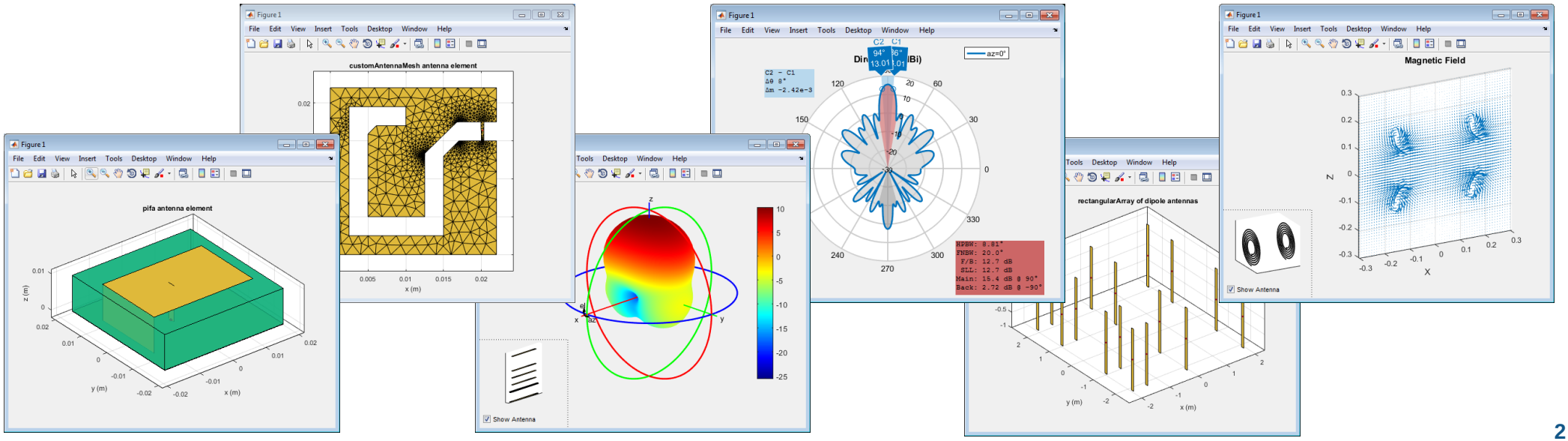
Antennas and Antenna Arrays Design and Analysis with MATLAB

Stefano Olivieri – Academia Group
Stefano.olivieri@mathworks.it

Agenda

- Introducing antenna design in MATLAB using full wave EM simulation
- Designing and analyzing your own custom antennas
- Addressing realistic antenna array modeling by including edge and coupling effects

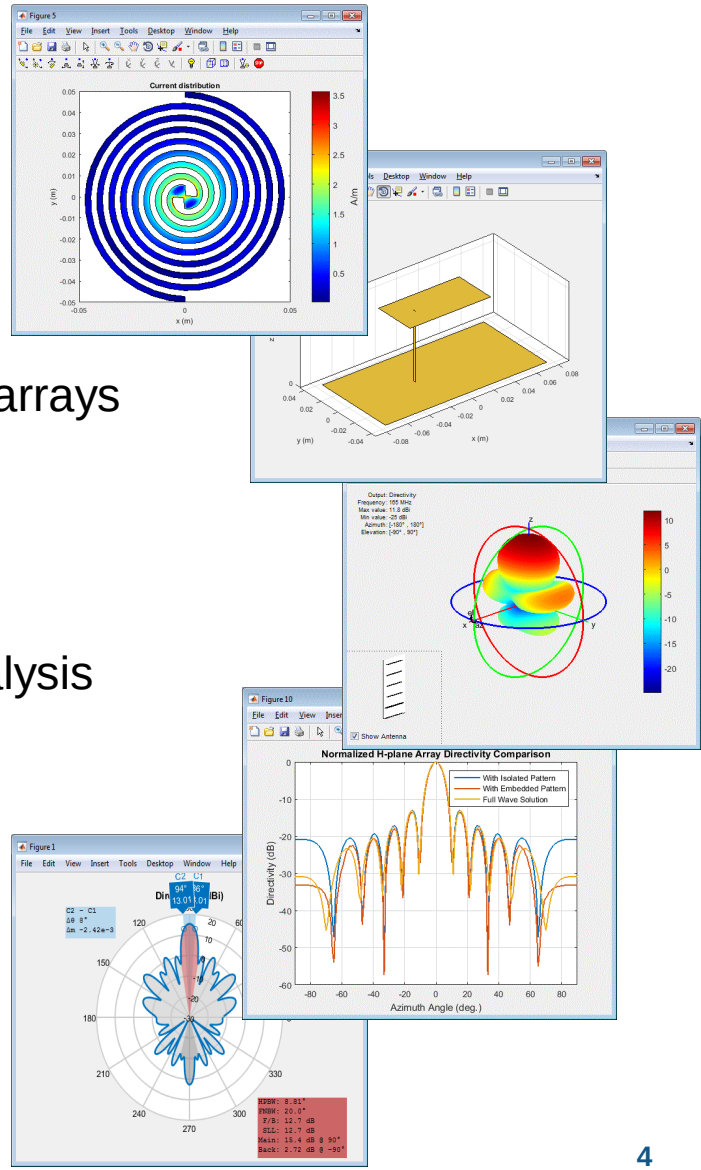


What Are the Challenges with Antenna Design?

- Understanding the requirements
  - Individual antenna parameters: frequency, directivity, geometry, material, efficiency
  - What antenna or antenna array do I use? Many types, very diverse, infinite configurations
- How to assess the antenna's performance
  - Port, field, surface analysis
  - Electromagnetic solvers: correct analysis set up
- Integrate into the system
  - Gain, impedance, coupling, leakage, pattern, feeding point … specific terminology
  - Model the antenna together with signal processing algorithms

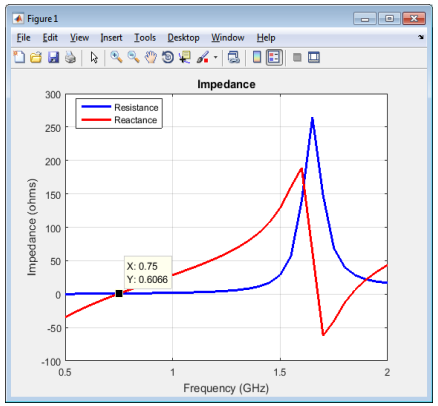
Antenna Toolbox

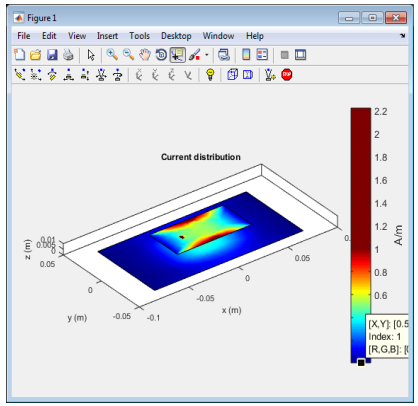
- Easy design
  - Library of parameterized antenna elements
  - Functionality for the design of linear and rectangular antenna arrays
  - No need for full CAD design
- Rapid simulation setup
  - Method of Moments field solver for port, field, and surface analysis
  - No need to be an EM expert
- Seamless integration
  - Model the antenna together with signal processing algorithms
  - Rapid iteration of different antenna scenarios for radar and communication systems design

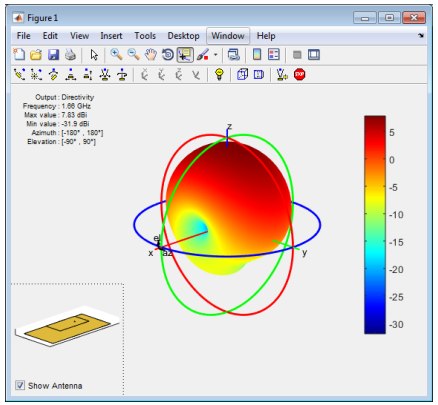


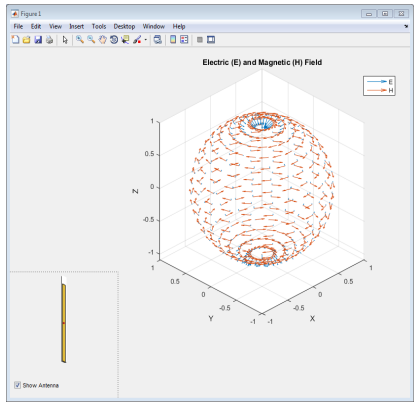
Analysis (and Visualization) Capabilities of Antenna Toolbox

- Port (RF termination characteristics)
  - Input Impedance, Resonance, Reflection Coefficient, Return Loss, Voltage Standing Wave Ratio (VSWR), Bandwidth
- Surface
  - Charge Distribution, Current Distribution
- Field
  - Radiation Pattern, Beamwidth, E-Plane and H-Plane, Polarization, Axial Ratio



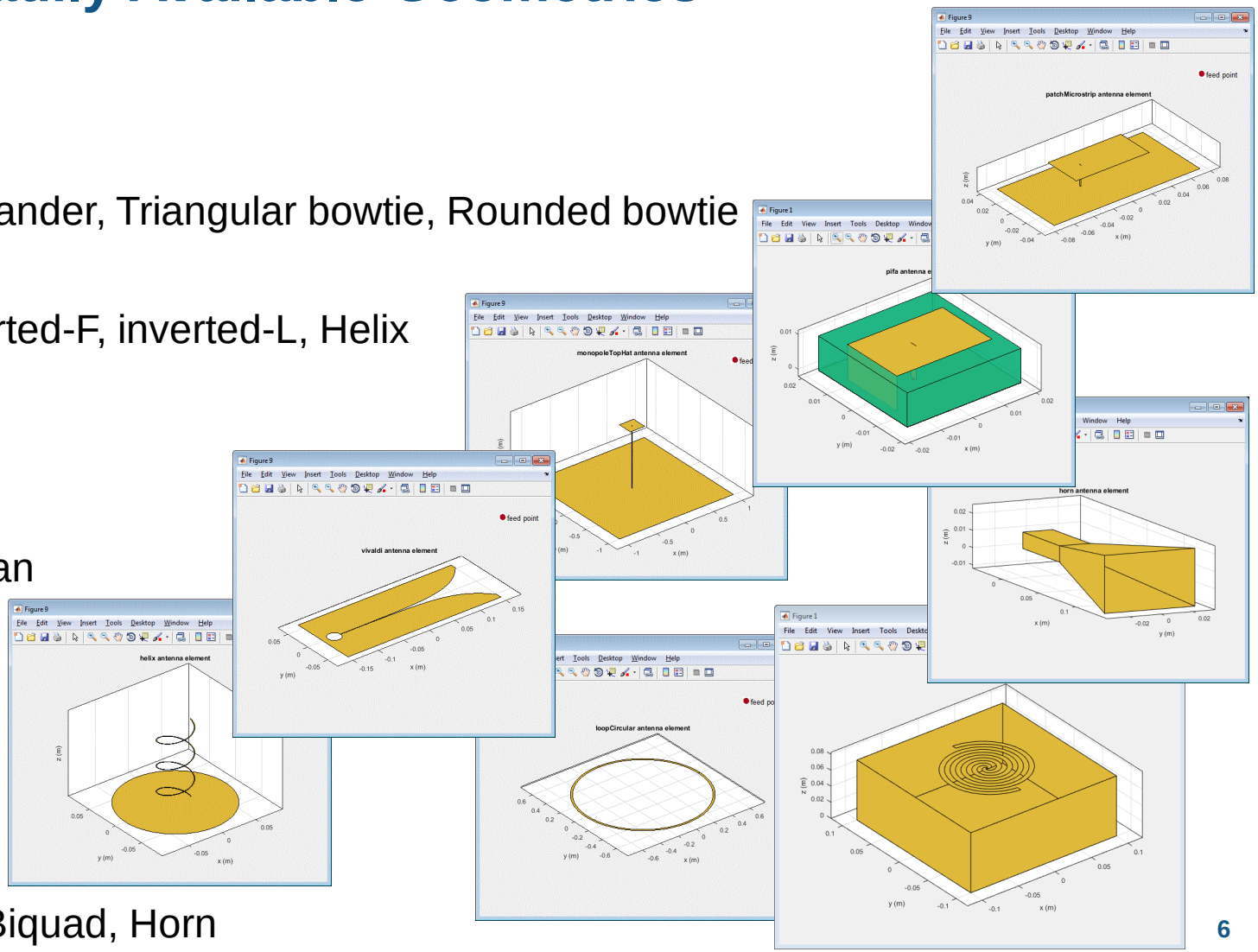






Antenna Library: Readily Available Geometries

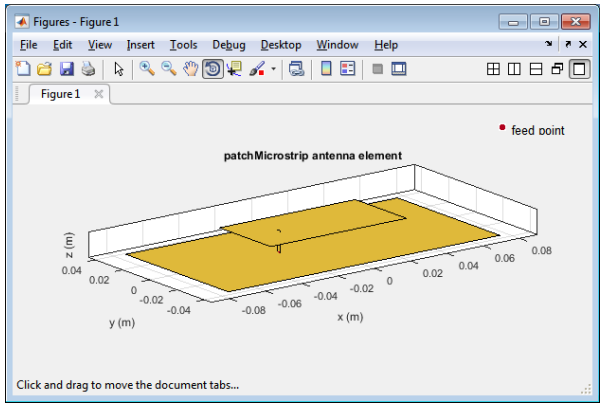
- Dipole antennas
  - Dipole, Vee, Folded, Meander, Triangular bowtie, Rounded bowtie
- Monopole antennas
  - Monopole, Top hat, Inverted-F, inverted-L, Helix
- Patch antennas
  - Microstrip patch, PIFA
- Spirals
  - Equiangular, Archimedean
- Loops
  - Circular, rectangular
- Backing structures
  - Reflector and cavity
- Other common antennas
  - Yagi Uda, Slot, Vivaldi, Biquad, Horn

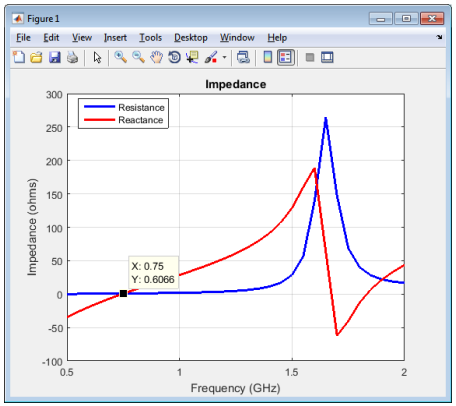


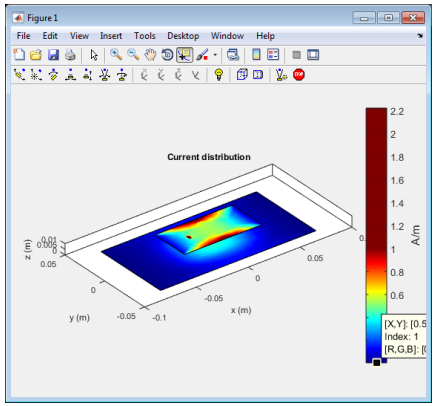
Antenna Toolbox Demo

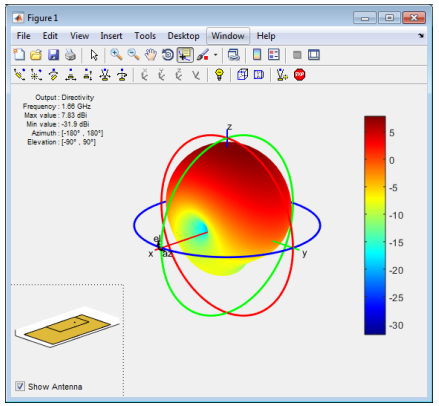
Design and analysis of one antenna element, in just 5 lines of MATLAB code

```
>> p = patchMicrostrip
>> p.Height = 0.01;
>> impedance(p, (500e6:10e6:2e9));
>> current(p, 1.66e9);
>> pattern(p, 1.66e9);
```







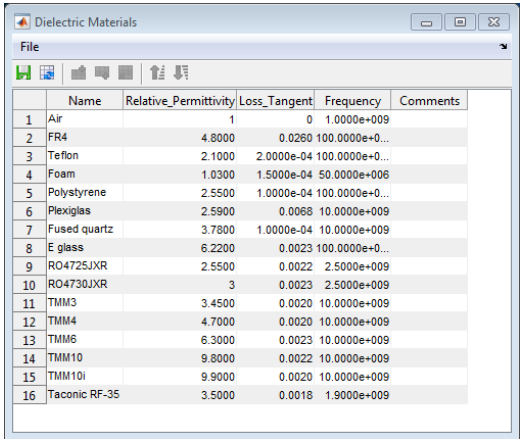


What if my Antenna is Mounted on a Dielectric Substrate?

- Antenna are often mounted on substrates
- Dielectric properties:

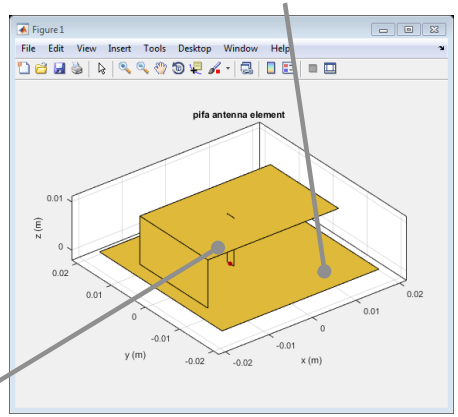
Dielectric	Relative permittivity	Loss Tangent
Air	1	0
Other	>1 (typically <10)	>0 (typically ~1e-3)

- Dielectric properties affect resonance, bandwidth, efficiency, pattern …
- Use the dielectric catalogue listing existing materials
- Define your own dielectric material

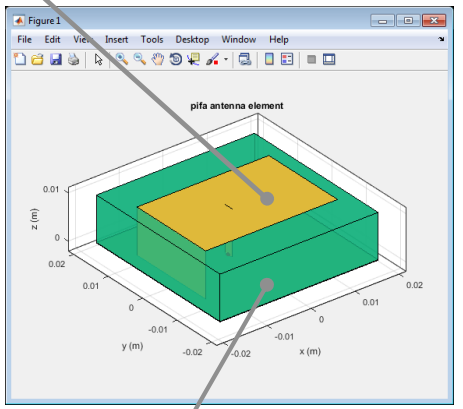


	Name	Relative_Permittivity	Loss_Tangent	Frequency	Comments
1	Air	1	0	1.0000e+009	
2	FR4	4.8000	0.0260	100.0000e+0...	
3	Teflon	2.1000	2.0000e-04	100.0000e+0...	
4	Foam	1.0300	1.5000e-04	50.0000e+006	
5	Polystyrene	2.5500	1.0000e-04	100.0000e+0...	
6	Plexiglas	2.5900	0.0068	10.0000e+009	
7	Fused quartz	3.7800	1.0000e-04	10.0000e+009	
8	E glass	6.2200	0.0023	100.0000e+0...	
9	RO4725JXR	2.5500	0.0022	2.5000e+009	
10	RO4730JXR	3	0.0023	2.5000e+009	
11	TMM3	3.4500	0.0020	10.0000e+009	
12	TMM4	4.7000	0.0020	10.0000e+009	
13	TMM6	6.3000	0.0023	10.0000e+009	
14	TMM10	9.8000	0.0022	10.0000e+009	
15	TMM10i	9.9000	0.0020	10.0000e+009	
16	Taconic RF-35	3.5000	0.0018	1.9000e+009	

Free space (isolation)



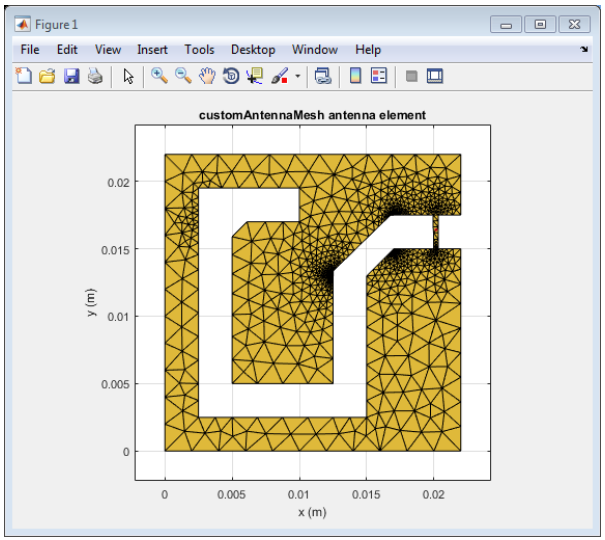
“metal” antenna (ideal conductor)



Dielectric substrate

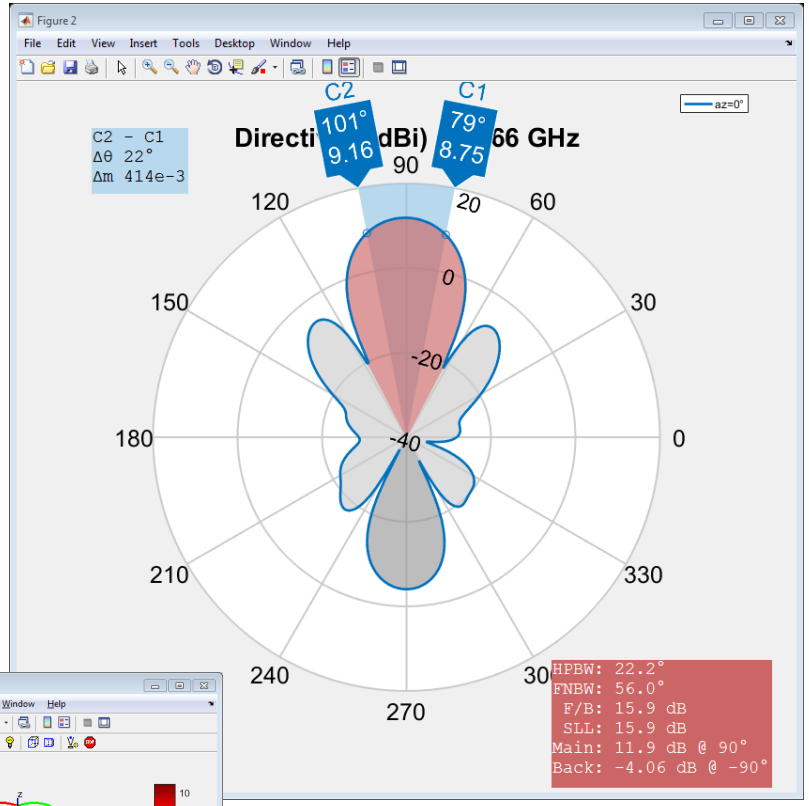
What if my Antenna is not in the Library?

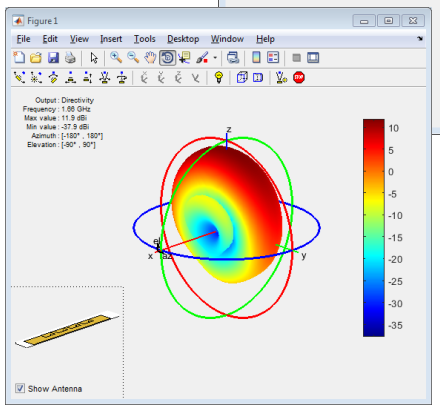
- Define your custom planar structure
  - Define the antenna geometry using PDE Toolbox
  - Define the mesh using MATLAB `delaunayTriangulation`
  - Use third party tools to generate a mesh structure
- Import 2D mesh with Antenna Toolbox
  - Define the feeding point
  - Analyse the antenna
- Integrate your custom antenna
  - Define a backing structure
  - Define a dielectric structure
  - Build an array with custom elements



Building your First Antenna Array

```
>> a = linearArray
>> a.Element = p;
>> a.ElementSpacing = 0.1;
>> a.NumElements = 4;
>> layout(a);
>> patternElevation(a, 1.66e9,0);
```

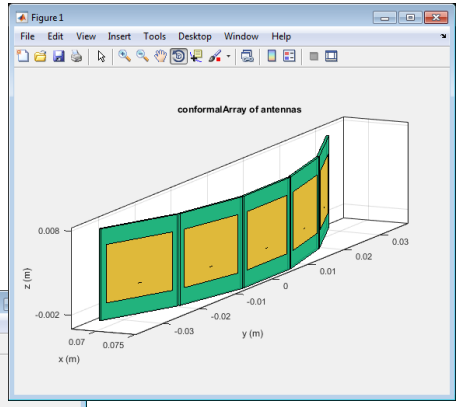


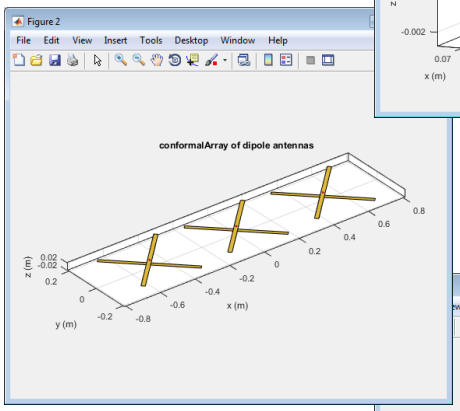


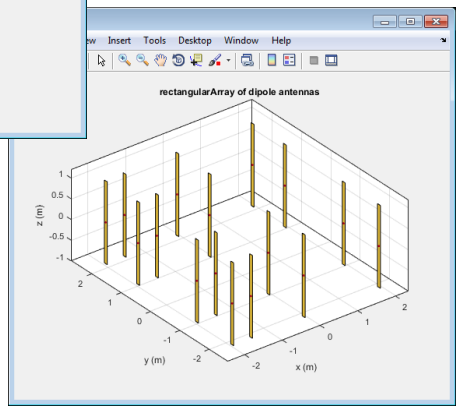
What if I Need to Customize my Array?

- Build regular arrays where you can change the properties of individual elements (rotation, size, tapering)
- Describe conformal (heterogeneous) arrays in terms of element type and arbitrary position

```
>> arr = conformalArray;
>> d = dipole;
>> b = bowtieTriangular;
>> arr.Element = {d, b};
>> arr.ElementPosition(1,:) = [0 0 0];
>> arr.ElementPosition(2,:) = [0 0.5 0];
```



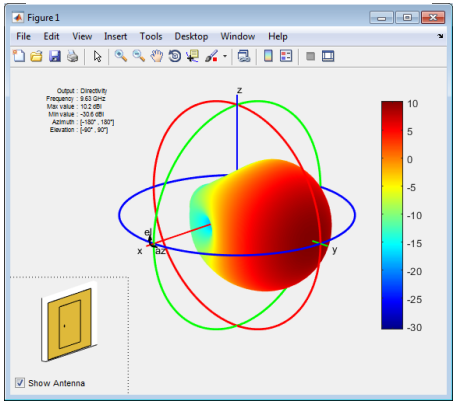




Computing the Radiation Pattern of Antenna Arrays

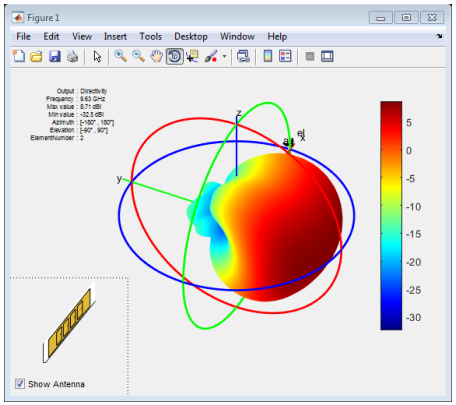
- Antenna Toolbox arrays perform full wave EM analysis
  - Isolated element vs embedded element vs full array

Isolated element



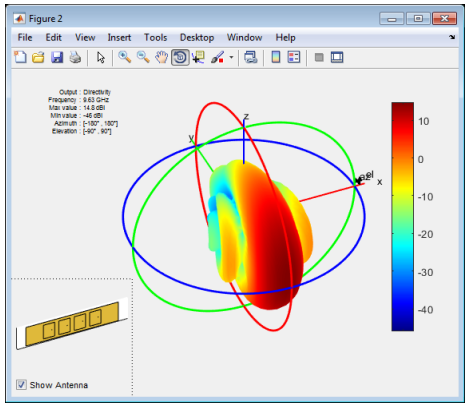

```
pattern(p, 10e9);
```

Embedded element




```
pattern(l, 10e9, ...
 'ElementNumber',2);
```

Full wave



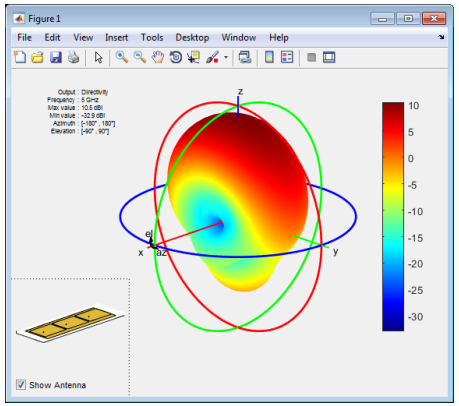

```
pattern(l, 10e9);
```

Antenna Toolbox Full Wave EM solver

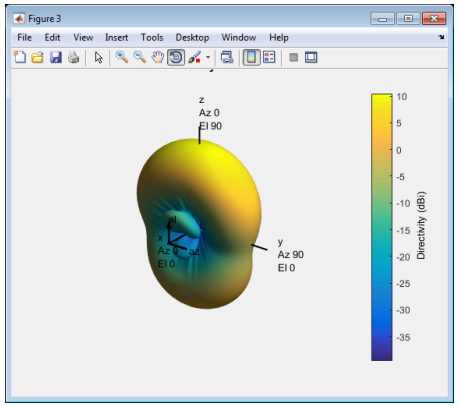
Interaction between antenna elements

- Estimate the effects of EM coupling on the pattern of each element when embedded in the array
- Estimate edge effects on the pattern of elements further away from the centre of the array
- Validate the assumption of pattern superposition by comparison with the full-wave EM solution

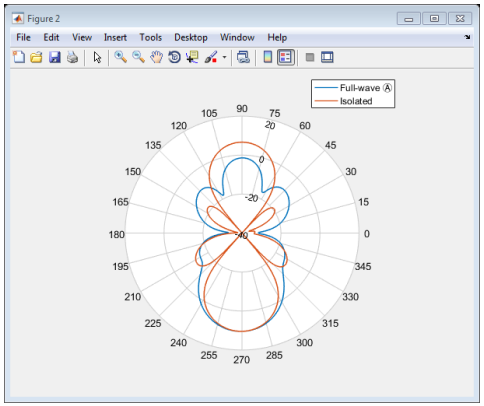
Full wave



Isolated element pattern superposition



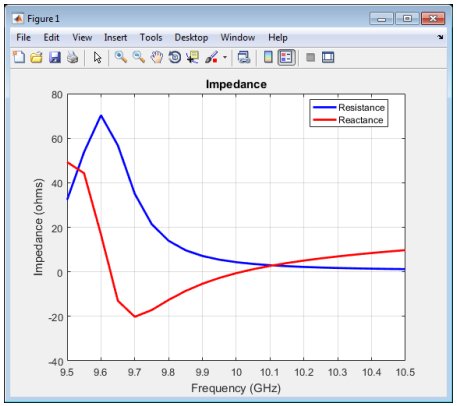
Comparison



Antenna Array, Impedance, and Electrical Coupling

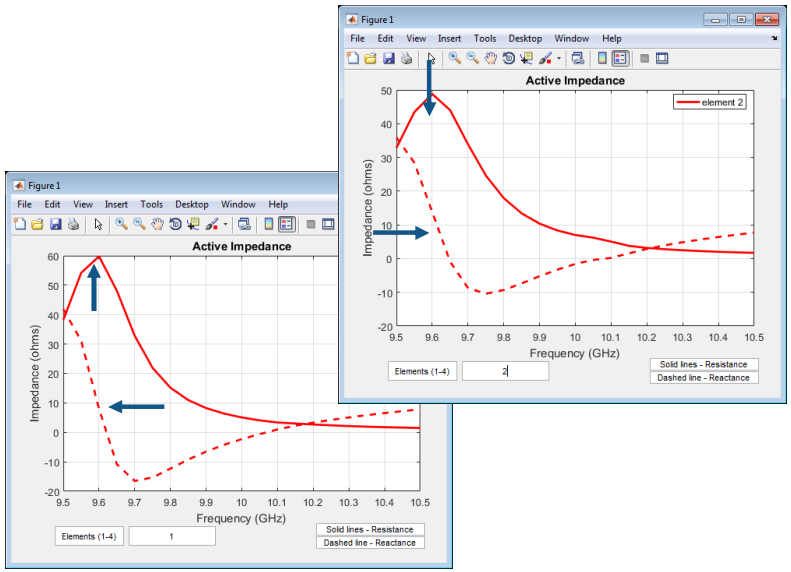
- Adjacent structures affect the impedance of an antenna embedded with an array
  - Resonant frequency
  - Electrical coupling in between antenna elements

Isolated element



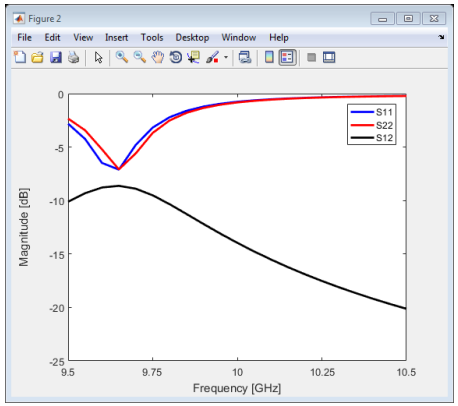

```
impedance(p, freq);
```

Active element




```
impedance(l, freq);
```

Full array




```
S=sparameters(l, freq);
```

Thanks for your attention

Questions?