Antenna Toolbox™

Getting Started Guide

MATLAB®

R2023a

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Antenna Toolbox™ Getting Started Guide

Revision History

March 2015	Online only	New for Version 1.0 (R2015a)
September 2015	Online only	Revised for Version 1.1 (R2015b)
March 2016	Online only	Revised for Version 2.0 (R2016a)
September 2016	Online only	Revised for Version 2.1 (R2016b)
March 2017	Online only	Revised for Version 2.2 (R2017a)
September 2017	Online only	Revised for Version 3.0 (R2017b)
March 2018	Online only	Revised for Version 3.1 (R2018a)
September 2018	Online only	Revised for Version 3.2 (R2018b)
March 2019	Online only	Revised for Version 4.0 (R2019a)
September 2019	Online only	Revised for Version 4.1 (R2019b)
March 2020	Online only	Revised for Version 4.2 (R2020a)
September 2020	Online only	Revised for Version 4.3 (R2020b)
March 2021	Online only	Revised for Version 5.0 (R2021a)
September 2021	Online only	Revised for Version 5.1 (R2021b)
March 2022	Online only	Revised for Version 5.2 (R2022a)
September 2022	Online only	Revised for Version 5.3 (R2022b)
March 2023	Online only	Revised for Version 5.4 (R2023a)

Contents

1 Introduction to Antenna Toolbox

2 Introduction to Arrays

3 Introduction to RF Propagation

1

Introduction to Antenna Toolbox

Antenna Toolbox Product Description

Design, analyze, and visualize antenna elements and antenna arrays

Antenna Toolbox provides functions and apps for the design, analysis, and visualization of antenna elements and arrays. You can design standalone antennas and build arrays of antennas using predefined elements with parameterized geometry, arbitrary planar structures, or custom 3D structures described with STL files.

Antenna Toolbox uses electromagnetic solvers, including the method of moments (MoM), to compute impedance, current distribution, efficiency, and near-field and far-field radiation patterns. To improve the antenna design, you can use manual methods or use the optimization methods provided in the toolbox. Antenna geometry and analysis results can be visualized in 2D and 3D. The toolbox lets you integrate antenna array patterns into wireless systems for simulating beamforming and beam steering algorithms. The impedance analysis results can be used to design matching networks for integration with the RF front-end. You can install the antennas on large platforms such as vehicles or aircraft and analyze the effects of the structure on antenna performance. You can import STL and Gerber files to analyze a pre-existing structure or export them to share or manufacture your design. A site viewer enables you to visualize antenna coverage on a 3D terrain map using a variety of propagation models, including ray tracing.

Antenna Modeling and Analysis

This example shows how to construct, visualize and analyze the antenna elements in the Antenna Toolbox.

Define Antenna Element Using the Antenna Library

Define a helix antenna using the `helix` antenna element in the Antenna Modeling and Analysis library.

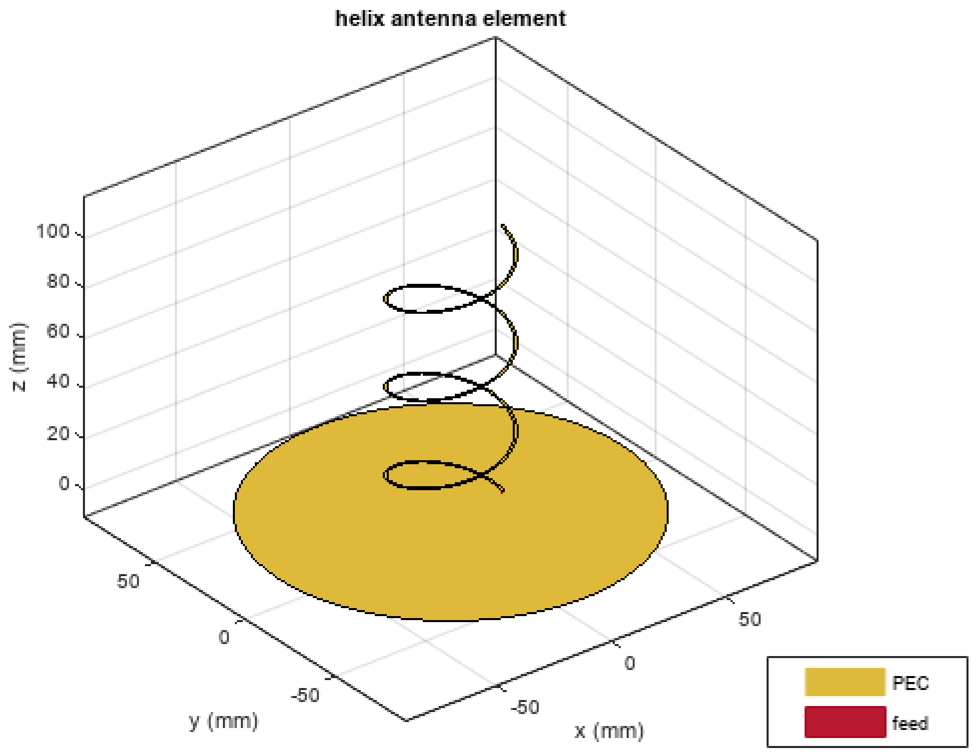
```
hx = helix
```

```
hx =
  helix with properties:

               Radius: 0.0220
                Width: 1.0000e-03
                Turns: 3
              Spacing: 0.0350
     WindingDirection: 'CCW'
       FeedStubHeight: 1.0000e-03
    GroundPlaneRadius: 0.0750
            Substrate: [1x1 dielectric]
            Conductor: [1x1 metal]
                 Tilt: 0
             TiltAxis: [1 0 0]
                 Load: [1x1 lumpedElement]
```

Show Structure of Antenna

Use the `show` function to view the structure of the helix antenna. A helical antenna consists of a helical shaped conductor on a ground plane. The ground plane of the antenna is in the X-Y plane.

```
show(hx)
```



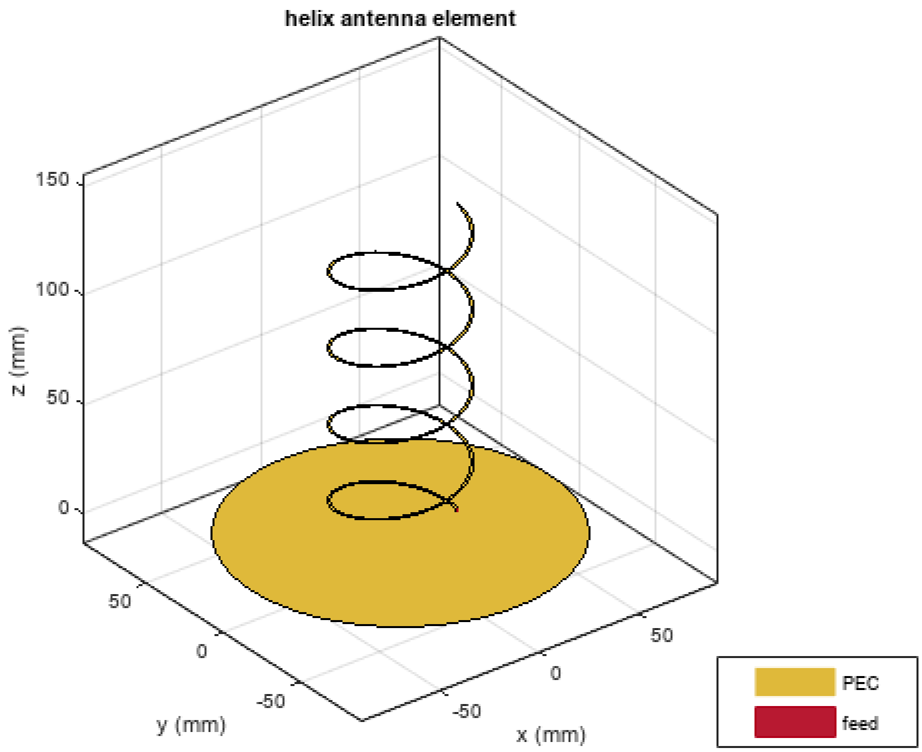
Modify Properties of Antenna

Modify the following properties of the helix antenna: Radius = 28e-3, Width = 1.2e-3, Number of Turns = 4 Display the properties of the antenna. View the antenna to see the change in structure.

```
hx = helix('Radius',28e-3,'Width',1.2e-3,'Turns',4)
```

```
hx = 
  helix with properties:

               Radius: 0.0280
                Width: 0.0012
                Turns: 4
              Spacing: 0.0350
     WindingDirection: 'CCW'
       FeedStubHeight: 1.0000e-03
    GroundPlaneRadius: 0.0750
            Substrate: [1x1 dielectric]
            Conductor: [1x1 metal]
                 Tilt: 0
             TiltAxis: [1 0 0]
                 Load: [1x1 lumpedElement]
```

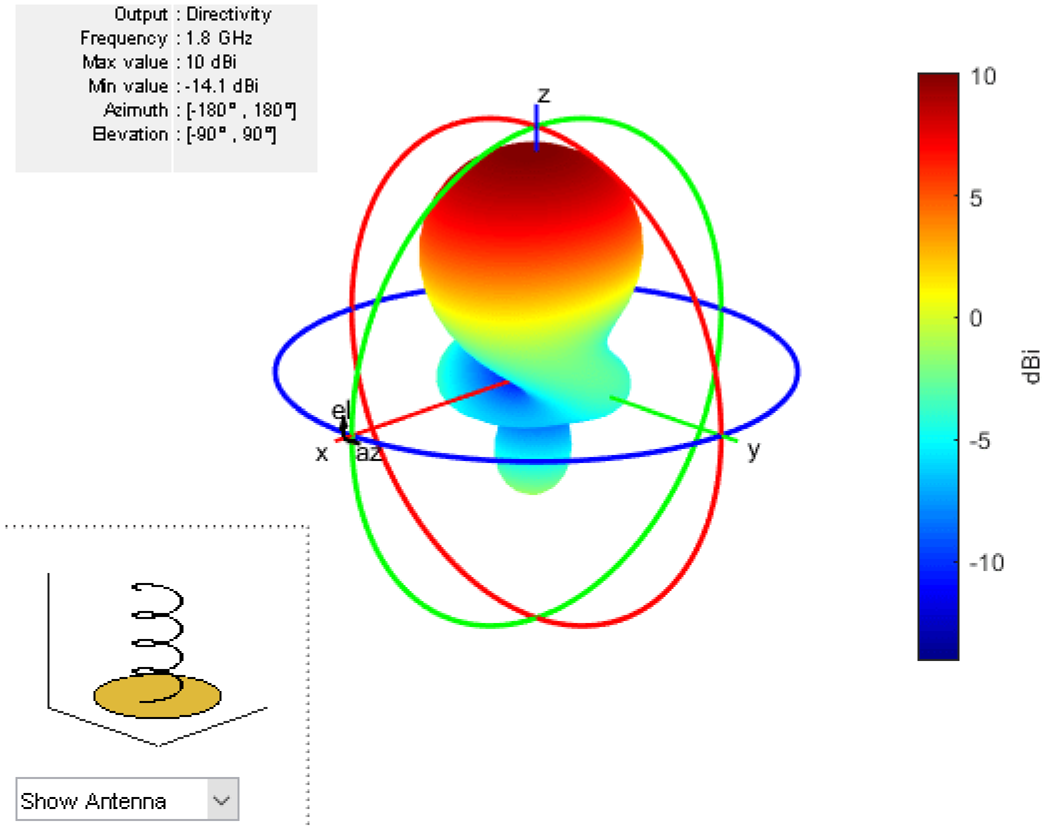
```
show(hx)
```



Plot Radiation Pattern of Antenna

Use `pattern` function to plot the radiation pattern of the helix antenna. The radiation pattern of an antenna is the spatial distribution of power of an antenna. The pattern displays the directivity or gain of the antenna. By default, the pattern function plots the directivity of the antenna.

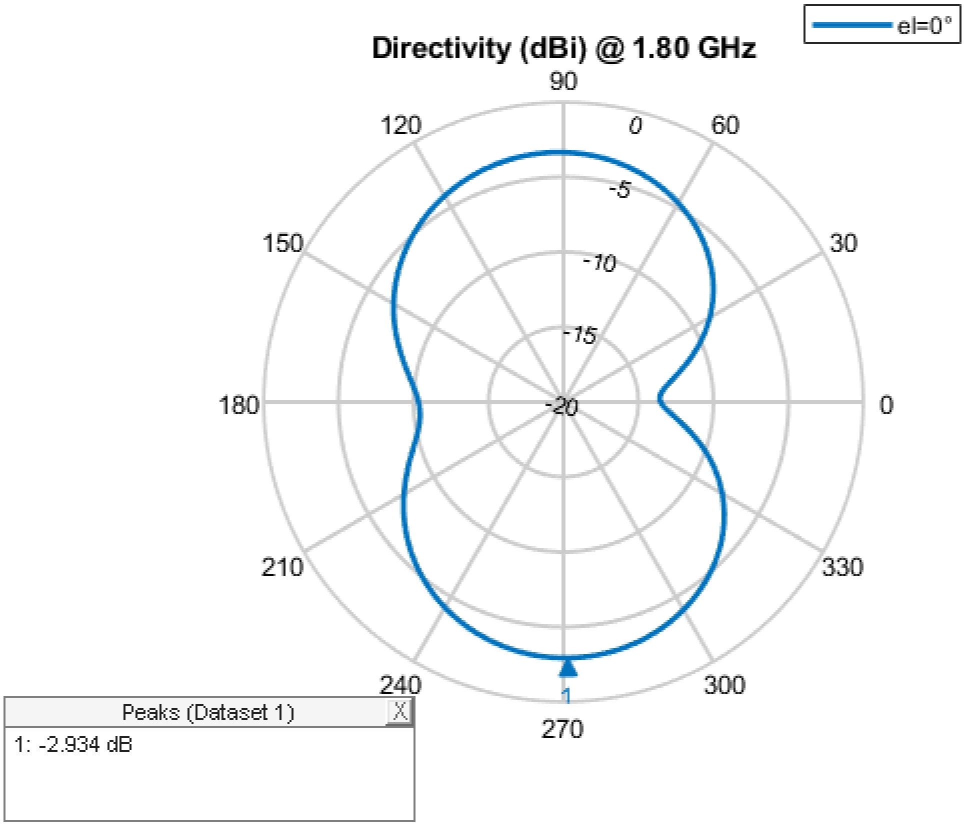
```
pattern(hx,1.8e9)
```



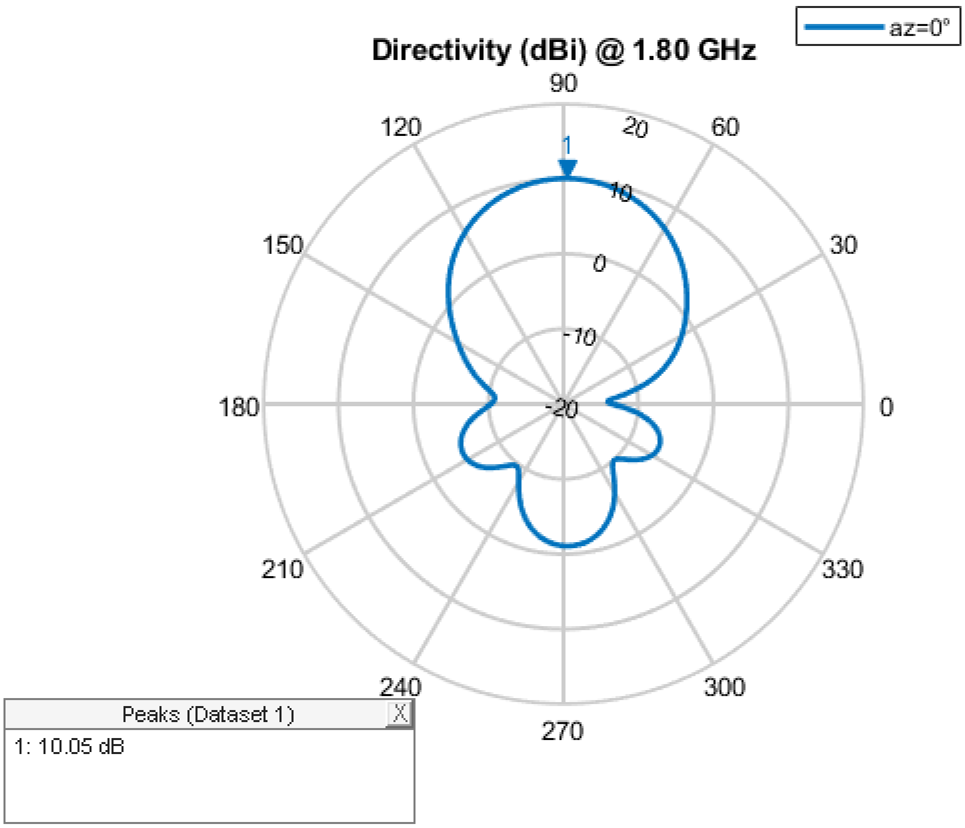
Plot Azimuth and Elevation Pattern of Antenna

Use `patternAzimuth` and `patternElevation` functions to plot the azimuth and elevation pattern of the helix antenna. This is the 2D radiation pattern of the antenna at a specified frequency.

```
patternAzimuth(hx,1.8e9)
```




```
figure
patternElevation(hx,1.8e9)
```



Calculate Directivity of Antenna

Use Directivity name-value pair in the output of the pattern function to calculate the directivity of helix antenna. Directivity is the ability of an antenna to radiate power in a particular direction. It can be defined as ratio of maximum radiation intensity in the desired direction to the average radiation intensity in all other directions. Note that the antenna Gain and Directivity are measured at a distance of `100*lambda`.

```
Directivity = pattern(hx,1.8e9,0,90)
```

```
Directivity = 10.0444
```

Calculate EHfields of Antenna

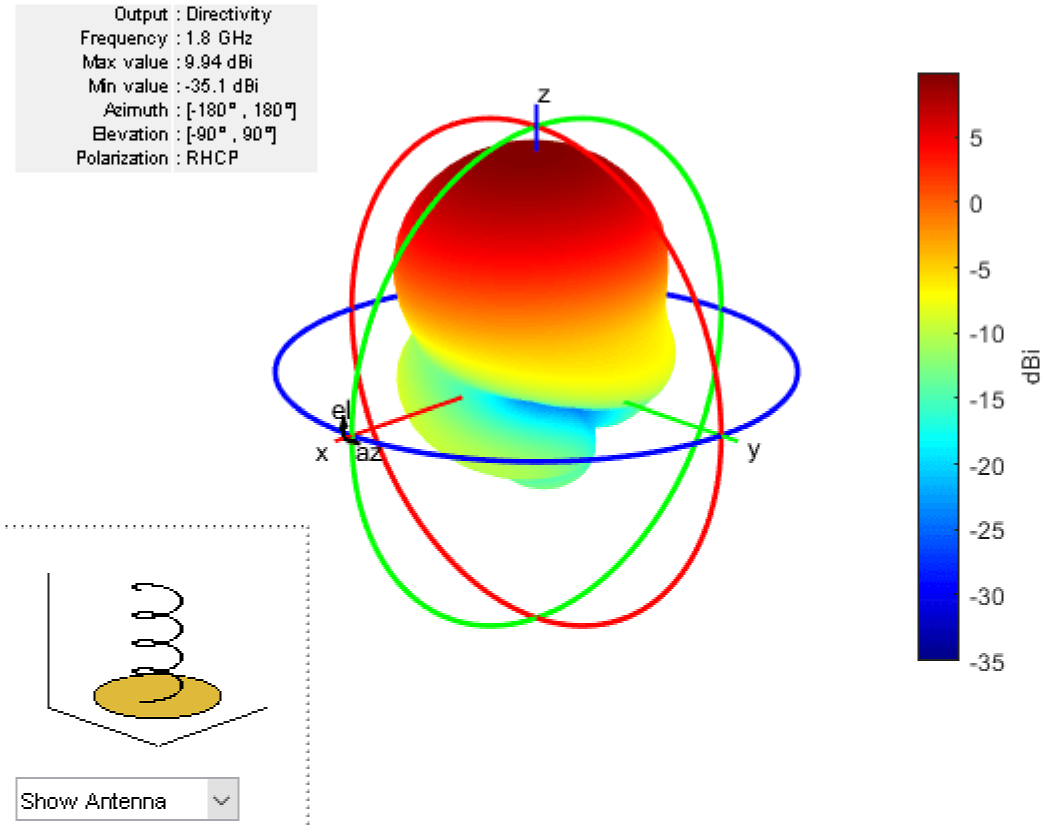
Use the `EHfields` function to calculate the EH fields of the helix antenna. EH fields are the x, y, z components of electric and magnetic fields of an antenna. These components are measured at a specific frequency and at specified points in space.

```
[E,H] = EHfields(hx,1.8e9,[0;0;1]);
```

Plot Different Polarizations of Antenna

Use the Polarization name-value pair in the pattern function to plot the different polarization patterns of the helix antenna. Polarization is the orientation of the electric field, or E-field, of an antenna. Polarization is classified as elliptical, linear, or circular. This example shows the Right-Hand Circularly Polarized(RHCP) radiation pattern of the helix.

```
pattern(hx,1.8e9,'Polarization','RHCP')
```



Calculate Axial Ratio of Antenna

Use the `axialRatio` function to calculate the axial ratio of the helix antenna. Antenna axial ratio (AR) in a given direction quantifies the ratio of two orthogonal field components radiated in a circularly polarized wave. An axial ratio of infinity, implies a linearly polarized wave. The unit of measure is dB.

```
ar = axialRatio(hx,1.8e9,20,30)
```

```
ar = 24.4335
```

Calculate Beamwidth of Antenna

Use the `beamwidth` function to calculate the beamwidth of the antenna. Antenna beamwidth is the angular measure of the antenna pattern coverage. Beamwidth angle is measured in plane containing the direction of main lobe of the antenna.

```
[bw, angles] = beamwidth(hx,1.8e9,0,1:1:360)
```

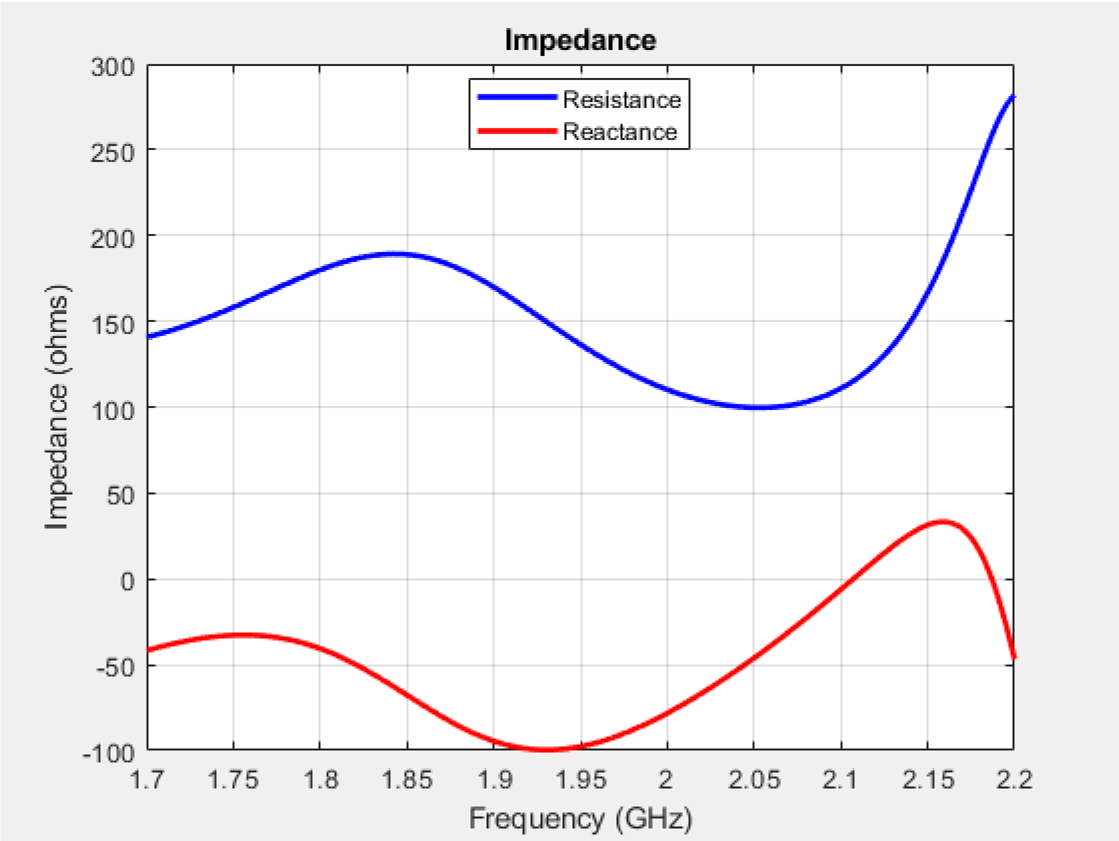
```
bw = 57.0000
```

```
angles = 1×2

    60   117
```

Calculate Impedance of Antenna

Use the `impedance` function to calculate and plot the input impedance of helix antenna. Input impedance is a ratio of voltage and current at the port. Antenna impedance is calculated as the ratio of the phasor voltage (which is 1V at a phase angle of 0 deg as mentioned earlier) and the phasor current at the port.

```
impedance(hx,1.7e9:1e6:2.2e9)
```



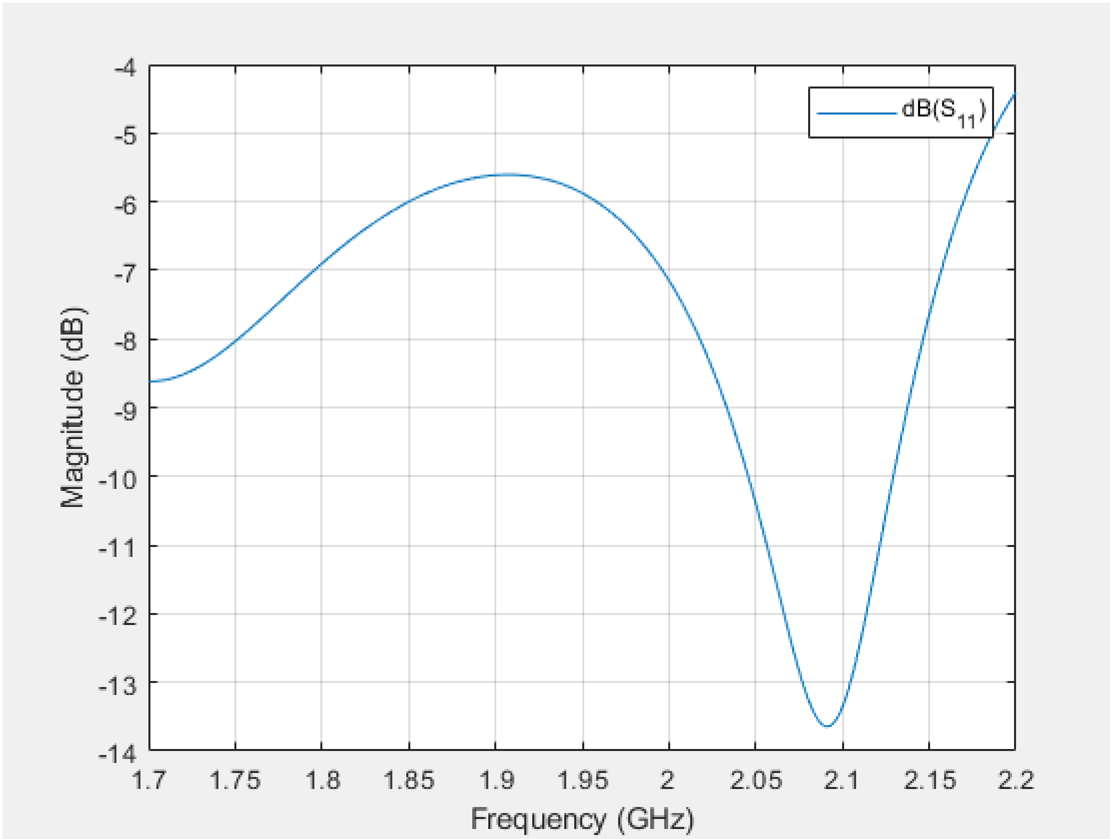
Calculate Reflection Coefficient of Antenna

Use the `sparameters` function to calculate the S11 of the helix antenna. Antenna reflection coefficient, or S_1_1, describes a relative fraction of the incident RF power that is reflected back due to the impedance mismatch.

```
S = sparameters(hx,1.7e9:1e6:2.2e9,72)
```

```
S = 
  sparameters: S-parameters object

       NumPorts: 1
    Frequencies: [501x1 double]
     Parameters: [1x1x501 double]
      Impedance: 72

  rfparam(obj,i,j) returns S-parameter Sij
```

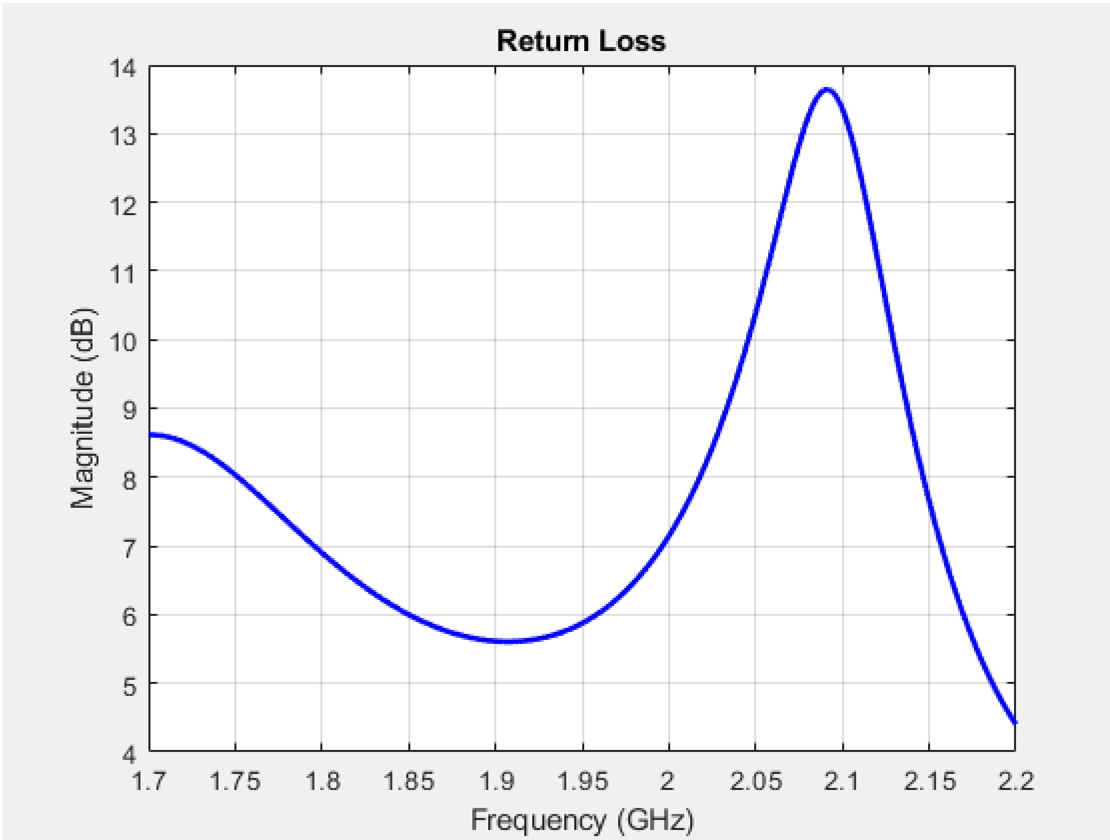
```
rfplot(S)
```



Calculate Return Loss of Antenna

Use the `returnLoss` function to calculate and plot the return loss of the helix antenna. Antenna return loss is a measure of the effectiveness of power delivery from a transmission line to a load such as antenna. The calculations are displayed in logscale.

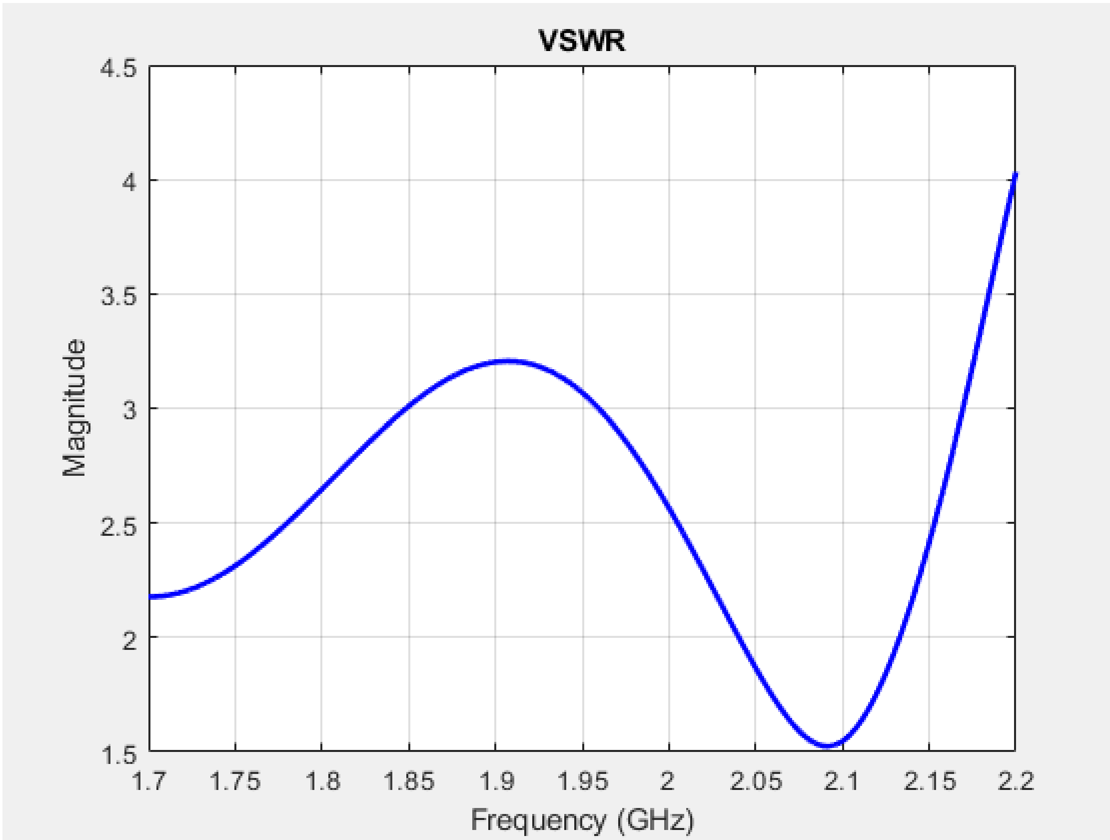
```
returnLoss(hx,1.7e9:1e6:2.2e9,72)
```



Calculate Voltage Standing Wave Ratio (VSWR) of Antenna

Use the `vswr` function to calculate and plot the VSWR of the helix antenna. The antenna VSWR is another measure of impedance matching between transmission line and antenna.

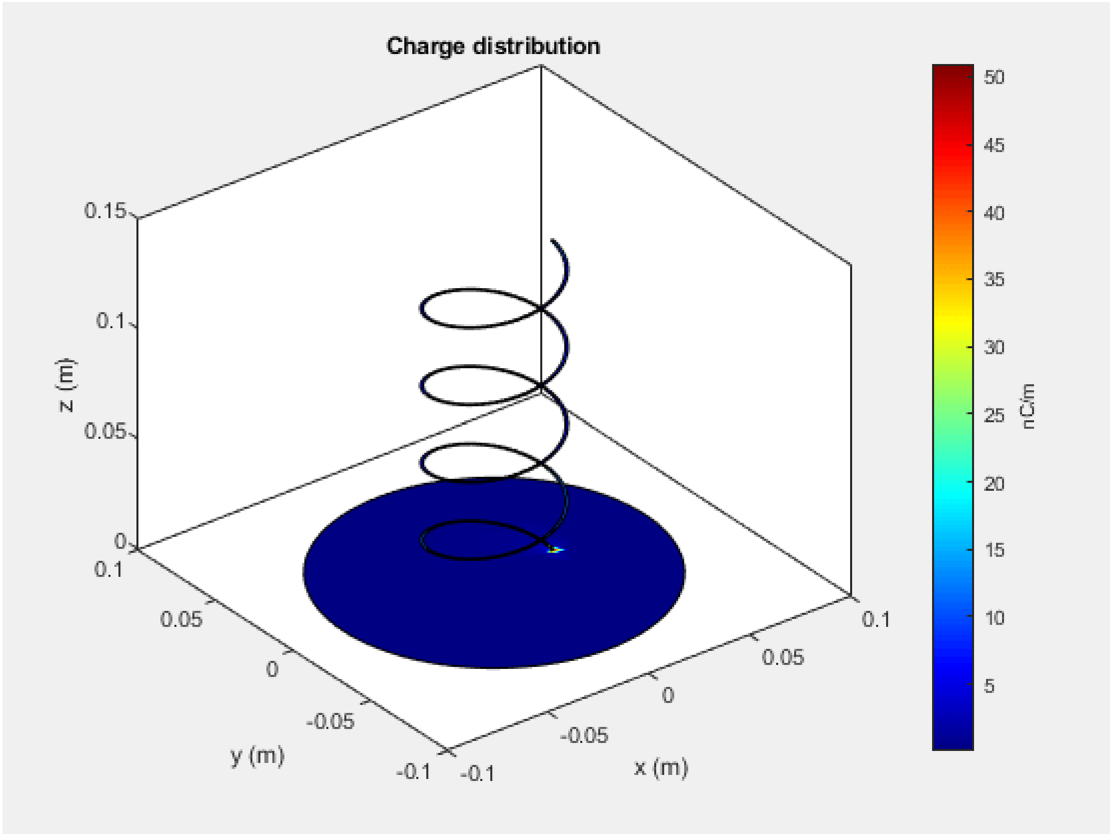
```
vswr(hx,1.7e9:1e6:2.2e9,72)
```



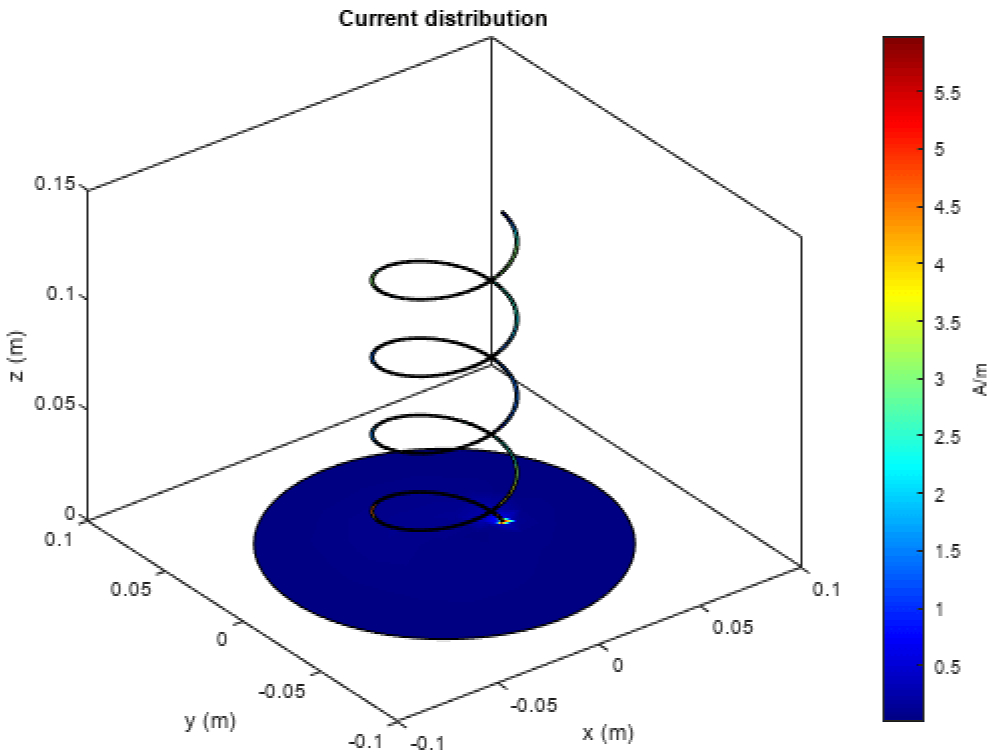
Calculate Current and Charge Distribution of Antenna

Use the `charge` function to calculate the charge distribution of the helix antenna. Charge distribution is the value of charge on the antenna surface at a specified frequency. Use the `current` function to calculate the current distribution of the helix antenna. Current distribution is the value of current on the antenna surface at a specified frequency.

```
charge(hx,2.01e9)
```



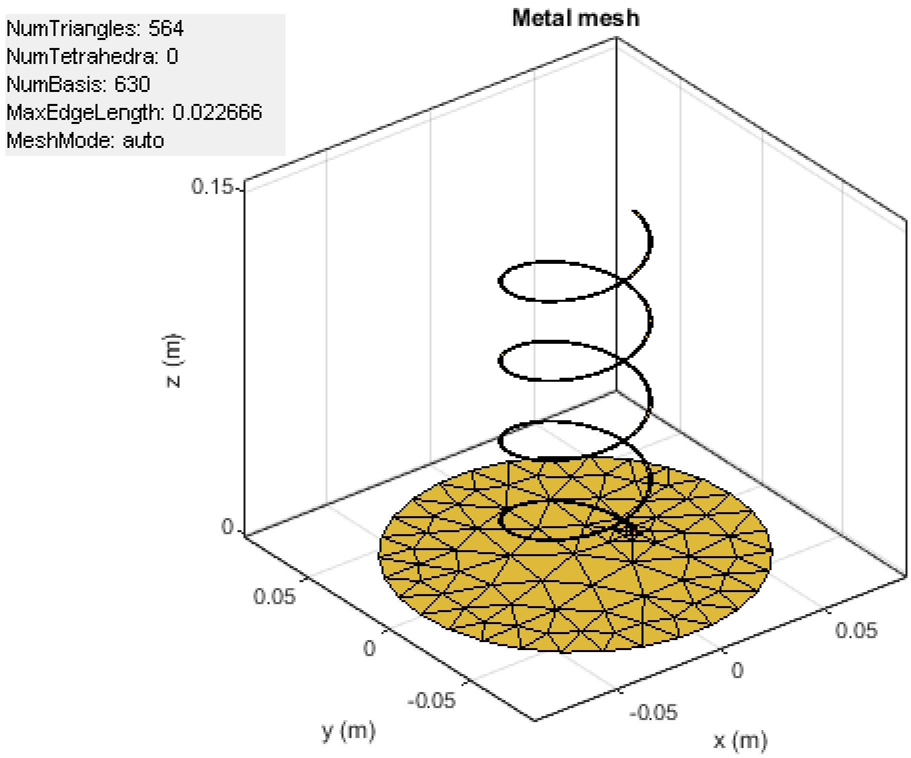

```
figure
current(hx,2.01e9)
```



Show Mesh of Antenna

Use the `mesh` function to create and show a mesh structure of the helix antenna. mesh is used to discretize antenna surface. In this process, the electromagnetic solver can process the geometry and material of the antenna. The shape of the basis or the discretizing element for subdividing the antenna surface is a triangle.

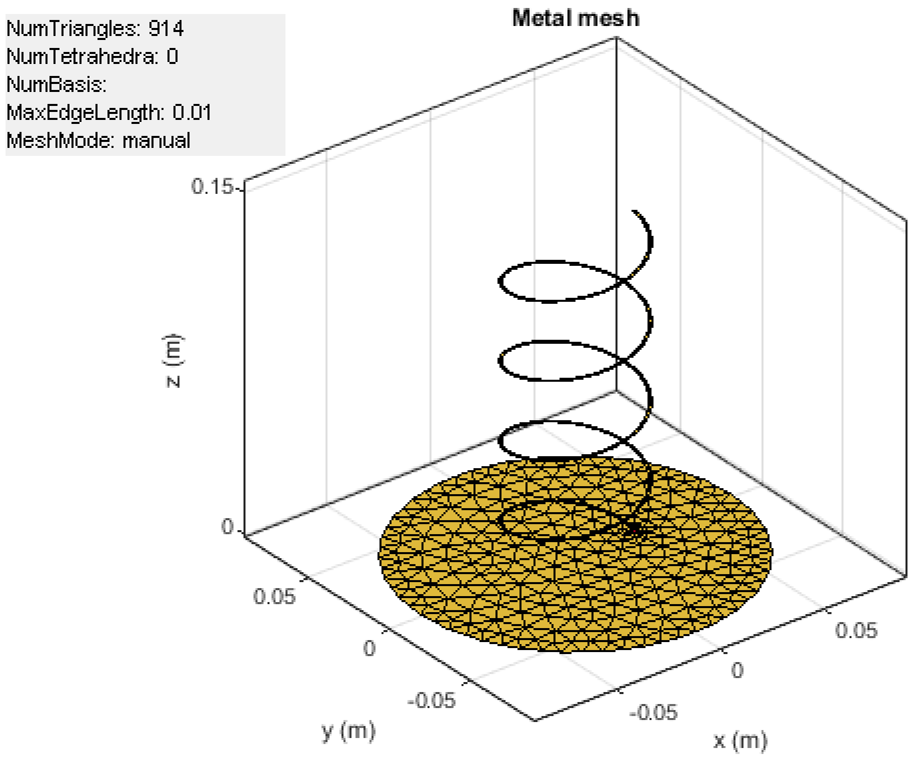
```
figure
mesh(hx)
```



Mesh Antenna Manually

Specify the maximum edge length for the triangles using the 'MaxEdgeLength' name-value pair. This name-value pair meshes the helix structure manually.

```
figure
mesh(hx,'MaxEdgeLength',0.01)
```



Change Meshing to Automatic

```
meshconfig(hx,'auto')
```

```
ans = struct with fields:
     NumTriangles: 914
    NumTetrahedra: 0
         NumBasis: []
    MaxEdgeLength: 0.0100
    MinEdgeLength: 0.0075
       GrowthRate: 0.9500
         MeshMode: 'auto'
```

See Also:

"Antenna Near-Field Visualization"

"Array Modeling and Analysis" on page 2-2

References

[1] Balanis, C.A. "Antenna Theory. Analysis and Design", p. 514, Wiley, New York, 3rd Edition, 2005.

Antenna Classification

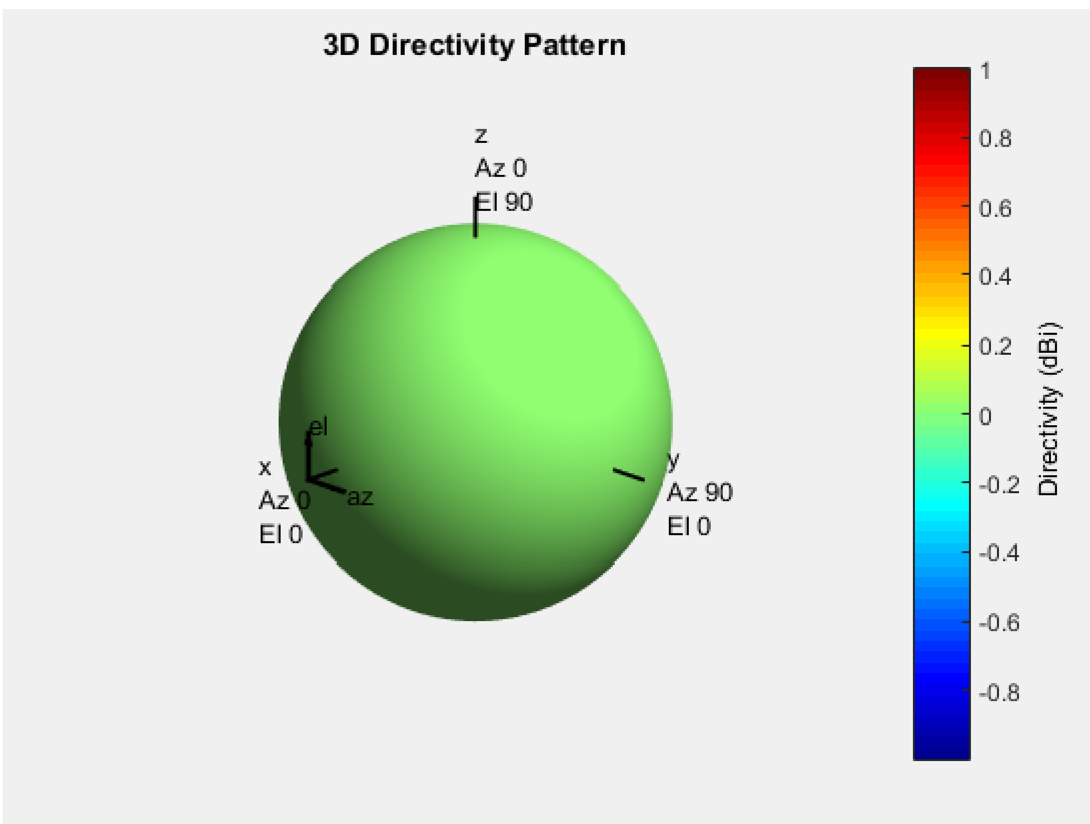
In this section...

Antennas are classified based on the radiation pattern or the feeding mechanism. Antenna radiation pattern is the angular variation of signal strength around the antenna. Feeding mechanism defines how the signal is fed into the antenna and the location of the feed point on the antenna.

Radiation Pattern

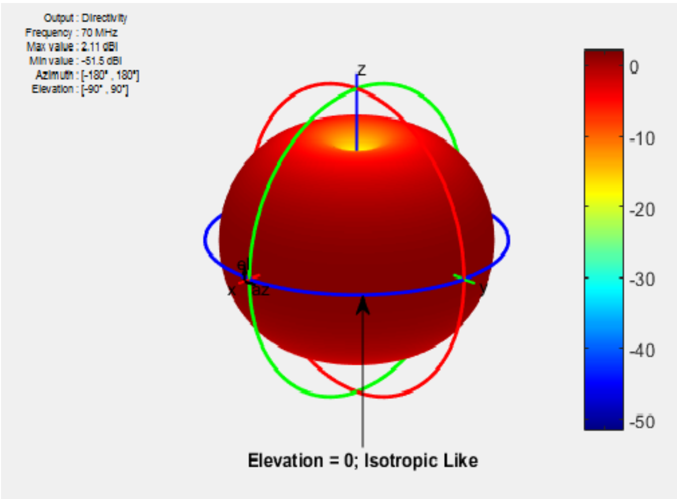
Isotropic Antenna

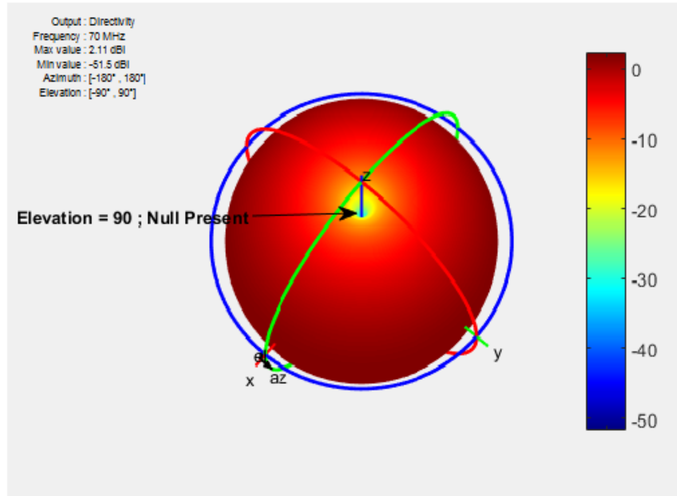
An *isotropic* antenna is an ideal lossless antenna that radiates uniformly in all directions. The antenna has no spatial selectivity or nulls. Practical antennas are compared against the isotropic antenna, but they rarely behaves like one.



Omnidirectional Antenna

Omnidirectional antennas behave like isotropic antennas in one plane. These antennas have nulls in the orthogonal plane. A common example of an omnidirectional antenna is the `dipole` antenna.

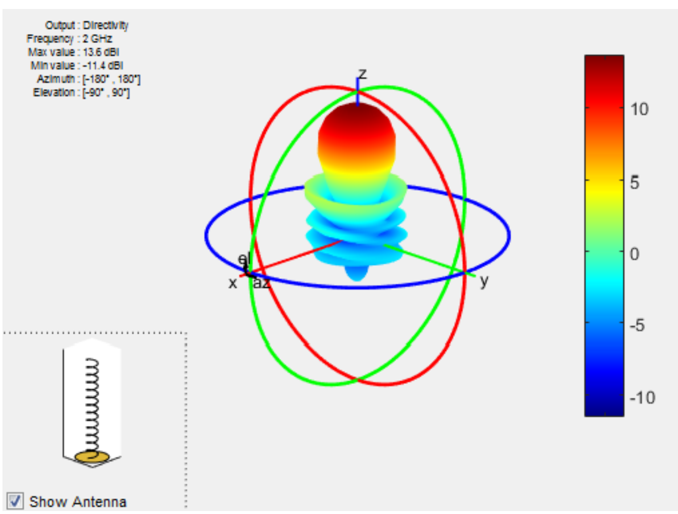


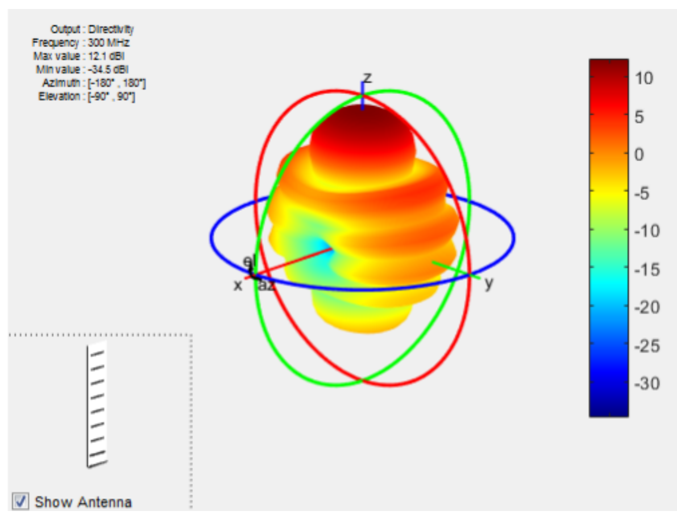


The dipole is omnidirectional around the E-plane, or elevation angle. The null is present in the H-plane, or azimuth angle.

Directional Antennas

Directional antennas are highly directive in a given direction. These antennas show high spatial selectivity, narrow bandwidth. They also have well defined major, or main, beam in the desired directions. Common examples of directional antennas are `helix` and `yagiUda`.

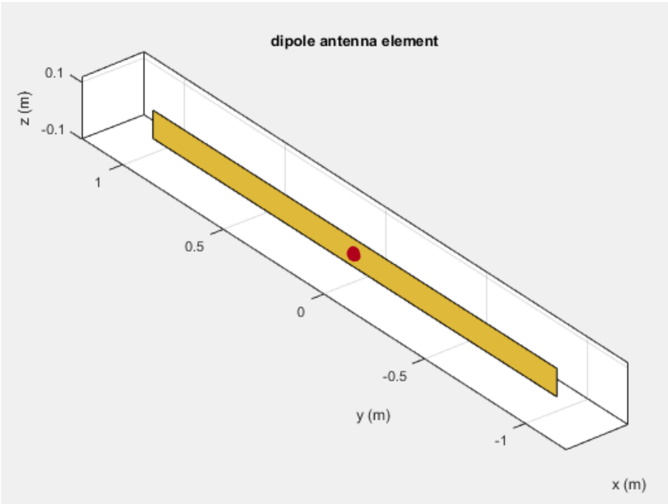


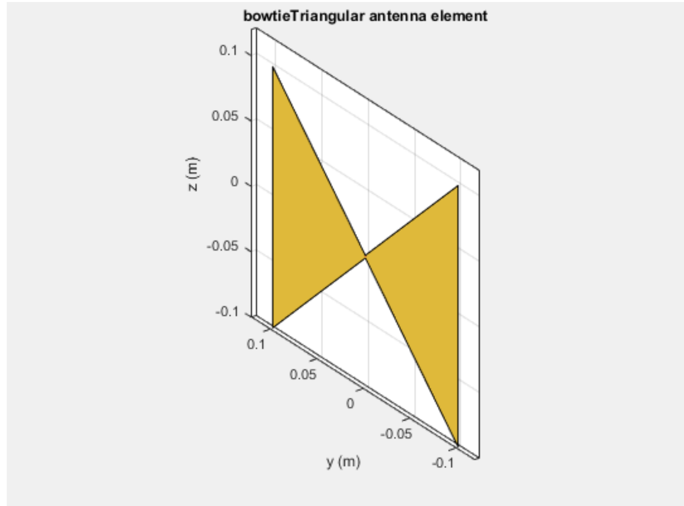


Antenna Feeding Mechanism

Balanced Antennas

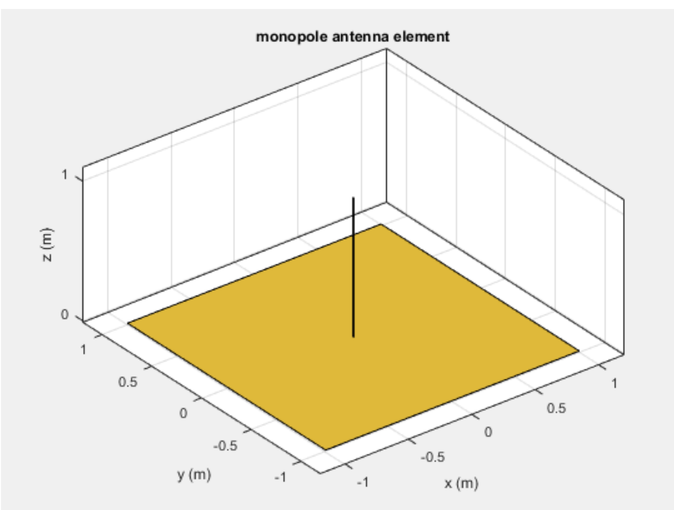
In *balanced* antennas, one side of the antenna is a mirror image of the other. These antennas require a balun to feed it, using a coaxial line. Common examples are: dipoles, bowties, spirals, and loops.

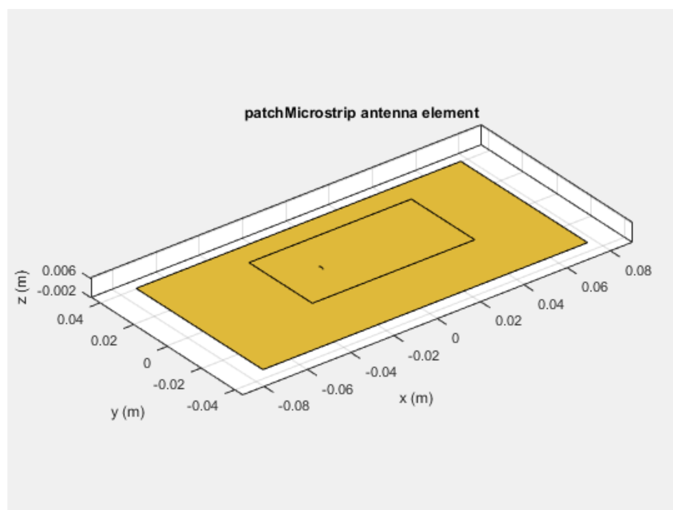




Unbalanced Antennas

Unbalanced antennas are end fed and mounted on top of a ground plane. The coaxial shield is connected to the ground, and the center conductor is connected to the antenna element. Common examples are monopoles and patches.





References

[1] Balanis, C.A. *Antenna Theory: Analysis and Design*. 3rd Ed. New York: Wiley, 2005.

Antenna Toolbox Coordinate System

In this section...

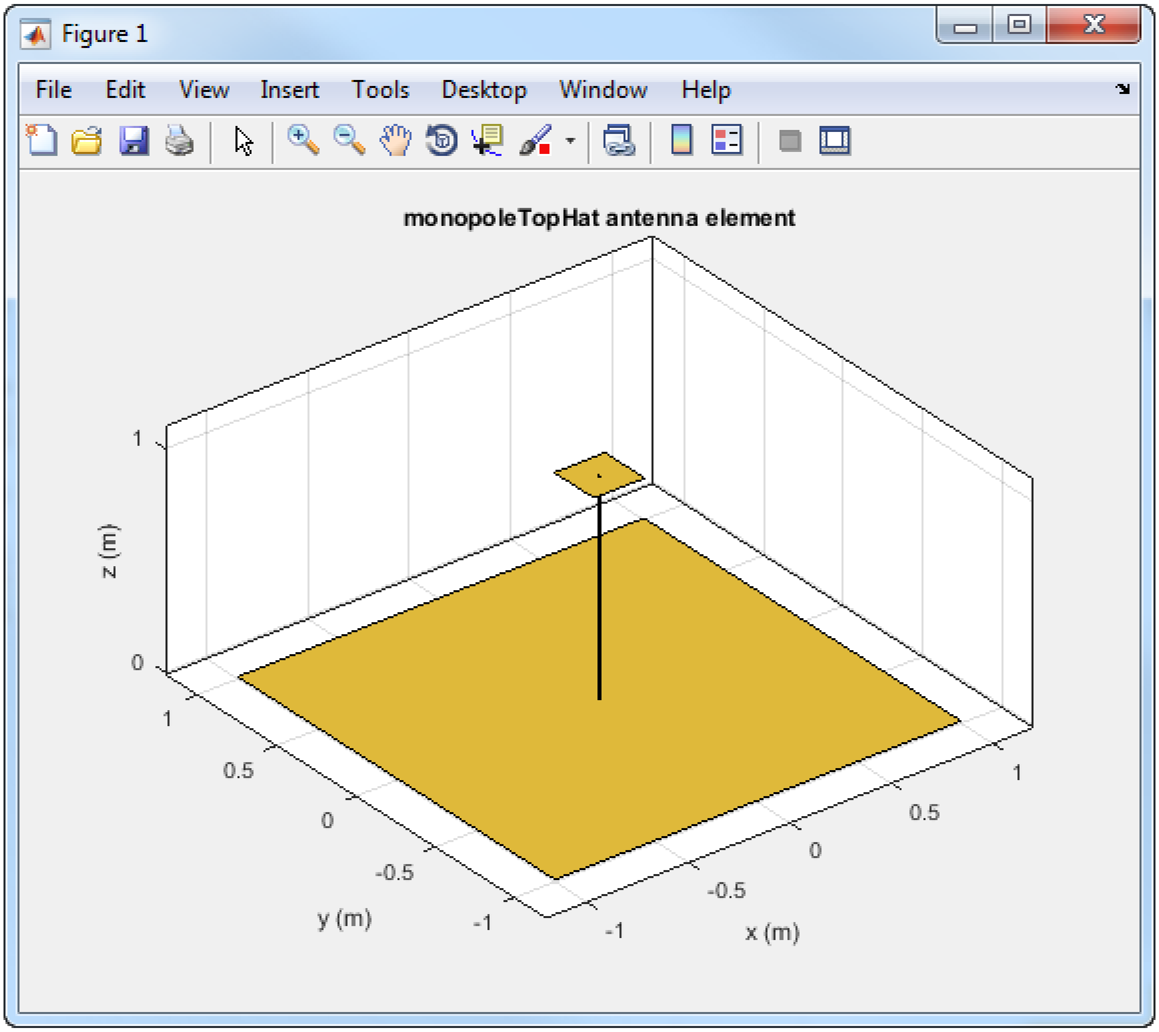
Antenna Toolbox uses two types of coordinate system: *rectangular coordinate system* and *spherical coordinate system* .

Antenna Toolbox uses the *rectangular coordinate system* to visualize antenna or array geometry. The toolbox uses the *spherical coordinate system* to visualize antenna radiation patterns.

Rectangular Coordinate System

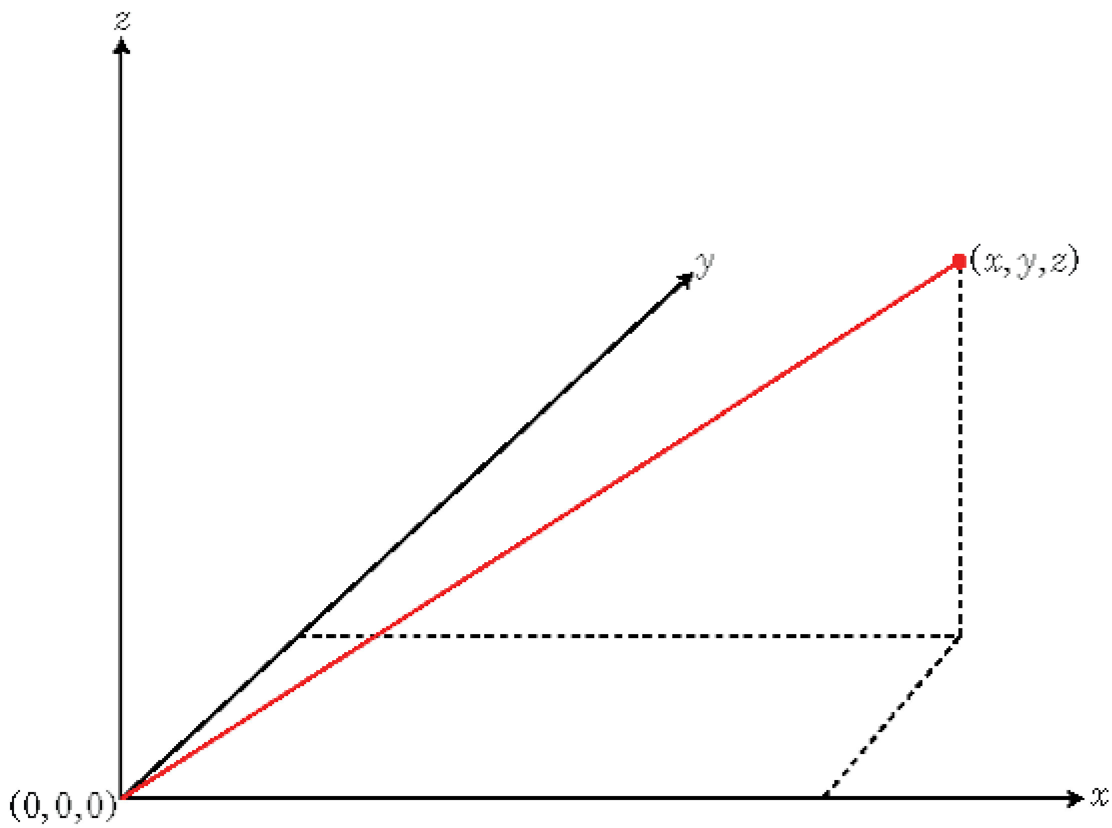
Visualize the geometry of a default `monopoleTopHat` antenna from the antenna library.

```
m = monopoleTopHat;
show(m);
```



The toolbox displays the top-hat monopole antenna in the *rectangular* or *Cartesian* coordinate system.

The *rectangular* coordinate system also called *Cartesian* coordinate system specifies a position in space as an ordered 3-tuple of real numbers, `(x,y,z)`, with respect to the origin `(0,0,0)`.



You can view the 3-tuple as a point in space, or equivalently as a vector in three-dimensional Euclidean space. When viewed as a vector in space, the coordinate axes are basis vectors and the vector gives the direction to a point in space from the origin. Every vector in space is uniquely determined by a linear combination of the basis vectors. The most common set of basis vectors for three-dimensional Euclidean space are the standard unit basis vectors:

$$\{[1\ 0\ 0], [0\ 1\ 0], [0\ 0\ 1]\}$$

.

Orthogonal Basis and Euclidean Norm

Any three linearly independent vectors define a basis for three-dimensional space. However, the Antenna Toolbox assumes that the basis vectors you use are orthogonal.

The standard distance measure in space is the l^2 norm, or Euclidean norm. The Euclidean norm of a vector [*x y z*] is defined by:

$$\sqrt{x^2 + y^2 + z^2}$$

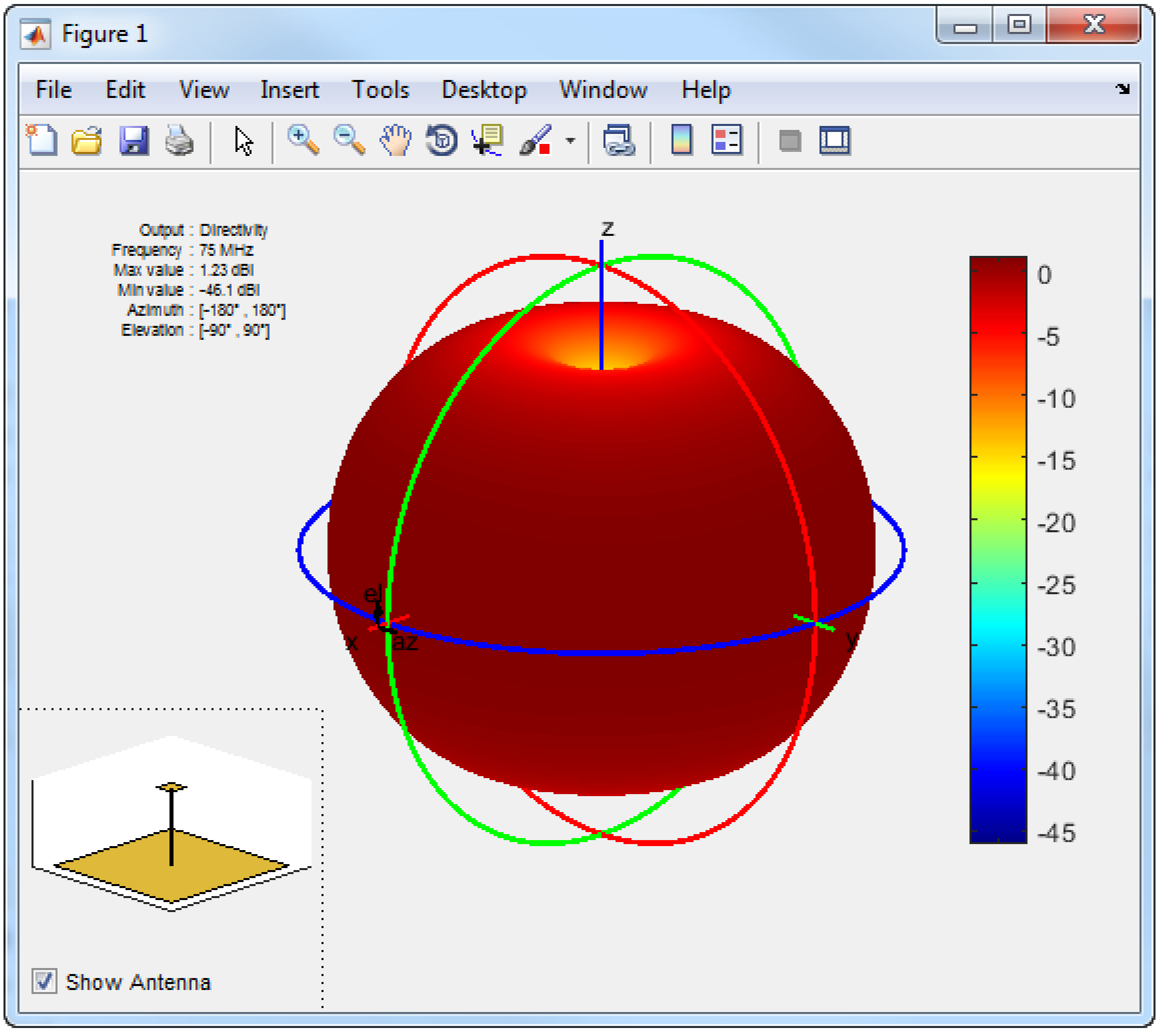
The Euclidean norm gives the length of the vector measured from the origin as the hypotenuse of a right triangle. The distance between two vectors [*x0 y0 z0*] and [*x1 y1 z1*] is:

$$\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2}$$

Spherical Coordinate System

Visualize the radiation pattern of the default `monopoleTopHat` antenna.

```
m = monopoleTopHat;
pattern(m,75e6);
```



The toolbox displays the radiation pattern of the top-hat monopole using *spherical* coordinate system represented by azimuth and elevation angles.

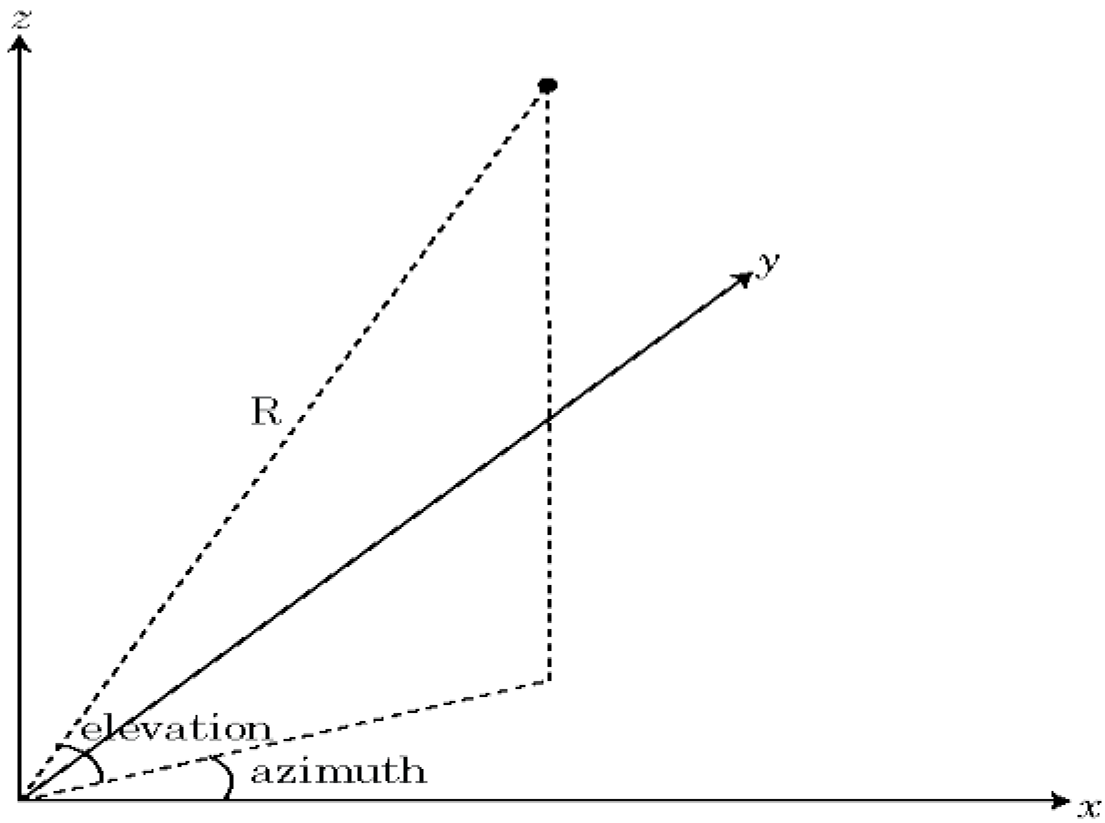
The *spherical* coordinate system defines a vector or point in space with a distance *R* and two angles. You can represent the angles in this coordinate system:

- Azimuth and elevation angles
- Phi (Φ) and theta (θ) angles
- *u* and *v* coordinates

Azimuth and Elevation Angles

The *azimuth angle* is the angle from the positive *x*-axis to the vector's orthogonal projection onto the *xy* plane, moving in the direction towards the y-axis. The azimuth angle is in the range –180 and 180 degrees.

The *elevation angle* is the angle from the vector's orthogonal projection on the *xy* plane toward the positive *z*-axis, to the vector. The elevation angle is in the –90 and 90 degrees.

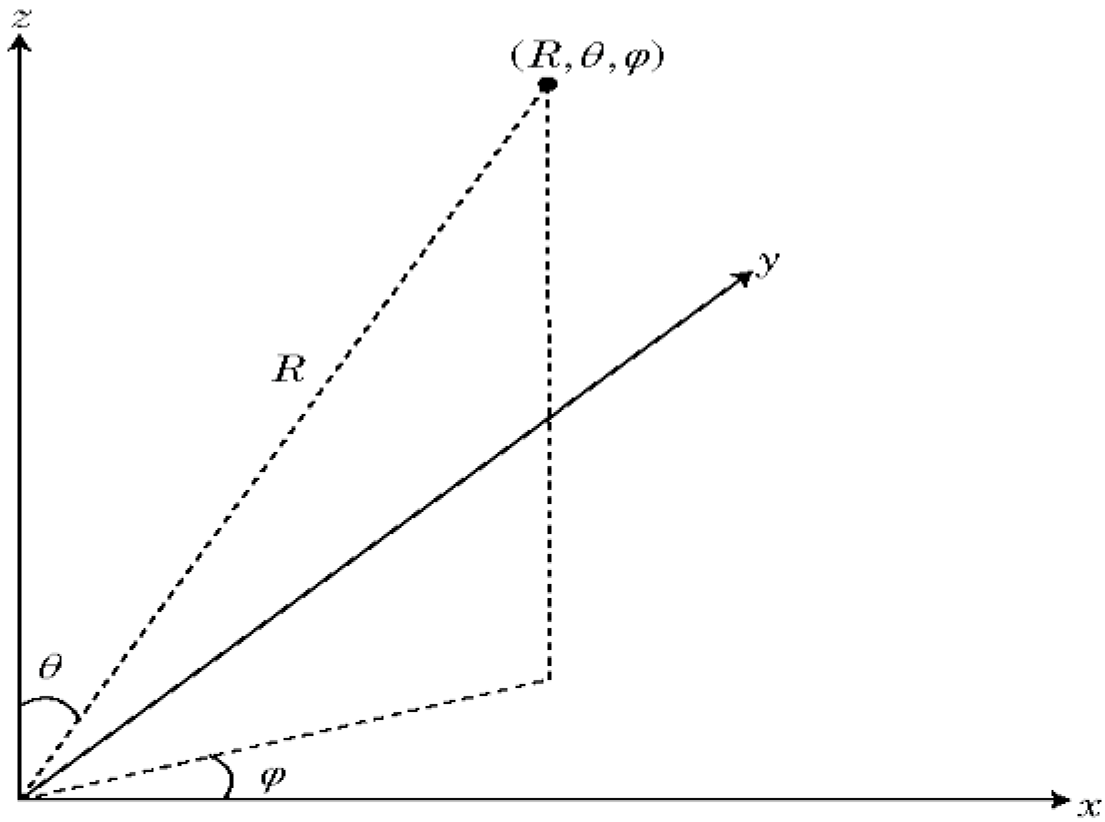


Phi (Φ) and Theta (θ) Angles

The φ angle is the angle from the positive *x*-axis to the vector's orthogonal projection onto the *xy* plane, moving in the direction towards the y-axis. The azimuth angle is between –180 and 180 degrees.

The θ angle is the angle from the positive *z*-axis to the vector itself. The θ angle is in the range 0 degrees and 180 degrees.

These angles are an alternative to using azimuth and elevation angles to express the location of point in a unit sphere.



u and v Coordinates

You can define *u* and *v* in terms of φ and θ:

$u = \sin\theta\cos\phi$

$v = \sin\theta\sin\phi$

In terms of azimuth and elevation angles, the *u* and *v* coordinates are:

$u = \cos el \sin az$

$v = \sin el$

The values of *u* and *v* satisfy the inequalities:

$-1 \leq u \leq 1$

$-1 \leq v \leq 1$

$u^2 + v^2 \leq 1$

The φ and θ angles in terms of *u* and *v* are:

$\tan\phi = u/v$

$\sin\theta = \sqrt{u^2 + v^2}$

The azimuth and elevation angles in terms of *u* and *v* are:

$$\sin el = v$$

$$\tan az = \frac{u}{\sqrt{1 - u^2 - v^2}}$$

Conversion Between Rectangular and Spherical Coordinates

Convert rectangular coordinates to spherical coordinates (*az*, *el*, *R*) using:

$$R = \sqrt{x^2 + y^2 + z^2}$$

$$az = \tan^{-1}\left(\frac{y}{x}\right)$$

$$el = \tan^{-1}\left(\frac{z}{\sqrt{x^2 + y^2}}\right)$$

Convert spherical coordinates (*az*, *el*, *R*) to rectangular coordinates using:

$$x = R\cos(el)\cos(az)$$
$$y = R\cos(el)\sin(az)$$
$$z = R\sin(el)$$

where:

- *R* is the distance from the antenna
- *el* and *az* are the azimuth and elevation angles

References

[1] Balanis, C.A. *Antenna Theory: Analysis and Design*. 3rd Ed. New York: Wiley, 2005.

Antenna Toolbox Limitations

The Antenna Toolbox does not support the following features.

Antenna Library

These antenna library objects do not support:

- PIFA and inverted–F antennas with the infinite ground plane.
- Antenna analysis at frequencies less than 1 kHz or greater than 200 GHz.
- `efficiency` works only with one feed location or port. And it does not support multiple non-air dielectric substrates.

Array Library

These array library objects do not support:

- Arrays using slot, and cavity antennas.
- Reflector-based arrays using helix, slot, Vivaldi, cavity, PIFA antennas as exciters.
- Building arrays using antennas with tilted ground planes.
- Conformal arrays created using unbalanced antennas with infinite ground plane.
- Infinite arrays using dielectric materials.

Interact with Polar Plot

This example shows how to interact with a polar plot created using `polarpattern` class.

Create Polar Plot of Helix Antenna

Create a helix antenna that has a 28 mm radius, a 1.2 mm width and 4 turns.

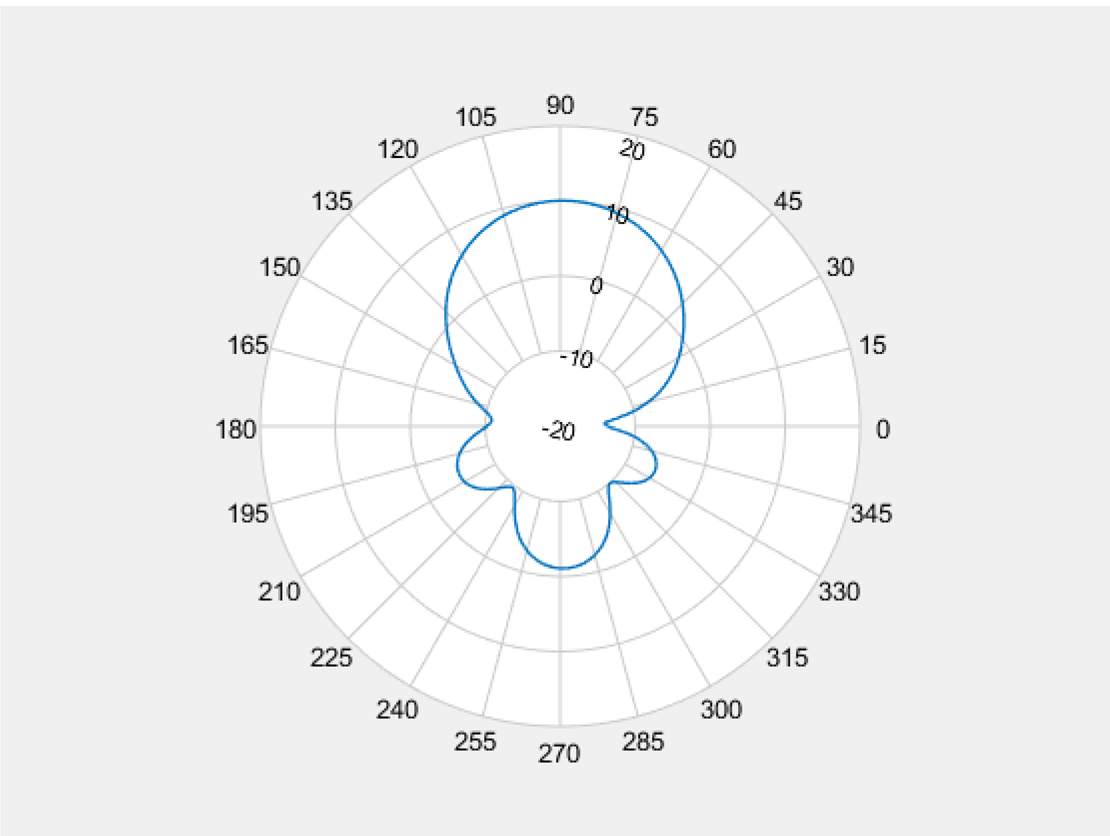
```
hx = helix('Radius',28e-3,'Width',1.2e-3,'Turns',4);
```

Calculate the directivity of the antenna at 1.8 GHz.

```
H = pattern(hx,1.8e9,0,0:1:360);
```

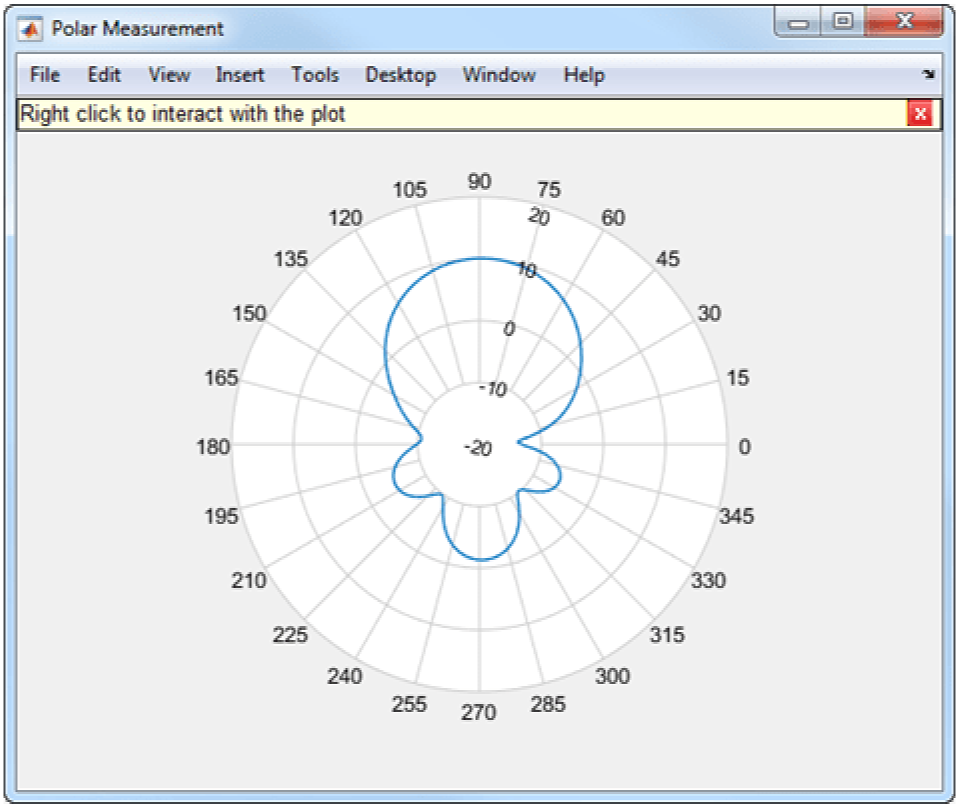
Display the polar plot of the antenna.

```
P = polarpattern(H);
```



Interact with Polar Plot

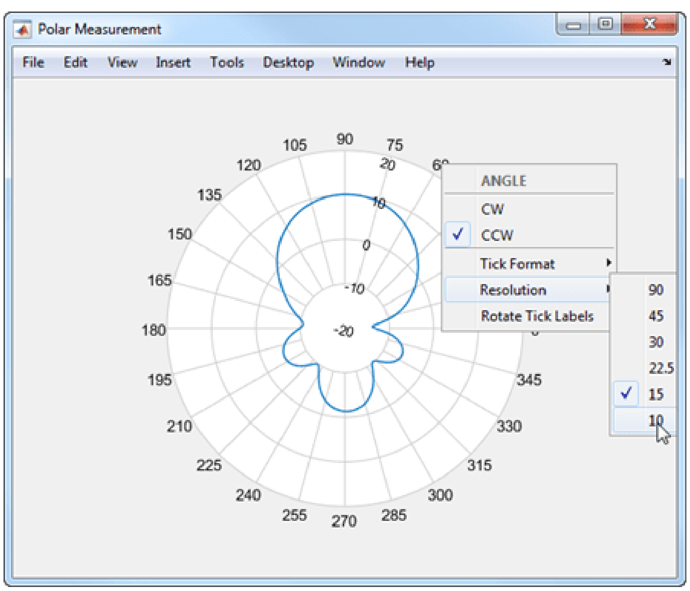
Hover over the plot. You see a message on top of the plot: **Right click to interact with the plot**. Right-click anywhere in the Polar Measurement window to display a context menu for interacting with the plot. For example, right-click outside the plot to show the **Main** context menu. Right-click inside the plot to show the **Display** context menu.

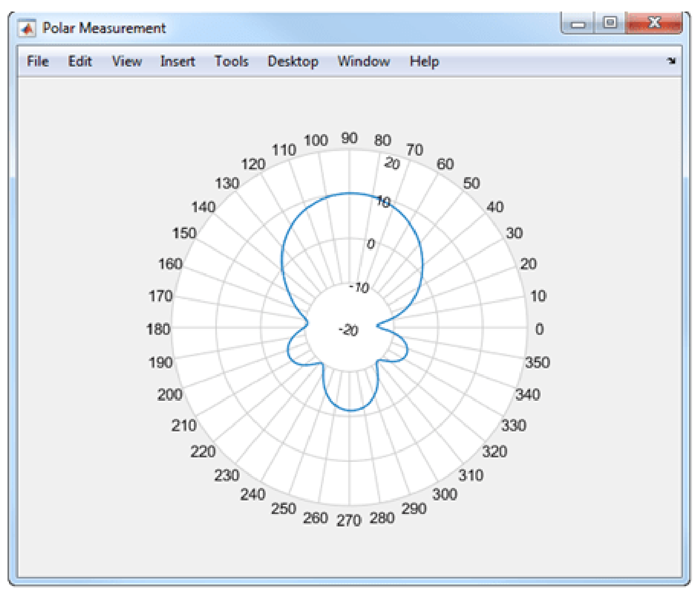


Update Angle and Magnitude Values

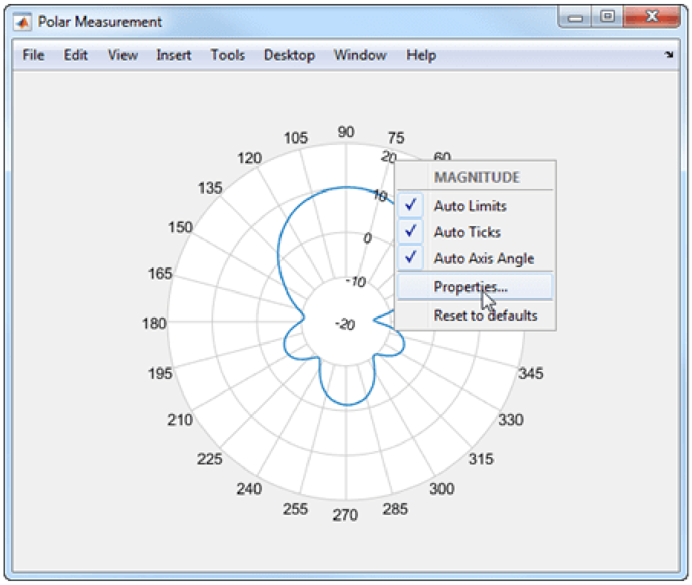
The angular values are around the circumference of the polar plot. Right-click any of the angle values to open **ANGLE** context menu. By default, the angles are displayed `CCW` (counterclockwise).

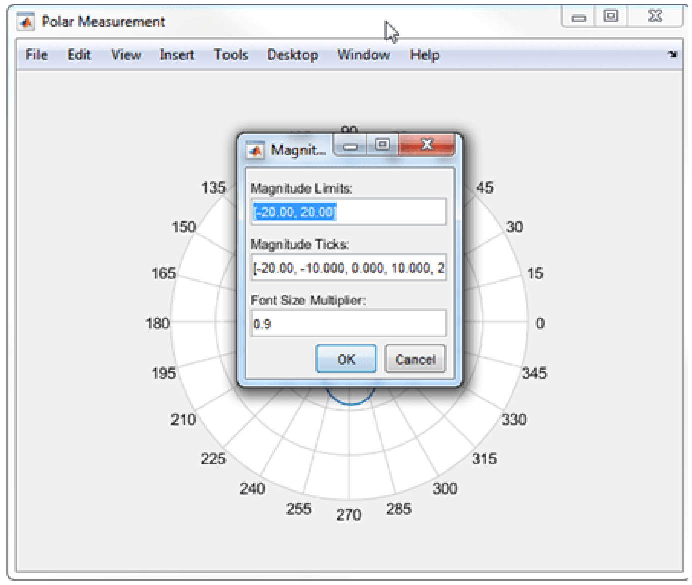
Change the resolution from 15 degrees to 10 degrees.





The magnitude values are on the radial lines of the plot. Right-click any of the magnitude values to open the **MAGNITUDE** context menu. Choose properties from the context menu to change the magnitude limits, magnitude ticks, or font size.

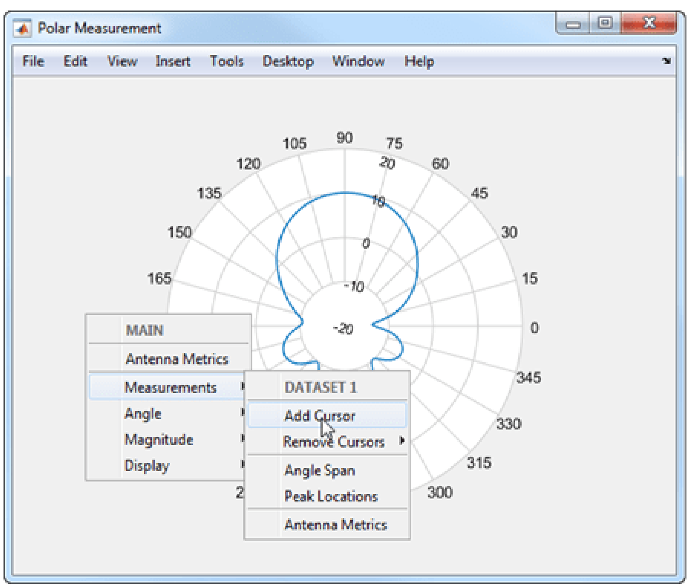


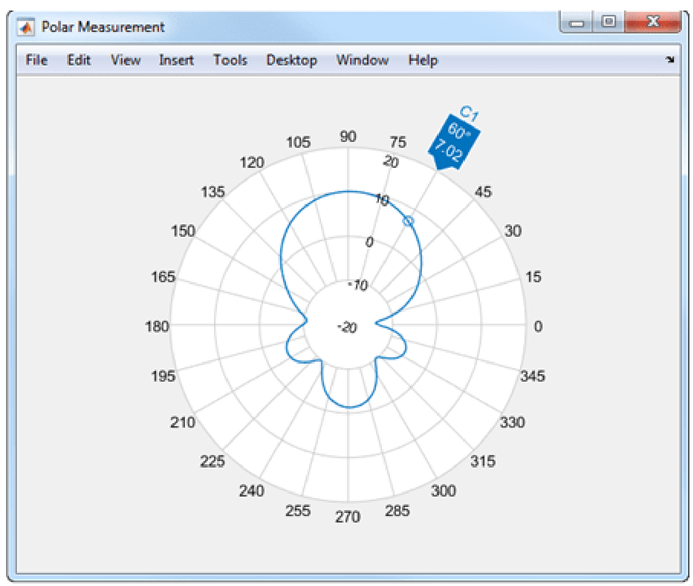


Add Cursors and Calculate Angle Span

Add cursors at 60 degrees and 105 degrees.

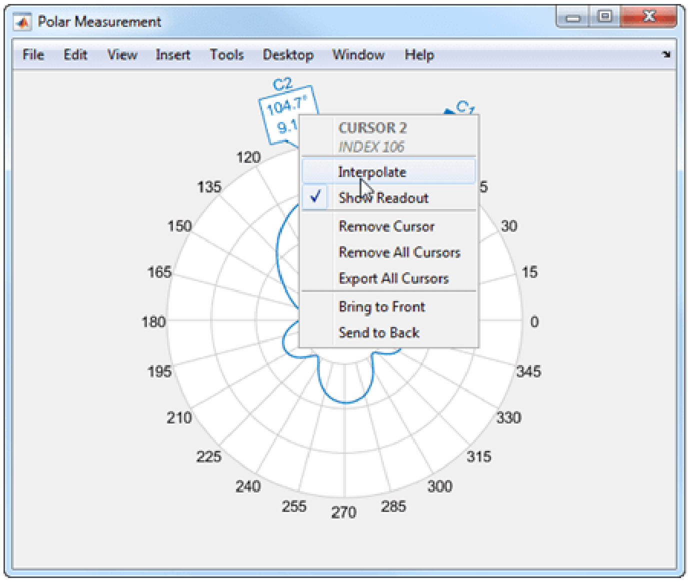
To add a cursor from the **MAIN** or **DISPLAY** context menus, select `Measurements` > **Add Cursor**. After adding the cursor, place the mouse pointer on the cursor and drag it to 60 degrees.

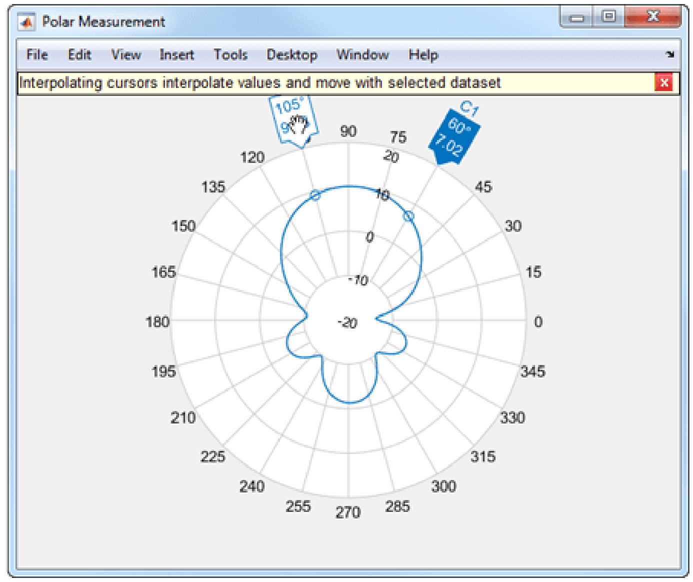




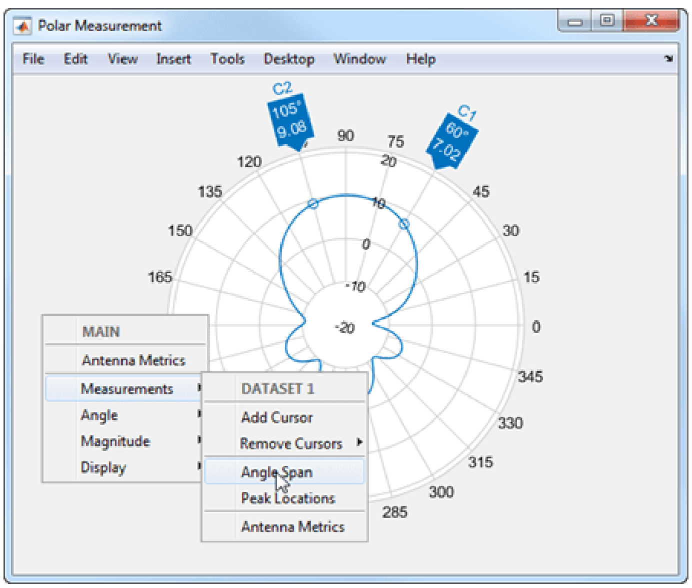
You can also add a cursor by double-clicking on the angle values. Double-click 105 to add a cursor. Right-click the newly added cursor and move the cursor to exact value of 105 degrees.

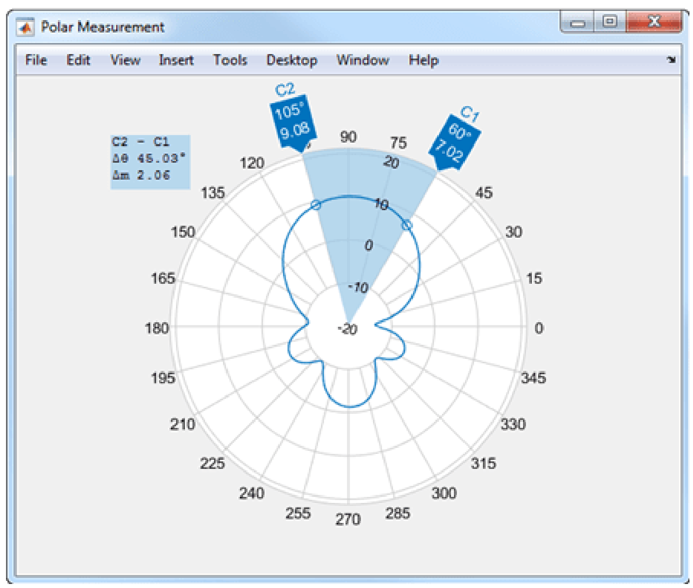
You can also interpolate the two angle values to 60 degrees and 150 degrees. Right-click on each cursor and choose **Interpolate** from the **CURSOR** context menu. To set the angle span, from the **MAIN** or **DISPLAY** context menu, select `Measurements` > **Angle Span**.





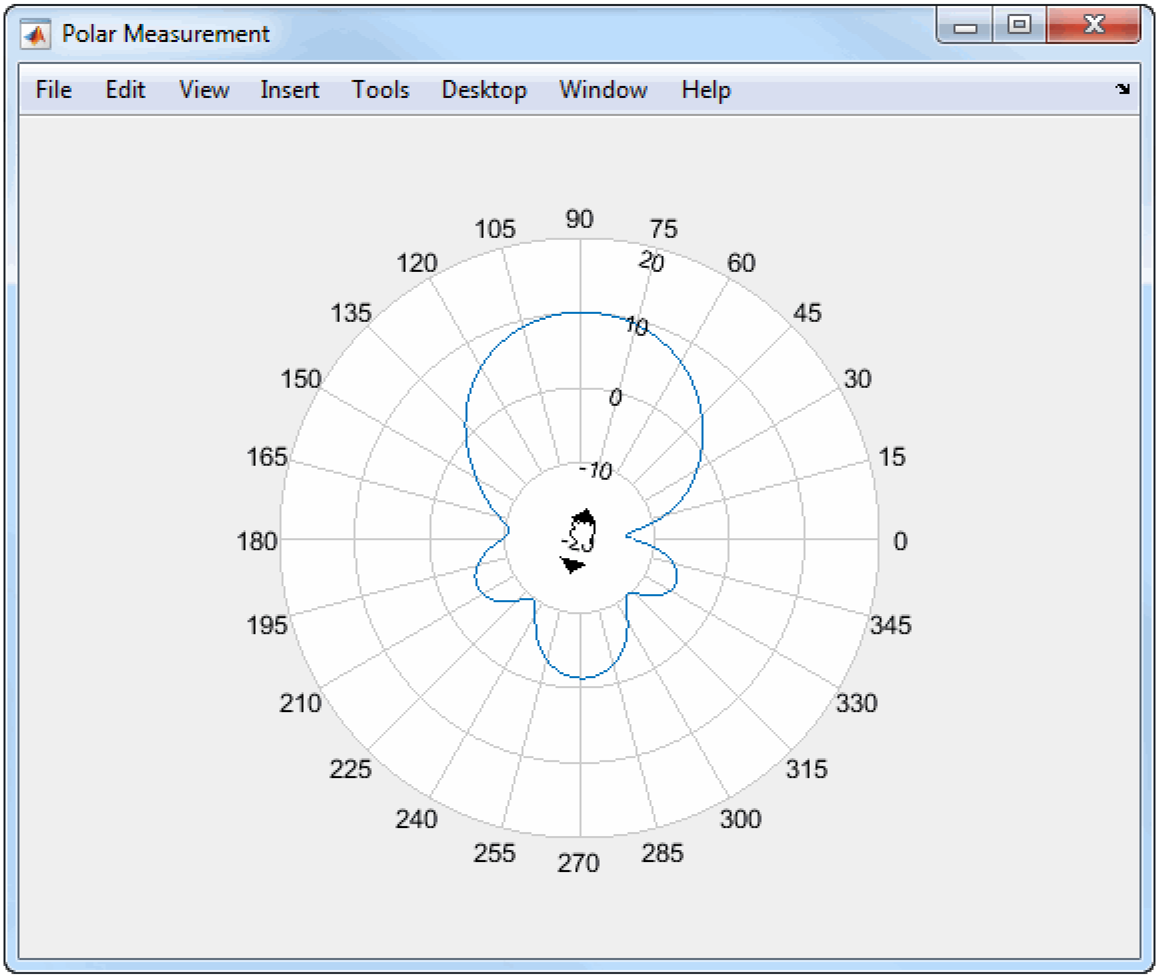
Calculate the counterclockwise angle span between 60 degrees and 105 degrees.





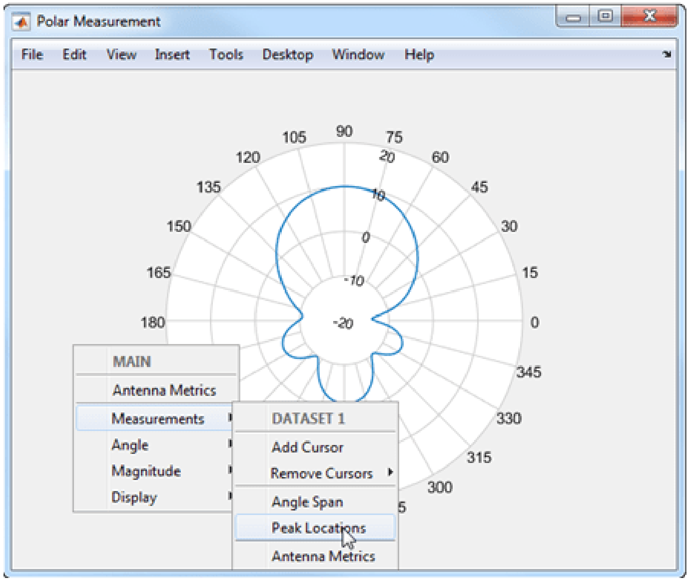
Zoom In and Zoom Out

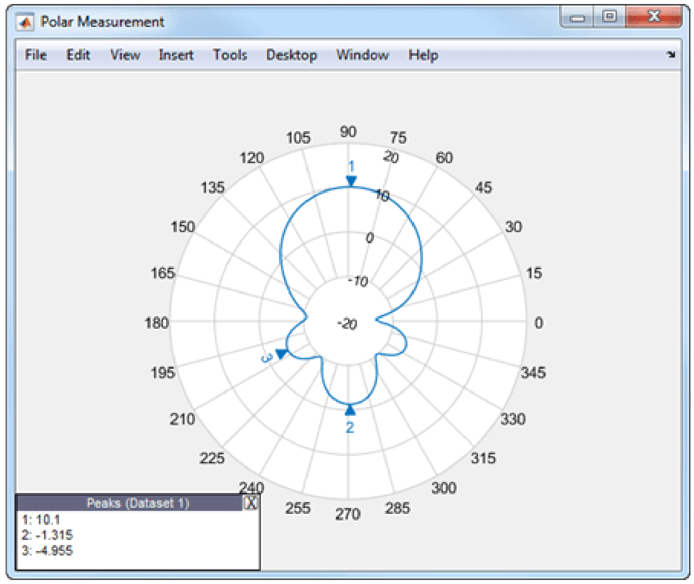
You can use the mouse pointer to zoom in and zoom out of the plot. Place the pointer at the center of the plot and drag radially outward.



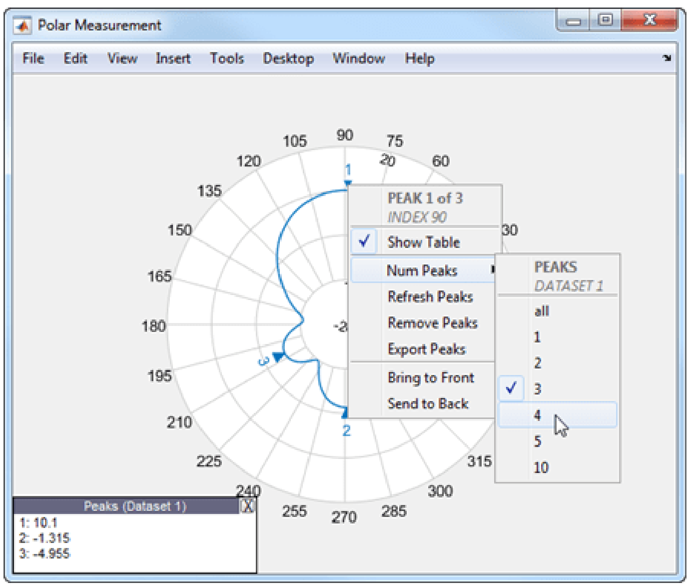
Display Peak Locations

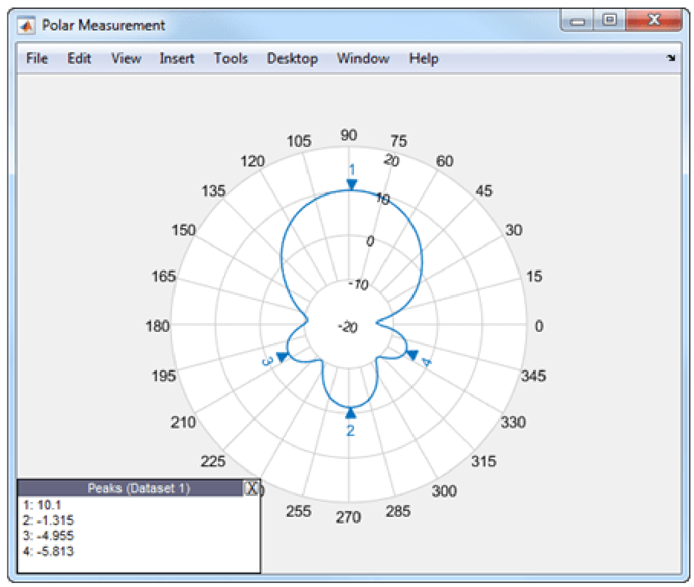
To display the peak locations from the **MAIN** or **DISPLAY** context menus, select `Measurements` > **Peak Location**.





Right-click any of the peak triangles and choose **NumPeaks**. Increase the number of peaks to 4.



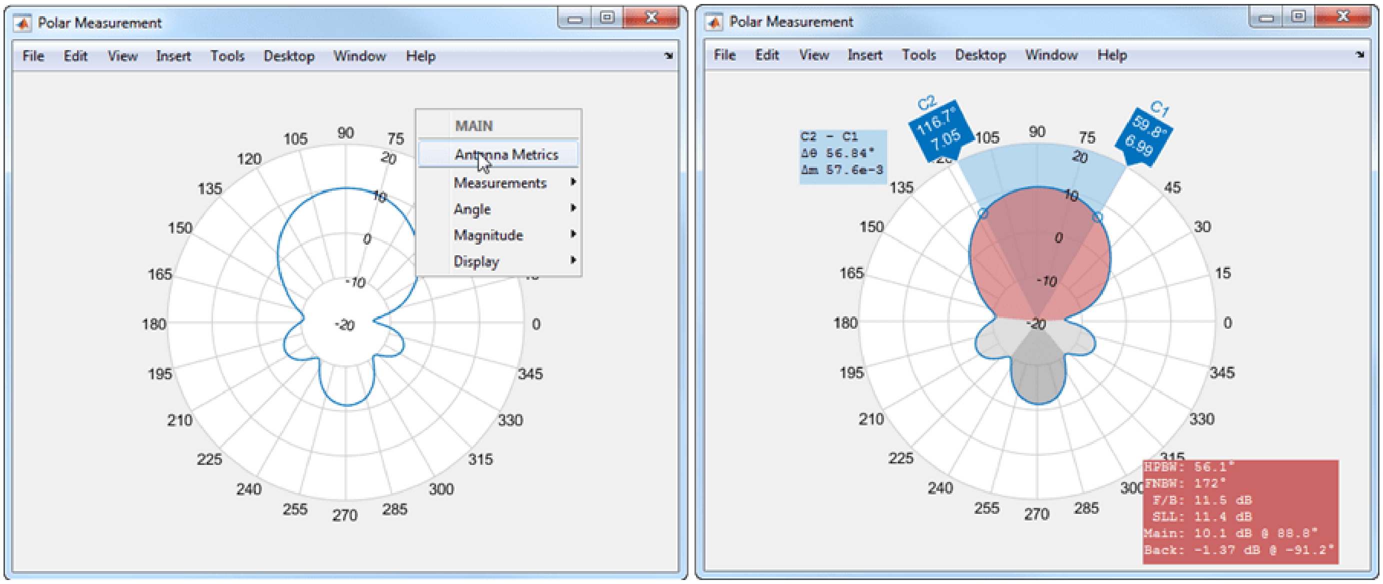


View Antenna Metrics

Antenna metrics in the polar plot display the main, back, and side lobes of the antenna. There are two ways to turn display antenna metrics on the plot:

1. Right-click within the polar figure to open the **Main** context menu. Choose **Antenna Metrics**.

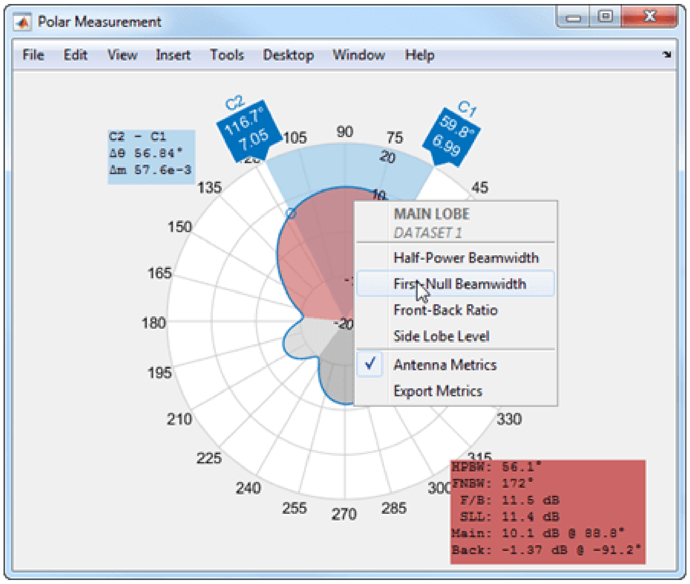
2. Right-click inside the polar plot to open the **Display** context menu. Inside the **Measurement** menu, choose Antenna Metrics.

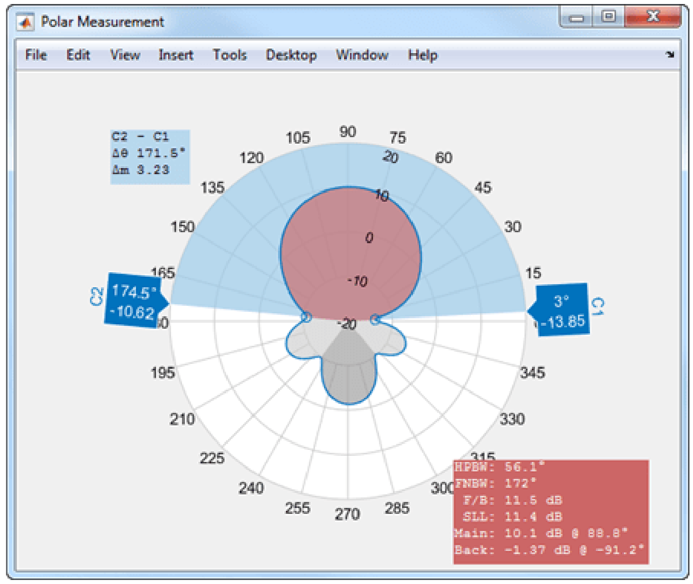


By default, the plot shows the HPBW (half-power beamwidth) of the antenna. The antenna measurements text box displays:

- `HPBW` (half-power beamwidth)
- `FNBW` (first-null beamwidth)
- `F/B` (front-to-back ratio)
- `SLL` (side lobe level)
- `Main` (main lobe peak value and corresponding angle)
- `Back` (back lobe peak value and corresponding angle)

To view the FNBW, right-click inside the red or gray polar plot region to open the **MAIN LOBE** or the **BACK LOBE** context menu and then choose **First-Null Beamwidth**.





See Also

"Antenna Far-Field Visualization"

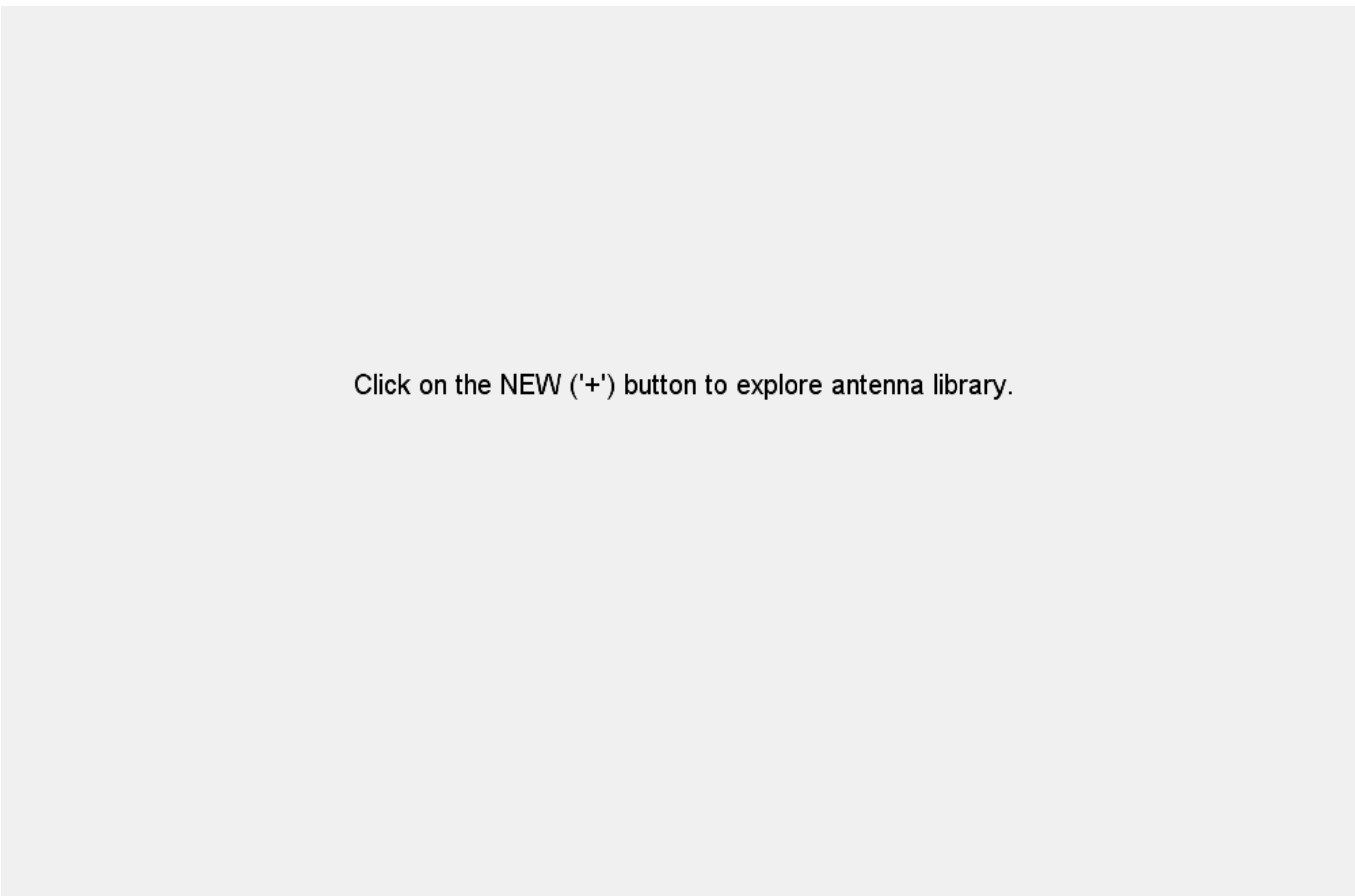
Design and Analysis Using Antenna Designer

This example shows how to construct, visualize, and analyze a helix antenna element using the Antenna Designer App.

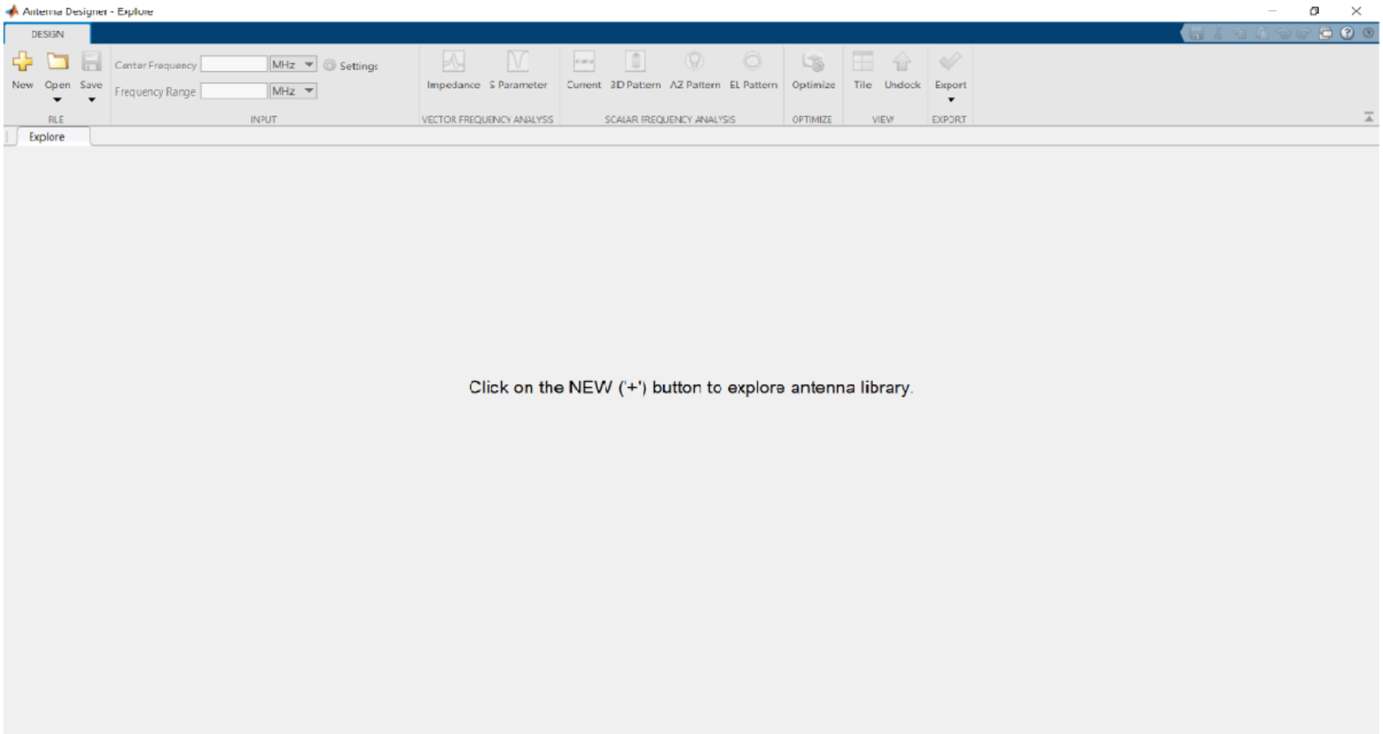
Open Antenna Designer App

To open the app, at the `MATLAB` command prompt, enter:

```
antennaDesigner
```

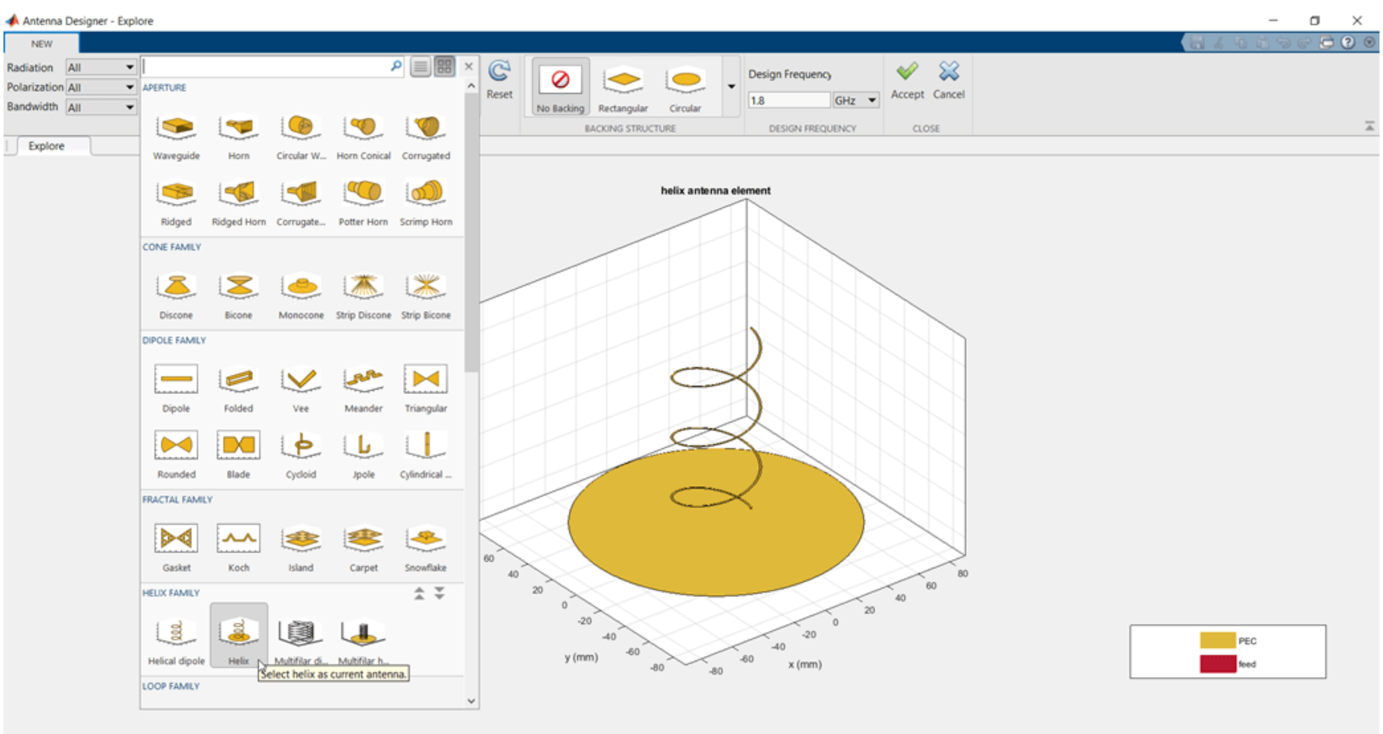


This command opens a blank canvas.



Design Helix Antenna

In the blank canvas, click **NEW**. In the **ANTENNA GALLERY**, under **HELIX FAMILY**, select a helix antenna. Set the **Design Frequency** to 1.8 GHz.



To analyze the helix antenna, click **Accept**.

Change Helix Antenna Properties

In the **Antenna Properties** tab, change the following:

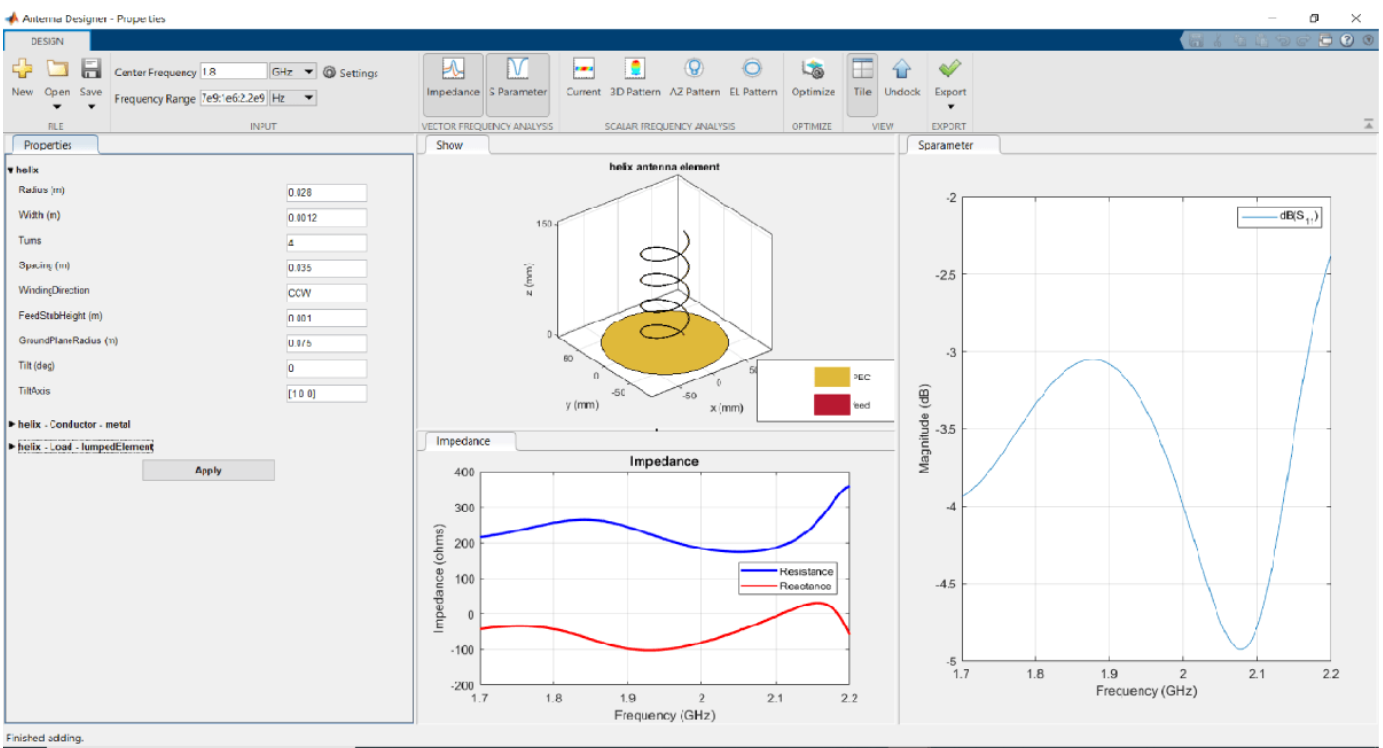
- **Radius** = 0.0280
- **Width** = 0.0012
- **Turns** = 4
- **Spacing** = 0.0350
- **GroundPlaneRadius** = 0.0750

Click **Apply** to see the change in the helix antenna structure.

Plot Impedance and S-Parameters

Open the **Load-helix** section and change the **Impedance** of the antenna to 72 ohms. Click **Apply**. In the toolstrip, under **INPUT** tab, change the **Frequency Range** to `1.7e9:1e6:2.2e9` Hz.

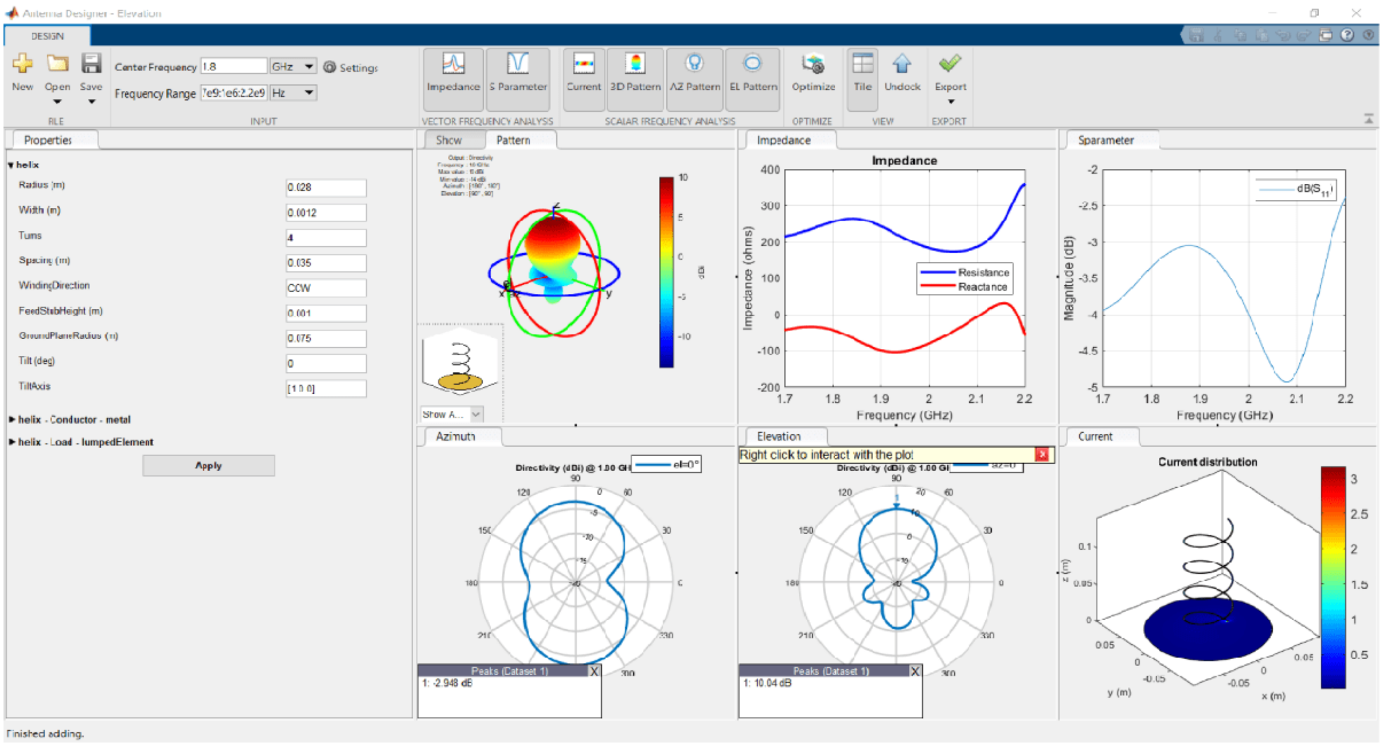
Click **Impedance** to plot the impedance of the helix antenna. Click **S Parameter** to plot the S11 value of the helix antenna. Click **Tile** to view the plots together.



Plot Current Distribution, 3-D, Azimuth, and Elevation Patterns

In the **SCALAR FREQUENCY ANALYSIS** section of the toolstrip, click **Current** to view the current distribution of the helix at 1.8 GHz.

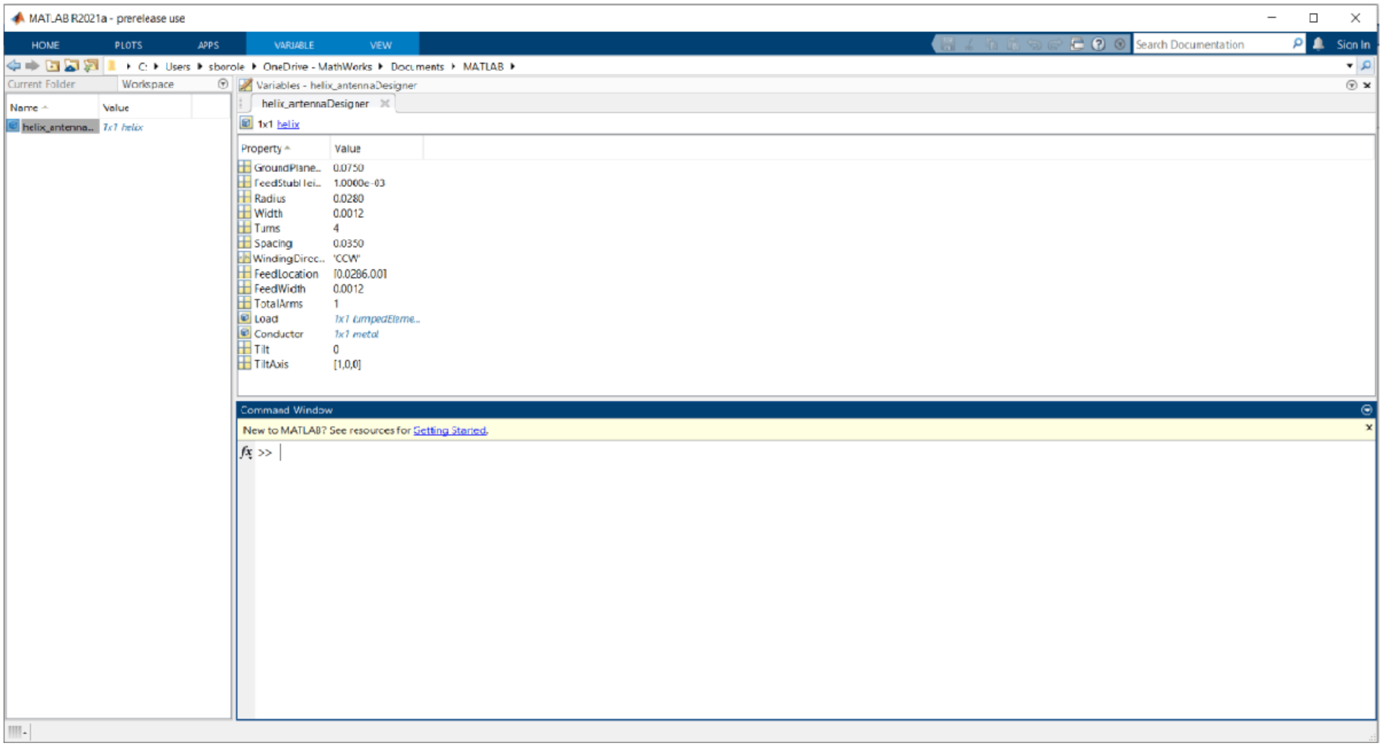
Click **3D Pattern**, **AZ Pattern**, and **EL Pattern** to view the radiation, azimuth, and elevation patterns of the helix antenna, respectively. Click **Tile** again to view all the plots together.



You can compare the results of this tutorial with the results of “Antenna Modeling and Analysis” on page 1-3 tutorial.

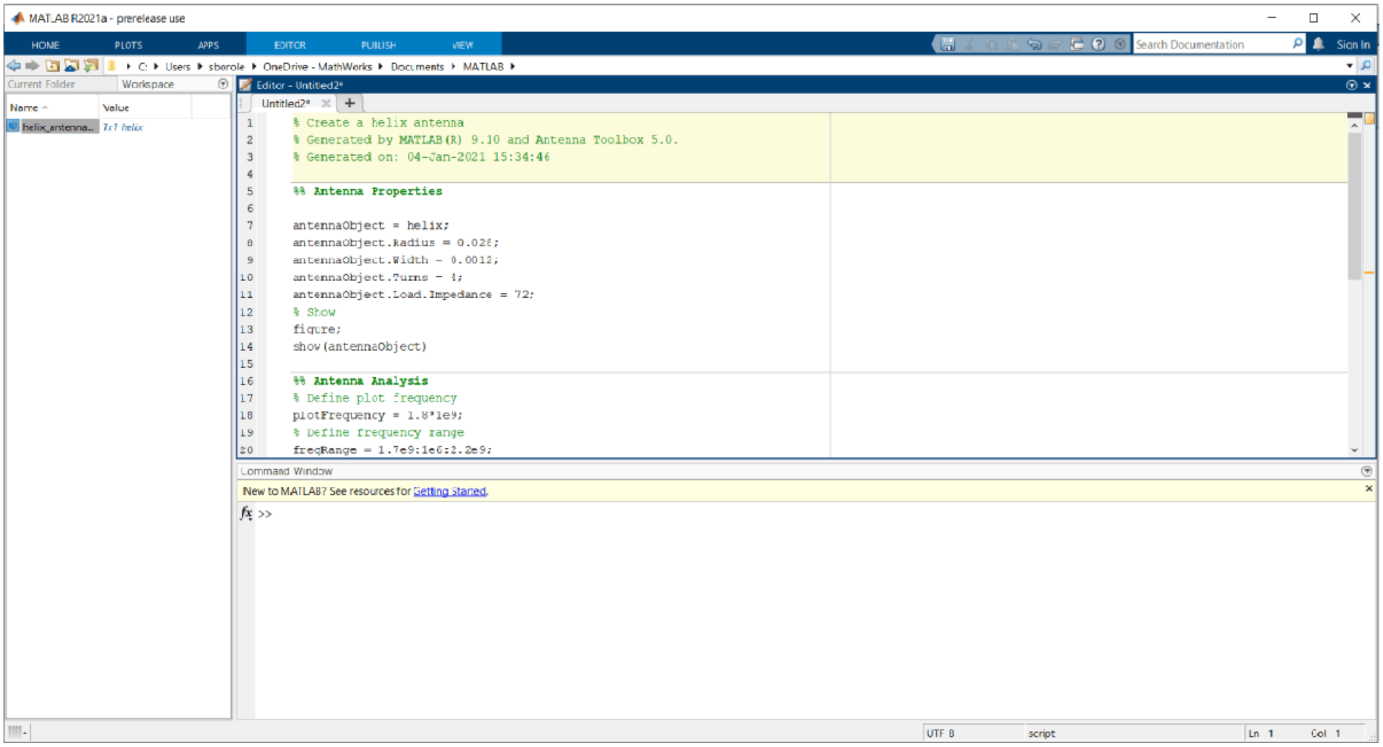
Export to MATLAB Workspace

Click the **Export** button arrow and then click **Export to workspace**. In the **Export to workspace** window, type the name of the antenna file. Click on the variable in the workspace to view the properties of the helix antenna.



Export to MATLAB Script

Click the **Export** button arrow again and then click **Export to script** to view the helix antenna and analysis in MATLAB script format. The script has two sections: **Antenna Properties** and **Antenna Analysis.**


MATLAB R2021a - prerelease use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
C: ▸ Users ▸ sborole ▸ OneDrive - MathWorks ▸ Documents ▸ MATLAB ▸
Current Folder
Workspace
Name
Value
helix_antenna...
1x1 helix
Editor - Untitled2*
Untitled2*
% Create a helix antenna
% Generated by MATLAB(R) 9.10 and Antenna Toolbox 5.0.
% Generated on: 04-Jan-2021 15:34:46
%% Antenna Properties
antennaObject = helix;
antennaObject.Radius = 0.028;
antennaObject.Width = 0.0012;
antennaObject.Turns = 4;
antennaObject.Load.Impedance = 72;
% Show
figure;
show(antennaObject)
%% Antenna Analysis
% Define plot frequency
plotFrequency = 1.8*1e9;
% Define frequency range
freqRange = 1.7e9:1e6:2.2e9;
Command Window
New to MATLAB? See resources for Getting Started.
fx >>
UTF-8
script
Ln 1
Col 1

Design, Analyze, and Optimize H-Notch Patch Using Design Variables

This tutorial shows how to define the metal patch, dielectric, and ground plane layers in H-notch using design variables in the PCB Antenna Designer app and optimize the design for maximum gain.

Open New Session

Type this command at the command line to open the **PCB Antenna Designer** app.

```
pcbAntennaDesigner
```

On the **Design** tab, click **New Session** to start a new session and open a blank canvas.

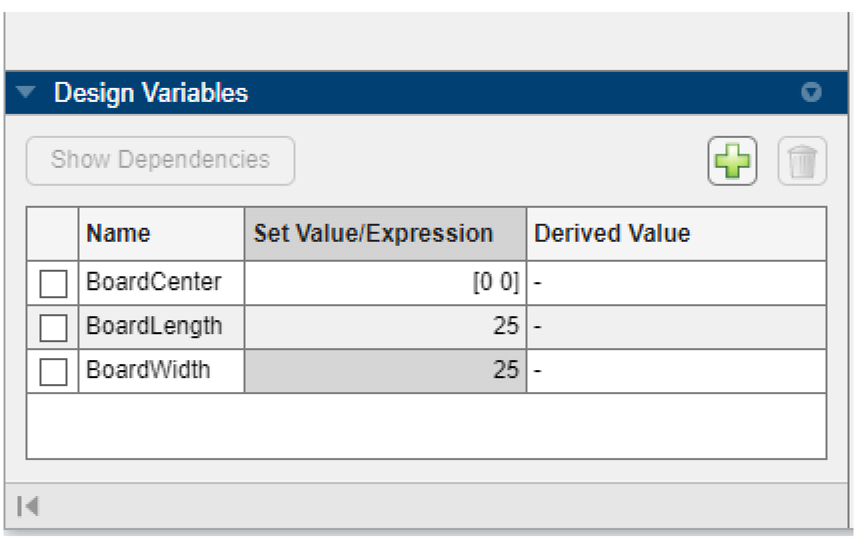
Define Board Shape

To design a PCB stack, you must first define a board shape. Select **Rectangle** from the `Shapes` section on the toolbar. Drag the shape on the canvas to create a rectangle.

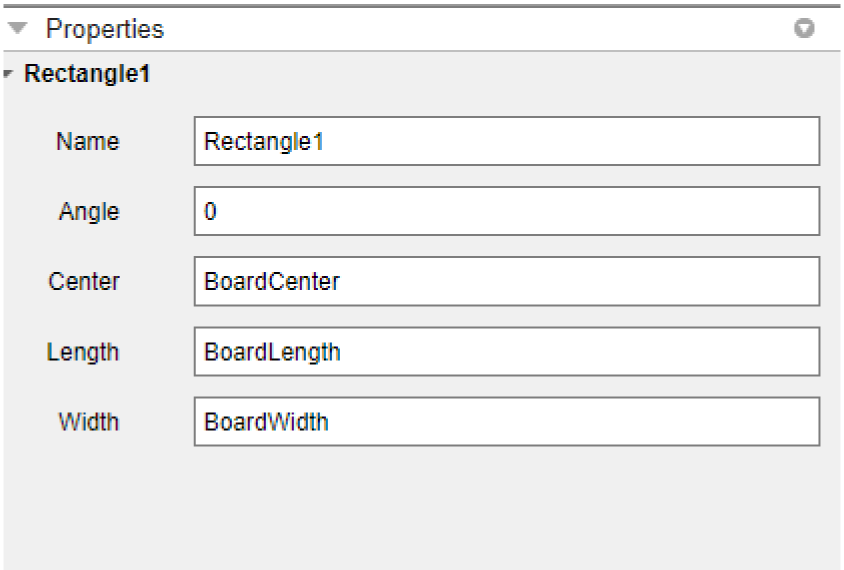
Use the **Design Variables** tab to add variables for the board center, board length, and board width.

To add a new variable, click . Add the following to the table:

- **BoardCenter** — `[0,0]`
- **BoardLength** — 25
- **BoardWidth** — 25



Add the variables to the `Rectangle1` properties tab.



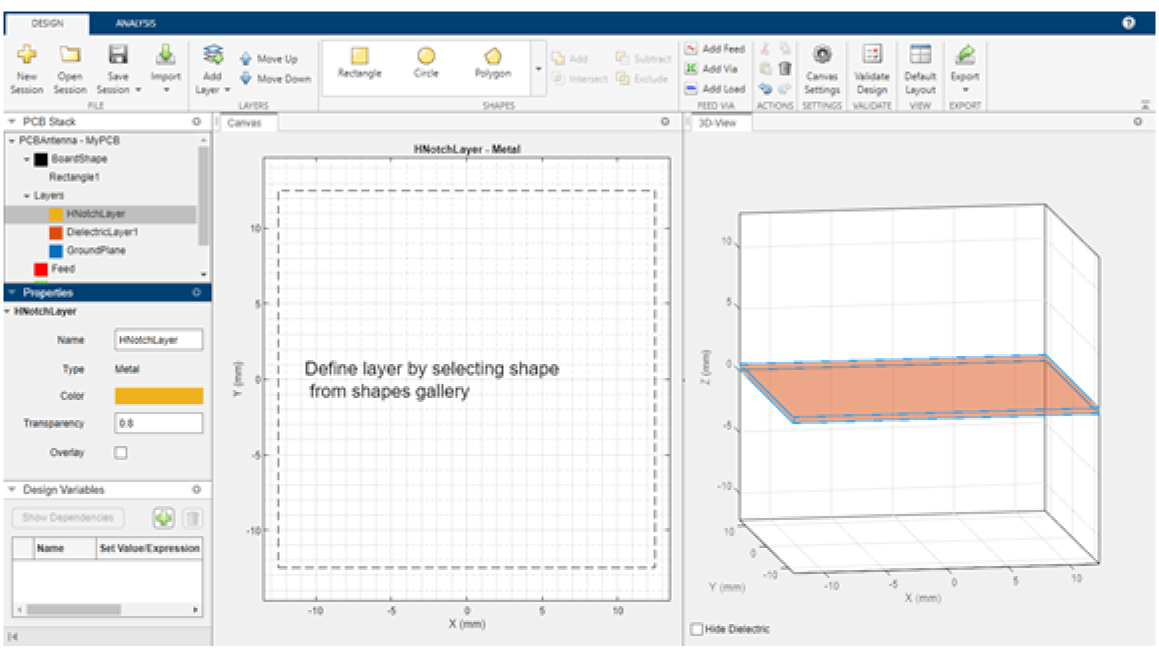
Add Ground, Metal, and Dielectric Layer

Click **Add Layer** on the toolbar and then select **Metal Layer** to add a metal layer.

Set the **Name** of the metal layer to `GroundPlane`.

Click **Add Layer** on the toolbar and then select **Dielectric Layer** to add a dielectric layer `DielectricLayer1`.

Click **Add Layer** on the toolbar and then select **Metal Layer** to add a metal layer. Set the name of this layer to `HNotchLayer`.

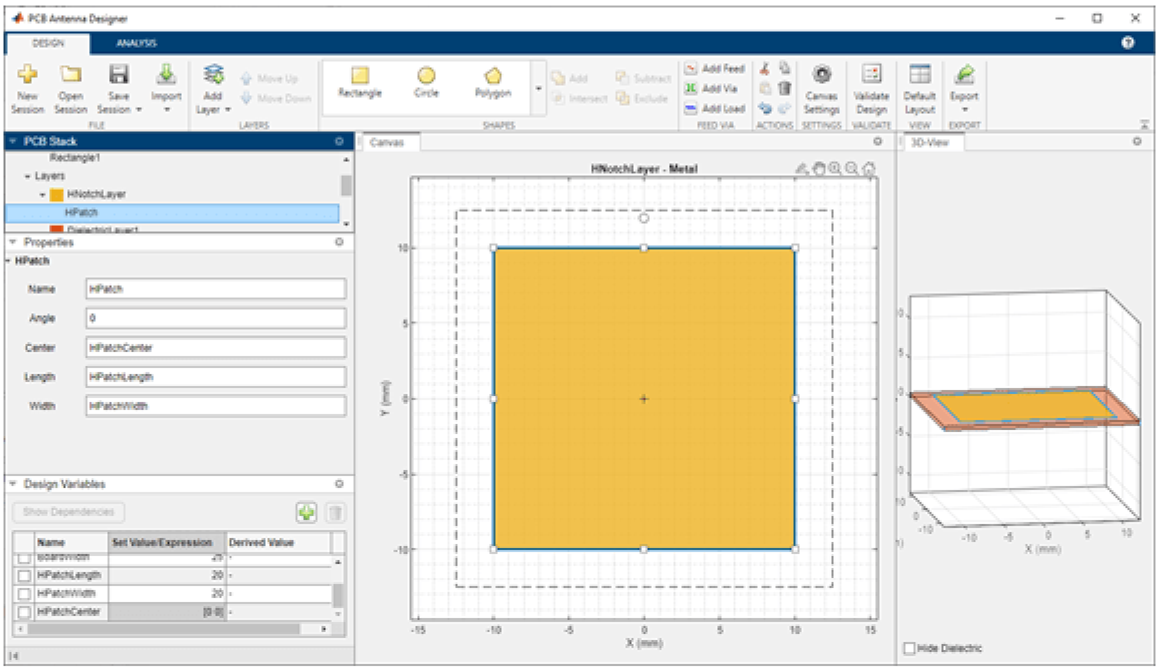


Create H-Notch

Select the `HNotchLayer` on the PCB Stack navigation tree and then select **Rectangle** from the **Shapes** section on the toolbar. Drag the shape onto the canvas to create a rectangular patch.

Use the **Design Variables tab** to set the properties of the HPatch.

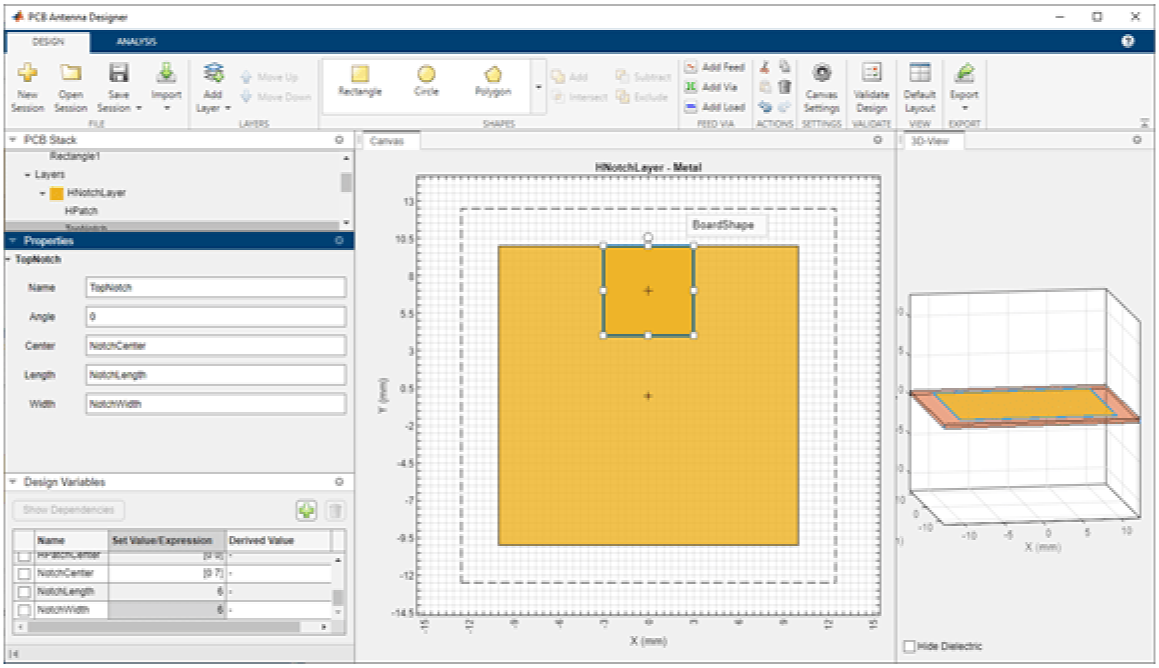
- **Center** — `[0,0]`
- **Length** — `20`
- **Width** — `20`



Create Top Notch

Select **Rectangle** from the **Shapes** section and drag the shape to the top of the shape to create a top notch in the patch layer. Set the properties of `Rectangle3` to the following using the **Design Variable** tab.

- **Name** — `TopNotch`
- **Center** — `[0,7]`
- **Length** — `6`
- **Width** — `6`

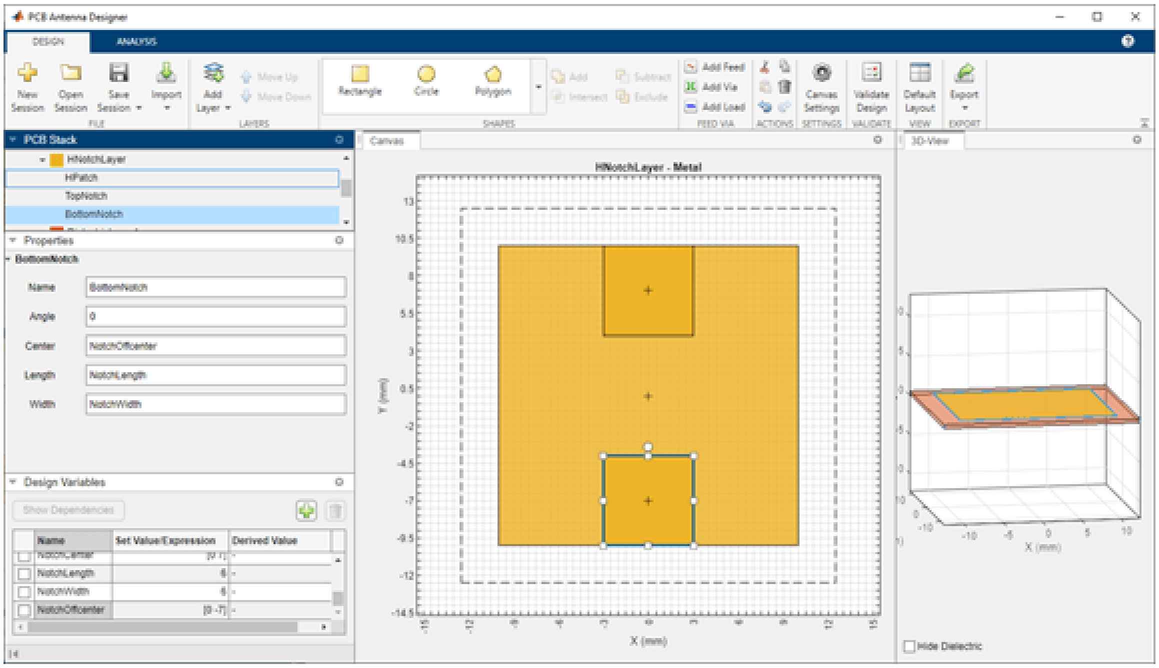


Create Bottom Notch

Select `TopNotch` from the canvas and then copy the layer by clicking **Copy** in the **Actions** section. Click **Paste** to paste a copy of `TopNotch`. Doing so creates a rectangle `TopNotch_Copy_1` with identical dimensions to the `TopNotch` layer.

Set the properties of `TopNotch_Copy_1` to the following:

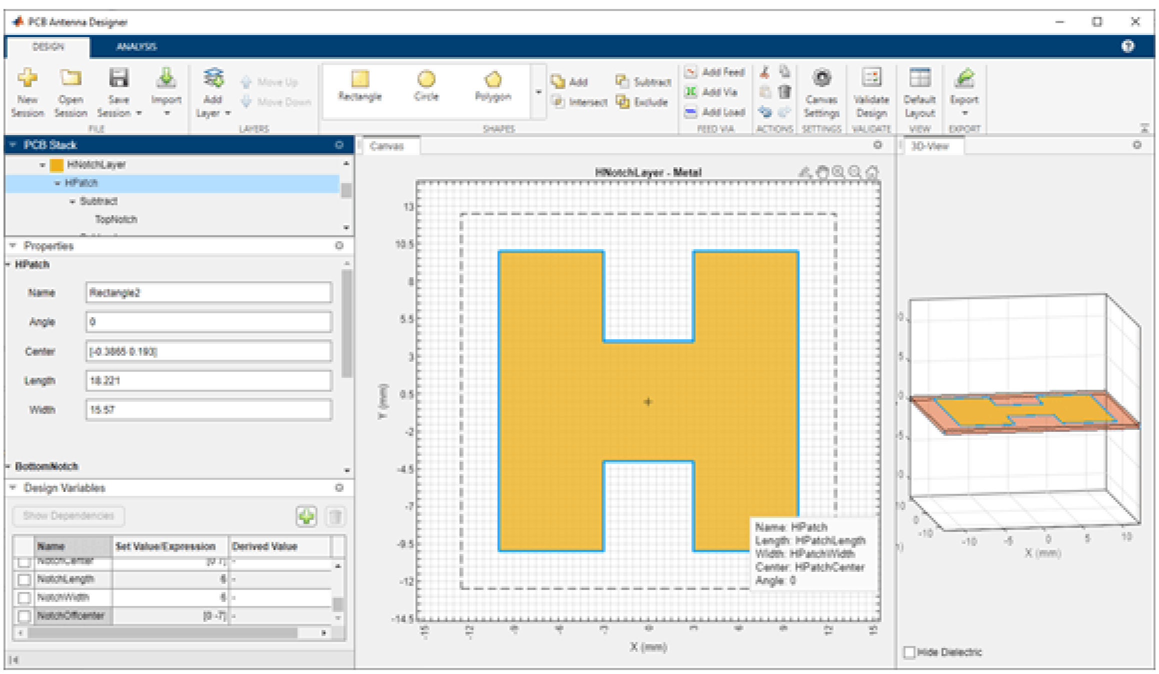
- **Name** — `BottomNotch`
- **Center** — `[0,-7]`



Subtract Top and Bottom Notch Layers from Metal Patch

Select `HPatch` and `TopNotch` and then click **Subtract** in the **Shapes** section of the toolbar to remove both rectangles.

Select the `HPatch` and `BottomNotch` and click **Subtract** in the **Shapes** section of the toolbar to remove both rectangles.



Set Dimensions for Ground Plane

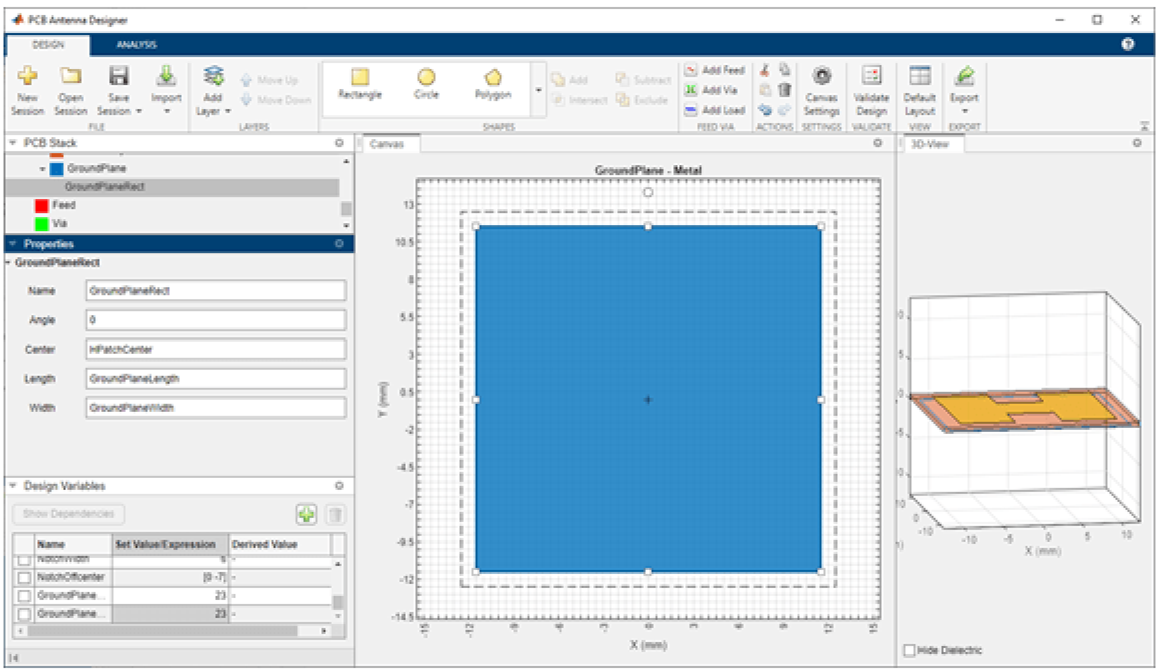
The ground plane must be same size as the HNotchLayer. To make the ground plane the same size as HPatch, right-click HPatch and select **Copy**.

Right-click the GroundPlane and click **Paste** on the toolbar to paste the rectangle in the GroundPlane.

The app copies the rectangle to the ground plane layer with the name HPatch_Copy(1). HPatch_Copy(1) and HPatch have the same dimensions.

Set the properties of HPatch_Copy(1) to the following:

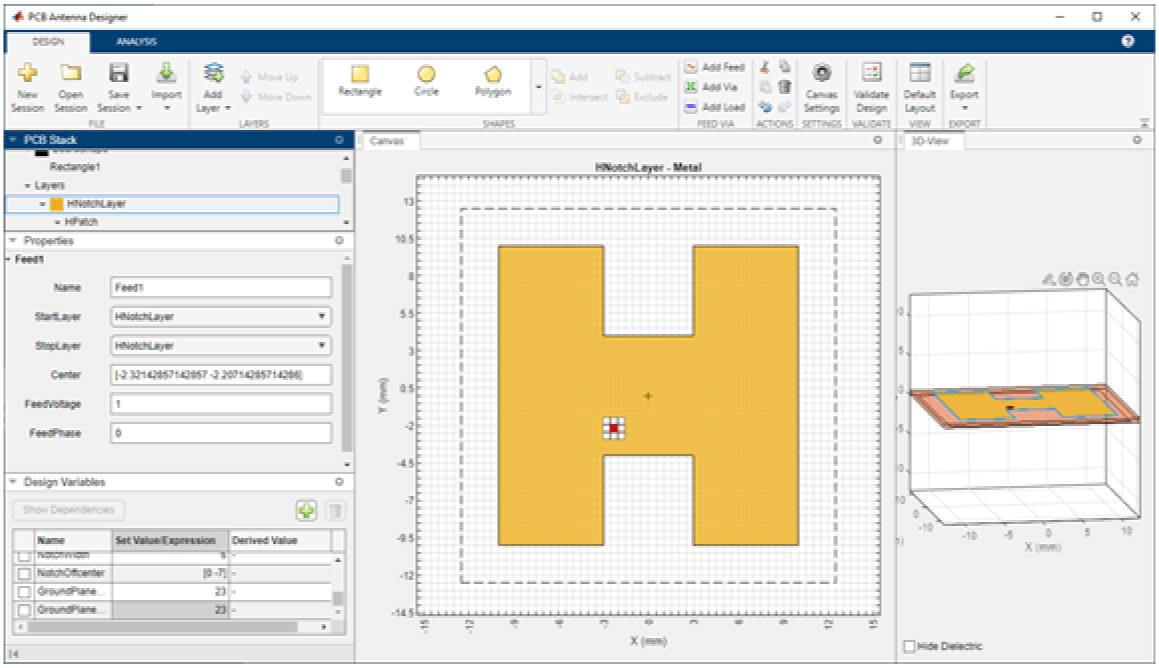
- **Name** — GroundPlaneRect
- **Center** — [0,0]
- **Length** — 23
- **Width** — 23



Add Feed

Select the HNotchLayer metal layer on the PCB Stack navigation tree and then click **Add Feed** in the **Feed Via** section on the toolbar. You can add a feed only to a metal layer.

As the PCB Stack navigation tree shows, the app adds the feed to HNotchLayer as Feed1.

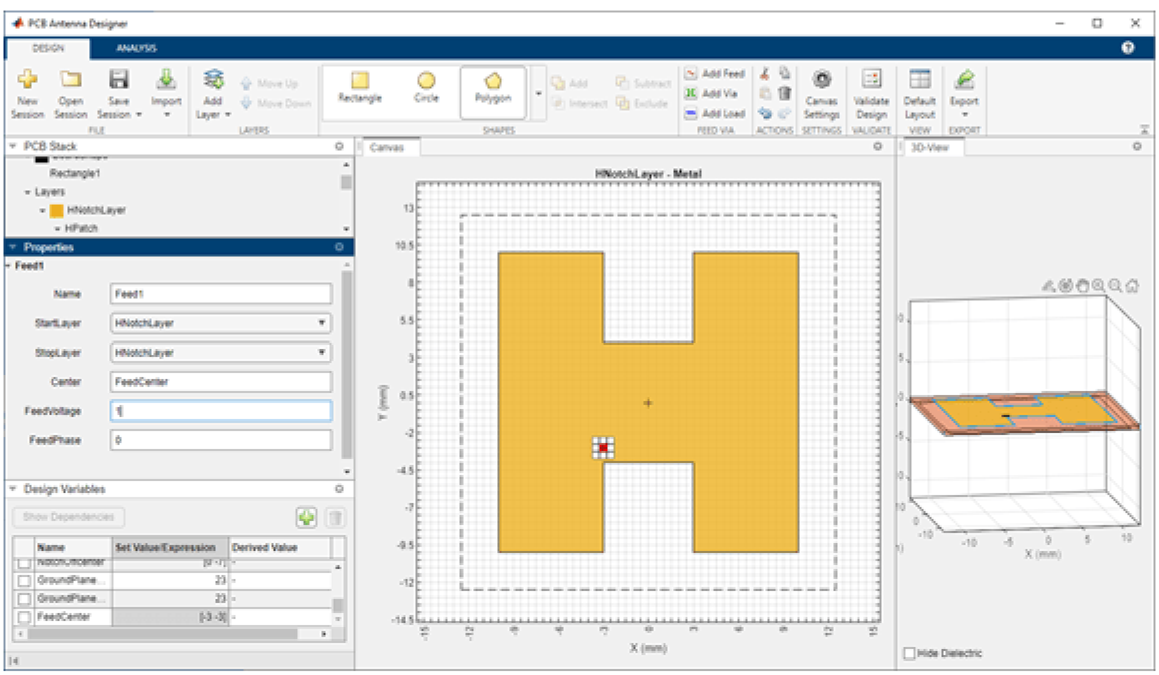


Select the parent `Feed` node on the PCB Stack navigation tree, and then set the **FeedDiameter** to `1` and **FeedViaModel** to `Strip`. **FeedDiameter** is a global property.

Change Position of Feed

Select `Feed1` on the PCB Stack navigation tree. In the **Properties** pane, set **Center** to `[-3,-3]`.

In the **3D-View** pane, you can see that the feed is connected only to `HNotchLayer`.



For more information on feeds and vias, refer to the `FeedLocation` and `ViaLocation` property descriptions in the `pcbStack` object documentation.

Change Start Layer and Stop Layer

Connect the feed from `HNotchLayer` to `GroundPlane` by setting **StartLayer** to `HNotchLayer` and **StopLayer** to `GroundPlane`. This creates a feed on the `GroundPlane` which connects to the `HNotchLayer`.

Change Metal Properties

Select the `Layers` in on the PCB Stack navigation tree and set the **Type** of the metal layer to `Copper` in the **Properties** pane.

Validate Design

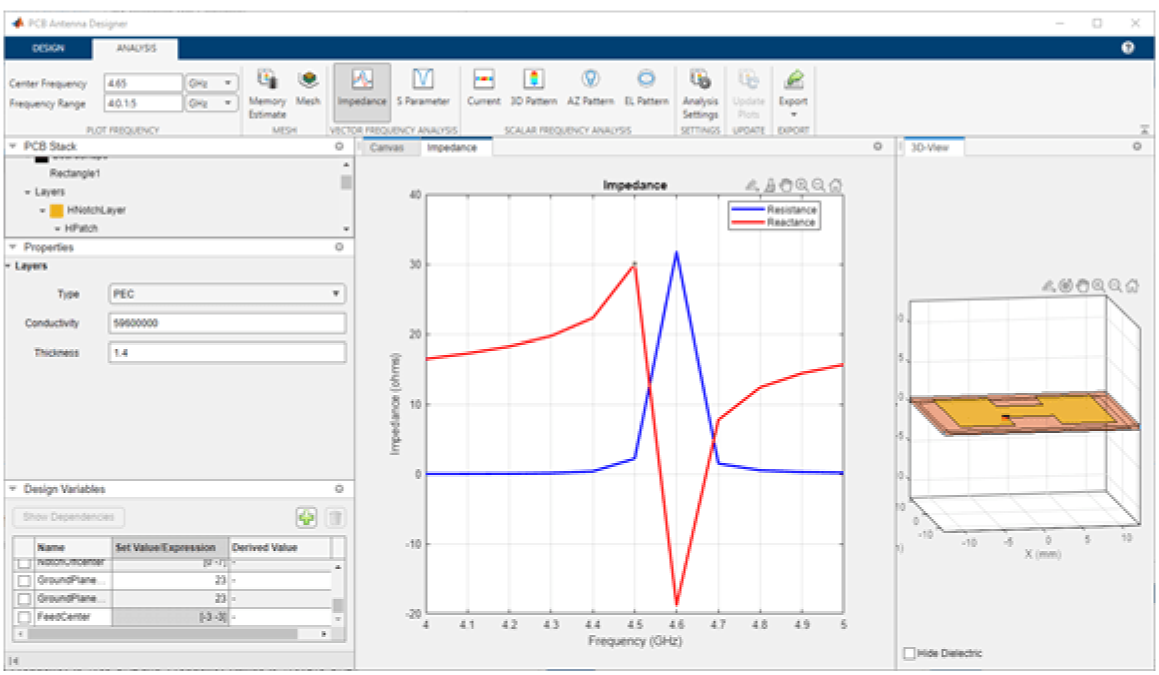
Click **Validate Design** on the toolbar to validate your board shape, layers, feed, via, and load.

Analyze H-Notch Unit Element

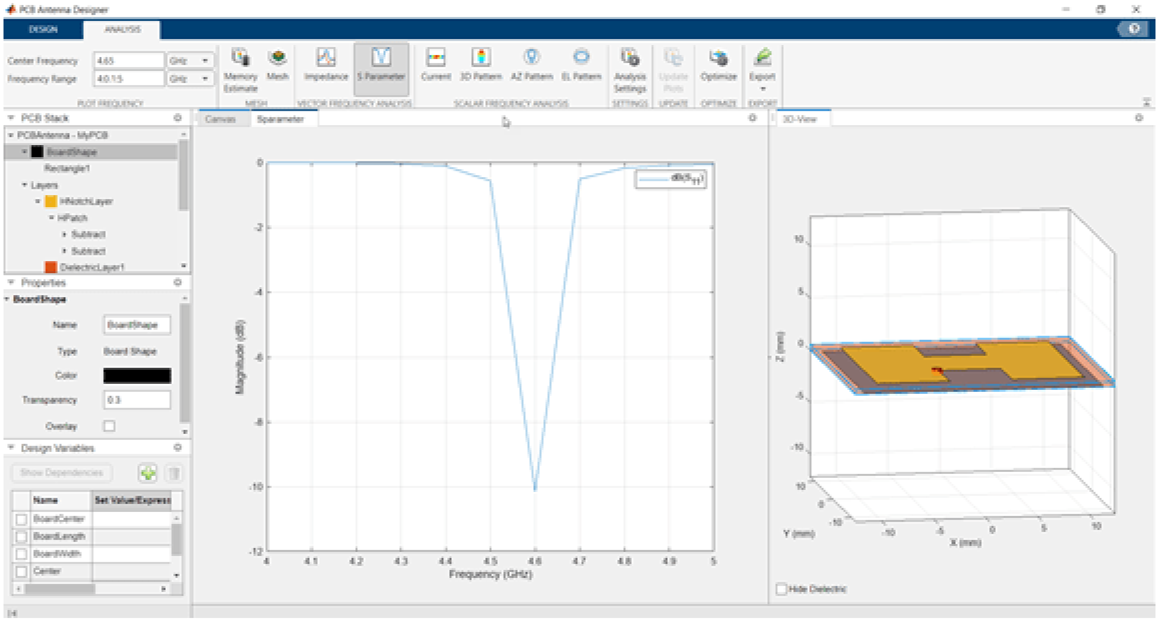
In the **Analysis** tab, set the **Center Frequency** to `4.65` GHz and **Frequency Range** to `4:0.1:5` GHz

Run Analysis

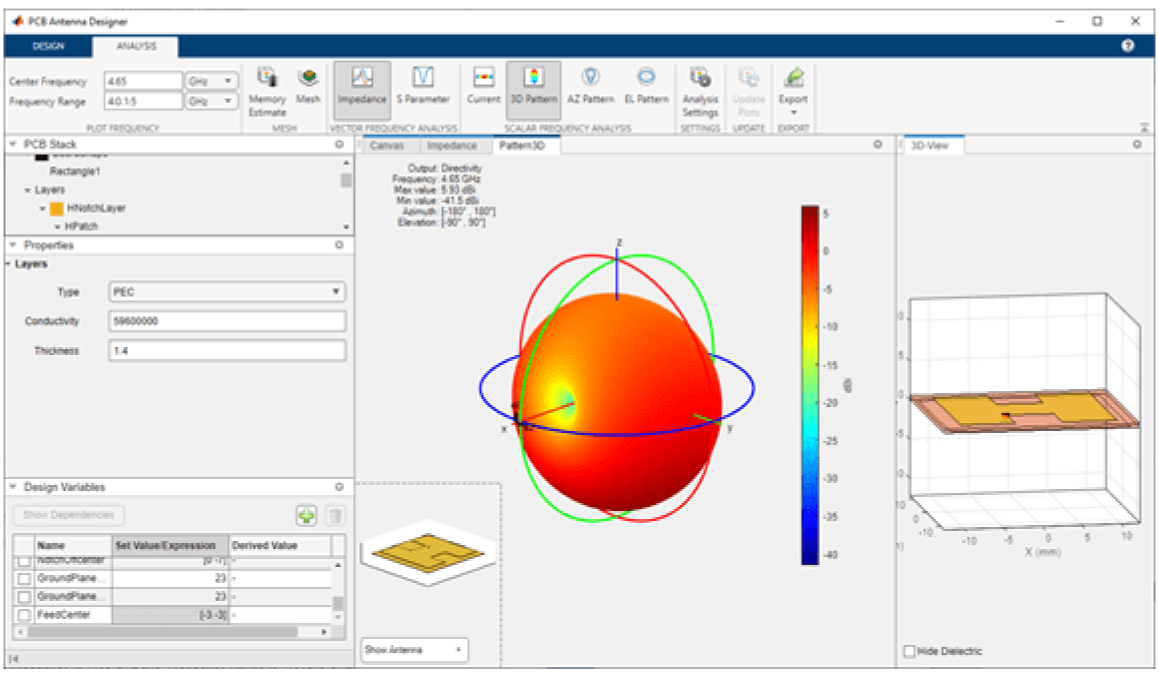
Select **Impedance** from the **Vector Frequency Analysis** section on the toolbar to plot the impedance plot. The impedance plot shows that the PCB stack antenna resonates at 4.6 GHz.



Select **S Parameter** from the **Vector Frequency Analysis** section on the toolbar to plot the S11.



Click **3D Pattern** from the **Scalar Frequency Analysis** section on the toolbar to plot the 3-D far-field radiation pattern of the antenna. The directivity of the H-notch patch unit element is 5.93 dBi.



Setup Optimization Problem

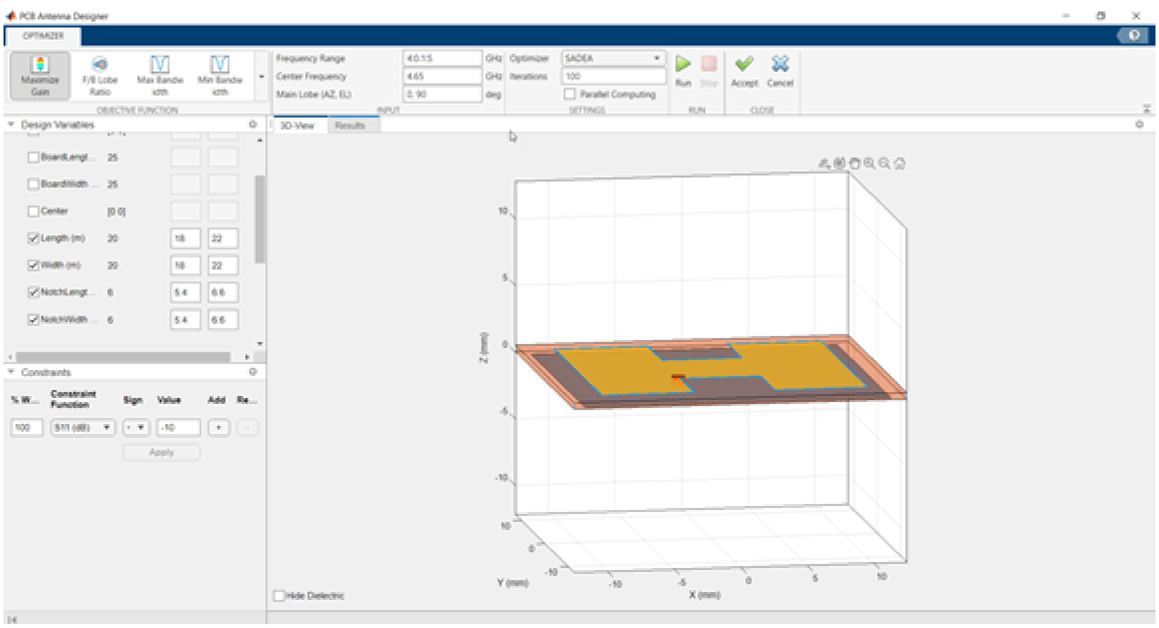
Any optimization problem typically requires following inputs.

1 Objective function: Main goal of the optimization. It evaluates the analysis function and minimizes or maximizes the output of the function. **In this tutorial, maximizing the gain of the antenna is the objective function.**

2 Design variables: The input variables to the objective function. These variables are changed by the optimizer within a pre-set range of values called as the bounds of the variables. **In this tutorial, the dimensions of the H-Notch patch are the design variables.**
3 Constraint functions (if necessary): Functions which restrict a desired analysis function value on the antenna. **In this tutorial, the constraint function is S11 less than -10 db to obtain an impedance bandwidth.**
4 Other inputs: Other inputs may include the number of iterations, the input center frequency, and the input frequency, the number of iterations, the frequency at which the analysis is performed, etc.

Optimization Function, Design Variables, and Constraints

To optimize the H-Notch patch, click on the **Optimize** button. To select an objective function, use the **OBJECTIVE FUNCTION** Gallery drop down. Since the goal is to maximize the gain of the antenna, click on **Maximize Gain**. To set up the design variables, click on the **Design Variables tab**. Click on the checkboxes present on the left-hand side of the properties to choose the required design variables. The optimizer would change these chosen properties to obtain a maximum gain for the antenna. To set up the constraints, click on the Constraints tab and select S11 (dB) from the Constraint Function. Select < operator from sign and enter value as -10.



Click **Apply**.

In the **SETTINGS** section, enter the number of **Iterations** to run for the optimizer. To start the optimization, click the **Run** button.

Optimization

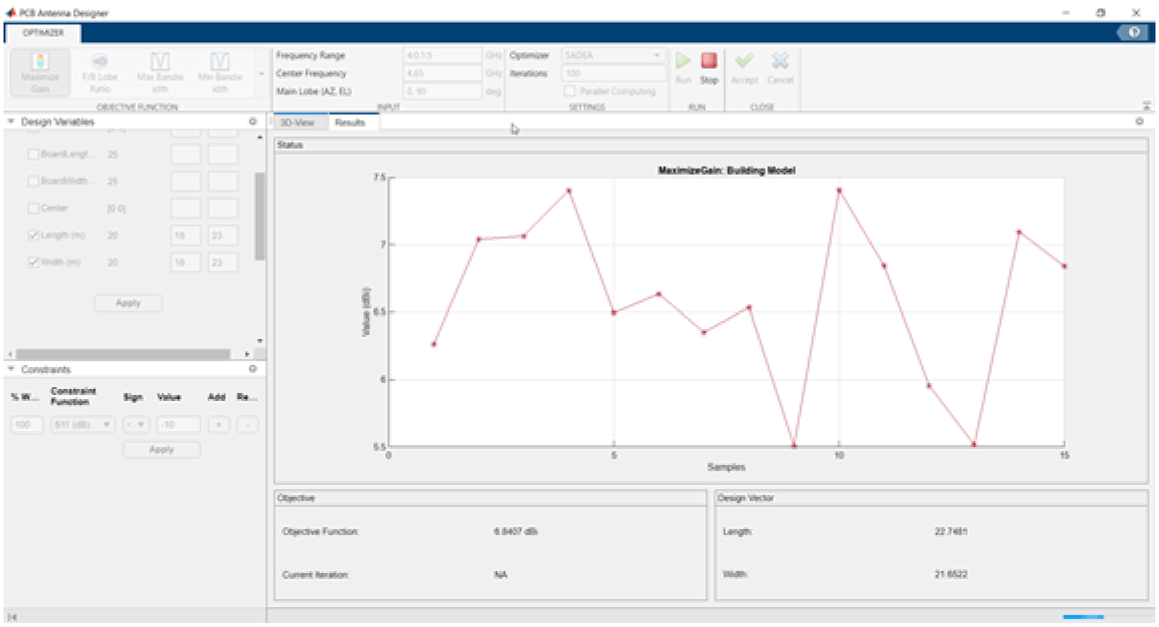
The **SADEA** optimization contains two stages:

1 Building model
2 Optimizing

Building Model

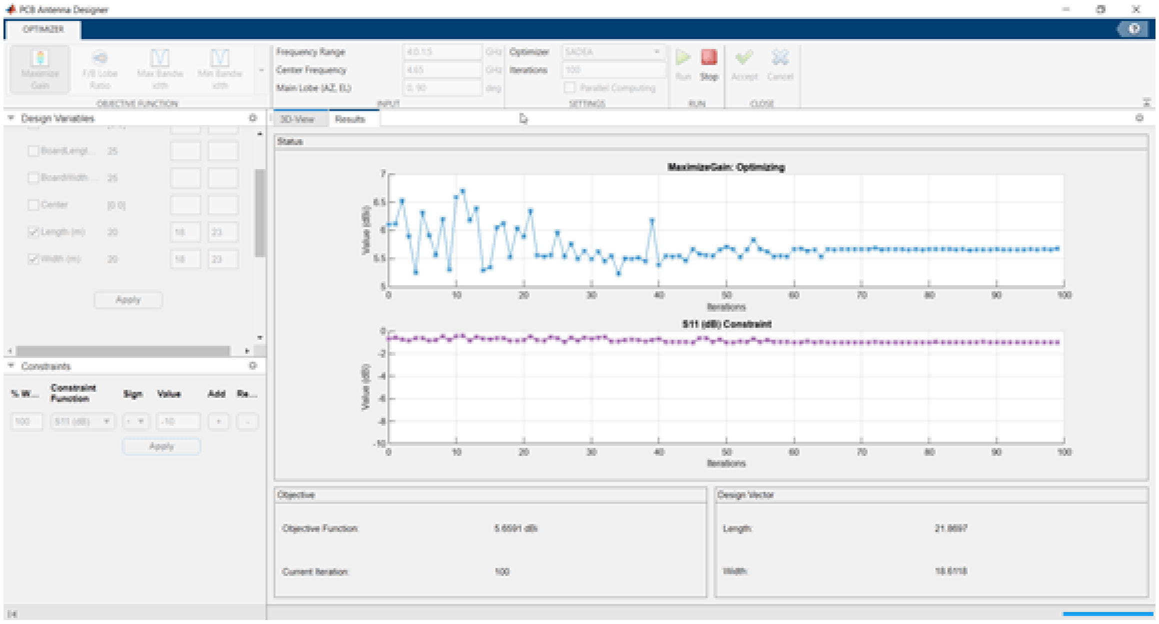
In the model building stage, the optimizer makes a surrogate model from the design space, and the specified objective and the constraints function. It diversely goes through the design space and performs analysis on these sample points.

So, the X-axis shows the number of samples and the Y-axis shows the value of the analysis function value at that sample. The bottom left side show the current sample value and the bottom right side shows the design variables. The optimizer within decides and takes appropriate number of samples to build the model. After the model is built, the optimizer starts running iterations.



Optimizing

In the optimizing stage, the X-axis shows the number of iterations and the Y-axis shows the objective function values. From the plots shown on the optimizing stage, you can understand the trend of convergence.



Once the optimization is complete, click on **Accept**. This takes you back to the **Anaysis** tab. And the dimensions are updated with the optimized values. Run the **Impedance**, **S Parameters**, and **3D Pattern** analysis again to compare optimization results with the original results.

See Also

Objects
`pcbStack`

Related Examples

- "Antenna Optimization Algorithm"
- "Maximizing Gain and Improving Impedance Bandwidth of E-Patch Antenna"

Design Variations On Microstrip Patch Antenna Using PCB Stack

Use the pcbstack to design basic, parasitic, direct-coupled, and CP patch antennas.

Setup parameters.

```
vp  = physconst('lightspeed');
f   = 850e6;
lambda = vp./f;
```

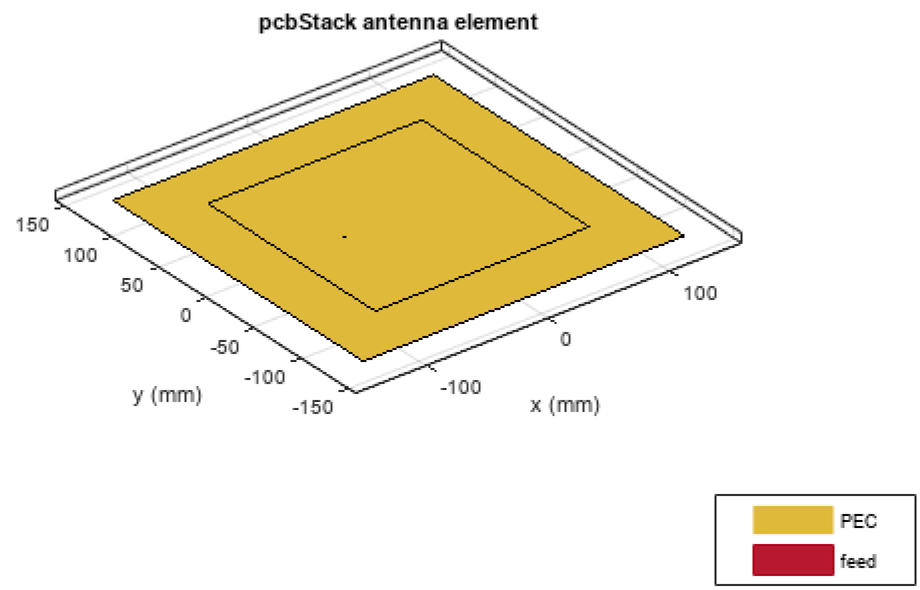
Design Basic Patch Antenna

Set the length and width of the patch and the groundplane.

```
Lp = lambda(1)/2;
Wp = lambda(1)/2;
Lgp = 0.75.*lambda(1);
Wgp = 0.75.*lambda(1);
h = 2.e-3;
p1 = antenna.Rectangle('Length',Lp,'Width',Wp,'NumPoints',30);
p2 = antenna.Rectangle('Length',Lgp,'Width',Wgp);
d1 = dielectric('Air');
```

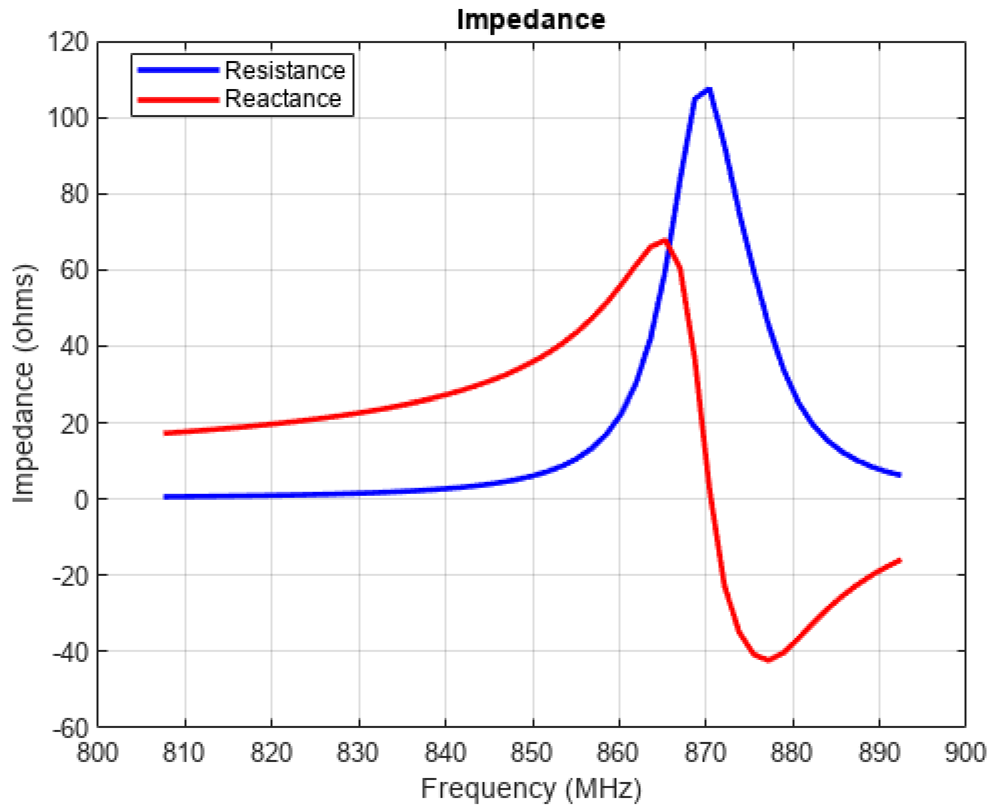
Define the properties of the PCB stack.

```
basicPatch = pcbStack;
basicPatch.Name = 'Basic Patch';
basicPatch.BoardThickness = h;
d1.Thickness = h;
basicPatch.BoardShape = p2;
basicPatch.Layers = {p1,d1,p2};
basicPatch.FeedLocations = [-lambda(1)/8 0 1 3];
figure
show(basicPatch)
```



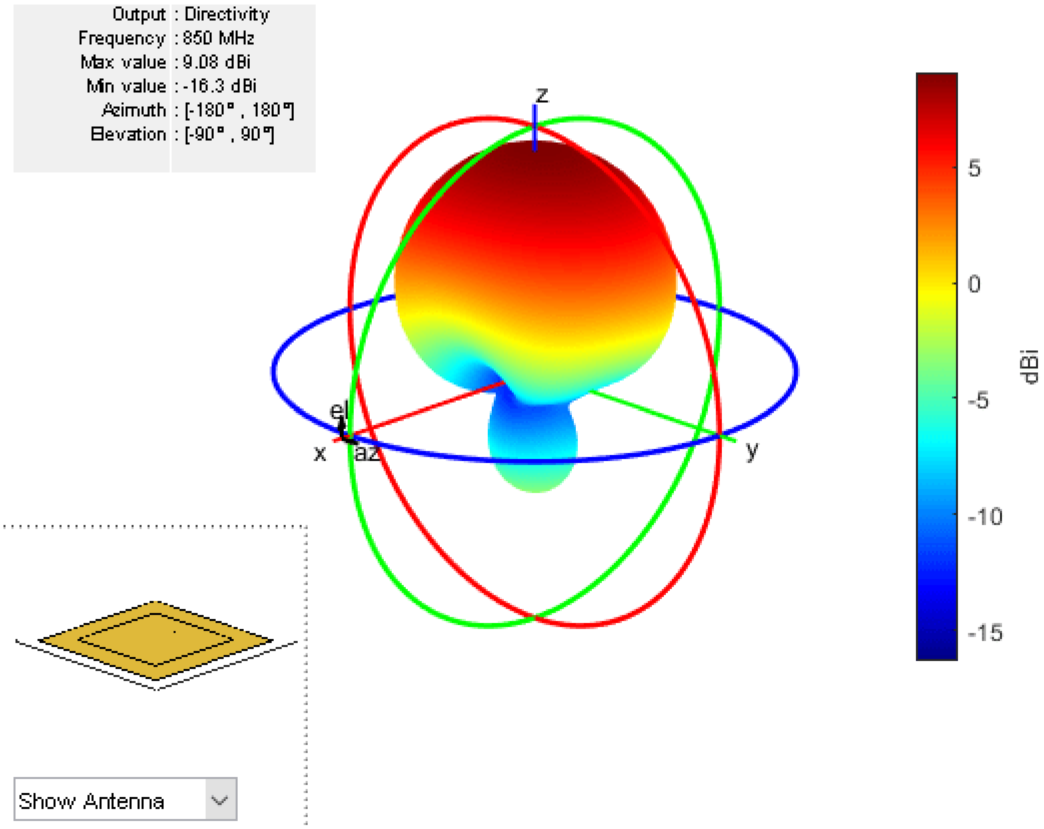
Plot the impedance of the basic patch antenna.

```
freq1 = linspace(f(1)-0.05*f(1),f(1) + 0.05*f(1),51);
figure
impedance(basicPatch,freq1)
```



Plot the radiation pattern of the basic patch antenna.

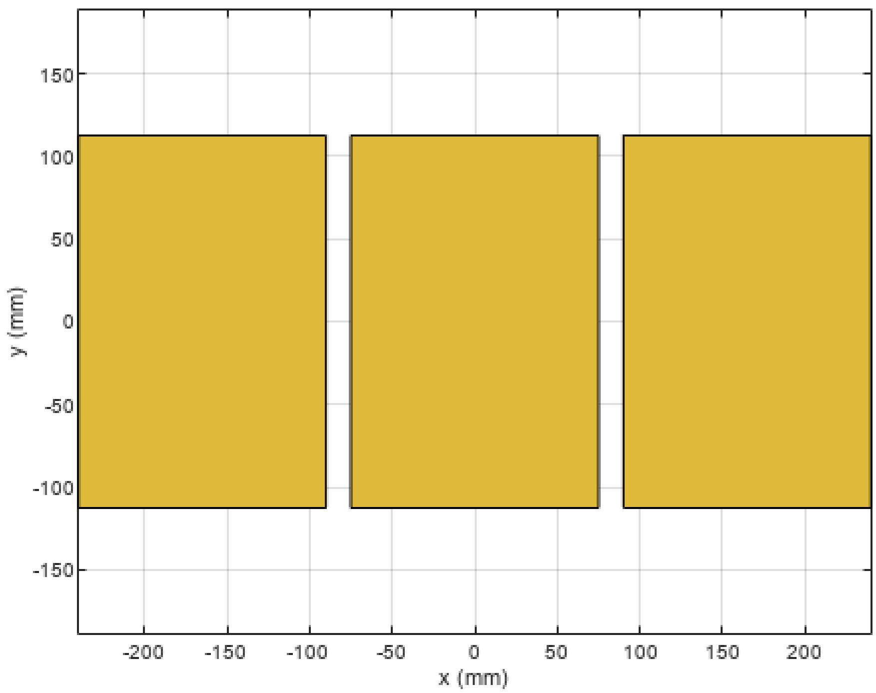
```
figure
pattern(basicPatch,f(1))
```



Design Parasitic Patch Antenna

Set the dimensions for the patch.

```
L = 0.15;
W = 1.5*L;
stripL = L;
gapx = .015;
gapy = .01;
r1 = antenna.Rectangle('Center',[0,0],'Length',L,'Width',W);
r2 = antenna.Rectangle('Center',[L/2+stripL/2+gapx,0],'Length',stripL,'Width',W,'NumPoints',[2 2
r3 = antenna.Rectangle('Center',[-L/2-stripL/2-gapx,0],'Length',stripL,'Width',W,'NumPoints',[2 2
r = r1+r2+r3;
figure
show(r)
```

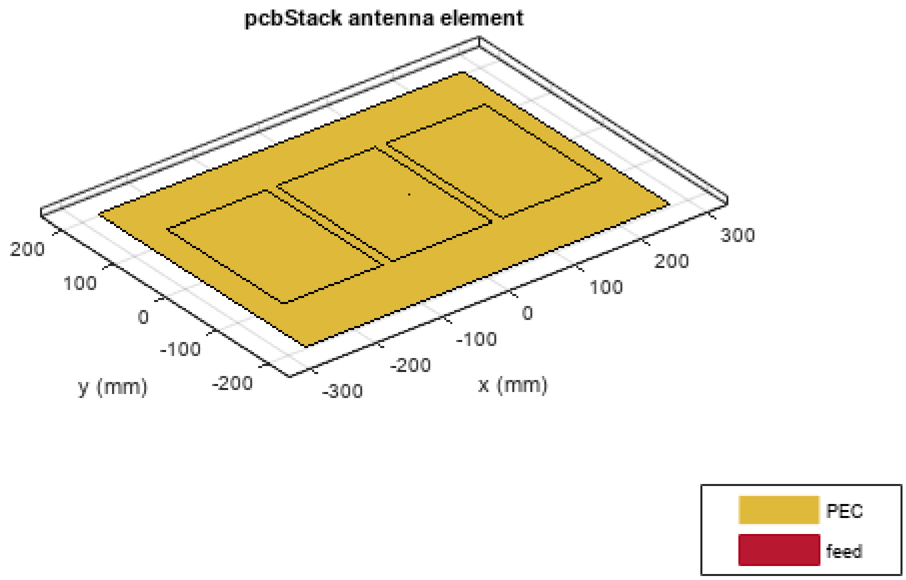


Set the dimensions of the groundplane.

```
Lgp = 0.55;
Wgp = 0.4;
g1  = antenna.Rectangle('Center',[0,0],'Length',Lgp,'Width',Wgp);
```

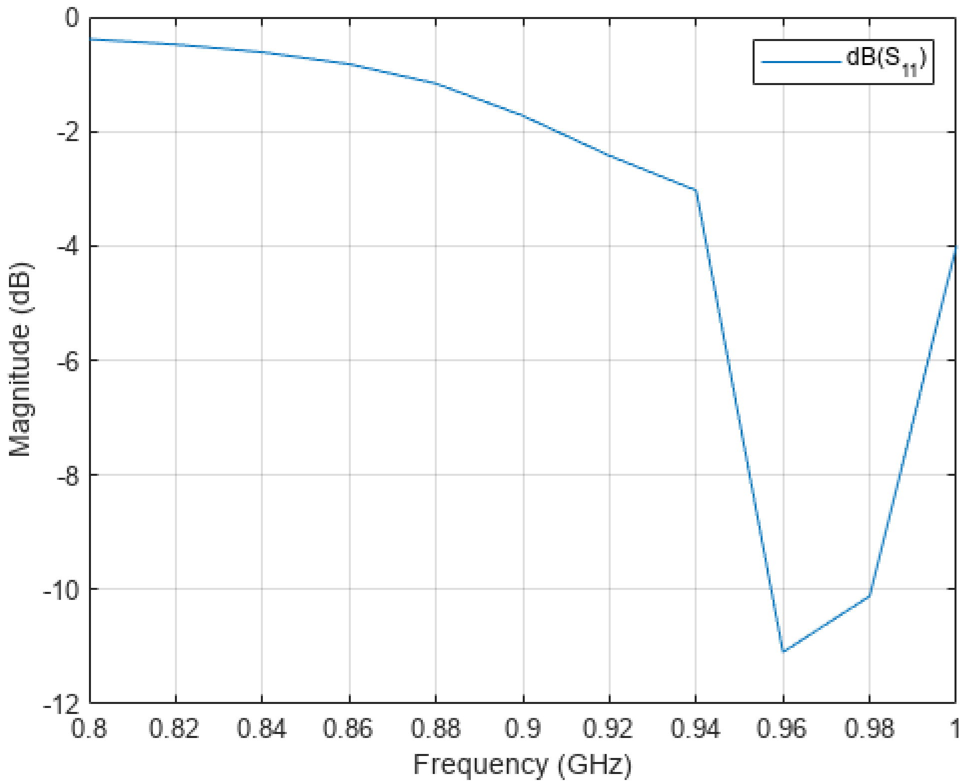
Define the properties of the PCB stack. Create a pcb stack by driving the center radiator.

```
parasitic_patch = pcbStack;
parasitic_patch.BoardShape = g1;
parasitic_patch.BoardThickness = .007;
parasitic_patch.Layers = {r,g1};
parasitic_patch.FeedLocations = [(L)/4 0 1 2];
figure
show(parasitic_patch)
```



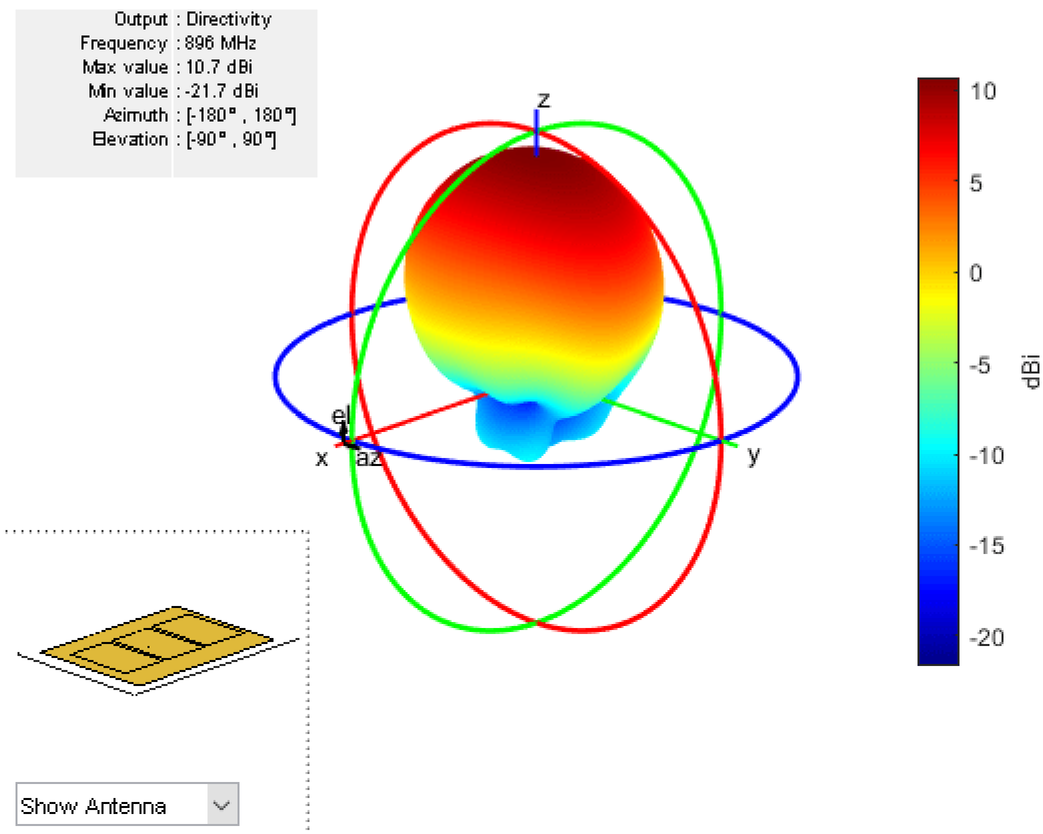
Plot the S-parameters of the parasitic patch antenna.

```
s = sparameters(parasitic_patch,linspace(0.8e9,1e9,11));
figure
rfplot(s)
```



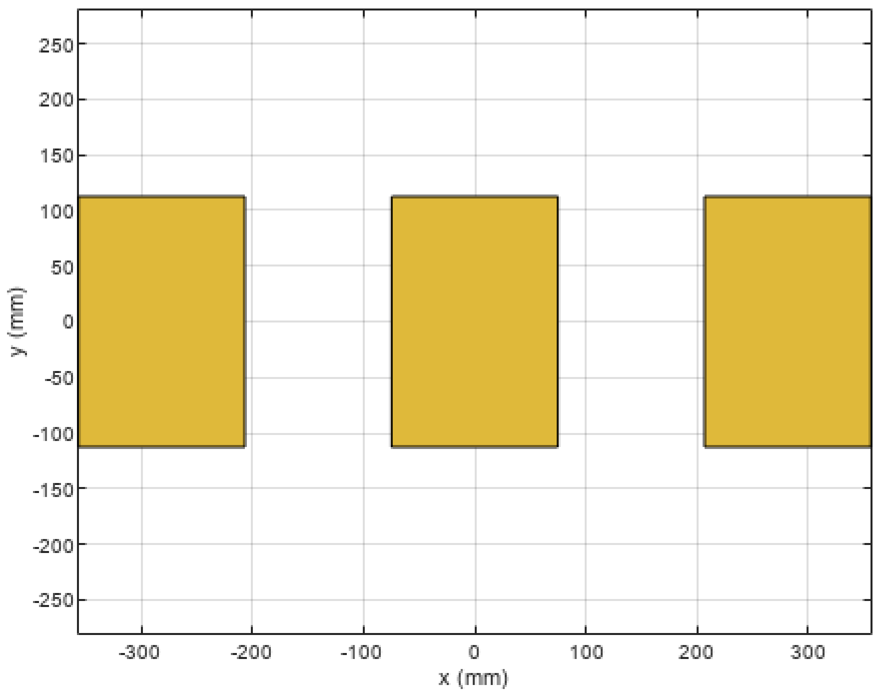
Plot the radiation pattern of the parasitic patch antenna.

```
figure
pattern(parasitic_patch,0.896e9)
```



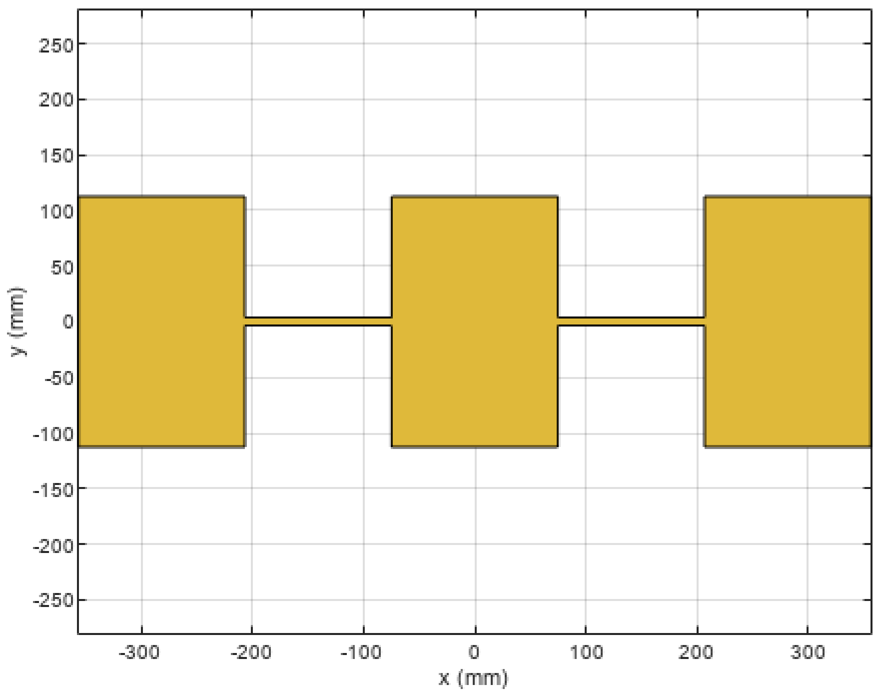
Design Direct-coupled Patch Antenna

```
r2 = copy(r1);
r2.Center = [lambda/1.25,0];
r3 = copy(r1);
r3.Center = [-lambda/1.25,0];
r = r1+r2+r3;
figure
show(r)
```



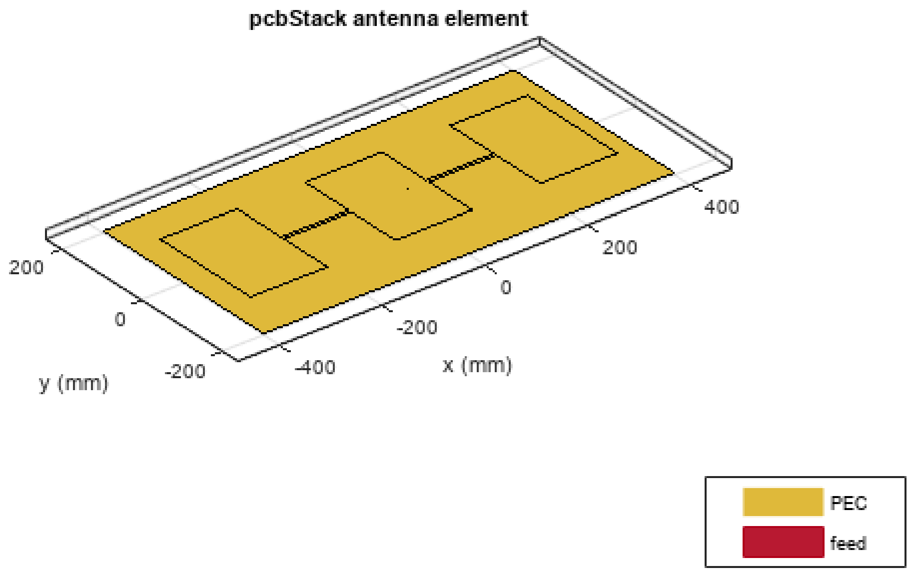
Strip join the sections.

```
r4 = antenna.Rectangle('Length',0.65*lambda,'Width',0.02*lambda,'Center',[lambda/2,0],'NumPoints
r5 = copy(r4);
r5.Center = [-lambda/2,0];
s = r + r4 + r5;
figure
show(s)
```



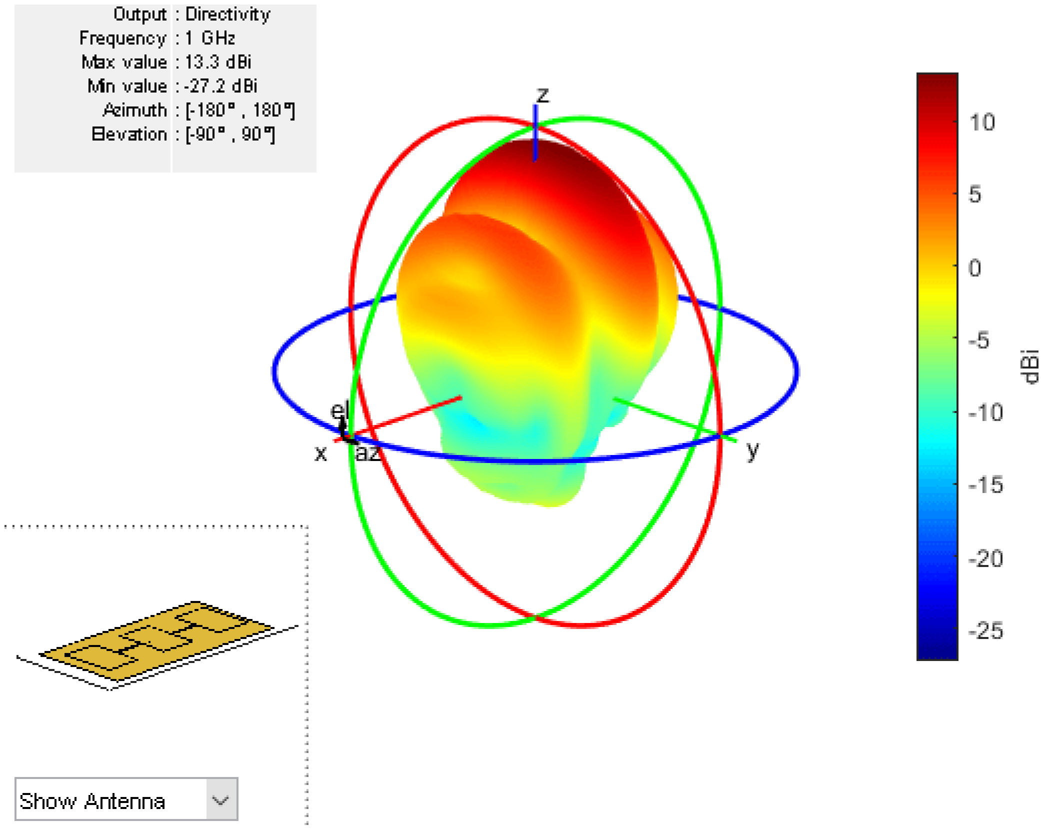
Define the properties of the PCB stack.

```
g1.Length = 0.8;
series_patch = pcbStack;
series_patch.BoardShape = g1;
series_patch.Layers = {s,g1};
series_patch.FeedLocations = [L/4 0 1 2];
figure
show(series_patch)
```



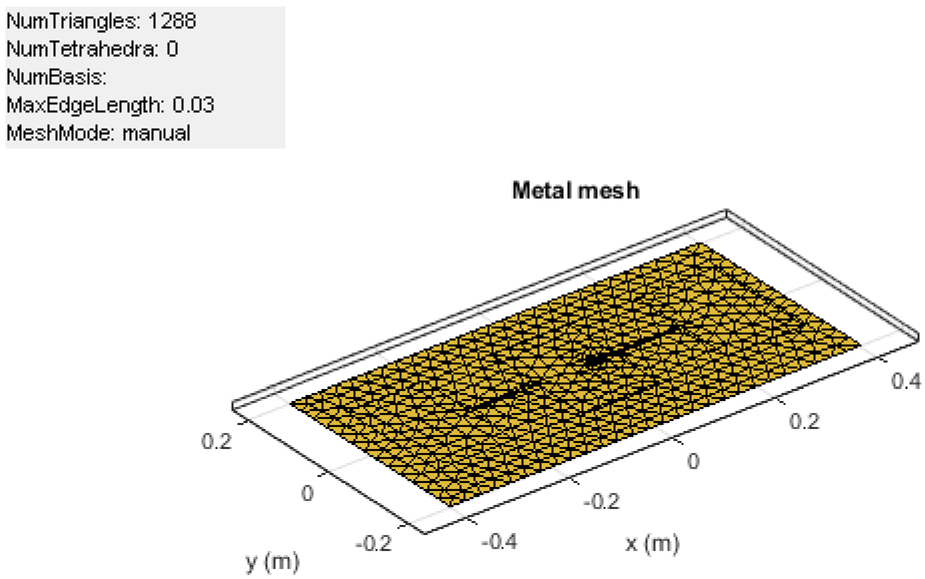
Plot the radiation pattern at 1 GHz for the direct-coupled patch antenna.

```
figure
pattern(series_patch,1e9)
```

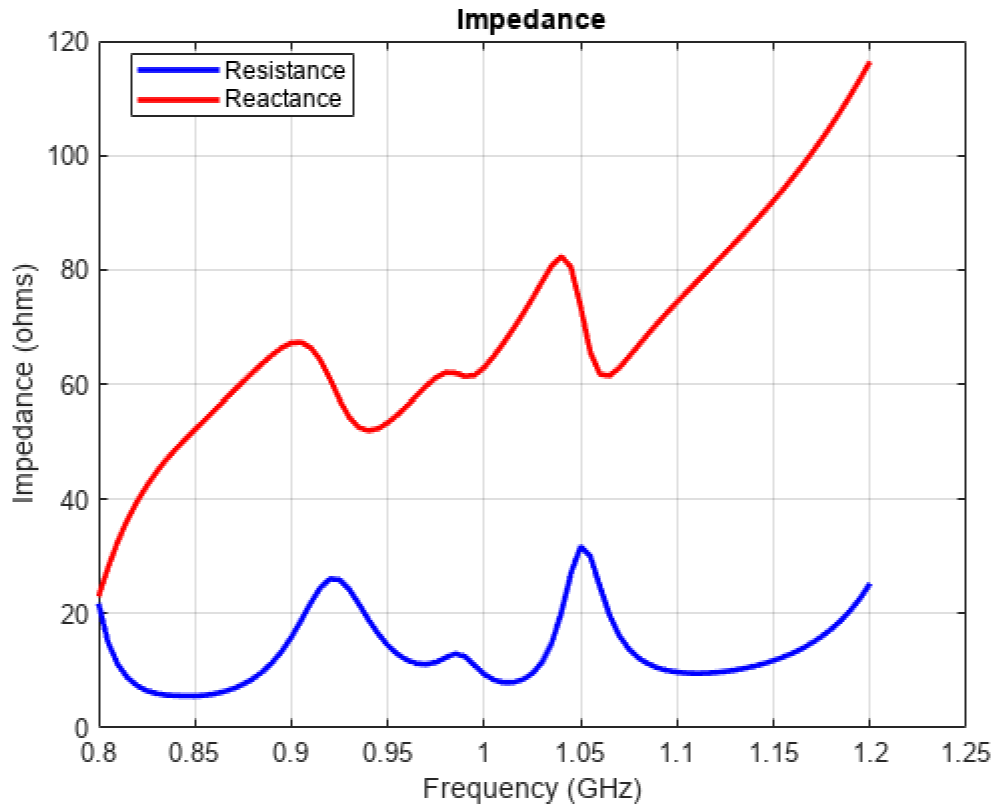


Mesh the antenna using maximum edge length of 0.03 m. Plot the impedance of the direct-coupled patch antenna for the frequency range, 0.8 GHz to 1.2 GHz.

```
figure
mesh(series_patch,'MaxEdgeLength',0.03)
```




```
figure
impedance(series_patch,linspace(0.8e9,1.2e9,81))
```



Design Circularly Polarized Patch - Truncated Corners

Set the length and width of the patch and the groundplane.

```
Lp = lambda(1)/2;
Wp = lambda(1)/2;
Lgp = 0.75.*lambda(1);
Wgp = 0.75.*lambda(1);
h = 2.e-3;
```

Create the base shape for the patch.

```
p1 = antenna.Rectangle('Length',Lp,'Width',Wp,'NumPoints',20);
```

Truncate the corners of the rectangle.

```
Lcorner = 0.25*Lp;
Wcorner = 0.25*Wp;
cornerCenter1 = [-Lp/2,Wp/2,0];
cornerCenter2 = [Lp/2,-Wp/2,0];
pcorner1 = antenna.Rectangle('Length',Lcorner,'Width',Wcorner);
pcorner1 = rotateZ(pcorner1,45);
pcorner1 = translate(pcorner1,cornerCenter1);
pcorner2 = antenna.Rectangle('Length',Lcorner,'Width',Wcorner);
pcorner2 = rotateZ(pcorner2,45);
pcorner2 = translate(pcorner2,cornerCenter2);
pradiator = p1 -pcorner1-pcorner2;
```

Create the groundplane shape.

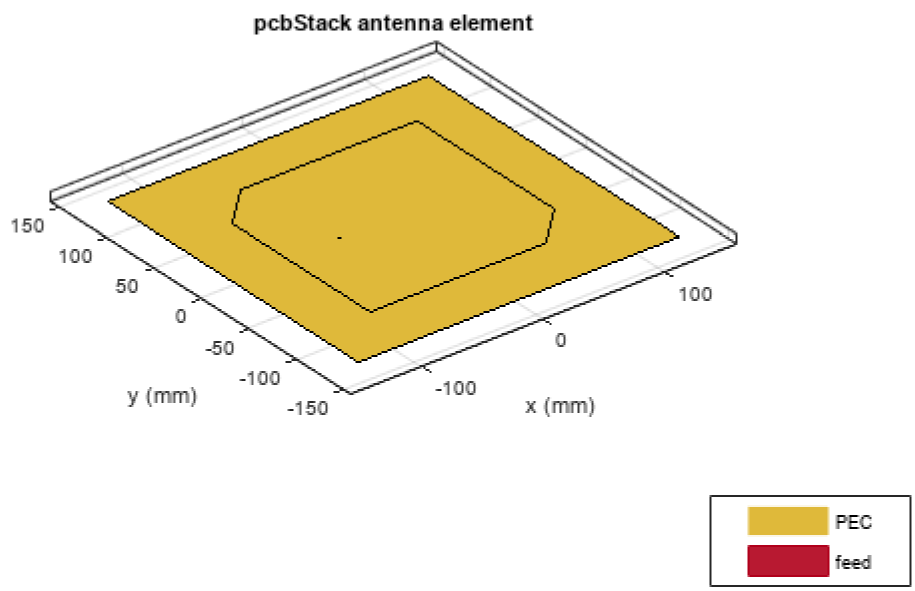
```
p2 = antenna.Rectangle('Length',Lgp,'Width',Wgp);
```

Define the dielectric Layer.

```
d1 = dielectric('Air');
```

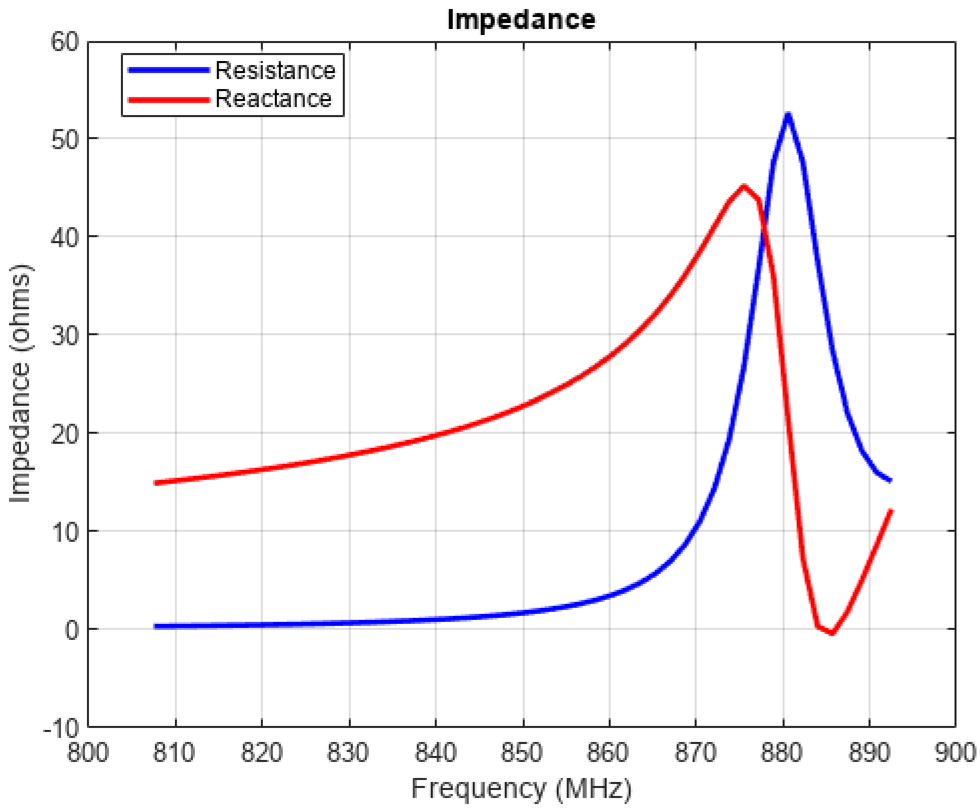
Define the properties of the PCB stack for the circularly polarized patch.

```
truncatedCornerPatch = pcbStack;
truncatedCornerPatch.Name = 'Basic Patch';
truncatedCornerPatch.BoardThickness = h;
d1.Thickness = h;
truncatedCornerPatch.BoardShape = p2;
truncatedCornerPatch.Layers = {pradiator,d1,p2};
truncatedCornerPatch.FeedLocations = [-lambda(1)/8 0 1 3];
figure
show(truncatedCornerPatch)
```



Plot impedance of the circularly polarized antenna.

```
figure
impedance(truncatedCornerPatch,freq1)
```



See Also

"Antenna Modeling and Analysis" on page 1-3

Introduction to Arrays

Array Modeling and Analysis

This example shows how to construct, visualize, and analyze an antenna array from the Antenna Toolbox.

Create Antenna Array Using Antenna Elements

Create a default rectangular antenna array using the `rectangularArray` element in the array library. By default, the array uses the dipole as an antenna element.

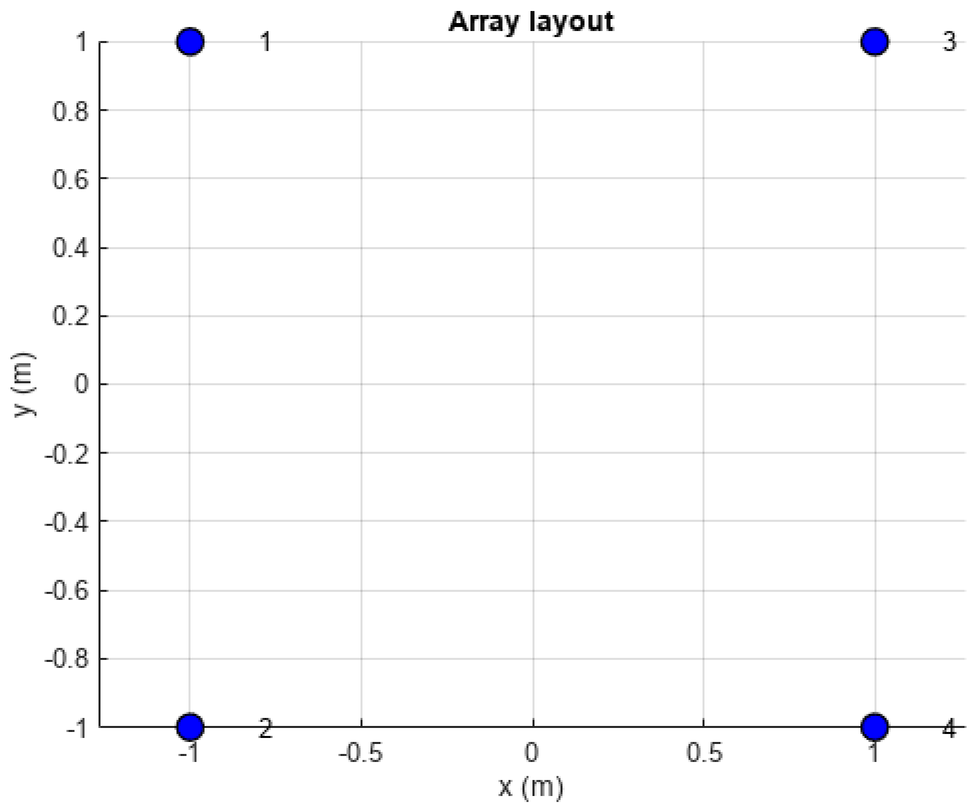
```
ra = rectangularArray
```

```
ra = 
  rectangularArray with properties:

           Element: [1x1 dipole]
              Size: [2 2]
        RowSpacing: 2
     ColumnSpacing: 2
           Lattice: 'Rectangular'
    AmplitudeTaper: 1
        PhaseShift: 0
              Tilt: 0
          TiltAxis: [1 0 0]
```

Visualize Layout of Array

Use the `layout` function to plot the position of array elements in the x-y plane. By default, the rectangular array is a 4-element dipole array in a 2x2 rectangular lattice.

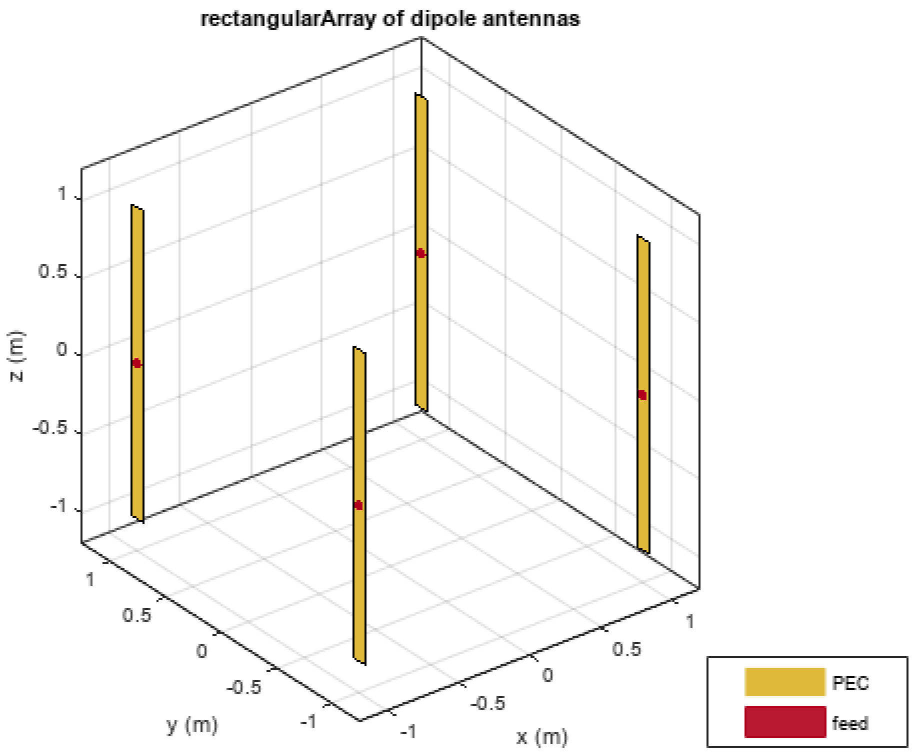
```
layout(ra)
```



Visualize Geometry of Array

Use the `show` function to view the structure of the rectangular antenna array.

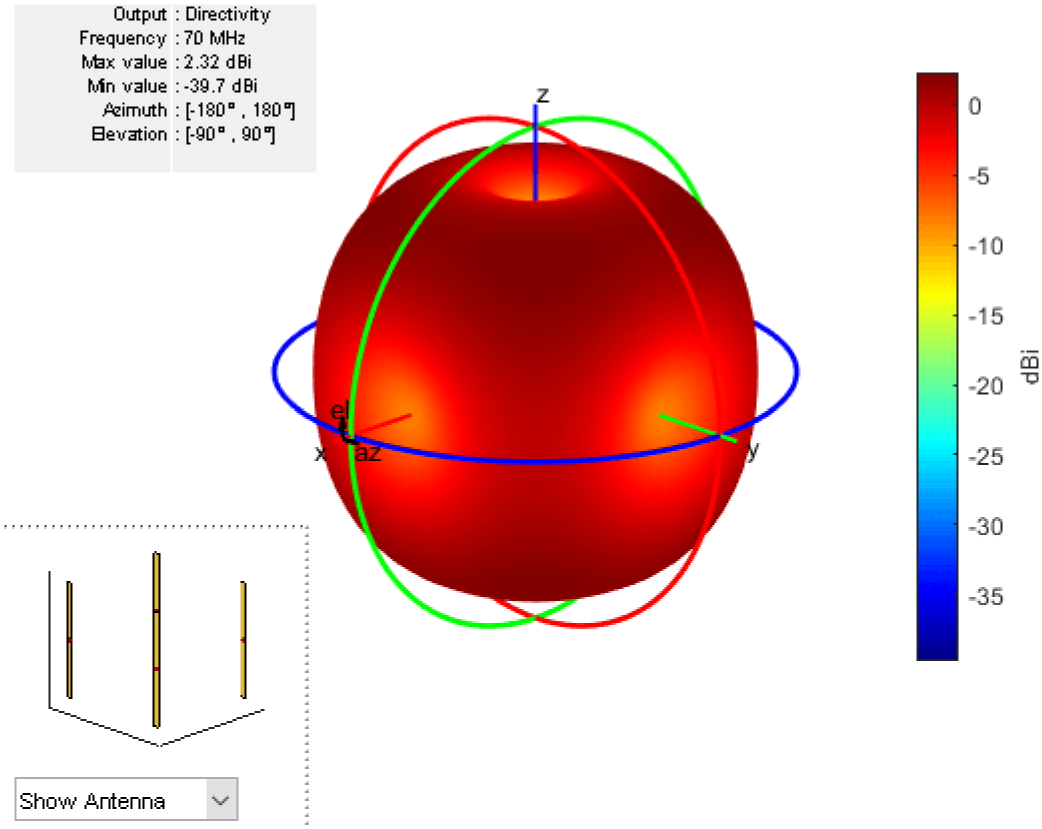
```
show(ra)
```



Plot Radiation Pattern of Array

Use the `pattern` function to plot the radiation pattern of the rectangular array. The radiation pattern is the spatial distribution of the power of an array. The pattern displays the directivity or gain of the array. By default, the pattern function plots the directivity of the array.

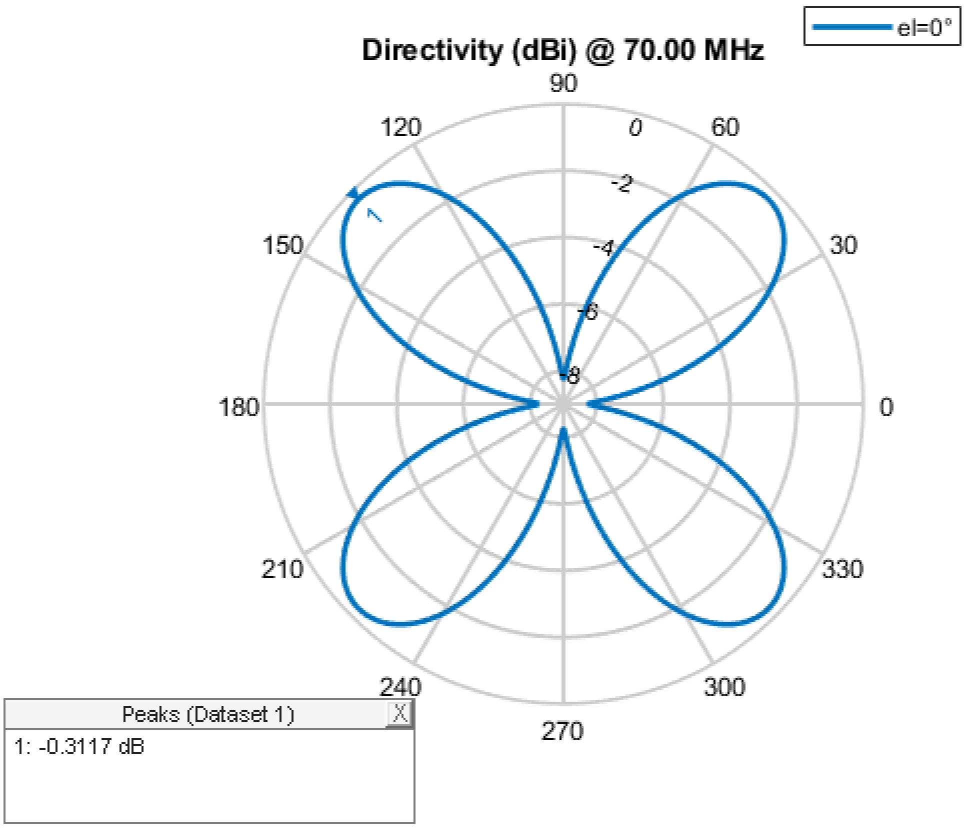
```
pattern(ra,70e6)
```



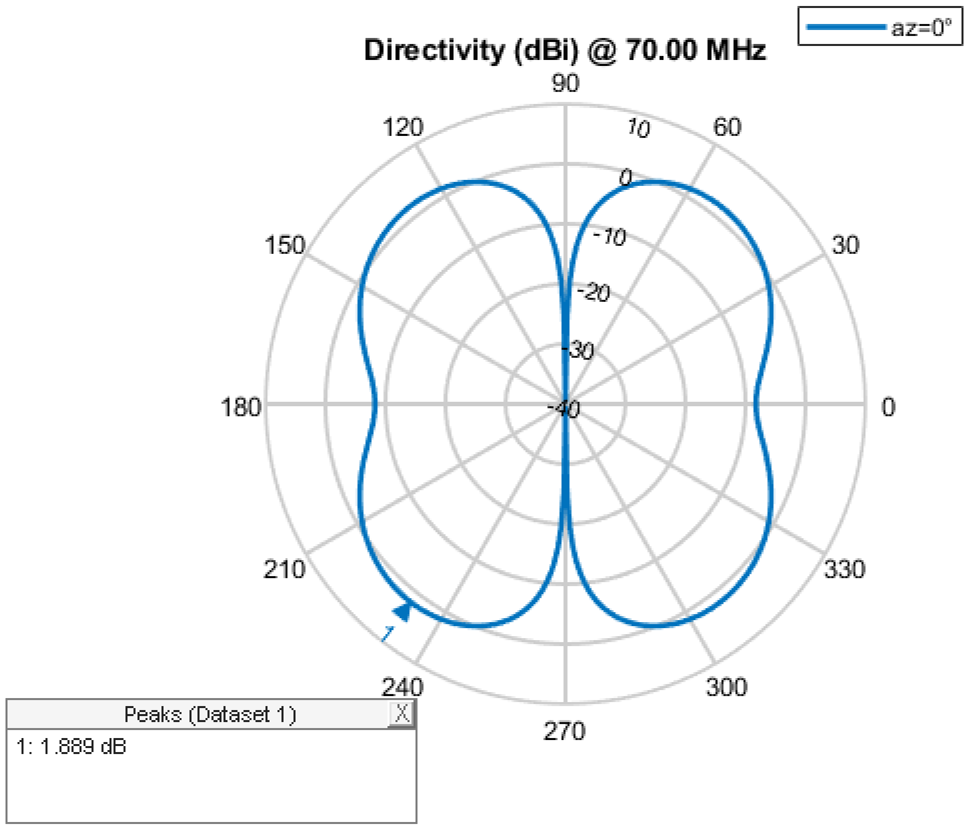
Plot Azimuth and Elevation Pattern of Array

Use `patternAzimuth` and `patternElevation` functions to plot the azimuth and elevation pattern of the rectangular array. These two patterns are the 2D radiation pattern of the array at a specified frequency.

```
patternAzimuth(ra,70e6)
```




```
figure
patternElevation(ra,70e6)
```



Calculate the Directivity of Array

Directivity is the ability of an array to radiate power in a particular direction. It can be defined as the ratio of the maximum radiation intensity in the desired direction to the average radiation intensity in all other directions. Use the `pattern` function to calculate the directivity of the rectangular array.

```
[Directivity] = pattern(ra,70e6,0,90)
```

```
Directivity = -39.7368
```

Calculate EH Fields of Array

Use the `EHfields` function to calculate the EH fields of the rectangular array. EH fields are the x, y, and z components of the electric and magnetic fields of an array. These components are measured at a specific frequency and at specified points in space.

```
[E,H] = EHfields(ra,70e6,[0;0;1])
```

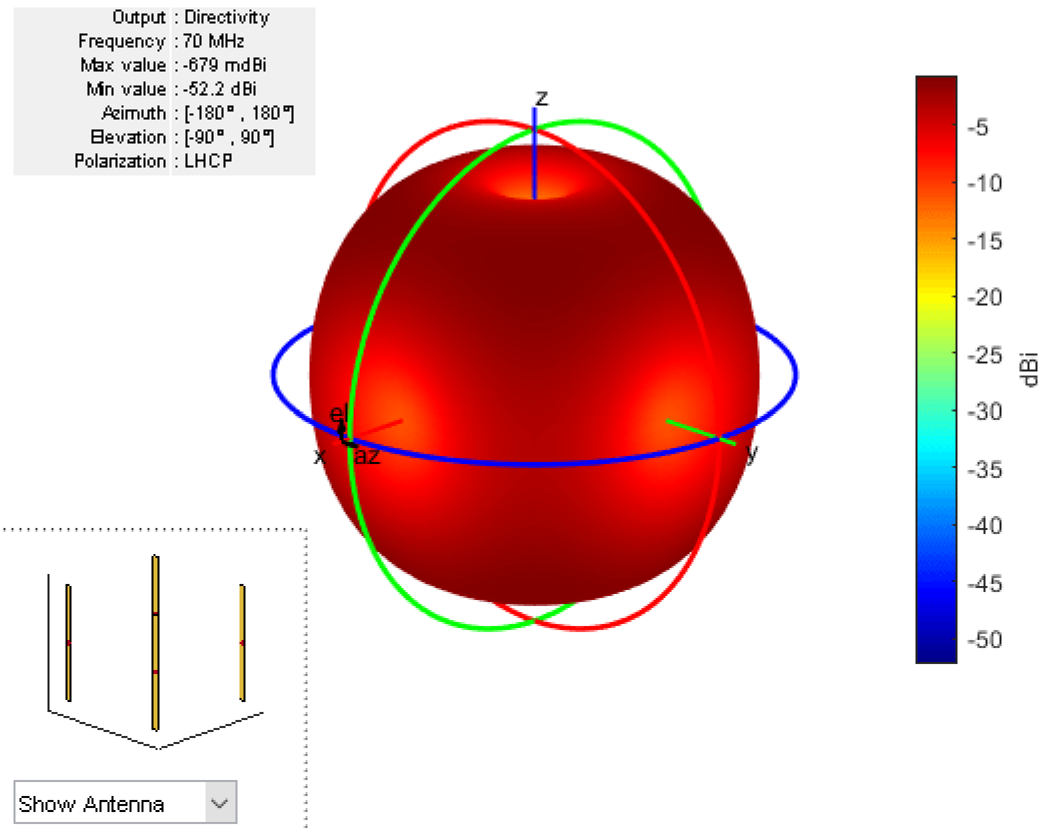
```
E = 3×1 complex

  -0.0000 - 0.0000i
   0.0003 + 0.0011i
  -1.3433 - 0.0695i


H = 3×1 complex
10^-5 ×
```

```
   0.0598 - 0.1672i
  -0.0000 - 0.0000i
   0.0000 - 0.0000i
```

Plot Different Polarizations of Array

Use the Polarization name-value pair in the pattern function to plot the different polarization patterns of the rectangular array. Polarization is the orientation of the electric field, or E-field, of an array. Polarization is classified as elliptical, linear, or circular. This example shows the left-hand circularly polarized (LHCP) radiation pattern of the rectangular array.

```
pattern(ra,70e6,'Polarization','LHCP')
```



Calculate Beamwidth of Array

Use the `beamwidth` function to calculate the beamwidth of the rectangular array. The beamwidth of an array is the angular measure of the array pattern coverage. The beamwidth angle is measured in the plane that contains the direction of main lobe of the array.

```
[bw,angles] = beamwidth(ra,70e6,0,1:1:360)
```

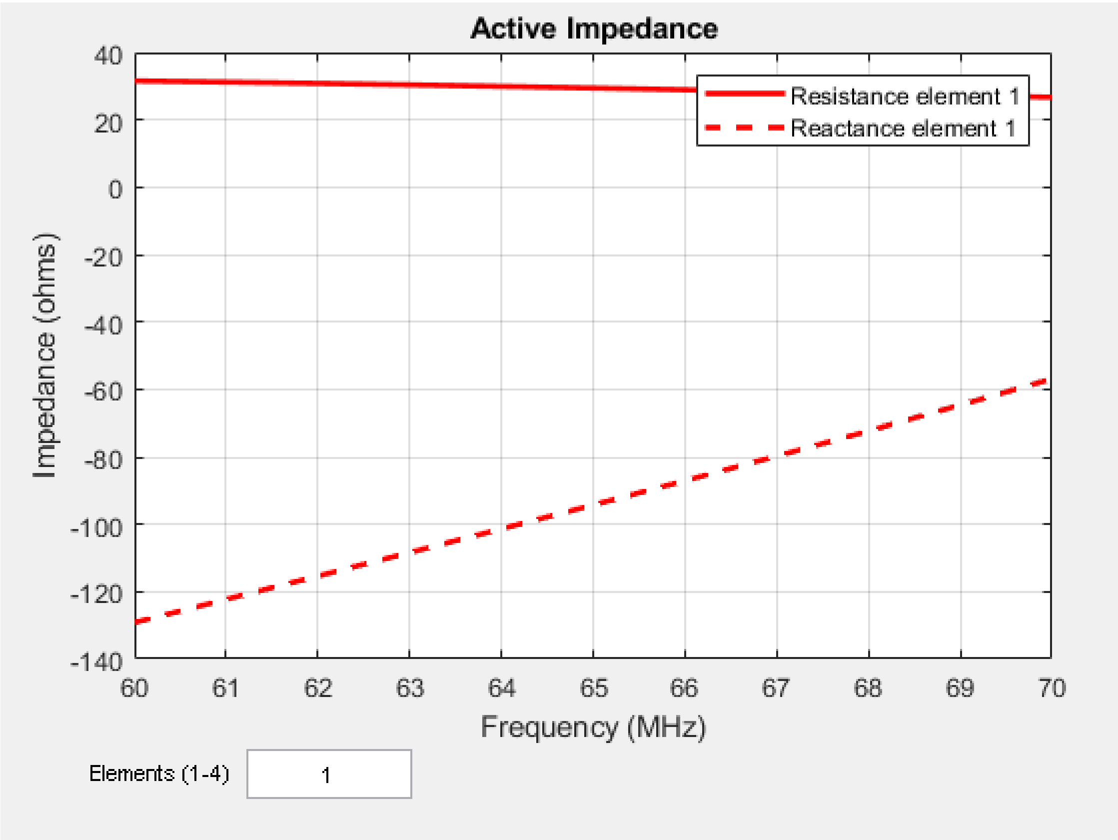
```
bw = 4×1

   44.0000
   44.0000
   44.0000
```

```
   44.0000


angles = 4×2

    28    72
   108   152
   208   252
   288   332
```

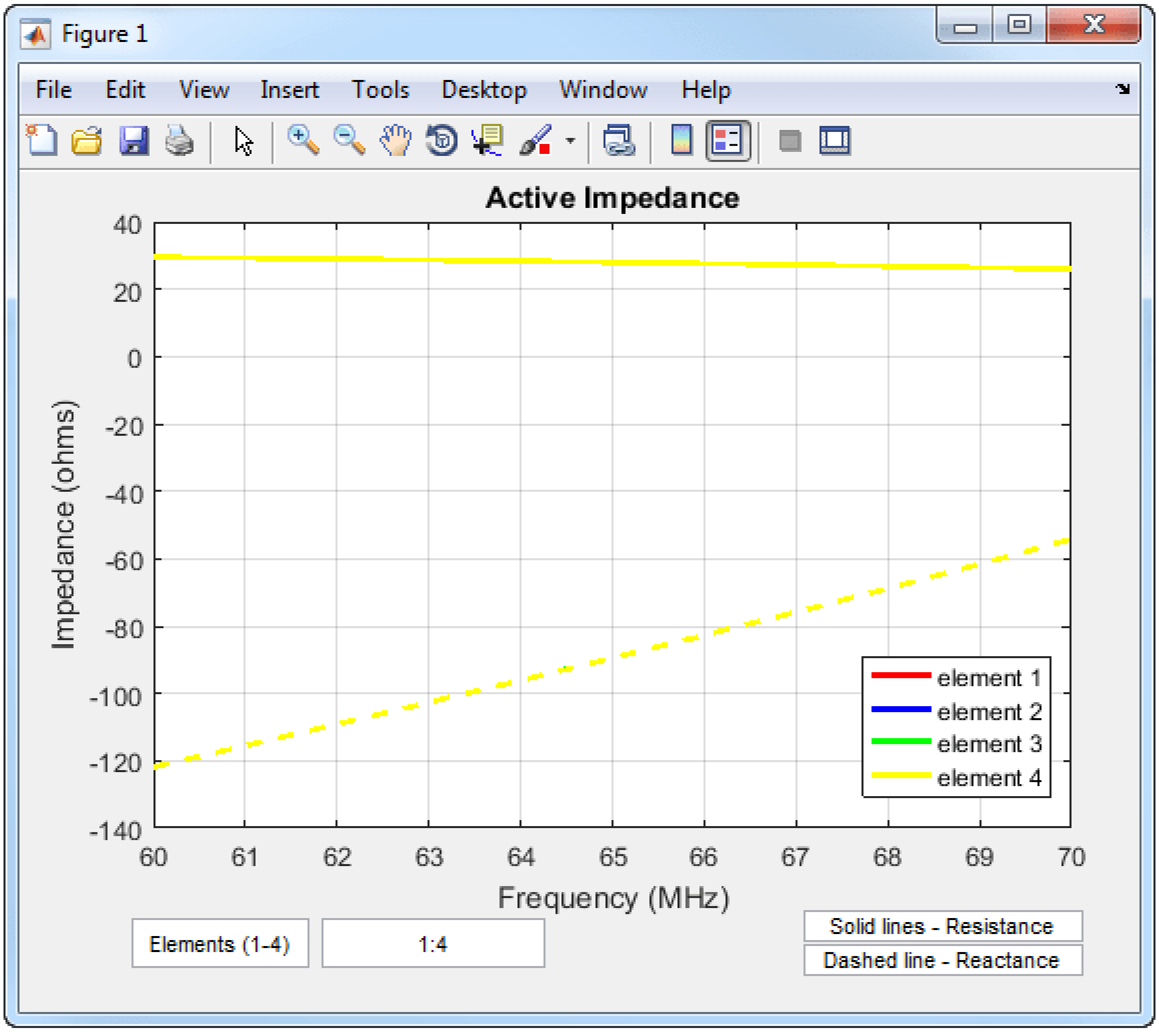
Calculate Scan Impedance of Array

Use the `impedance` function to calculate and plot the input impedance of rectangular array. Active impedance, or scan impedance, is the input impedance of each antenna element in an array, when all elements are excited.

```
impedance(ra,60e6:1e6:70e6)
```



You can also view the impedance of all four elements by changing the number of elements on the plot from 1 to 1:4. See figure.



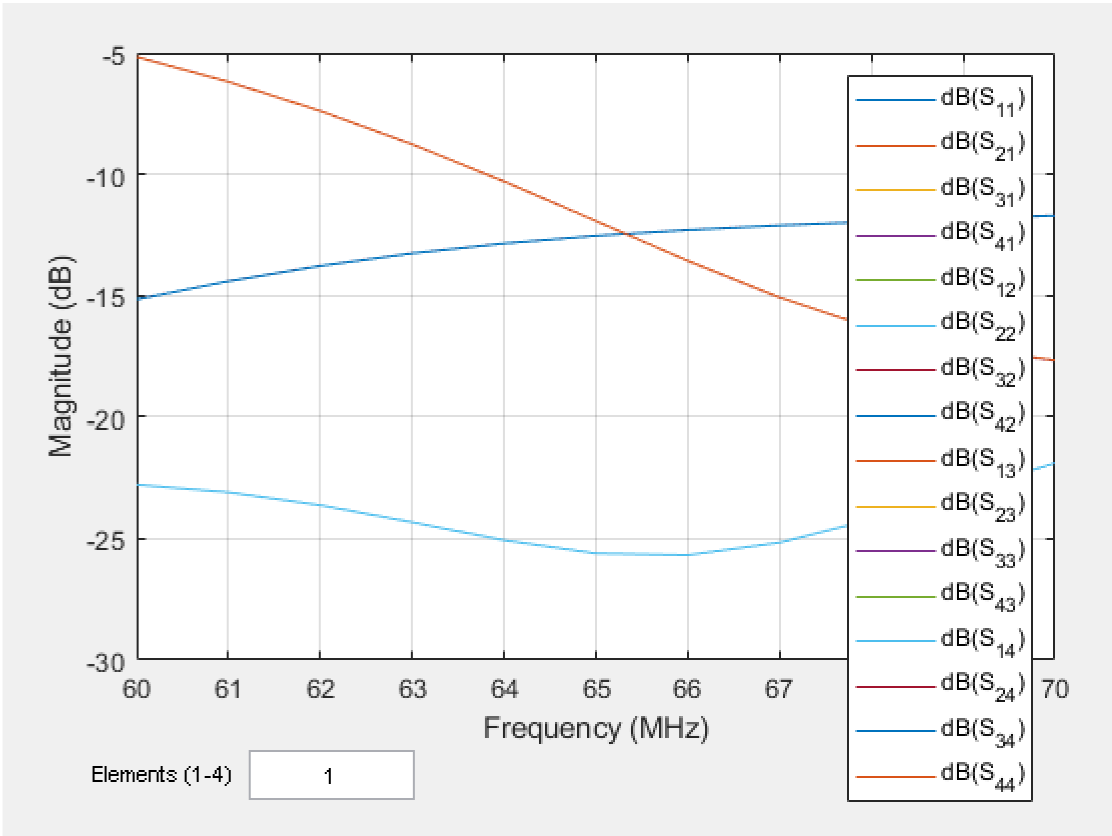
Calculate Reflection Coefficient of Array

Use the `sparameters` function to calculate the S11 value of the rectangular array. S11 value gives the reflection coefficient of the array.

```
S = sparameters(ra,60e6:1e6:70e6,72)
```

```
S =
  sparameters: S-parameters object

       NumPorts: 4
    Frequencies: [11x1 double]
     Parameters: [4x4x11 double]
      Impedance: 72

  rfparam(obj,i,j) returns S-parameter Sij
```

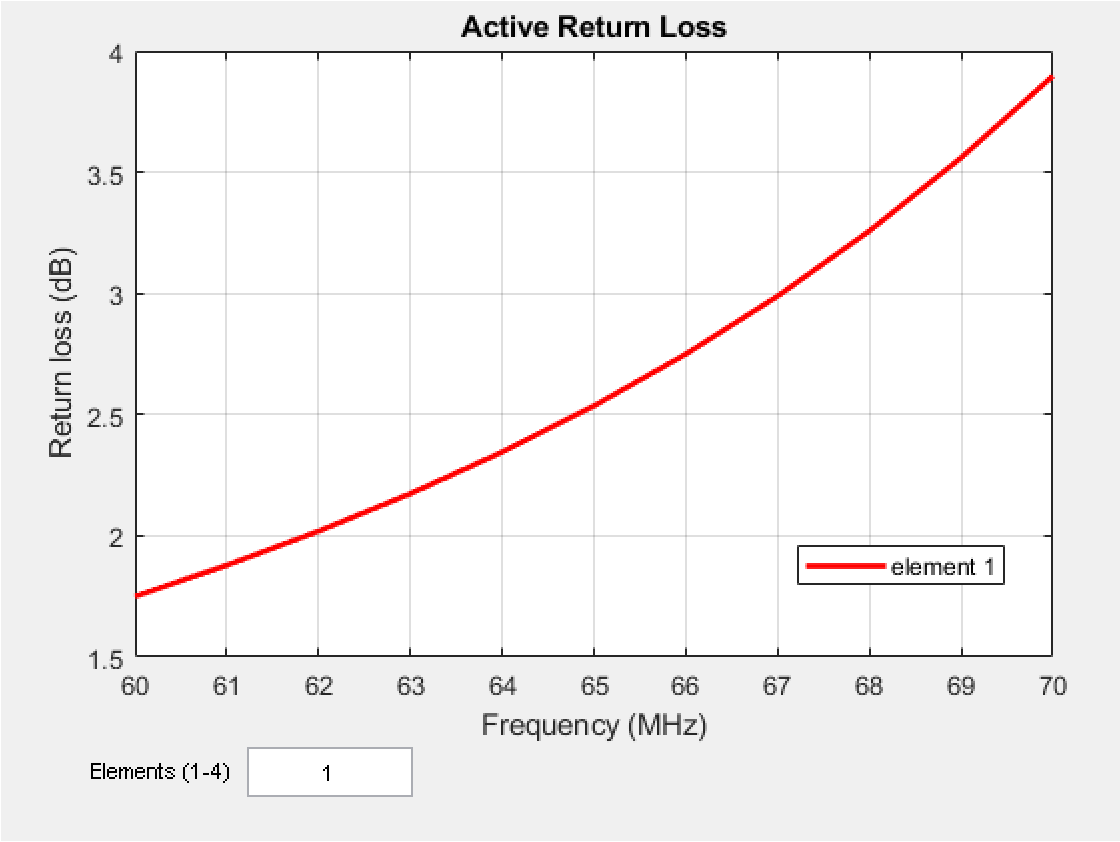
```
rfplot(S)
```



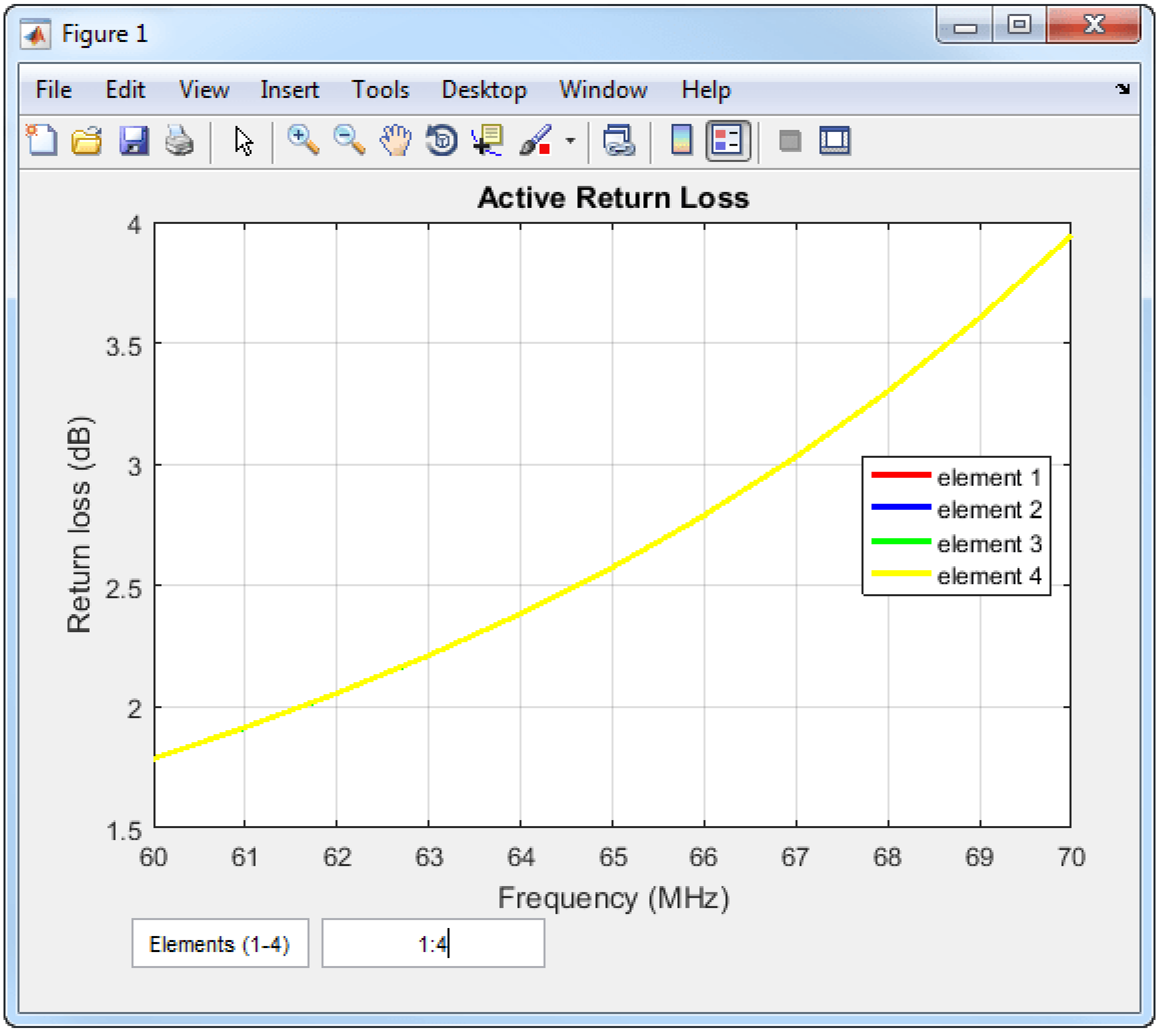
Calculate Return Loss of Array

Use the `returnLoss` function to calculate and plot the return loss of the rectangular array.

```
returnLoss(ra,60e6:1e6:70e6,72)
```



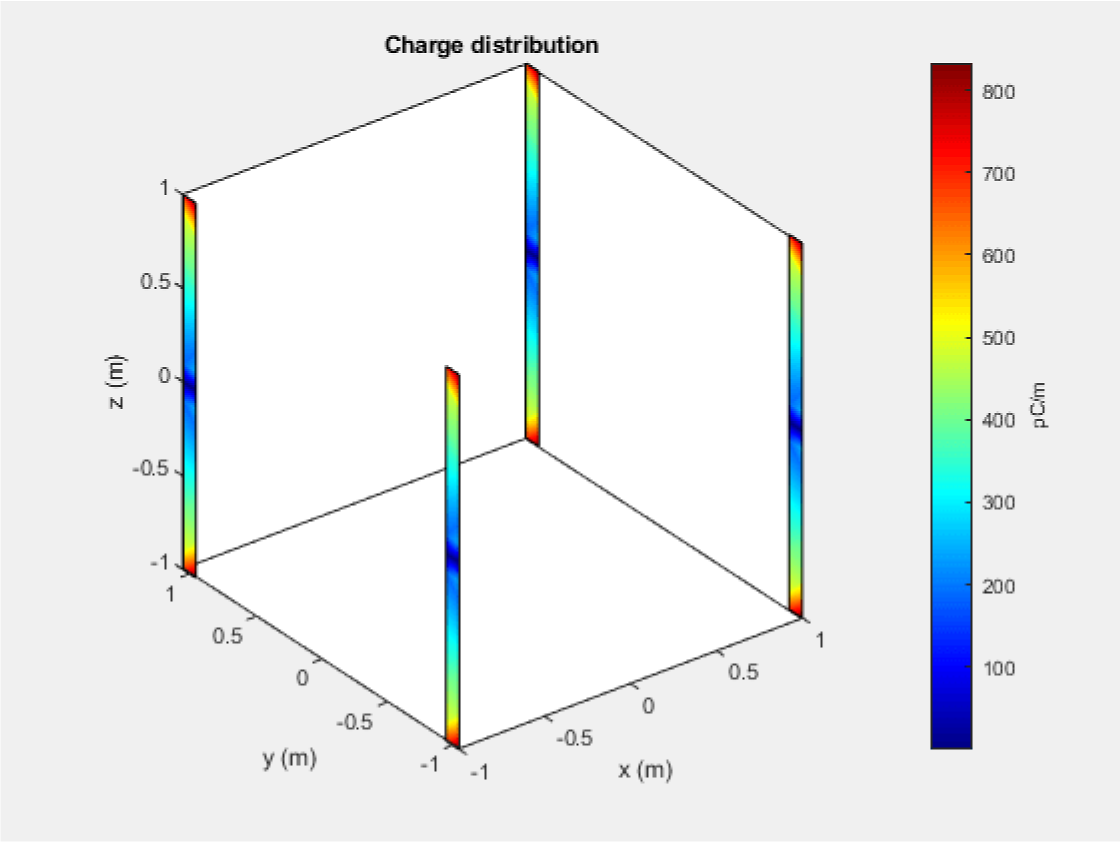
You can also view the return loss of all four elements by changing the number of elements on the plot from 1 to 1:4. See figure.



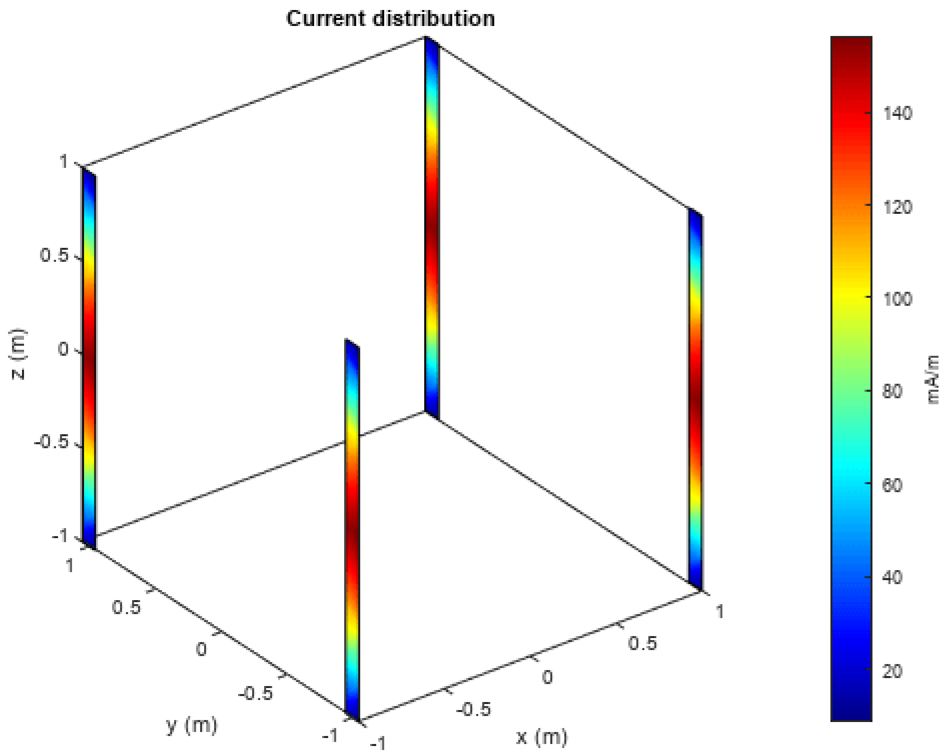
Calculate Charge and Current Distribution Of Array

Use the `charge` and `current` functions to calculate the charge and current distribution on the rectangular array surface.

```
charge(ra,70e6)
```



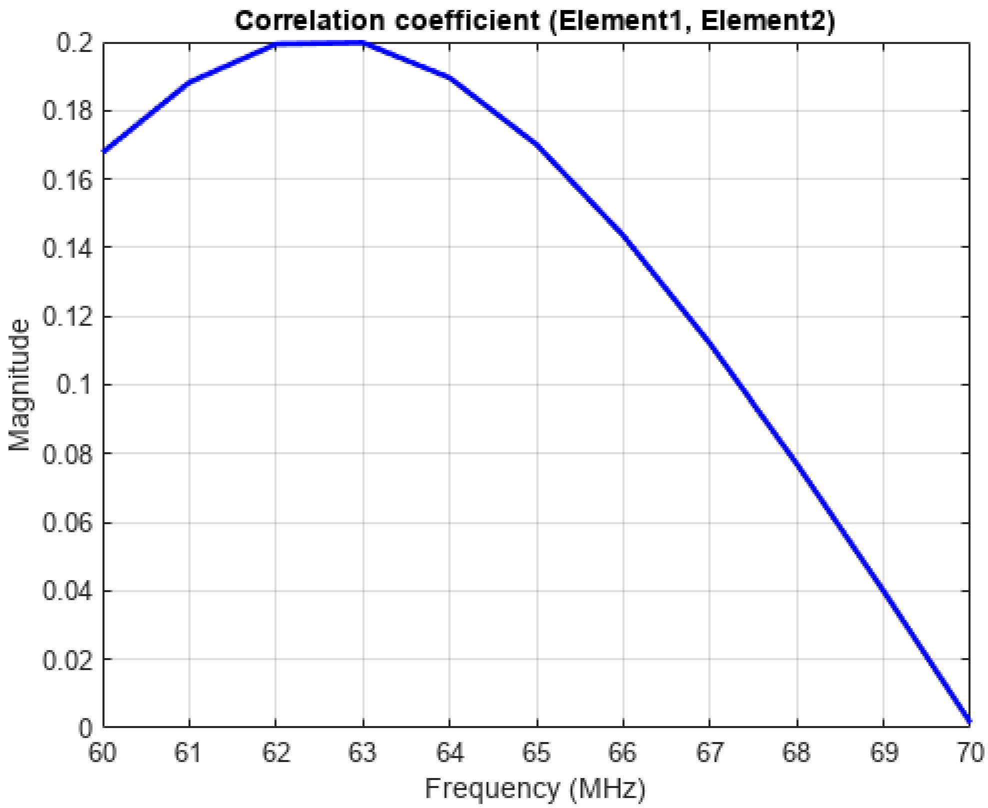

```
figure
current(ra,70e6)
```



Calculate Correlation Coefficient of Array

Use the `correlation` to calculate the correlation coefficient of the rectangular array. The correlation coefficient is the relationship between the incoming signals at the antenna ports in an array.

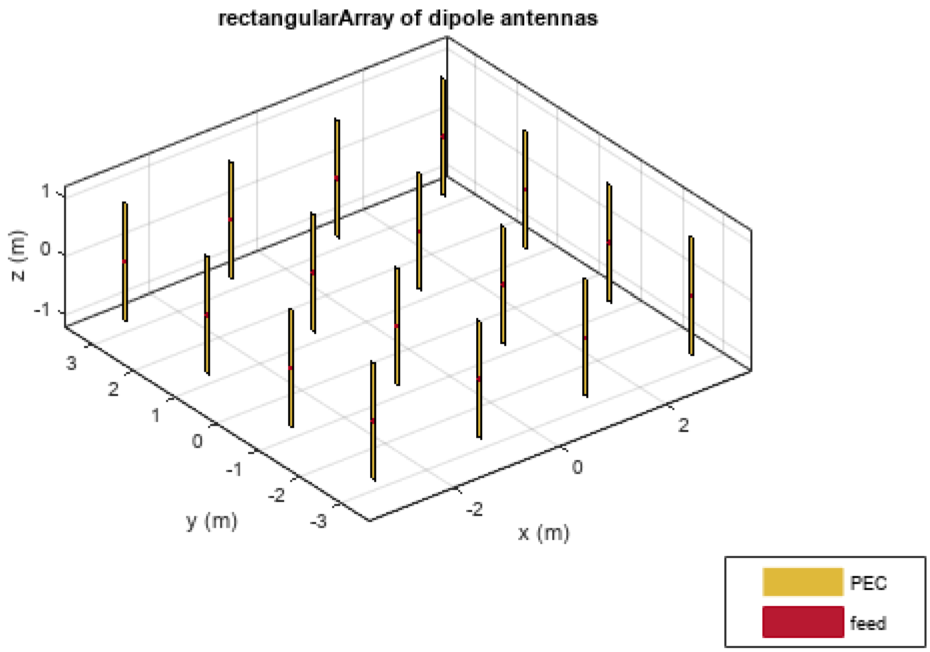
```
correlation(ra,60e6:1e6:70e6,1,2)
```



Change Size of Array and Visualize Layout

Use the 'Size' property of the rectangular array to change it to a 16-element dipole array.

```
ra.Size = [4 4];
show(ra)
```



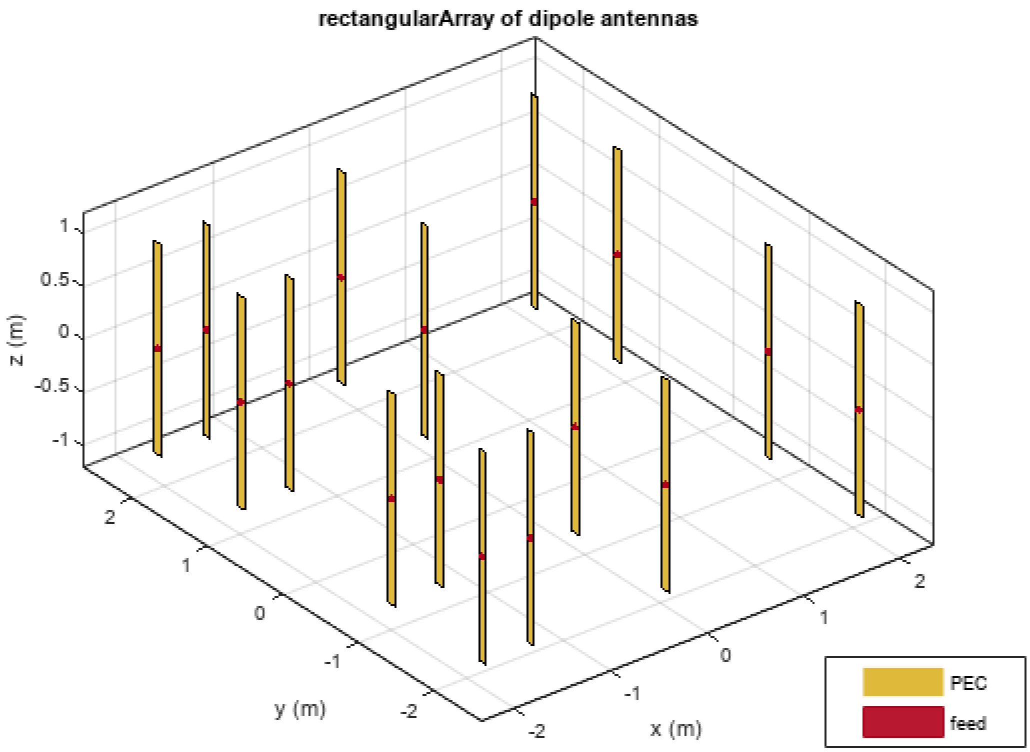
Change Array Elements Spacing To Nonuniform

Use the 'RowSpacing' and 'ColumnSpacing' properties of rectangular array to change the spacing between the antenna elements.

```
ra.RowSpacing = [ 1.1 2 1.2];
ra.ColumnSpacing =[0.5 1.4 2]
```

```
ra =
  rectangularArray with properties:

           Element: [1x1 dipole]
              Size: [4 4]
        RowSpacing: [1.1000 2 1.2000]
     ColumnSpacing: [0.5000 1.4000 2]
           Lattice: 'Rectangular'
    AmplitudeTaper: 1
        PhaseShift: 0
              Tilt: 0
          TiltAxis: [1 0 0]
```

```
show(ra)
```



References

[1] Balanis, C.A. "Antenna Theory. Analysis and Design", p. 514, Wiley, New York, 3rd Edition, 2005.

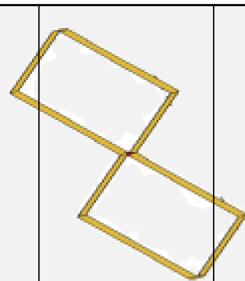
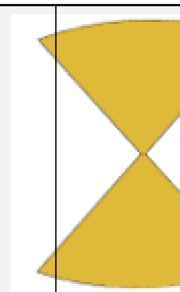
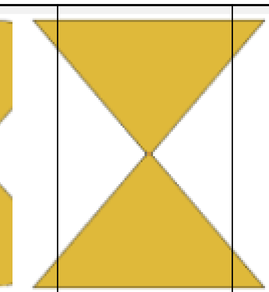
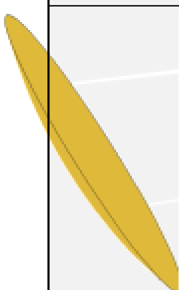
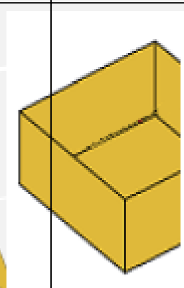
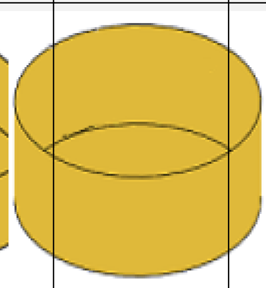
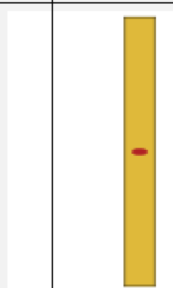
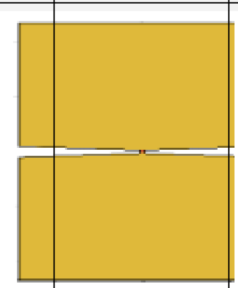
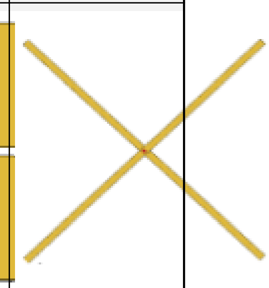
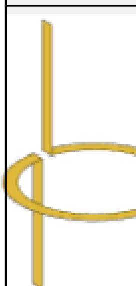
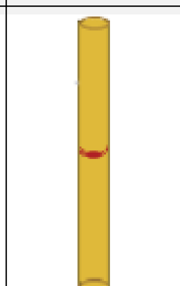
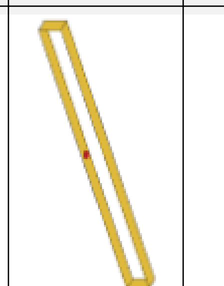
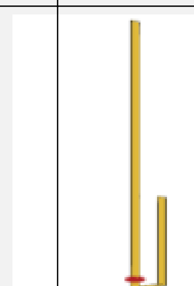
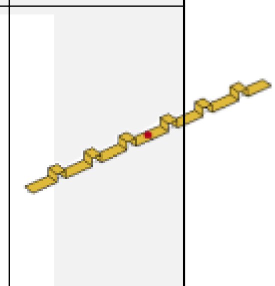
See Also

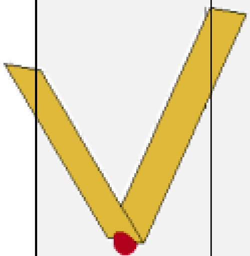
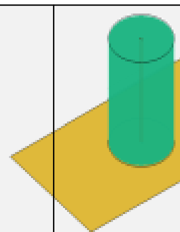
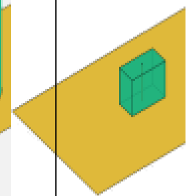
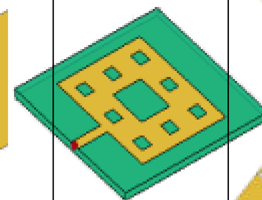
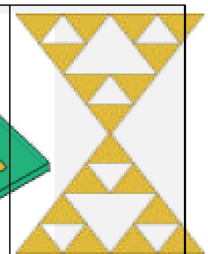
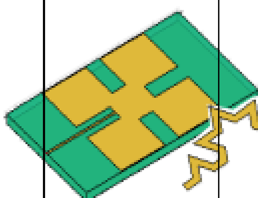
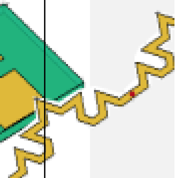
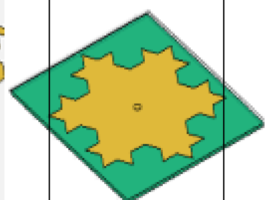
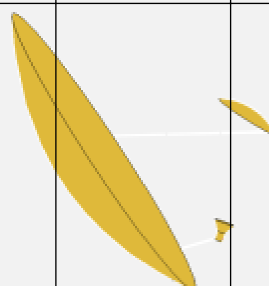
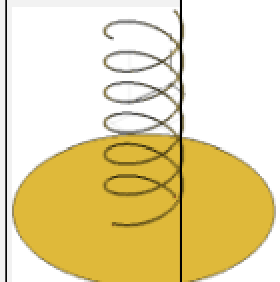
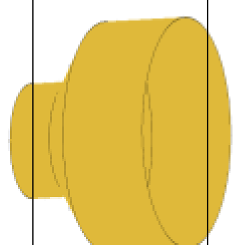
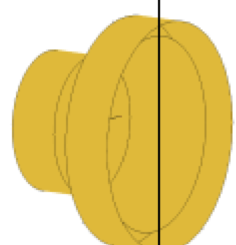
"Surrogate Based Optimization Design of Six-Element Yagi-Uda Antenna"

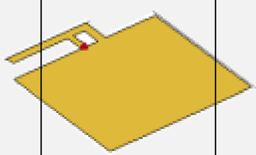
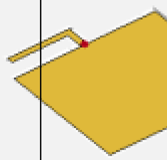
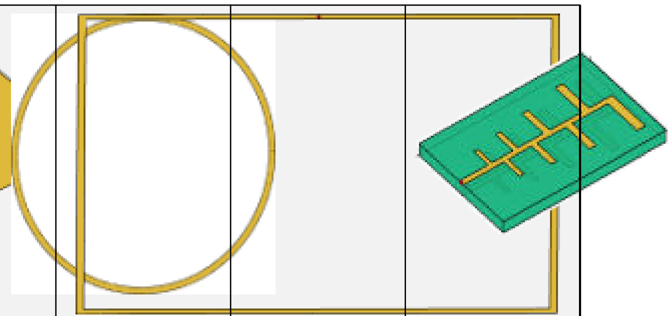
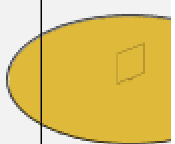
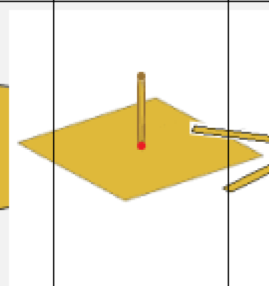
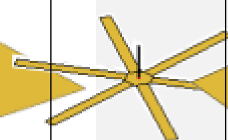
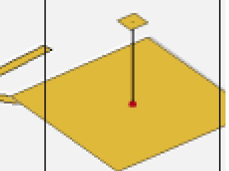
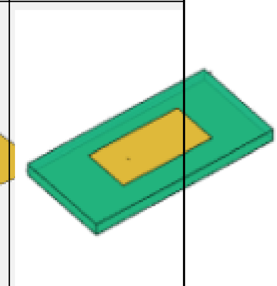
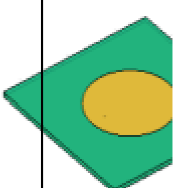
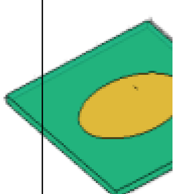
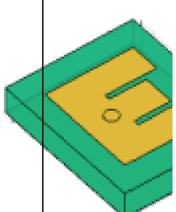
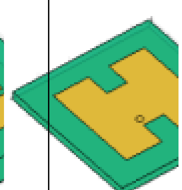
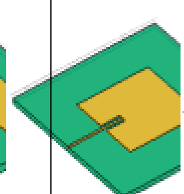
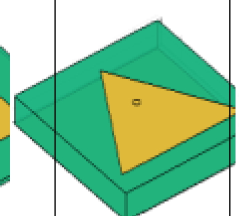
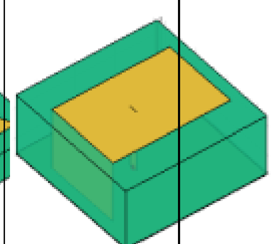
Antenna Element Catalog

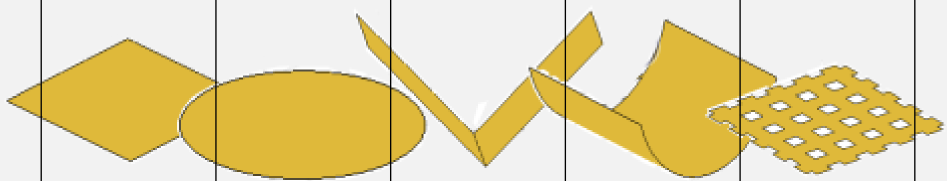
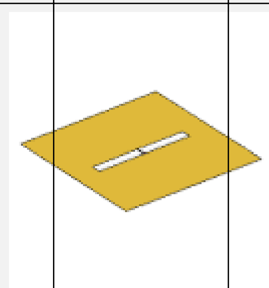
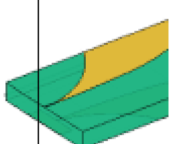
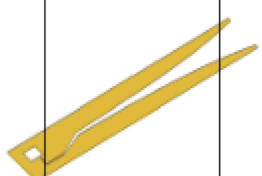
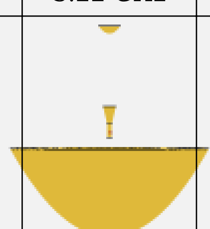
The Antenna Toolbox consists of two catalogs: Antenna and Array. This catalog illustrates all the antenna elements in Antenna Toolbox. The frequency in this catalog denotes the default operating frequency of every antenna.

Catalog Elements

						
`bicone` Frequency: 2.3 GHz	`biconeStrip` Frequency: 363.2 MHz	`biquad` Frequency: 2.8 GHz	`birdcage` Frequency: 64 MHz	`bowtieRounded` Frequency: 490 MHz	`bowtieTriangular` Frequency: 410 MHz	`cassegrain` Frequency: 18.51 GHz
						
`cassegrainOffset` Frequency: 17.8 GHz	`cavity` Frequency: 1 GHz	`cavityCircular` Frequency: 1 GHz	`cloverleaf` Frequency: 5.8 GHz	`dipole` Frequency: 75 MHz	`dipoleBlade` Frequency: 600 MHz	`dipoleCrossed` Frequency: 6 GHz
						
`dipoleCycloid` Frequency: 48 MHz	`dipoleCylindrical` Frequency: 70 MHz	`dipoleFolded` Frequency: 70.5 MHz	`dipoleHelix` Frequency: 2 GHz	`dipoleHelixMultifilar` Frequency: 2 GHz	`dipoleJ` Frequency: 144 MHz	`dipoleMeander` Frequency: 200 MHz

						
dipoleVee Frequency: 75 MHz	**discone** Frequency: 2.12 GHz	**disconeStrip** Frequency: 147.38 MHz	**draCylindrical** Frequency: 1.5 GHz	**draRectangular** Frequency: 3.3 GHz	**fractalCarpet** Frequency: 5.45 GHz	**fractalGasket** Frequency: 1.3 GHz
						
fractalIsland Frequency: 6 GHz	**fractalKoch** Frequency: 800 MHz	**fractalSnowflake** Frequency: 4.15 GHz	**gregorian** Frequency: 18.3 GHz	**gregorianOffset** Frequency: 17.76 GHz	**helix** Frequency: 2 GHz	**helixMultifilar** Frequency: 2 GHz
						
horn Frequency: 15 GHz	**hornConical** Frequency: 7.58 GHz	**hornConicalCorrugated** Frequency: 9.54 GHz	**hornCorrugated** Frequency: 15.28 GHz	**hornPotter** Frequency: 3.8 GHz	**hornRidge** Frequency: 11 GHz	**hornScrimp** Frequency: 4 GHz

						
invertedF Frequency: 1.7 GHz	**invertedFcoplanar** Frequency: 1.7 GHz	**invertedL** Frequency: 1.7 GHz	**invertedLcoplanar** Frequency: 1.7 GHz	**loopCircular** Frequency: 75 MHz	**loopRectangular** Frequency: 53 MHz	**lpda** Frequency: 5.5 GHz
						
monocone Frequency: 3.8 GHz	**monopole** Frequency: 75 MHz	**monopoleCustom** Frequency: 1.24 GHz	**monopoleCylindrical** Frequency: 70 MHz	**monopoleRadial** Frequency: 75 MHz	**monopoleTopHat** Frequency: 75 MHz	**patchMicrostrip** Frequency: 1.67 GHz
						
patchMicrostripCircular Frequency: 1 GHz	**patchMicrostripElliptical** Frequency: 5.45 GHz	**patchMicrostripEnotch** Frequency: 6.6 GHz	**patchMicrostripHnotch** Frequency: 3.49 GHz	**patchMicrostripInsetfed** Frequency: 4.5 GHz	**patchMicrostripTriangular** Frequency: 12.5 GHz	**pifa** Frequency: 2.4 GHz

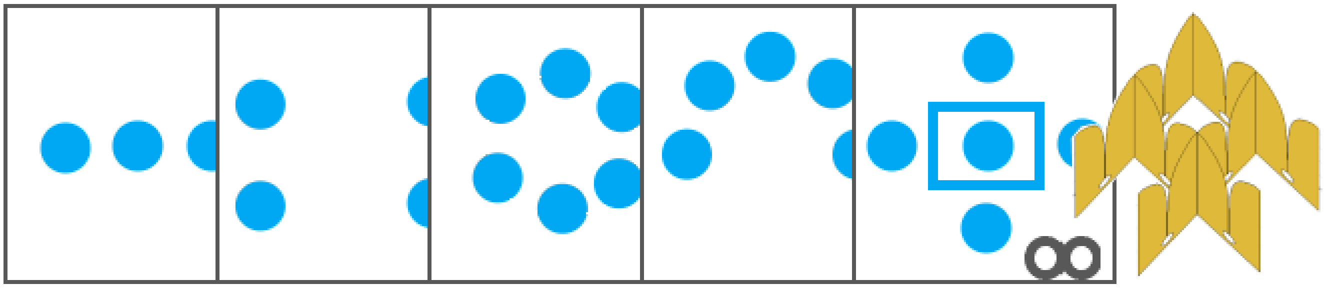
						
quadCustom Frequency: 2.4 GHz	**reflector** Frequency: 1 GHz	**reflectorCircular** Frequency: 1 GHz	**reflectorCorner** Frequency: 1 GHz	**reflectorCylindrical** Frequency: 1 GHz	**reflectorGrid** Frequency: 1 GHz	**reflectorParabolic** Frequency: 10 GHz
						
reflectorSpherical Frequency: 10 GHz	**rhombic** Frequency: 510 MHz	**sectorInvertedAmos** Frequency: 2.45 GHz	**slot** Frequency: 130 MHz	**spiralArchimedean** Frequency: 5 GHz	**spiralEquiangular** Frequency: 5 GHz	**spiralRectangular** Frequency: 7.68 GHz
						
vivaldi Frequency: 3.2 GHz	**vivaldiAntipodal** Frequency: 3.22 GHz	**vivaldiOffsetCavity** Frequency: 18 GHz	**waveguide** Frequency: 6.5 GHz	**waveguideCircular** Frequency: 8.42 GHz	**waveguideRidge** Frequency: 9.45 GHz	**waveguideSlotted** Frequency: 2.45 GHz
						

`yagiUda` Frequency: 300 MHz	**`customDualReflectors`** Frequency: 18.51 GHz					

See Also

"Array Catalog Elements" on page 2-25 | "Antenna Radiation Patterns" on page 2-26

Array Catalog Elements



`linearArray` `rectangularArray` `circularArray` `conformalArray` `infiniteArray` `eggCrate`

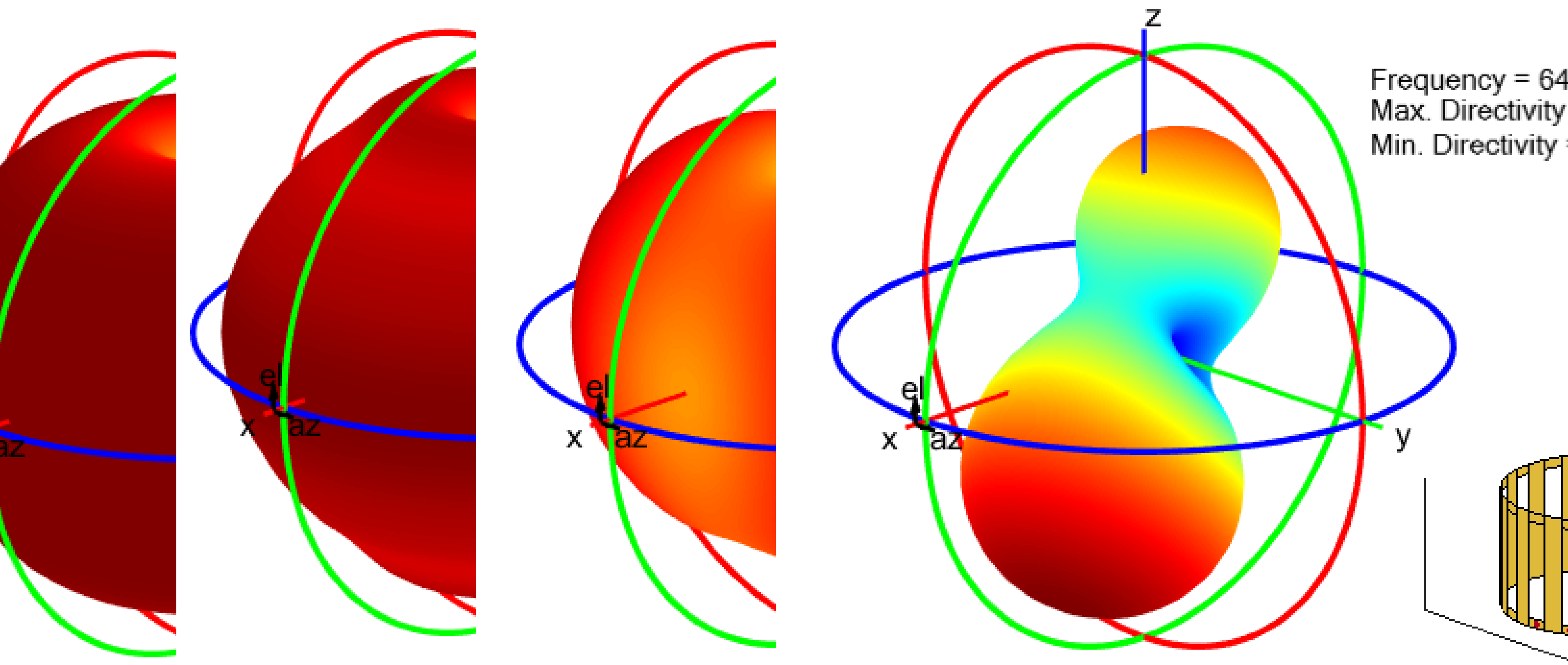
See Also

"Antenna Element Catalog" on page 2-19

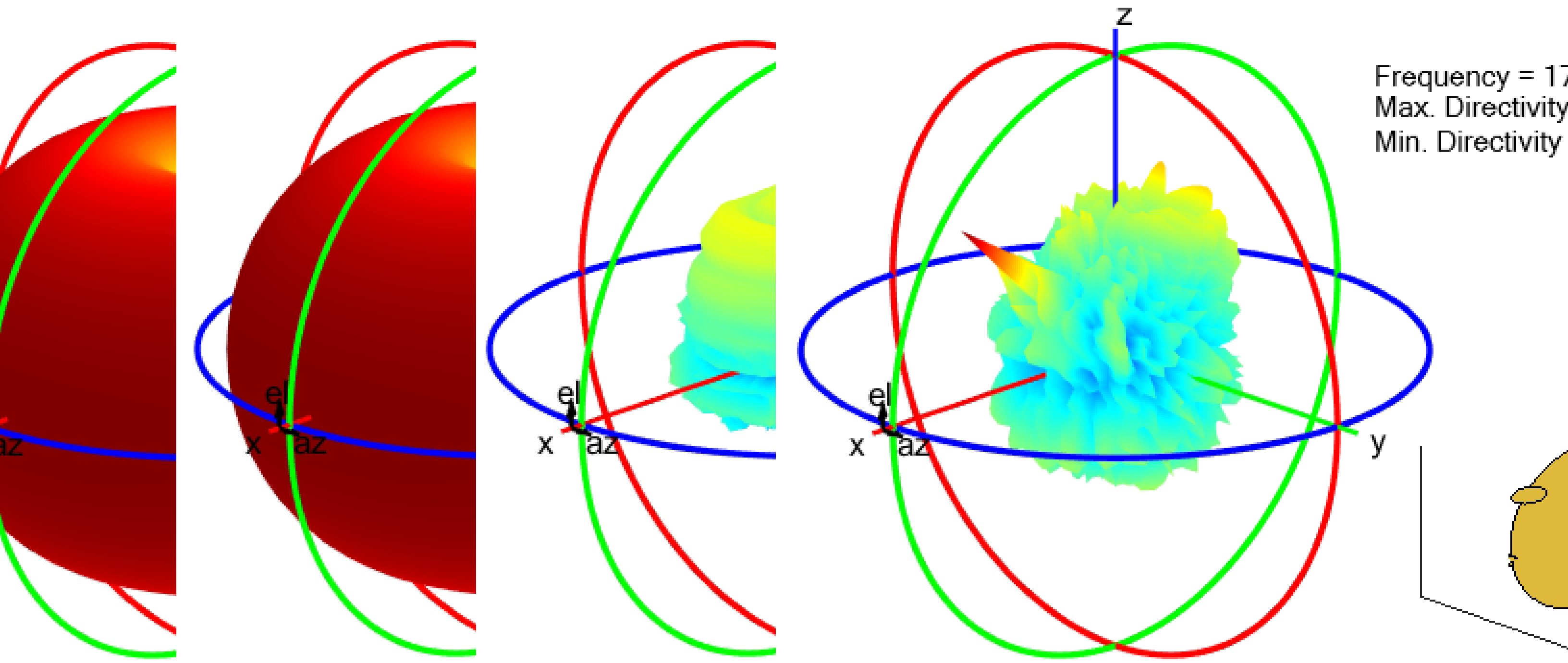
Antenna Radiation Patterns

The Antenna Toolbox allows to plot 3-D radiation pattern of the antenna or array object over a specified Frequency This table illustrates all the antenna elements in Antenna Toolbox. To know more about antenna radiation patterns see `pattern`.

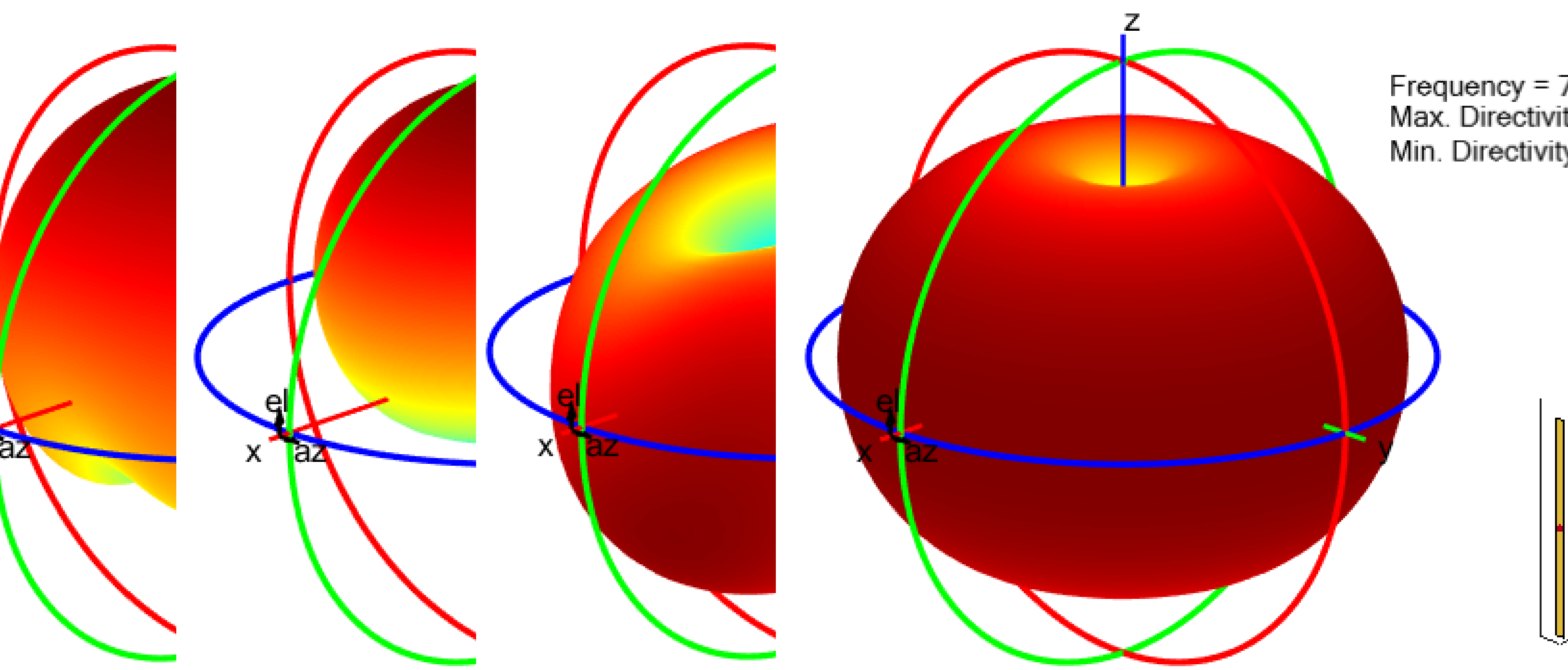
`bicone`	**`biconeStrip`**	**`biquad`**	**`birdcage`**

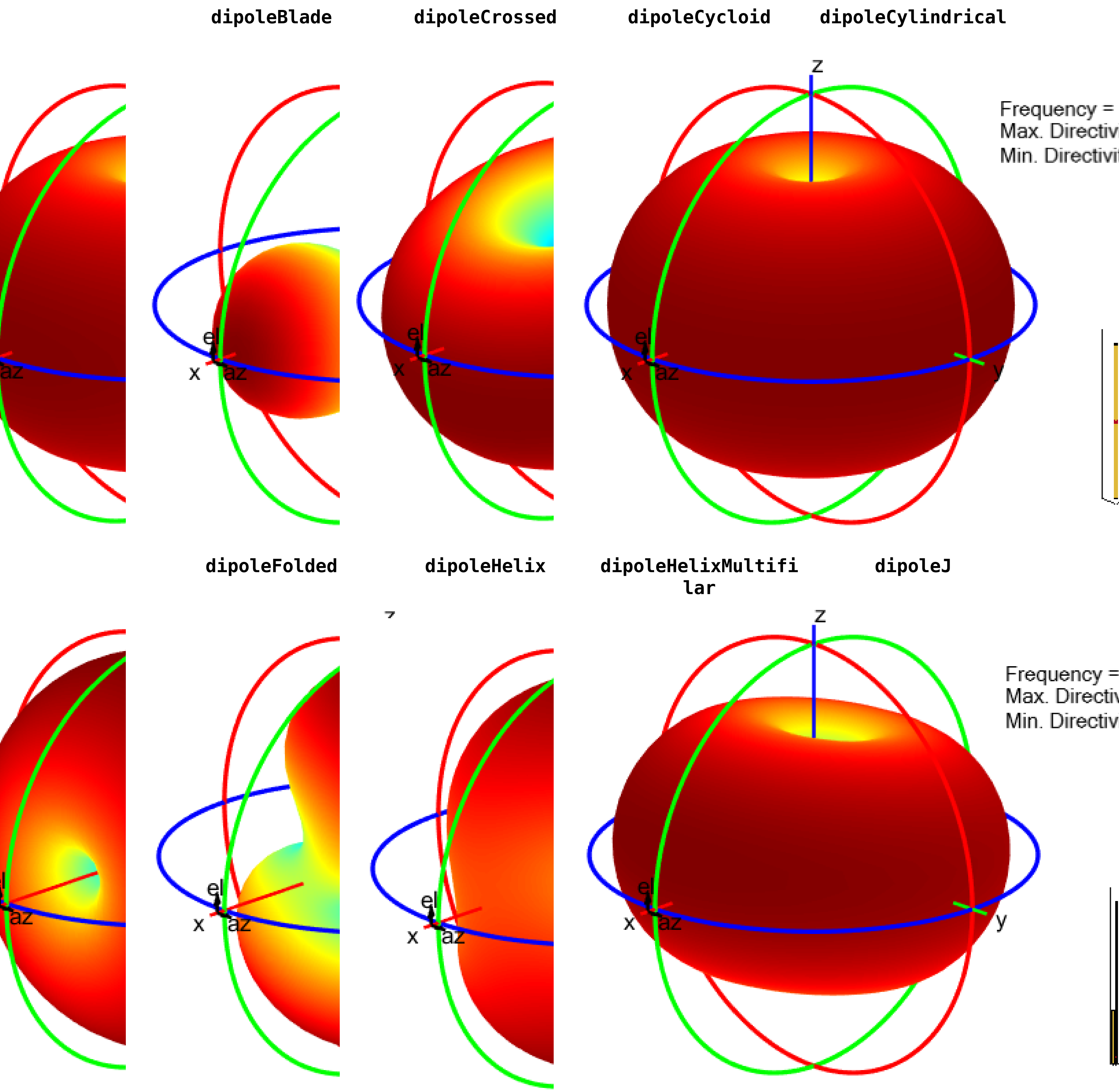


bowtieRounded	bowtieTriangular	cassegrain	cassegrainOffset

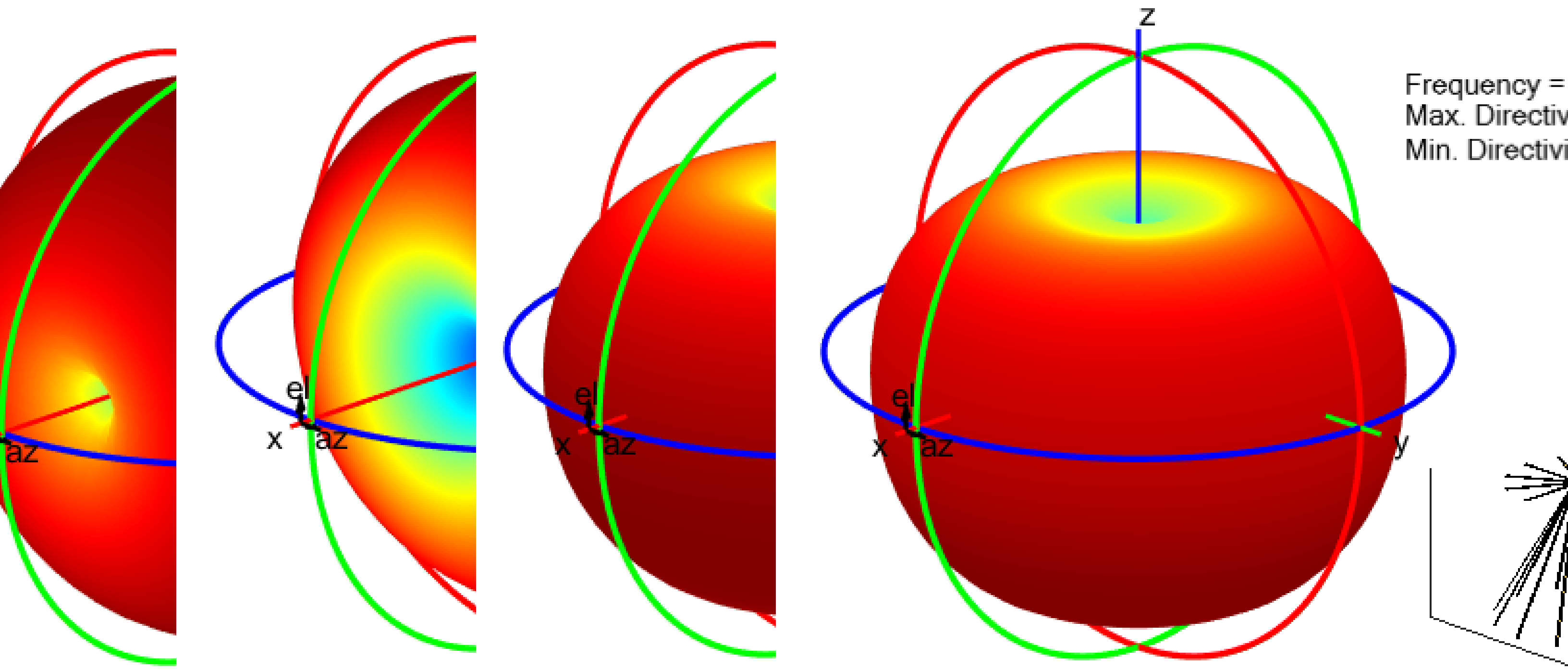


cavity	cavityCircular	cloverleaf	dipole

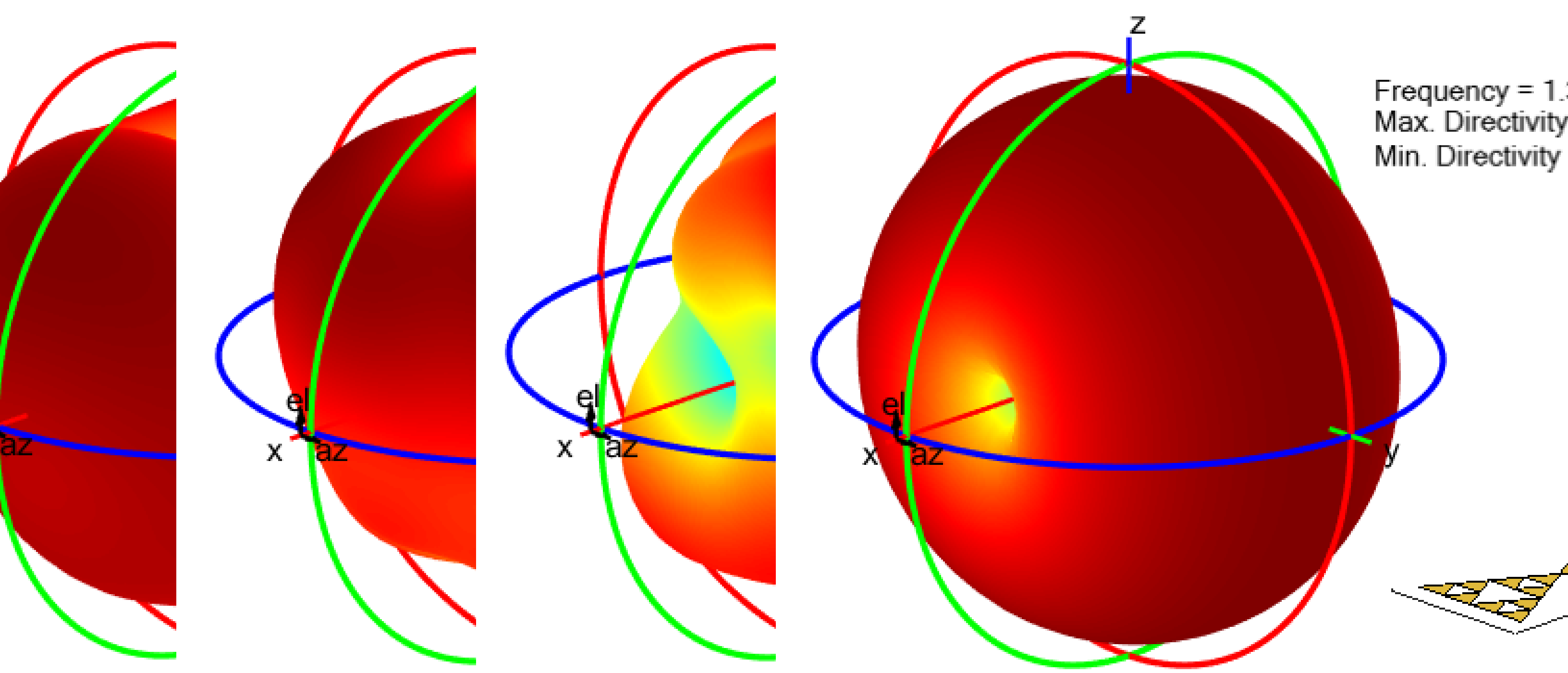


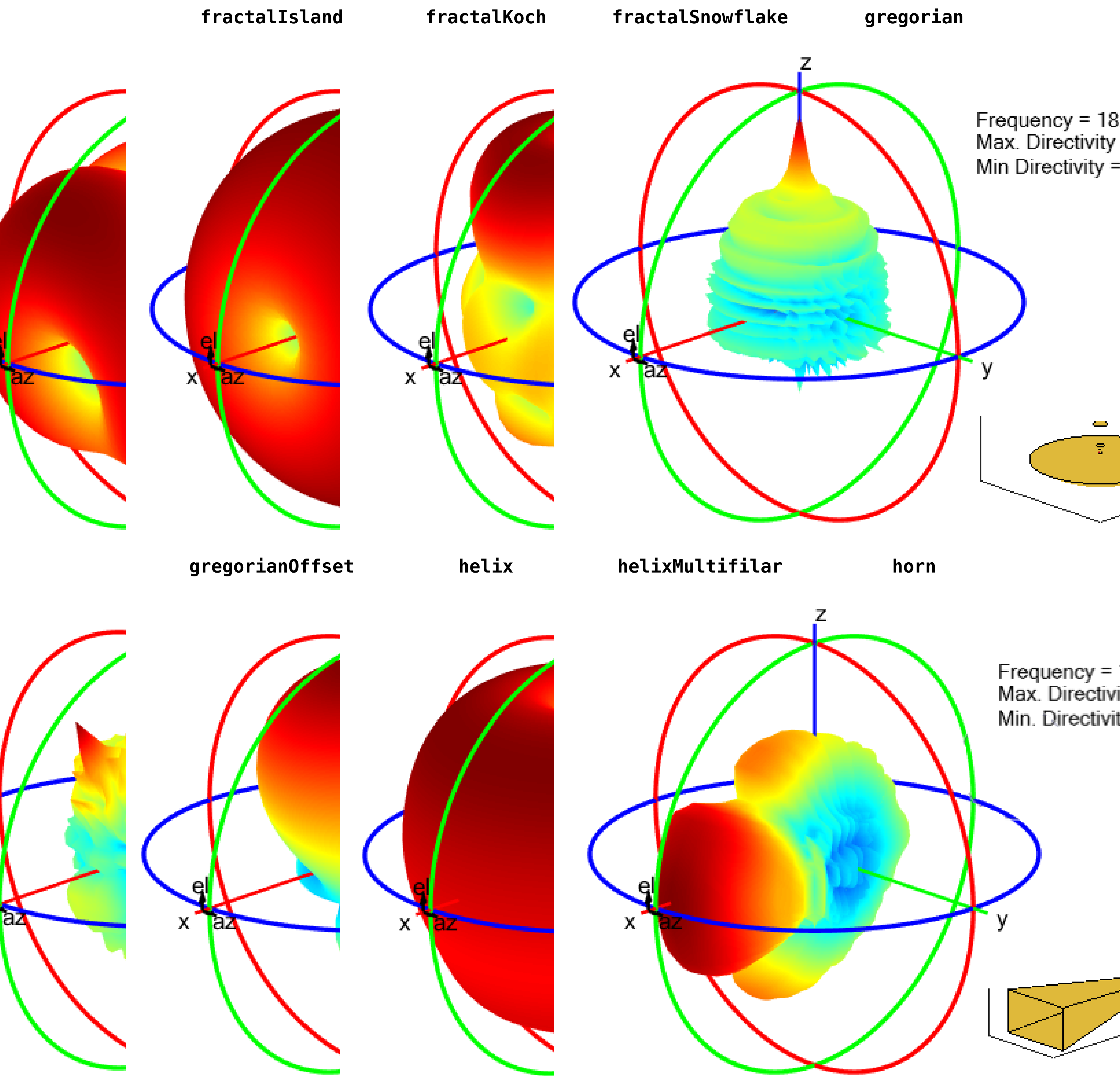

dipoleBlade
dipoleCrossed
dipoleCycloid
dipoleCylindrical
dipoleFolded
dipoleHelix
dipoleHelixMultifilar
dipoleJ
Frequency =
Frequency =
z
x
y
el
az

`dipoleMeander` `dipoleVee` `discone` `disconeStrip`

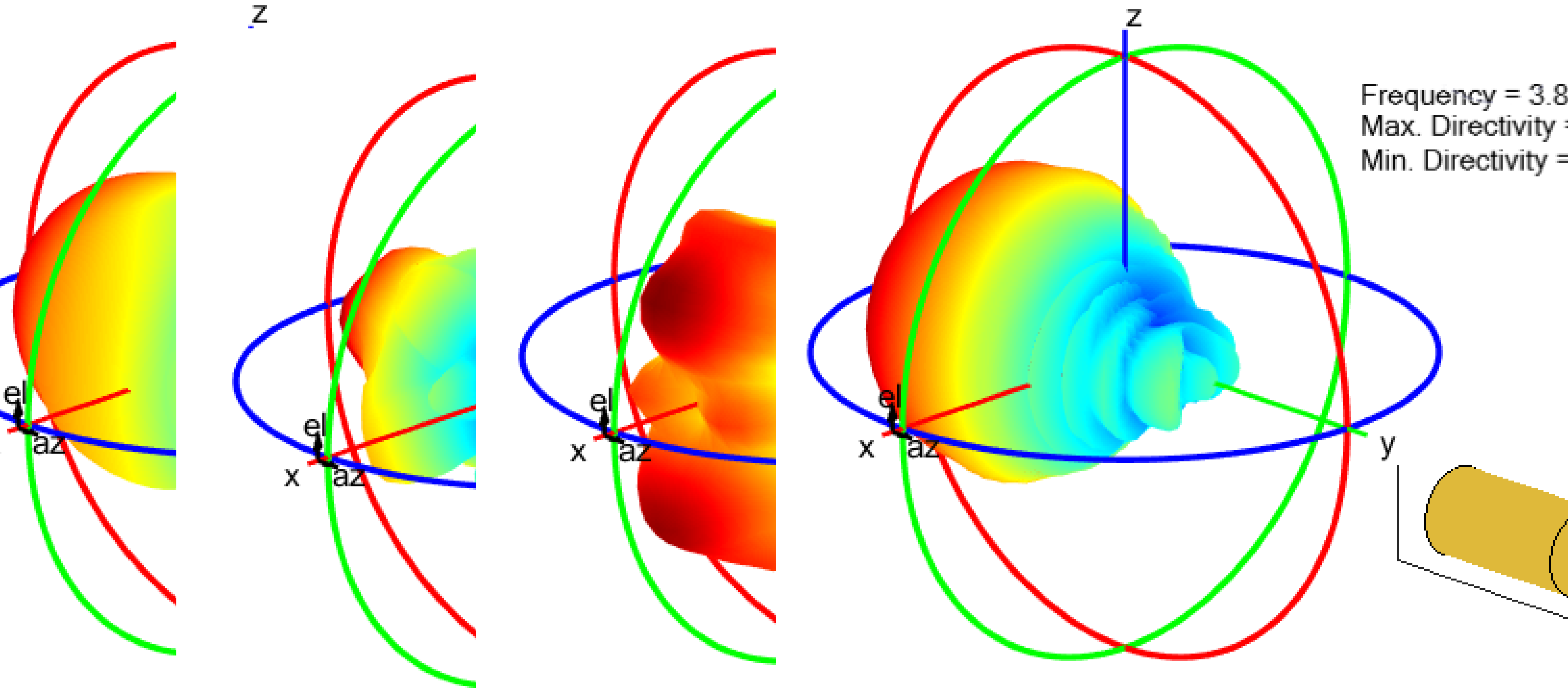


`draCylindrical` `draRectangular` `fractalCarpet` `fractalGasket`

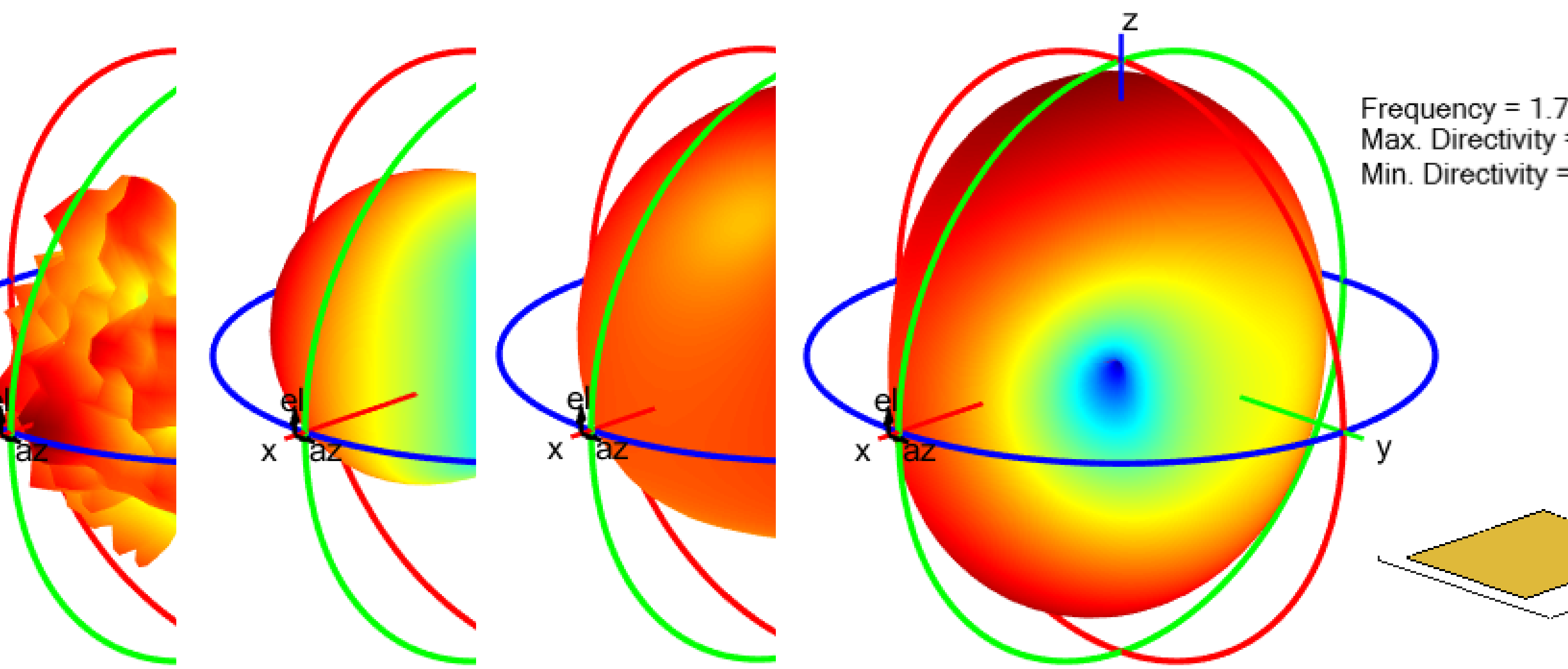


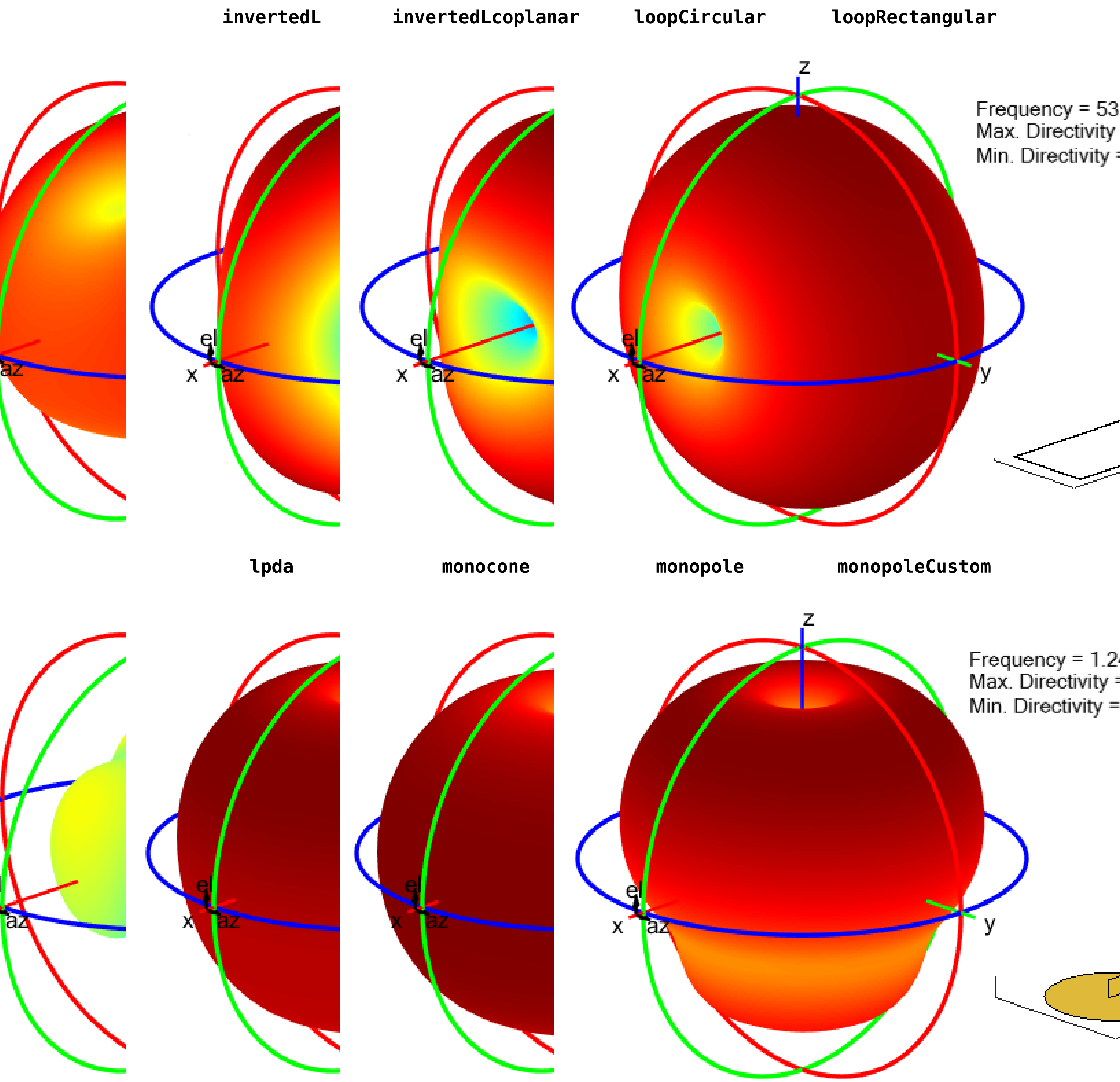

fractalIsland
fractalKoch
fractalSnowflake
gregorian
gregorianOffset
helix
helixMultifilar
horn
z
x
y
el
az
Frequency = 18
Max. Directivity
Min Directivity =

hornConical	hornConicalCorrugated	hornCorrugated	hornPotter

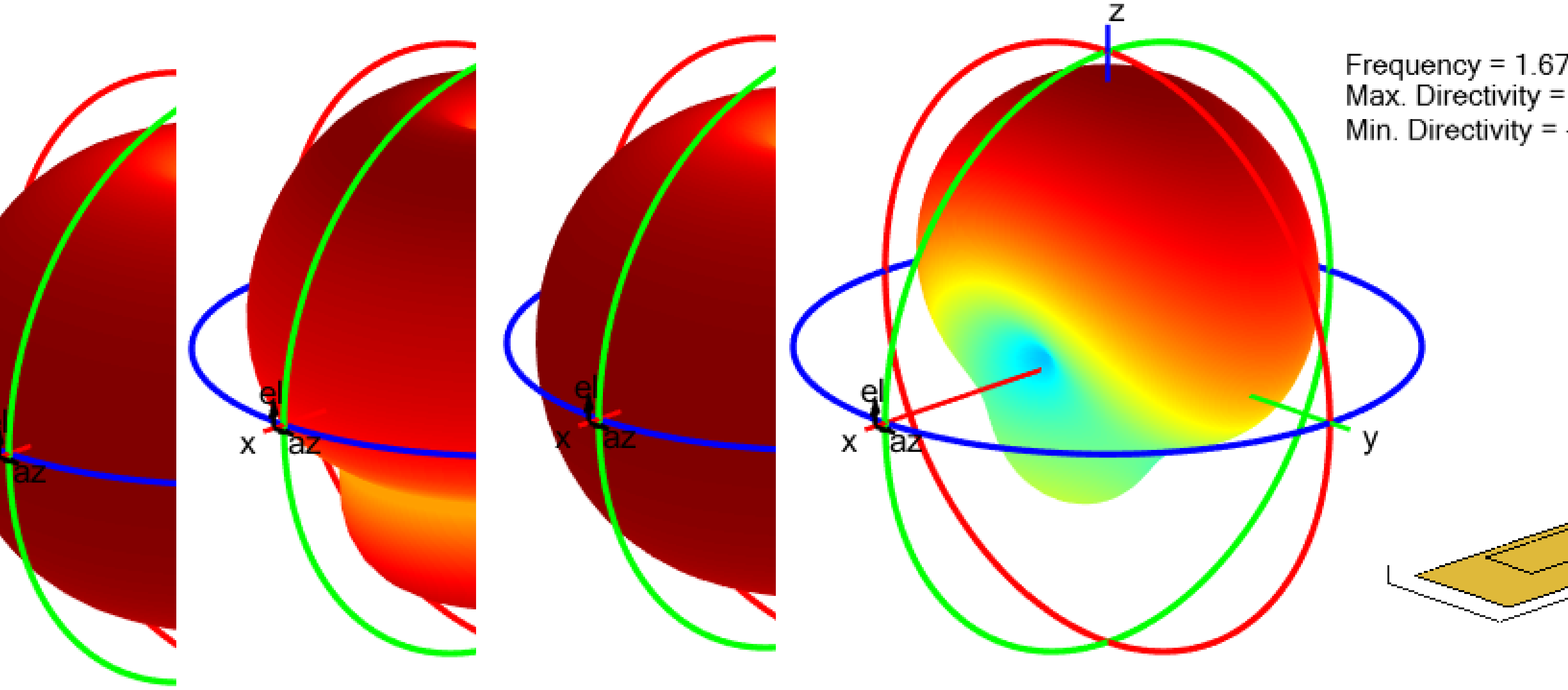


hornRidge	hornScrimp	invertedF	invertedFcoplanar

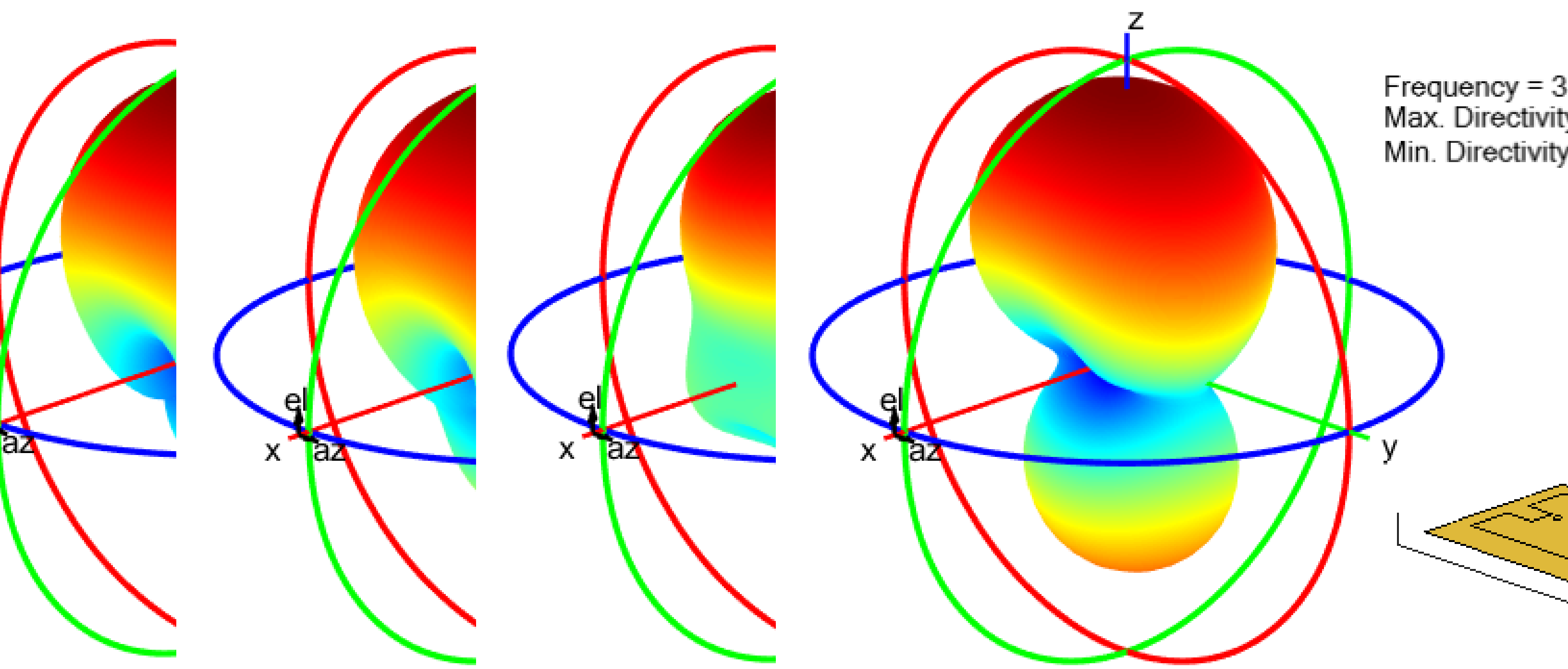


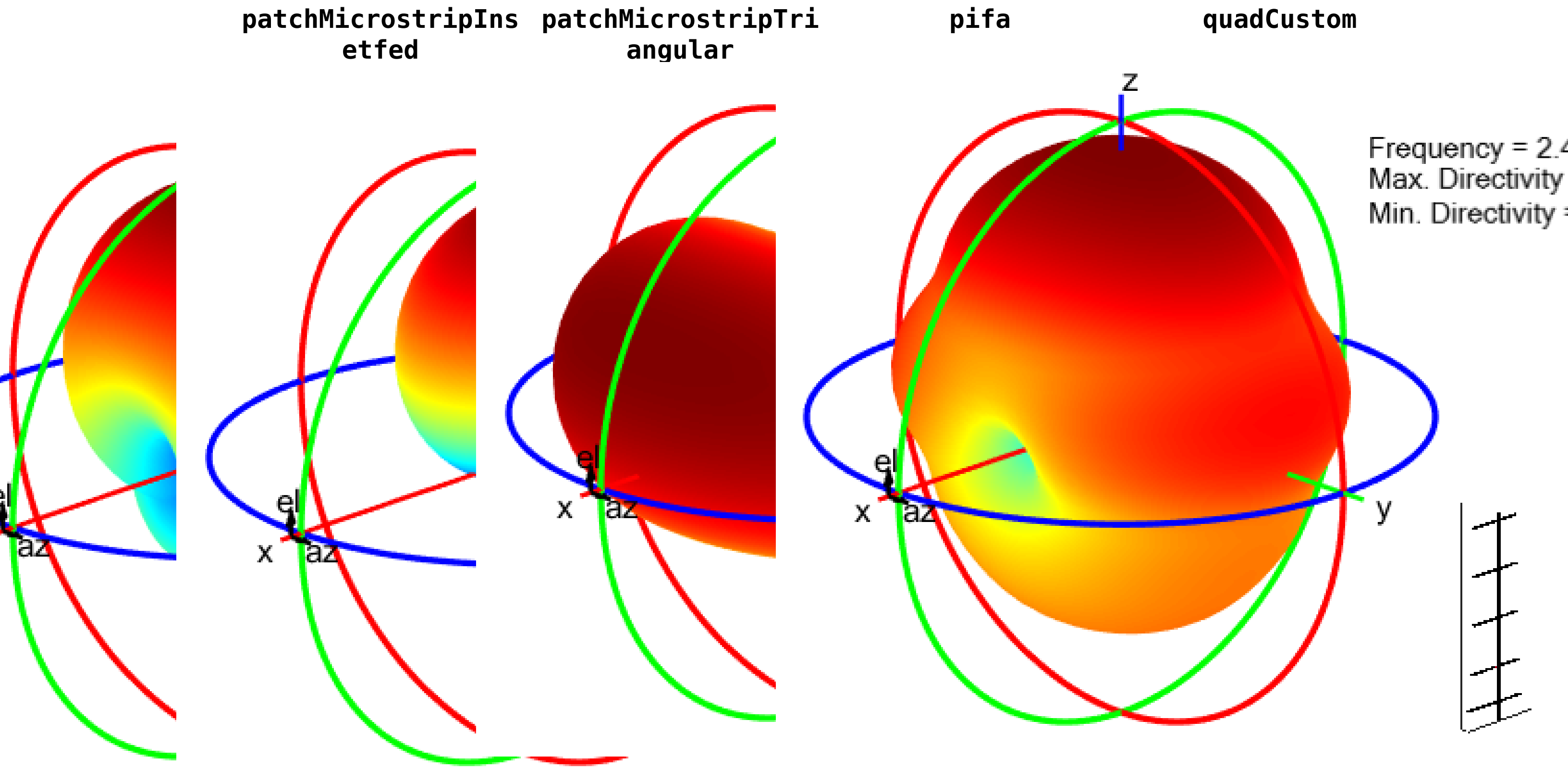

invertedL
invertedLcoplanar
loopCircular
loopRectangular
z
Frequency = 53
Max. Directivity
Min. Directivity =
el
x
az
y
lpda
monocone
monopole
monopoleCustom
Frequency = 1.2
Max. Directivity =
Min. Directivity =

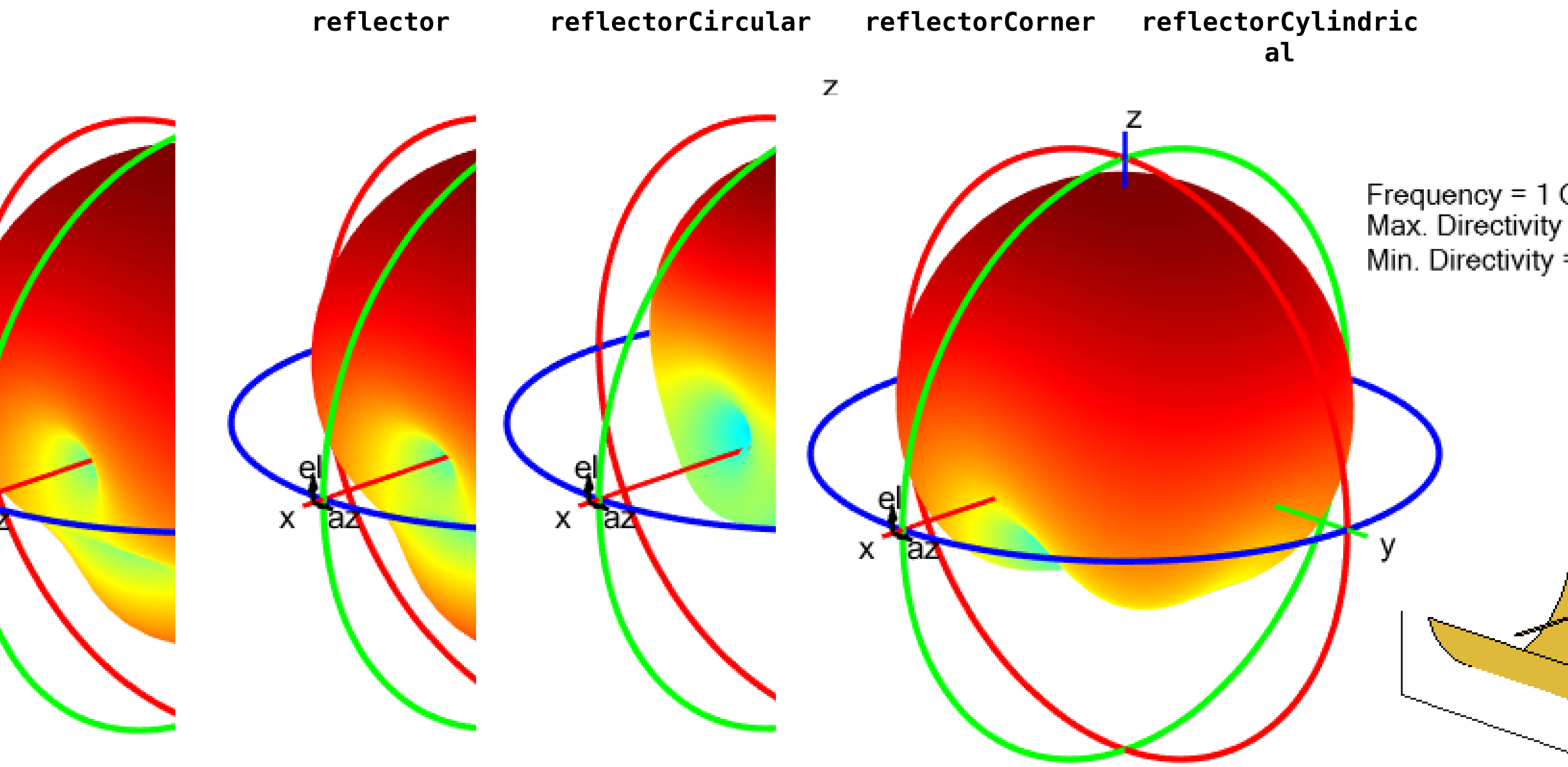
monopoleCylindrical | **monopoleRadial** | **monopoleTopHat** | **patchMicrostrip**

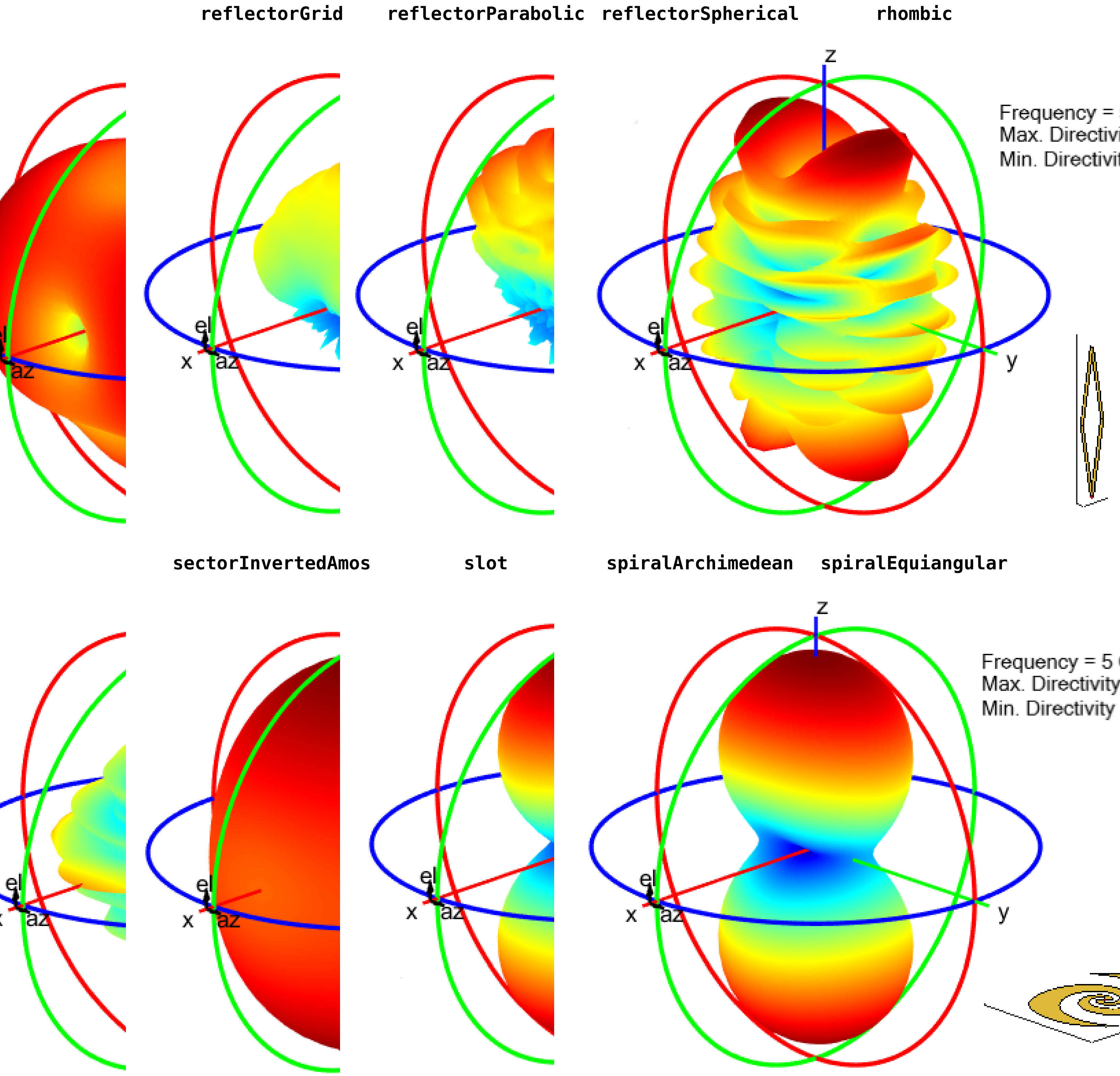


patchMicrostripCircular | **patchMicrostripElliptical** | **patchMicrostripEnotch** | **patchMicrostripHnotch**

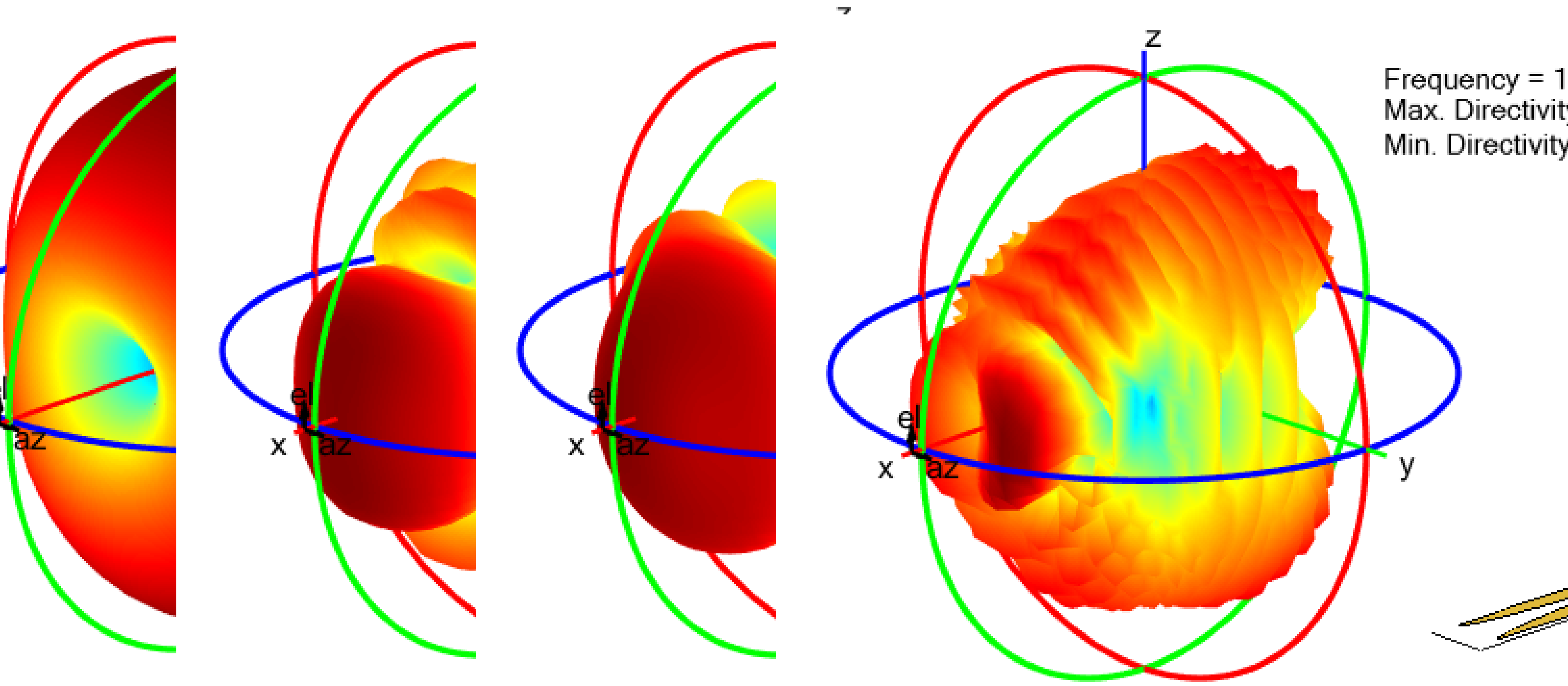



patchMicrostripInsetfed
patchMicrostripTriangular
pifa
quadCustom
z
x
y
el
az
Frequency = 2.4
Max. Directivity
Min. Directivity

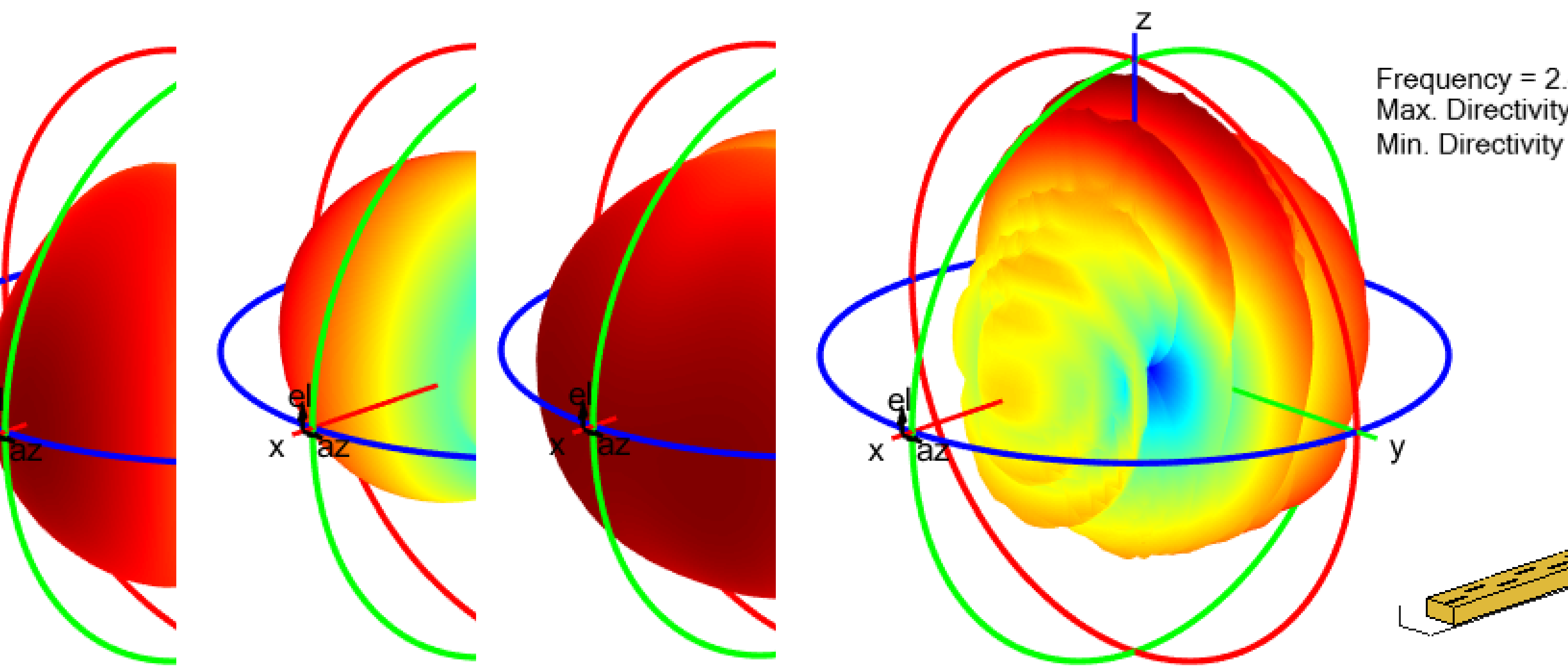

reflector
reflectorCircular
reflectorCorner
reflectorCylindrical
z
x
y
el
az
Frequency = 1 G
Max. Directivity
Min. Directivity


reflectorGrid
reflectorParabolic
reflectorSpherical
rhombic
sectorInvertedAmos
slot
spiralArchimedean
spiralEquiangular
z
x
y
el
az
Frequency =

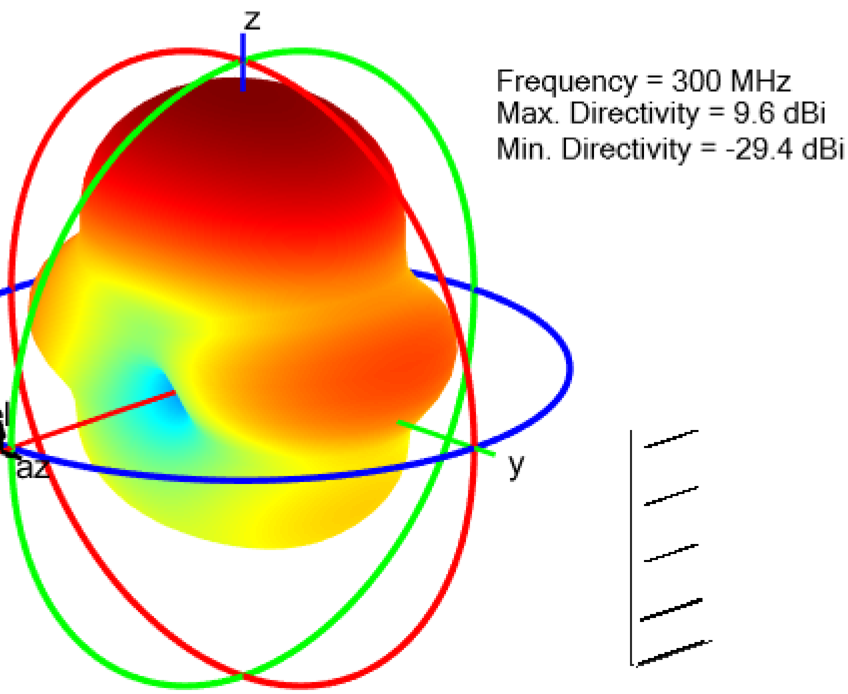
spiralRectangular	vivaldi	vivaldiAntipodal	vivaldiOffsetCavity



waveguide	waveguideCircular	waveguideRidge	waveguideSlotted



yagiUda



See Also

"Antenna Element Catalog" on page 2-19

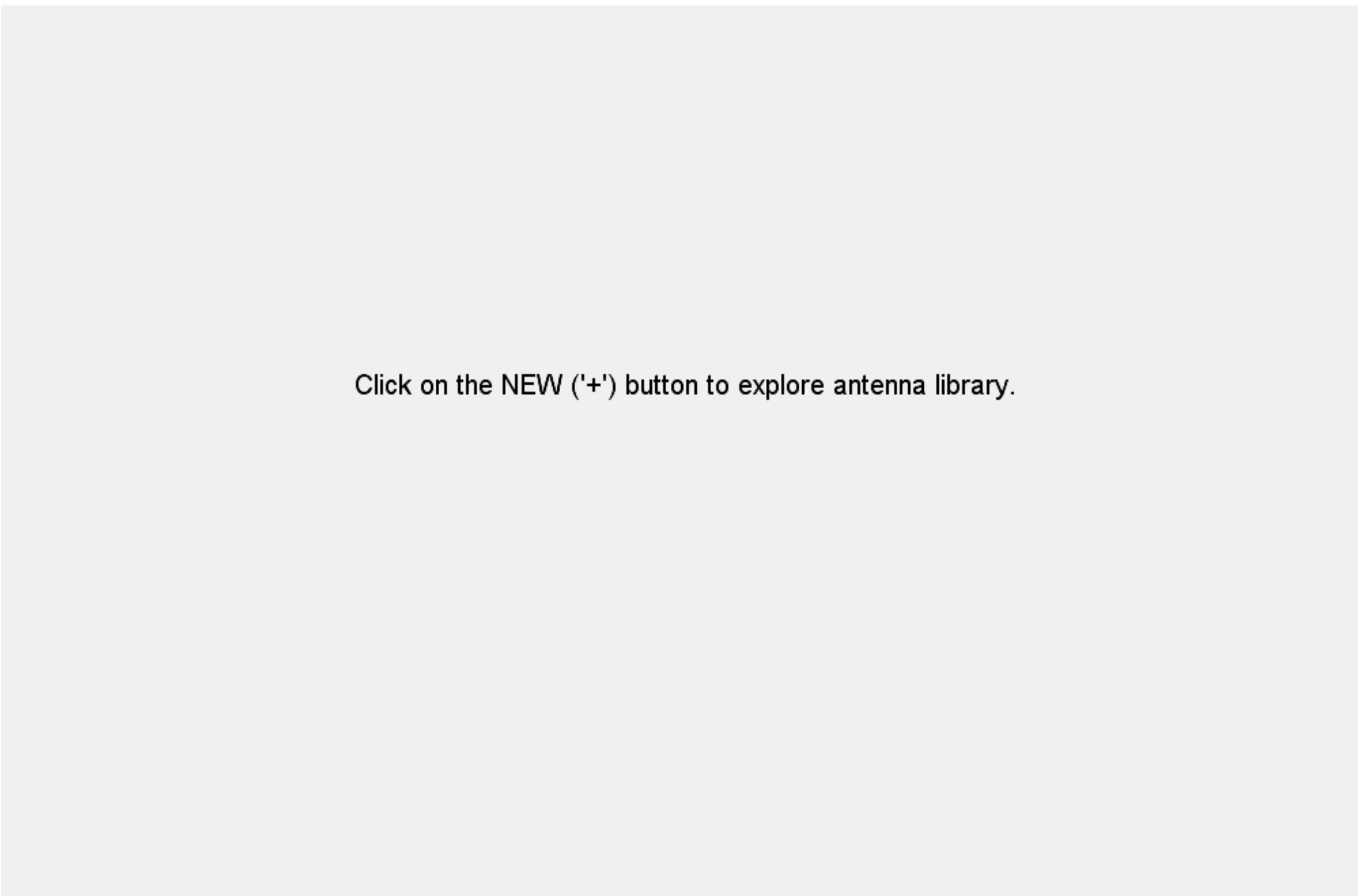
Design and Analysis Using Antenna Array Designer

This example shows how to create and analyze a 6-element linear array of half-wavelength dipoles using Antenna Array Designer app in the Antenna Toolbox™. The design and analysis are performed at 2.1GHz

Open Antenna Array Designer App

To open the app, at the MATLAB command prompt enter: The command opens a blank canvas.

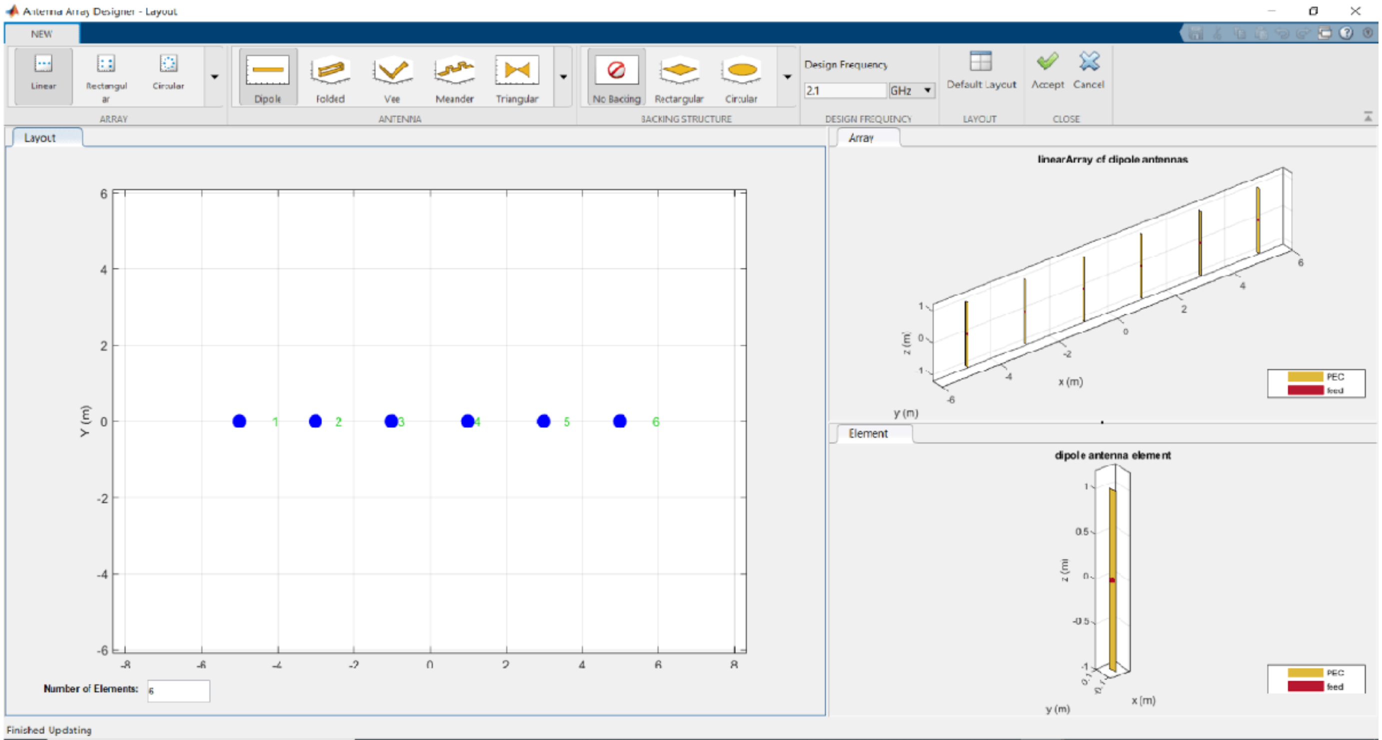
```
antennaArrayDesigner
```



Design Antenna Array

In the blank canvas, click New. In the **ARRAY GALLERY**, select **Linear Array**.

Set the **Design Frequency** value to **2.1GHz**. Set the **Number of Elements** to **6**.

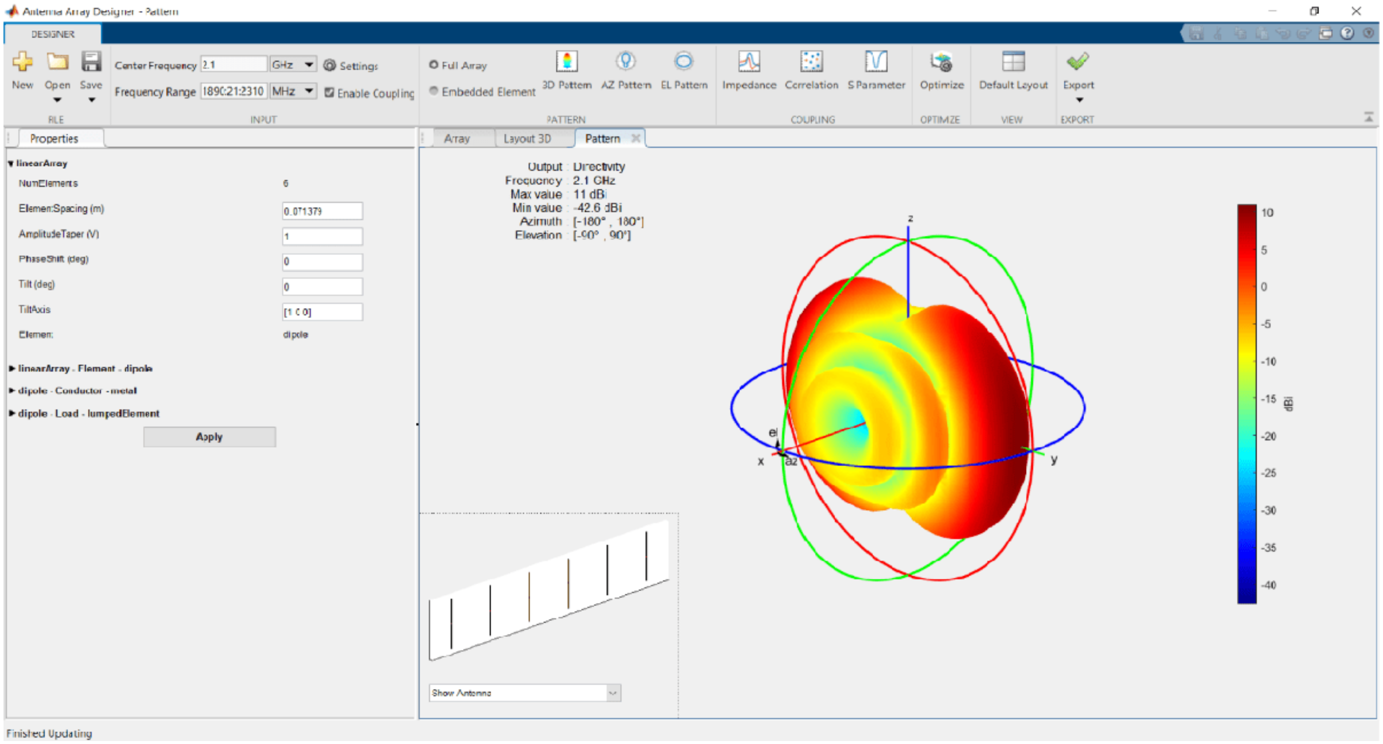


To analyze this antenna array, click **Accept.**

Plot 3-D Pattern

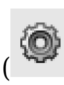
Observe the array geometry and the dipole's geometry at 2.1GHz in the **Array** and **Layout** figure tabs.

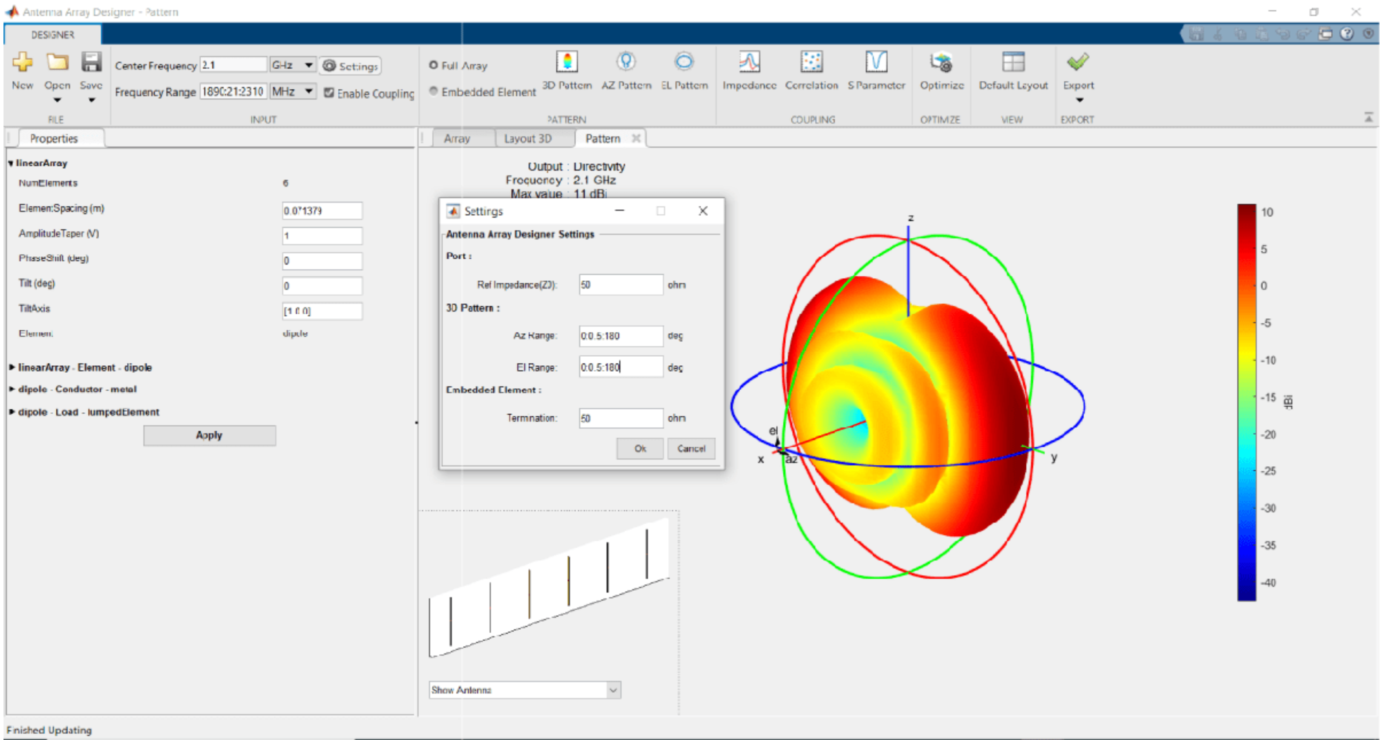
In the toolstrip under the **PATTERN** section, click **3D Pattern** to visualize the pattern for the linear array at the design frequency.



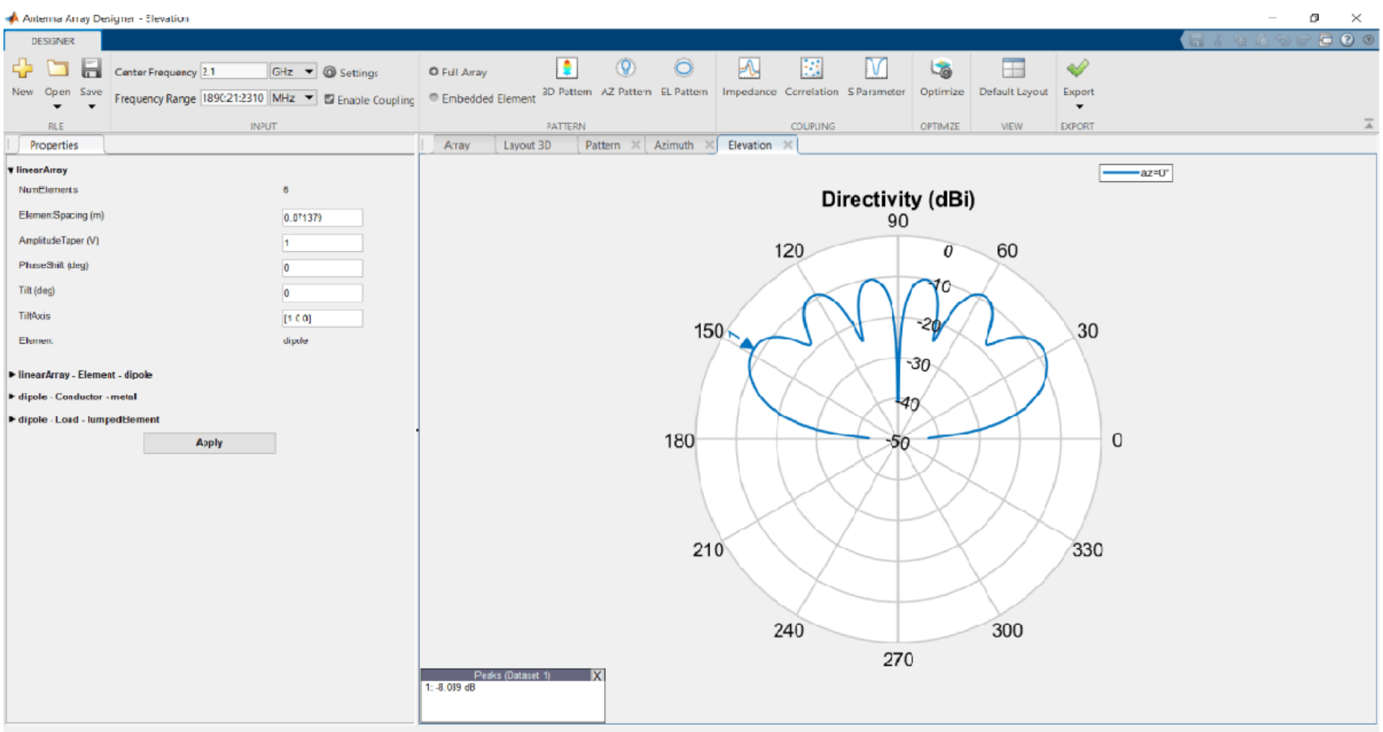
The array has a peak directivity of **11 dBi** with maximum beam directed at 90 degree azimuth.

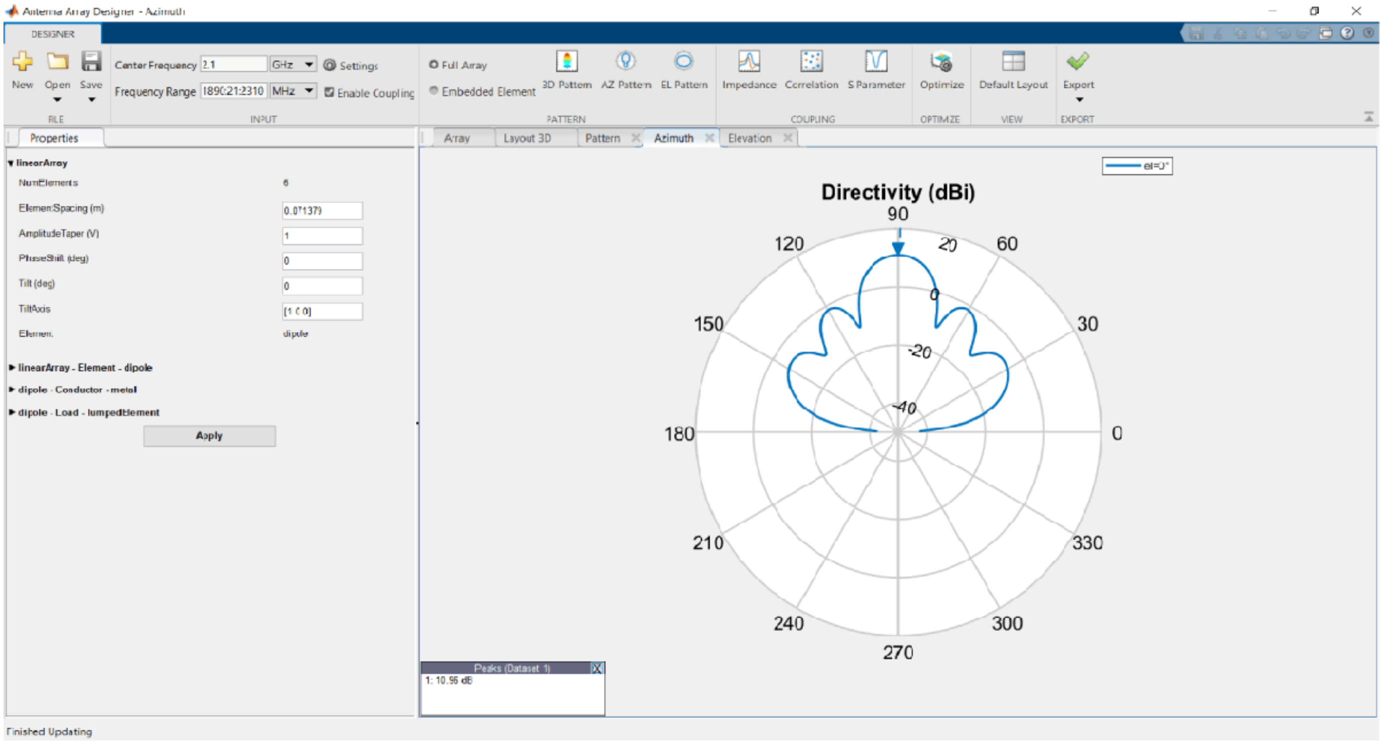
Plot Azimuth and Elevation Pattern

In the toolstrip, under the **INPUT** section click on the **Settings** ( icon) to change the azimuth and elevation range values. Change the **Az Range** and **El Range** to **0:0.5:180**. Click **Ok**.



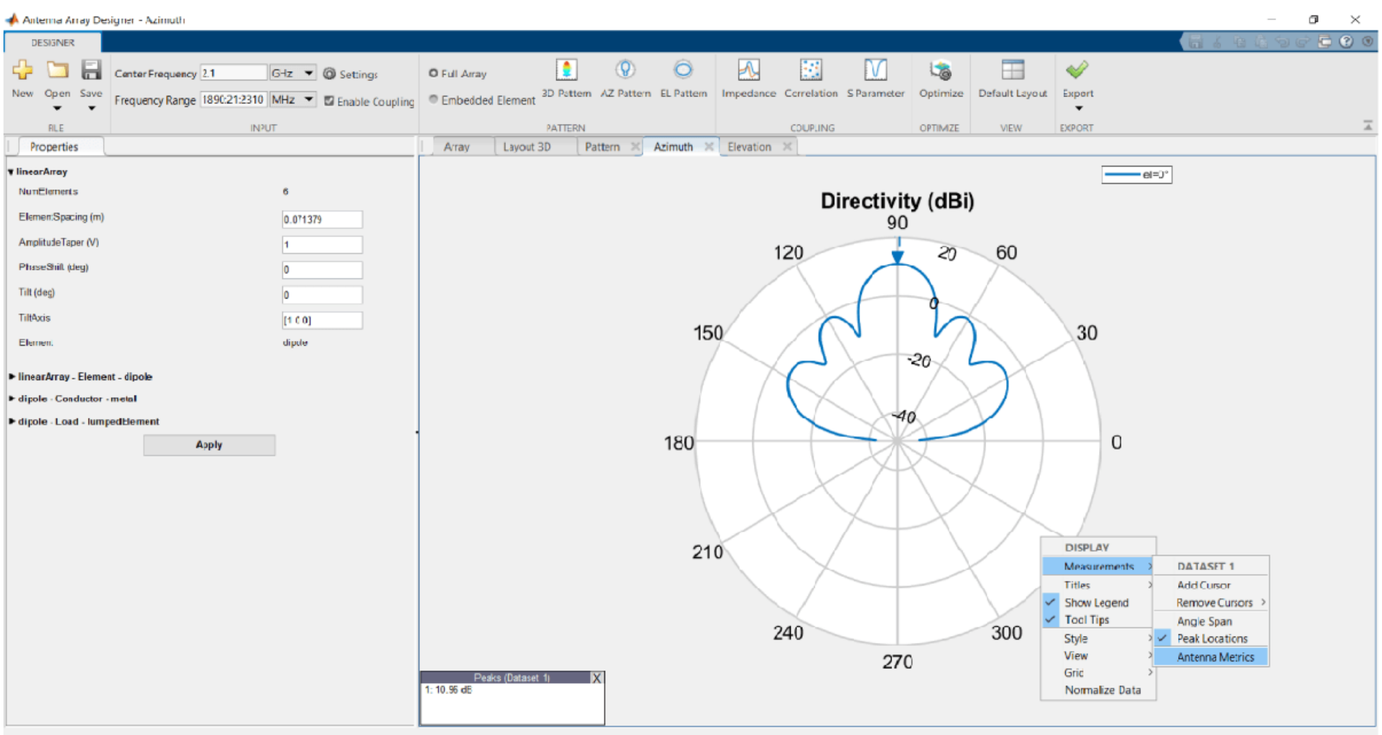
Under **PATTERN** section, click on the **AZ Pattern** and **EL Pattern** to view azimuth and elevation patterns of the linear array. Observe the azimuth and elevation range.



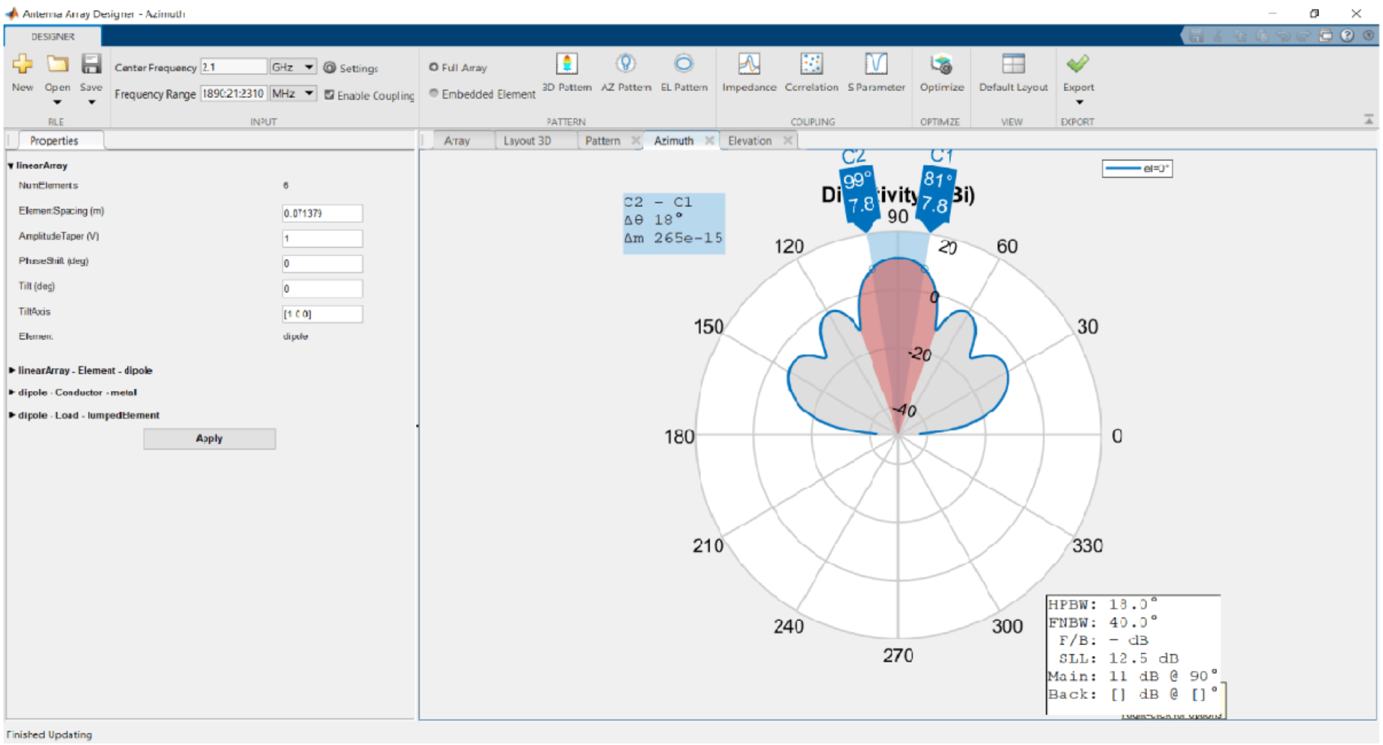


Half-power beamwidth (HPBW) and Sidelobe level (SLL)

Click on the AZ Pattern tab in the designer. Right click on the polar plot. Select **Antenna Metrics** from the **Measurements** tab of the context menu.



The half-power beamwidth is **18 degrees** and the sidelobe level is **12.5 dB**.



Change Maximum Beam Angle

Phase shift property of the array allows to direct the maximum beam to a specific angle. Steer the beam direction to azimuth 80 degree.

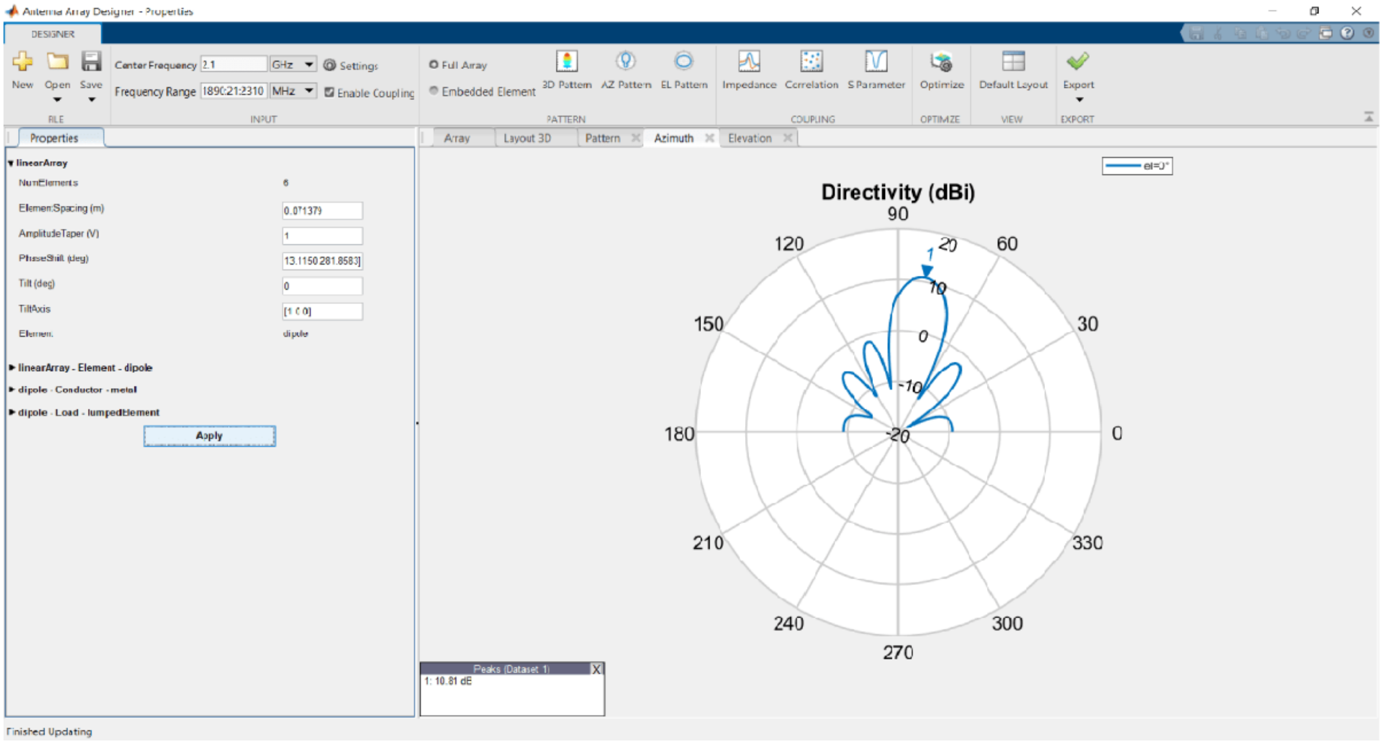
Antenna Toolbox's **phaseShift** method can be used to compute the progressive phase shift required to steer the beam to 80 degrees azimuth.

The values for phase shift are obtained as below

```
linArray = design(linearArray('NumElements',6),2.1e9);
ps = phaseShift(linArray,2.1e9,[80 0])';
```

In the **Property Panel**, under **Geometry - linearArray** change the **PhaseShift** property to **[78.1417 46.8850 15.6283 344.3717 313.1150 281.8583].**

Click **Apply**. All the analysis gets updated to account for the changes to array configuration. Click on **AZ Pattern plot** tab to see the maximum beam which is now at 80 degree azimuth.

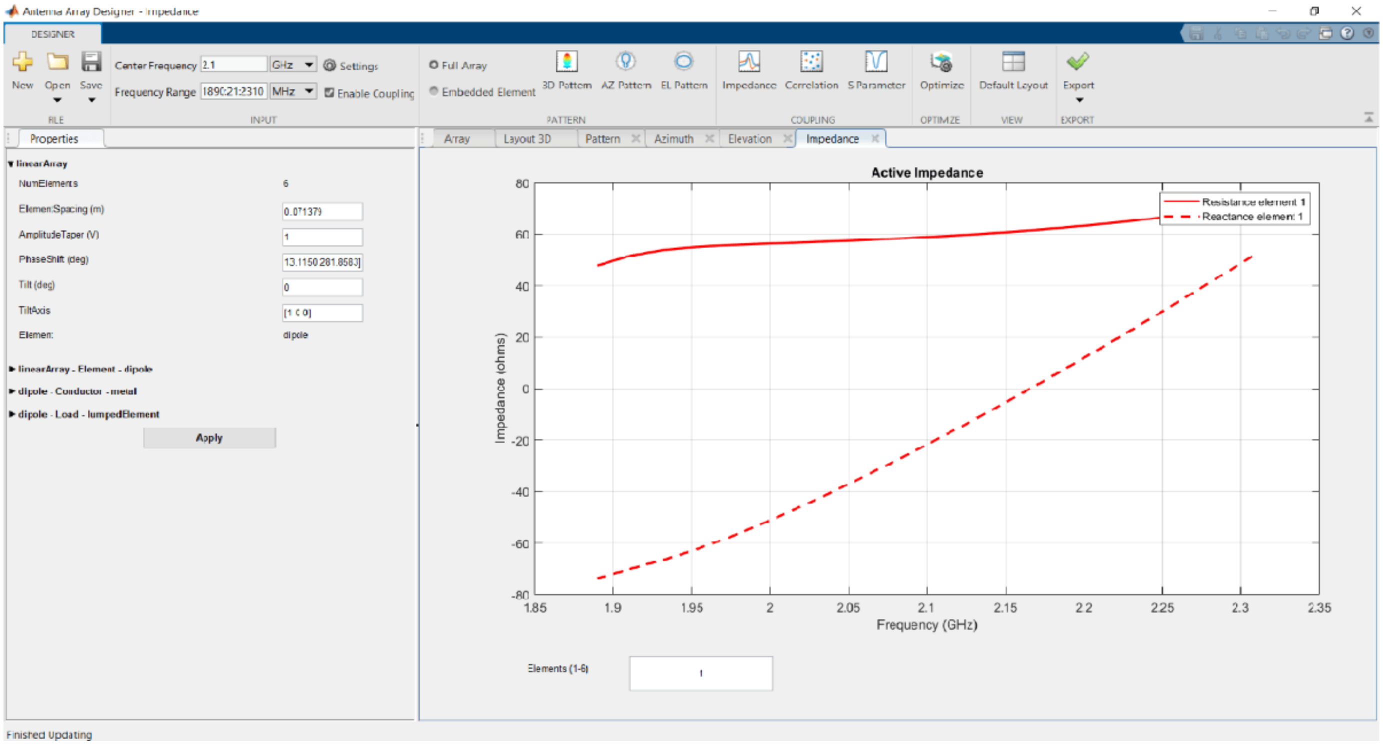


Coupling Analysis

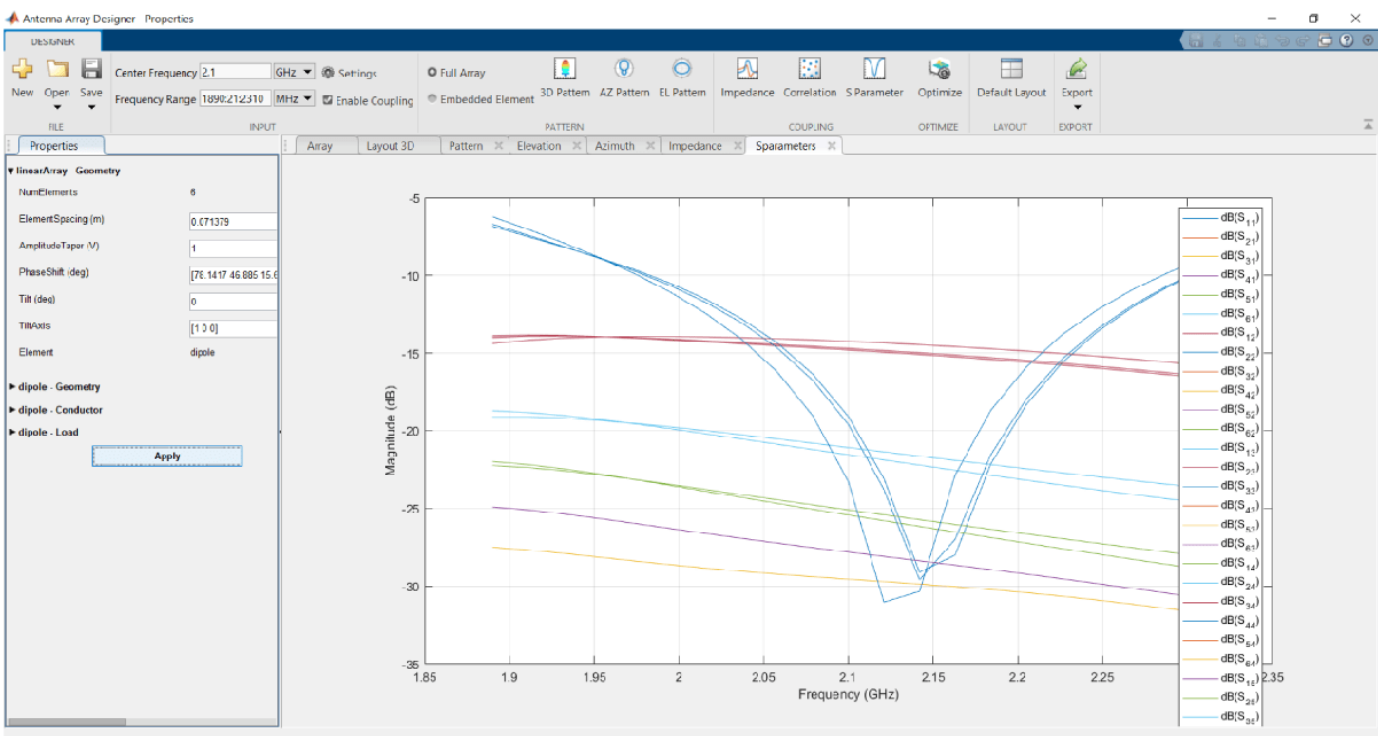
Plot Impedance and S-Parameter

In the toolstrip under **COUPLING** section, click **Impedance** to plot the impedance of each element. Change the edit field **Element** to plot the impedance of a different element.

The default reference impedance for S-Parameters plot is 50 Ohms. To change this value, click on **Settings** under the **INPUT** section of toolstrip. Change the **Ref Impedance(Z0)**value to **75** Ohms and click **Ok**.

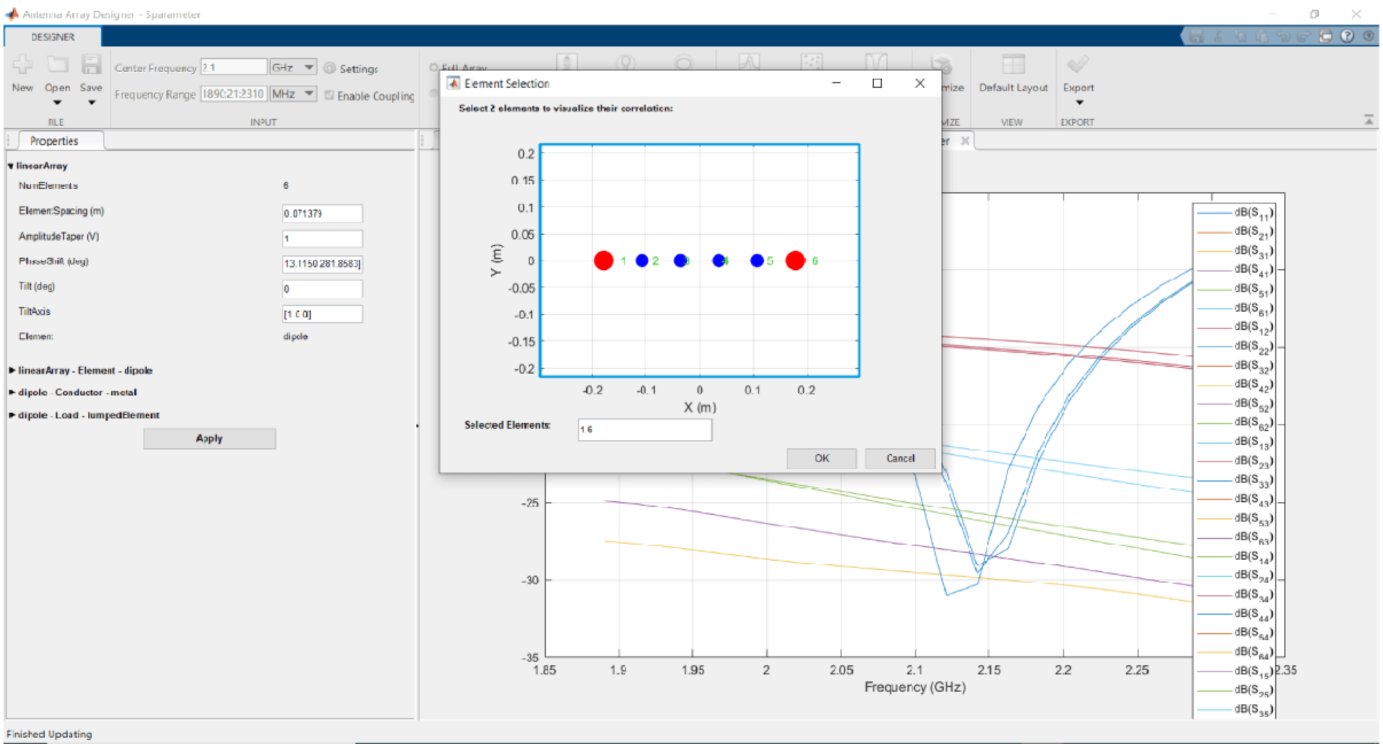


Click **S-Parameter**, under **COUPLING** section to plot the 6 port S-parameters of the array.

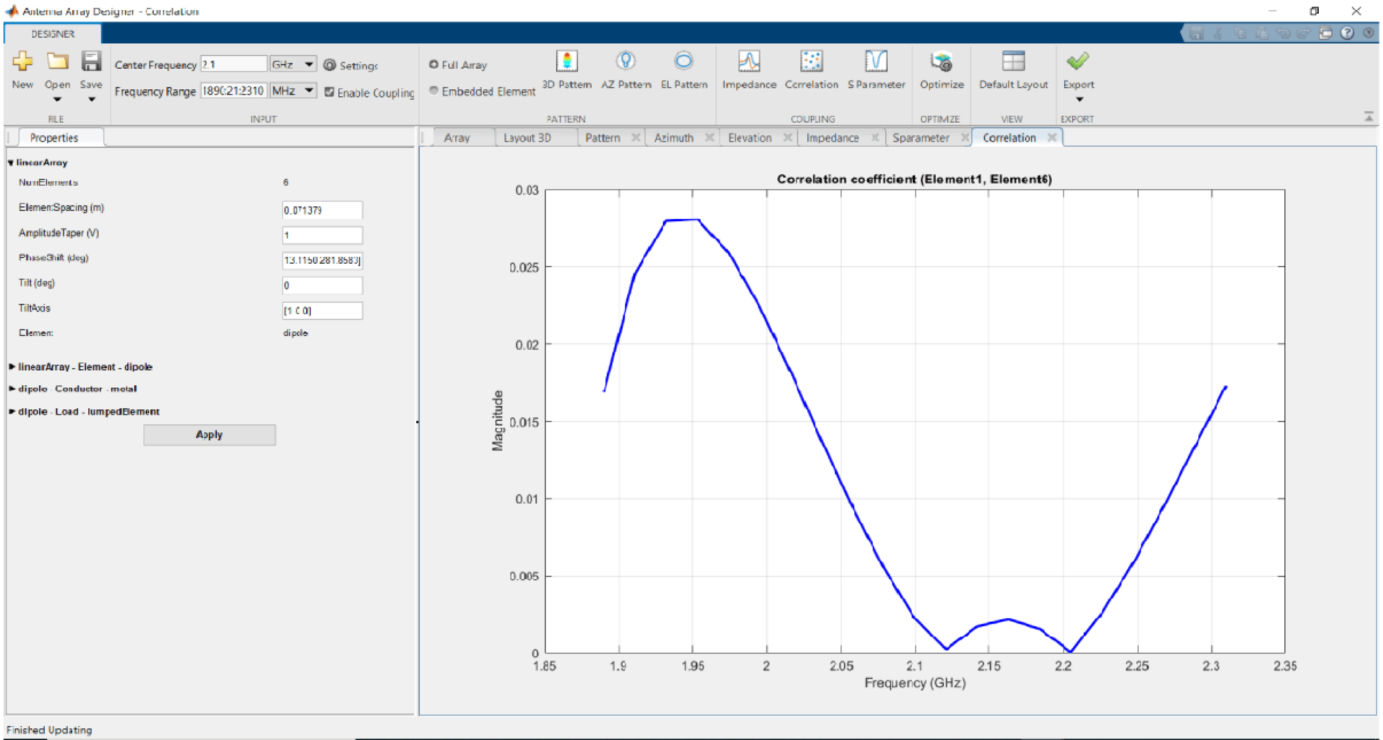


Correlation Analysis

In the toolstrip, under **COUPLING** section click on **Correlation.** In the **Element Selection** window pop-up, click on 1 and 6 to select element **1** and element of the array.



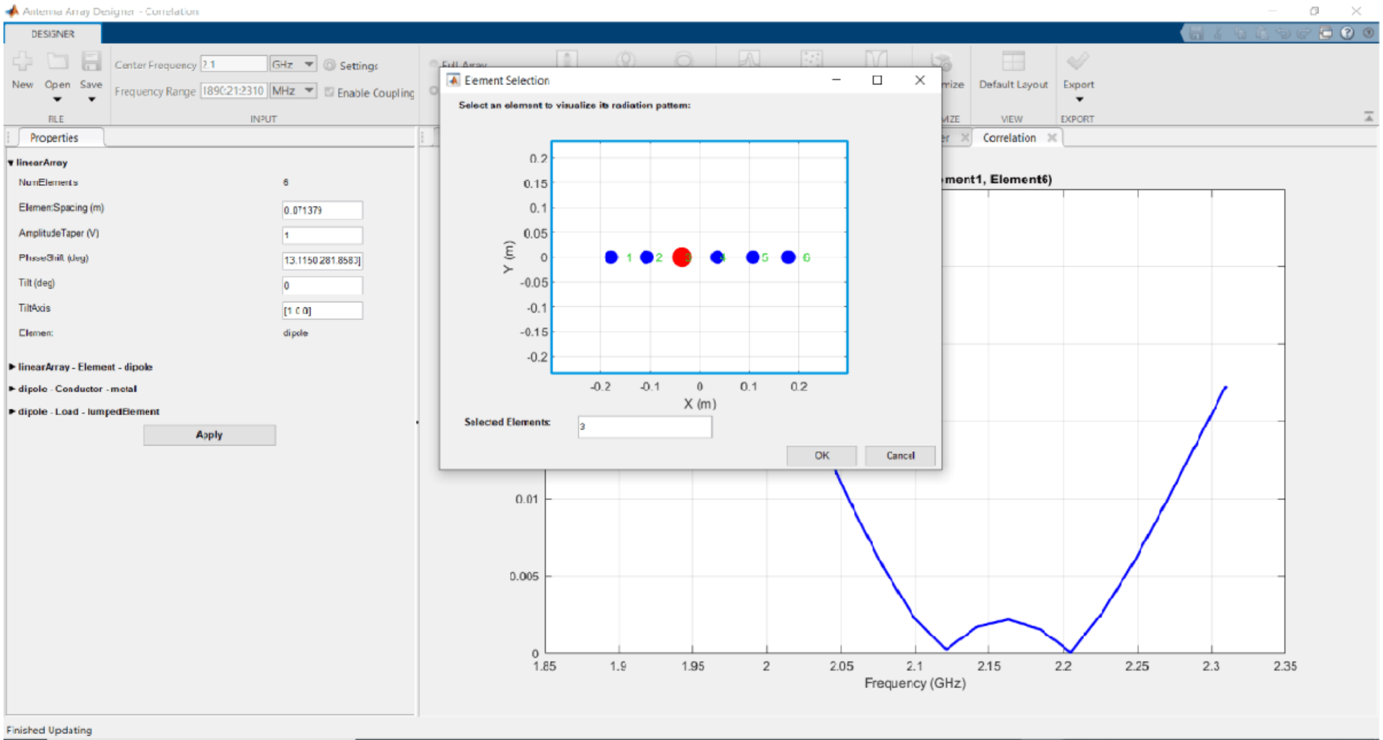
Click **OK.**



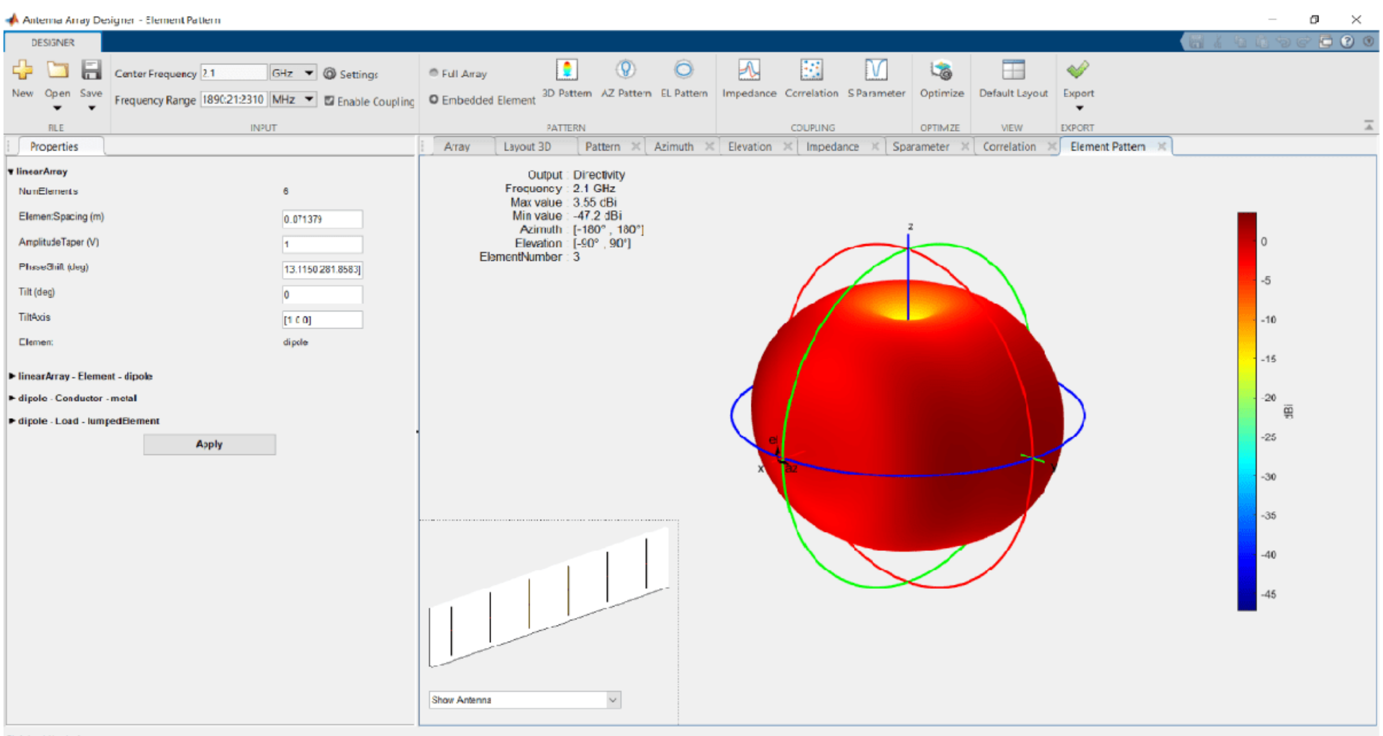
Embedded Element Analysis

Plot 3D, Azimuth and Elevation Pattern

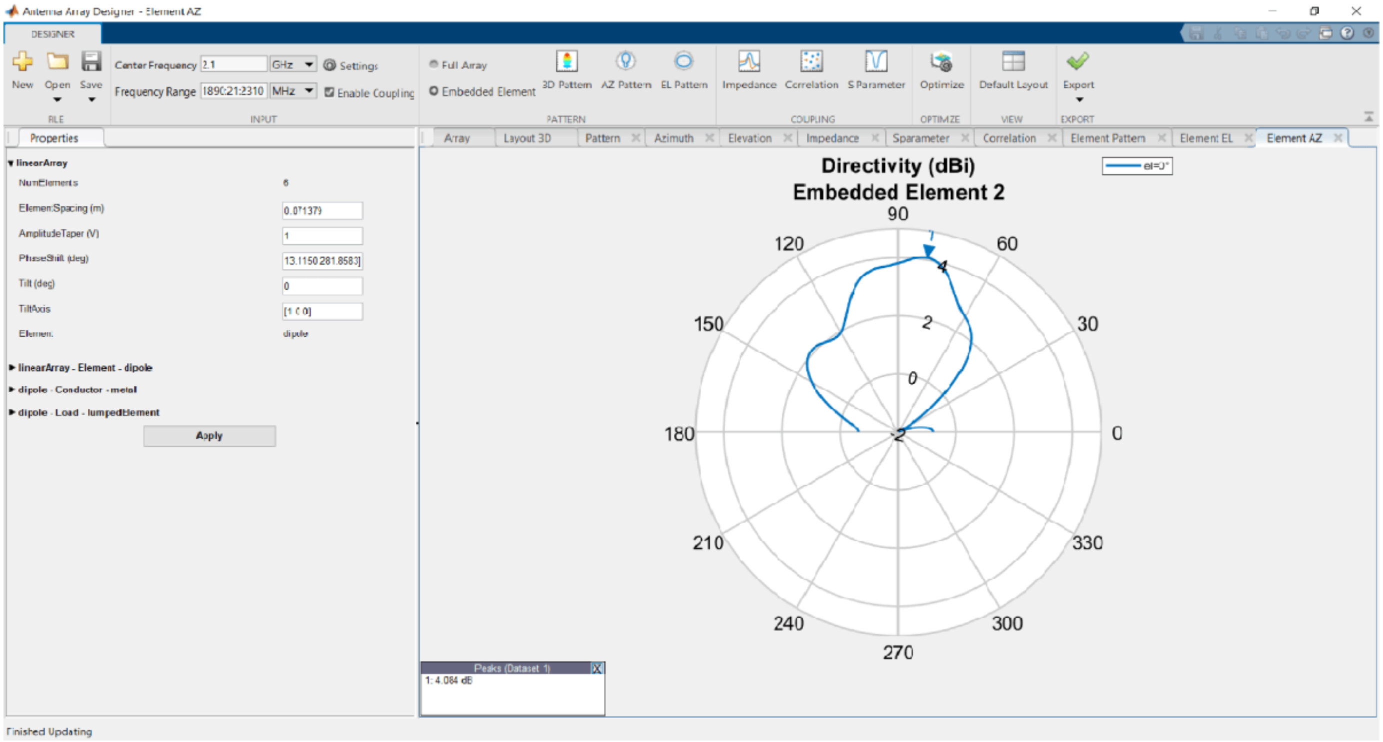
To visualize the radiation properties of an individual element in the array, click on **Embedded Element** under **PATTERN** section of the toolstrip. Click on **3D Pattern**. An **Element Selection** window opens which allows to choose an element from the array.

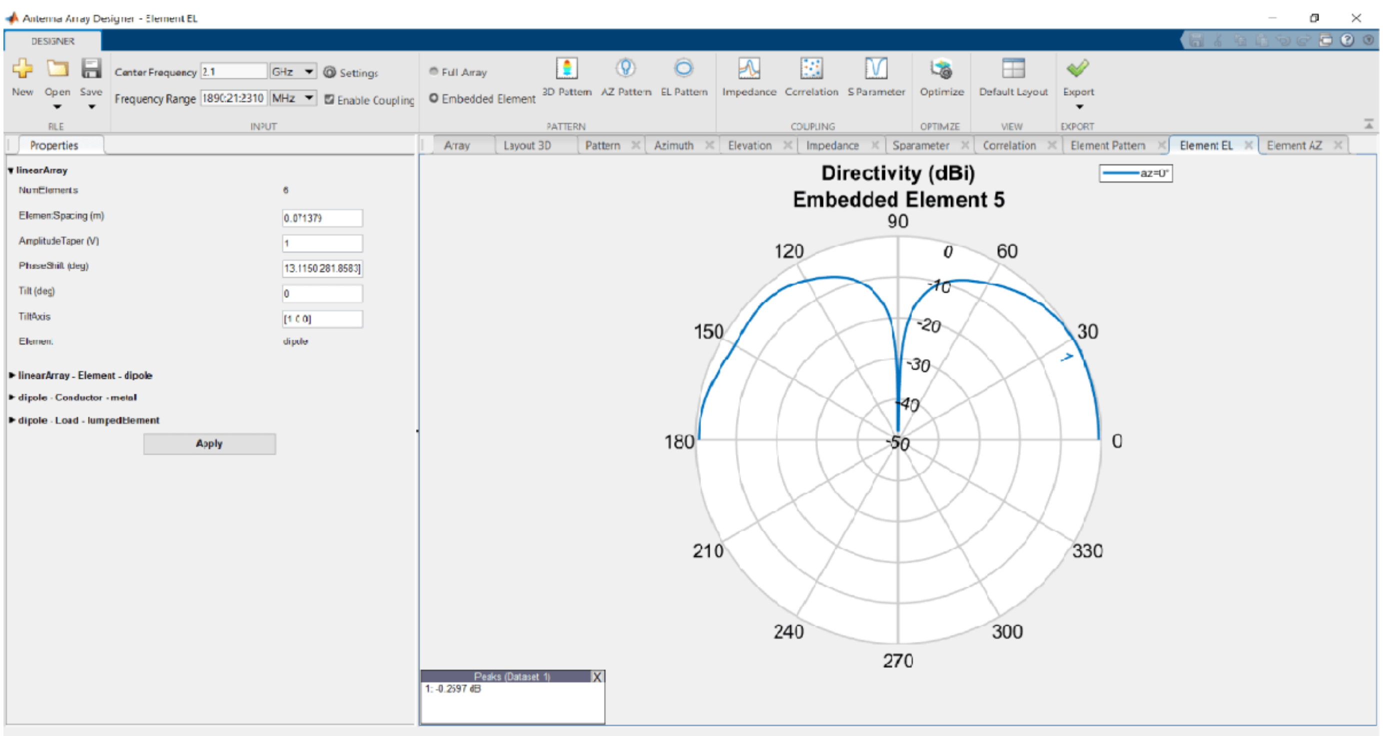


Select element 3 to visualize its 3D radiation pattern. Click **OK**. A new figure named **Element Pattern** is displayed



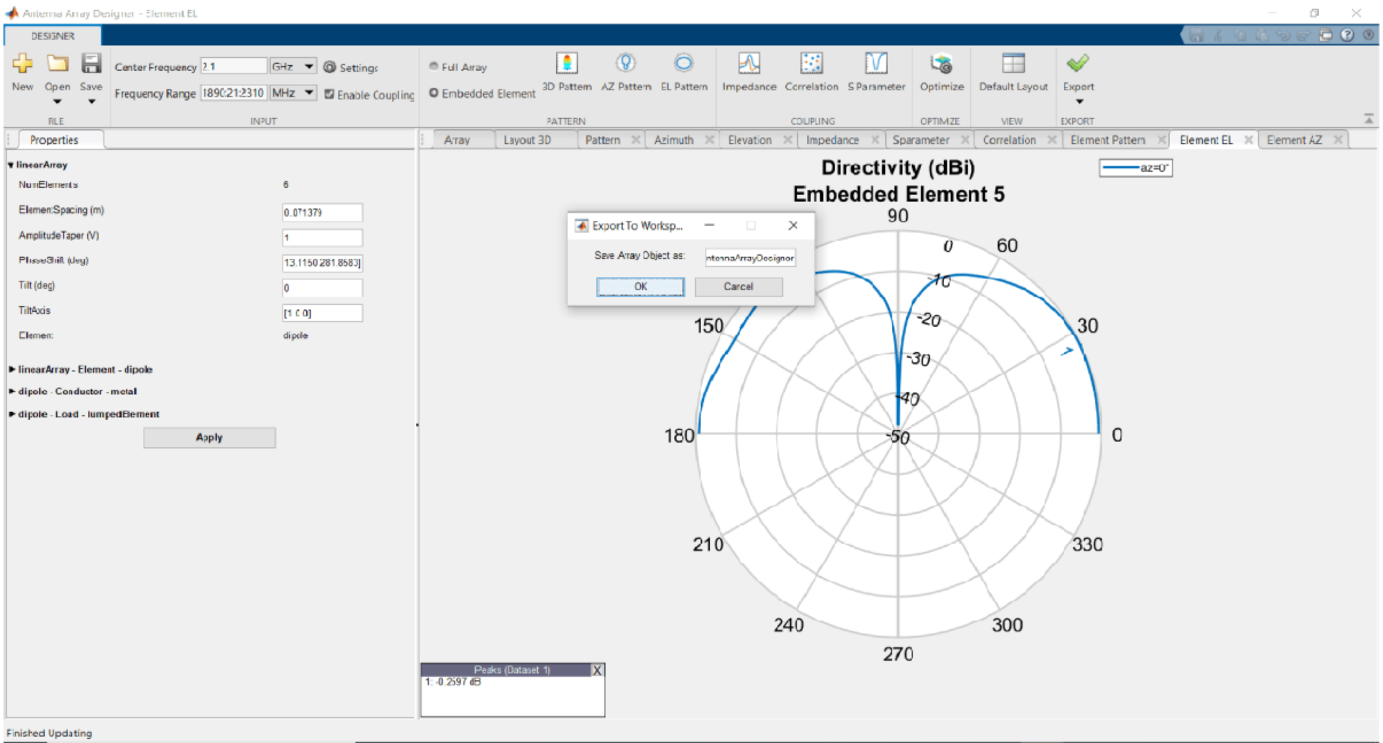
Click on **AZ Pattern**, and select the element **5**. Similarly, click on **EL Pattern** and select element **2**.


Antenna Array Designer - Element AZ
DESIGNER
New
Open
Save
Center Frequency
2.1
GHz
Settings
Frequency Range
MHz
Enable Coupling
FILE
INPUT
Full Array
Embedded Element
3D Pattern
AZ Pattern
EL Pattern
PATTERN
Impedance
Correlation
S Parameter
COUPLING
Optimize
OPTIMIZE
Default Layout
VIEW
Export
EXPORT
Properties
Array
Layout 3D
Pattern
Azimuth
Elevation
Impedance
Sparameter
Correlation
Element Pattern
Element EL
Element AZ
linearArray
NumElements
6
ElementSpacing (m)
AmplitudeTaper (V)
1
PhaseShift (deg)
Tilt (deg)
0
TiltAxis
Element
dipole
linearArray - Element - dipole
dipole - Conductor - metal
dipole - Load - lumpedElement
Apply
Directivity (dBi)
Embedded Element 2
el=0°
Peaks (Dataset 1)
1: 4.084 dB
Finished Updating


Antenna Array Designer - Element EL
DESIGNER
New
Open
Save
Center Frequency
2.1
GHz
Settings
Frequency Range
MHz
Enable Coupling
FILE
INPUT
Full Array
Embedded Element
3D Pattern
AZ Pattern
EL Pattern
PATTERN
Impedance
Correlation
S Parameter
COUPLING
Optimize
OPTIMIZE
Default Layout
VIEW
Export
EXPORT
Properties
Array
Layout 3D
Pattern
Azimuth
Elevation
Impedance
Sparameter
Correlation
Element Pattern
Element EL
Element AZ
linearArray
NumElements
6
ElementSpacing (m)
AmplitudeTaper (V)
1
PhaseShift (deg)
Tilt (deg)
0
TiltAxis
Element
dipole
linearArray - Element - dipole
dipole - Conductor - metal
dipole - Load - lumpedElement
Apply
Directivity (dBi)
Embedded Element 5
az=0°
Peaks (Dataset 1)
1: -0.2397 dB
Finished Updating

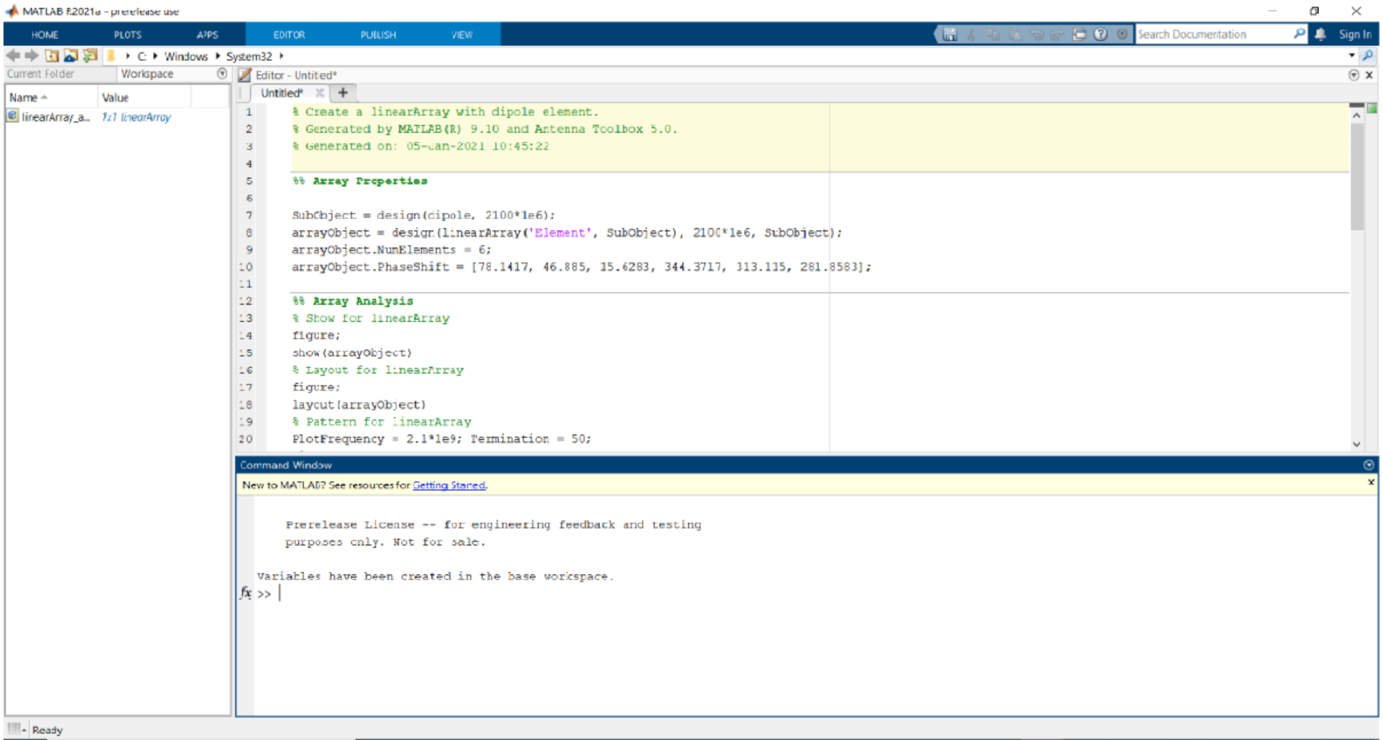
Export to MATLAB Workspace

Click the **Export** button arrow and then click **Export to workspace**. In the **Export to workspace** window, give a name to the array that you've designed. Click on the variable in the workspace to view the properties of the antenna array.



Export to MATLAB Script

Click the **Export** button arrow again and then click **Export to script** to view the linear array and analysis in MATLAB script format. The script has two sections: **Array Properties** and **Array Analysis.**



See Also

"Array Modeling and Analysis" on page 2-2 | "Optimization of Antenna Array Elements Using Antenna Array Designer App"

Introduction to RF Propagation

RF Propagation and Visualization

RF propagation models describe the behavior of signals as they travel through the environment. You can display transmitter sites, receiver sites, and RF propagation visualizations by using Site Viewer, an interactive 3-D viewer. Site Viewer enables you to visualize propagation models in both outdoor and indoor environments.

Visualize Outdoor Wireless Coverage

Display transmitter and receiver sites on a 3-D globe, calculate the distance and angles between the sites, and analyze the signal strength of the transmitter at the receiver site. Display a communication link, a coverage map, and a signal-to-interference-plus-noise ratio (SINR) map.

Display Sites

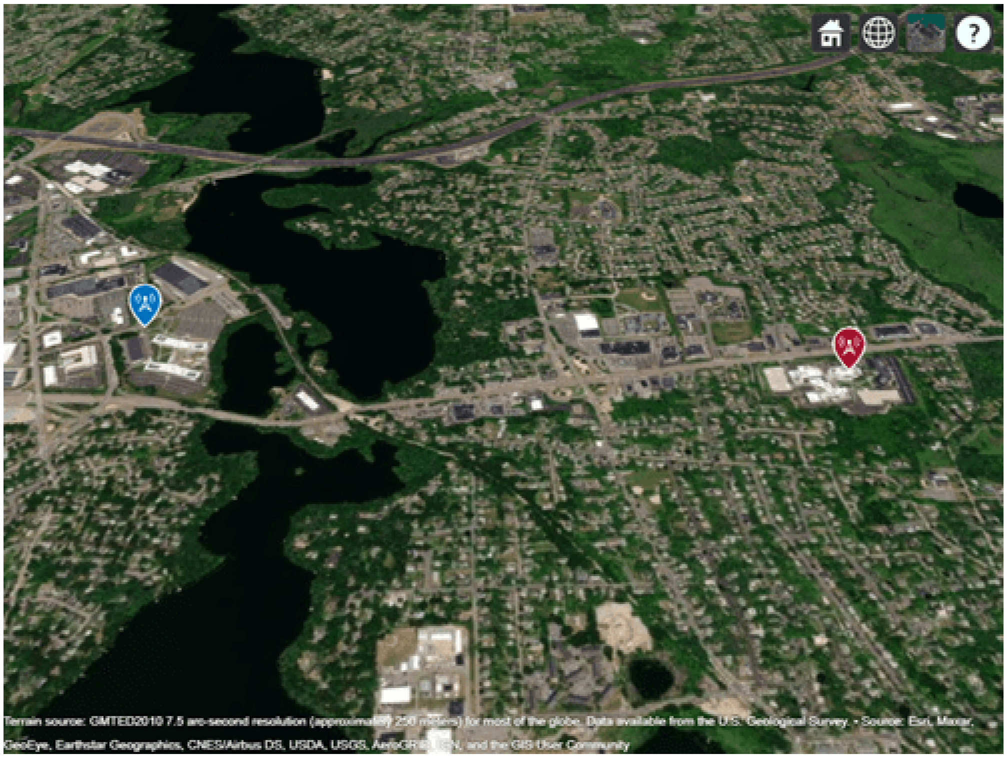
Create a transmitter site and a receiver site. Specify the position using geographic coordinates in degrees.

```
tx = txsite("Latitude",42.3001,"Longitude",-71.3504);
rx = rxsite("Latitude",42.3021,"Longitude",-71.3764);
```

Display the sites in Site Viewer. Site Viewer displays geographic sites on an interactive 3-D globe. You can customize the propagation environment of the 3-D globe by using DTED terrain and OpenStreetMap® buildings.

```
show(tx)
show(rx)
```

Pan the map by clicking and dragging. Zoom out by using the scroll wheel.



Find Distance and Angles

Calculate the distance between the sites in meters. By default, the `distance` function calculates the distance along a straight line between the sites. This straight-line path is called the Euclidean path and ignores all obstructions, including the Earth.

```
dm = distance(tx,rx)
```

```
dm = 2.1556e+03
```

You can also calculate distance using a great circle path, which considers the curvature of the Earth.

Calculate the azimuth and elevation angles between the sites. For geographic sites, the `angle` function returns the azimuth angle in degrees, measured counterclockwise from the east. The `angle` function returns the elevation angle in degrees from the horizontal plane.

```
[az,el] = angle(tx,rx)
```

```
az = 174.0753
```

```
el = -0.7267
```

Analyze Signal Strength

The signal strength of a transmitter at a receiver site is given by the following equation:

$$P_{\text{rx}} = P_{\text{tx}} + G_{\text{tx}} + G_{\text{rx}} - \text{pathloss}$$

where:

- P_{rx} is the power available at the receiver.
- P_{tx} is the transmitter output power.
- G_{tx} is the transmitter gain.
- G_{rx} = is the receiver gain.
- pathloss is the RF attenuation suffered by the transmitter signal when it arrives at the receiver.

Calculate the signal strength at the desk receiver site. By default, the `sigstrength` function calculates signal strength in power units (dBm). You can also calculate the signal strength in electric field strength units (dBµV/m).

```
ss = sigstrength(rx,tx)
```

```
ss = -67.0767
```

The link margin measures the robustness of the communication link. Calculate the link margin by subtracting the required receiver sensitivity from the signal strength.

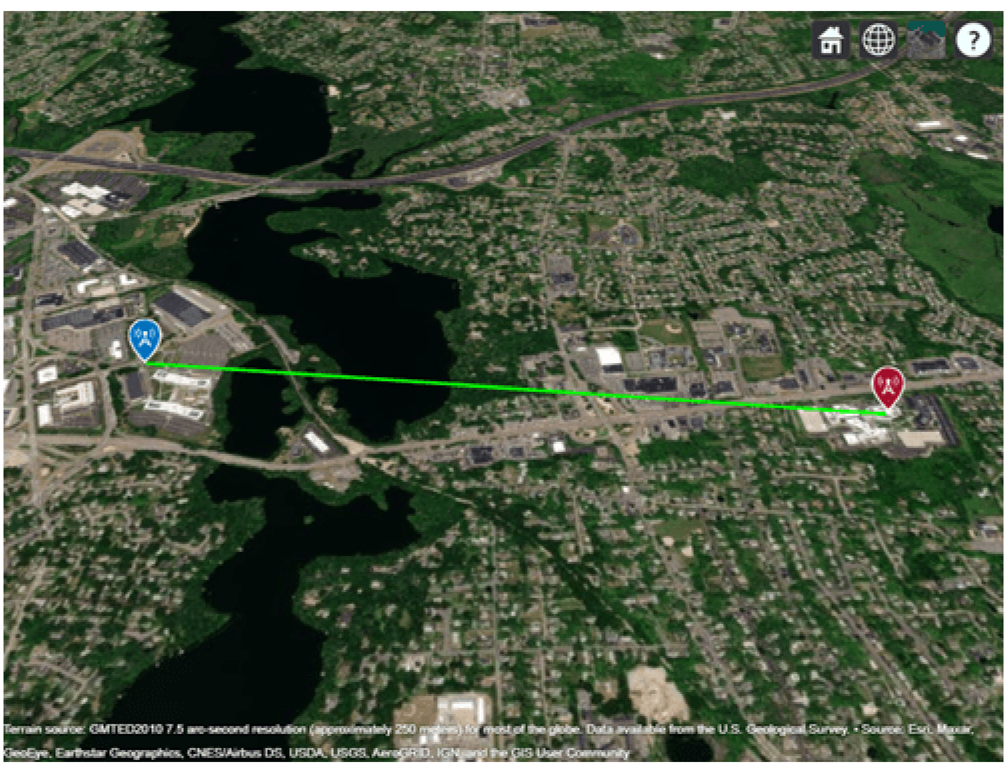
```
margin = abs(rx.ReceiverSensitivity - ss)
```

```
margin = 32.9233
```

Display Communication Link

Display the communication link status between the sites. The success of the link depends on the power received by the receiver from the transmitter. By default, a green line indicates that the received power meets or exceeds the receiver sensitivity. A red line indicates unsuccessful communication.

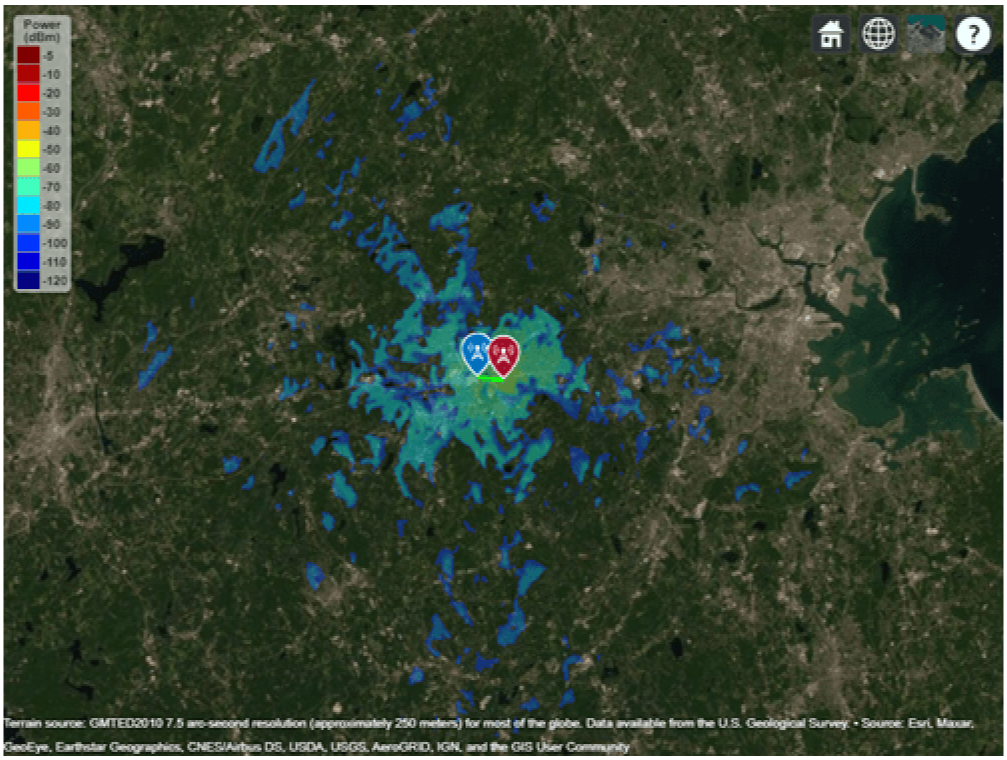
```
link(rx,tx)
```



Display Coverage Map

Display the coverage map of the transmitter. A coverage map visualizes the service area of the transmitter, which is where the received signal strength for a reference receiver meets its sensitivity. You can create coverage maps that depict signal strength as either a power quantity (typically dBm) or a voltage quantity (typically dBµV/m).

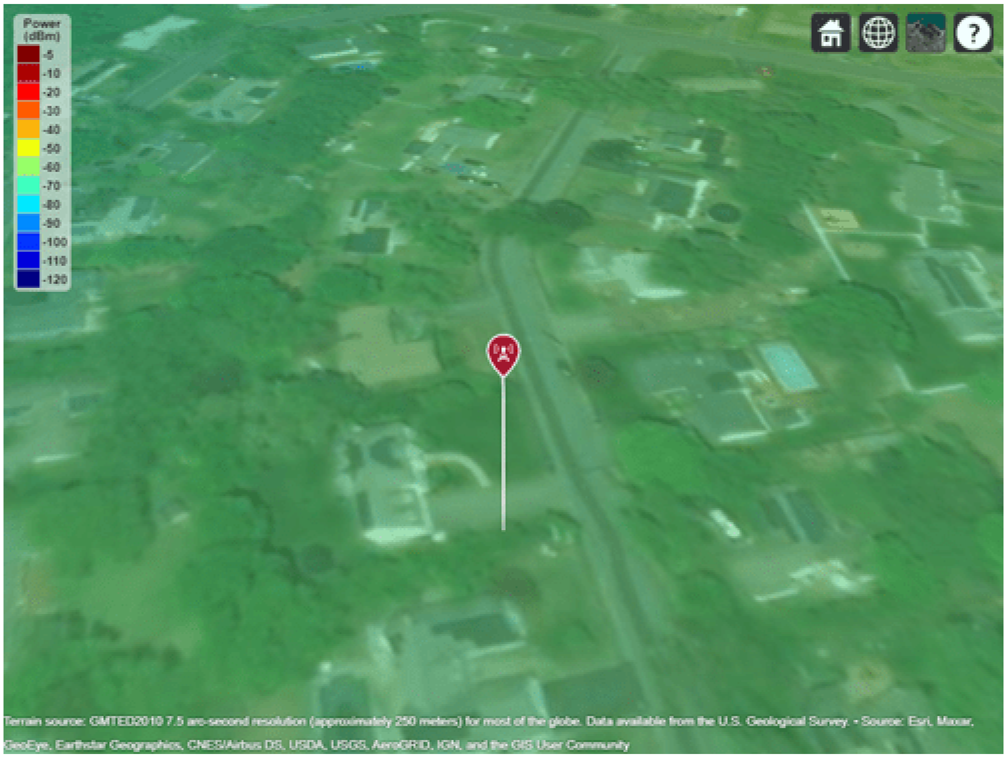
```
coverage(tx,"SignalStrengths",-100:5:-60)
```



Find New Transmitter Site

Create and display a new transmitter site that is 1 km north of the existing transmitter site. Specify the antenna height as 30 m.

```
[lat,lon] = location(tx,1000,90);
tx2 = txsite("Latitude",lat,"Longitude",lon,"AntennaHeight",30);
show(tx2)
```



Calculate SINR

Calculate the SINR in decibels. The SINR of a receiver is given by the following equation:

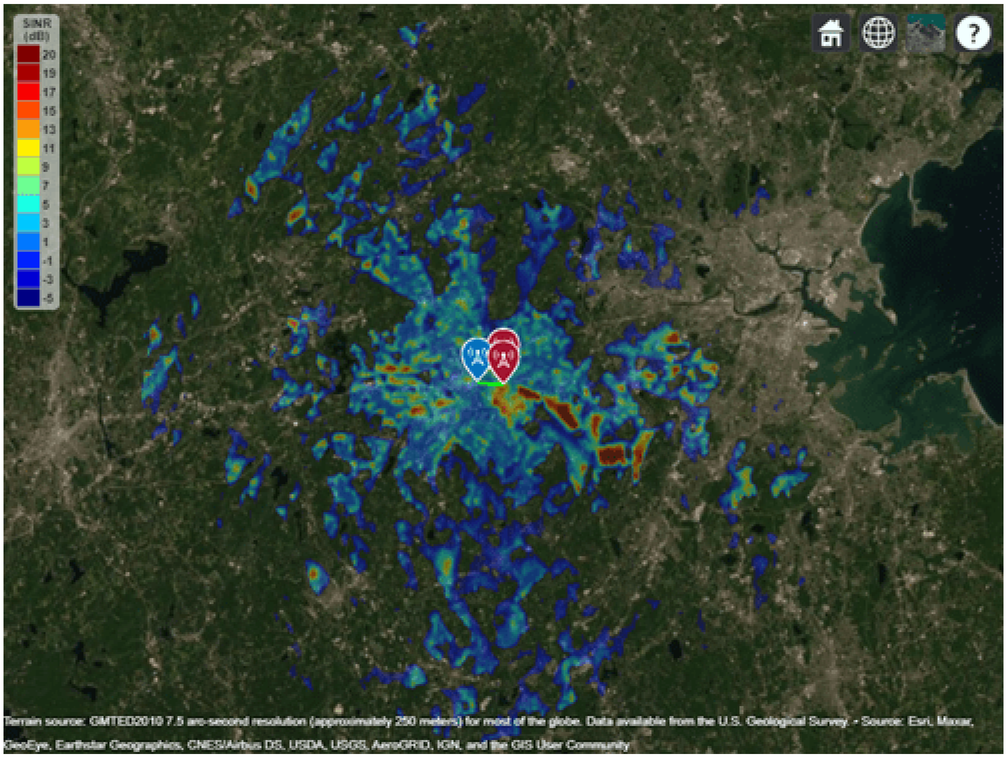
$$\text{SINR} = \frac{S}{I + N}$$

where:

- S is the received power of the signal of interest.
- I is the received power of interfering signals in the network.
- N is the total received noise power.

When Site Viewer has terrain data, the `sinr` function incorporates the terrain into the calculations.

```
sinr([tx,tx2])
```



Visualize Indoor Propagation Paths

Import a 3-D scene model of a conference room. Display sites and find propagation paths between the sites.

Import Scene

Import and view an STL file. The file models an indoor office with a conference room and open space separated by a partial wall. STL files contain geometry information and do not contain information about colors, surfaces, or textures.

```
viewer = siteviewer("SceneModel","office.stl","ShowOrigin",false);
```

Display Sites

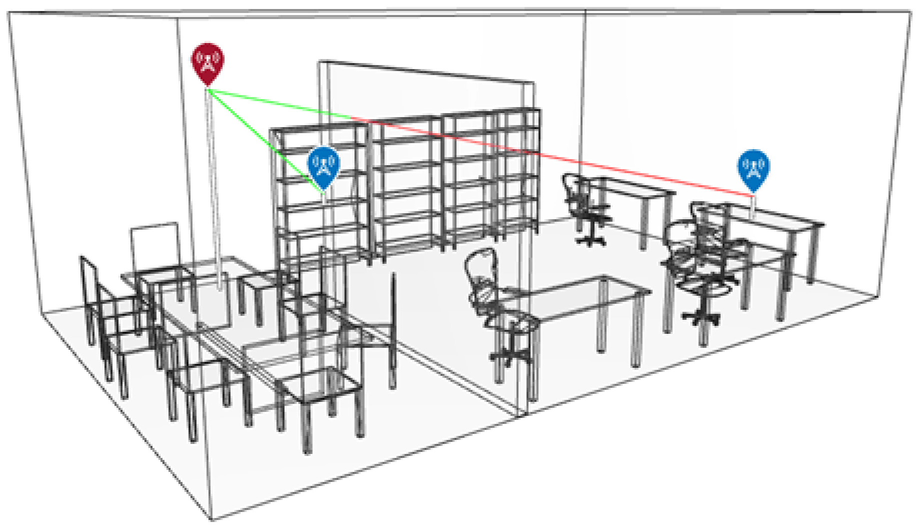
Place one transmitter near the ceiling in the conference room. Place one receiver on a desk in the open space and another receiver on a shelf. Specify the position using Cartesian coordinates in meters.

```
tx = txsite("cartesian","AntennaPosition",[2; 1.3; 2.5]);
rx_desk = rxsite("cartesian","AntennaPosition",[3.6; 7.5; 1]);
rx_shelf = rxsite("cartesian","AntennaPosition",[0.4; 3.3; 1]);
```

Display the receivers and the line-of-sight paths.

```
los(tx,[rx_desk rx_shelf])
```

Pan the scene by left-clicking, zoom by right-clicking or by using the scroll wheel, and rotate by clicking the middle button and dragging or by pressing **Ctrl** and left-clicking and dragging.



The path to the shelf receiver is clear and the path to the desk receiver is obstructed.

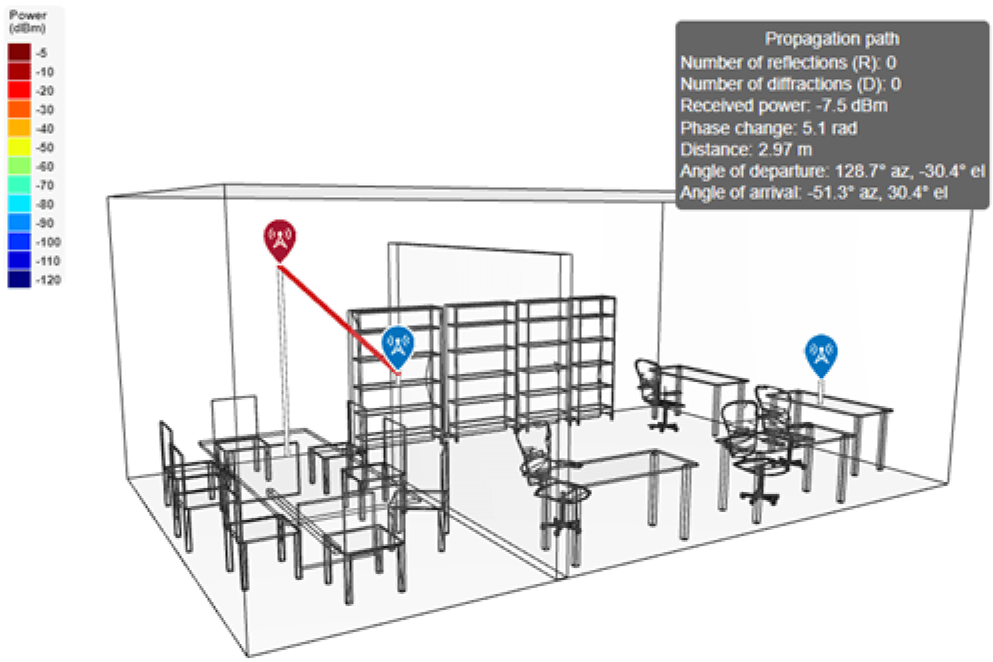
Display Propagation Paths

Create a ray tracing propagation model, which MATLAB® represents using a `RayTracing` object. Configure the model to use a Cartesian coordinate system and wooden surface materials. By default, the model uses the shooting and bouncing rays (SBR) method.

```
pm = propagationModel("raytracing", ...
    "CoordinateSystem","cartesian", ...
    "SurfaceMaterial","wood");
```

Display propagation paths that are within the line of sight by setting the `MaxNumReflections` property to `0`. Unlike the `los` function, the `raytrace` function does not show obstructed paths.

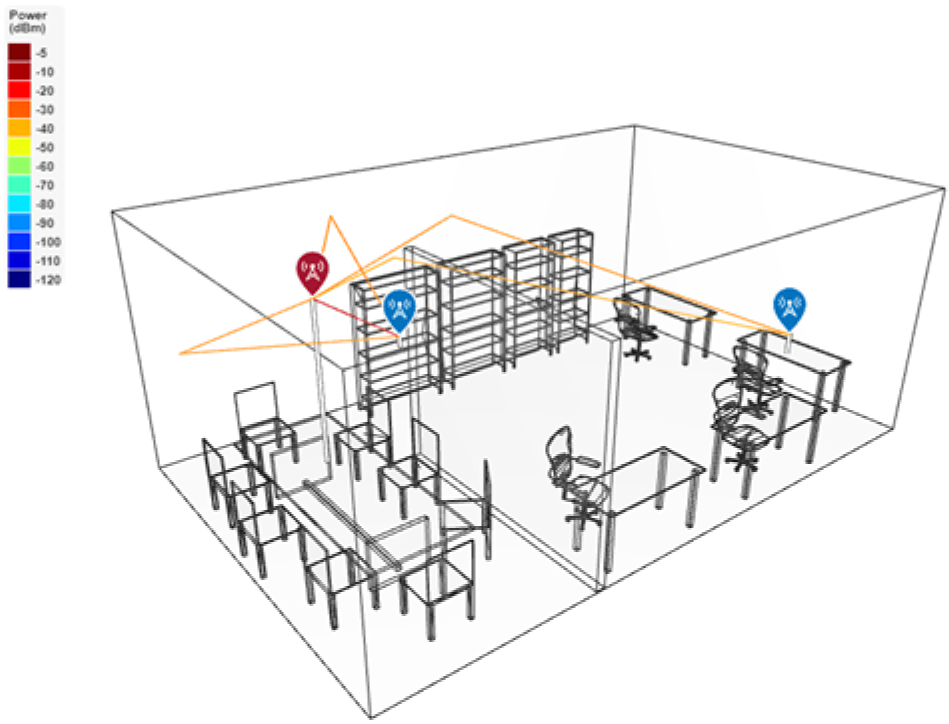
```
pm.MaxNumReflections = 0;
clearMap(viewer)
raytrace(tx,[rx_desk rx_shelf],pm)
```



The `raytrace` function finds one line-of-sight path. You can view information about the path, such as the received power, by clicking on the path.

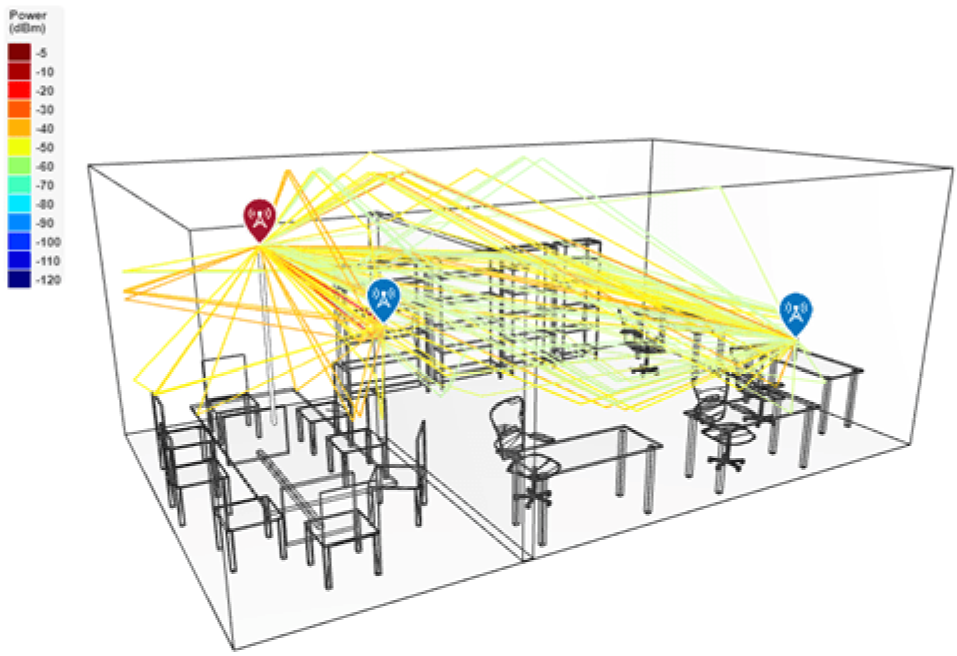
Display propagation paths with up to one reflection.

```
pm.MaxNumReflections = 1;
raytrace(tx,[rx_desk rx_shelf],pm)
```



Display propagation paths with up to one reflection and one diffraction.

```
pm.MaxNumDiffractions = 1;
raytrace(tx,[rx_desk rx_shelf],pm)
```



See Also

Functions
`coverage` | `sigstrength` | `link` | `sinr` | `raytrace`

Objects
`siteviewer` | `txsite` | `rxsite`

More About

- "Planning Radar Network Coverage over Terrain"
- "Visualize Antenna Field Strength Map on Earth"
- "Urban Link and Coverage Analysis Using Ray Tracing"