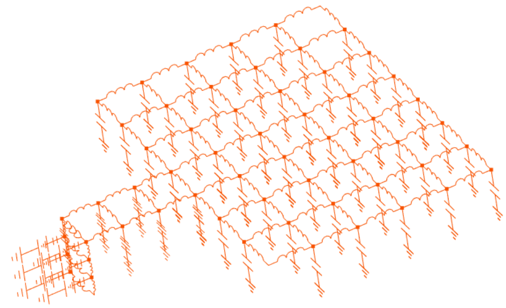


What is unique?

- Adjustment of maximum internal setting (along with auto-recommendation)
- PEEC has its own benefits like circuit extraction
- Un-attended simulations with integrations (batch mode simulation)

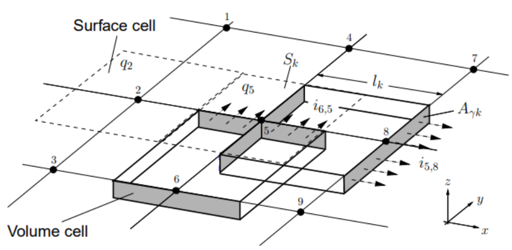
Partial Electrical Equivalent Circuits (PEEC) generates the circuit model of the design prior to simulation. This circuit models are used for analysis to obtain final results. TaraNG also includes different types of topologies. What are benefits?

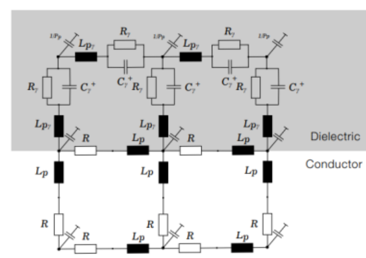
- ✓ Both time and frequency domain analysis using single method
- ✓ DC to daylight spectrum for simulation, as DC solution of circuit exists
- ✓ Integrated with any circuit component externally
- ✓ Adding components at desired point or location
- ✓ Understand physical behavior of the designed problem



TaraNG PEEC Features:

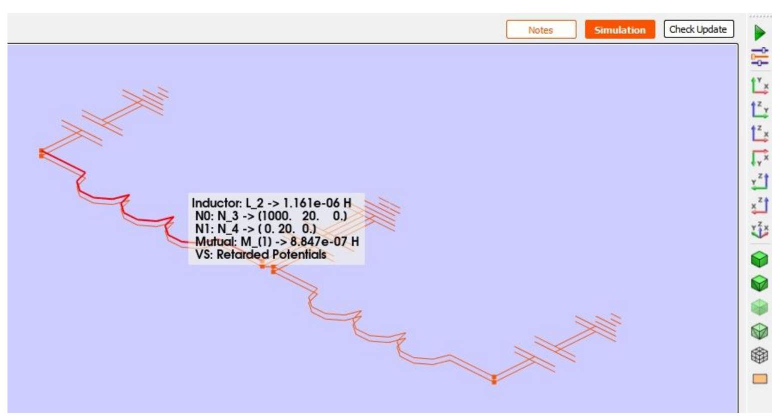
1. **Equivalent Circuit Modelling of EM Structures:** Divide the structure into multiple segments called filaments using appropriate meshing technique. Compute inductive and capacitive interactions between those filaments with self-mutual partial inductances and partial potential coefficients.



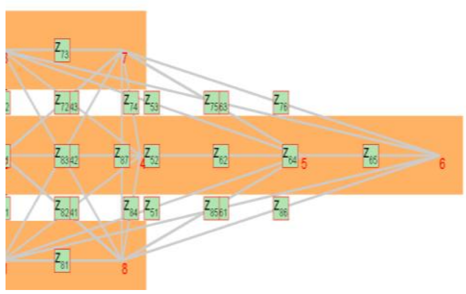


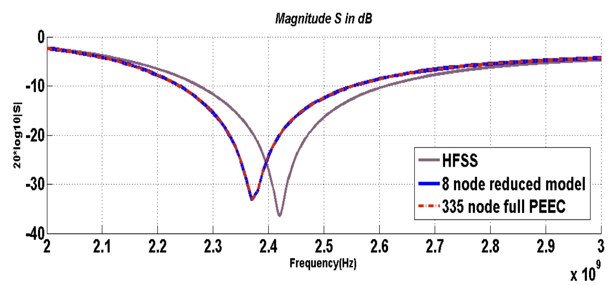
2. **Visualizing the equivalent circuit in 3D environment:** Visualize the complete equivalent circuit model in 3D space, with appropriate locations of circuit components in accordance with the model geometry.

3. **Annotating and interacting with circuit elements:** Labelling (annotation) values of different component values and mutual interactions. Selecting each component, doble click over it to extract the details. Rotate, Zoom and Pan the 3D circuit model.



4. **Modelling of exponential green's function for network topology:** For high frequency electromagnetic designs the exponential terms in green's function has significant role, those terms can be modelled using time delays between the mutual coupling i.e. retarted RLC topology (RLCt) for full wave analysis.
5. **EM characterization of Structures into different Topologies**: For 3D electromagnetic structures generate equivalent circuit with desired topology viz. RC Network Topology, LC Network Topology and RLC Network Topology using partial elements. The user will have a choice on which circuit topology to be extracted like RC topology, RL topology, LC topology, RLC topology or full wave modelling of exponential green's function into RLCt topology
6. **Model order reduction (MOR):** In general, the circuit size is larger when we model a high frequency electromagnetic design using PEEC, which becomes overwhelming for designer. Using network reduction approach like star to delta conversion the unwanted circuit nodes can be eliminated. which reduces the order of the circuit model to make the circuit model more insightful and easier to understand.
7. **Node definition for MOR:** In order to reduce the circuit model, the software allows unwanted nodes removal by defining the wanted nodes or the nodes of our interest to be kept for generating the equivalent circuit model between given node location. The facility for defining such nodes on a 3D environment is made easy.





8. **Pole-zero extraction**: Pole zero analysis is useful for stability analysis and response studies. By extracting the pole-zeros and considering only dominating poles or zeros can also reduce the order of the circuit model.
9. **Circuit matrix in tabular format:** It is easy to extract all necessary circuit matrices viz Partial Inductance matrix, Partial capacitance matrix (potential coefficients), Resistance matrix, Retardations with self and mutual coupling component values and plotting options. These matrices can also be exported to excel or csv format.
10. **Exporting the circuit netlist to Hspice format:** Exporting the generated circuit model to compatible spice format for further processing the circuits or adding new circuit components like transistor or so on to run third party simulation.
11. **Third party component insertion:** The 3D circuit UI allows adding or inserting external component between node point pairs at given location to define connection just by picking up the desired nodes.
12. **Time Domain and Frequency Domain analysis:** As the circuits can be simulated both in time domain and frequency domain, it's easy to analyze the EM structure after converting to equivalent circuit topology.

Chapter 6: Result Visualization

After performing the simulation, the next step is to visualize the results and understand various parameters of design. Now, by looking at the results we can check if design goal is achieved or not, If not then what are the further optimization required, we can also understand physical behavior of the system by looking at equivalent circuit.

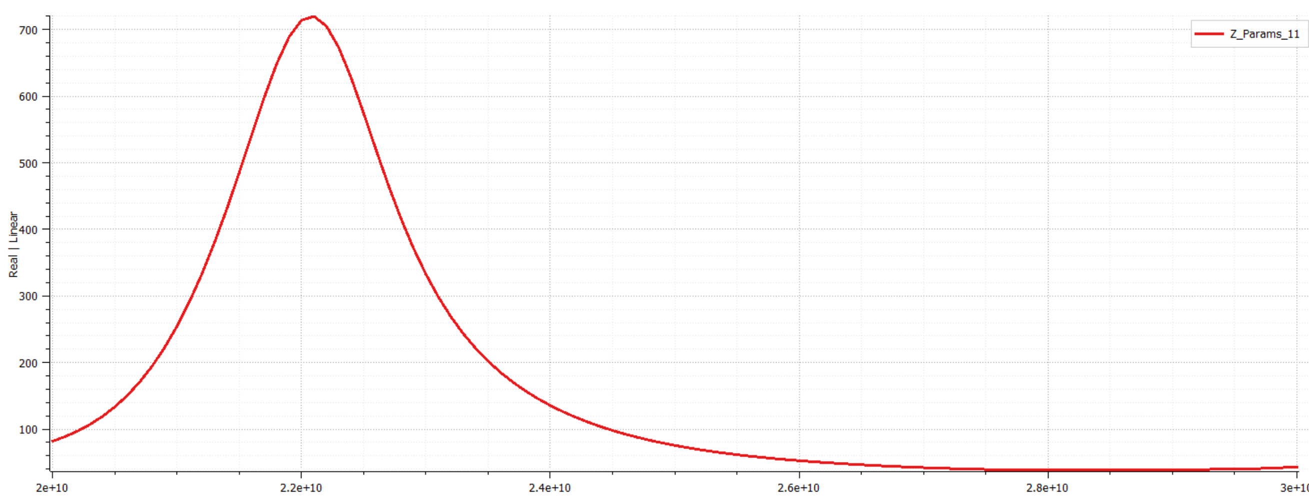
TaraNG offers all necessary 2D and 3D results plots with the necessary controls like pan, rotation, zoom and animations, export options are available. Here is the list of processing features

- 3D equivalents circuit extraction and components annotations
- 3D display of models with superimposed results, e.g. far-field, near-field, surface currents on structures, iso-surfaces for near-field plots, location of peak localized SAR cubes.
- 2D result plots in either linear or polar plots, e.g. Smith charts for input impedance, polar plots for RCS, linear plots for S-parameters, VSWR, radiated power etc.
- Real-time updating and visualization of parametric variations (tunning).
- Radiation pattern (3-D in model, 2-D XY/polar)
- SAR (IEEE standard compliant whole-body average, 10g &1g cube localized)
- Full multi-port S-Parameter extraction and touchstone file export in standard format.
- Several visualization options for surfaces, incl. iso-surfaces, 2D field cuts and 3D contours, streamlines, quivers.
- Multiple results displayable in same viewport for comparison.
- Export views to image formats and web based virtual reality formats.

Visualizing Results in 2D

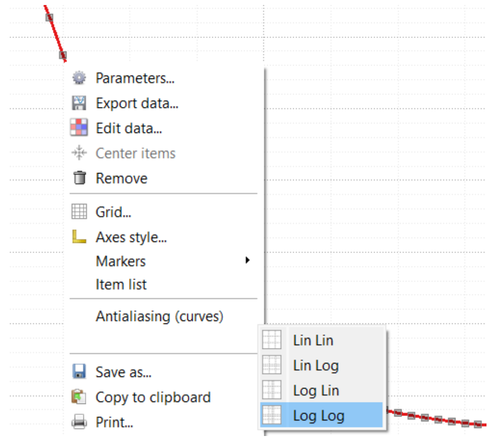
After performing analysis, we can plot the results 2D space (X and Y axis). Such 2D results can be different Port Parameters, VSWR, Power densities that can be plotted by clicking on the tool,


in the result visualization tool bar.



We have different useful 2D Plot Features that can be accessed by right click on object to like

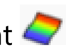
- Adding Marker
- Changing Labels
- Changing Annotations
- Changing Fonts and color of plot
- Exporting plot formatting in csv format
- Visualizing data in table format
- Saving data in image format
- Changing axes style
- Data scaling to linear/Logarithmic
- Adding computatin domain

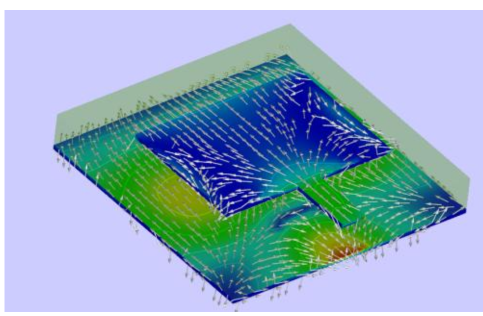
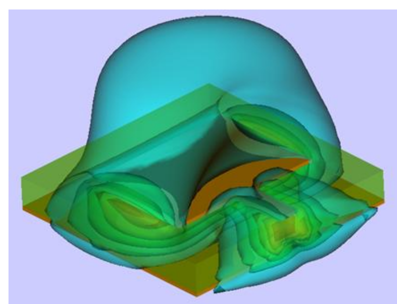


Visualizing Results in 3D

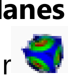
TaraNG offers result visualization and animation in 3D. Using 3D results we can understand vector fields around the problem. These plots include Streamline, Contour, Quiver/Glymph, Cut-plane etc.

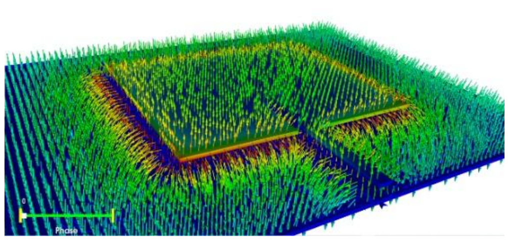
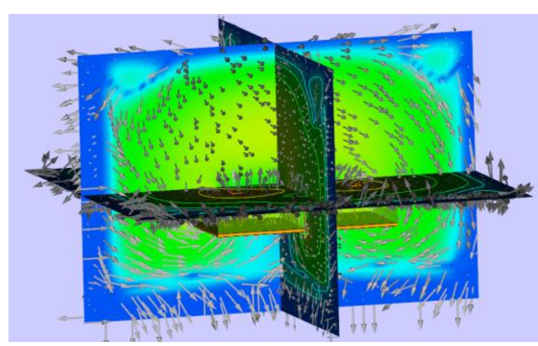
Drawing Surface Currents

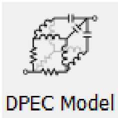
To draw surface currents, click on Surface current  tool from the result toolbar

Drawing Field Contours, Quivers and Cut-planes

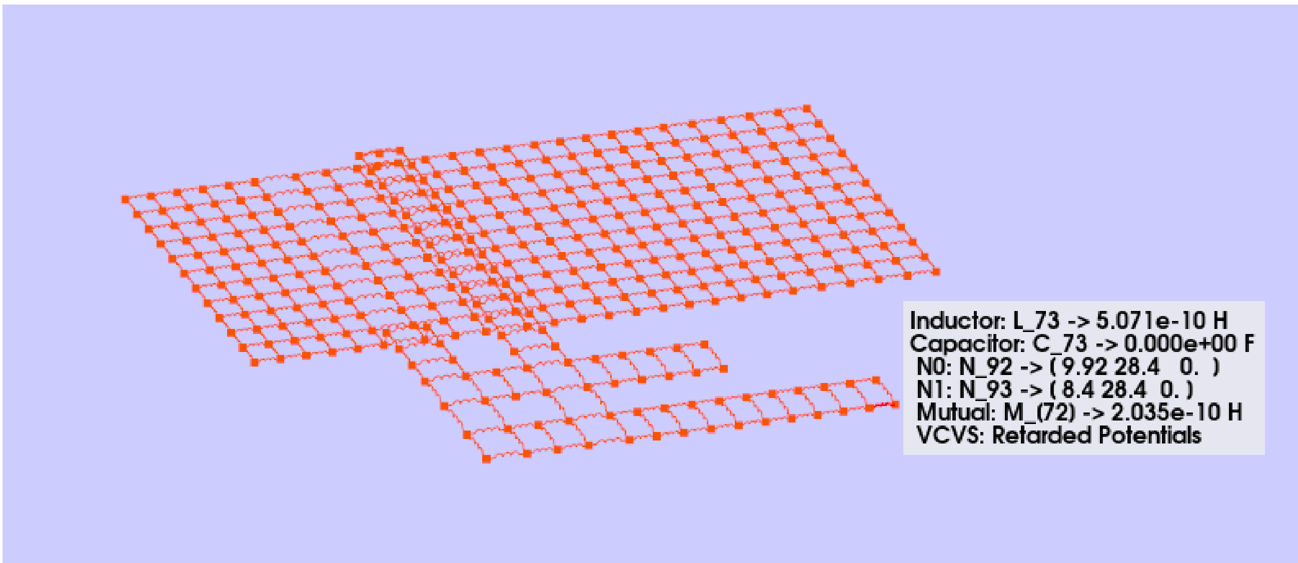
To draw surface currents, click on Field contour  tool from the result toolbar, we can also change data type (electric or magnetic field) and the scale of data.



Visualizing 3D PEEC Circuit

To visualize the equivalent Partial Electrical Equivalent Circuit user can switch to PEEC mode this will show the PEEC Circuit in 3D space that will comprise of Partial Inductors and Capacitors. User can howe on the particular component to know its value.



Utilities

- Optimization: Interfacing facility with Particle Swarm Optimization (PSO) code
- Fourier Transforms: Use Fourier Transforms to get time domain information from frequency domain simulations and vice-versa
- Curve fitting and interpolation: TaraNG uses curve fitting to model and interpolate the parameters of interest, for example the input impedance, S-parameters, currents, far- and near-fields, etc. Reduce heavy dataset into small equation weights that can be used for data training and machine leaning.
- Access to intermediate data tables and workspace.
- TaraNG has capability to provide interface with different open-source libraries such as TensorFlow, Keras etc. which are necessary to perform data driven techniques

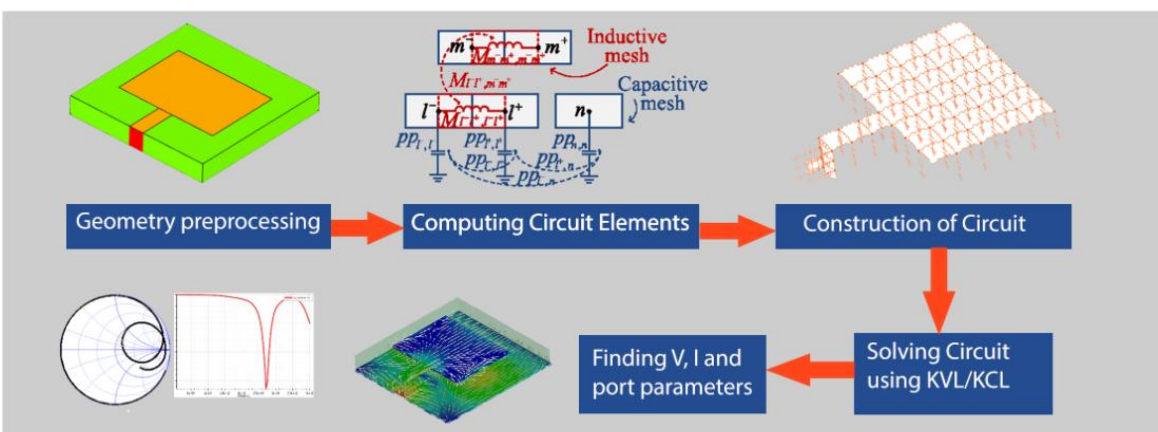
Chapter 7: A Note on Solvers

The three important techniques within the area of computational electromagnetics are the finite element method (FEM), the moment method (MoM), and the finite difference (FD) method all have their strengths and weaknesses. The FEM and FD methods are both based on a differential formulation of Maxwells equations making them feasible for complex geometries and materials. In contrast to the FEM and FD methods are the MoM and PEEC integral based formulations useful for open-air problems.

The main feature with the PEEC method is the transformation of the electromagnetic problem to the circuit domain. This simplifies the extraction of equivalent circuits for discontinuities and offers the solution both in the time and frequency domain for the same equivalent circuit

As the density, complexity, and speed of VLSI circuits are continuing to increase, the management of on-chip interconnects becomes of major concern to the I.C. designer. When group of conducting bars are kept together then field coupling occurs which may result in induced field on nearby bars or interconnects. The effects like proximity effect, Skeen effect need to be accurately modelled while designing VLSI interconnects or IC packaging. This module allows to extract these parasitic effects and tells behaviors. Although circuit performance can always be evaluated by a circuit solver like SPICE, this is usually very slow and fails to reveal relationship between interconnect structure and interconnect delays.

TaraNG PEEC: This module allows on-chip interconnect wires to be considered in circuit analysis for high precision analysis. The goal of this module is to include interconnect parasitic in a circuit simulation as efficiently as possible, without significantly compromising accuracy. The range of applications covered by this module starting from computing flux linkages of circuit components to simulating radiation effects from various Antennas.



- Inductors and Transformers design**:** Inductors such as solenoid, cylindrical coil, toroidal core, flat spiral, multilayer air core coil. Air core transformer, toroidal transformer, center tap transformer, polyphase transformer.
- Capacitor design: Capacitors such as interdigital capacitor, cylindrical capacitor.

- High voltage Application: Design of air core reactor.
- High frequency Microwave Circuits: Design of multi-port passive circuits such as filters, couplers, power divider and modelling and analysis of junctions and discontinuity.
- EMI/EMC: Calculation of induced field due to external sources, mutual coupling between the objects.
- Electronic IC packaging: Signal Integrity & crosstalk analysis.
- VLSI Interconnect: High frequency parasitic calculation, 3D inductive and capacitive coupling.
- Power Electrical and drives: High voltage application, design of reactors, solenoid coils etc.
- Antennas and Radiation Effects

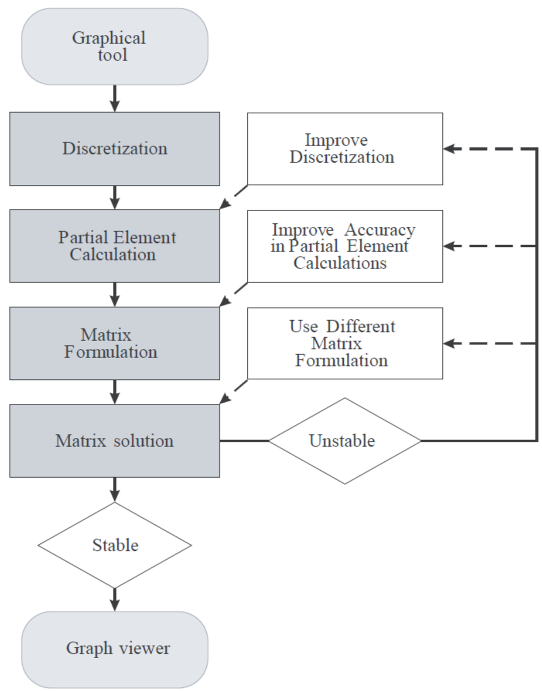
PEEC Workflow:

a) The problem is discretized in different cells called as filaments

b) The partial inductances and capacitances and the associated mutual couplings are calculated for each filament

c) The equivalent circuit is constructed using those inductors and capacitors

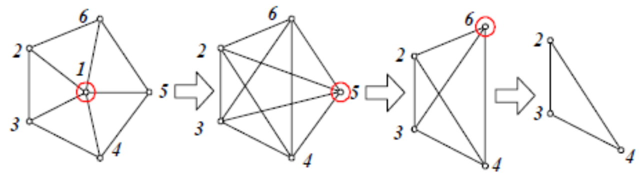
d) This circuits can be solved in time domain or frequency domain to find the different parameters



Derived Physically Expressible Circuit

With the fast development of low-loss and high-density integrated packaging technologies such as system-on-package (SoP) is considered as one of the most promising solutions for integrated electronic systems and wireless products. TaraNG is a computer-aided design (CAD) tools based on an algorithm that can systematically generate a physically meaningful circuit model for large-scale embedded RF passives and interconnection traces.

The needs for such a tool mainly come from two aspects, which are: 1) a co-simulation of a mixed-signal system which are modeled by a circuit simulator and passive circuits, which are modeled by electromagnetic (EM) simulation and 2) the prediction of electromagnetic interference (EMI) among the RF passive circuits.



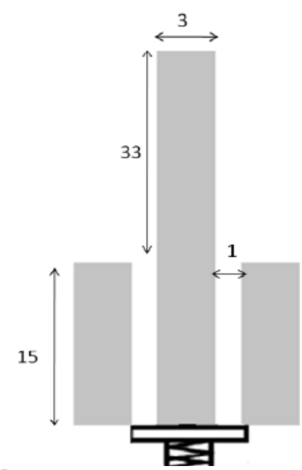
. The process of DPEC for a six-node network.

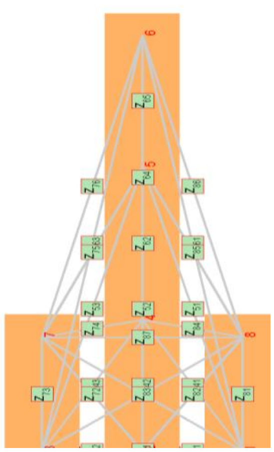
A number of techniques for extraction of lumped element equivalent circuits for embedded RF passives have been developed over the past years. The most popular technique is to construct an equivalent-circuit model based on a predefined circuit topology from physical intuition. The component values of the circuit model can be determined by empirical formulas or by curve fitting. The PEEC model can generate a frequency-independent circuit model containing all self and mutual capacitance and inductance, can be regarded as a primitive equivalent circuit, the number of the circuit elements is excessive to handle for practical cases and the capacitors and inductors in the model do not have apparent physical meanings.

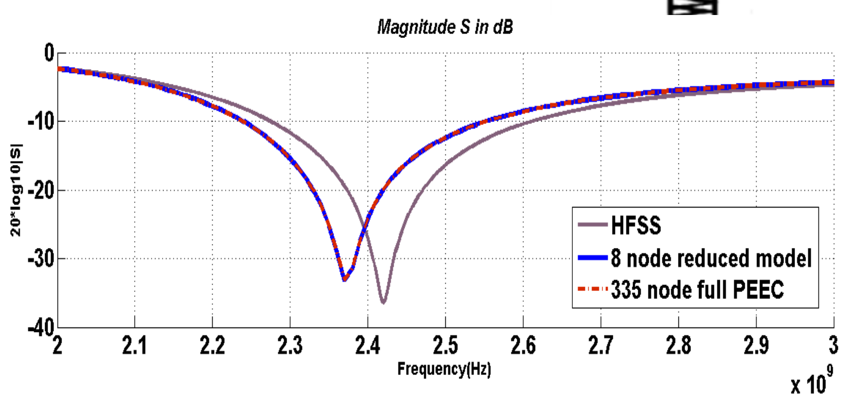
Simulation Example using PEEC & DPEC:

1. A CPW fed monopole antenna is considered for this example, Partitioning scheme (+PEEC) is applied on generated mesh elements with the common nodes are combined. Thus, at first level the circuit is converted to 335 nodes.

2. At second level of reduction the approach of Model Order Reduction (MOR) is applied







The model is reduced for 8 nodes using DPEC and the comparison is made for the PEEC and DPEC. It can be noted that both the results are well in agreement with existing commercial package.

Examples

1. Half-wavelength Dipole Antenna

This example demonstrates how to compute the input impedance and radiation pattern (Directivity and Gain) for a lossless half wavelength dipole antenna. The length of the dipole is 200 cm and it is a half wavelength long at 75 MHz. The dipole is located along the z-axis and has a radius of 1 cm. The excitation used in the analysis is a voltage source with a voltage applied at the midpoint of the dipole. The analysis is done for a linear frequency interval of 101 frequencies from 25 MHz to 500MHz.

Geometry Design

Dipole length plays an important role while designing an antenna. The length of this antenna is equal to half of its wavelength ($l = \lambda/2$ or $\beta l = \pi$). According to the formula,

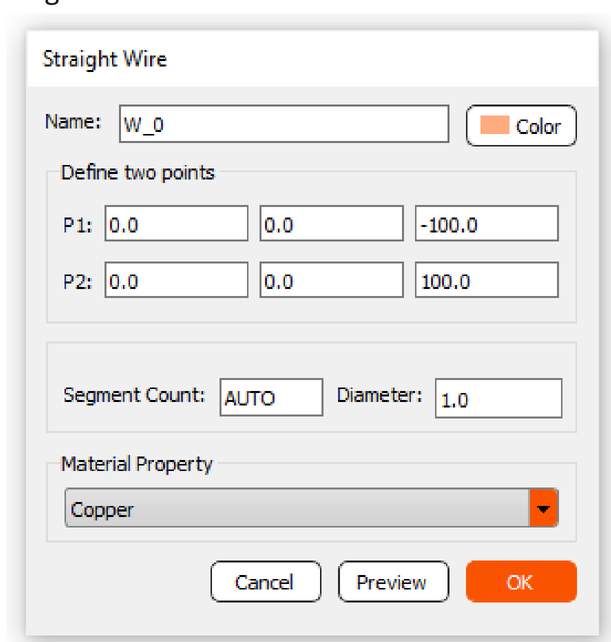
$$f = c / \lambda$$

Where, f is frequency in Hz, c is speed of light (3 x 10^8 m/s) and λ is wavelength

To find the length of the half-wavelength dipole antenna at frequency 75MHz

Length = $\lambda/2 = (c/f)/2$ = 3e8/75e6/2 = 300/75/2 = 2 meter = 200 cm

1. Initially, click on the 'Wire Segment'  icon in the toolbar. A wire appears on the screen with a dialogue box shown in figure 1.3

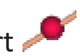


For the length 200 cm along z direction, we will use two points P1 = (0,0, -100) and P2 = (0,0,100) respectively.

2. After defining start and end coordinates of wire we will define Diameter as 1.0 cm and set the Material Property as copper for this example. In this experiment, we have used copper as the material however these properties can be changed per the requirement of the user.
3. After the details are entered, clicking OK will show the dipole antenna wire. In the main 3D Design window of the software.

Setting up the Excitation

The dipole is a balanced antenna, that is, neither of the two parts or sides of the antenna is connected to the ground. The next step is to add the antenna feed in the center of the dipole wire.

1. To add the feed, click on the 'Port'  icon in the taskbar and slide the cursor near the dipole to get the visuals. By clicking anywhere on the dipole, the feed will be placed.
2. To make sure the port is exactly at the center of the dipole, double click on the port  and make X, Y and Z as 0.0, 0.0 and 0.0 respectively
3. Click OK to complete the step. The dipole antenna with these easy steps has been created using TaraNG Software.

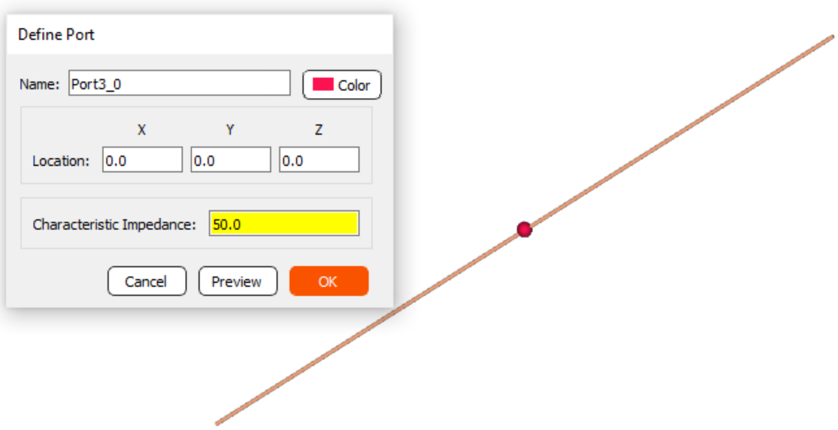
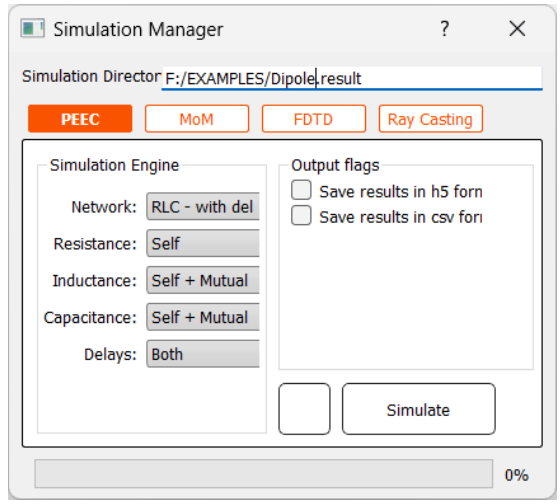


Fig. Feeding the Dipole Antenna

4. After the successful creation of the dipole antenna and applying excitation, the next step is to run the simulation to perform the computations or analyses.

Running the Simulation



1. To run the simulation, click on the Run button  in the taskbar.
2. Setup Analysis settings are useful for setting up solver method and inversion methods. We will keep these parameters as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.

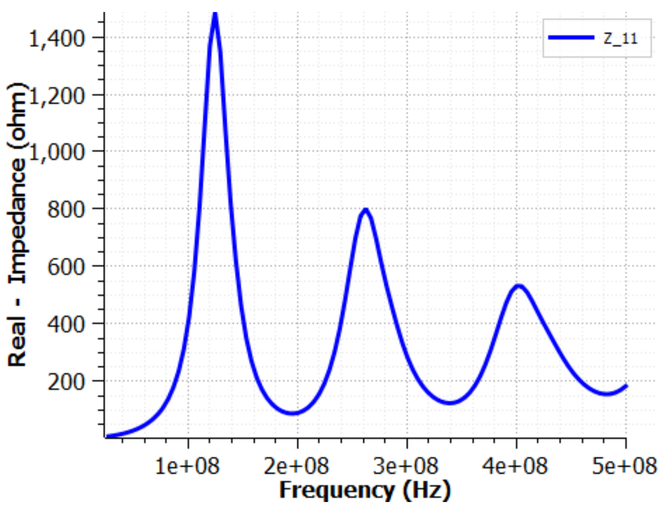
Visualize the Results

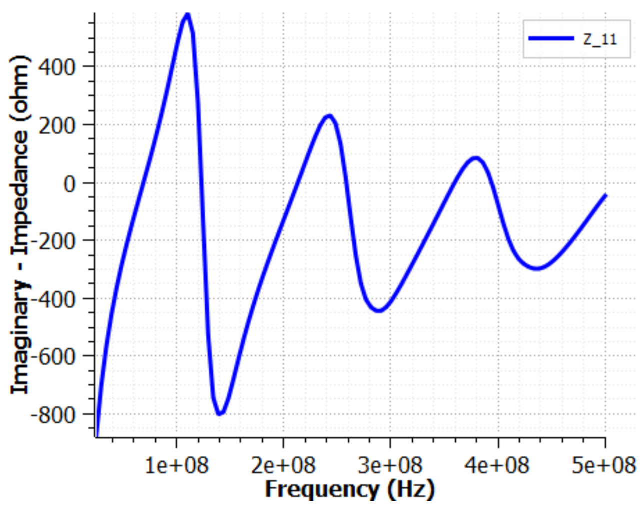
Clicking on the bar to the right shows the required plots.

A) Port Parameters

Plotting Input Impedance: Clicking on the right bar shows the real part of the input impedance 

The very next tab shows the result of the imaginary part of the input impedance 

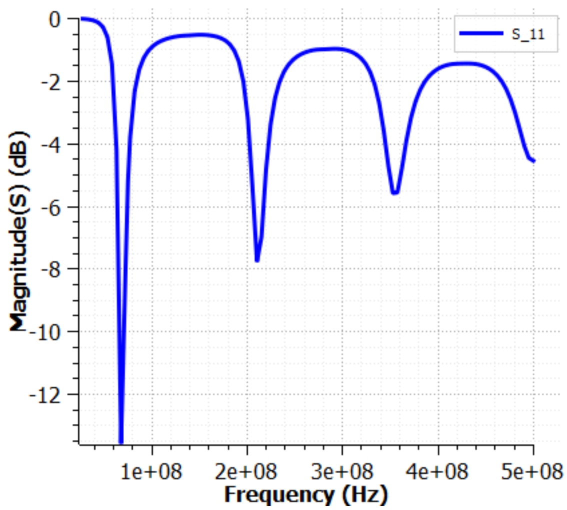


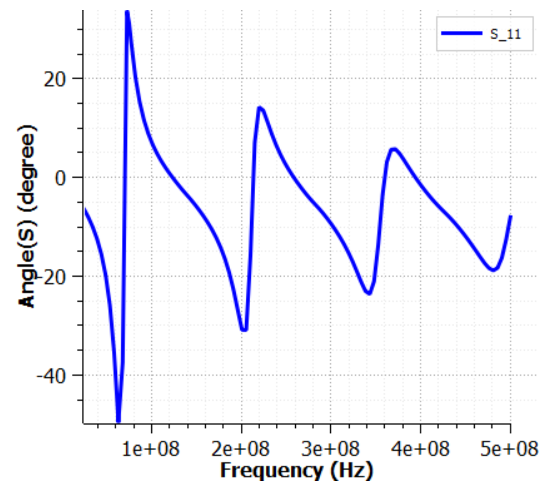


Here we can see that the antenna shows capacitive behavior at the lower frequencies. At a particular frequency, it changes from capacitive to inductive. This is the resonant frequency when the reactance is zero. This behavior changes between capacitive to inductive and so on.

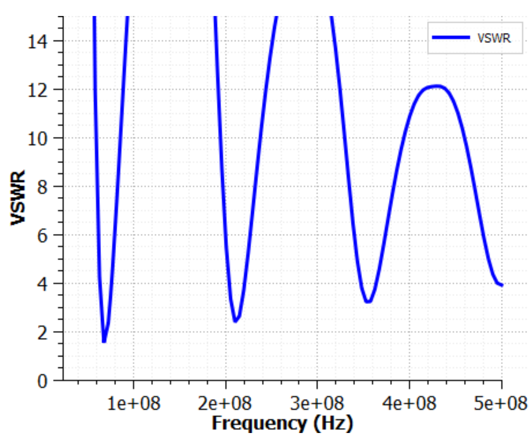
Plotting Scattering Parameters:

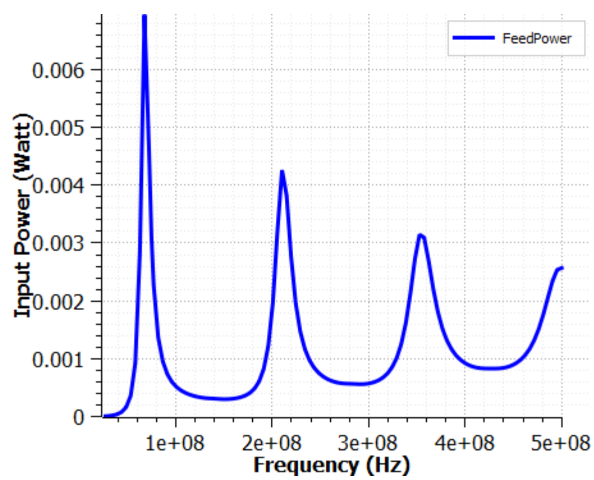
1. In the S parameter, dip can be seen close to the resonant frequency (in this case, 75MHz). For the proper operation, S11 should be less than -10dB.
2. In the phase angle of the S parameter, we can see 0-degree phase resonance.





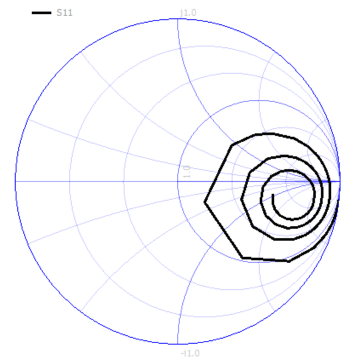
Plotting VSWR and Input Power: Next, we will plot the SWR (Standing Wave Ratio), the value of the SWR will be less than 2 in the operation range. The plot can be viewed by zooming in on the graph in the software.





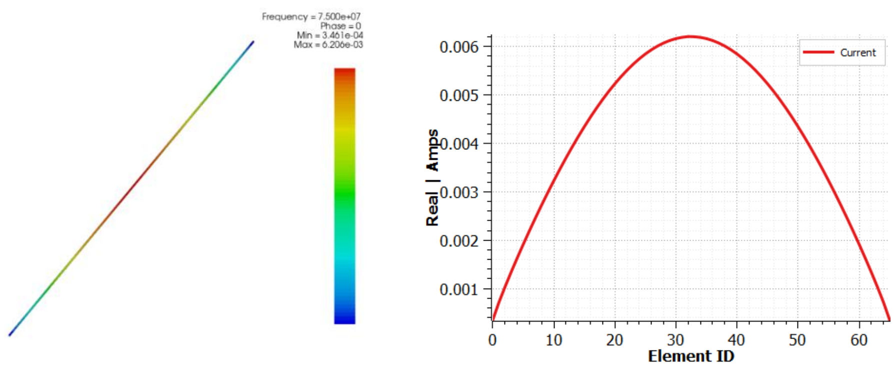
The next plot shows the fed power to the antenna. The fed power will be maximum at the resonant frequencies.

Plotting Smith Chart: TaraNG also has the Smith chart feature. We can see that the lower side depict the capacitive type of behavior and the upper side depict the inductive behavior.

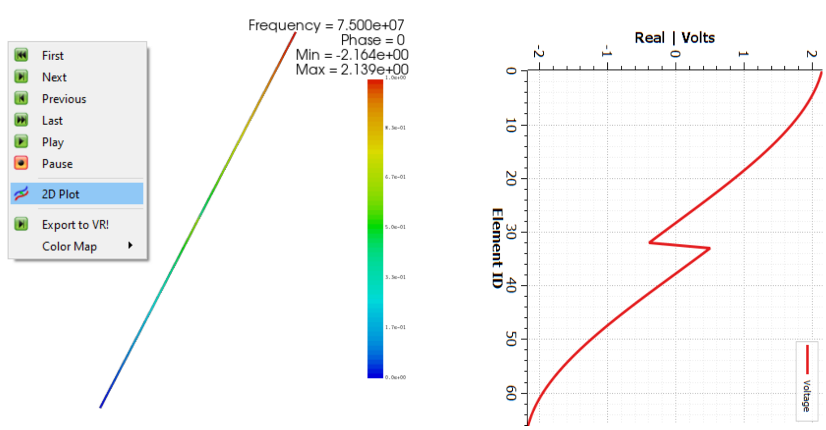


B) Near and Far Fields:

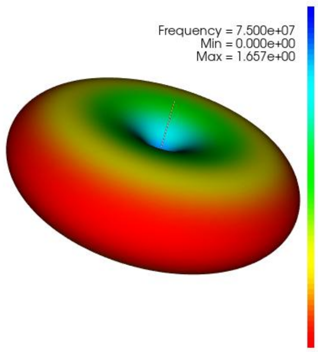
Visualizing Current Distribution: The current distribution can also be view as the 3D and as the 2D plot. At the center, the current will be maximum and at the ends, the current will be zero. It looks more like the half sinusoidal.

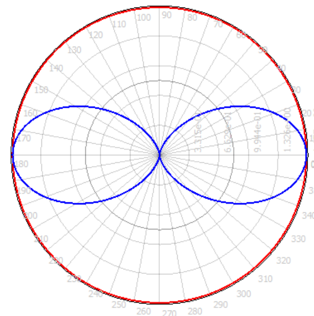


Visualizing Voltage Distribution: The first plot in the near-far field is the voltage distribution antenna where one side has a positive voltage and on the other end the negative voltage. The 2D plot of the same can also be viewed.

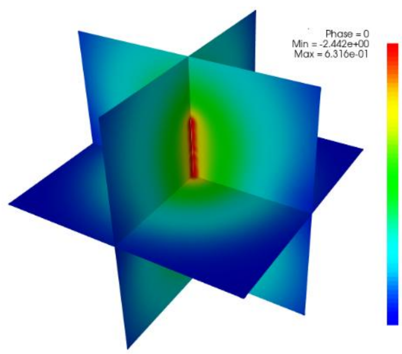


Visualizing Radiation Patterns: The 3D radiation pattern can be viewed in the TaraNG software by clicking on button with  icon. Dipole antenna's radiation pattern looks like a Doughnut shape with a gain of around 1.6. The polar plot can be viewed which looks like the shape of '8'.

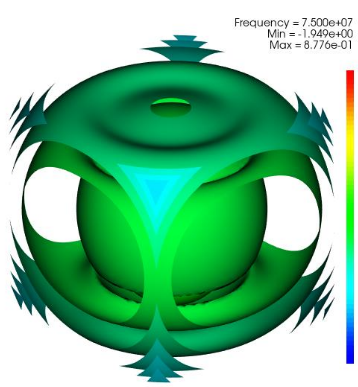




Visualizing Near-fields over Cut-planes: Next plot is of the field cut plane pattern of the electric field. Unique feature of the software is that the animation can be played simply by right clicking and playing the same.



Visualizing Near-fields Contours: 3D near field contour plane is the next icon on the left bar in the software. By clicking on the run button, the animation can be played for visualization. With this graph, one can view, how the waves propagate out from the antenna. (if the plot gets too zoomed/shrinked, 'R' button on the keyboard can be pressed to reset the view)



2. Patch Antenna

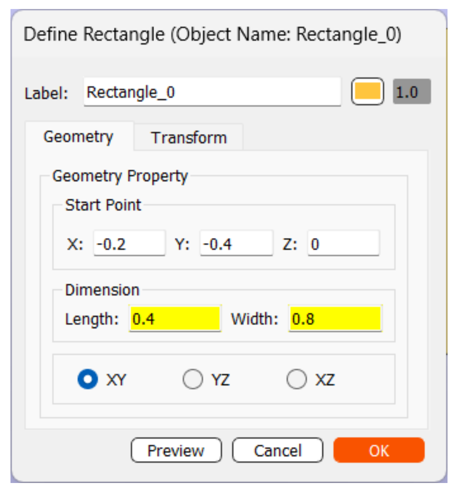
This example demonstrates how to simulate Patch Antenna with a probe feed. The height of the patch is 0.05 m. The feeding point is moved from the patch boundary to inner point at [0.1,0.166] used as excitation in the analysis is a voltage source. The probe radius is 0.004 m. The analysis is done for a linear frequency interval of 101 frequencies from 25 MHz to 500MHz.

Geometry Design

To draw this design, we need to draw two rectangular sheets for upper patch and the ground plane, one box for dielectric and a line for applying the feed point on a wire.

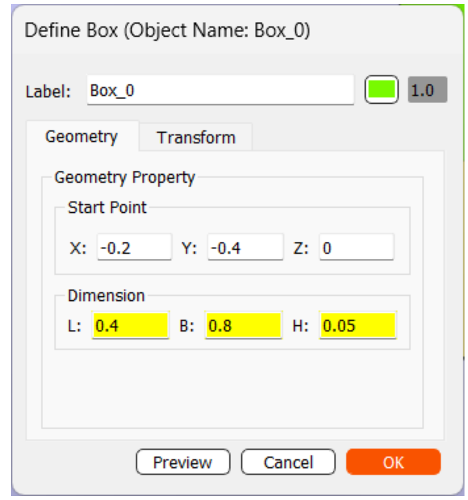
Step I:
Drawing the Design in CAD Modeller

1. Draw the Rectangle for Ground Plane



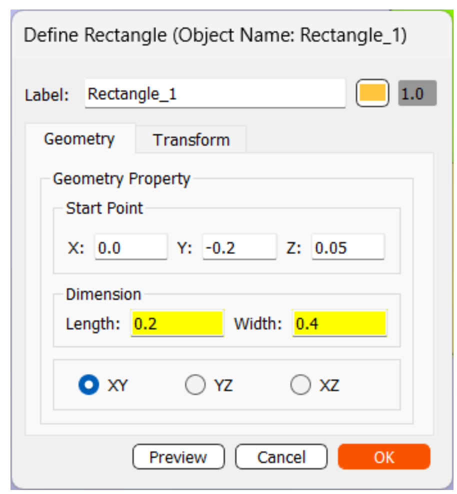
- To draw a rectangle, click on Rectangle tool  under *Shapes* toolbar.
- Click on the two points (*Start Point* and *End Point*) on the design grid where we want to draw the rectangle
- Doble click on the drawn rectangle to assign the exact dimensions
- Enter the *Start Point* of rectangle [**-0.2, -0.4, 0**] to define the position of rectangle in *Start Point* box under *Geometry Property* as shown in the figure.
- Enter the Length as **0.4** and width as **0.8** in *Length* and *Width* box under *Dimension* of *Geometry Property* as shown in figure, this defines dimensions of the rectangle.
- Set the Plane of drawing as **XY**, which is by-default
- Click on Preview and then *OK*

2. **Draw the Box for Dielectric**



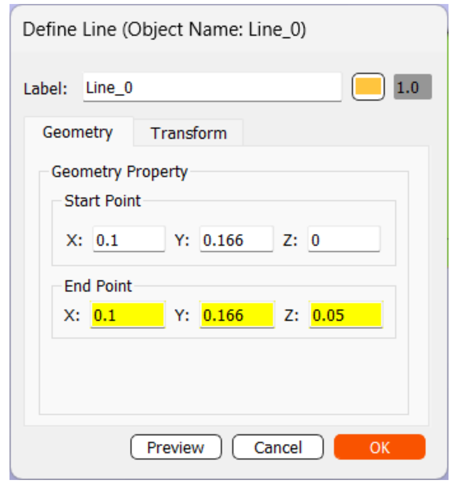
- To draw a box, click on Box tool under *Shapes* toolbar.
- Click on the three points (*Start Point* and Two points for Dimension *L, B, H*) on the design grid where we want to draw the box
- Doble click on the drawn box to assign the exact dimensions
- Enter the *Start Point* of box [**-0.2, -0.4, 0**] to define the position of box in *Start Point* under *Geometry Property* as shown in the figure.
- Enter the Length (L) as **0.4** and *Breadth (B)* as **0.8** and Height (H) as **0.05** in *L, B* and *H* box under *Dimension* of *Geometry Property* as shown in figure, this defines dimensions of the box.
- Click on Preview and then *OK*

3. **Draw the Rectangle for Patch**

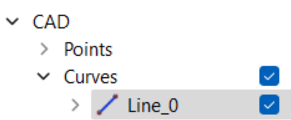


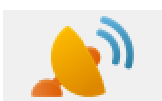
- To draw a rectangle, click on Rectangle tool under *Shapes* toolbar.
- Click on the two points (*Start Point* and *End Point*) on the design grid where we want to draw the rectangle
- Doble click on the drawn rectangle to assign the exact dimensions
- Enter the *Start Point* of rectangle [**0.0, -0.2, 0.05**] to define the position of rectangle in *Start Point* box under *Geometry Property* as shown in the figure.
- Enter the Length as **0.2** and width as **0.4** in *Length* and *Width* box under *Dimension* of *Geometry Property* as shown in figure, this defines dimensions of the rectangle.
- Set the Plane of drawing as **XY**, which is by-default
- Click on Preview and then *OK*

4. **Draw the Line for the probe feed**



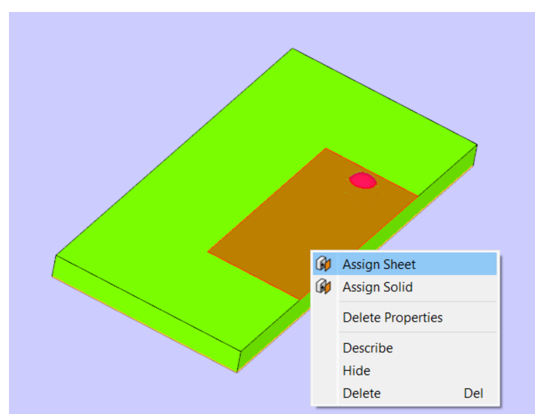
- To draw a line, click on Line tool in the *Curve Toolbar*
- Click on any two points (*Start Point* and *End Point*) on the design grid where we want to draw the line. Press Esc key on the keyboard.
- The clicks may not be precise, in order to enter the dimensions precisely doble click on the line drawn in the design window.
- Enter the values of *Start Point* as [**0.1,0.166,0**] and *End Point* [**0.1,0.166,0.05**] in the window as shown
- Click on the *OK* button and the window will be closed
- The line will be updated with desired coordinate points
- After successfully adding the Line into design it will be reflected under *Design Tree under CAD> Curves*



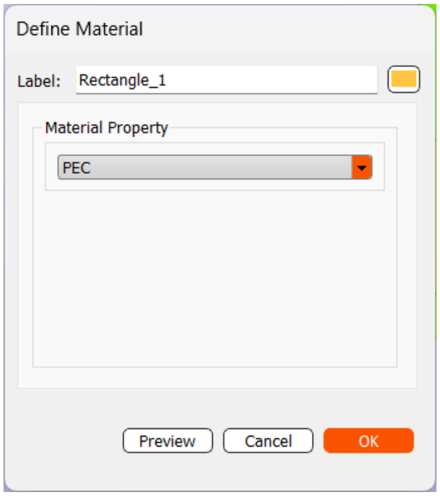


Step II:
Assigning Electromagnetic Properties in CEM

Assigning Perfect Electric Conductor (PEC) to Patch



- Select the patch surface **Rectangle_1**
- Right click on it and choose *Assign Sheet* from the context menu
- Choose *PEC* from the drop-down menu in the *Material Property group box*
- *Click on OK*

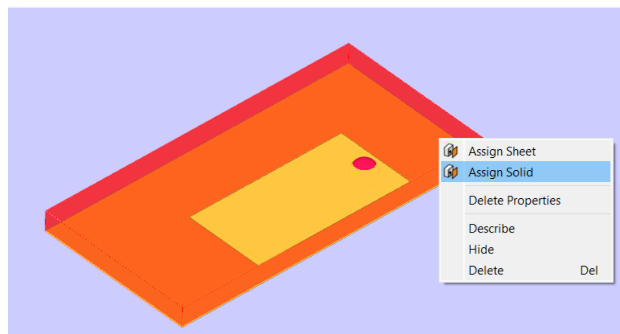


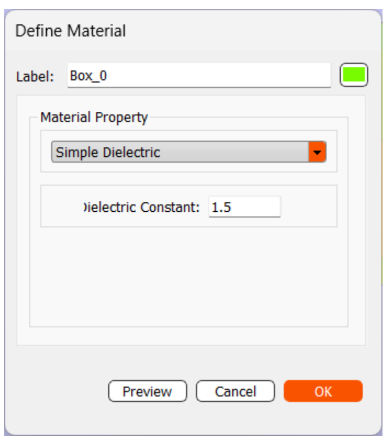
Assigning Perfect Electric Conductor (PEC) to Ground

- Select the ground surface **Rectangle_0**
- Right click on it and choose *Assign Sheet* from the context menu
- Choose *PEC* from the drop-down menu in the *Material Property group box*
- *Click on OK*

Assigning Dielectric Properties (PEC) to Box_0

- Select the dielectric slab volume **Box_0**
- Right click on it and choose *Assign Solid* from the context menu

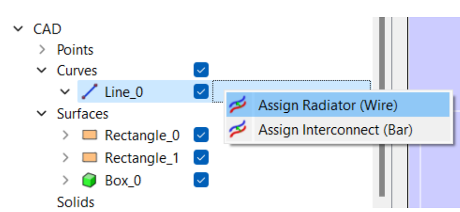


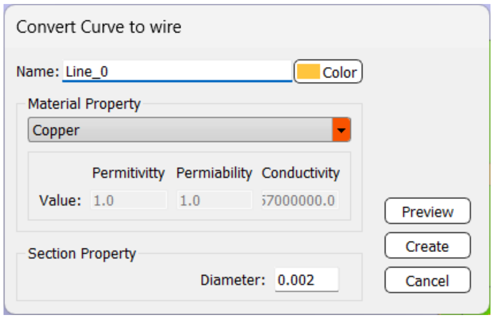


- Choose *Simple Dielectric* from the drop-down menu in the *Material Property group box*
- Enter the value of *Dielectric Constant as* **1.5**
- *Click on OK*

Assigning Wire to Line Element

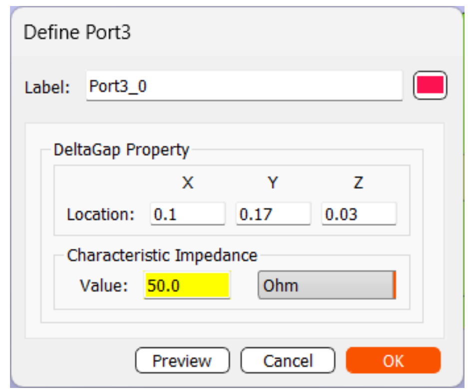
- To assign the wire properties right click on the **Line_0** from *Design Tree* as the line may not be visible that is inside the dielectric slab and select *Assign Radiator (Wire)* from the context menu.
- The Wire window will appear





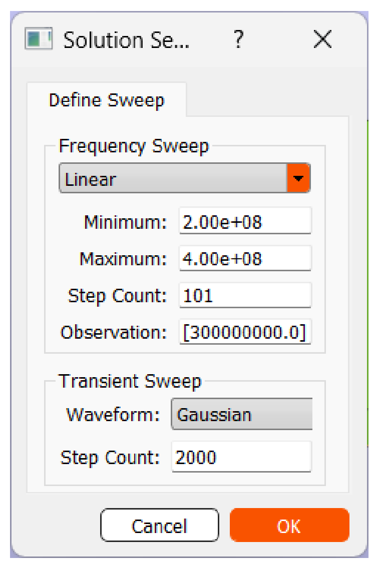
- Select the material from the drop-down menu. By default, the Copper material will be assigned
- Enter the diameter of the wire and click on the *Create* button

Assigning the Port Excitation

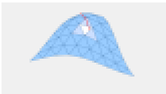


- To assign the feed, click on the 'Port' icon in the taskbar
- Drag the cursor near the wire center to get the visuals.
- By clicking anywhere on the wire, the feed will be placed.
- To make sure the port is exactly at the given point, double click on the port
- Set the port location X, Y, Z as **0.1, 0.17, 0.03** respectively
- The default characteristic impedance is 50 ohms
- Click OK to complete the step.

Assigning Frequency Sweep



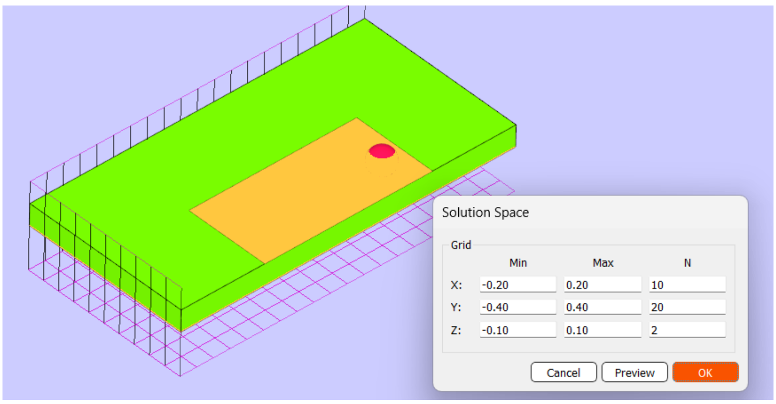
- Click on icon in the Electromagnetics Toolbar
- Choose the frequency sweep as Linear or Logarithmic
- Enter Minimum and Maximum frequency as **200e8** and **400e8**
- The Step Count is set to **101** points
- Define the observation frequency 300e8
  for obtaining near and far field, radiation patterns

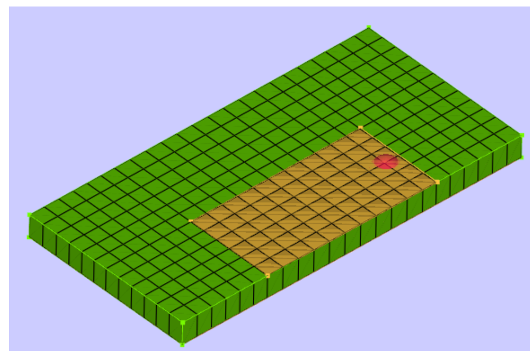


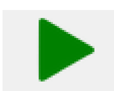
Step III: Voxelization

TaraNG automatically sets the rules for meshes and suggests the lower and upper boundary and the number of cells in each direction.

- To perform meshing click on  tool from the meshing toolbar
- The Solution Space box will appear which will include pre-computed values
- User can change these values (optional) and click on OK button

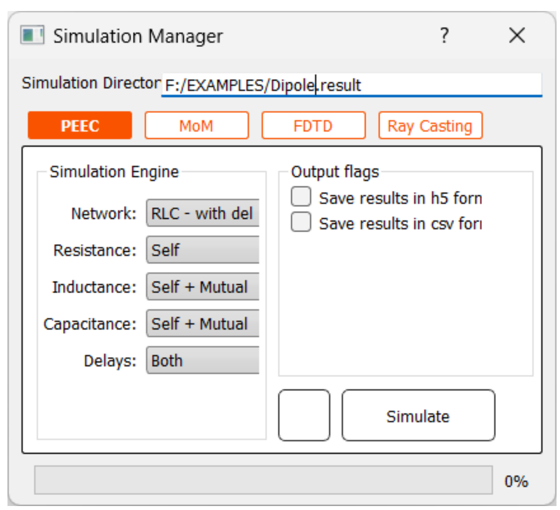


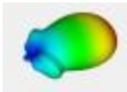




Step IV: Running the Simulation

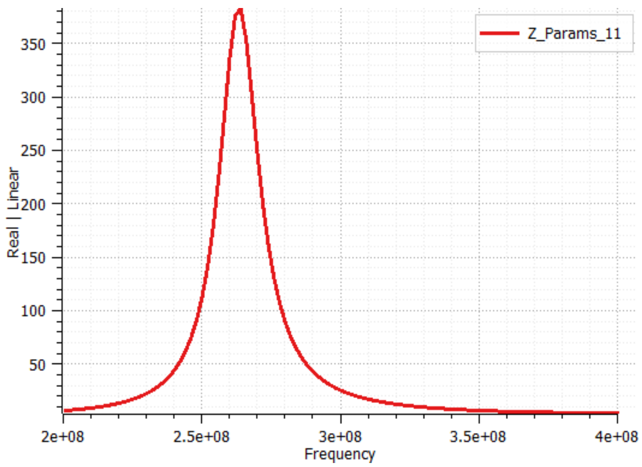
- To run the simulation, click on  the Run button
- Setup Analysis settings are useful for setting up solver method and inversion methods. We will keep these parameters as default.
- Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.

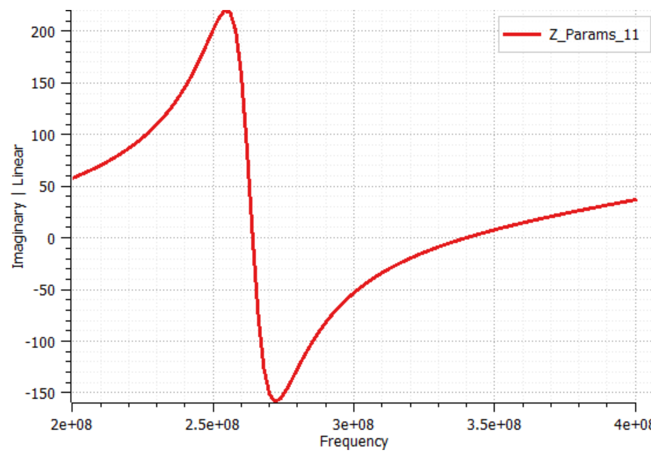




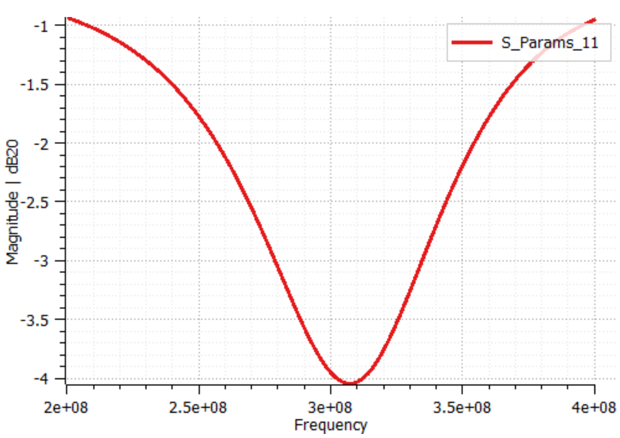
Step V:
Visualize the Results

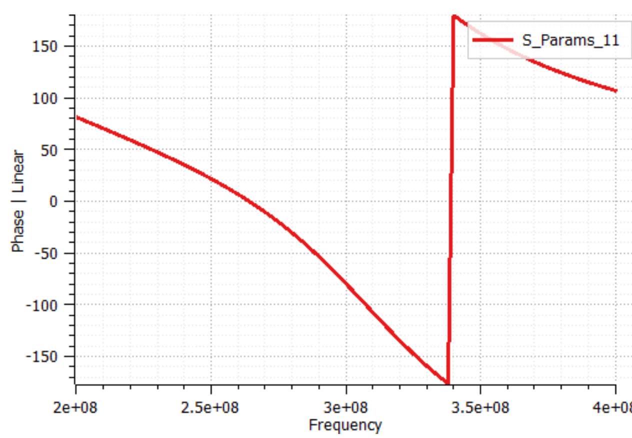
Plotting Input Impedance: Clicking on the *Result Toolbar* shows the real part of the input impedance  The very next tab shows the result of the imaginary part of the input impedance 





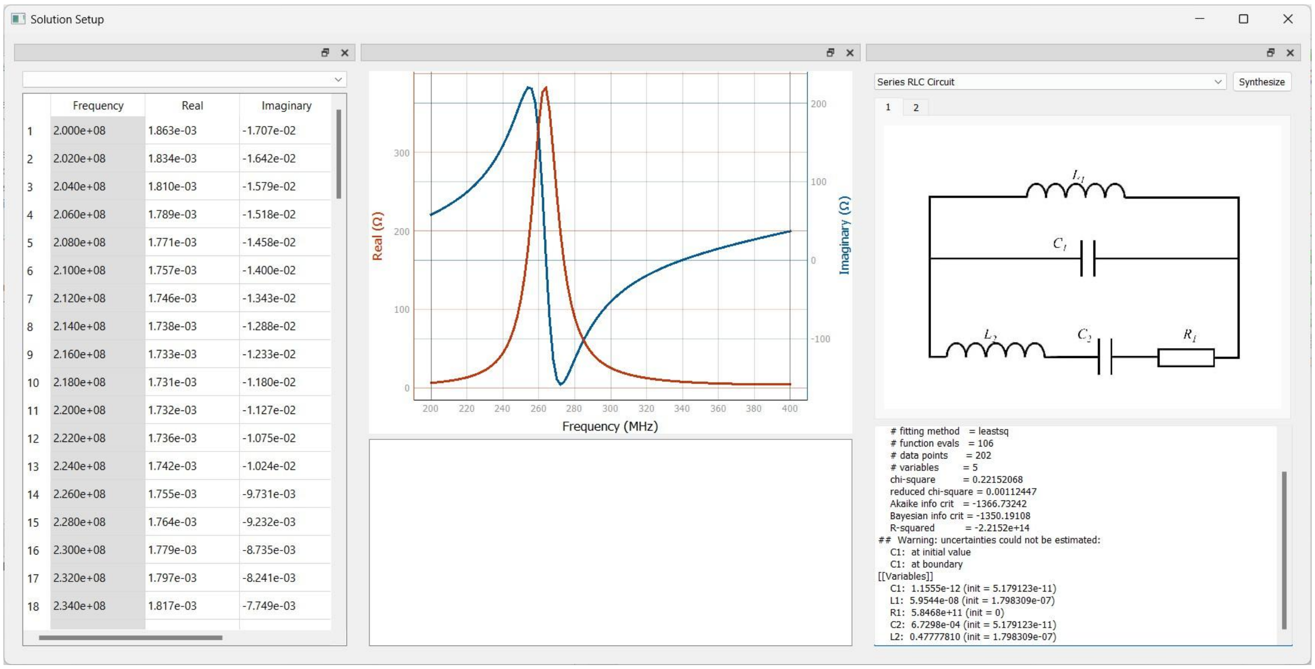
Plotting Scattering Parameters: Clicking on the *Result Toolbar* shows the magnitude s-parameters and the phase of s-parameter plot





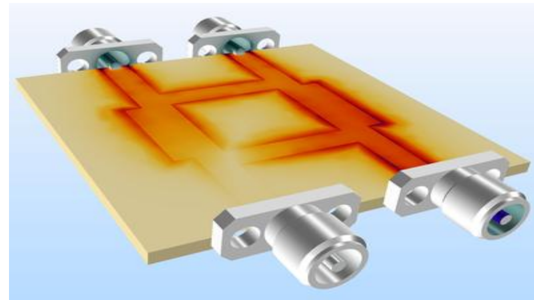
Step VI:
Visualize the DPEC Circuit

The Derived Physically Expressible Circuit between the desired nodes can be visualized and understood from the DPEC window as shown below. This window is further explained in the appendix.



3. Branchline Coupler

The Branch-line coupler, also known as the quadrature hybrid coupler, is a foundational device in the field of microwave and RF Engineering.



It is widely used for signal splitting, combining, and phase manipulation. Its unique ability to divide power with specific amplitude and phase relationships makes it indispensable in numerous applications, including radar systems, satellite communication, and signal processing circuits.

The branch-line coupler has four ports: input, through, coupled and isolated. It is often built using transmission lines like microstrip or stripline technology.

The conventional design has two pairs of quarter-wavelength (λ/4) transmission lines in a rectangular shape. These lines are carefully designed to produce the necessary coupling, impedance matching and phase characteristics.

The quarter-wavelength transmission lines are critical for the coupler's functionality. At the design frequency, the electrical length of these lines guarantees that the input power is evenly distributed between the through and coupled ports, with a 90-degree phase difference.

This phase shift is critical for applications that need accurate signal handling.

Operating Principal

The branchline coupler's operation is based on the principles of transmission line theory. When a signal is applied to the input port, it propagates through the coupler and splits between the through and coupled ports. The degree of power division is determined by the transmission lines' impedance and coupling coefficient.

In an ideal coupler:

- The input power is distributed evenly between the through and coupled ports, with each receiving 50% of the power.
- The isolated port stays signal-free, allowing efficient energy transmission to the required ports.
- The signals at the through and coupled ports have a 90-degree phase difference, identifying the branchline coupler as a quadrature device.

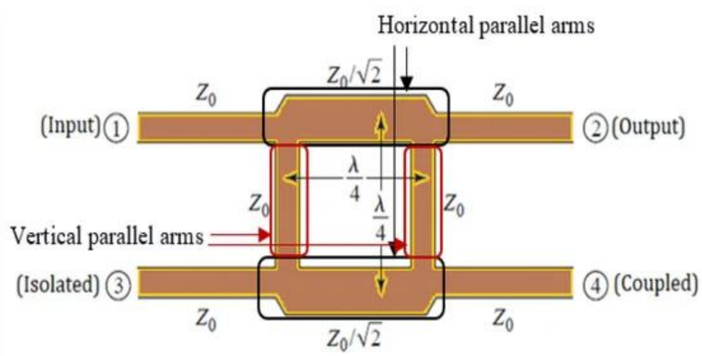
Key Parameters

Several parameters define the performance of a branchline coupler:

- **Coupling Coefficient:** Determines the fraction of input power transferred to the coupled port.
- **Isolation:** Indicates how effectively the isolated port is decoupled from the input signal, with higher isolation being desirable.
- **Return Loss:** Reflects the quality of impedance matching, with lower return loss indicating better performance.
- **Insertion Loss:** Represents the power lost within the coupler itself, which ideally should be minimal.
- **Bandwidth:** Defines the range of frequencies over which the coupler operates efficiently.

Design Variants

Several variants of the branchline coupler have been developed to address specific requirements.

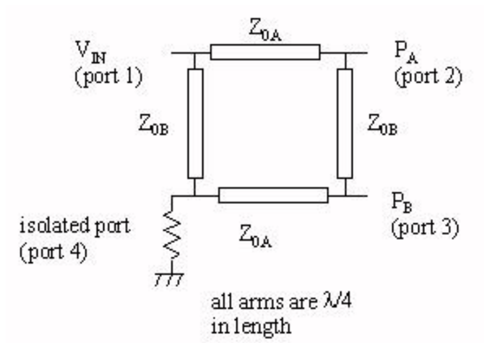


The standard 3-dB branchline coupler provides equal power splitting, but other designs allow for unequal power division. Advanced configurations may also extend the bandwidth or improve isolation, expanding the range of potential applications. Shown below is an **equal branchline coupler.**

Unequal Branchline Coupler

A branch line coupler is 3-dB directional coupler with a 90-degree phase difference in the outputs of the through and coupled arms. This can be made unequal power BLC by selecting the branch line impedances to provide various power division ratios. This type of unequal branch line coupler hybrid is often made in microstrip line or stripline form as shown below:

With all ports matched, power entering port 1 is evenly divided between ports 2 and 3, with a 90-degree phase shift between these outputs. No power is coupled to port 4 (the isolated port).



Formulae

$$Z_1 = Z_0 \times \left(\frac{\frac{P_A}{P_B}}{1 + \frac{P_A}{P_B}} \right)^{0.5}$$

$$Z_2 = Z_0 \times \left(\frac{P_A}{P_B} \right)^{0.5}$$

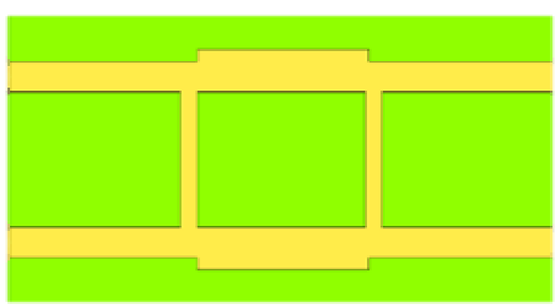
For this experiment the power ratio is assumed to be 0.5 and the value of characteristic impedance Z0 is 50 Ohms.

Calculations

Using the above formula, the length and width of the Z1 and Z2 microstrip patches can be deduced as follows:

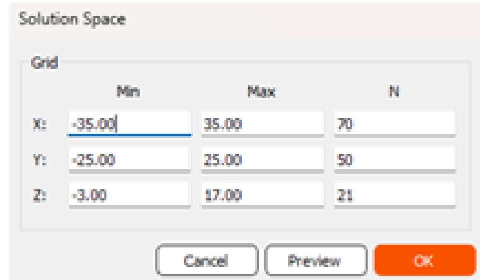
Microstrip patch	Width (mm)	Length (mm)
Z1	4.20209	13.3985
Z2	1.58002	14.0464

Design In TaraNG

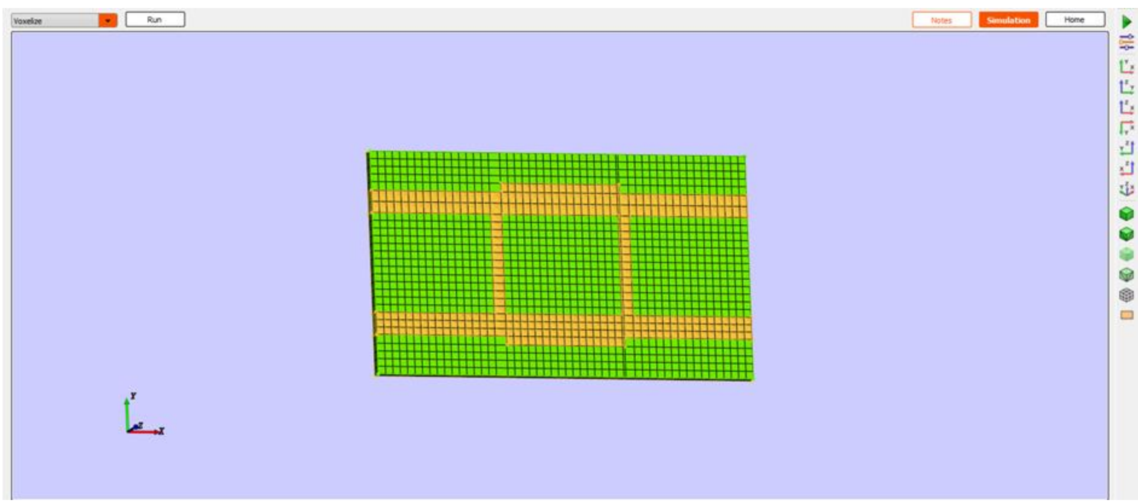


Meshing

A 70mm*50mm*20mm mesh was used to discretize the structure into small mesh cells. This is essential to determine how the electromagnetic wave interact with each mesh cell and to study the response by applying Maxwell’s equations.

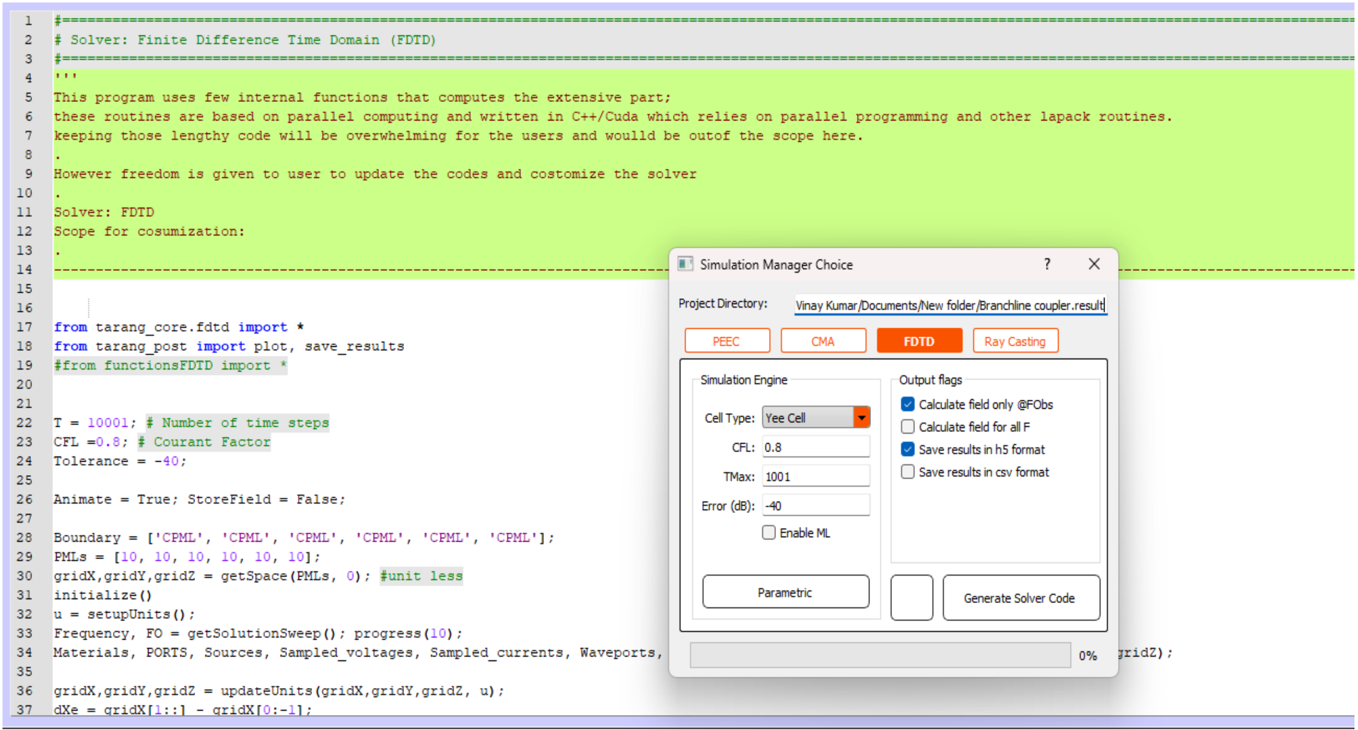


The accuracy of the simulation heavily depends on the quality and refinement of the mesh. A finer mesh generally captures more details of the geometry and field variations, leading to more precise results. Fields such as electromagnetic fields, currents, or stress distributions are solved on the mesh, so poor meshing can result in significant errors. From the images, it can be seen that each and every component has been meshed properly.



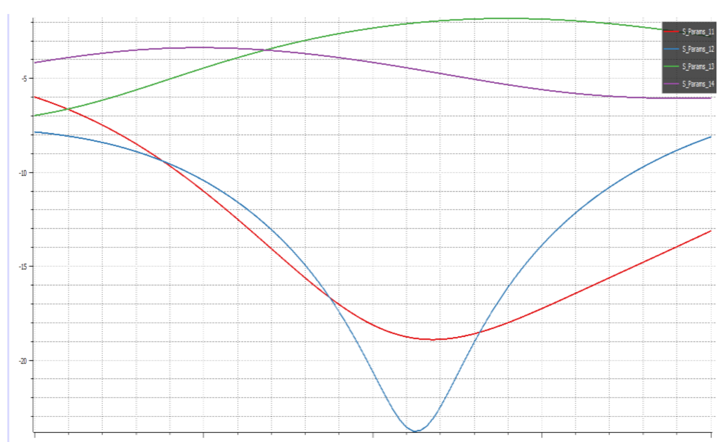
Solver Method

The solver method used for this problem was Finite Difference Time Domain (FDTD). The Finite-Difference Time-Domain (FDTD) method is a widely used computational technique for solving Maxwell's equations and simulating electromagnetic wave propagation in various materials and environments. FDTD directly solves Maxwell's curl equations in the time domain, making it suitable for transient and broadband analysis.



The computational domain is divided into a uniform or non-uniform grid (e.g., Yee grid), where field components are calculated at discrete time and space intervals. FDTD uses explicit time-stepping schemes, meaning future field values are computed directly from current and past values. FDTD can handle large computational domains and a wide range of frequencies.

Simulation Result in TaraNG



Applications

The branchline coupler’s versatility makes it suitable for a wide range of applications:

- **Power Division and Combining:** It is commonly used to split a signal into two paths with specific phase and amplitude characteristics or to combine multiple signals into one.
- **Phase Shifting:** The 90-degree phase difference between output ports is crucial for applications such as phased array antennas and quadrature modulators.
- **Signal Processing:** It plays a pivotal role in mixers, modulators and demodulators, especially in systems requiring precise quadrature signals.
- **Radar and Communication Systems:** The branchline coupler is used in the design of hybrid circuits, duplexers and beamforming networks.

Advantages and Limitations

The branchline coupler has various advantages, including simplicity, compact size, and accurate amplitude and phase balancing. However, it does have limits. Its performance is heavily reliant on design precision, and it often has a restricted bandwidth compared to other coupler types.
Furthermore, fabrication tolerances can considerably impact its performance, therefore precision manufacture is crucial.

Conclusion

The branchline coupler is a fundamental component in microwave engineering, enabling crucial activities such as power division, phase shifting and signal processing.
While it has certain restrictions, its simple design and consistent performance keep it a popular choice for many high-frequency applications. As technology improves, further developments in materials and fabrication processes are anticipated to improve the branchline coupler's capabilities, assuring its continued use in future microwave systems.

4. Simulation and Analysis of an Asymmetric U-shaped Monopole Antenna Embedded with a T-shaped Strip

The Federal Communication Commission (FCC) approved and allocated the frequency range from 3.1-10.6GHz for ultra-wideband (UWB) communication systems. In the recent years, these UWB systems have emerged as the most prominent technologies for the wireless industries. UWB designs have the advantages such as low profile, low cost for consumer electronics, compact size, high data transmission rate, easy integration with microwave circuits, and easy design.

This example demonstrates how to design the following an Asymmetric U-shaped Monopole Antenna embedded with a T-shaped Strip for Ultrawideband characteristics on TaraNG, to simulate and analyse its various parameters such as Reflection coefficient, Gain, Radiation pattern and so on. The antenna implements a defected ground plane for wideband characteristics.

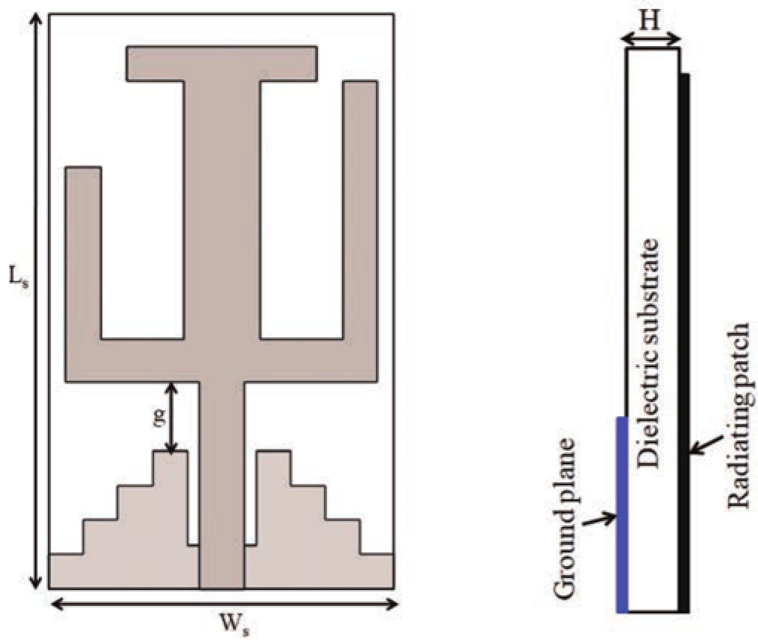


Fig. Asymmetric U-shaped Monopole Antenna with a defected ground plane

The overall dimensions of this antenna including the ground plane are 34 (L_S)x 20 (W_S) x 1.6 (H) mm^3, and the antenna is printed on a low-cost FR4 material and fed by a 50-ohm microstrip line.

Geometry Design

Using the appropriate dimensions for all the elements in the following structure, it can be designed using a combination of code and CAD interface. This structure can be effortlessly designed using polygons and Boolean operations.

The Asymmetric U-shaped Monopole Antenna structure:

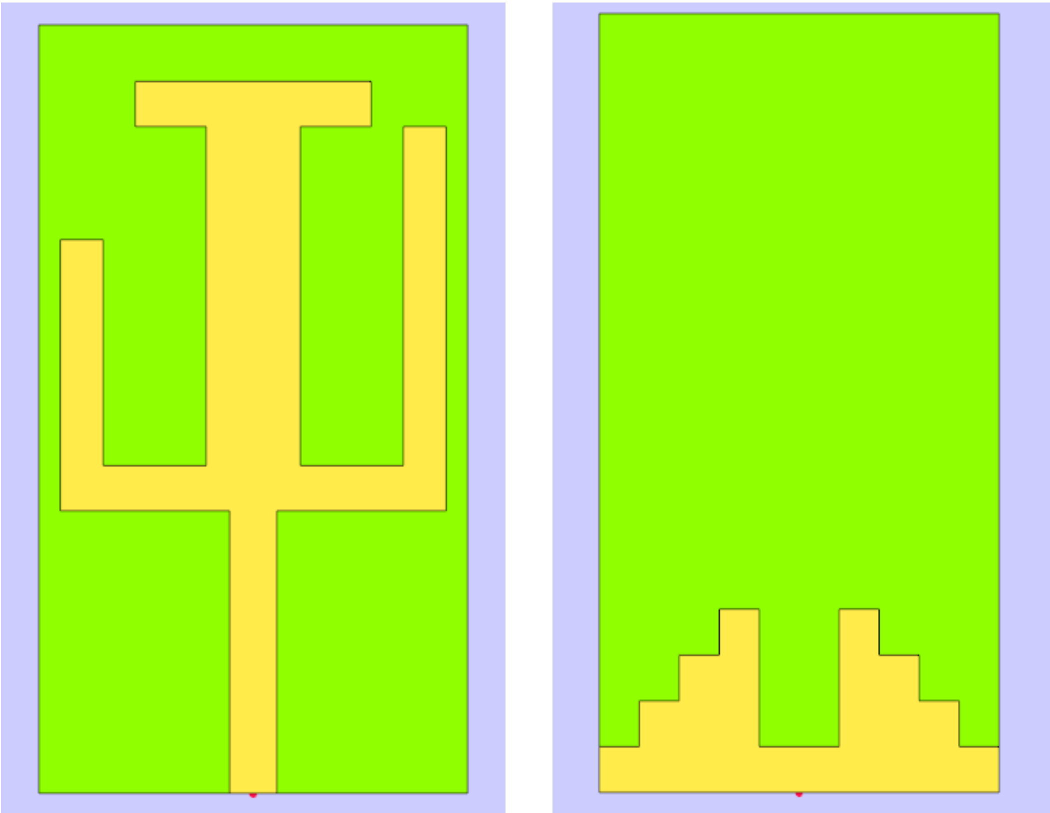


Fig. Asymmetric U-shaped Monopole Antenna Design front and rear view respectively

Using the Code to design this structure makes the process much quicker and more efficient. Shown below is the code that achieves the same purpose with the added advantage of easy modifications with the use of variables. Polygons are easy to code on TaraNG and having a knowledge of all the vertices of the structure's outline makes the process effortless. Employing if-else statements to design two different polygons under the same function makes the code much lighter and less redundant. One can also employ the use of loops and nested loops to make the code more efficient.

Code (Mode: Enable AUTO)

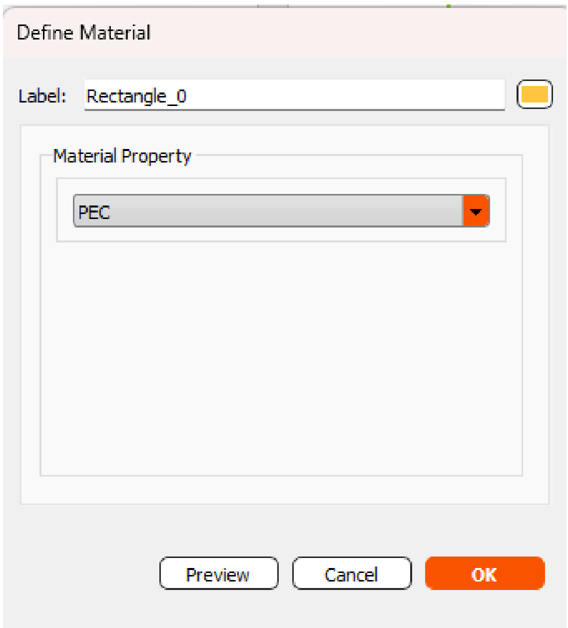
```
lg=8; wg=20; g=4.5;
l=2; w=2; wh=4.8; whl=3.3; lh=5;
dl=1.5; tc=0.035;
config=0; t=3000;
def myPoly(config=0):
        points = [];
        ws=20; ls=34; lp=19; wp=18;
        l1=12.5; wl=2.2; l2=10;
        w2=2; l3=15; w3=4.4;
        l4=2; w4=11; l5=15; d=2;
        ts=1.6; ln=6; wn=4;
        lg=8; wg=20; g=4.5;
        l=2; w=2; wh=4.8; whl=3.3; lh=5;
        dl=1.5; tc=0.035;
        if config==0:
                points.append([wl/2, -ls/2,0]); points.append([wl/2, -ls/2+l1,0]);
                points.append([wp/2, -ls/2+l1,0]); points.append([wp/2, -ls/2+l1+d+l5,0]);
                points.append([wp/2-w2, -ls/2+l1+d+l5,0]); points.append([wp/2-w2, -ls/2+l1
                points.append([wp/2-w2-wh, -ls/2+l1+d,0]); points.append([wp/2-w2-wh, -ls/
                points.append([wp/2-w2-wh+whl, -ls/2+l1+d+l3,0]); points.append([wp/2-w2-w
                points.append([wp/2-w2-wh+whl-w4, -ls/2+l1+d+l3+l4,0]); points.append([wp/
                points.append([wp/2-w2-wh+whl-w4+whl, -ls/2+l1+d+l3,0]); points.append([wp
                points.append([wp/2-w2-wh+whl-w4+whl-wh, -ls/2+l1+d,0]); points.append([-w
                points.append([-wp/2, -ls/2+l1+d+l2,0]); points.append([-wp/2, -ls/2+l1,0])
                points.append([-wl/2, -ls/2+l1,0]); points.append([-wl/2, -ls/2,0]);
                return points;
        else:
                points.append([ws/2, -ls/2,-ts]); points.append([ws/2, -ls/2+l,-ts]);
                points.append([ws/2-w, -ls/2+l,-ts]); points.append([ws/2-w, -ls/2+2*l,-ts]
                points.append([ws/2-2*w, -ls/2+2*l,-ts]); points.append([ws/2-2*w, -ls/2+3
                points.append([ws/2-3*w, -ls/2+3*l,-ts]); points.append([ws/2-3*w, -ls/2+4
                points.append([ws/2-4*w, -ls/2+4*l,-ts]); points.append([ws/2-4*w, -ls/2+4
                points.append([ws/2-4*w-wn, -ls/2+4*l-ln,-ts]); points.append([ws/2-4*w-wn,
                points.append([ws/2-5*w-wn, -ls/2+4*l,-ts]); points.append([ws/2-5*w-wn, -
                points.append([ws/2-6*w-wn, -ls/2+3*l,-ts]); points.append([ws/2-6*w-wn, -
                points.append([ws/2-7*w-wn, -ls/2+2*l,-ts]); points.append([ws/2-7*w-wn, -
                points.append([ws/2-8*w-wn, -ls/2+l,-ts]); points.append([ws/2-8*w-wn, -ls,
                return points;
Polygon_0 = Polygon(Points=myPoly(config=0))
Polygon_0.AddTo(Project,'Polygon_0')
Polygon_1 = Polygon(Points=myPoly(config=1))
Polygon_1.AddTo(Project,'Polygon_1')
Box_0 = Box(StartPoint = [-ws/2, -ls/2, 0], L = ws, B = ls, H = -ts)
Box_0.AddTo(Project,'Box_0')
```

Code for the Asymmetric U-shaped Monopole Antenna

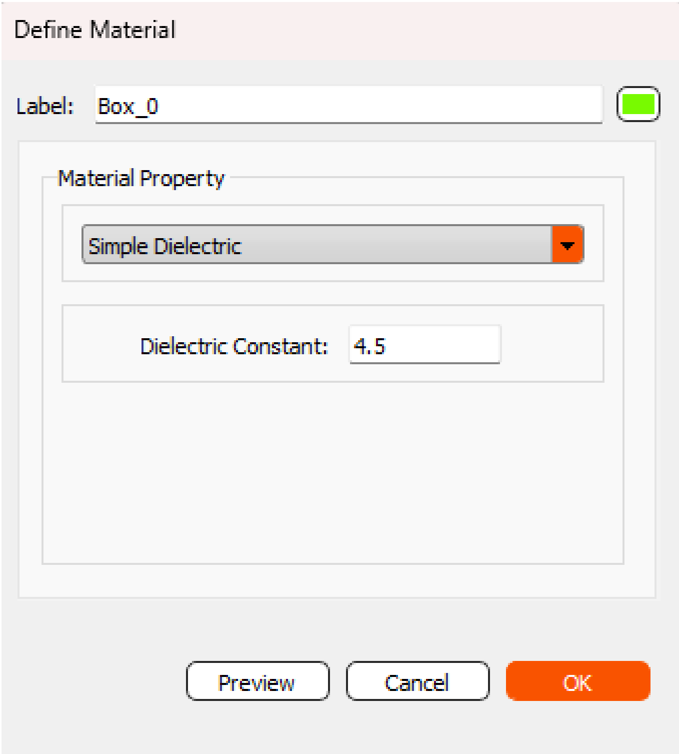
Setting up the Excitation and assigning material properties

To assign material properties and port excitation click on CEM operation in the TaraNG Taskbar.

1. Assign PEC material to the patch and the ground



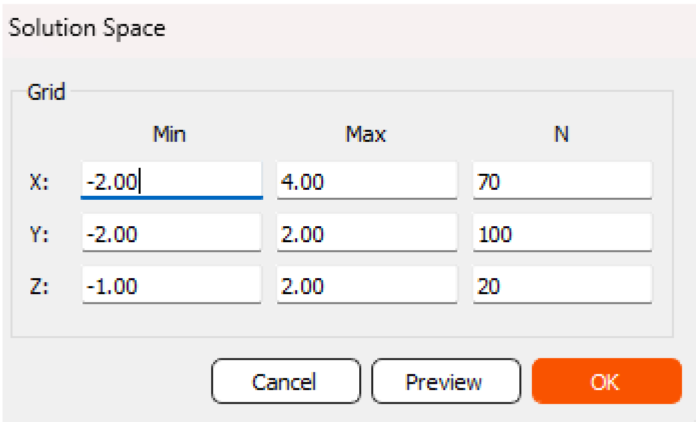
2. To add the feed, click on the 'Port'  icon and trace the arrow joining any two edges of the rectangle in the XZ plane alongside the feedline.
3. Assign Simple Dielectric property to the Substrate box with the necessary Effective permittivity value. In this instance it is 4.5



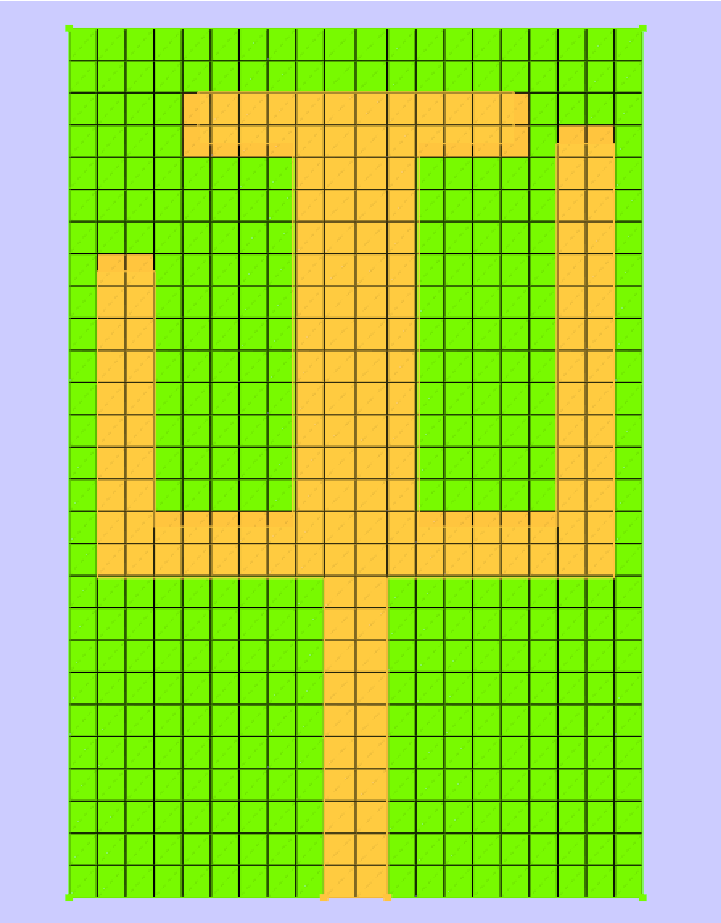
Meshing and Voxelization

To assign material properties and port excitation click on Mesh operation in the TaraNG Taskbar.

1. Click on the  icon in the tool bar and provide the appropriate dimensions for the meshing box along with the necessary number of divisions for the mesh cells

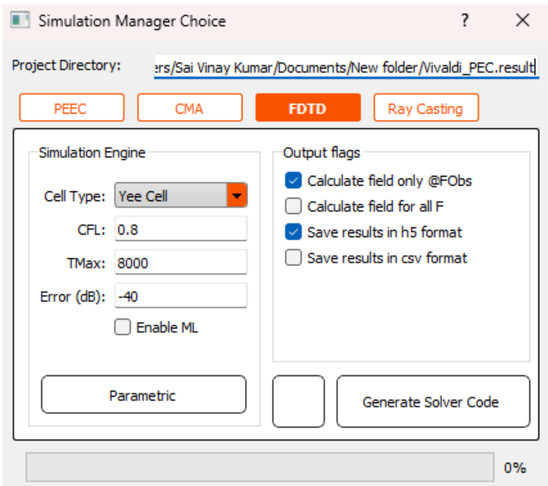


2. After selecting the grid of appropriate dimensions and divisions click on Run to get the following meshed structure.



Meshed View

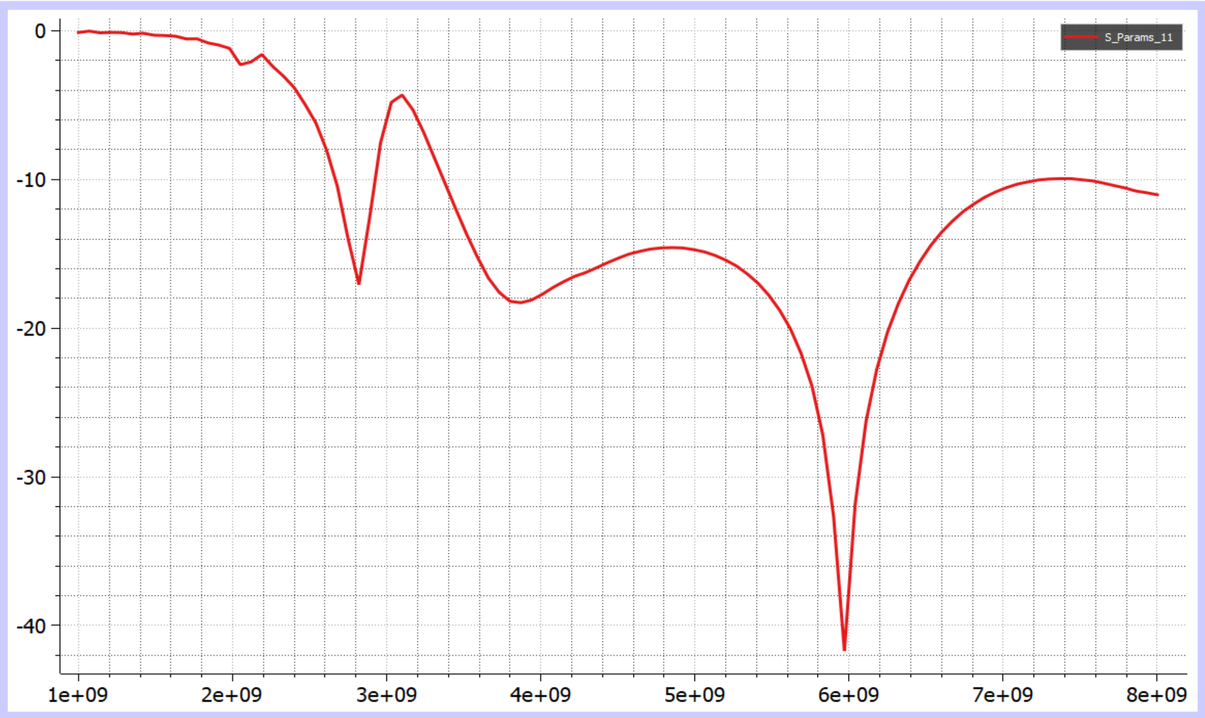
Running the Simulation



1. To run the simulation, click on the Run button  in the taskbar.
2. Select FDTD setup solver and assign the necessary time steps (Tmax). The other parameters can be kept as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.

Obtaining the Scattering parameters

1. To view the results of the simulation, click on the results operation in the taskbar.
2. Select the plots that require analysis and view them through the toolbar in the results tab.
3. Here are the S11 parameters obtained after simulation through the FDTD solver over 15000-time steps.



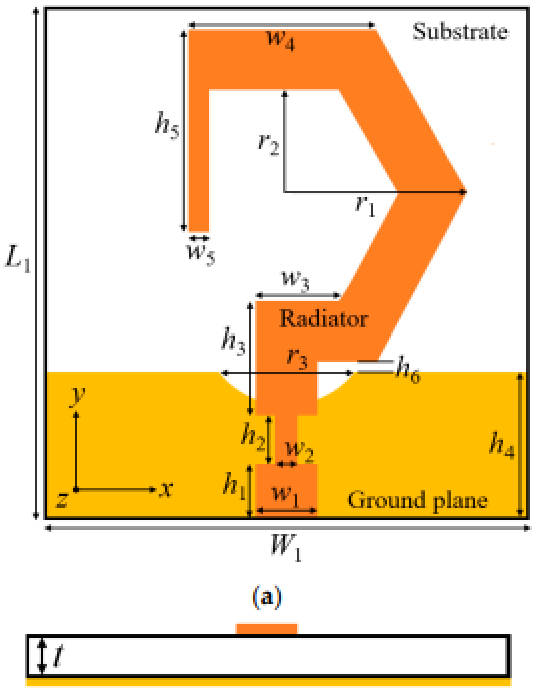
S11 results for Asymmetric U-shaped Monopole Antenna

Result Analysis

From the obtained results, it can be observed that the Asymmetric U-shaped Monopole Antenna has an operating bandwidth of 3.25Ghz to 8Ghz and it has a wide bandwidth. Hence the obtained results are consistent with the dimensions of the design and the Asymmetric U-shaped Monopole Antenna can be used for ultrawideband applications.

5. Simulation and Analysis of a circularly polarized (CP) monopole antenna for MIMO applications

This example demonstrates how to design the following circularly polarized (CP) monopole antenna on TaraNG, to simulate and analyse its various parameters such as Reflection coefficient, Gain, Radiation pattern and so on. The circularly polarized monopole antenna is to be placed with three other such antennas for MIMO application that require enhanced spatial diversity.



Circularly polarized monopole antenna

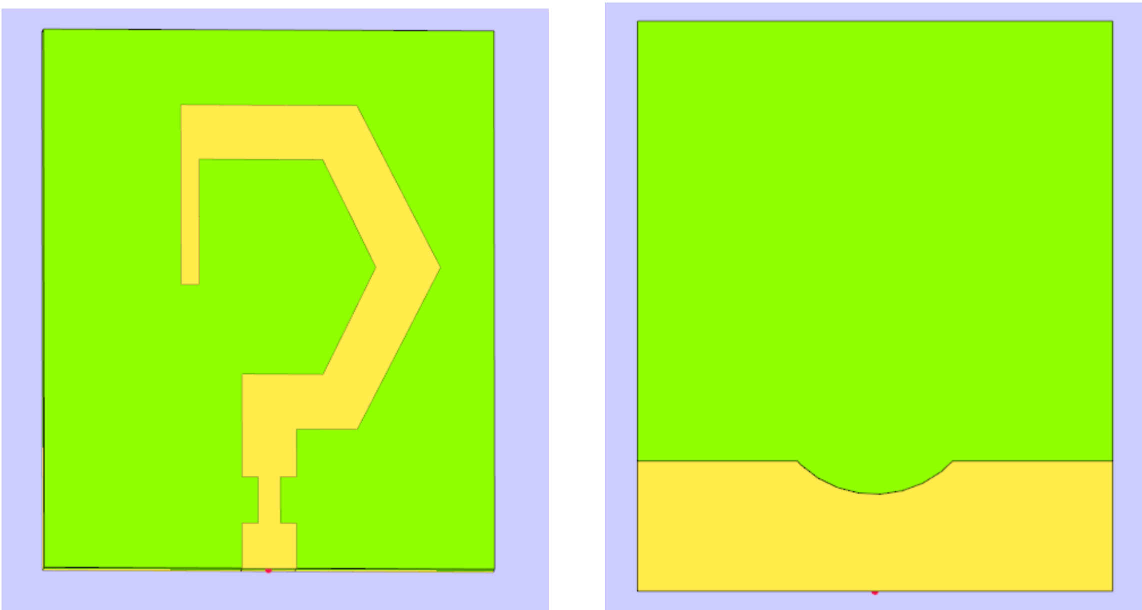
The following design is based on the below cited reference paper:

[Kumar, Sachin & Lee, Gwan & Kim, Donghwi & Choi, Hyunchul & Kim, Kang. (2020). Dual Circularly Polarized Planar Four-Port MIMO Antenna with Wide Axial-Ratio Bandwidth. Sensors. 20. 5610. 10.3390/s20195610.]

Geometry Design

Using the dimensions of all the elements mentioned in the Cited reference paper the following structure can be designed using a combination of code and CAD interface. This structure can be effortlessly designed using polygons and Boolean operations.

The circularly polarized monopole antenna structure:



Circularly polarized monopole antenna Design front and rear view respectively

Using the Code to design this structure makes the process much quicker and more efficient. Shown below is the code that achieves the same purpose with the added advantage of easy modifications with the use of variables. Polygons are easy to code on TaraNG and having a knowledge of all the vertices of the structure's outline makes the process effortless.

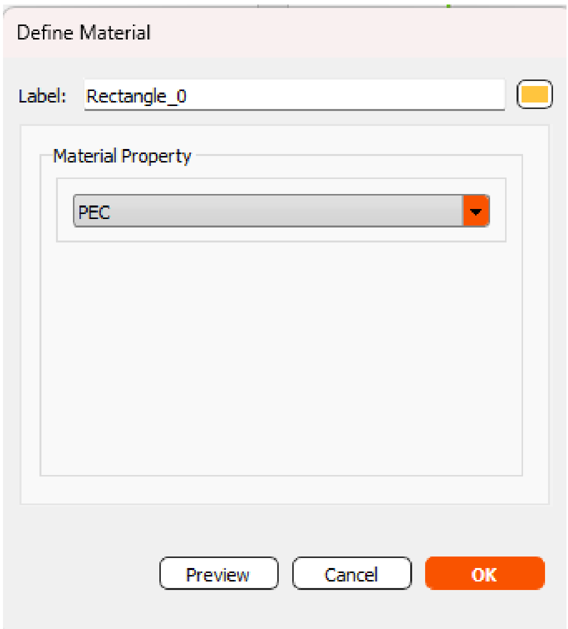
```
Project = initProject()
l1=25; swl=25; l2=70; sw2=68;
h1=2.5; h2=2.6; h3=5.7; h4=6.85;
h5=10; h6=0.92; h7=5.65; h8=15; h9=5;
r1=9.5; r2=6; r3=4.5; ts=1.6;
w1=3; w2=1.2; w3=4.5; w4=9.75; w5=1; w6=0.5;
w8=1; w9=4; w10=w11=2.5; w12=19.5;
offset=5; t=10000;
def myPoly():
        points = [];
        l1=25; swl=25; l2=70; sw2=68;
        h1=2.5; h2=2.6; h3=5.7; h4=6.85;
        h5=10; h6=0.92; h7=5.65; h8=15; h9=5;
        r1=9.5; r2=6; r3=4.5; ts=1.6;
        w1=3; w2=1.2; w3=4.5; w4=9.75; w5=1; w6=0.5;
        w8=1; w9=4; w10=w11=2.5; w12=19.5;
        points.append([w1/2, -l1/2,0]); points.append([w1/2, -l1/2+h1,0]);
        points.append([w2/2, -l1/2+h1,0]); points.append([w2/2, -l1/2+h1+h2,0]);
        points.append([w1/2, -l1/2+h1+h2,0]); points.append([w1/2, -l1/2+h4+h6,0])
        points.append([w4/2, -l1/2+h4+h6,0]); points.append([r1, -l1/2+h1+h2+h3+r2
        points.append([w4/2, -l1/2+h1+h2+h3+2*r2+w1,0]); points.append([-w4/2, -l1
        points.append([-w4/2, -l1/2+h1+h2+h3+2*r2+w1-h5,0]); points.append([-w4/2+
        points.append([r1-3.55, -l1/2+h1+h2+h3+r2,0]); points.append([-w1/2+w3, -l
        points.append([-w1/2, -l1/2+h1+h2+h3,0]); points.append([-w1/2, -l1/2+h1+h
        points.append([-w2/2, -l1/2+h1+h2,0]); points.append([-w2/2, -l1/2+h1,0]);
        points.append([-w1/2, -l1/2+h1,0]); points.append([-w1/2, -l1/2,0]);
        return points;
Polygon_0 = Polygon(Points=myPoly())
Polygon_0.AddTo(Project,'Polygon_0')
Box_0 = Box(StartPoint = [-swl/2, -l1/2, 0], L = swl, B = l1+offset, H = -ts)
Box_0.AddTo(Project,'Box_0')
Rectangle_0 = Rectangle(StartPoint=[-swl/2, -l1/2, -ts], A=swl, B=h4)
Rectangle_0.AddTo(Project,'Rectangle_0')
Circle_0 = Circle(Center = [0, -l1/2+h1+h2+h3, -ts], Radius = h3)
Circle_0.AddTo(Project,'Circle_0')
Difference_0 = Difference(Rectangle_0, Circle_0)
Difference_0.AddTo(Project)
```

Code for the Circularly polarized monopole antenna

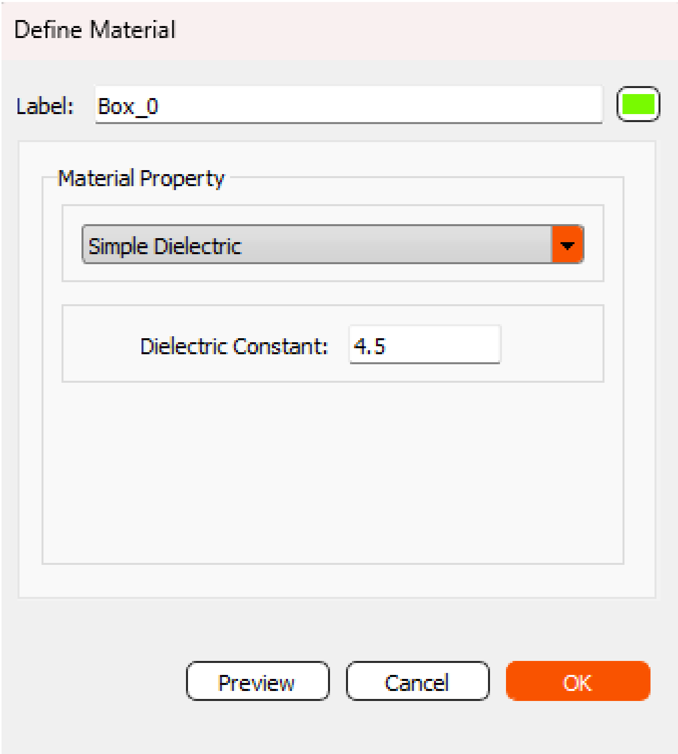
Setting up the Excitation and assigning material properties

To assign material properties and port excitation click on CEM operation in the TaraNG Taskbar.

1. Assign PEC material to the patch and the ground



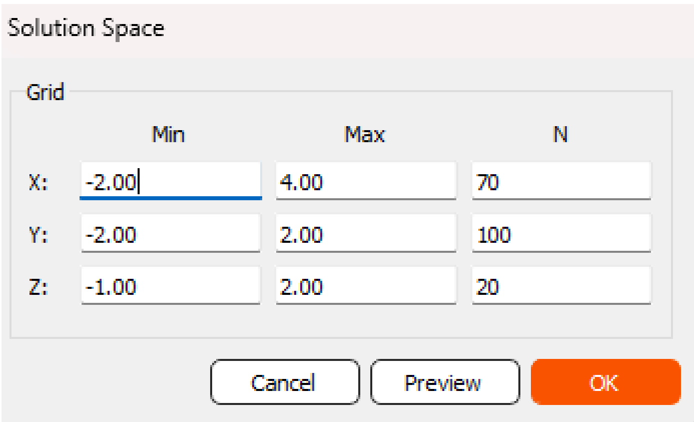
2. To add the feed, click on the 'Port'  icon and trace the arrow joining any two edges of the rectangle in the XZ plane alongside the feedline.
3. Assign Simple Dielectric property to the Substrate box with the necessary Effective permittivity value. In this instance it is 4.5



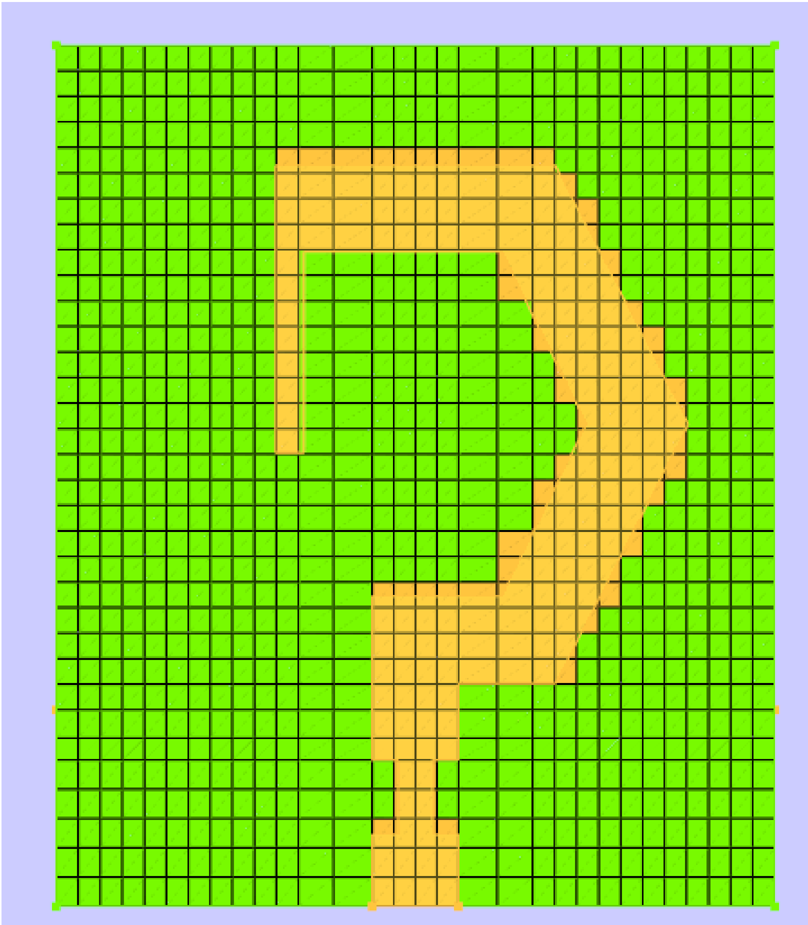
Meshing and Voxelization

To assign material properties and port excitation click on Mesh operation in the TaraNG Taskbar.

1. Click on the  icon in the tool bar and provide the appropriate dimensions for the meshing box along with the necessary number of divisions for the mesh cells

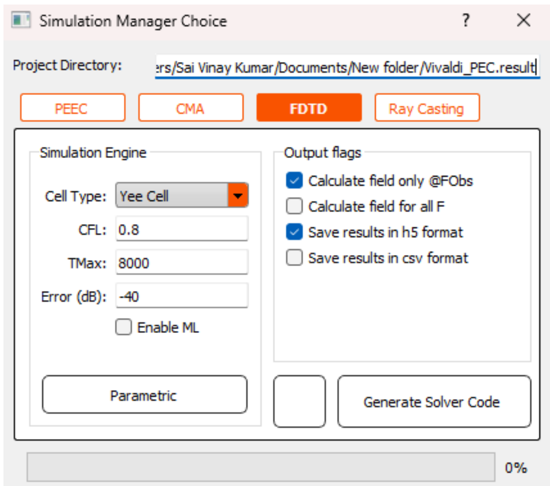


2. After selecting the grid of appropriate dimensions and divisions click on Run to get the following meshed structure.



Meshed View

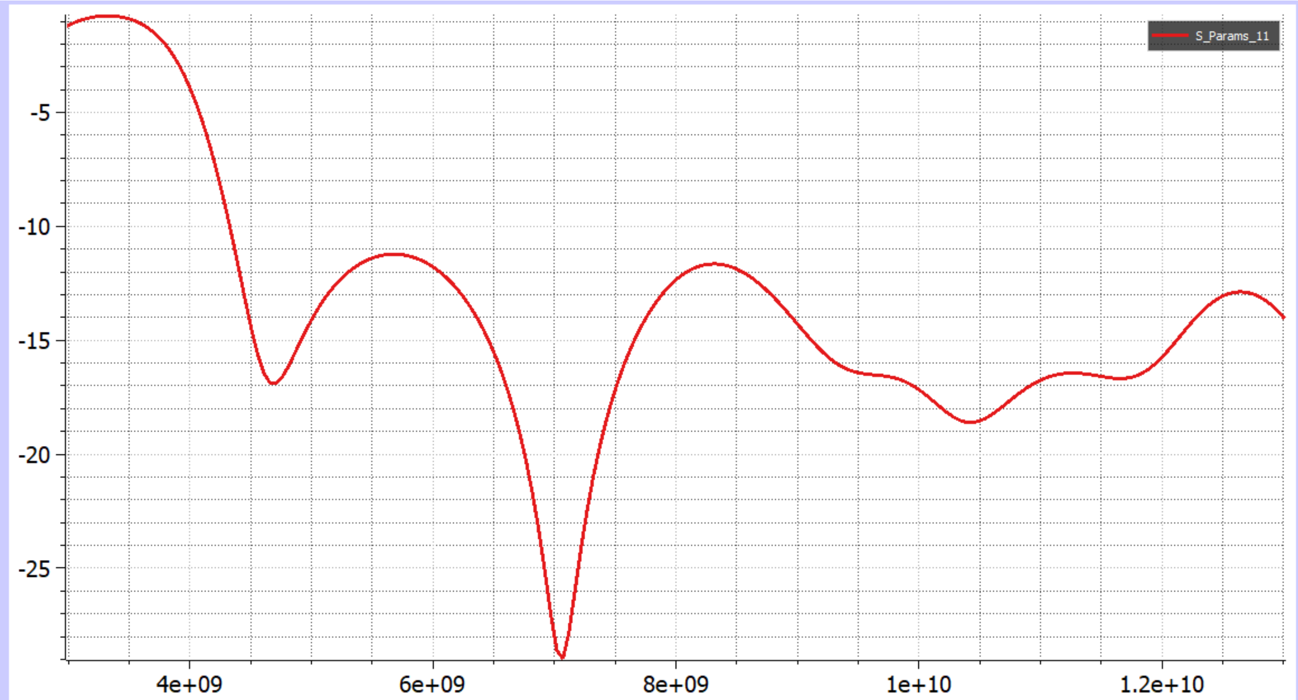
Running the Simulation



1. To run the simulation, click on the Run button  in the taskbar.
2. Select FDTD setup solver and assign the necessary time steps (Tmax). The other parameters can be kept as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.

Obtaining the Scattering parameters

1. To view the results of the simulation, click on the results operation in the taskbar.
2. Select the plots that require analysis and view them through the toolbar in the results tab.
3. Here are the S11 parameters obtained after simulation through the FDTD solver over 15000-time steps.



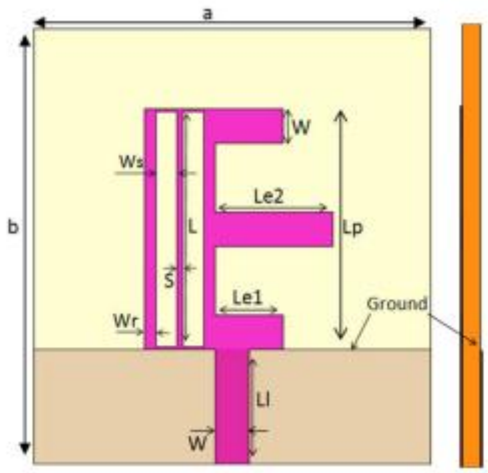
S11 results for Circularly polarized monopole antenna

Result Analysis

From the obtained results, it can be observed that the Circularly polarized monopole antenna has an operating bandwidth of 4.25GHz to 13GHz. This coincides with the simulated results in the above cited reference paper. Hence the obtained results are consistent with the dimensions of the design and the Circularly polarized monopole antenna works as intended.

6. Simulation and Analysis of a Novel Triband E-Shaped Printed Monopole Antenna

The E-shaped monopole antenna can create a single resonance within the WLAN range. Placement of two slots within the E-shaped monopole antenna creates two extra resonances whose centre frequencies can be adjusted by the E-shaped monopole and the slots parameters.



E-shaped monopole antenna

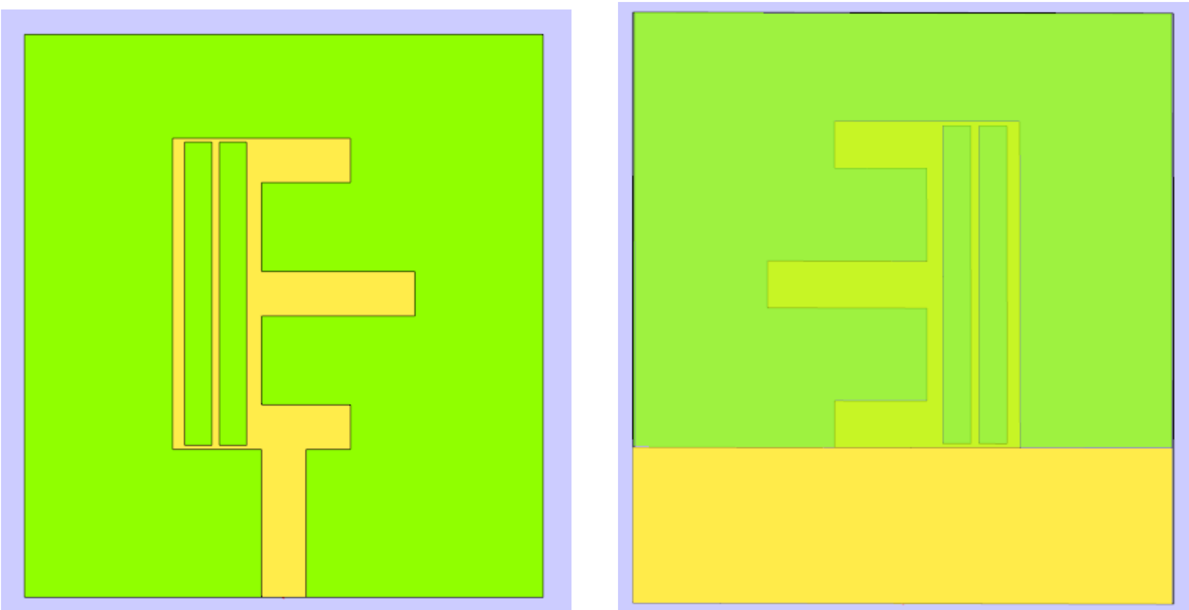
This example demonstrates how to design an E-shaped monopole antenna on TaraNG to simulate and analyse its various parameters such as Reflection coefficient, Gain, Radiation pattern and so on. an E-shaped microstrip antenna is a good candidate for MIMO applications. As such, the E-shaped patch is used in the present printed monopole antenna work. This design is based on the reference paper cited below

Mohammad-Ali-Nezhad, Sajad & Hassani, H.R.. (2010). A Novel Triband E-Shaped Printed Monopole Antenna for MIMO Application. Antennas and Wireless Propagation Letters, IEEE. 576 - 579. 10.1109/LAWP.2010.2051131.

Geometry Design

Using the dimensions of all the elements mentioned in the Cited reference paper the following structure can be designed using a combination of code and CAD interface. This structure can be effortlessly designed using polygons and Boolean operations.

The E-shaped monopole antenna structure:



E-shaped monopole antenna Design front and rear view respectively

Using the Code to design this structure makes the process much quicker and more efficient. Shown below is the code that achieves the same purpose with the added advantage of easy modifications with the use of variables. Polygons are easy to code on TaraNG and having a knowledge of all the vertices of the structure's outline makes the process effortless. Additionally, Boolean operations such as difference operation must be used to carve out the two rectangular slots. This can be seen in the code.

Code (Mode: Enable AUTO)

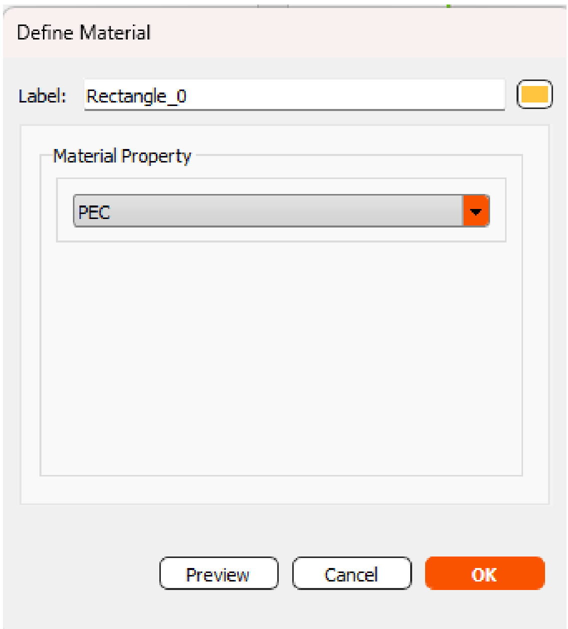
```
Project = initProject()
a=35; lp=21; l=20.47; l1=10;
le1=6; le2=10.35; b=38;
w=3; ws=1.85; wr=1;
s=0.5; ts=0.8; tc=0.035;
t=3000;
def myPoly():
        points = [];
        a=35; lp=21; l=20.47; l1=10;
        le1=6; le2=10.35; b=38;
        w=3; ws=1.85; wr=1;
        s=0.5; ts=0.8; tc=0.035;
        points.append([w/2, -b/2,0]); points.append([w/2, -b/2+l1, 0]);
        points.append([-w/2+le1, -b/2+l1, 0]); points.append([-w/2+le1, -b/2+l1+w, 0]);
        points.append([-w/2, -b/2+l1+w, 0]); points.append([-w/2, -b/2+l1+w+le1, 0]);
        points.append([-w/2+le2, -b/2+l1+w+le1, 0]); points.append([-w/2+le2, -b/2+l1+w+le1+w, 0]);
        points.append([-w/2, -b/2+l1+w+le1+w, 0]); points.append([-w/2, -b/2+l1+w+le1+w+le1, 0]);
        points.append([-w/2+le1, -b/2+l1+w+le1+w+le1, 0]); points.append([-w/2+le1, -b/2+l1+w+le1+w+le1+
        points.append([-w/2-le1, -b/2+l1+w+le1+w+le1+w, 0]); points.append([-w/2-le1, -b/2+l1+w+le1+w+le
        points.append([-w/2, -b/2+l1+w+le1+w+le1+w-lp, 0]); points.append([-w/2, -b/2+l1, 0]); points.ap
        return points;
Polygon_0 = Polygon(Points=myPoly())
Polygon_0.AddTo(Project,'Polygon_0')
Box_0 = Box(StartPoint = [-a/2, -b/2, 0], L = a, B = b, H = -ts)
Box_0.AddTo(Project,'Box_0')
Rectangle_0 = Rectangle(StartPoint=[-w/2-wr, -b/2+l1+(lp-l)/2, 0], A=-ws, B=l)
Rectangle_0.AddTo(Project,'Rectangle_0')
Rectangle_1 = Rectangle(StartPoint=[-w/2-wr-ws-s, -b/2+l1+(lp-l)/2, 0], A=-ws, B=l)
Rectangle_1.AddTo(Project,'Rectangle_1')
Rectangle_2 = Rectangle(StartPoint=[-a/2, -b/2, -ts], A=a, B=l1)
Rectangle_2.AddTo(Project,'Rectangle_2')
Rectangle_3 = Rectangle(StartPoint=[-1.0, -18.5, -0.02], A=2.42, B=-1.5)
Rectangle_3.SetPlane('XZ')
Rectangle_3.AddTo(Project,'Rectangle_3')
Rectangle_3.Modify(StartPoint=[-w/2,-b/2,-ts], A=w, B=ts)
Rectangle_3.Refresh();
Difference_0 = Difference(Polygon_0, Rectangle_0)
Difference_0.AddTo(Project)
Difference_1 = Difference( Difference_0, Rectangle_1)
Difference_1.AddTo(Project)
```

Code for the E-shaped monopole antenna Design

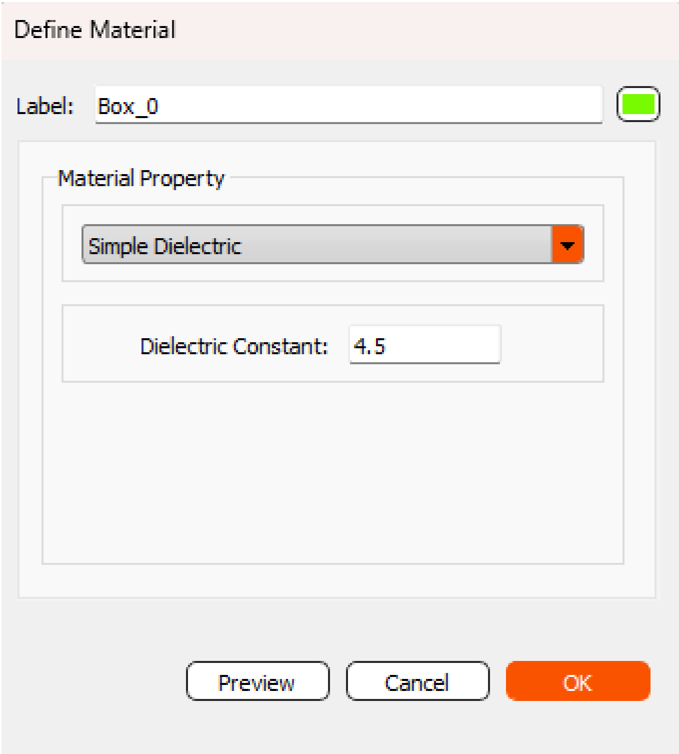
Setting up the Excitation and assigning material properties

To assign material properties and port excitation click on CEM operation in the TaraNG Taskbar.

1. Assign PEC material to the patch and the ground



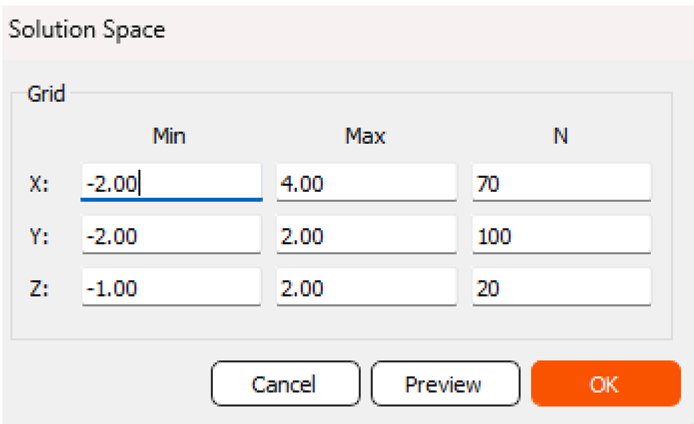
2. To add the feed, click on the ‘Port’  icon and trace the arrow joining any two edges of the rectangle in the XZ plane alongside the feedline.
3. Assign Simple Dielectric property to the Substrate box with the necessary Effective permittivity value. In this instance it is 4.5



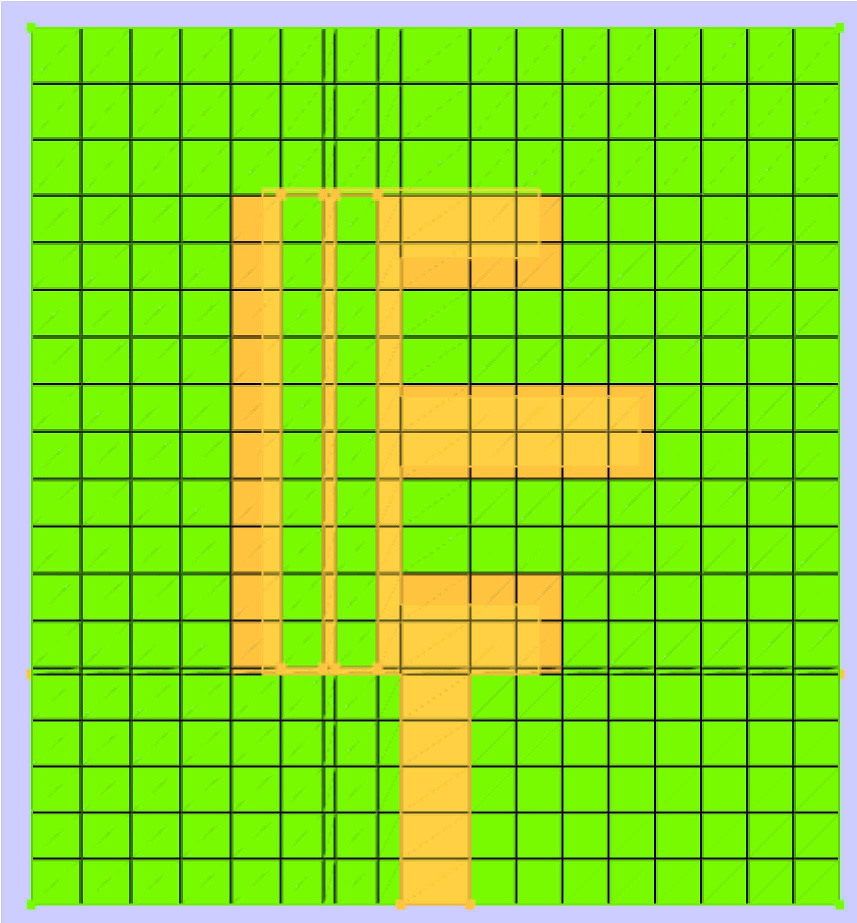
Meshing and Voxelization

To assign material properties and port excitation click on Mesh operation in the TaraNG Taskbar.

1. Click on the  icon in the tool bar and provide the appropriate dimensions for the meshing box along with the necessary number of divisions for the mesh cells



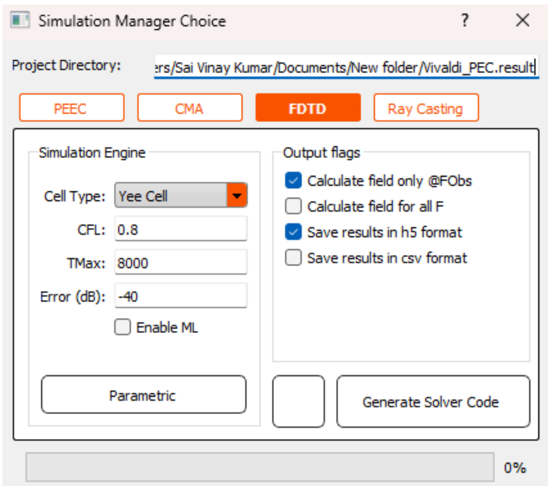
2. After selecting the grid of appropriate dimensions and divisions click on Run to get the following meshed structure.



Meshed View

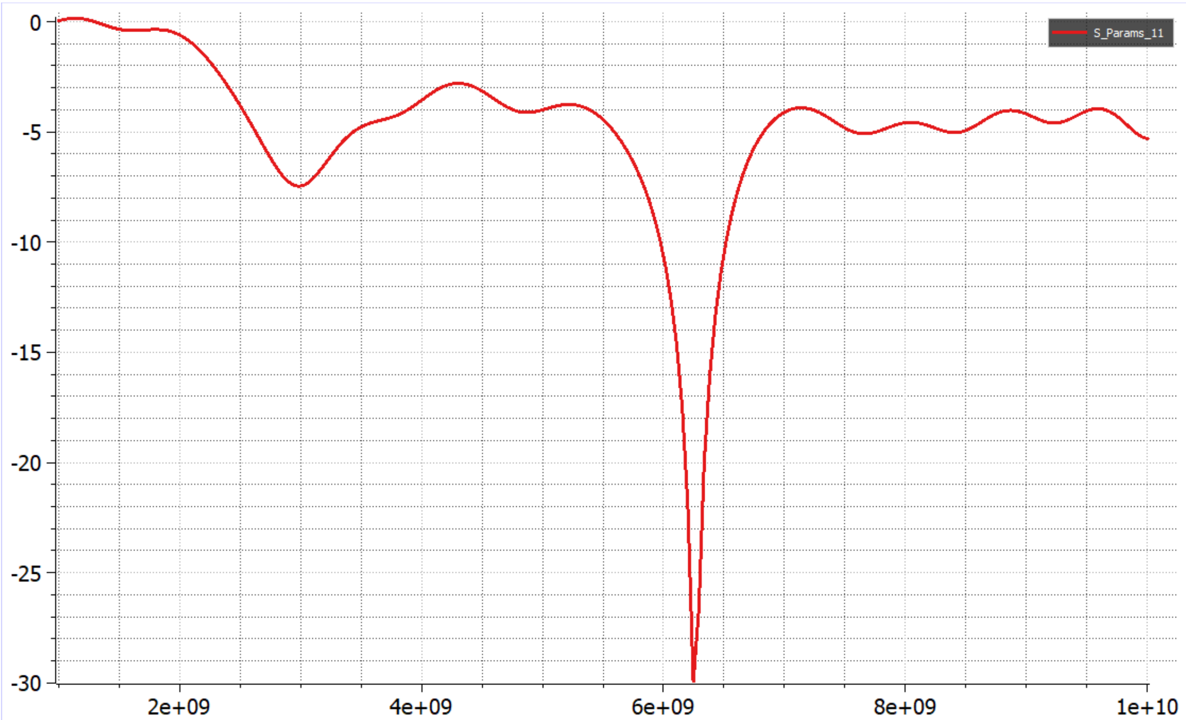
Running the Simulation

1. To run the simulation, click on the Run button  in the taskbar.
2. Select FDTD setup solver and assign the necessary time steps (Tmax). The other parameters can be kept as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.



Obtaining the Scattering parameters

1. To view the results of the simulation, click on the results operation in the taskbar.
2. Select the plots that require analysis and view them through the toolbar in the results tab.
3. Here are the S11 parameters obtained after simulation through the FDTD solver over 15000-time steps.



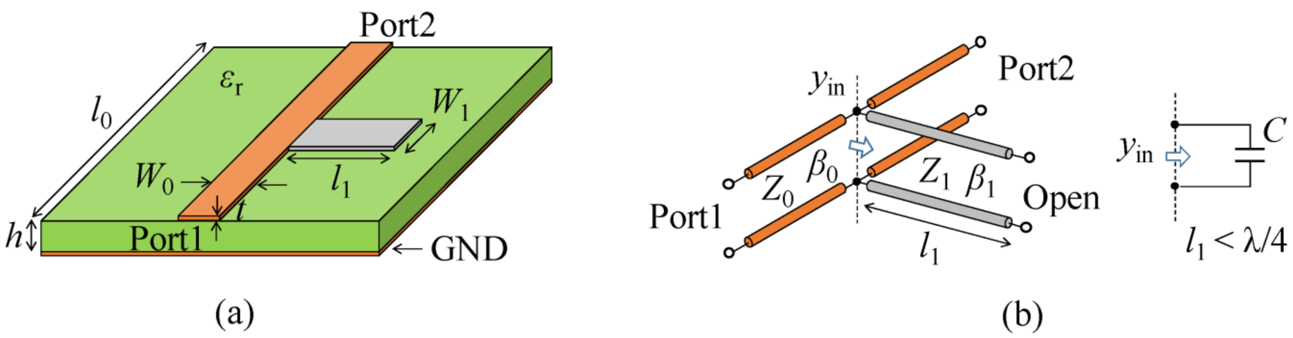
S11 results for E-shaped monopole antenna

Result Analysis

From the obtained results, it can be observed that the E-shaped monopole antenna has high a steep dip at the 6.25Ghz mark which coincides with the obtained result in the above cited reference paper. Hence the obtained results are consistent with the dimensions of the design and the E-shaped monopole antenna works as intended.

7. Simulation and Analysis of a Stub Based Band Stop Filter

A stub-based band-stop filter is an RF/microwave filter that attenuates a certain range of frequencies while allowing signals outside this range to flow through with minimum loss. It is commonly used in wireless communication, radar, and signal processing applications to remove undesirable interference or harmonic frequencies. The filter is built using transmission line stubs that may be either open or short-circuited. These stubs are carefully positioned along the main transmission line, generating destructive interference at undesirable frequencies and essentially blocking them. The stubs' resonance frequency controls the centre of the stopband, while their quantity and design regulate the bandwidth and depth of attenuation. Stub-based band-stop filters have various advantages, including small size, minimal insertion loss, and easy integration onto printed circuit boards. They are especially effective in high-frequency applications where lumped-element filters are unsuitable owing to parasitic effects. However, these filters are susceptible to manufacturing tolerances, which can impair their effectiveness. Overall, stub-based band-stop filters are an effective and adaptable method for rejecting certain frequencies in RF systems. Their straightforward design and efficacy make them a popular choice for modern communication and signal processing applications.



Stub based Band Stop Filter

This example demonstrates how to design a Stub based Band Stop Filter on TaraNG to simulate and analyse its various parameters such as Reflection coefficient, Gain, Radiation pattern and so on.

Geometry Design

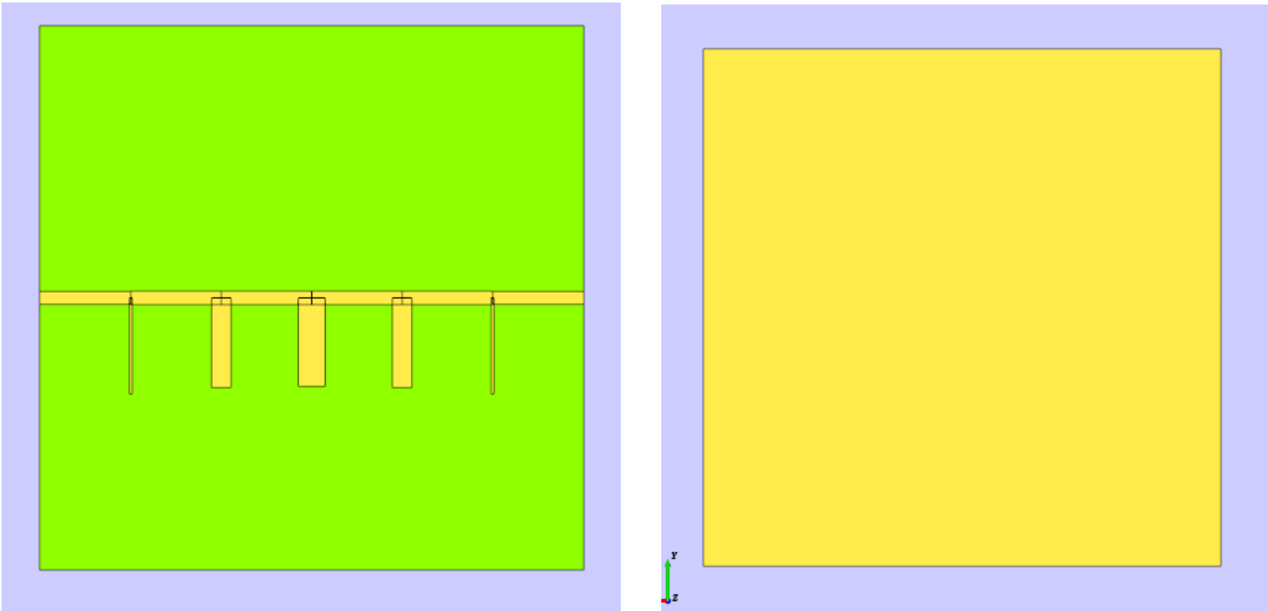
Depending on the passband and stop band frequency ranges, the number of stubs, types of stubs (shunt stubs or series stubs), and the width of the stubs will change. The designed band stop filter is supposed to have a stop band between the frequencies 1.5Ghz and 2.5Ghz. Using appropriate formulae to suit this criterion we can find the width and length of each of the individual stubs.

The construction of the Band stop filter can be divided into two parts.

- Substrate
- Patch (microstrip transmission line and the stubs and the ground)

<u>The Stub Based Band Stop Filter structure:</u>

This structure follows the two elliptic curves forming a polygon outwardly by connecting with their left and right edges respectively and meeting up at the bottom edge.



Stub based Band Stop Filter Design front and rear view respectively

This design can be achieved by knowing the dimensions of each of the individual components (microstrip line, Stubs and substrate and the Ground) and using the cad environment to design the boxes and rectangles with the corresponding dimensions. Alternatively, the Code can also be used to achieve the same. Shown below is the code that achieves the same purpose with the added advantage of easy modifications with the use of variables.

Code (Mode: Enable AUTO)

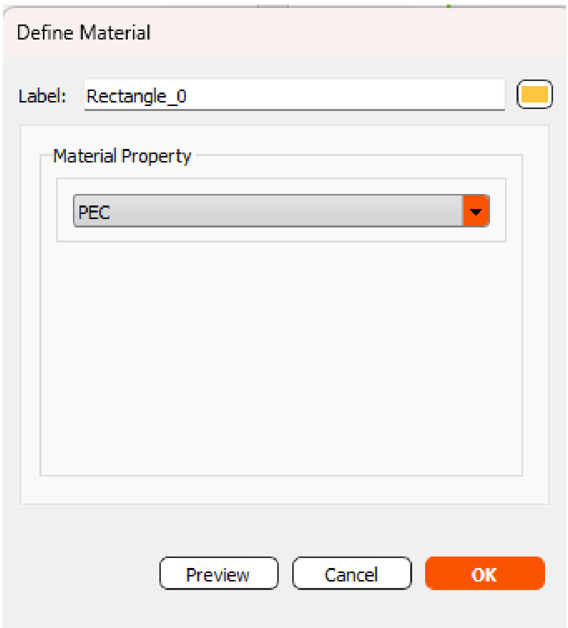
```
Project = initProject()
a=60.71; l=20.35; h=1.6; w=2.9; w1=3.01604; l1=20.18; l2=l1; w2=w1; l3=l1; w3=w1; l4=l1; w4=w1;
wo1= 0.618664; wo2= 4.39313; wo3=5.95366; wo4=wo2; wo5=wo1;
lo1= 21.44015; lo2= 20.02986; lo3= 19.760415; lo4=lo2; lo5=lo1;
Box_0 = Box(StartPoint = [-a, -a, 0], L = 2*a, B = 2*a, H = h)
Box_0.AddTo(Project,'Box_0')
Rectangle_0 = Rectangle(StartPoint=[-60.71, 60.71, 0], A=121.42, B=-121.42)
Rectangle_0.AddTo(Project,'Rectangle_0')
Rectangle_1 = Rectangle(StartPoint=[-a, -w/2, 1.6], A=l, B=w)
Rectangle_1.AddTo(Project,'Rectangle_1')
Rectangle_2 = Rectangle(StartPoint=[-a+l, -w1/2, 1.6], A=l1, B=w1)
Rectangle_2.AddTo(Project,'Rectangle_2')
Rectangle_3 = Rectangle(StartPoint=[-a+l+l1, -w2/2, 1.6], A=l2, B=w2)
Rectangle_3.AddTo(Project,'Rectangle_3')
Rectangle_4 = Rectangle(StartPoint=[a-l-l4, -w3/2, 1.6], A=-l3, B=w3)
Rectangle_4.AddTo(Project,'Rectangle_4')
Rectangle_5 = Rectangle(StartPoint=[a-l, -w4/2, 1.6], A=-l4, B=w4)
Rectangle_5.AddTo(Project,'Rectangle_5')
Rectangle_6 = Rectangle(StartPoint=[a, -w/2, 1.6], A=-l, B=w)
Rectangle_6.AddTo(Project,'Rectangle_6')
Rectangle_7 = Rectangle(StartPoint=[-a+l-wo1/2, -lo1, 1.6], A=wo1, B=lo1)
Rectangle_7.AddTo(Project,'Rectangle_7')
Rectangle_8 = Rectangle(StartPoint=[-a+l+l1-wo2/2, -lo2, 1.6], A=wo2, B=lo2)
Rectangle_8.AddTo(Project,'Rectangle_8')
Rectangle_9 = Rectangle(StartPoint=[-wo3/2, -lo3, 1.6], A=wo3, B=lo3)
Rectangle_9.AddTo(Project,'Rectangle_9')
Rectangle_10 = Rectangle(StartPoint=[a-l-l4+wo4/2, -lo4, 1.6], A=-wo4, B=lo4)
Rectangle_10.AddTo(Project,'Rectangle_10')
Rectangle_11 = Rectangle(StartPoint=[a-l+wo5/2, -lo5, 1.6], A=-wo5, B=lo5)
Rectangle_11.AddTo(Project,'Rectangle_11')
Rectangle_12 = Rectangle(StartPoint=[-60.71, 1.45, 1.6], A=-2.9, B=-1.57)
```

Code for the Stub based Band Stop Filter Design

Setting up the Excitation and assigning material properties

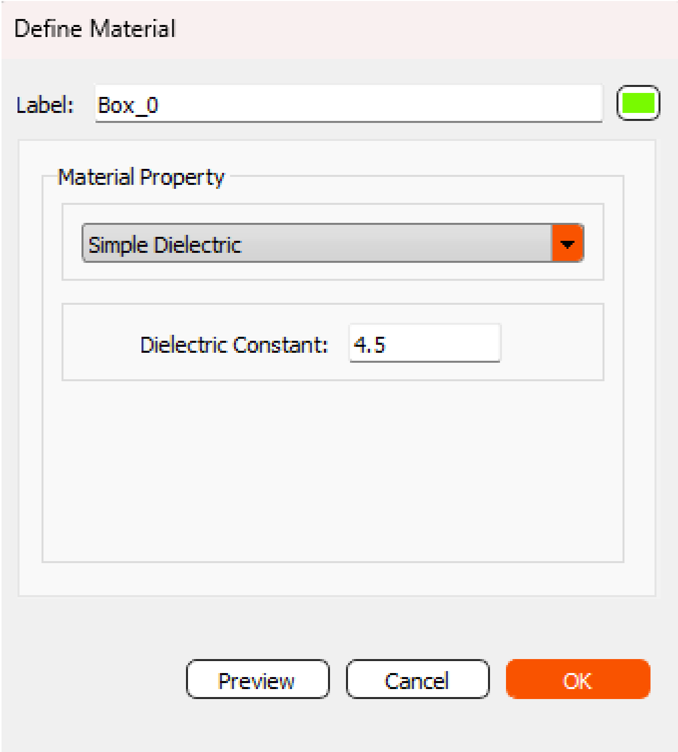
To assign material properties and port excitation click on CEM operation in the TaraNG Taskbar.

1. Assign PEC material to the microstrip line, the stubs and the Ground



2. To add the feed, click on the 'Port'  icon and trace the arrow joining any two edges of the rectangle in the XZ plane alongside the feedline.

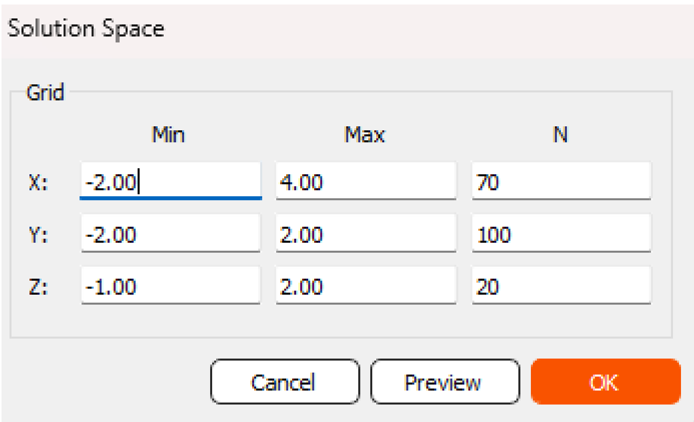
3. Assign Simple Dielectric property to the Substrate box with the necessary Effective permittivity value. In this instance it is 4.5



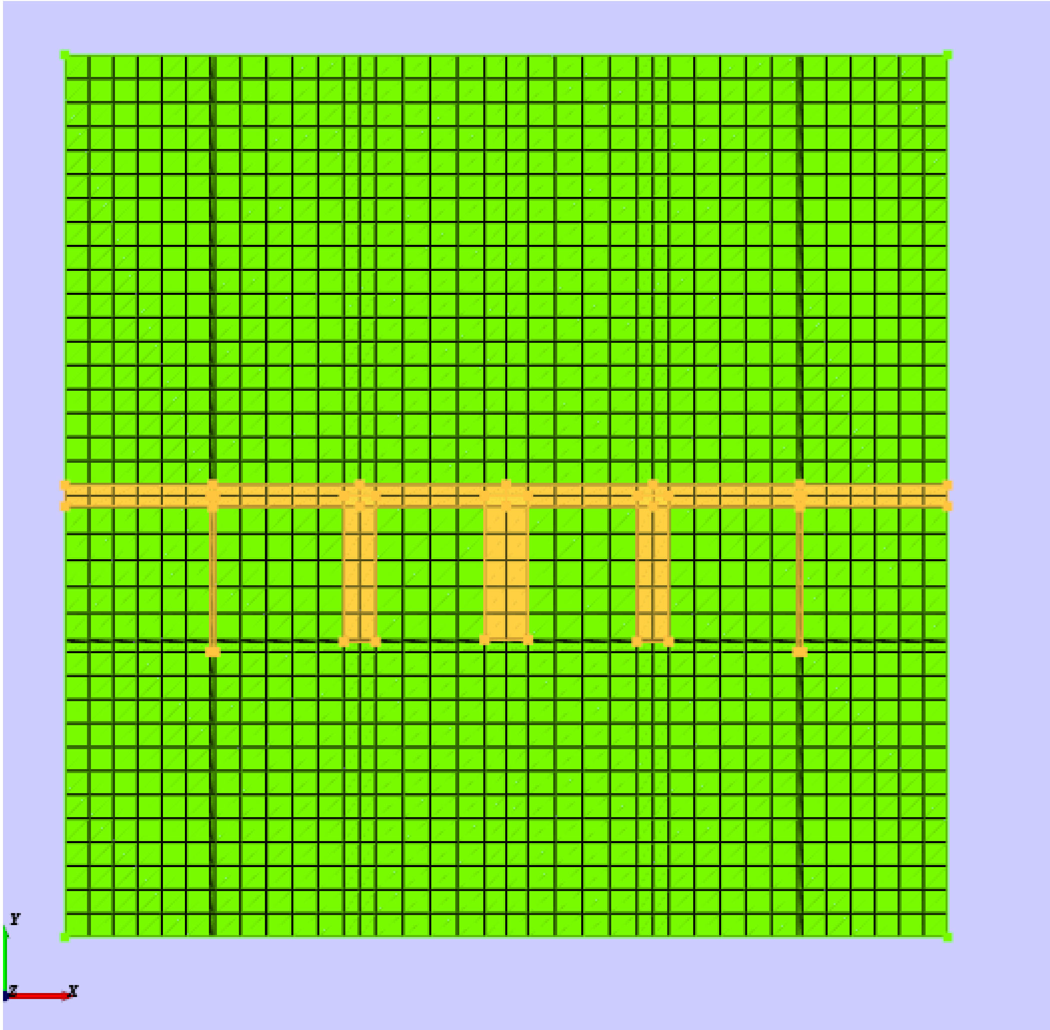
Meshing and Voxelization

To assign material properties and port excitation click on Mesh operation in the TaraNG Taskbar.

1. Click on the  icon in the tool bar and provide the appropriate dimensions for the meshing box along with the necessary number of divisions for the mesh cells



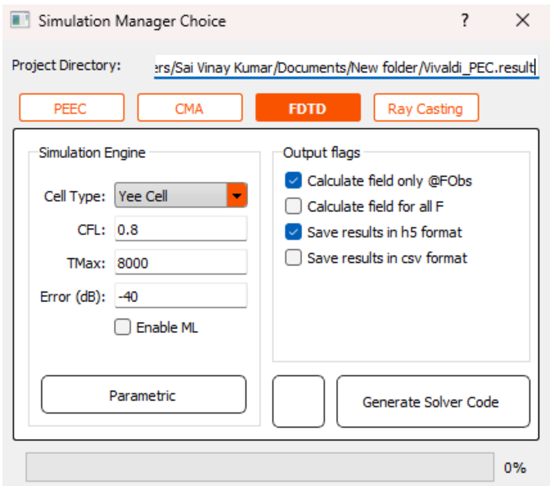
2. After selecting the grid of appropriate dimensions and divisions click on Run to get the following meshed structure.



Meshed View

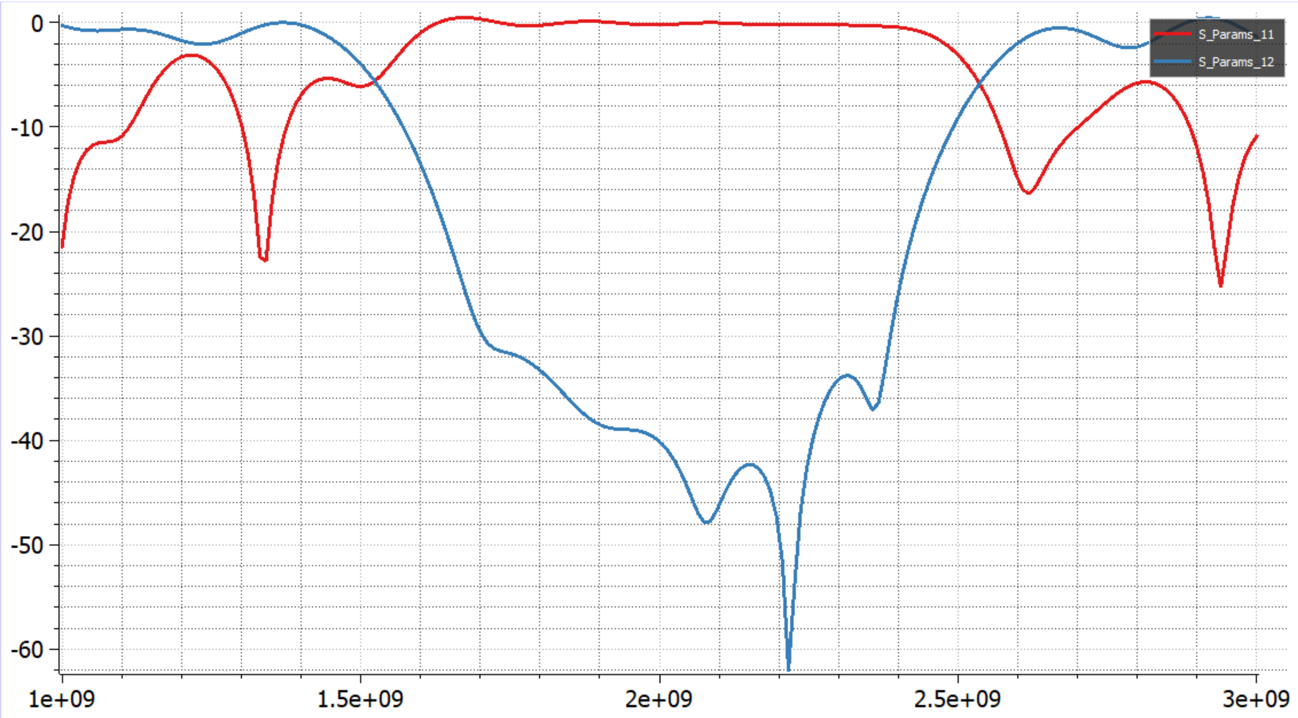
Running the Simulation

1. To run the simulation, click on the Run button in the taskbar.
2. Select FDTD setup solver and assign the necessary time steps (Tmax). The other parameters can be kept as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.



Obtaining the Scattering parameters

1. To view the results of the simulation, click on the results operation in the taskbar.
2. Select the plots that require analysis and view them through the toolbar in the results tab.
3. Here are the S11 and S12 parameters obtained after simulation through the FDTD solver over 15000-time steps.



S11 and S12 results for Stub Based band stop filter

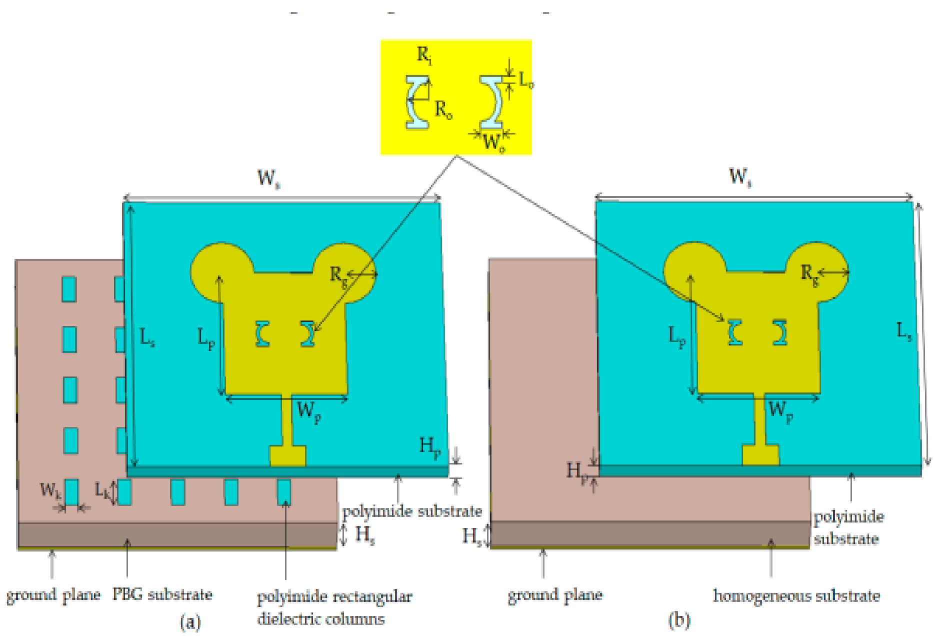
Result Analysis

From the obtained results, it can be observed that the Band stop filter has high reflection and low transmission between the frequencies 1.5GHz and 2.5GHz. This means that the mentioned frequency range is the stop band. Hence the obtained results are consistent with the dimensions of the design and the band stop filter works as intended.

8. Simulation and Analysis of a High-Gain and Tri-Band Terahertz Microstrip Antenna Using a Polyimide Rectangular Dielectric Column Photonic Band Gap Substrate

A high-gain and tri-band terahertz microstrip antenna with a photonic band gap (PBG) substrate finds its application in terahertz communications and other Photonics applications. Polyimide dielectric columns are inserted into the silicon substrate to form the PBG substrate to improve the gain of the antenna. The PBG substrate and polyimide substrate constituted a multilayer substrate structure and enabled the multi-band operation of the antenna.

This example demonstrates how to design a High-Gain and Tri-Band Terahertz Microstrip Antenna Using a Polyimide Rectangular Dielectric Column Photonic Band Gap Substrate on TaraNG, to simulate and analyse its various parameters such as Reflection coefficient, Gain, Radiation pattern and so on.



Tri-Band Terahertz Microstrip Antenna

This design is based on the below cited reference paper

[A. A. Megahed, M. Abdelazim, E. H. Abdelhay and H. Y. M. Soliman, "Sub-6 GHz Highly Isolated Wideband MIMO Antenna Arrays," in IEEE Access, vol. 10, pp. 19875-19889, 2022, doi: 10.1109/ACCESS.2022.3150278.]

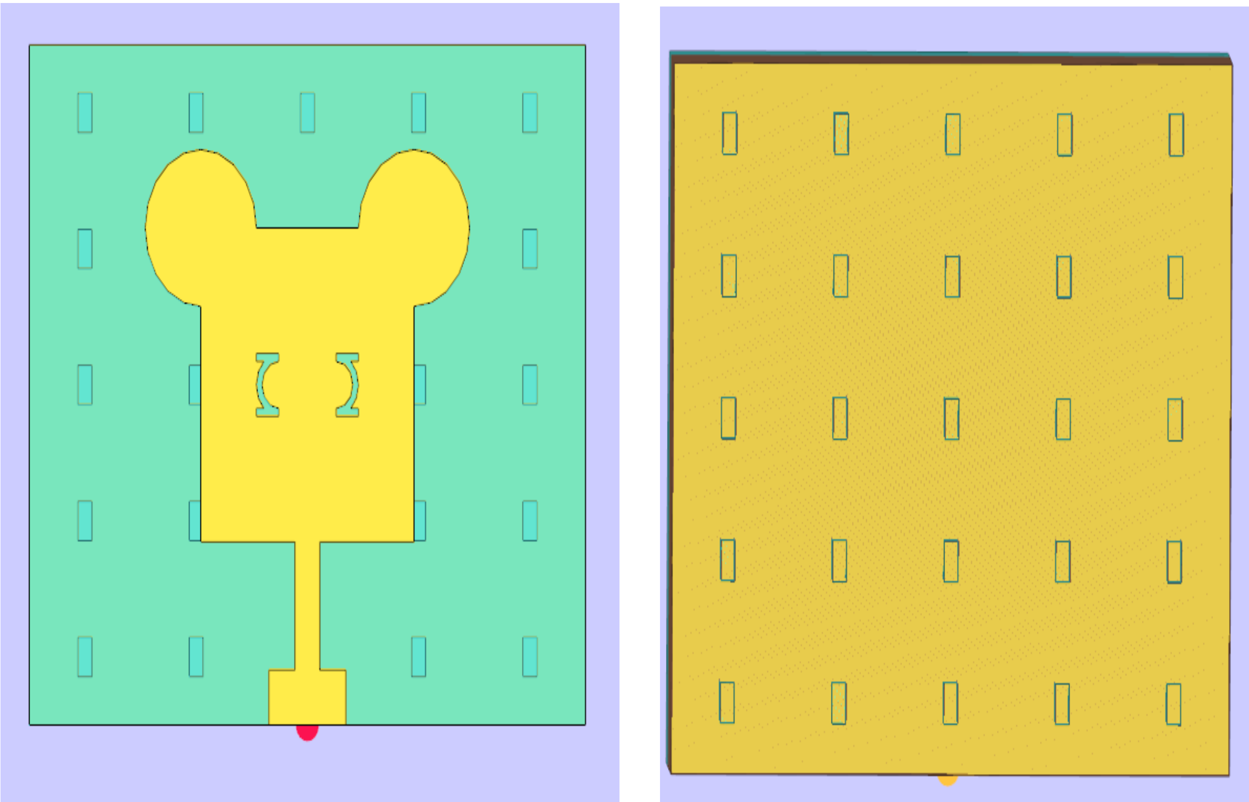
Importing Structures

Structures that are large or complicated to design can be imported from other CAD software and environments using the following procedure.

1. Save the designs exported from other CAD software as ". stl" files
2. Start Tarang and select the import icon in the CAD toolbar shown as 
3. Once a dialog box opens asking for the structures that are meant to be imported select the appropriate .stl files from their respective locations.

This specific structure has been imported to TaraNG

Tri-Band Terahertz Microstrip Antenna structure:

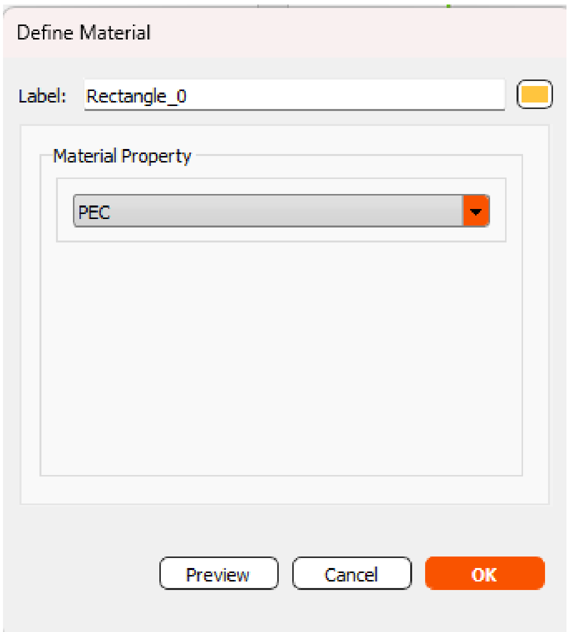


Tri-Band Terahertz Microstrip Antenna design front and rear view respectively

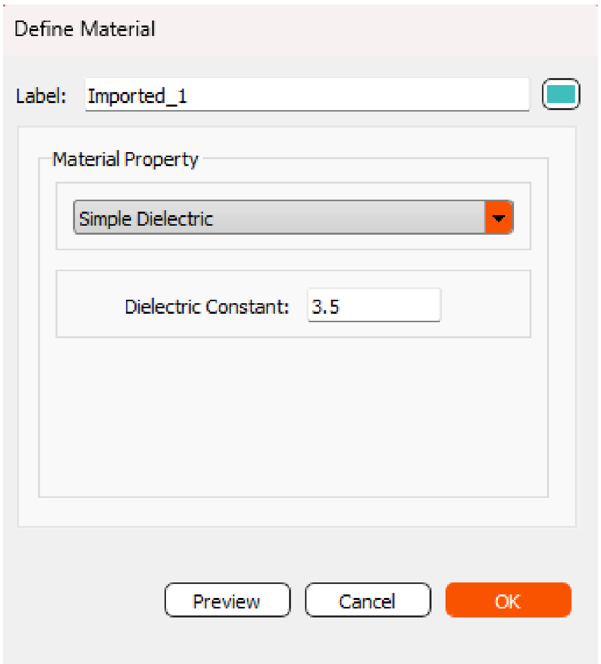
Setting up the Excitation and assigning material properties

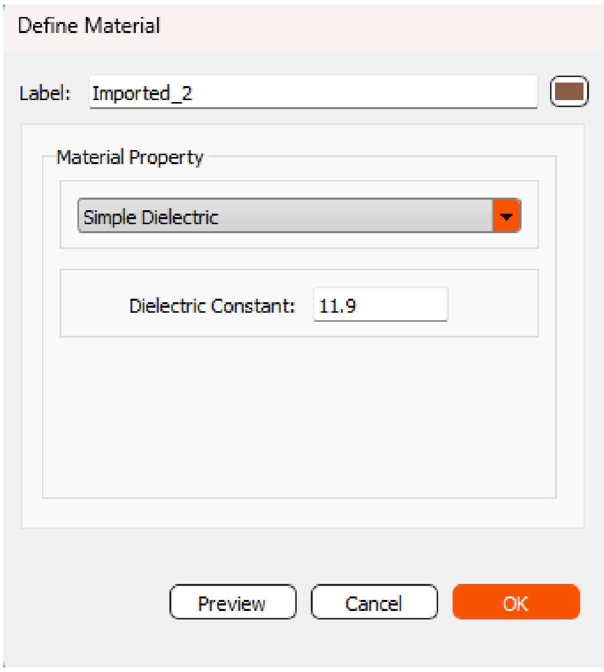
To assign material properties and port excitation click on CEM operation in the TaraNG Taskbar.

1. Assign PEC material to the patch and the ground



2. To add the feed, click on the 'Port'  icon and trace the arrow joining any two edges of the rectangle in the XZ plane alongside the feedline.
3. Assign Simple Dielectric property to the Substrate box with the necessary Effective permittivity value. In this instance it is 11.9 for the silicon substrate and 3.5 for the Polymide substrate

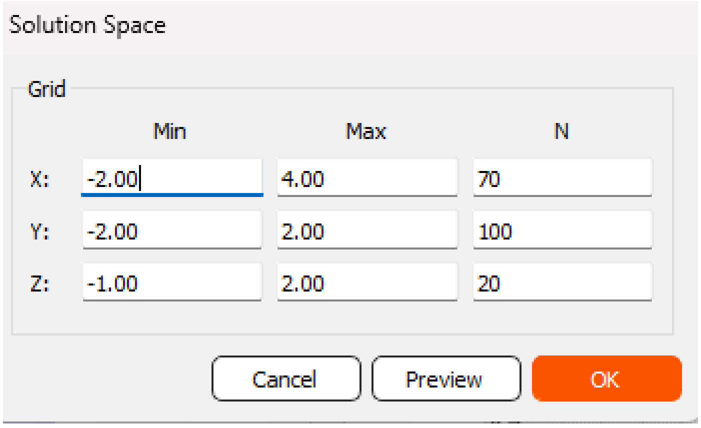




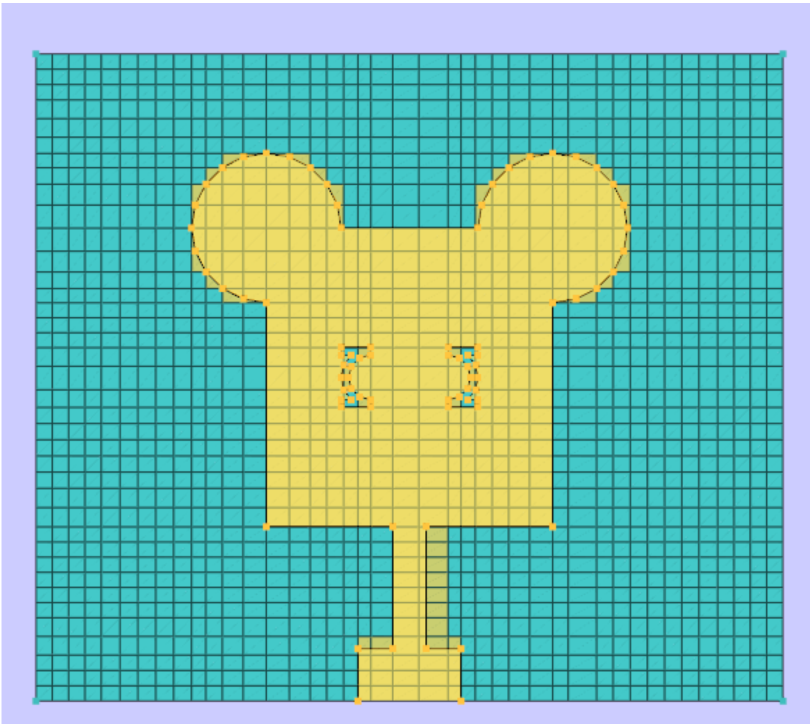
Meshing and Voxelization

To assign material properties and port excitation click on Mesh operation in the TaraNG Taskbar.

1. Click on the  icon in the tool bar and provide the appropriate dimensions for the meshing box along with the necessary number of divisions for the mesh cells

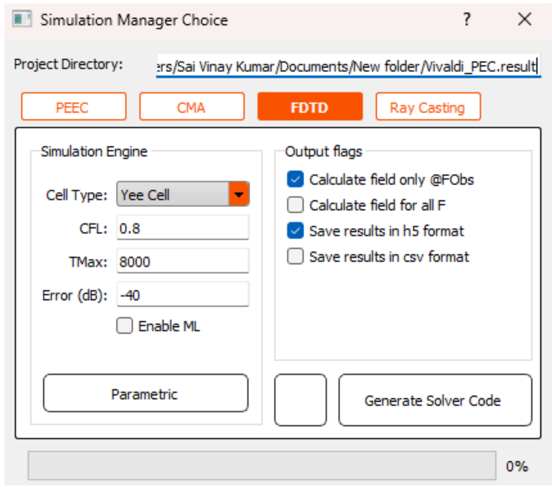


2. After selecting the grid of appropriate dimensions and divisions click on Run to get the following meshed structure.



Meshed View

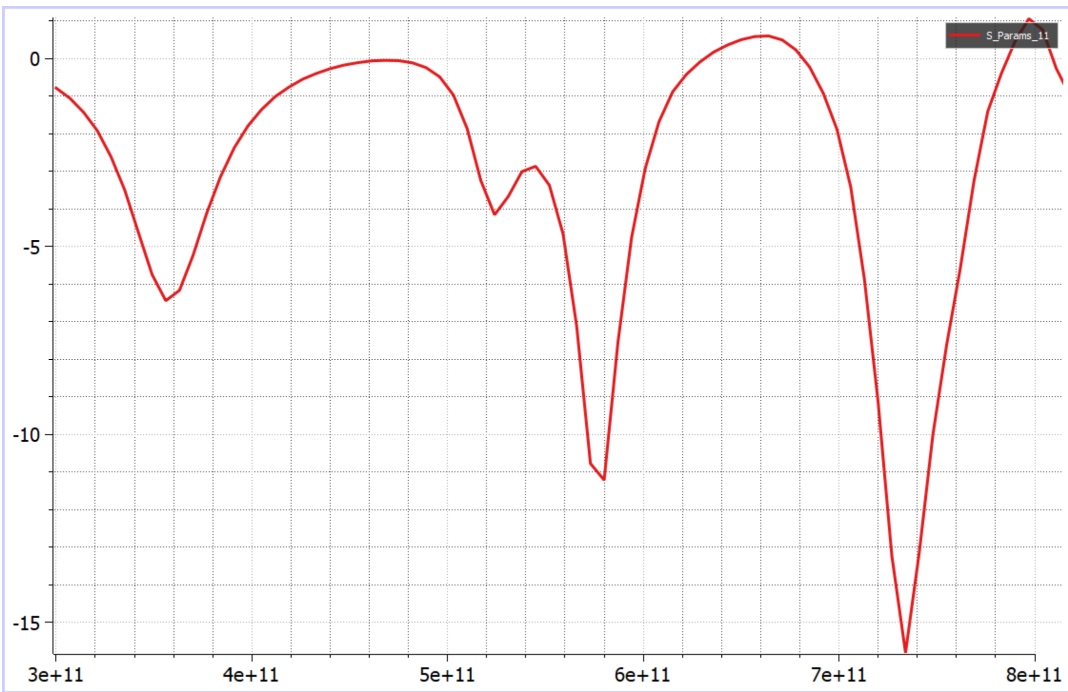
Running the Simulation



1. To run the simulation, click on the Run button  in the taskbar.
2. Select FDTD setup solver and assign the necessary time steps (Tmax). The other parameters can be kept as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.

Obtaining the Scattering parameters

1. To view the results of the simulation, click on the results operation in the taskbar.
2. Select the plots that require analysis and view them through the toolbar in the results tab.
3. Here are the S11 parameters obtained after simulation through the FDTD solver over 15000-time steps.



S parameter results for the Tri-Band Terahertz Microstrip Antenna

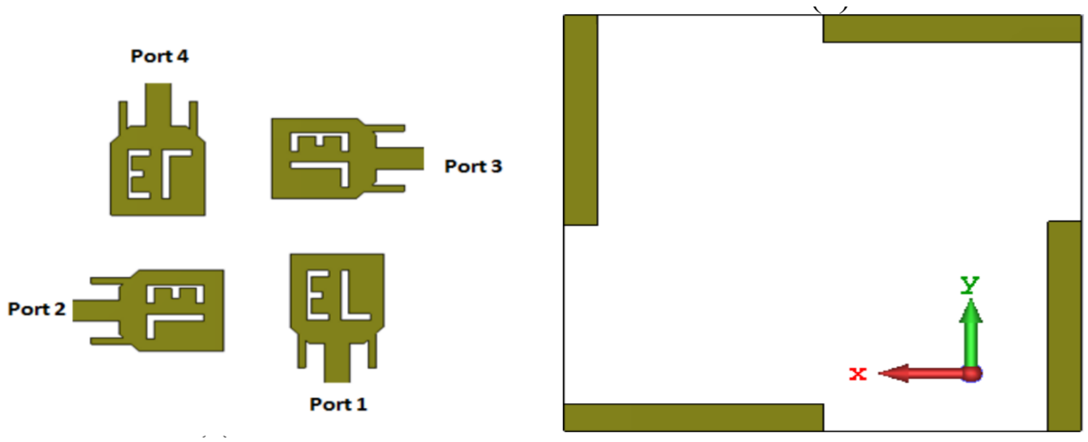
Result Analysis

From the obtained results, it can be observed that the Tri-Band Terahertz Microstrip Antenna Has three different bands in which it radiates. It displays the best results between 0.71THz to 0.75THz. This makes it useful for very high frequency applications.

9. Simulation and Analysis of Sub-6 GHz Highly Isolated Wideband MIMO Antenna Array

Small size wideband microstrip patch antenna with partial ground design is a major challenge. Designing a small, wideband microstrip patch antenna with partial grounding is a significant task. MIMO (Multiple Input Multiple Output) antenna technology is one of the most essential technologies in the 5G wireless communication system. This technology develops spectrum efficiency, energy, and cost efficiency. MIMO has been widely used in different systems to significantly develop channel capacity.

This example demonstrates how to design a Sub-6 GHz Highly Isolated Wideband MIMO Antenna Array on TaraNG, to simulate and analyse its various parameters such as Reflection coefficient, Gain, Radiation pattern and so on. This structure is a 4 port MIMO antenna that is useful for Ultra-wideband applications while still remaining compact in nature. It incorporates a DGS (Defected Ground System) for enhanced isolation and better Ultrawideband characteristics.



Sub-6 GHz MIMO Antenna Array

This design is based on the below cited reference paper

[A. A. Megahed, M. Abdelazim, E. H. Abdelhay and H. Y. M. Soliman, "Sub-6 GHz Highly Isolated Wideband MIMO Antenna Arrays," in IEEE Access, vol. 10, pp. 19875-19889, 2022, doi: 10.1109/ACCESS.2022.3150278.]

Importing Structures

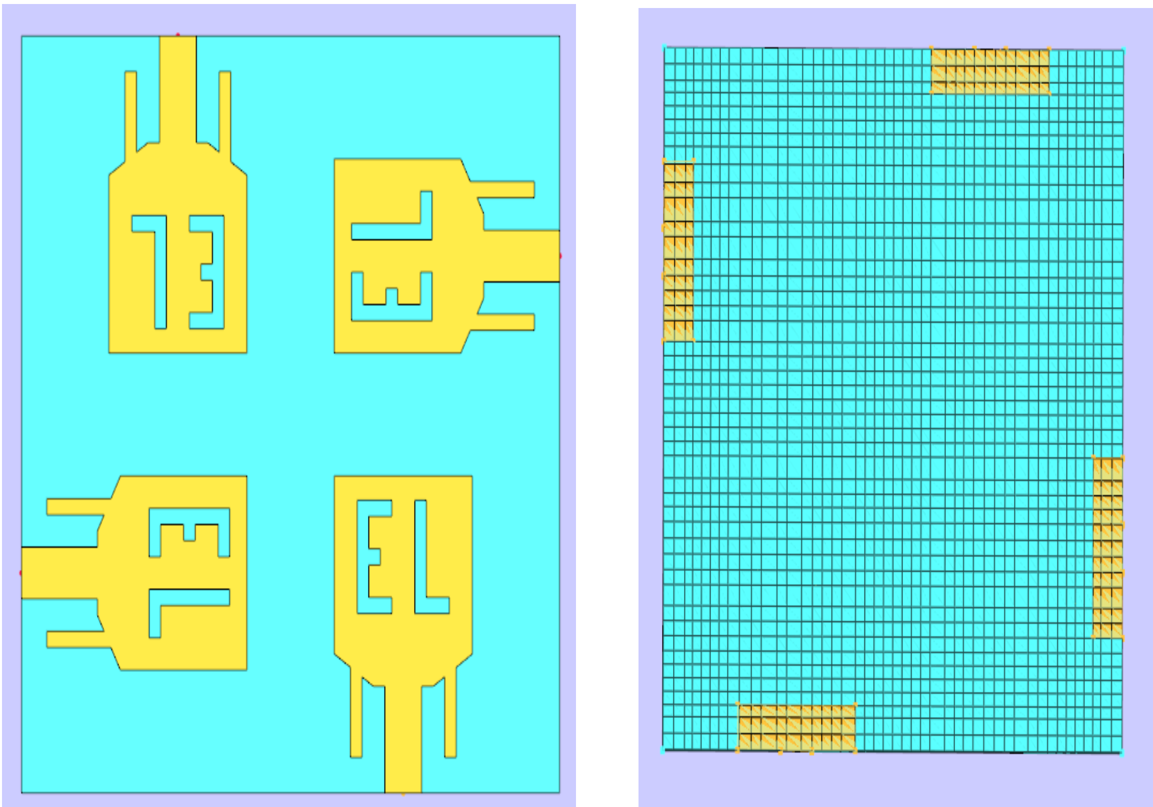
Structures that are large or complicated to design can be imported from other CAD software and environments using the following procedure.

1. Save the designs exported from other CAD software as ". stl" files

2. Start Tarang and select the import icon in the CAD toolbar shown as 
3. Once a dialog box opens asking for the structures that are meant to be imported select the appropriate .stl files from their respective locations.

This specific structure has been imported to TaraNG

<u>The Sub-6 GHz MIMO Antenna Array structure:</u>

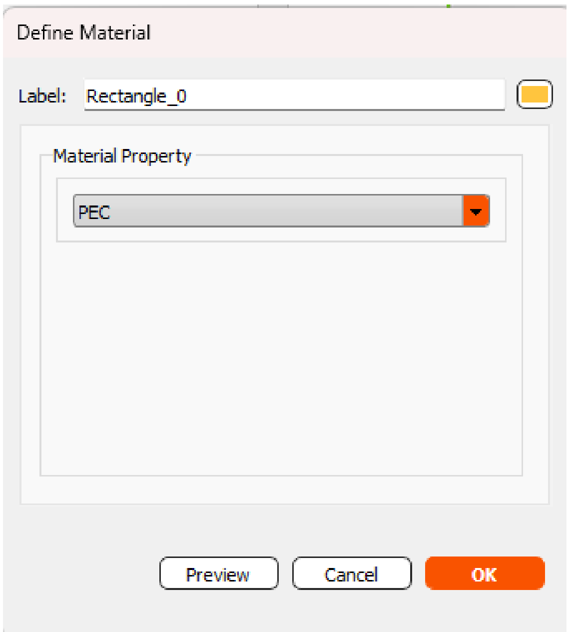


Sub-6 GHz MIMO Antenna Array design front and rear view respectively

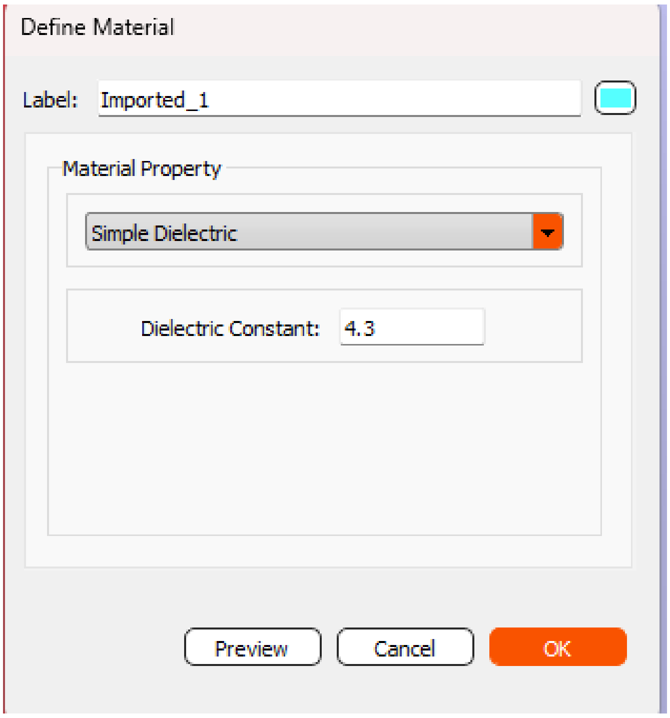
Setting up the Excitation and assigning material properties

To assign material properties and port excitation click on CEM operation in the TaraNG Taskbar.

1. Assign PEC material to the patch and the ground



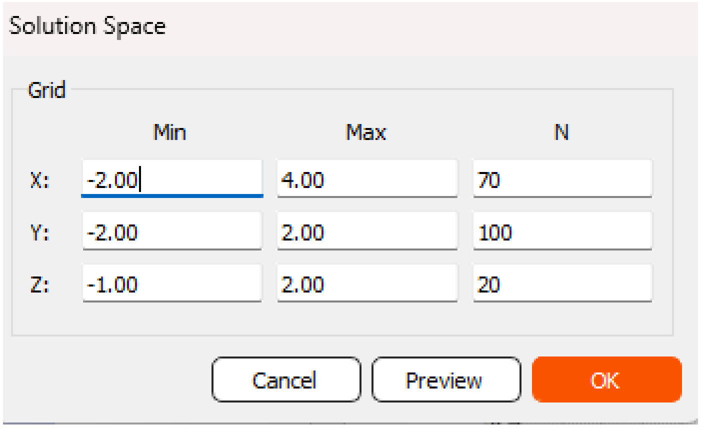
2. To add the feed, click on the 'Port'  icon and trace the arrow joining any two edges of the rectangle in the XZ plane alongside the feedline.
3. Assign Simple Dielectric property to the Substrate box with the necessary Effective permittivity value. In this instance it is 4.3



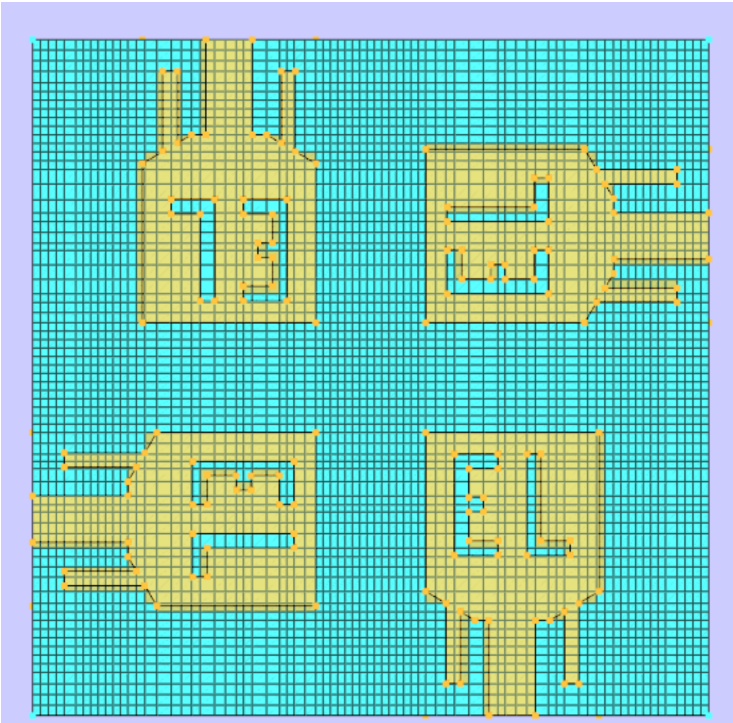
Meshing and Voxelization

To assign material properties and port excitation click on Mesh operation in the TaraNG Taskbar.

1. Click on the  icon in the tool bar and provide the appropriate dimensions for the meshing box along with the necessary number of divisions for the mesh cells



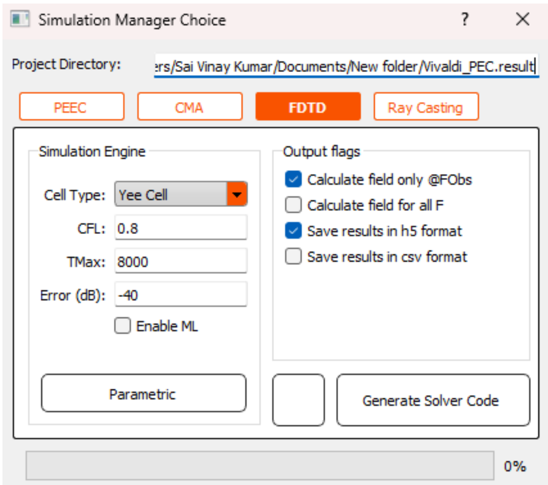
2. After selecting the grid of appropriate dimensions and divisions click on Run to get the following meshed structure.



Meshed View

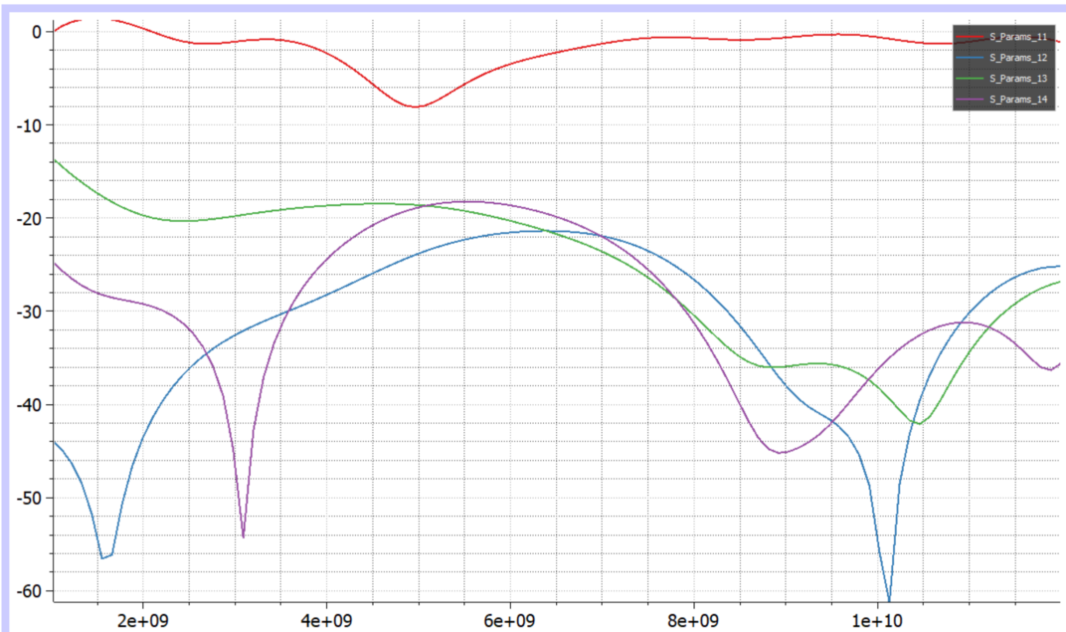
Running the Simulation

1. To run the simulation, click on the Run button  in the taskbar.
2. Select FDTD setup solver and assign the necessary time steps (Tmax). The other parameters can be kept as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.



Obtaining the Scattering parameters

1. To view the results of the simulation, click on the results operation in the taskbar.
2. Select the plots that require analysis and view them through the toolbar in the results tab.
3. Here are the S parameters obtained after simulation through the FDTD solver over 15000-time steps.



S parameter results Sub-6 GHz MIMO Antenna Array

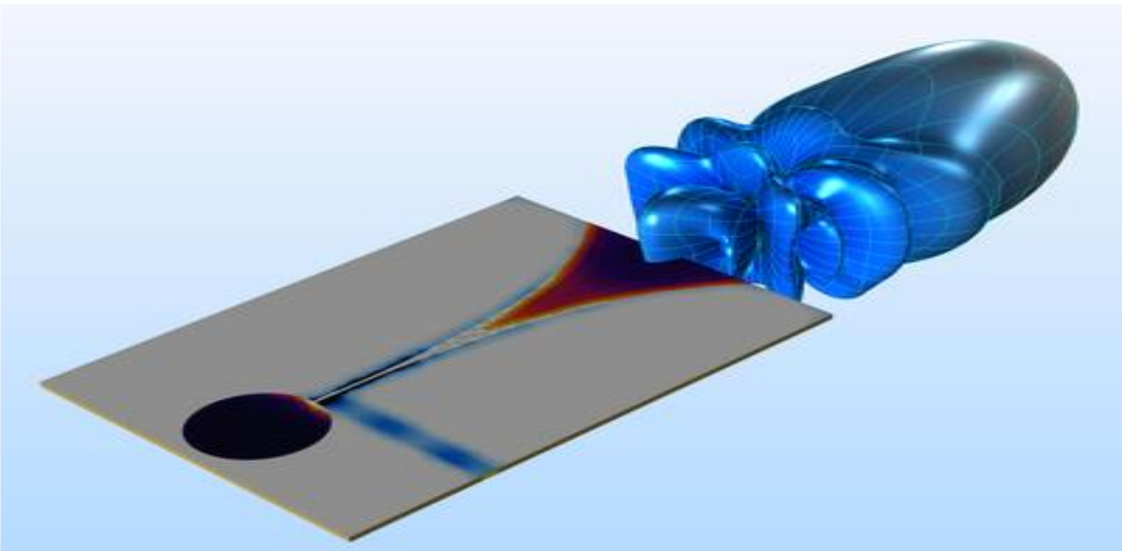
Result Analysis

From the obtained results, it can be observed that Sub-6 GHz MIMO Antenna Array displays ultrawideband properties between 3 to 12GHz. Thus, the MIMO antenna can be used for various UWB applications.

10. Simulation and Analysis of a High Frequency Fluidic Vivaldi antenna

A Vivaldi antenna is a type of tapered slot antenna designed for wide bandwidth and high directivity. Its low cost and ease of fabrication make it ideal for various applications, especially in wireless communications. By utilizing printed circuit technology, these antennas achieve compact designs without compromising performance.

The Vivaldi antenna is a technological breakthrough, providing increased capabilities for a wide range of applications. The Vivaldi antenna, with its characteristic tapered slot shape, blends form and function to provide excellent performance over a wide range of frequencies. This has made it the preferred alternative for engineers and technology enthusiasts looking for dependability and efficiency. Its capacity to accommodate a wide bandwidth while maintaining good directivity makes it suitable for both commercial and scientific applications, such as radar systems, wireless communications, and even space exploration. As we look more into its revolutionary characteristics and practical applications, it becomes evident why the Vivaldi antenna is unique in the world of antenna technology.



Vivaldi Antenna radiation pattern

This example demonstrates how to design a Vivaldi antenna on TaraNG to simulate and analyse its various parameters such as Reflection coefficient, Gain, Radiation pattern and so on.

Geometry Design

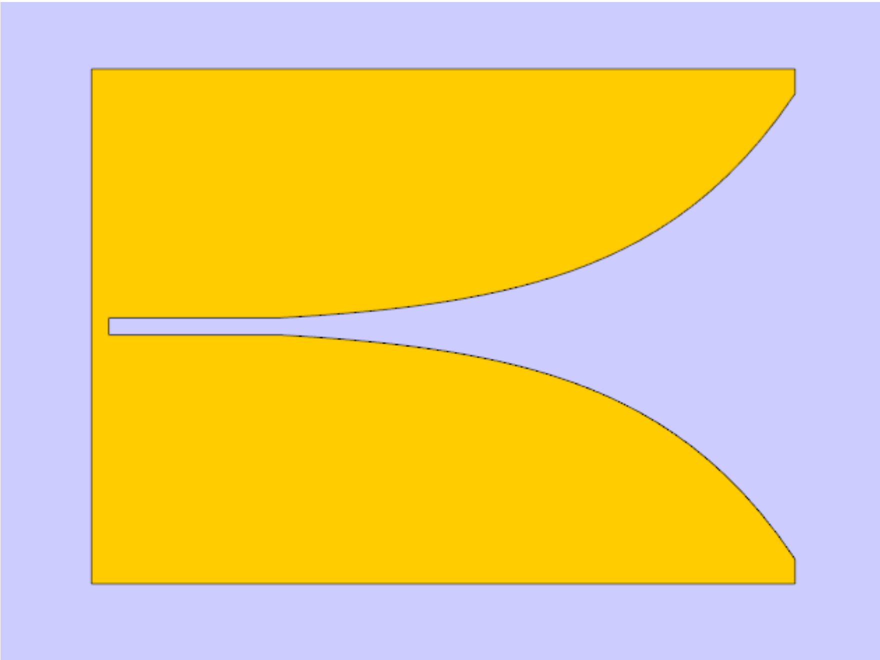
A Vivaldi antenna is not a conventional structure with straight edges. In order to design the structure on the CAD environment the use of Polygon is necessary. Structures like these are more easily and efficiently designed via code. Designing it via code also makes it easier to modify the antenna depending on the user's requirements such as the antenna's resonant frequency and so on. The Polygon is a design tool in TaraNG that is used to design irregularly shaped objects. It requires the user to specify the various points contribute to all the segments in the structure. If the structure is simple enough then the polygon tool can be used directly from the CAD toolbar. However, for a structure that requires the knowledge of curves between two points it is faster to use coding instead.

The construction of the Vivaldi antenna can be divided into two parts.

- The Outer Vivaldi structure
- The Inner Vivaldi structure

<u>The Outer Vivaldi structure:</u>

This structure follows the two elliptic curves forming a polygon outwardly by connecting with their left and right edges respectively and meeting up at the bottom edge.



Outer Vivaldi structure

The simple code to design this structure is shown below. It requires the knowledge of the two exponential curves, the rectangular slot extending below the curves and the various corners connecting the outer edges of this structure.

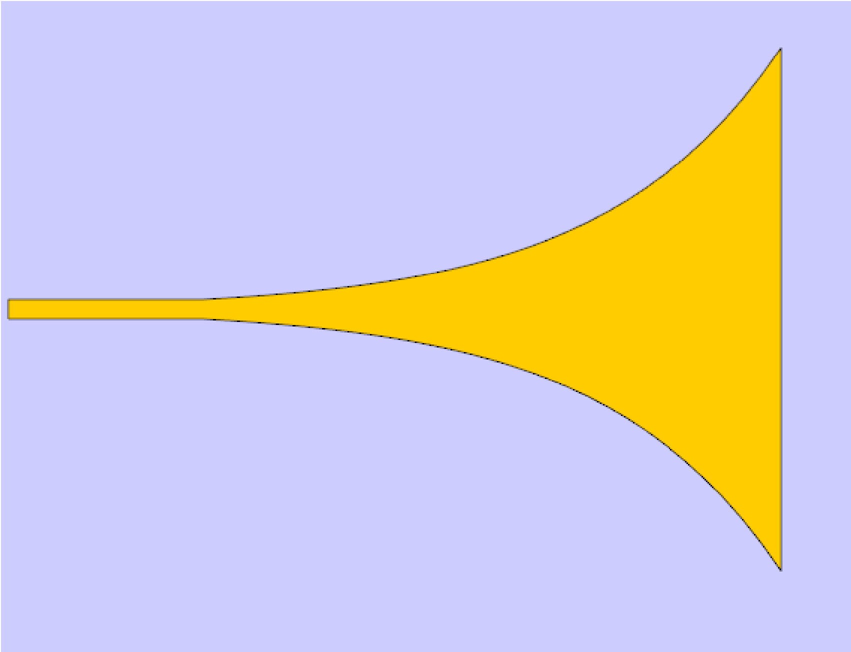
```
Project = initProject()
sl=1; ext=0.1; s=0.05; r=1.1;
def myVivaldi(r=1.1, s=0.05, sl=1):
        points = []; Segs = 30;  sl=1;
        for T in range(Segs+1):
                t = 3*T/Segs;
                points.append([t, -s*exp(r*t),0]);
        points.append([t, -s*exp(r*t)-0.144, 0]); points.append([-sl-0.1, -s*exp(r*t)-0.144, 0]);
        points.append([-sl-0.1, s*exp(r*t)+0.144, 0]); points.append([t, s*exp(r*t)+0.144, 0]);
        for T in range(Segs, -1, -1):
                t = 3*T/Segs;
                points.append([t, s*exp(r*t),0]);
        points.append([-sl, s, 0]); points.append([-sl, -s, 0]);
        return points

Polygon_0 = Polygon(Points=myVavaldi(r=1.1, s=0.05, sl=1))
Polygon_0.AddTo(Project,'Polygon_0')
Polygon_0.SetColor('#ffaa00');
Polygon_0.Refresh();
```

Outer Vivaldi structure Code

From the code it is evident that the function "myVivaldi "is responsible for collecting all the points in the polygon. The reason it's more efficient to use coding to design the structure is while the polygon tool in the CAD toolbar is useful for structures with straight edges, for curves however they cannot be defined solely by their start and end points. Since the polygon is a collection of points, that would mean that the user would have to manually type out all the X and Y coordinates of all the points on the curve. This is a tedious task manually. This task is instead handled by the two "for" loops. The two for loops collect all the points on the curve and add them to the "points []" array. As can be seen in the code the equation of the elliptic curves is given by "-s*exp(r*t)" and "s*exp(r*t)" for the upward elliptic curve and the downward curve respectively. Here the term "s" stands for Slot width which is the width of the rectangular slot extending below the curves. The term "sl" stands for slot length which is the length of the rectangular slot. From the code it can be seen that the curves have 30 well defined points referred to as "Segs". These 30 points are calculated and appended to the points [] array via the for loop. If we increase the number of points on the curve i.e. increase the value of Segs from 30, then the elliptic curve becomes even more smooth however this increases the time it takes to design and simulate the structure. Upon constructing the outer Vivaldi polygon, the polygon can be provided with a certain thickness using the Sweep operation. The outer Vivaldi structure is enough to construct a simple Vivaldi antenna without fluid. Note that this structure changes depending on the resonant frequency of the Vivaldi antenna being designed. This would mean the size of the structure and the curvature of the exponentials would change depending on the design frequency.

The Inner Vivaldi structure:

The Inner Vivaldi structure follows the two elliptic curves forming a polygon inwardly by connecting with the top edge of the structure.



Inner Vivaldi structure

The simple code to design this structure is shown below. It only requires the knowledge of the two exponential curves and the rectangular slot extending below the curves. Upon providing these necessary points the polygon automatically connects the ends of the two elliptic curves to form a closed structure.

```
Project = initProject()
sl=1; ext=0.1; s=0.05; r=1.1;
def myVivaldi(r=1.1, s=0.05, sl=1):
        points = []; Segs = 30;  sl=1;
        for T in range(Segs+1):
                t = 3*T/Segs;
                points.append([t, -s*exp(r*t),0]);
        for T in range(Segs, -1, -1):
                t = 3*T/Segs;
                points.append([t, s*exp(r*t),0]);
        points.append([-sl, s, 0]); points.append([-sl, -s, 0]);
        return points

Polygon_0 = Polygon(Points=myVivaldi(r=1.1, s=0.05, sl=1))
Polygon_0.AddTo(Project,'Polygon_0')
Polygon_0.SetColor('#ffaa00');
Polygon_0.Refresh();
```

Inner Vivaldi structure Code

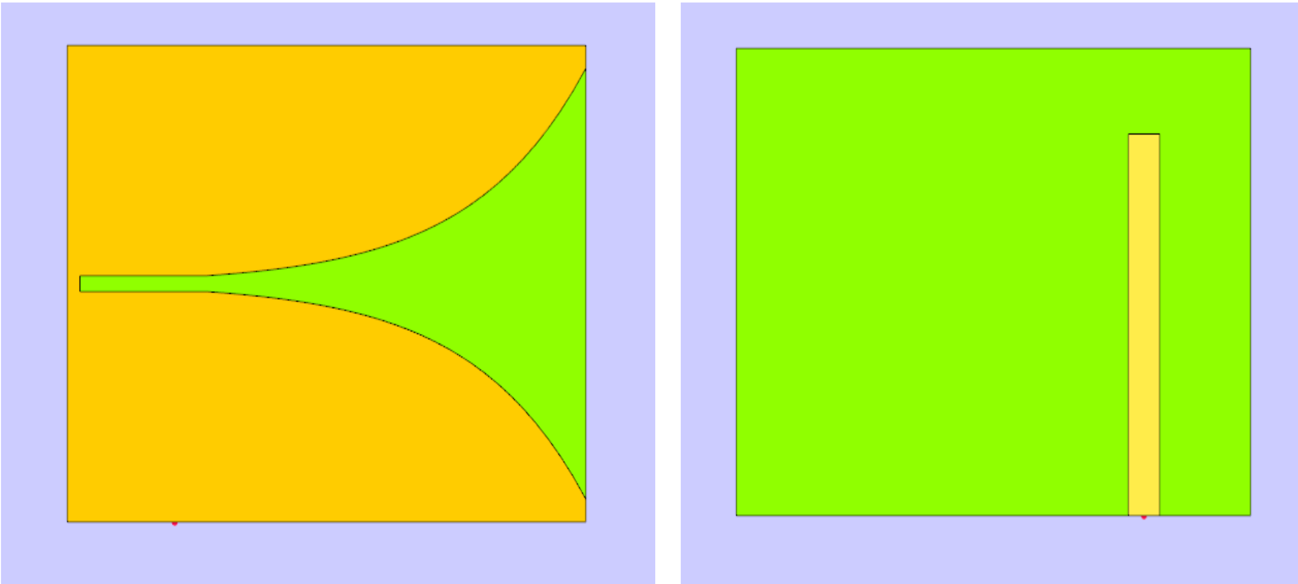
As is evident from the code, it is relatively the same as the outer Vivaldi structure but devoid of the edge points. This makes it so the designer takes the shortest path between the two end points of the curve thus forming the inner Vivaldi structure. The parameters are the same as the outer Vivaldi structure. The inner Vivaldi structure will be used when designing the fluidic Vivaldi antenna to provide it with fluidic properties. Note that this structure changes depending on the resonant frequency of the Vivaldi antenna being designed. This would mean the size of the structure and the curvature of the exponentials would change depending on the design frequency.

While defining polygons it is important to note that the points in the polygon are supposed to be defined in order i.e. the points should form a continuous loop. Changing the order in which the points are defined in the polygon may result in undesirable structures.

Vivaldi antenna design without Fluid

A Vivaldi antenna without fluid can be designed with these simple steps

1. Design the outer Vivaldi structure as shown previously.
2. Design the substrate using “Box” tool in the toolbar with the appropriate dimensions.
3. Design the feed line on the opposite side of the substrate to the outer Vivaldi structure.
4. Design a rectangle at the end of the feedline in the XZ plane to assign port excitation
5. Sweep the outer Vivaldi structure and the feedline for the necessary thickness.

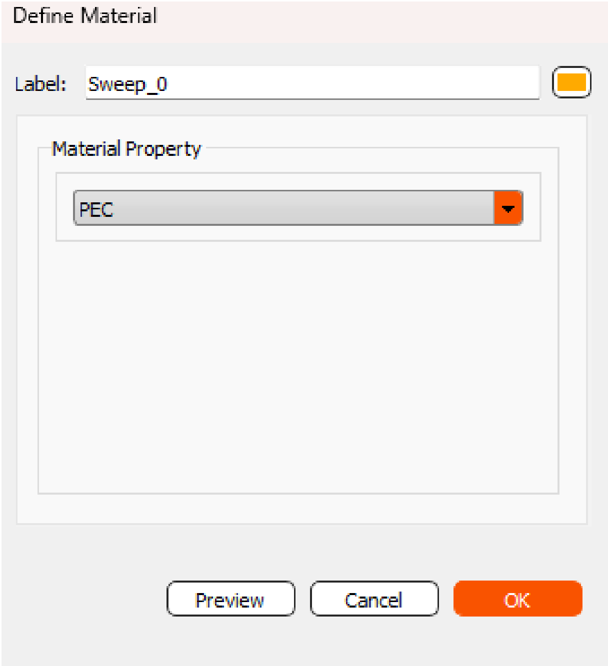


Front View and Rear view

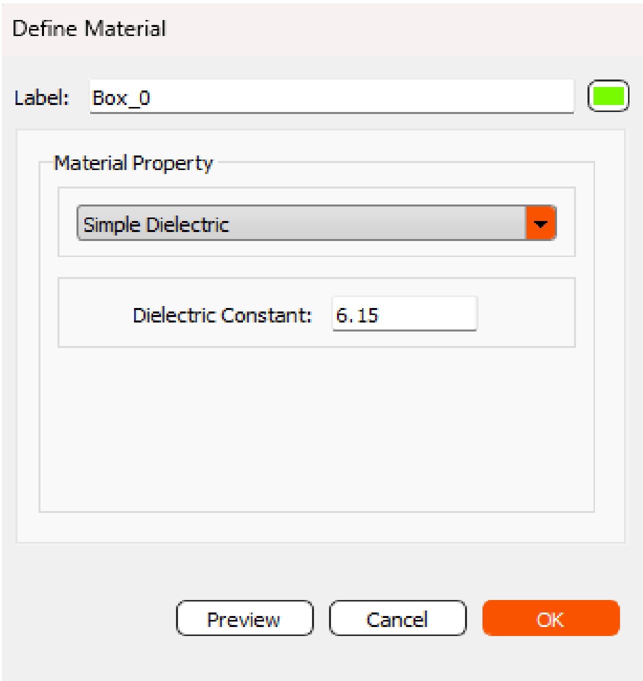
Setting up the Excitation and assigning material properties

To assign material properties and port excitation click on CEM operation in the TaraNG Taskbar.

1. Assign PEC material to the Swept outer Vivaldi structure and the feedline



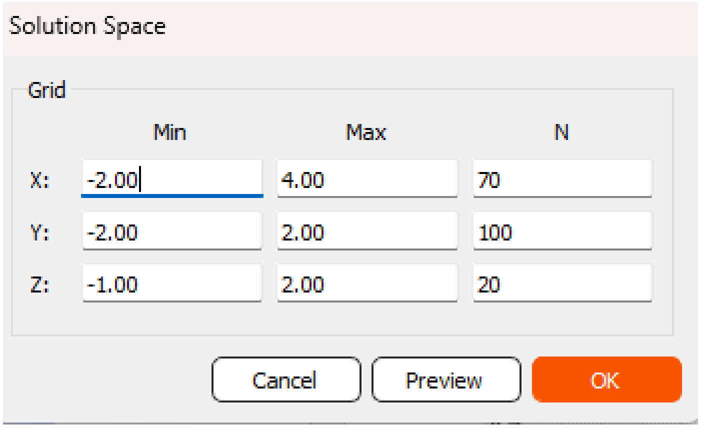
2. To add the feed, click on the ‘Port’  icon and trace the arrow joining any two edges of the rectangle in the XZ plane alongside the feedline.
3. Assign Simple Dielectric property to the Substrate box with the necessary Effective permittivity value. In this instance it is 6.15



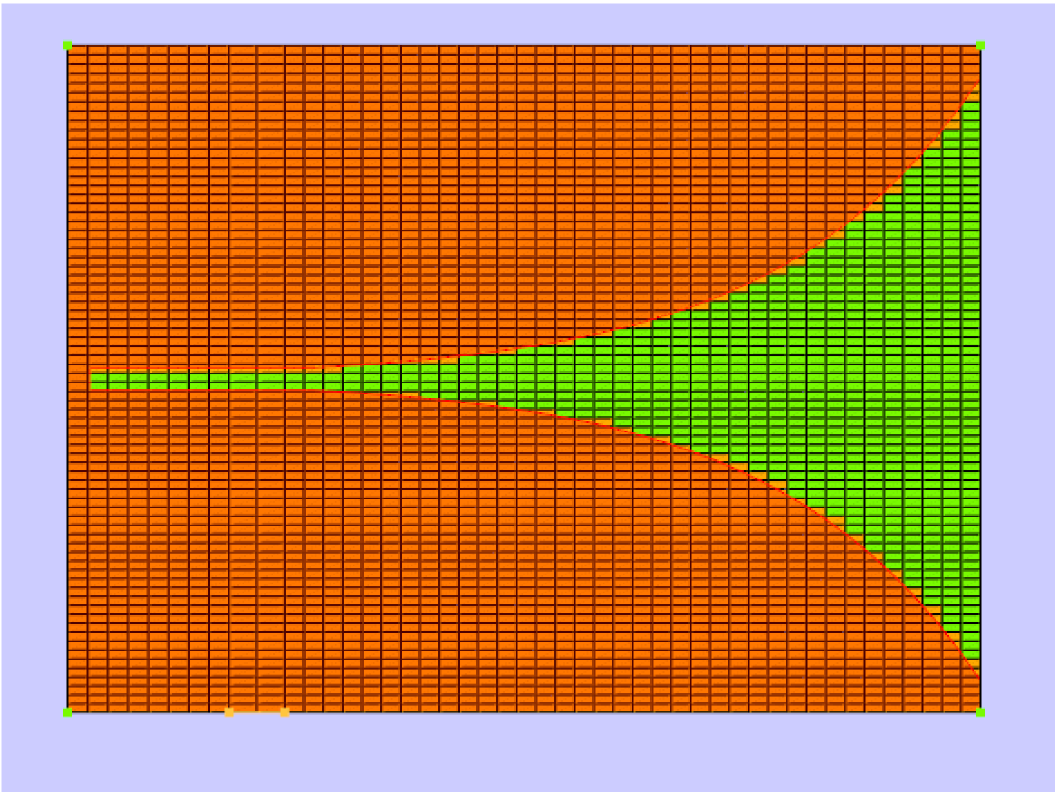
Meshing and Voxelization

To assign material properties and port excitation click on Mesh operation in the TaraNG Taskbar.

1. Click on the  icon in the tool bar and provide the appropriate dimensions for the meshing box along with the necessary number of divisions for the mesh cells

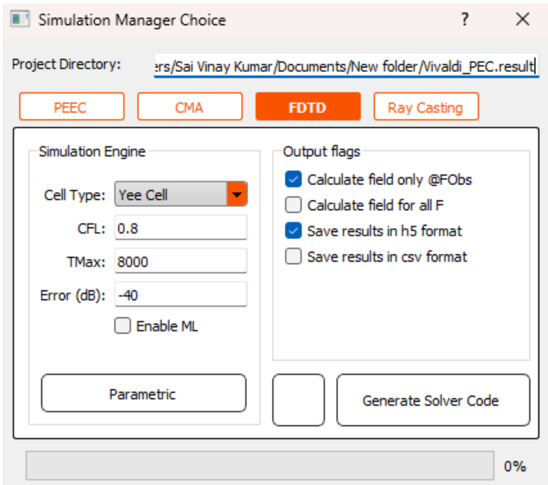


2. After selecting the grid of appropriate dimensions and divisions click on Run to get the following meshed structure.



Meshed View

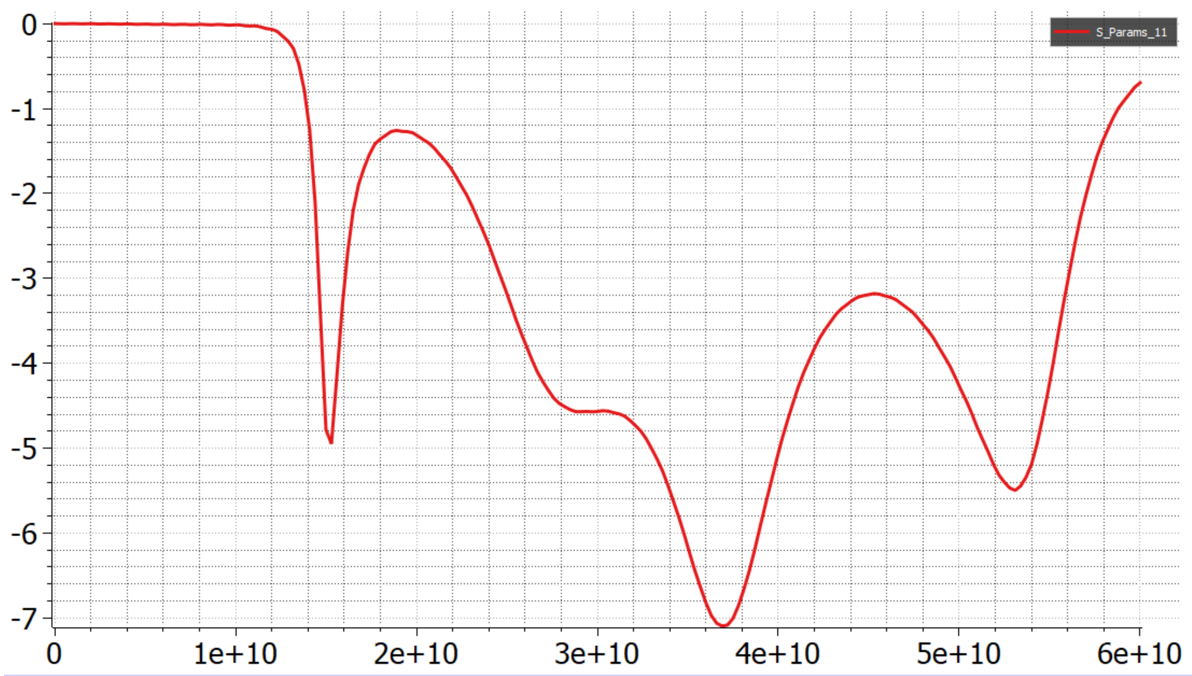
Running the Simulation



1. To run the simulation, click on the Run button  in the taskbar.
2. Select FDTD setup solver and assign the necessary time steps (Tmax). The other parameters can be kept as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.

Obtaining the Scattering parameters

1. To view the results of the simulation, click on the results operation in the taskbar.
2. Select the plots that require analysis and view them through the toolbar in the results tab.
3. Here are the S11 parameters obtained after simulation through the FDTD solver over 15000-time steps.

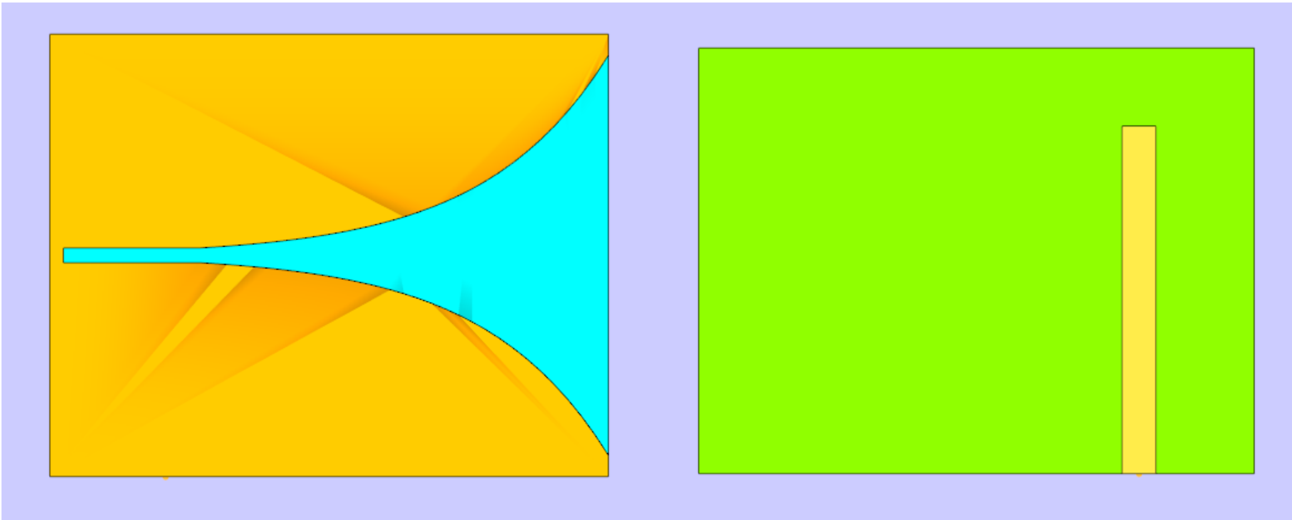


S11 results for fluid less Vivaldi antenna

Vivaldi antenna design with Sea Water

A Vivaldi antenna without fluid can be designed with these simple steps

1. Design the outer Vivaldi structure as shown previously.
2. Additionally design another polygon for the inner Vivaldi structure.
3. Design the substrate using “Box” tool in the toolbar with the appropriate dimensions.
4. Design the feed line on the opposite side of the substrate to the outer Vivaldi structure.
5. Design a rectangle at the end of the feedline in the XZ plane to assign port excitation.
6. Sweep the outer Vivaldi structure, Inner Vivaldi structure and the feedline for the necessary thickness.

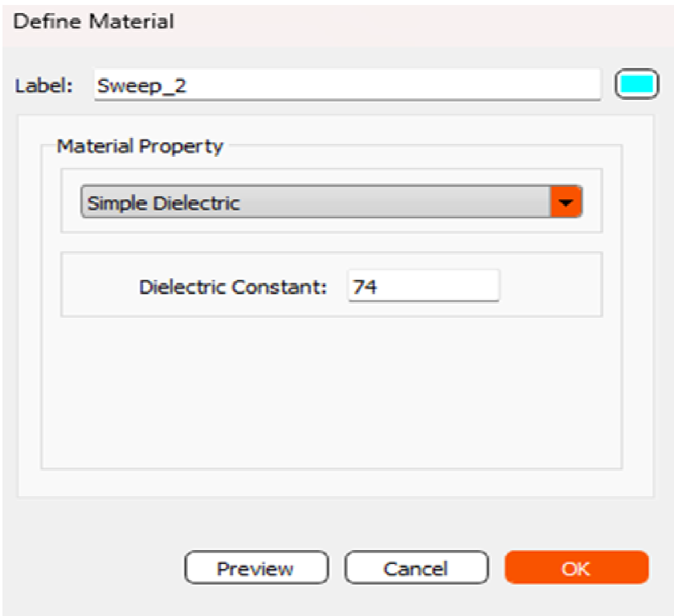


Front view and Rear View

Setting up the Excitation and assigning material properties

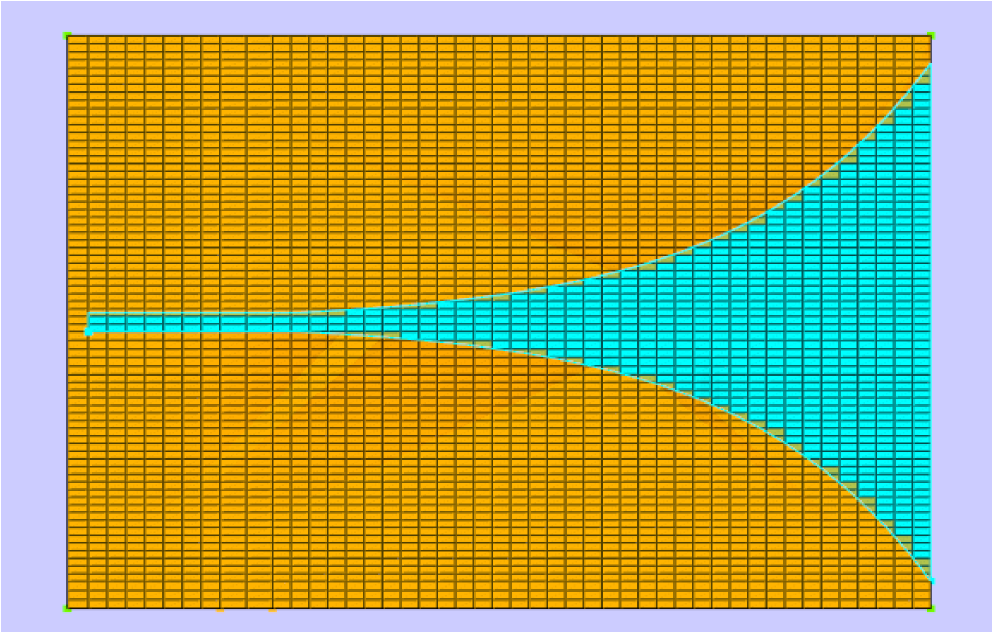
1. Assigning material properties and excitation is the same as for the Vivaldi antenna without fluid.
2. The only addition here is that the inner Vivaldi antenna must also be assigned the properties of the fluid through the same process

.



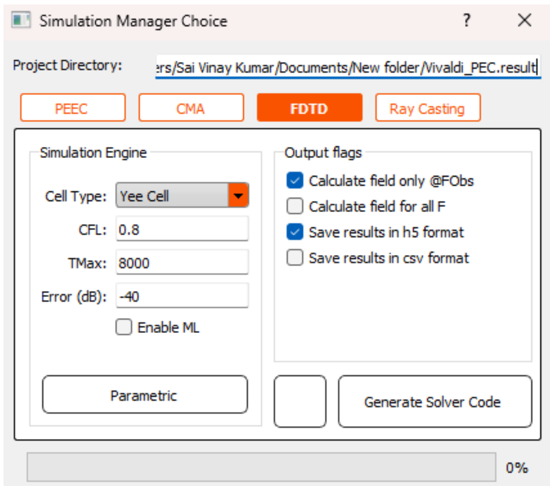
Meshing and Voxelization

The same procedure that was used for the Vivaldi antenna without fluid must be used here as well to yield the meshed structure shown below



Meshed View

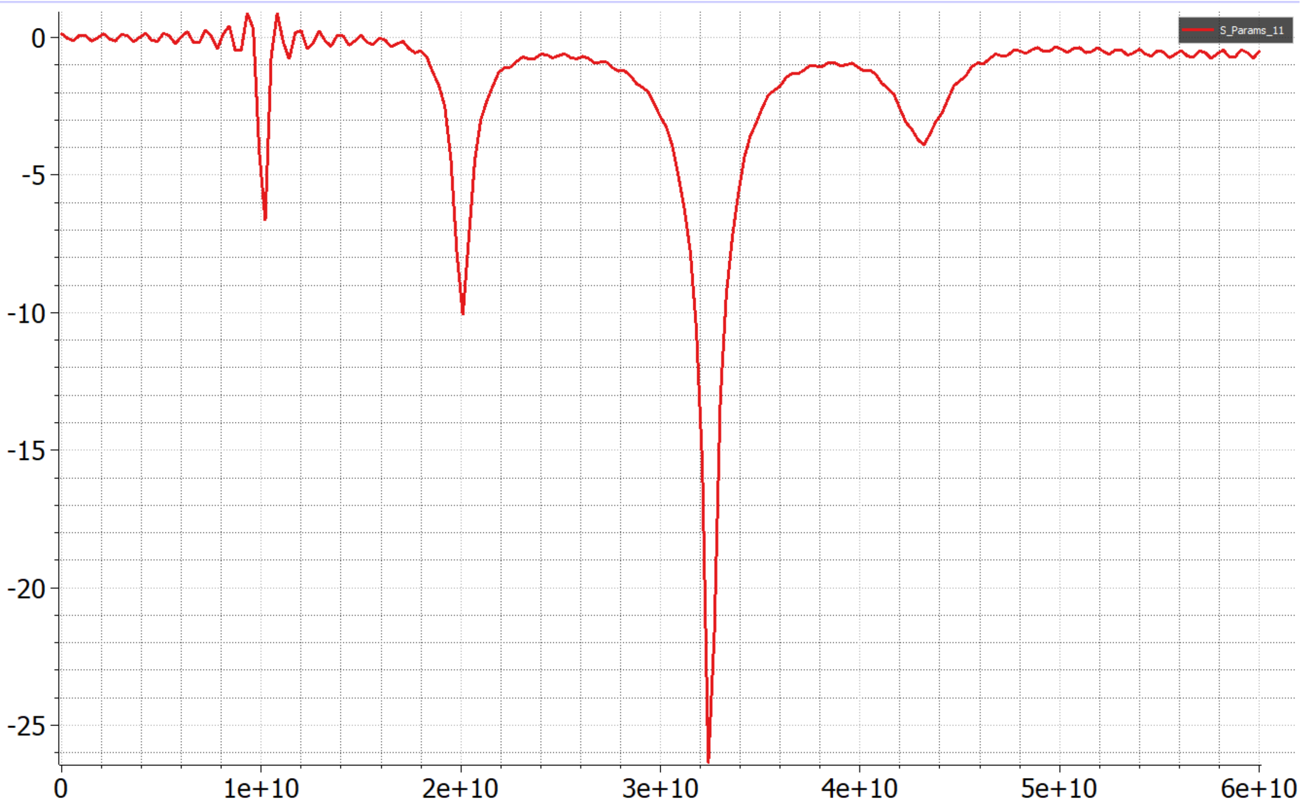
Running the Simulation



1. To run the simulation, click on the Run button  in the taskbar.
2. Select FDTD setup solver and assign the necessary time steps (Tmax). The other parameters can be kept as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.

Obtaining the Scattering parameters

1. To view the results of the simulation, click on the results operation in the taskbar.
2. Select the plots that require analysis and view them through the toolbar in the results tab.
3. Here are the S11 parameters obtained after simulation through the FDTD solver over 15000-time steps.



S11 results for fluidic Vivaldi antenna

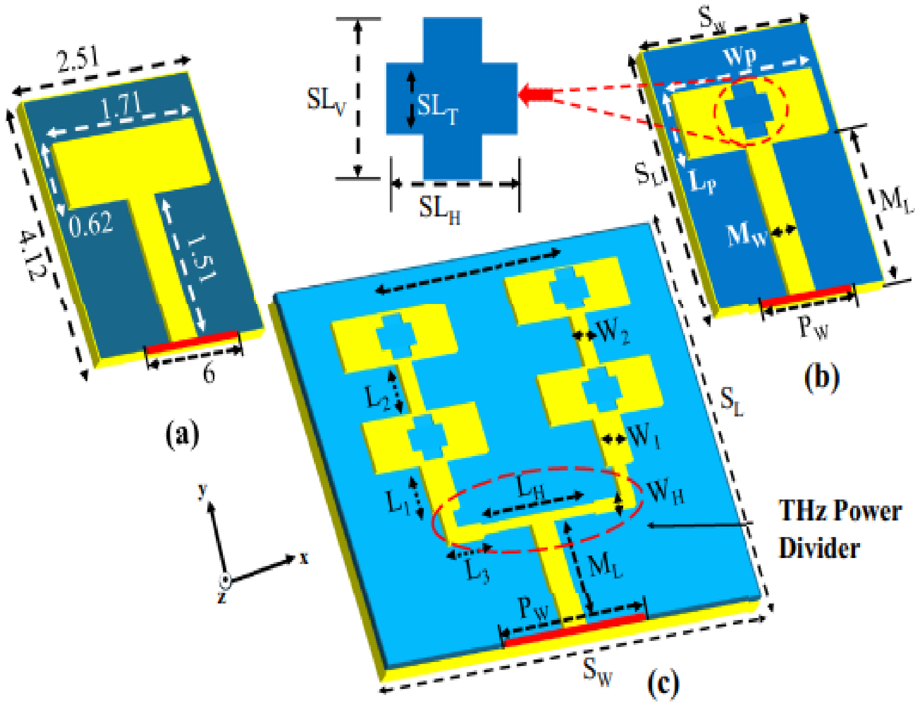
Comparison between Fluidic and Non Fluidic Vivaldi antennas

From the obtained S11 results, the difference that the introduction of sea water makes to the behaviour of a Vivaldi antenna can be observed. With the introduction of Sea water, the antenna radiates even better at 20GHz and 30GHz than its non-fluidic counterpart. The comparison can be further understood by studying the results of the fluidic Vivaldi antenna with a combination of other fluids like mercury and so on to see how each fluid impacts the antenna's behaviour differently.

11. Simulation and Analysis of a High-performance Sub-THz planar antenna array

The Terahertz (THz) spectral range, ranging from 0.1 to 3 THz, holds significant potential for future wireless technology. Recent study has focused on the terahertz gap because of its unique channel capabilities. The physical design and production of THz devices, particularly antennas, are significant impediments to realizing their full potential.

This example demonstrates how to design a high-performance sub-THz planar antenna array on TaraNG, to simulate and analyse its various parameters such as Reflection coefficient, Gain, Radiation pattern and so on. The antenna holds the potential to achieve terabits per second data rates and futuristic high-resolution short-range THz imaging application. The antenna is useful for THz sensing and imaging applications.



Sub THz planar antenna array

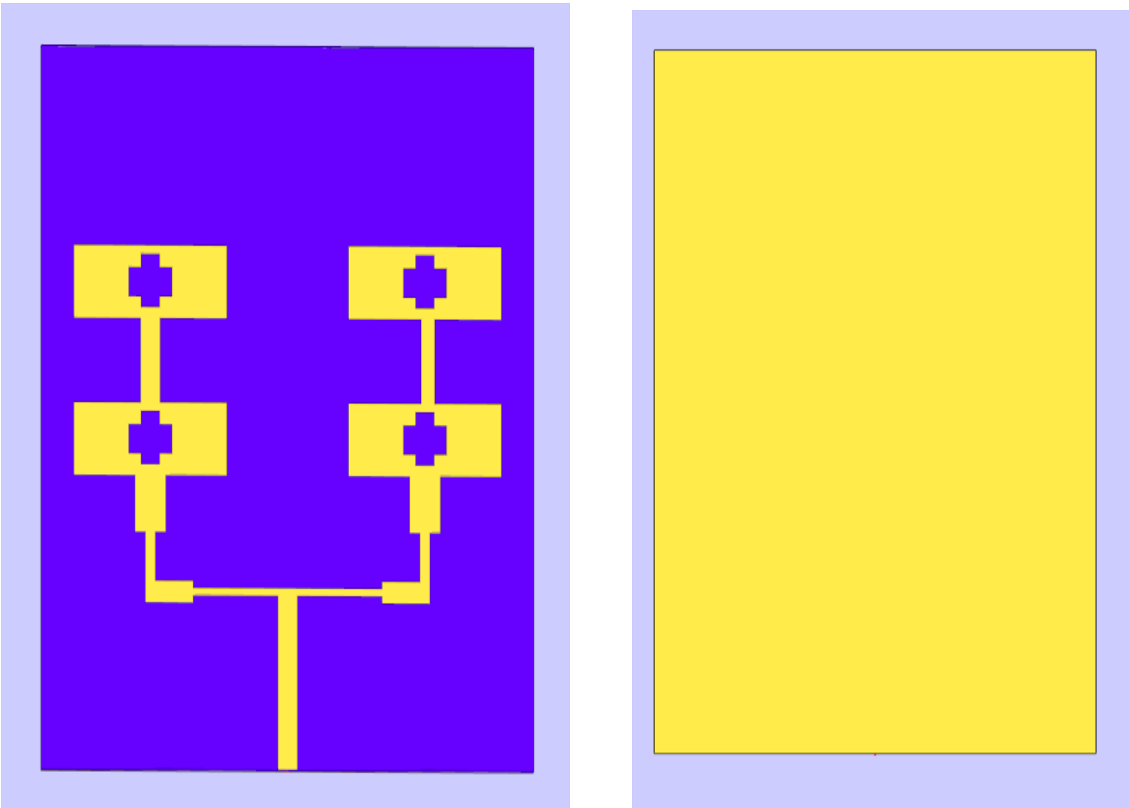
Importing Structures

Structures that are large or complicated to design can be imported from other CAD software and environments using the following procedure.

1. Save the designs exported from other CAD software as ". stl" files
2. Start Tarang and select the import icon in the CAD toolbar shown as 
3. Once a dialog box opens asking for the structures that are meant to be imported select the appropriate .stl files from their respective locations.

This specific structure has been imported to TaraNG

<u>The Sub THz planar antenna array structure:</u>

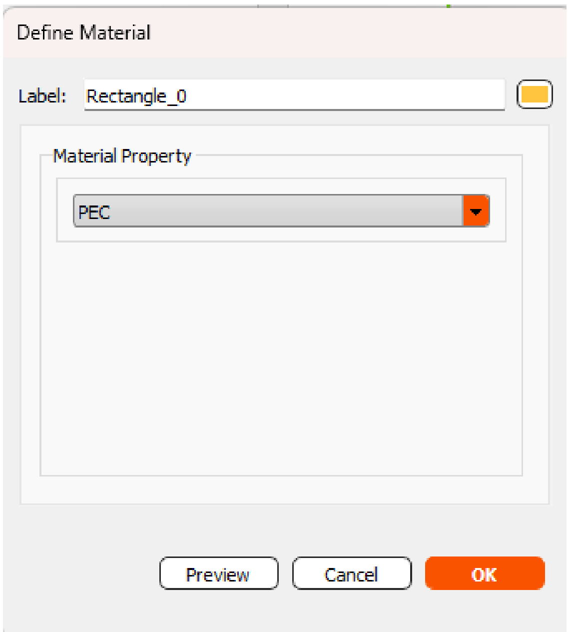


Sub THz planar antenna array design front and rear view respectively

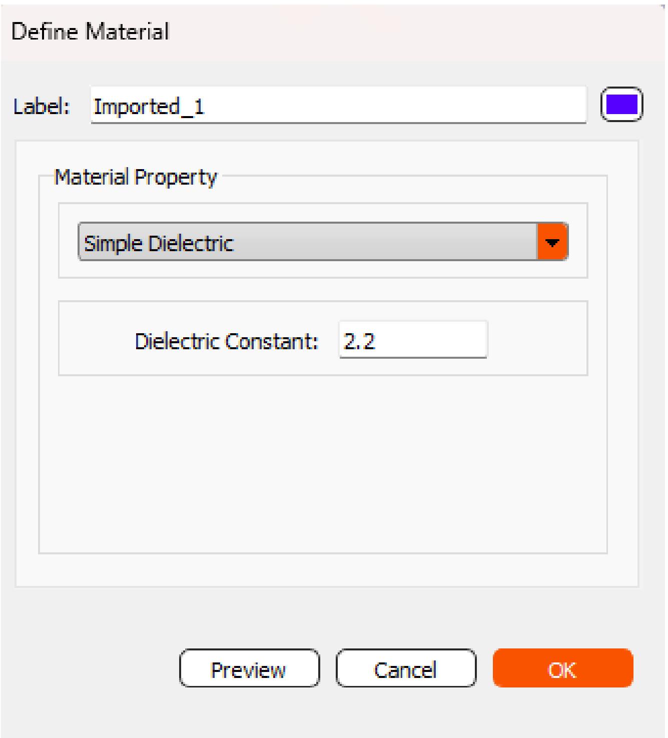
Setting up the Excitation and assigning material properties

To assign material properties and port excitation click on CEM operation in the TaraNG Taskbar.

1. Assign PEC material to the patch and the ground



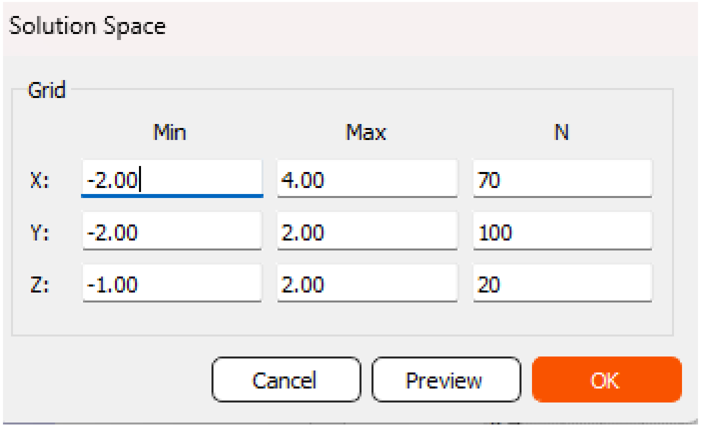
2. To add the feed, click on the 'Port'  icon and trace the arrow joining any two edges of the rectangle in the XZ plane alongside the feedline.
3. Assign Simple Dielectric property to the Substrate box with the necessary Effective permittivity value. In this instance it is 2.2



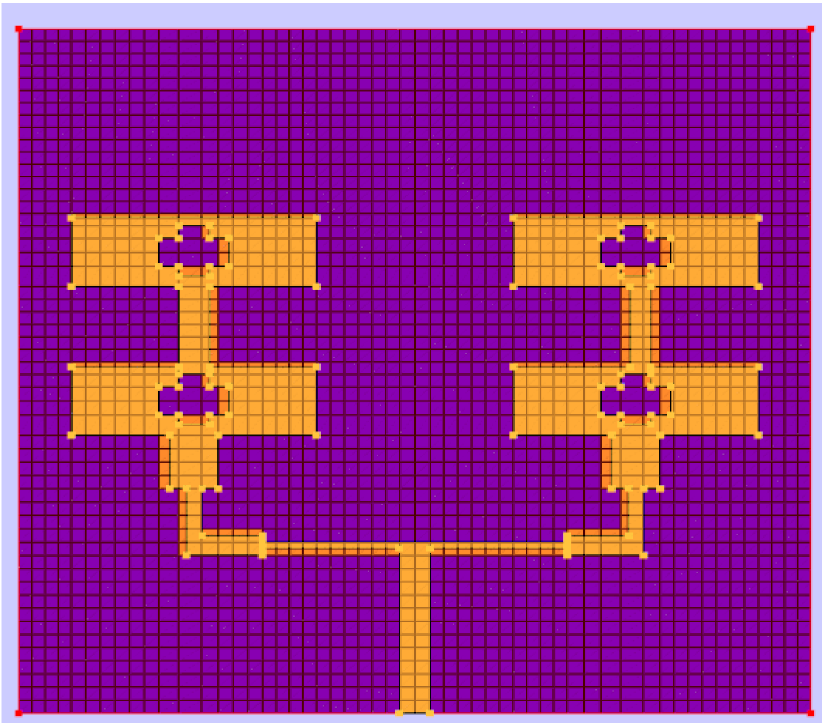
Meshing and Voxelization

To assign material properties and port excitation click on Mesh operation in the TaraNG Taskbar.

1. Click on the  icon in the tool bar and provide the appropriate dimensions for the meshing box along with the necessary number of divisions for the mesh cells

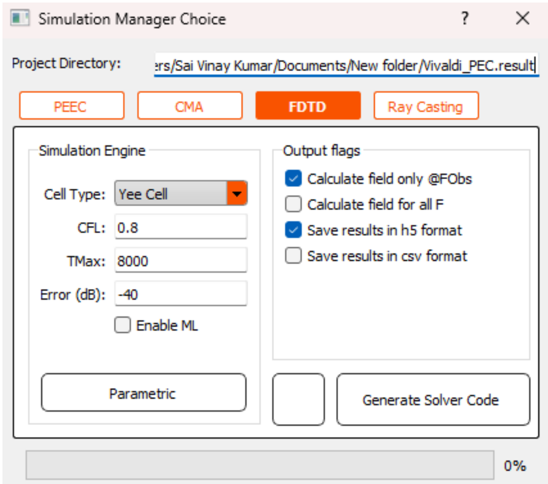


2. After selecting the grid of appropriate dimensions and divisions click on Run to get the following meshed structure.



Meshed View

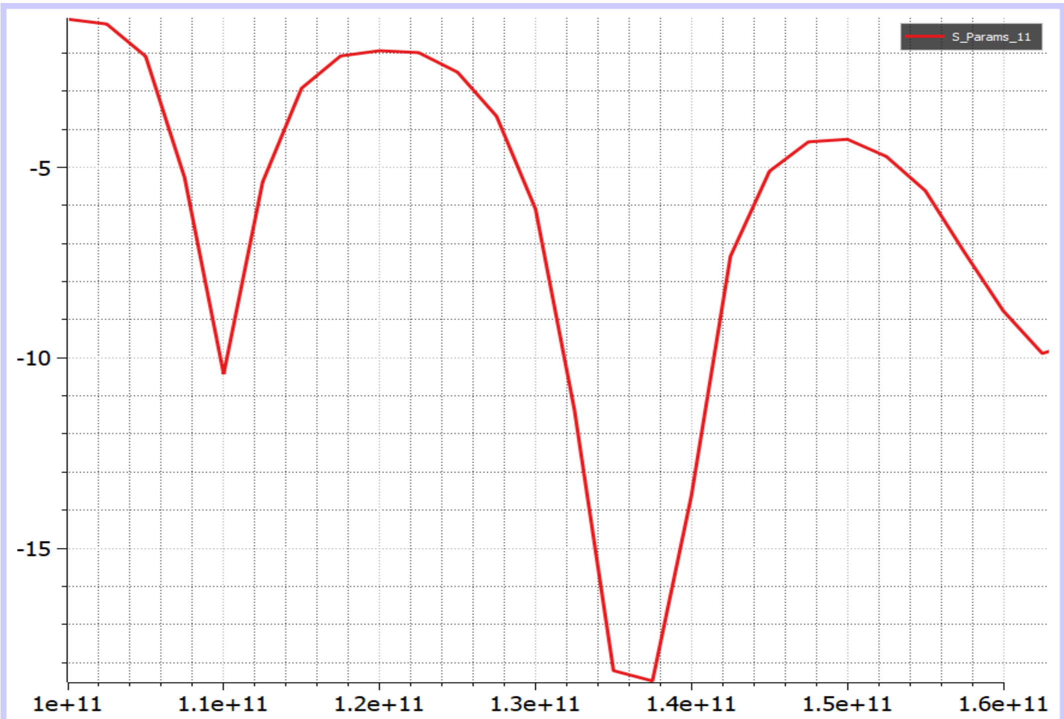
Running the Simulation



1. To run the simulation, click on the Run button  in the taskbar.
2. Select FDTD setup solver and assign the necessary time steps (Tmax). The other parameters can be kept as default.
3. Further click on the simulate button. The software solves the problem in a few seconds the progress bar will reach to 100% once the simulation is complete.

Obtaining the Scattering parameters

1. To view the results of the simulation, click on the results operation in the taskbar.
2. Select the plots that require analysis and view them through the toolbar in the results tab.
3. Here are the S11 parameters obtained after simulation through the FDTD solver over 15000-time steps.



S11 results for Sub THz planar antenna array

Result Analysis

From the obtained results, it can be observed that the Sub THz planar antenna array performs well between 130 and 150 GHz. Thus, the antenna array can be used for terahertz frequencies and terahertz applications.