4-29 Array Design and Analysis using GRASP MoM (TICRA) Wire Elements

We can use the voltage generator excitation in GRASP MoM add-on program to analyze phased arrays with wire type elements. A later section considers waveguide/coax port excitation. This method accounts for the full coupling between elements and automatically computes the gain efficiency and element input impedances. By feeding a single central element of an array and computing the UV-plane contour pattern scan blindness problems can be identified.

We prepare the array using voltage generator excitation for analysis in GRASP MoM by using the program TICMARW for arrays by combining geometry contained in mesh files (*.msh) and wire elements with a single feed point on each element. TICMARW generates a GRASP TOR file edit that includes scattering objects for each array element whose placement is specified by a separate coordinate system. Element locations are determined by the array position file (*.isp) with feed excitations specified by one or more *.exi files. The method shown below starts with an existing GRASP project and modifies the TOR file using a text editor. The existing geometry and excitations are replaced by the text file output of the program TICMARW. TICMARW generates TOR file scatter cluster of the elements of array each being a scatter cluster combining the mesh file and wire elements.

Arrays of Dipoles over Ground Plane

Single Element Design

We start with a GRASP input files for a single dipole located over a ground plane. The program GMDIPOLE writes the mesh file of the ground plane and the wire elements of a tilted dipole in GRASP format. The program is either run with keyboard data entry or from an input file. Inputs of a single dipole with a square ground plane is given below. Here is an example input text file (with explanations not in file):

```
4               cm
gmdipol1.txt    output file with TOR file elements
1               ground plane every element
dipgrd1.msh     TOR mesh file of ground plane
14.25           dipole length
.125            dipole radius
8               height over gr.pl.
20              tilt towared gr.pl.
1               single dipole
15.,15.         ground plane height, top width
15.             center width of gr.pl.
No              circular cup
dipole_coor     object file of element coordinate system incremented for each element
dipl_grd        tabulated mesh object name incremented
xdipole         Straight wire element incremented
```

GMDIPOLE writes a portion of the TOR geometry of the ground plane and dipole of a single element used in the program TICMARW to generate the TOR file edit for an array.

We need to edit the output of GMDIPOLE: gmdipol1.txt to add the object line count for each array element required as input to TICMARW to generate the full array. The listing below shows added line counts including blank lines (shown in red).

```
6
dipl_grd  tabulated_mesh
(
 coor_sys       : ref(dipole_coor)
 file_name      : dipgrd1.msh
)

14
xdipole  piecewise_straight_wire
(
 coor_sys       : ref(dipole_coor),
 nodes          : sequence
  (
   struct(x: -6.695310E+00 cm, y:  0.000000E+00 cm, z:  5.563107E+00 cm),
   struct(x: -7.125000E-01 cm, y:  0.000000E+00 cm, z:  8.000000E+00 cm),
   struct(x:  0.000000E+00 cm, y:  0.000000E+00 cm, z:  8.000000E+00 cm),
   struct(x:  7.125000E-01 cm, y:  0.000000E+00 cm, z:  8.000000E+00 cm),
   struct(x:  6.695310E+00 cm, y:  0.000000E+00 cm, z:  5.563107E+00 cm),
  ),
 radius         :  1.250000E-01 cm
)
```

The input file to TICMARW for the single dipole element is as follows with explanations in red.

Ticmarwd1.txt	TOR file edit output to modify an existing GRASP MoM project
Dipol1.isp	XADEF generates a single array element and TICARR writes array element file: *.ISP
1	# array elements
dipole_elem	Name of array elements in TOR file (incremented)
2	# of scatterers in each element
dipl_grd	tabulated mesh object name incremented
xdipole	Straight wire element incremented (dipole)
gmdipol1.txt	Modified output of GMDIPOLE
1	add voltage generators
Dipol1.exi	Element input voltages
0.,0.,8.	feed position on single element in gmdipole1.txt
Generator	Name of voltage array
array_spar	Impedance parameter

0 no other voltage excitations

The TOR file of an existing project is edited using the output file: TICMARWD1.TXT

```
dipole_elem  scatterer_cluster
(
  scatterers      : sequence(ref(elem1))
)

elem1  scatterer_cluster
(
  scatterers      : sequence(ref(dipl_grd1),ref(xdipole1))
)

array_coor_sys  coor_sys
(
)

coor_sys_elem1  coor_sys_grasp_angles
(
  origin          : struct(x: 0.0 cm, y: 0.0 cm, z: 0.0 cm),
  base            : ref(array_coor_sys)
)

dipl_grd1  tabulated_mesh
(
  coor_sys        : ref(coor_sys_elem1),
  file_name       : dipgrd2.msh
)

xdipole1  piecewise_straight_wire
(
  coor_sys        : ref(coor_sys_elem1),
  nodes           : sequence
    (    struct(x: -6.69531 cm, y: 0.0 cm, z: 5.563107 cm),
    struct(x: -0.7125 cm, y: 0.0 cm, z: 8.0 cm),
    struct(x: 0.0 cm, y: 0.0 cm, z: 8.0 cm),
    struct(x: 0.7125 cm, y: 0.0 cm, z: 8.0 cm),
    struct(x: 6.69531 cm, y: 0.0 cm, z: 5.563107 cm)
    ),
  radius          : 0.125 cm
)
```

```
generator  voltage_generator
(
  generators       : sequence
   (    struct(x: 0.0 cm, y: 0.0 cm, z: 8.0 cm, amplitude: 1.0 V, phase: 0.0)
   ),
  parameter_selection : s_parameters,
  parameter_file   : array_spar,
  coor_sys         : ref(array_coor_sys)
)
```

The rest of the TOR file contains essential elements to the GRASP project and must retained when adding the output to TICMARW.

```
wavelength  frequency_range
(
  frequency_range  : struct(start_frequency: 1.0 GHz, end_frequency: 1.0 GHz, number_of_frequencies:
1)
)

mom  mom
(
  frequency        : ref(wavelength),
  scatterer        : ref(dipole_elem),
  relative_geom_tolerance : 0.1E-03,
  iterative_solution : struct(use_mlfmm: automatic, relative_error: 0.1E-02, mlfmm_use_disk: allow,
mlfmm_precision: normal, preconditioner_accuracy: normal, obsolete_group_size: 4.0),
  keep_matrix      : on,
  polynomial_precision : 1,
  integration_precision : 2,
  advanced_polynomial : struct(edge: 1, wedge: 0, junction: 0, pec: 0, dielectric: 0, wire: 0),
  file_name        : array.cur,
  colour_plot_file : array.cpf
)

array_pattern  spherical_cut
(
  coor_sys         : ref(array_coor_sys),
  theta_range      : struct(start: -180.0, end: 180.0, np: 361),
  phi_range        : struct(start: 0.0, end: 90.0, np: 3),
  polarisation     : linear,
  file_name        : array01.cut,
  frequency        : ref(wavelength)
```

```
)

complete  spherical_cut
(
  coor_sys        : ref(array_coor_sys),
  theta_range     : struct(start: 0.0, end: 358.0, np: 180),
  phi_range       : struct(start: 0.0, end: 180.0, np: 91),
  comment         : "Field data in cuts",
  frequency       : ref(wavelength)
)

//DO NOT MODIFY OBJECTS BELOW THIS LINE.
//THESE OBJECTS ARE CREATED AND MANAGED BY THE
//GRAPHICAL USER INTERFACE AND SHOULD NOT BE
//MODIFIED MANUALLY!
view_1  view
(
  objects         :
sequence(ref(view_1_mom_plot),ref(view_1_mom_source_plot),ref(view_1_output_points_plot),
ref(view_1_coor_sys_plot),ref(view_1_tabulated_mesh_plot),ref(view_1_wires_plot),ref(view_1_box_pl
ot))
)
…
  //$$ Saved at 12:00:09 on 27.12.2017 by GRASP ver. 10.6.0 SN=005400
```

For comparison a tilted dipole was placed over a 3λ x 3λ square ground plane and analyzed using GRASP MoM and PO method developed by Diaz and Milligan, *Antenna Engineering using Physical Optics*, Artech 1996 (Section 5-5a). Figure 1 gives the GRASP MoM pattern output in the *E*-, *H*-, and diagonal-planes. Figure 2 shows the pattern obtained by using the Diaz and Milligan PO analysis including the ground plane in the *E*- and *H*-planes. We see that they match.

The PO analysis uses a sinusoidal current distribution on the dipole to compute currents on the 3λ x 3λ square ground plane. The reactance theorem is used to compute the self-impedance of the dipole and the mutual impedance between the dipole and the finite ground plane currents to compute the dipole input impedance. The currents on the dipole and ground plane are scaled for a 1W input and used to compute far-field patterns.

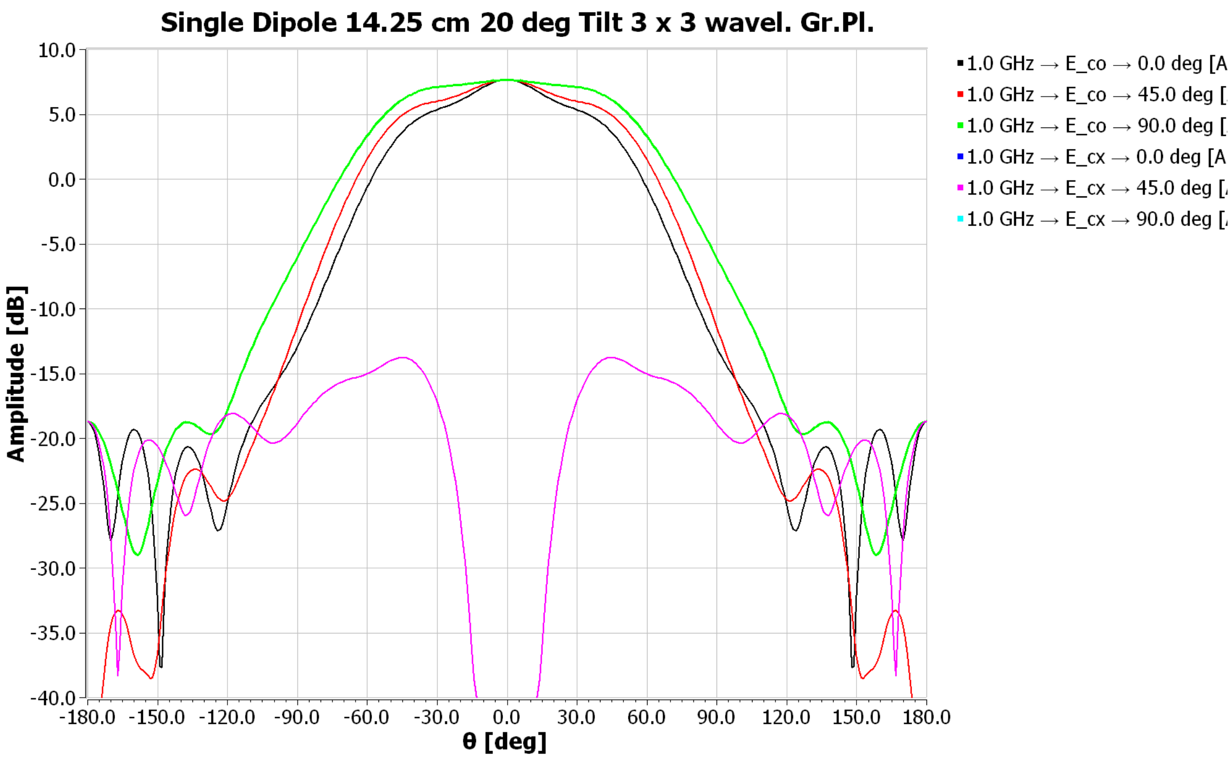


Figure 1 GRASP MoM analysis over 3- x 3-λ ground plane

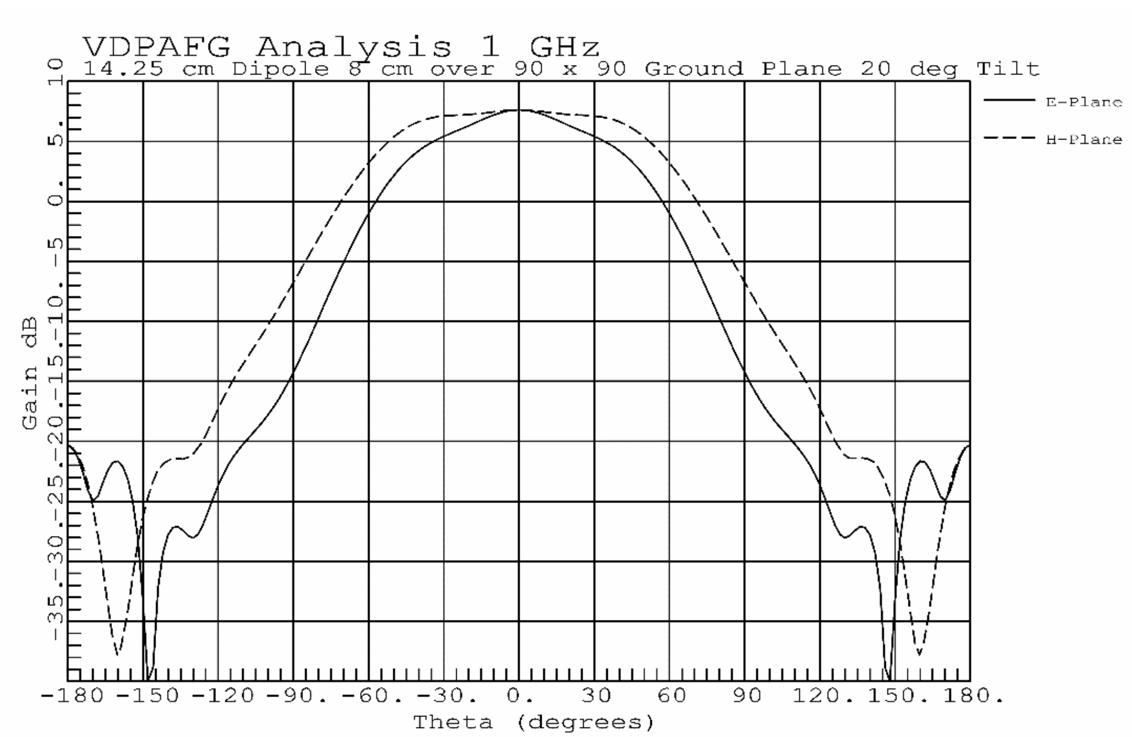


Figure 2 PO Analysis (Diaz and Milligan) analysis over 3- x 3-λ ground plane

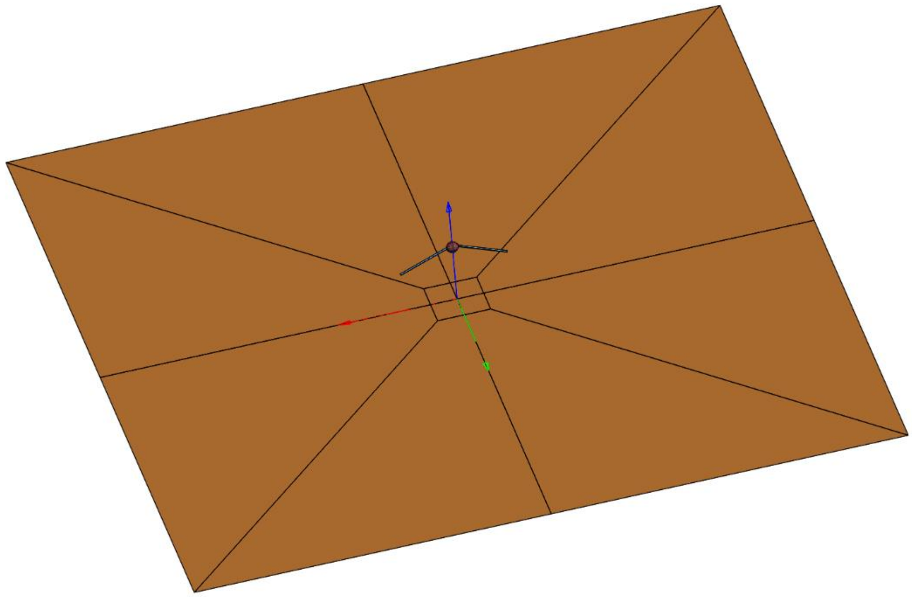


Figure 3 GRASP MoM Model automatically re-meshes the large patches

Both analyses predict a boresight gain of 7.63 dB and they match the computation of directivity using the pattern over the full radiation sphere. This illustrates my contention that PO can be related to MoM provided the interaction is accounted for between currents or we use an iterative PO analysis. If the coupling between various currents in a PO (or MoM) analysis is small, then the step-by-step propagation between objects in an analysis produces correct results as done in GRASP PO.

The gain of the same dipole over an infinite ground plane using image analysis is 7.29 dB. The high gain of the dipole element when spaced λ/2 in an array the mutual coupling reduces the gain of the active element because the pattern gain cannot exceed π (4.97 dB) in a large array (the associated area of each element).

Dipole Array

We can write the TOR file edit using TICMARW using the same single dipole by using a GRASP array geometry input file (*.ISP) and excitation files (*.EXI) to generate array and scanned (active) array element patterns when analyzed in GRASP MoM using voltage generator excitation. Below are the inputs to TICMARW for a 25-element array.

ticmarwd1.txt TOR file edit output of TICMARW
dipole25.isp GRASP array geometry file (positions) 25 elements 15 cm spacings
25 # array elements
dipole_elem
2 # of scatterers in each element
dipl_grd Ground plane mesh file prefix *.MSH

xdipole Dipole straight element prefix
gmdipol1.txt Modified output of GMDIPOLE with added # elements
1 add voltage generators
Dipole25.exi GRASP uniform amplitude and phase excitations
0.,0.,8. feed position on element
Generator Object voltage-generator name
array_spar Output impedance parameters
1 add voltage generators
Dipole2513.exi GRASP excitation of middle element (13) only
0.,0.,8. feed position on element
generator_13 Object voltage-generator name
array_spar13 Output impedance parameters
1 add voltage generators
Dipole251.exi GRASP excitation of corner element (1) only
0.,0.,8. feed position on element
generator_1 Object voltage-generator name
array_spar1 Output impedance parameters
0 no other voltages

Dipole elements are arrayed with a 15 cm spacing, λ/2 at 1 GHz in a 5 x 5 array with 15- x 15-cm square ground planes under each dipole. Figure 4 shows the GUI illustration of array input of the 25-element array with joined individual ground planes. Figure 5 plots the array pattern when excited uniformly.

Most dipole arrays have extended ground planes so that the edge elements do not spill their patterns over the edge and have greatly altered element patterns compared to central elements. A separate GRASP MoM mesh file can be generated to surround the array elements by using the program GMEXGRD which produces a single mesh file. For the case shown below XADEF generates the geometry of 9 x 9 element array using 15 cm spacings. The program XADSUB subtracts the common elements between the 81- and 25-element arrays and generates the difference array of 56 elements. TICARR converts the XADEF array geometry file to a GRASP array geometry file. GMEXGRD can be run from an input file and produce the GRASP MoM mesh file.

exgrd2.msh
diplc56.isp
56 number of empty array elements
15.,15. Y-axis, X-axis Top patch dimensions
15. X-axis center width
0 end of isp files (multiple *.ISP files can be concatenated into single mesh file)

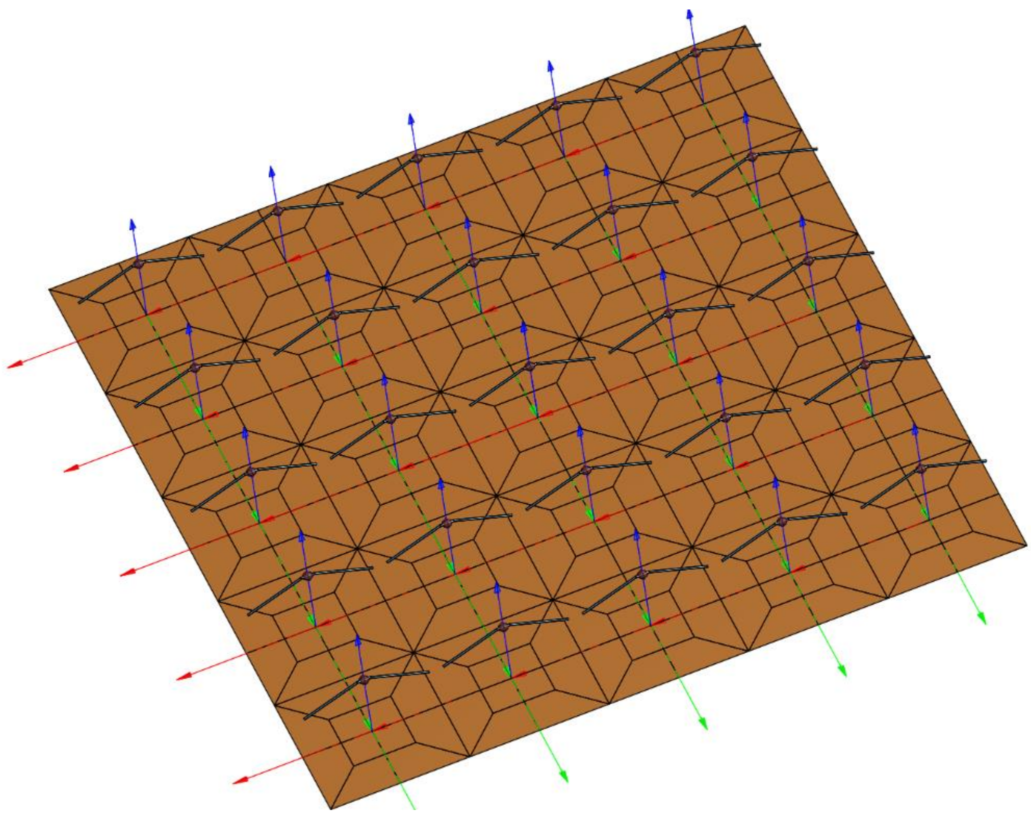


Figure 4 25 element tilted dipole spaced λ/2 array over finite ground plane

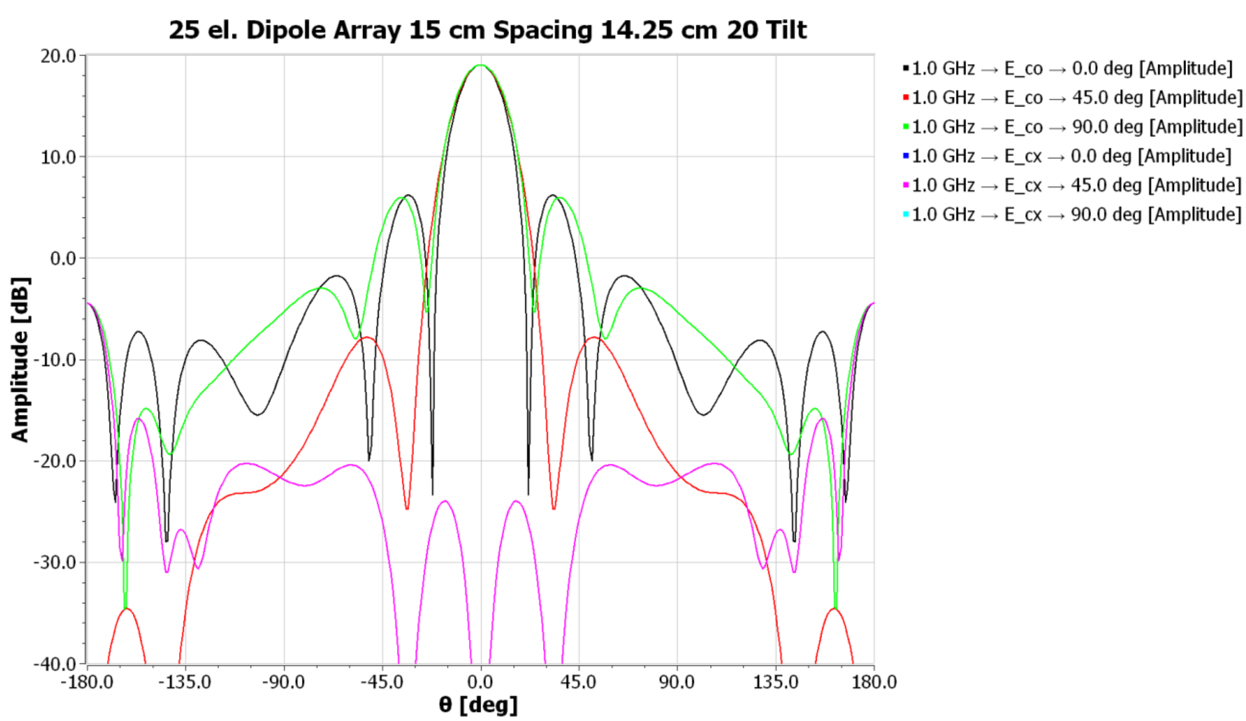


Figure 5 25 element tilted dipole spaced λ/2 array over limited ground plane GRASP pattern

The “create scatterer” command in GRASP adds the *.msh geometry to produce the geometry in figure 6. The meshing of the patches around the array elements match those produced by GMDIPOLE. It is necessary to add “tabulated_msh_grd” to scatterer cluster “dipole_elem” in the GUI.

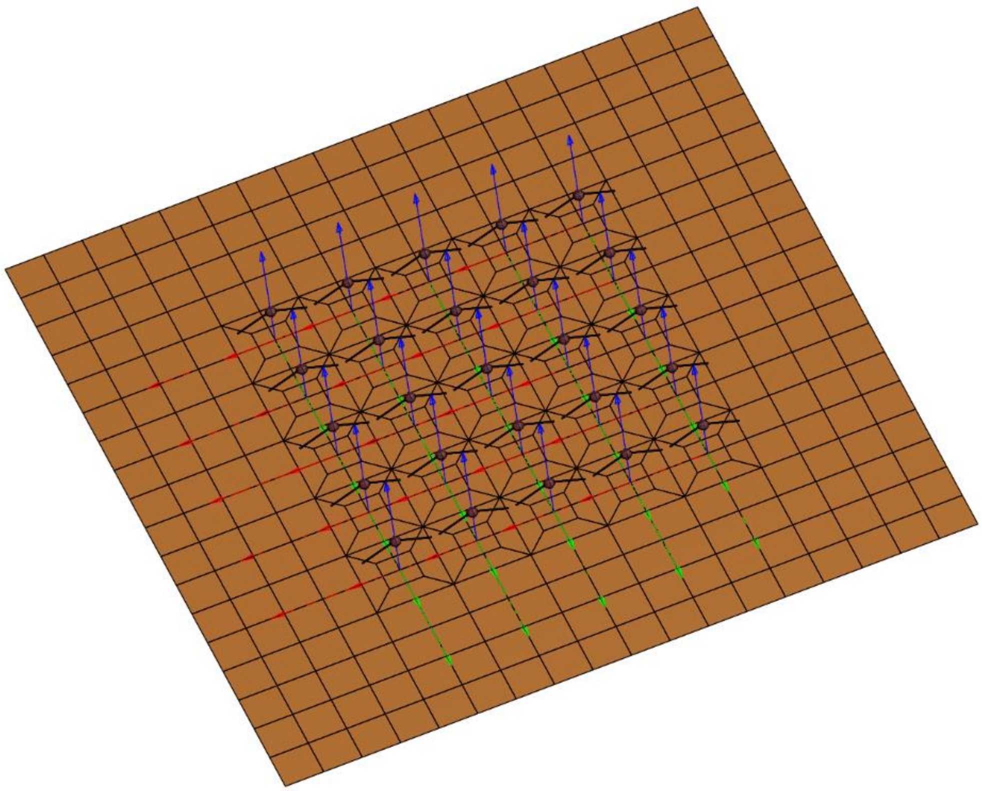


Figure 6 25 element tilted dipole spaced λ/2 array over extended ground plane

Section 4-27 discusses equally spaced uniform amplitude thinned arrays to produce a given aperture distribution (see Figure 28). It is necessary to fill in ground plane between elements because TICMARW will only place ground under elements which exist. By subtracting the thinned array XADEF file from an un-thinned (filled) array, the missing elements are separated into a separate XADEF file. GMEXGRD will generate the combined mesh file of these ground planes and remove the holes in a GRASP MoM analysis.

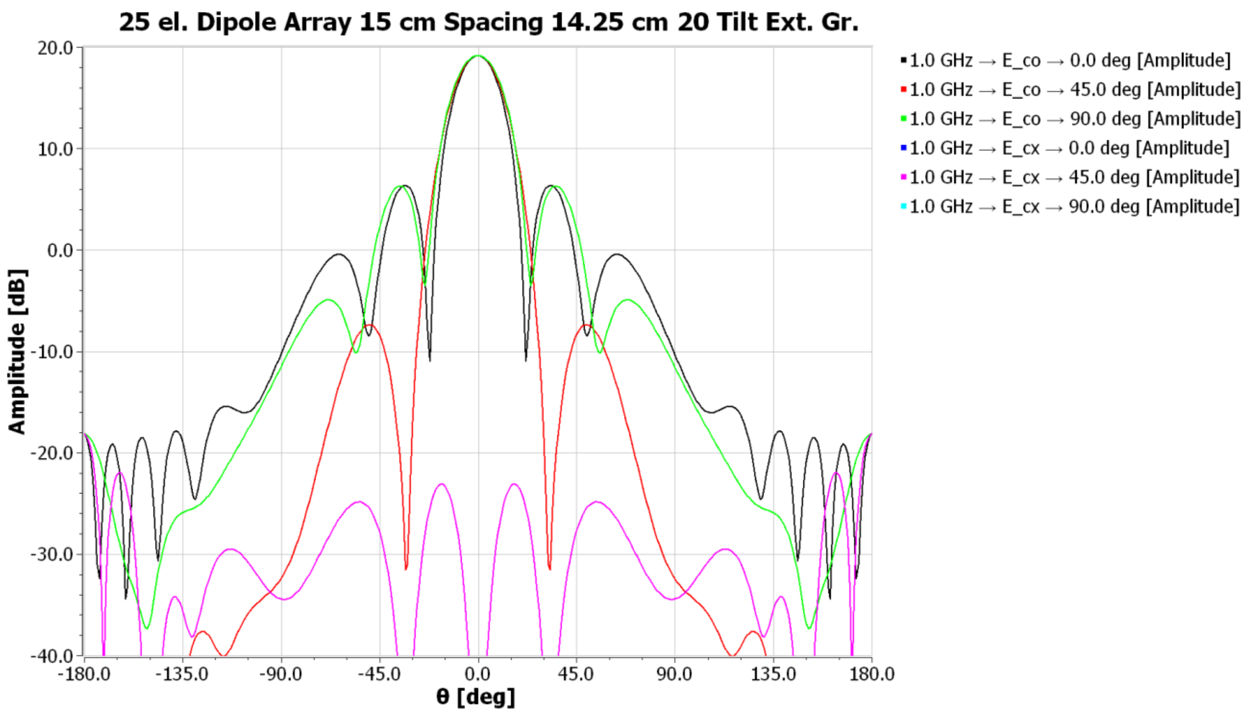


Figure 7 25 element tilted dipole spaced λ/2 array over extended ground plane GRASP pattern

Figure 5 shows a peak gain of 19.02 dB. We compare this to 25 elements (13.97 dB) plus the gain of this dipole over an infinite ground plane of 7.29 dB, a sum of 21.27 dB. The closely spaced elements cause a gain reduction of 2.25 dB which matches Table 2, Section 4-28. The extended ground plane reduces the backlobe by 14 dB, Figure 7. Figure 8 shows the active or scan element pattern of the central element and how its pattern dips at boresight due to interelement coupling. When we add the extended ground plane the central element active- or scan-element pattern in Figure 9 shows a wider pattern with more ripples. The UV-space patterns of figures 10 and 11 illustrate a wide dip that hints of a scan blindness problem as the array grows in size.

Figures 12 and 13 demonstrate that the pattern of the corner element and that every element in the array has a different response due to the combination of different coupling discussed in Section 4-28 and their position relative to the finite ground plane. The UV-contour of the corner element, figure 14 and 15, show less pattern variation across the broad pattern of the dipole mounted over a ground plane.

We create field storage objects for the spherical cut and grid files of the individually fed elements. To feed a single element either the center element (13) or corner element we set the amplitude to 100 dB in XADEF to overshadow all other elements to be used in TICARR for the excitation for the active- or scan-element pattern. We add commands illustrated below for the excitation voltage generators of the single element to first compute currents and followed by pattern "get fields" commands. We can use the "keep matrix" setting in the mom object because only one frequency is computed and we use the inverted matrix over and over with new excitations: array, element 13, and element 1.

	Command Type	Objects
1	Get Currents	Source : generator Target : mom
2	Get Field	Source : mom Target : array_pattern
3	Get Field	Source : mom Target : complete
4	Get Field	Source : mom Target : spherical_grid
5	Get Currents	Source : generator_1 Target : mom
6	Get Field	Source : mom Target : spherical_grid_01
7	Get Field	Source : mom Target : spherical_cut_01
8	Get Currents	Source : generator_13 Target : mom
9	Get Field	Source : mom Target : spherical_grid_13
10	Get Field	Source : mom Target : spherical_cut_13

Command file tasks to generate array and element patterns and UV-contours after setting the mom object to “keep matrix” to reduce run time using a single frequency.

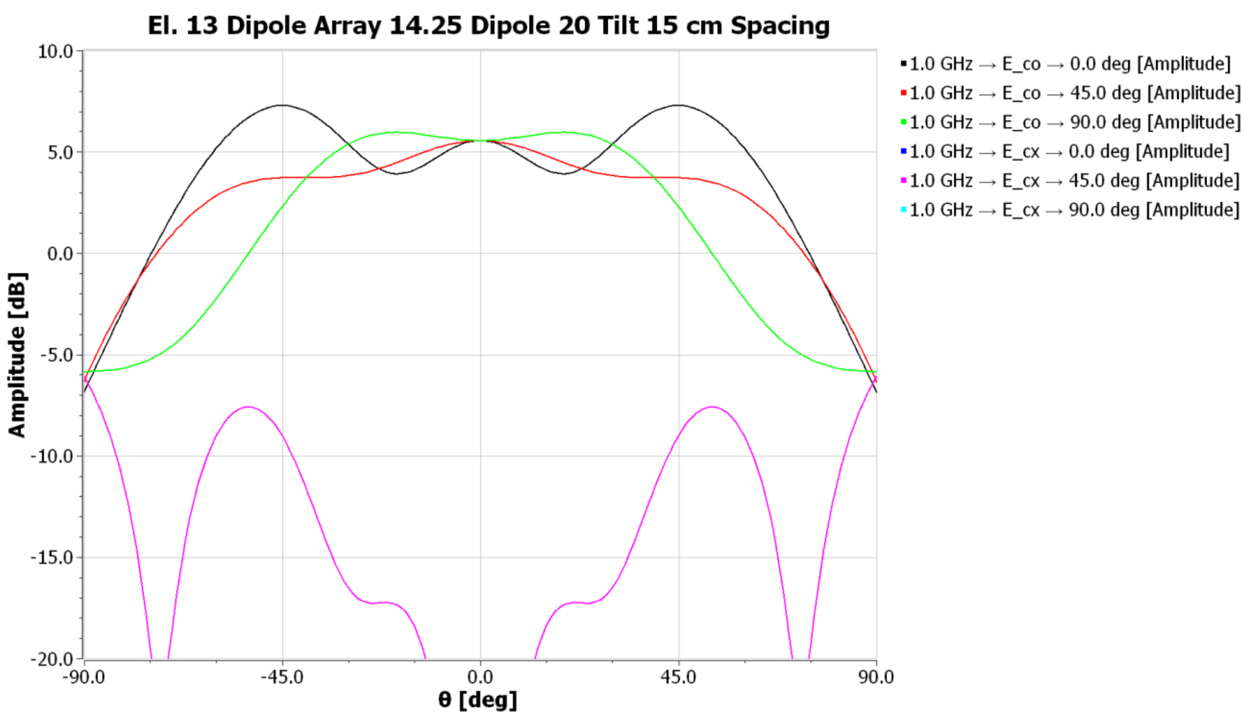


Figure 8 Active or scan element pattern of central element of 25 element dipole square array spaced λ/2 without extended ground plane

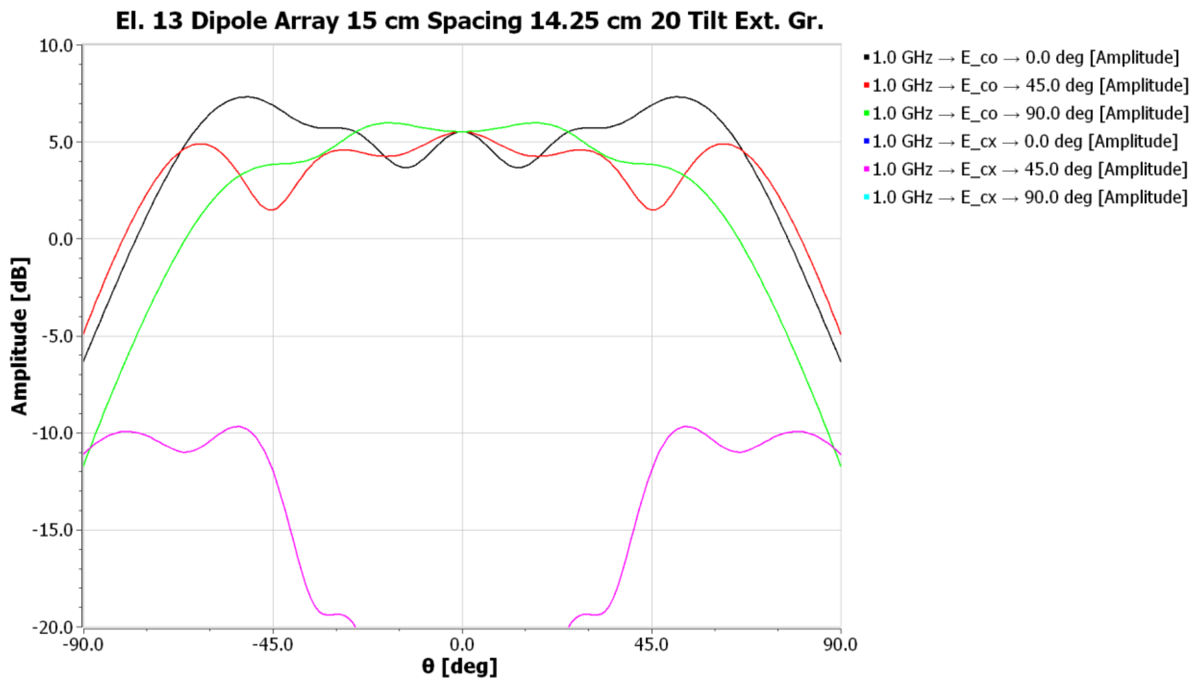


Figure 9 Active or scan element pattern of central element of 25 element dipole square array spaced λ/2 with extended ground plane

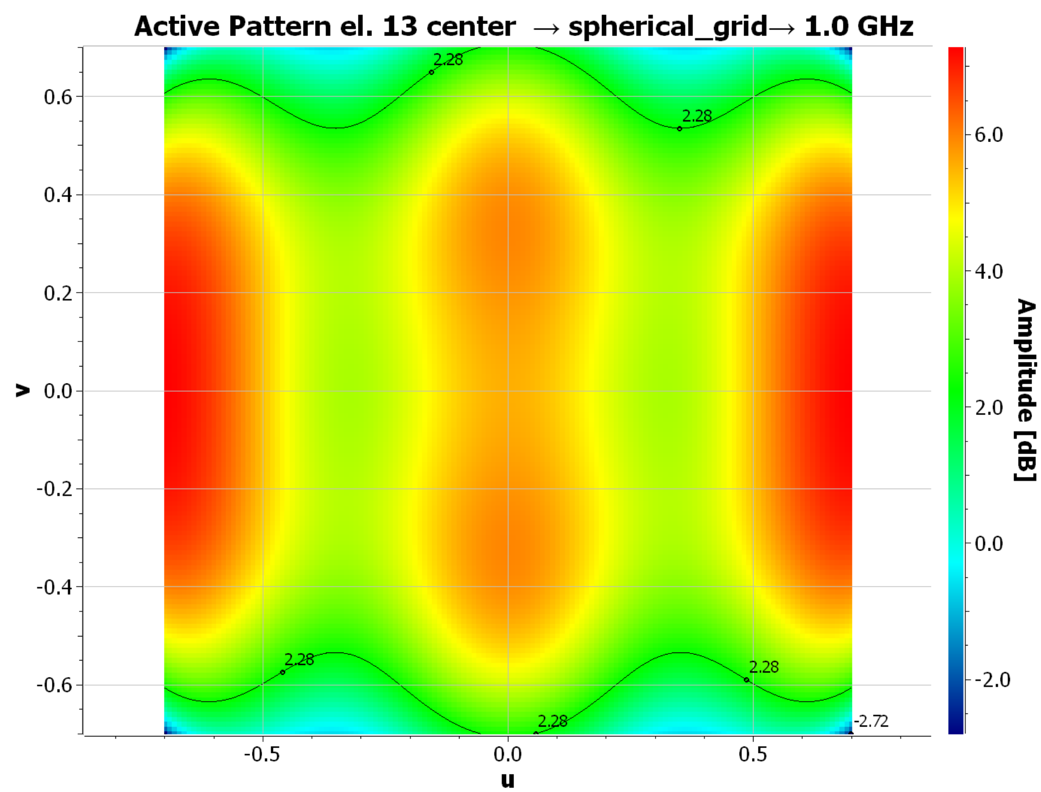


Figure 10 Active or scan-element pattern of central element of 25 element square λ/2 spaced array

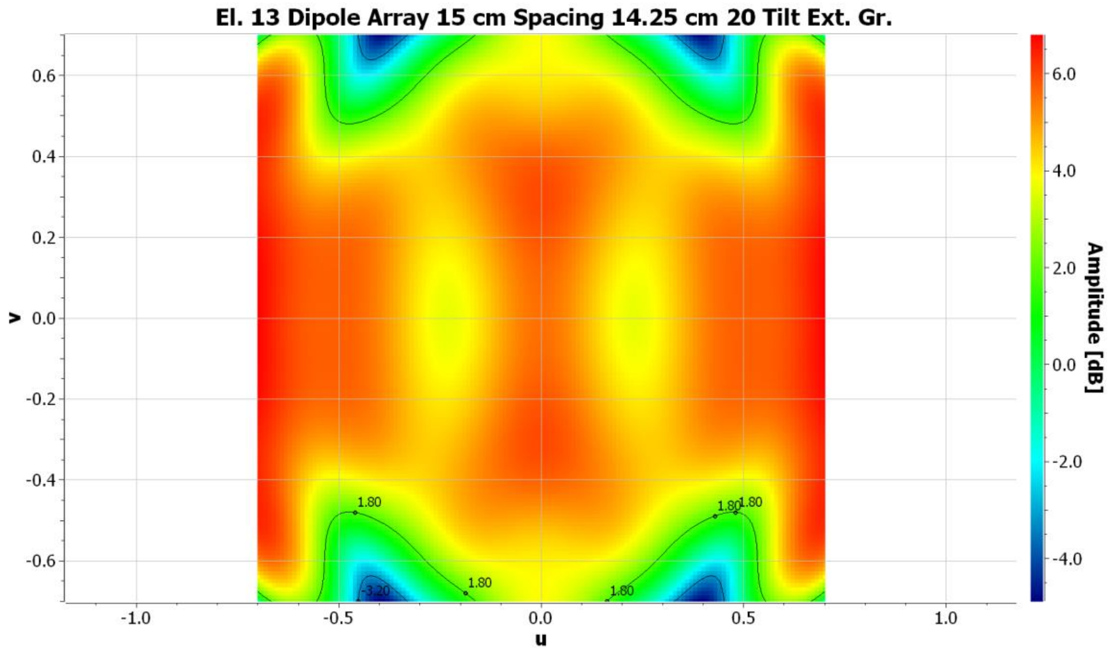


Figure 11 Active or scan-element pattern of central element of 25 element square λ/2 spaced array with extended Ground Plane

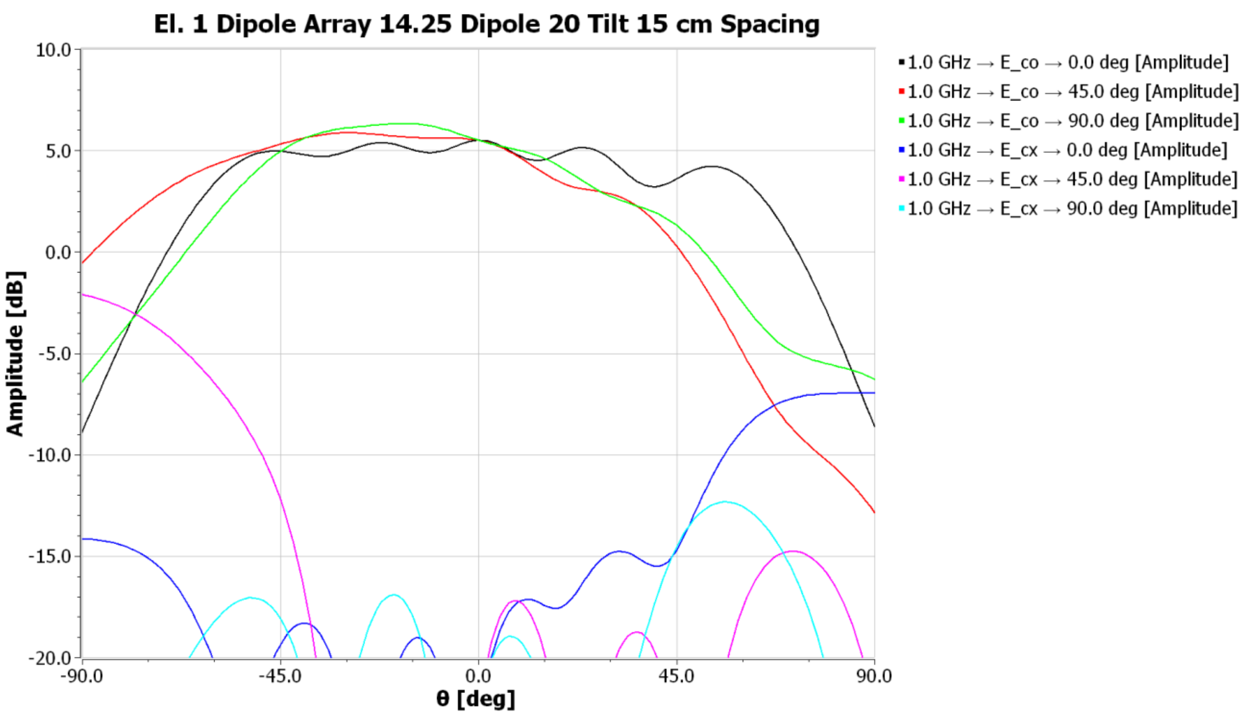


Figure 12 Active or scan element pattern of corner element of 25 element dipole square array spaced λ/2.

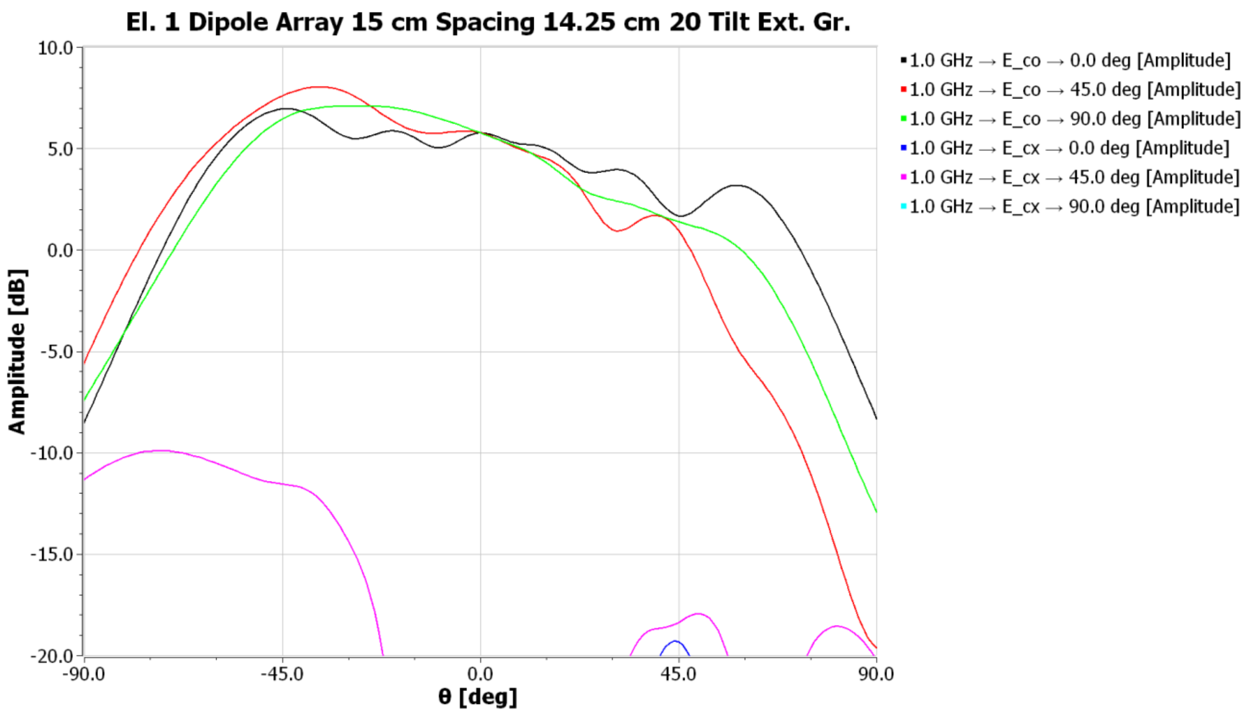


Figure 13 Active or scan element pattern of corner element of 25 element dipole square array spaced λ/2 with extended Ground Plane

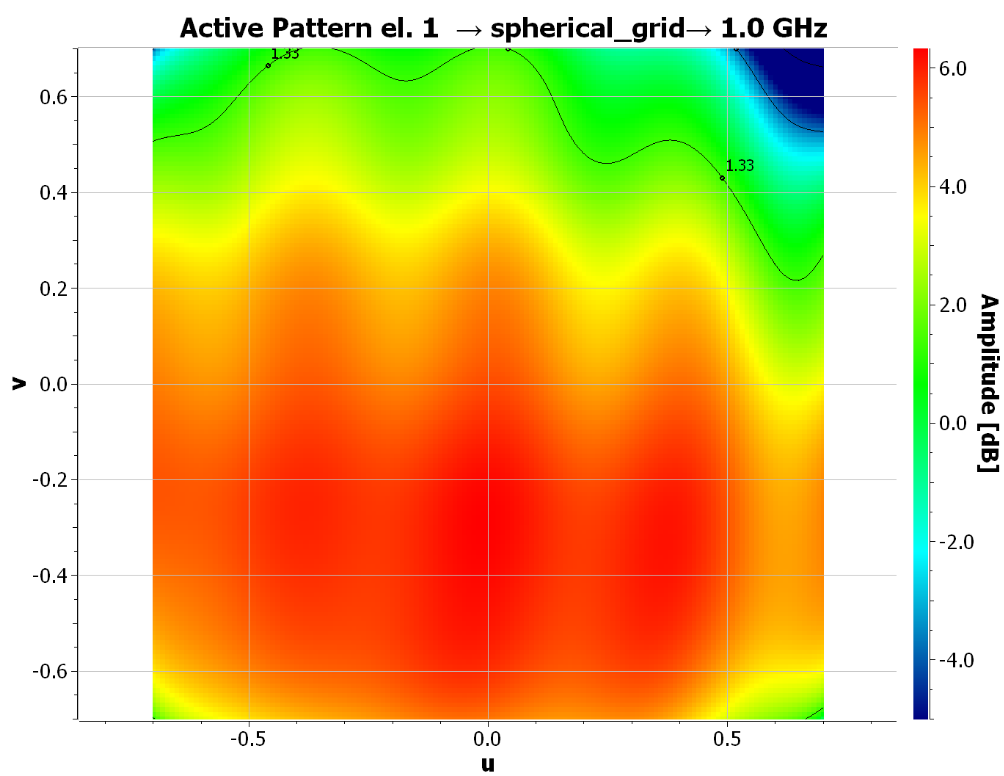


Figure 14 Active or scan-element pattern of corner element of 25 element square λ/2 spaced array

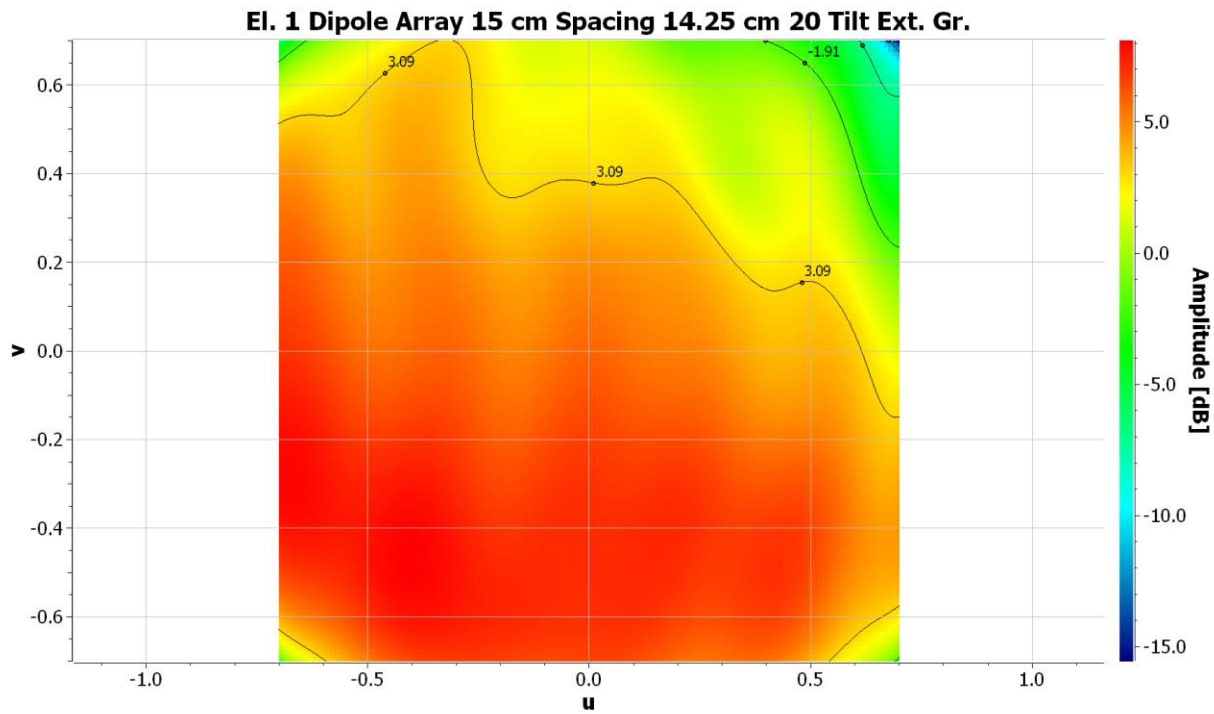


Figure 15 Active or scan-element pattern of corner element of 25 element square λ/2 spaced array with extended Ground Plane

Table 2 predicts that the gain reduction due to interelement coupling can be reduced by spacing the elements 0.65λ. By repeating the steps given above, GRASP MoM predicts the pattern given by Figure 16. The pattern peak increases to 21.08 dB. The gain reduction due to overlapping effective area of elements becomes 0.2 dB which is slightly different from Table 2 of the 121-element array in Section 4-28. The active- or scan-element pattern of the central element has a reduced dip, figure 18, and a wide uniform UV-contour beam, figure 19. Matching the allotted area to the array element gain produces greatly improved patterns.

Figures 20 and 21 of the pattern and UV-contour of the corner element show that this element still has pattern variation due to unbalanced coupling to neighboring elements and the closeness of the ground plane edge. They are improved, but not to the extent of the central element.

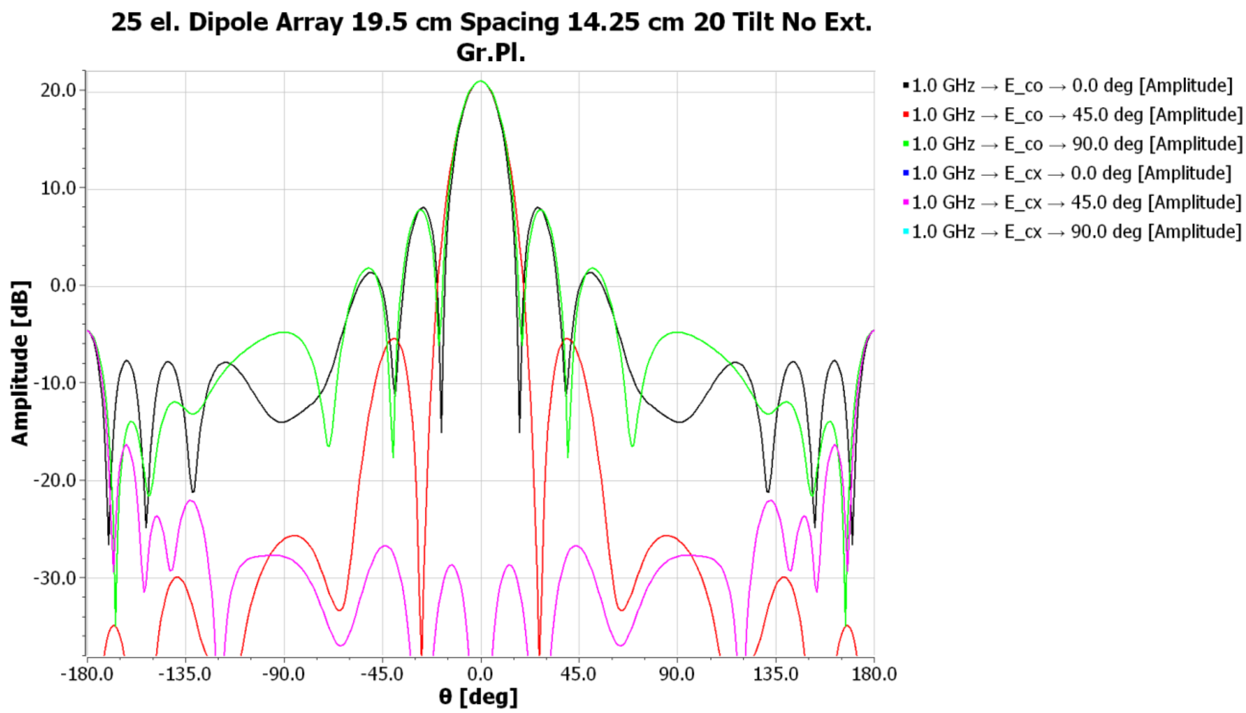


Figure 16 25 element tilted dipole spaced 0.65λ array over limited ground plane GRASP pattern

Figure 17 25 element tilted dipole spaced 0.65λ array over extended ground plane GRASP pattern

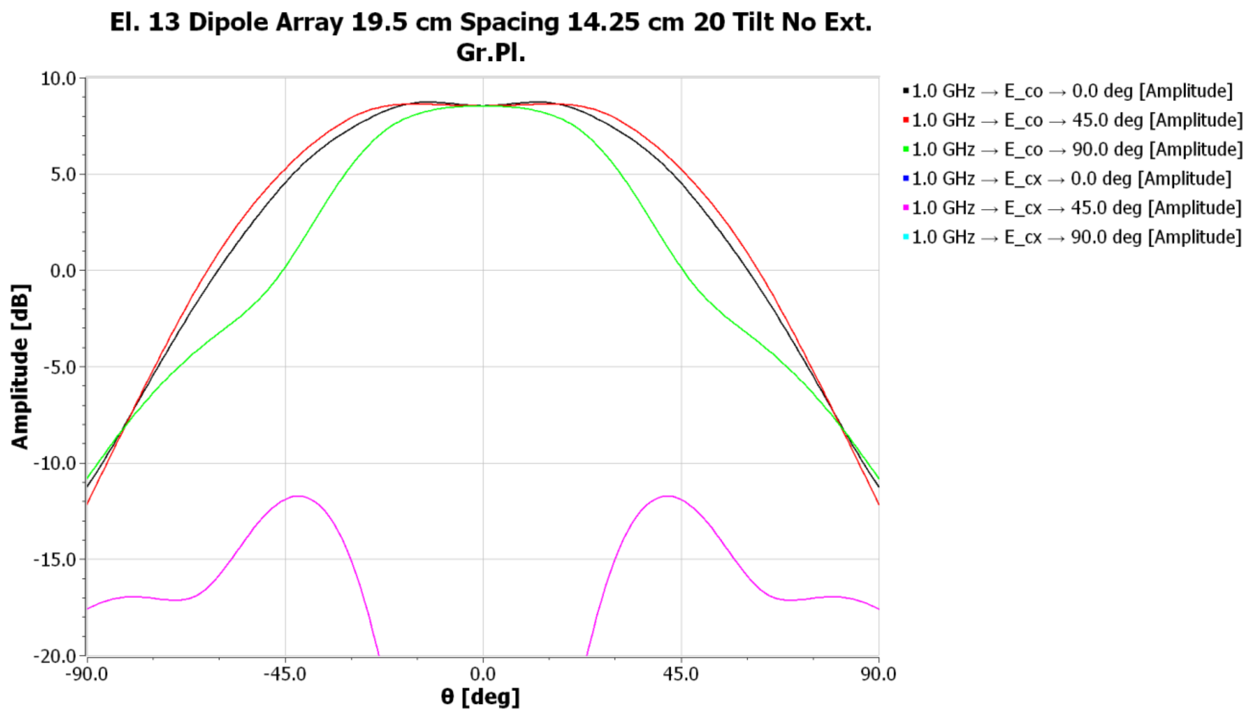


Figure 18 Active or scan element pattern of central element of 25 element dipole square array spaced 0.65λ

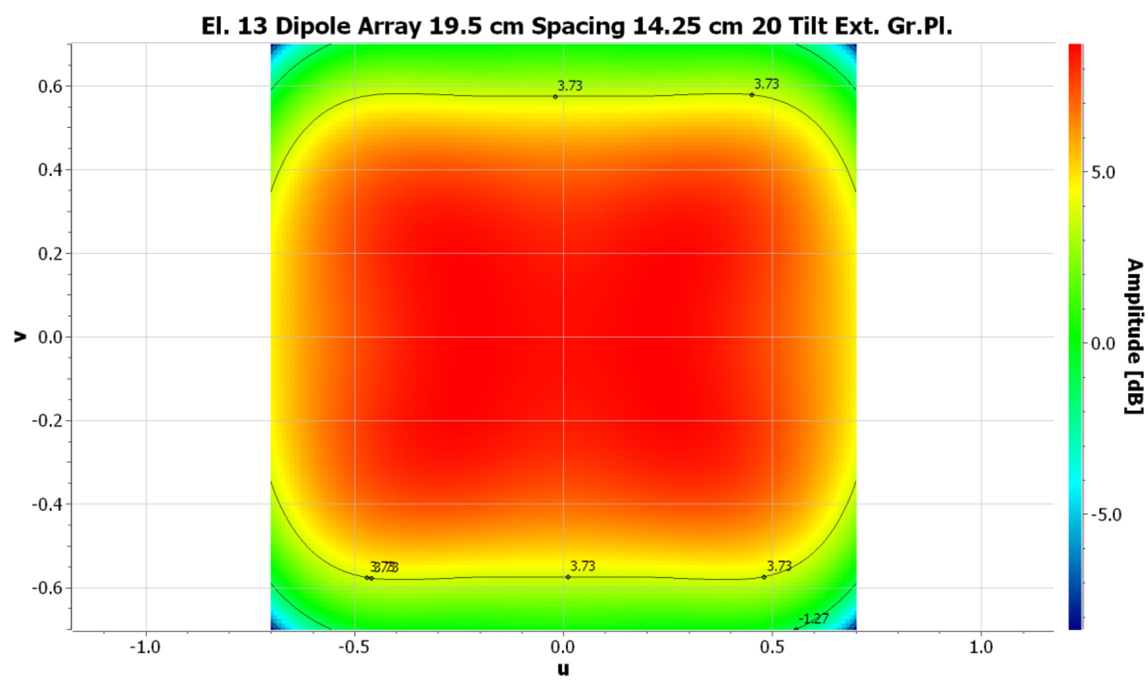


Figure 19 Active or scan-element pattern of central element of 25 element square 0.65λ spaced array

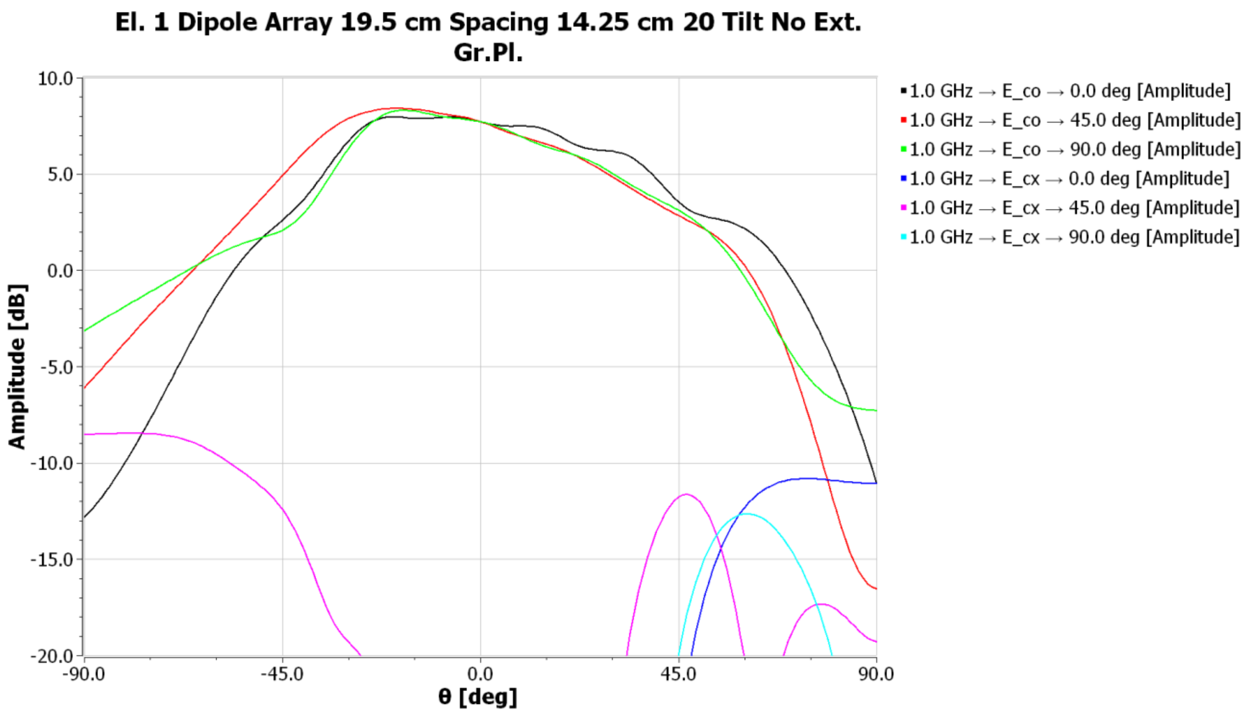


Figure 20 Active or scan element pattern of corner element of 25 element dipole square array spaced 0.65λ

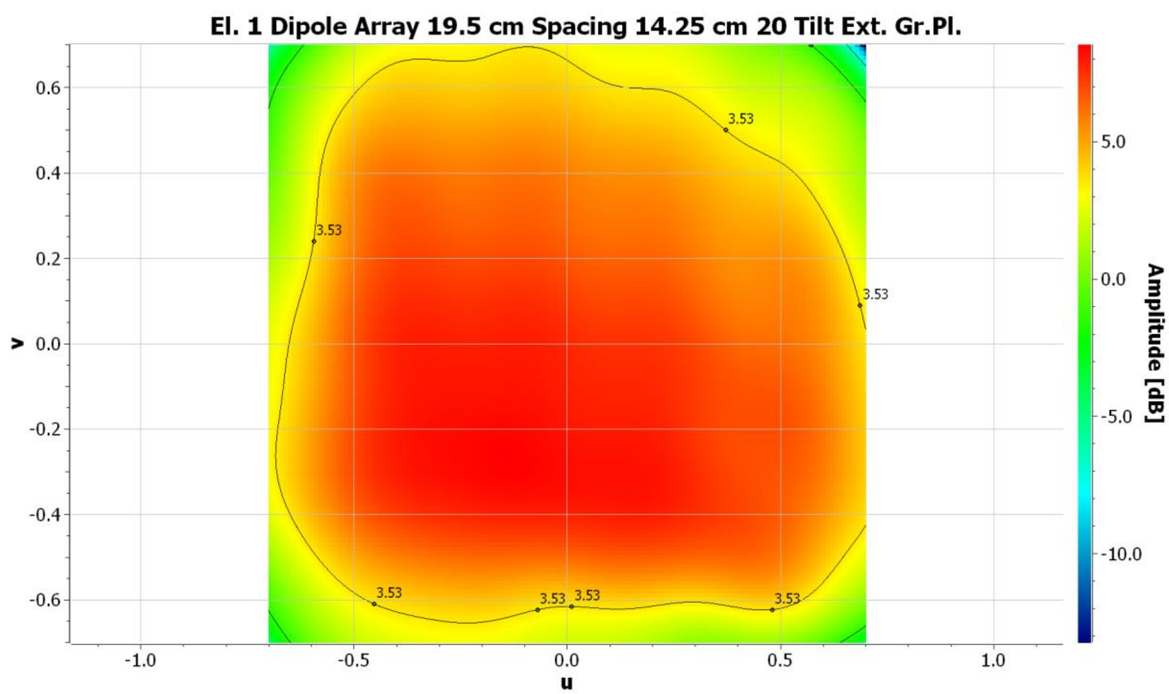


Figure 21 Active or scan-element pattern of corner element of 25 element square 0.65λ spaced array

Conformal Dipole Array on a Cylinder

The program XADEF is able to roll an array around a cylinder. The distance between elements is along the arc length of the cylinder. This causes a geometric problem because we use flat ground plane elements that produces ground gaps if we do not account for this effect. If done incorrectly, it can also cause overlap or intersection of the pieces of ground plane in a non-physical manner. Figure 22 shows half of two plates joined outside the rolling cylinder where the distance between elements is 2*arclength.

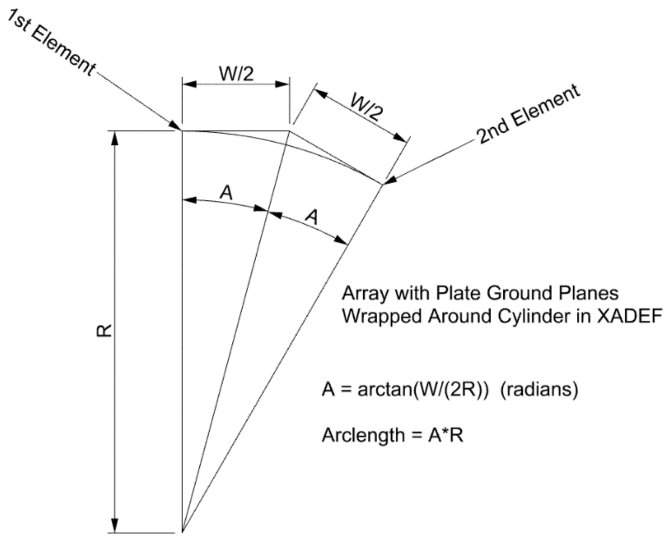


Figure 22 Geometry of construction of Flat Plates rolled in a Cylinder

Consider an array of dipoles over 19.5 cm square ground planes rolled on a cylinder of radius = 80 cm. The equation on Figure 22 computes the distance between elements = 2*80*arctan(19.5/160)) = 19.4043 cm in the roll direction in XADEF.

Command Prompt - xadef

```
C:\arrays>xadef
Enter input 0 keyboard, 1 file 0
Enter File Name cyl_dipole_25.arr
Enter units: 1 in., 2 ft, 3 cm, 4 m 3
New File? y
Enter Frequency (GHz) 1
xadef: ad,re,25
Enter Ampl (dB), Phase 0,0
Enter Number of Elements along X-axis 5
Enter initial spacings in X,Y axes cm  19.4043,19.5
Array along X-axis
Enter Axis Spacing: 1 Uniform, 2 Bratkovic  3 Geometric, 4 Uneven Taylor 1
Enter 1) Uniform Amplitude Distribution
      2) Area Sampling of Taylor Distribution
      3) Point Sampling of Taylor Distribution
      4) Zero Sampled Taylor Distribution
      5) Point Sampled Bayliss Distribution
      6) Zero Sampled Bayliss Distribution
      7) Chebyshev array 1
Enter Quadratic Phase Factor, S 0
Array along Y-axis
Enter Axis Spacing: 1 Uniform, 2 Bratkovic  3 Geometric, 4 Uneven Taylor 1
Enter 1) Uniform Amplitude Distribution
      2) Area Sampling of Taylor Distribution
      3) Point Sampling of Taylor Distribution
      4) Zero Sampled Taylor Distribution
      5) Point Sampled Bayliss Distribution
      6) Zero Sampled Bayliss Distribution
      7) Chebyshev array 1_
Enter Quadratic Phase Factor, S 0
Array along Y-axis
Enter Axis Spacing: 1 Uniform, 2 Bratkovic  3 Geometric, 4 Uneven Taylor 1
Enter 1) Uniform Amplitude Distribution
      2) Area Sampling of Taylor Distribution
      3) Point Sampling of Taylor Distribution
      4) Zero Sampled Taylor Distribution
      5) Point Sampled Bayliss Distribution
      6) Zero Sampled Bayliss Distribution
      7) Chebyshev array 1
Enter Quadratic Phase Factor, S 0
move: pr,co
Projection of Array from X-Y plane to Cone
Is the array orientated so that the X-axis will be around
the cone and Y axis along the cone axis? y
Enter Radius of Cone at Center of Array  cm  80
Enter Cone Angle (0 for cylinder) 0
Enter Rotation from X-axis of Center of array on Cone 0
Enter Orientation of Cone Axis: Theta, Phi 0,0
move: ro,ax
Array must orientated in X-Y plane before it is Rotated, ready? y
Enter Lower End Point (X,Y,Z) of Axis of Rotation 0,0,0
Enter Upper End Point (X,Y,Z) of Axis of Rotation 0,1,0
Enter Rotation Angle about Axis -90
Rotate element pointing? y
move: tr,ge
Enter Translation X,Y,Z  cm  0,0,-80
move: ex
Enter Final New Z axis Rotation of Antennas (array) 0
xadef:
```

We use TICARR to generate GRASP input files

```
Enter input 0 keyboard, 1 file 0
Enter XADEF array File Name cyl_dipole_25.arr
Number of Elements:     25
Frequency:             1.000GHz
Enter Ticra element position output filename (.isp) cyl_dipole_25.isp
Enter units: 1 in, 2 ft, 3 mm, 4 cm, 5 m 4
Enter label
19.5 cm square gr.pl. with tilted dipoles
Enter TICRA source object name for element pattern dipole_elem
Enter Ticra excitation output filename (.exi) cyl_dipole_25.exi
Enter quadratic phase taper (deg) X,Y, Max Radius X,Y 0,0,10,10
Enter array efficiency dB 0
Enter 1 to form beam of array 1
Enter scan direction of array Theta, Phi 0,0
Enter beam pointing frequency (GHz) 1
Enter phase center (X, Y, Z) cm
0,0,0
Quantize Phase? n
Enter Feed Error (1 Sigma) Ampl(dB), Phase(deg) 0,0
Another beam excitation file? n
Process another array file? n
Stop - Program terminated.
```

We perform the same steps as the 25 element flat ground plane array on the cylindrical array by substituting the cyl_dipole_25.isp array geometry file and cyl_dipole_25.exi excitation file into a edited TICMARW input file. The TICMARW output is used to edit the GRASP TOR geometry file. Figure 23 shows the GRASP geometry of the 25-element dipole array rolled on a cylinder.

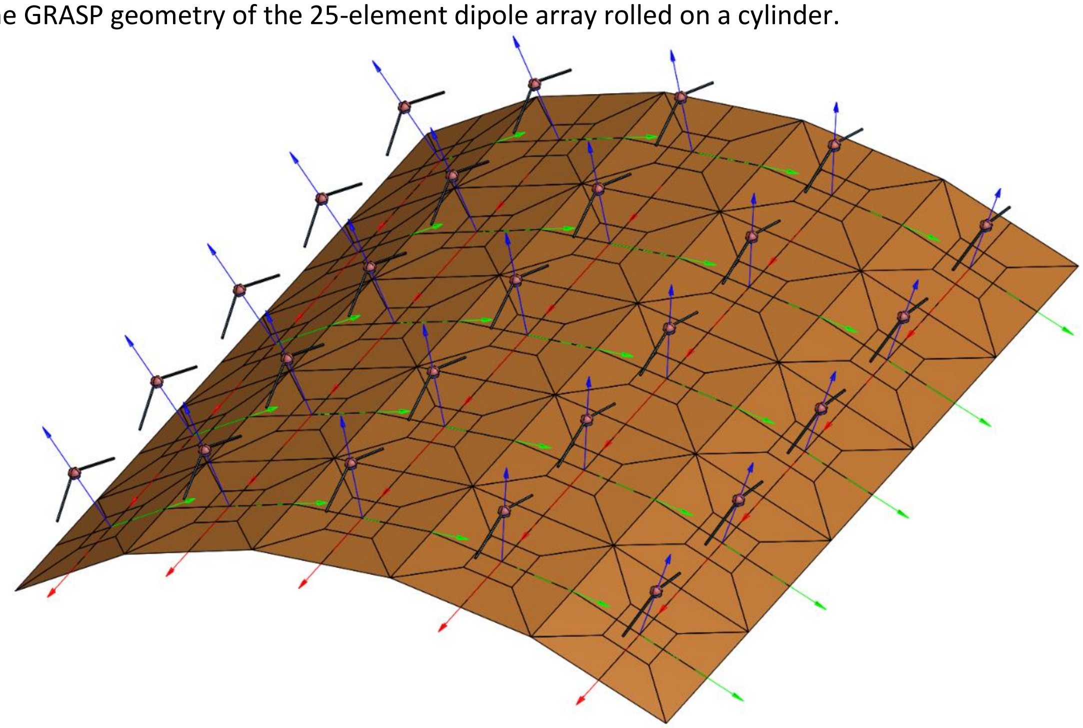


Figure 23 25-element tilted dipole spaced 0.65λ array over cylindrical ground plane R = 80 cm

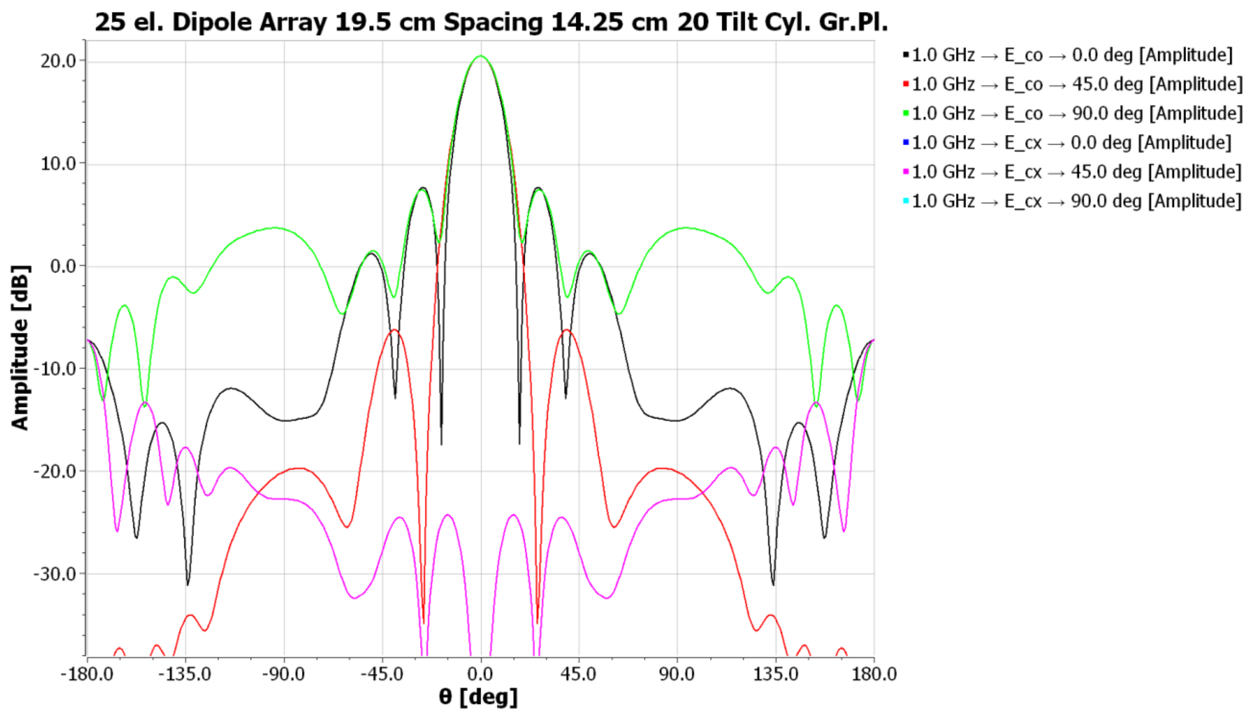


Figure 24 25-element Dipole Array 0.65λ spacings rolled on Cylinder

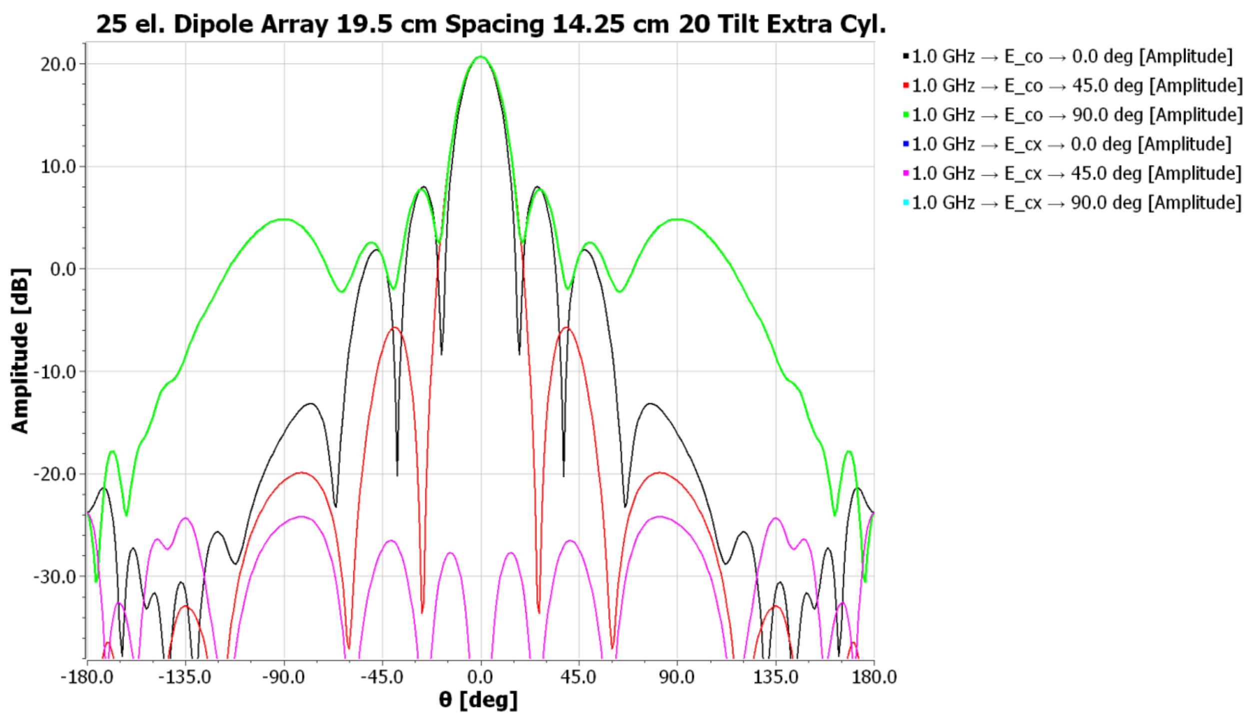


Figure 25 25-element Dipole Array 0.65λ spacings rolled on Extended Ground Plane Cylinder

We use XADEF to design another array with 81 elements by using the steps given above to roll it on a cylinder. XADSUB subtracts the two arrays to form the 56-element array to extended the ground plane.

The program GMEXGRD generates a mesh file with empty plates. The extended ground plane mesh is added to the GRASP project by the create command with the mesh file placed in working directory. The last step is to add the extended ground plane mesh file to the scatterer cluster: dipole_elem cluster so it is included in the MOM analysis. Notice that the dipoles are aligned with the x-axis (cylinder axis) even though XADEF rolls the x-axis of the array around the cylinder. This operation only effects the pointing of elements in the array and the coordinate system of each dipole is based only on the array coordinate system. To rotate the elements coordinates systems would be required to rotate elements relative to each “element coordinates systems” which TICMARW does not generate.

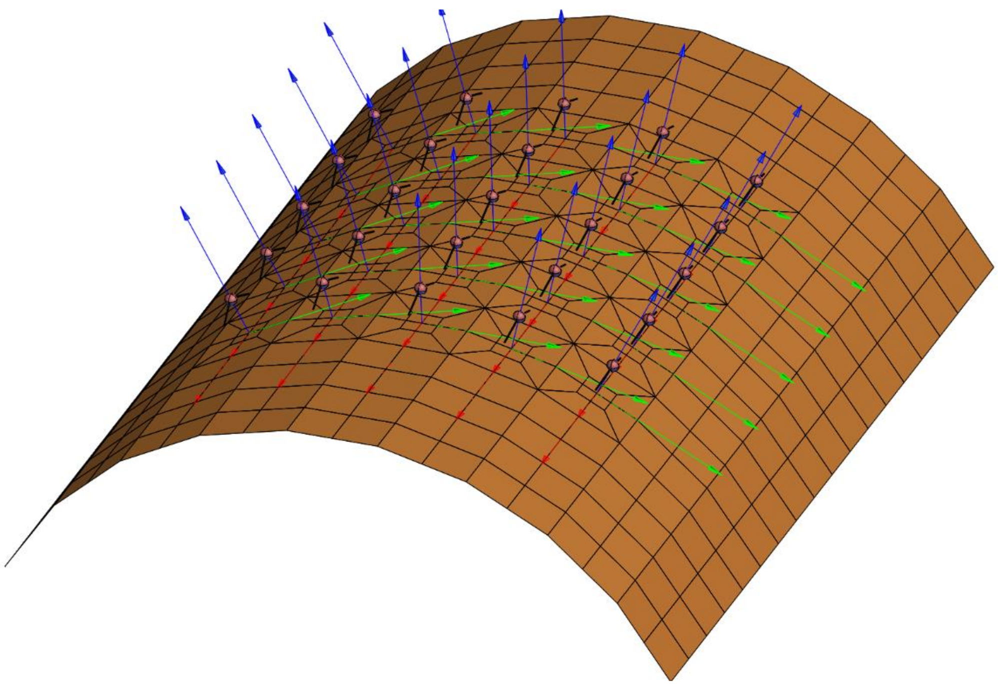


Figure 26 25-element tilted dipole spaced 0.65λ array over cylindrical ground plane R = 80 cm extended by 56 empty patches.

Arrays of Cup Dipoles

Single Element Design

A dipole mounted in a circular cavity (cup), Section 5-11 and supplemental material, is an ideal primary focus feed for a reflector. GMDIPOLE includes an optional circular cavity in the ground plane mesh file for GRASP MoM. A design with a 0.7λ diameter cavity, 0.5λ deep, using a 0.4λ long dipole mounted 0.21λ above the of the cavity was selected from the supplemental catalog of designs. The NEC wire analysis of the antenna predicts nearly equal beamwidths in all planes an optimum to feed a reflector with *f/D* = 0.4. We use the following input file to GMDIPOLE (with added explanations).

4 cm
gmdipol4.txt list of partial TOR objects for use in TICMARW
1 ground plane every element

```
dipgrd4.msh     mesh output file containing ground plane and cavity
12.3            dipole length
.125            dipole radius
6.3             height over gr.pl.
0,0             tilt towared gr.pl., Phi rotation
1               single dipole
23.,23          ground plane height, top width
23.             center width of gr.pl.
yes             circular cup
21.,15.         cup diameter, height
dipole_coor     object file of element coordinate system incremented for each element
dipl_grd        tabulated mesh object name incremented
xdipole         Straight wire element incremented
```

We add the count of the number of lines in each GRASP TOR object to the output file: gmdipol4.txt.

```
6
dipl_grd  tabulated_mesh
(
  coor_sys        : ref(dipole_coor)
  file_name       : dipgrd4.msh
)

14
xdipole  piecewise_straight_wire
(
  coor_sys        : ref(dipole_coor),
  nodes           : sequence
    (
     struct(x: -6.150000E+00 cm, y:  0.000000E+00 cm, z:  6.300000E+00 cm),
     struct(x: -6.150000E-01 cm, y:  0.000000E+00 cm, z:  6.300000E+00 cm),
     struct(x:  0.000000E+00 cm, y:  0.000000E+00 cm, z:  6.300000E+00 cm),
     struct(x:  6.150000E-01 cm, y:  0.000000E+00 cm, z:  6.300000E+00 cm),
     struct(x:  6.150000E+00 cm, y:  0.000000E+00 cm, z:  6.300000E+00 cm),
    ),
  radius          :  1.250000E-01 cm
)
```

We need to use the program TICMARW with this output file and an array with a single element (*.ISP and *.EXI) files to generate the edit file for an existing GRASP project TOR file.

ticmarwd4.txt	TOR edit file with cavity mounted dipole
dipol2.isp	Single element GRASP array geometry file
1	# array elements
dipole_elem	scatterer cluster name in GRASP project
2	# of scatterers in each element
dipl_grd	name of mesh ground plane and cup GRASP object
xdipole	name of piecewise linear wire GRASP object
gmdipol4.txt	modified output file from GMDIPOLE
1	add voltage generator
dipol2.exi	Single element GRASP array excitation file
0.,0.,6.3	feed position on element
Generator	voltage feed GRASP object name
array_spar	impedance parameters GRASP output object
0	no other voltages

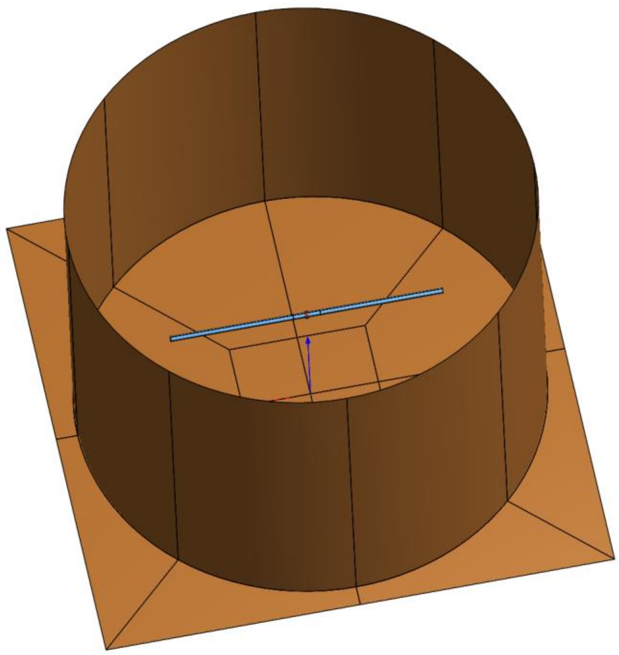


Figure 27 GRASP MoM model of Dipole mounted in Cavity

Figure 28 gives the GRASP MoM analysis of the cup-dipole to linear polarization while figure 29 shows the corresponding NEC (MoM) analysis using a wire mesh cavity in the model. Two methods produce very similar patterns. Both methods predict a resonance impedance with low reactance, but differ in the value of resistance. The exact procedure in the two MoM analyses for computing input resistance differ which we can expect from different MoM solutions. When we add the feeding balun in the final antenna, it will need to be tuned to include all parts of the design.

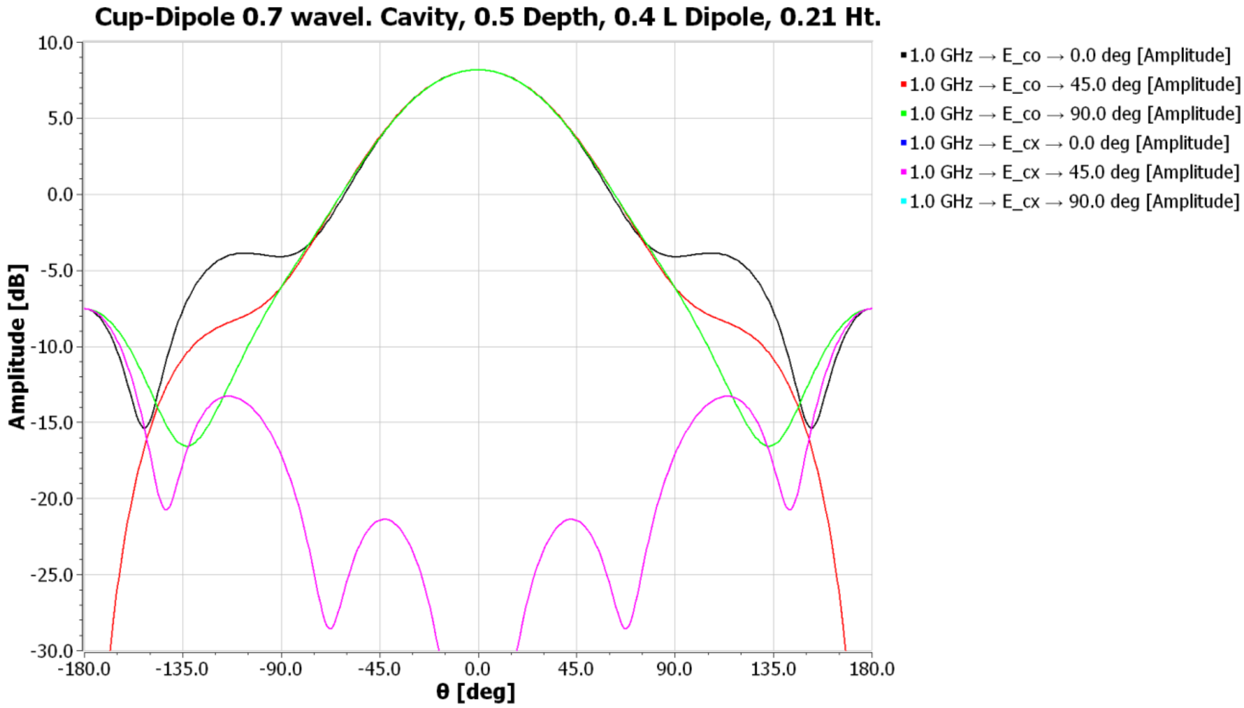


Figure 28 GRASP MoM pattern of Dipole mounted In Cavity

Linear Polarization Blue: ϕ = 0, Red: ϕ = 90, Black: ϕ = 45

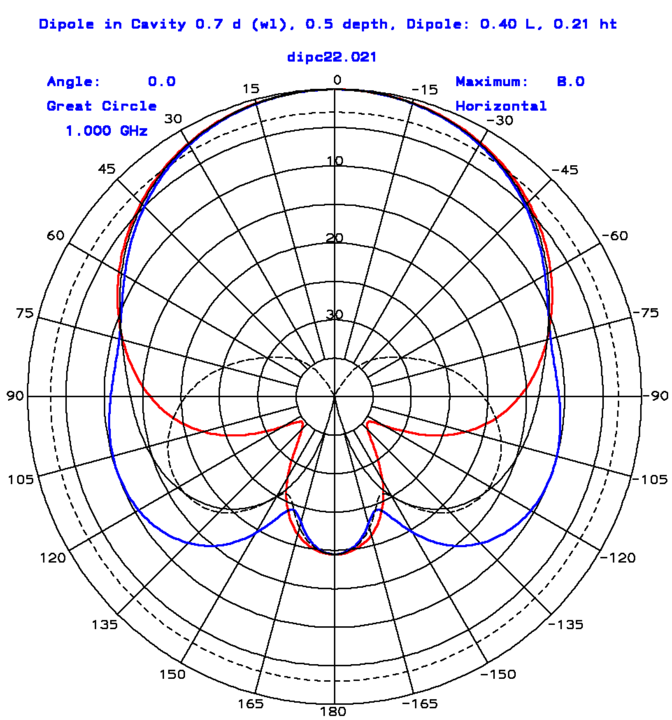


Figure 29 NEC wires (MoM) Analysis of Dipole mounted in Cavity

Cup Dipole Array

To design an array of cup dipoles we use XADEF to generate a 25-element evenly spaced array with 23 cm spacings and feed it uniformly. We run TICARR to write the GRASP array geometry file cup_dipole_25.isp and the corresponding excitation file (*.exi). The excitation files from the flat 25-element dipole array can be reused. This includes the cases where either the center element (13) or corner element amplitude is set to 100 dB to overshadow all other elements when used in TICARR for the excitation for the active- or scan-element pattern. The GRASP objects for the single element can be reused in the input file to TICMARW to generate the TOR file edits.

ticmarwd8.txt TOR file edits output file
cup_dipole_25.isp GRASP array geometry file
25 # array elements
dipole_elem scatterer cluster name in GRASP project
2 # of scatterers in each element
dipl_grd name of mesh ground plane and cup GRASP object
xdipole name of piecewise linear wire GRASP object
gmdipol4.txt modified output file from GMDIPOLE
1 add voltage generators
dipol1.exi uniform 25-element excitation file
0.,0.,6.3 feed position on element
Generator voltage feed GRASP object name
array_spar impedance parameters GRASP output object
dipol113.exi central element (13) excitation of 25 elements
0.,0.,6.3 feed position on element
generator_13 voltage feed GRASP object name
array_spar13 impedance parameters GRASP output object
1 add voltage generators
dipol11.exi corner element (1) excitation of 25 elements
0.,0.,6.3 feed position on element
generator_1 voltage feed GRASP object name
array_spar1 impedance parameters GRASP output object
0 no other voltages

We run the single element GRASP project and use “save as” to generate the 25-element project. The output of TICMARW is used to edit the project TOR file. When we open the GRASP project, it contains all 25 elements, Figure 30.

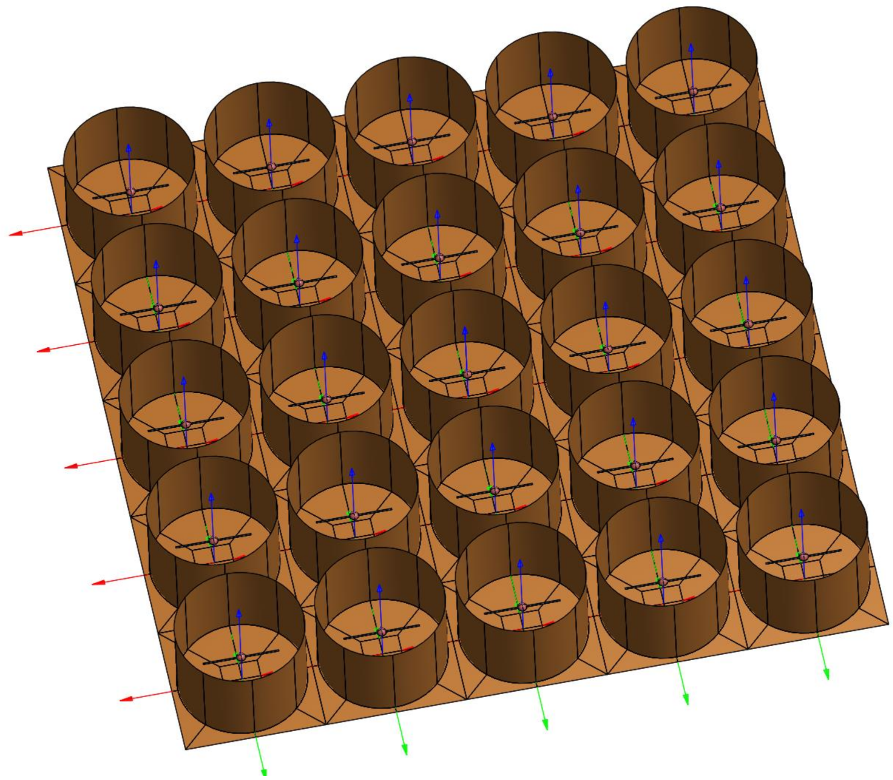


Figure 30 25-element array of cup dipoles

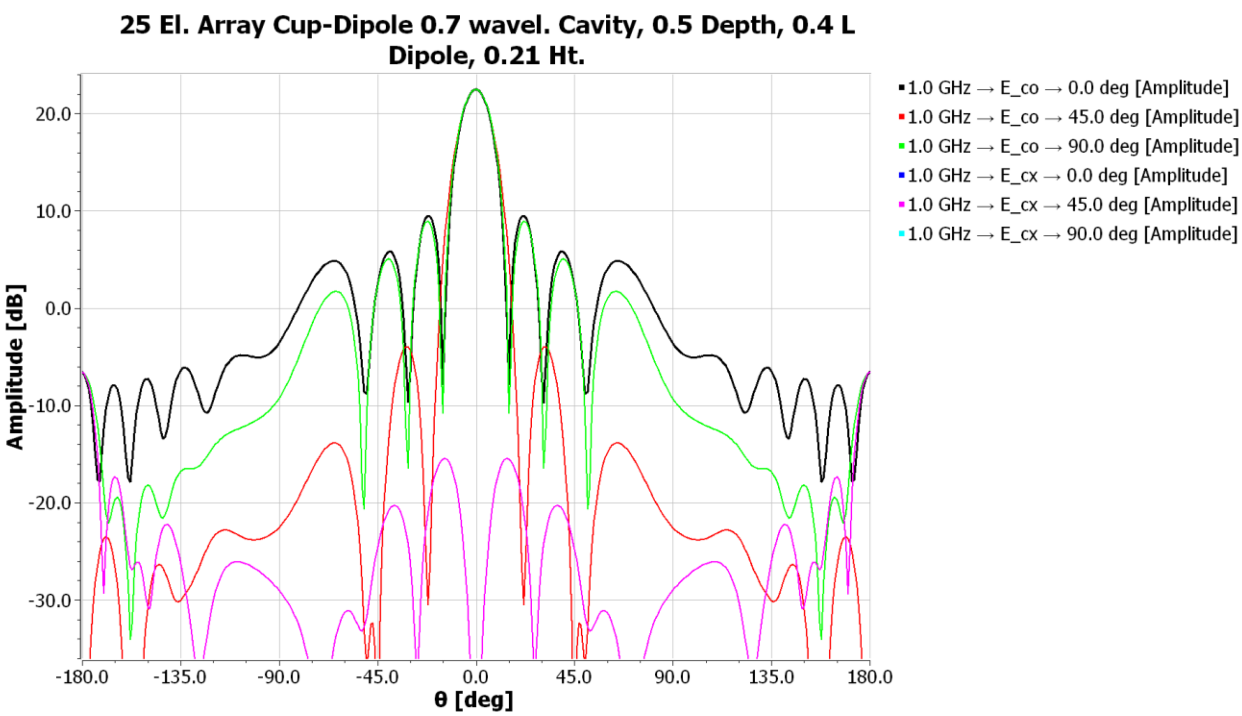


Figure 31 Pattern of 25-element uniformly fed array of cup dipoles

The peak gain of figure 31 is 22.44 dB while the element pattern, figure 28, is 8.17 dB. When we add the 25-element factor of 13.97 dB to the element pattern gain, the 22.15 dB is 0.29 dB less than the actual gain. As shown in Section 4-28 the element spacing and coupling between elements affects realized gain. The cup dipoles are aperture elements, like horns, and we expect gain close to (element gain) X (number of elements). The difference is small but not zero.

The active- or scan-element patterns of individual elements where one element is fed and the others loaded show deviations figures 32 and 33 from the isolated element, figure 28. The full UV contours, figures 34 and 35 illustrate the effects of element coupling on these patterns due to the radiation of neighboring elements.

Both figure 32 and figure 34 show a boresight pattern dip which means other elements in the array must have increased gains to make up the difference to produce the boresight gain, figure 31, slightly greater than the sum of 25 isolated elements. The UV-contour plots of these two active- or scan-element antennas show less variation than dipoles on a flat ground plane because the cup modifies the element to become closer to an aperture element. Larger apertures better isolate array elements and produce gains due to the sum of individual isolated elements.

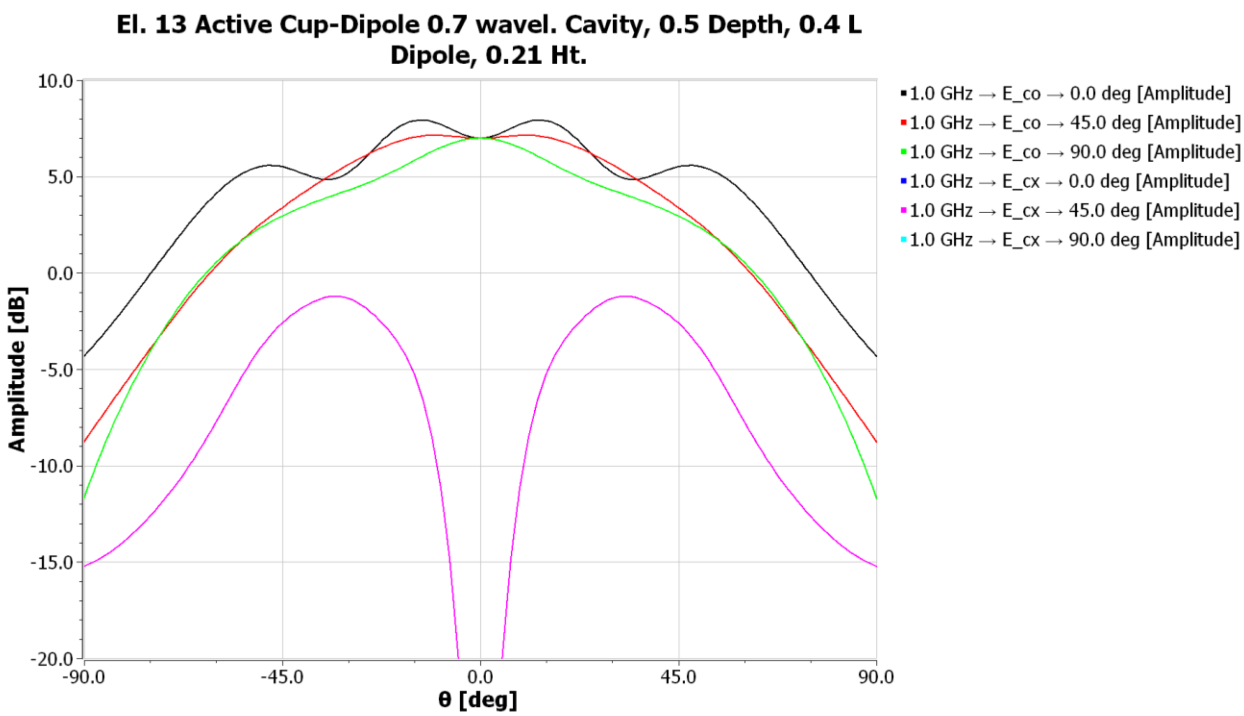


Figure 32 Active- or scan-element pattern of central element (13) in 25-element cup dipole array

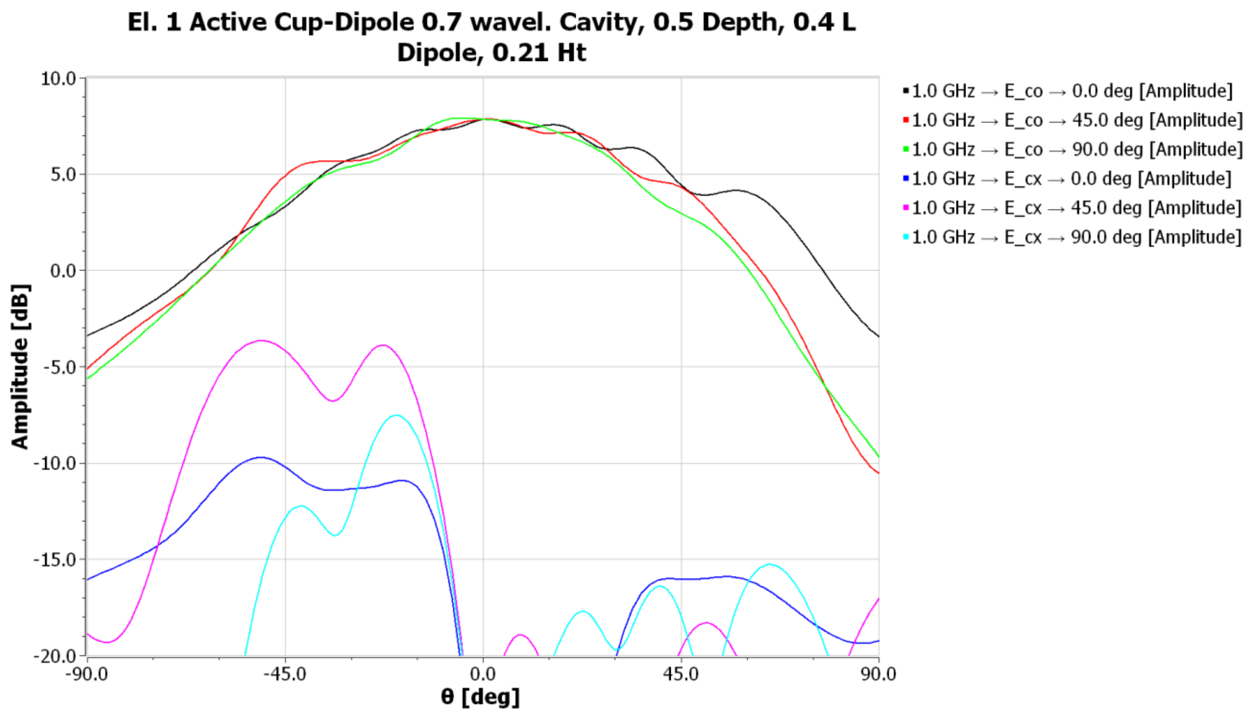


Figure 33 Active- or scan-element pattern of corner element (1) in 25-element cup dipole array

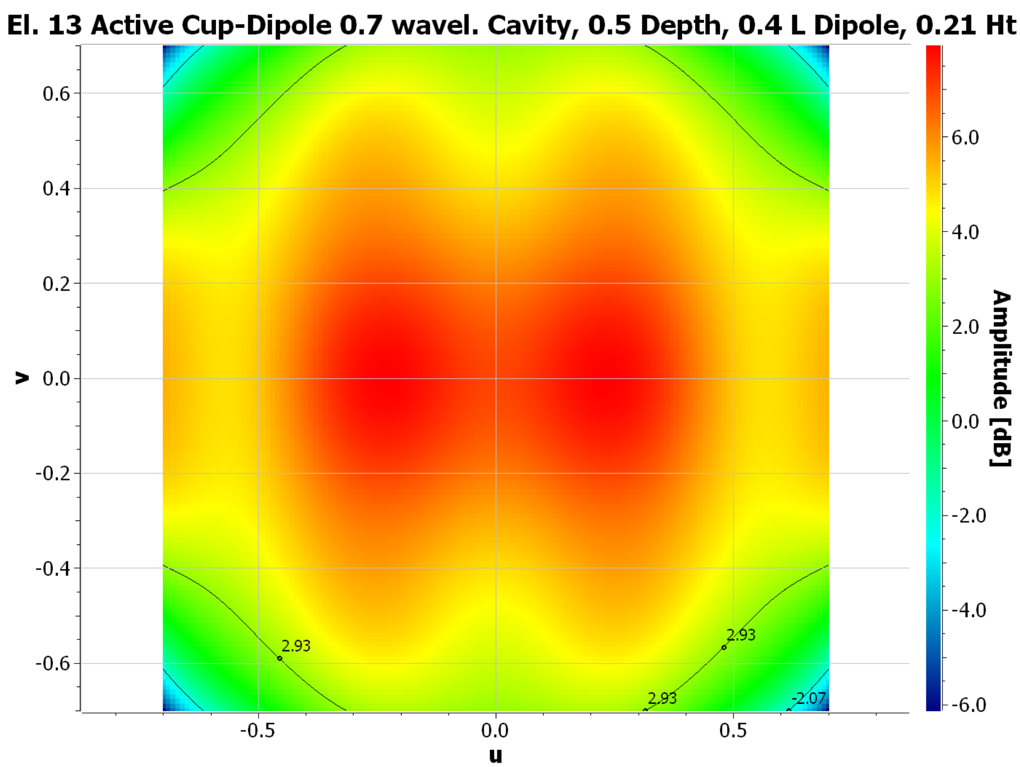


Figure 34 Active- or scan-element UV-contour of central element (13) in 25-element cup dipole array

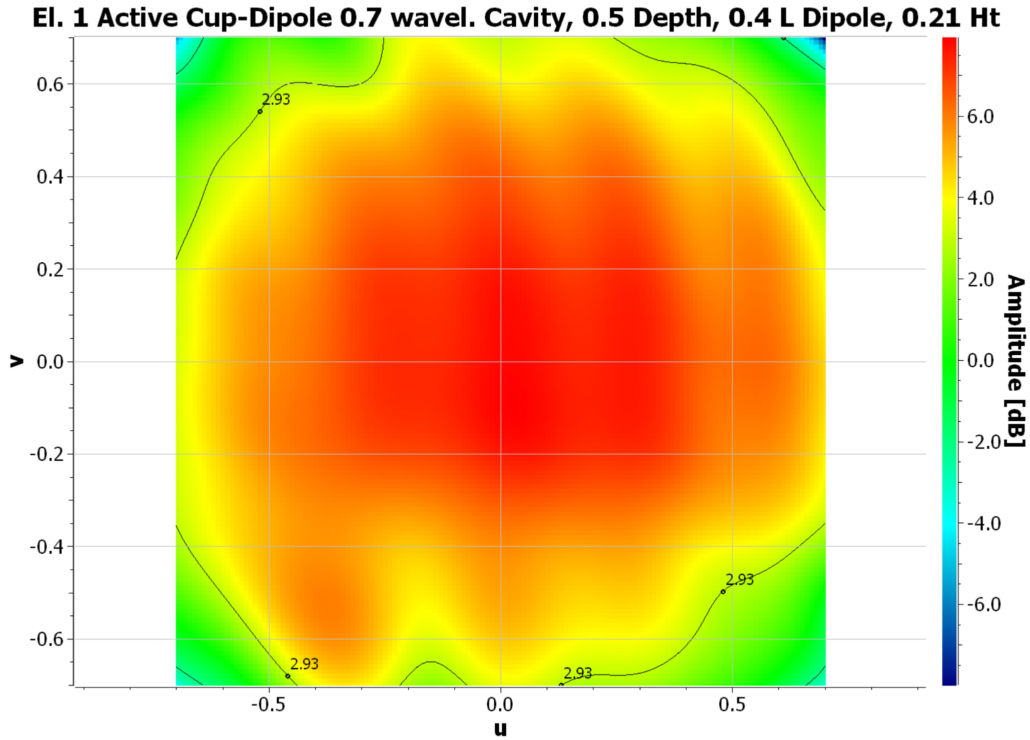


Figure 35 Active- or scan-element UV-contour of corner element (1) in 25-element cup dipole array

Arrays of Corner Antennas

Single Element Design

A simple model of a corner reflector has a piecewise wire element dipole and two meshed plates. GMCORNR generates the GRASP MoM model of a single corner reflector. The model has the option of a flat center plate and allows the angle between the sides to be arbitrary. Section 5-8 contains examples of various angled sides corner reflector and pattern results versus size and plate size. The program generates an output which needs to be used with TICMARW to generate an edit file for the GRASP project. Here is an input file:

4	cm units
gmcornr1t.txt	GRASP TOR objects to be used with TICMARW
cornplt1.msh	GRASP MoM mesh file
13.1	dipole length
0.125	dipole radius
11.25	dipole to vertex
0	dipole tilt
1	single dipole
30.,30.	corner reflector ht (x-axis), side plate (y-axis)
90	angle between plates (degrees)
0.	center plate width (y-axis) (none)

dipole_coor   object file of element coordinate system incremented for each element
cornr_grd   tabulated mesh object name incremented
xdipole   Straight wire element incremented

The output file gmcornr1t.txt needs to be edited to add the number of lines for each GRASP object. Except for the corner reflector mesh file, it has the same line counts as one for a dipole over a ground plane.

```
6
cornr_grd  tabulated_mesh
(
 coor_sys       : ref(dipole_coor)
 file_name      : cornplt1.msh
)

14
xdipole  piecewise_straight_wire
(
 coor_sys       : ref(dipole_coor),
 nodes          : sequence
  (
   struct(x: -6.550000E+00 cm, y:  0.000000E+00 cm, z:  1.125000E+01 cm),
   struct(x: -6.550000E-01 cm, y:  0.000000E+00 cm, z:  1.125000E+01 cm),
   struct(x:  0.000000E+00 cm, y:  0.000000E+00 cm, z:  1.125000E+01 cm),
   struct(x:  6.550000E-01 cm, y:  0.000000E+00 cm, z:  1.125000E+01 cm),
   struct(x:  6.550000E+00 cm, y:  0.000000E+00 cm, z:  1.125000E+01 cm),
  ),
 radius         :  1.250000E-01 cm
)
```

An existing GRASP MoM project, for example, a single dipole is opened and “saved as” a corner reflector file to have its TOR file edited by the output of TICMARW to generate the corner reflector, Figure 36. The pattern, Figure 37, matches the analysis using iterative PO in pattern shape, peak gain, and backlobe for the same size antenna. If we generate an array of these element they are spaced 1λ in the *E*-plane and $\sqrt{2}\lambda$ in the *H*-plane. We use XADEF to generate an array and follow it up by TICARR to convert the array geometry output to GRASP *.isp array geometry and *.exi excitation files. After we run TICMARW using inputs for the array geometry and excitations, we generate the TOR file edits to modify an existing project.

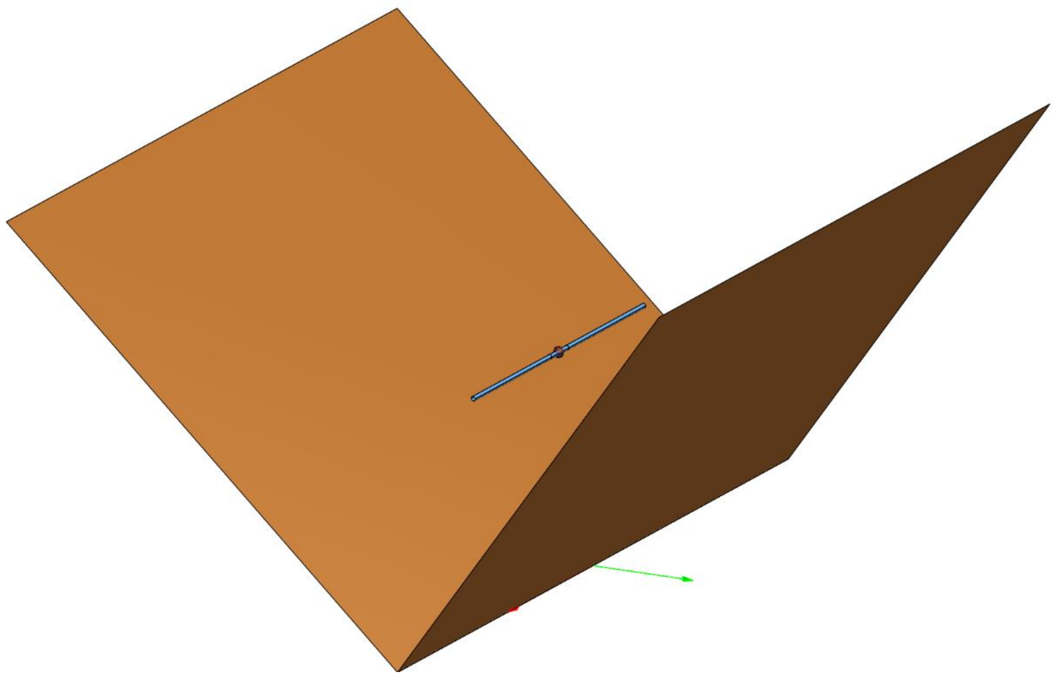


Figure 36 GRASP MoM model of corner reflector

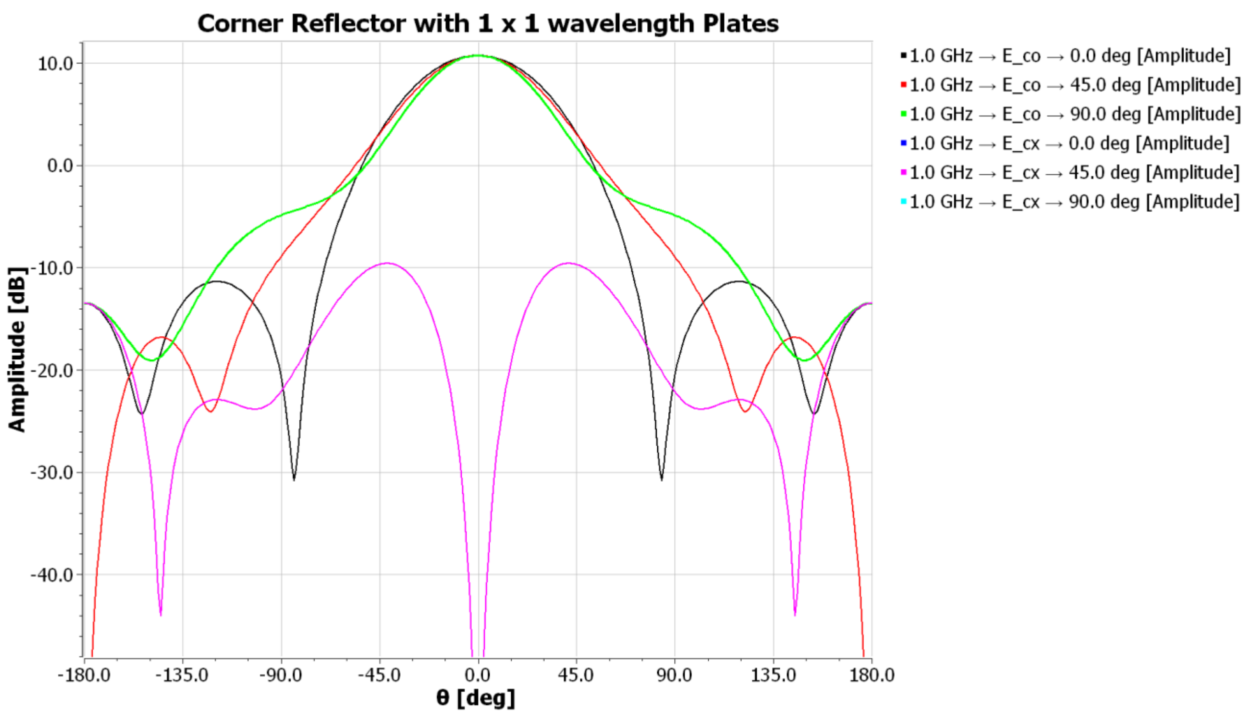


Figure 37 Single corner reflector pattern (10.72 dB gain)

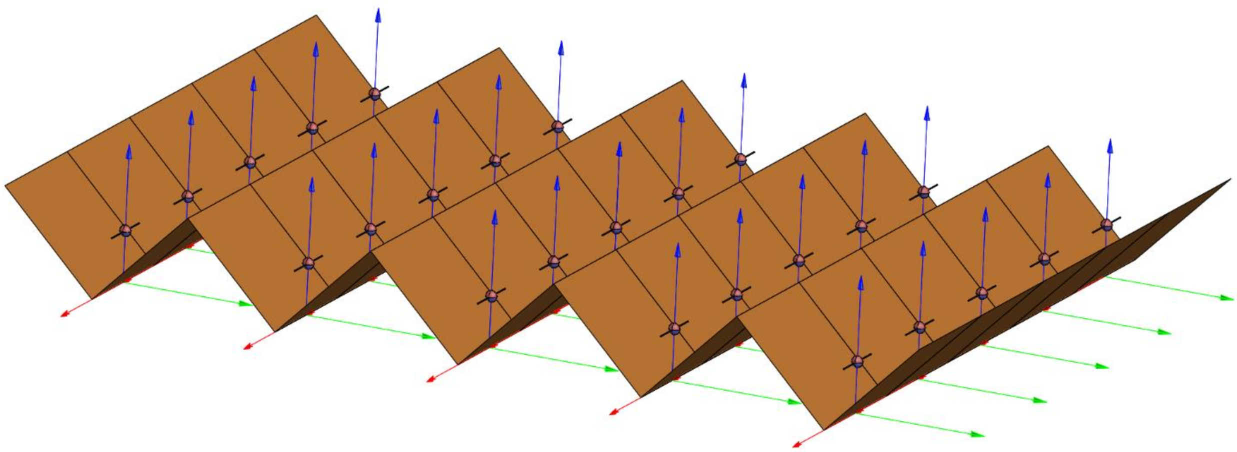


Figure 38 25-element Corner Reflector Array for 1λ x 1.4142λ spacings

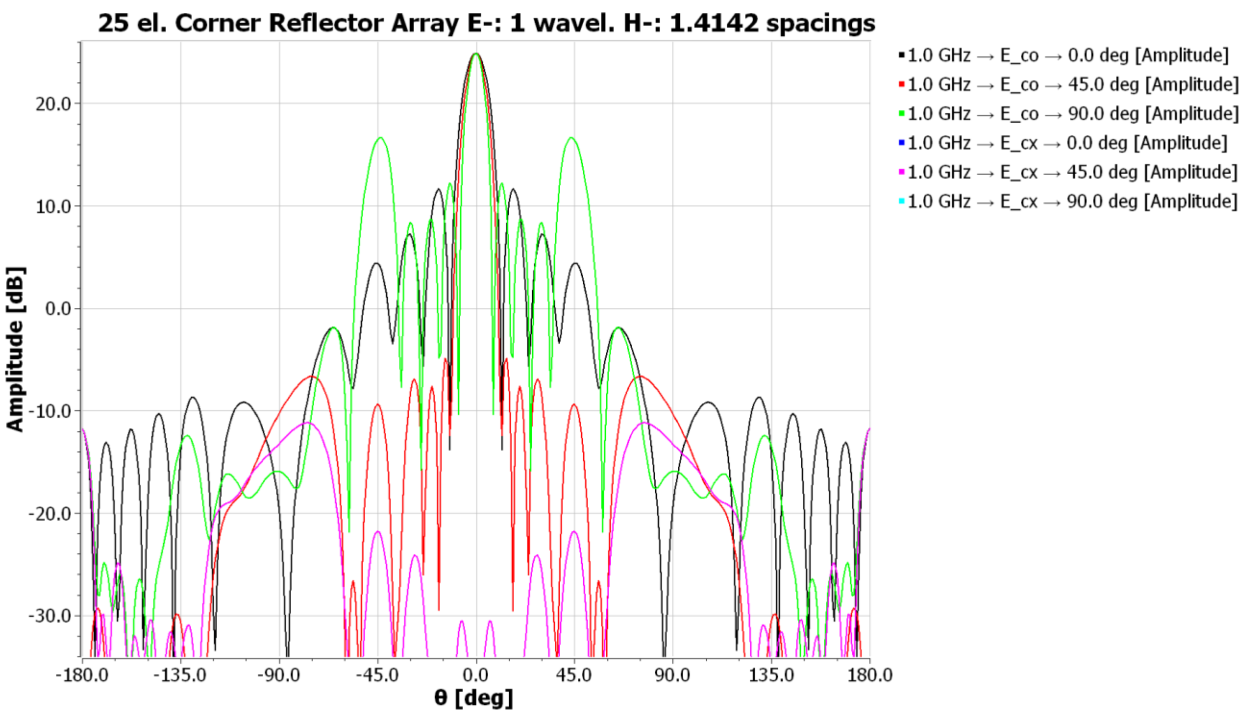


Figure 39 25-element Corner Reflector Array with 1λ x 1.4142λ spacings pattern 24.90 dB gain.

Element gain times number of elements gives a gain of 24.70 dB. Even though the pattern has gratings lobes in the *H*-plane, it produces a pattern with 0.2 dB greater gain than the simple (element power gain) x (number of elements).

To reduce the grating lobes in the *H*-plane we can reduce the length of the plates in the corner reflector to 0.707λ and reduce the gain of the single element (Figure 40) to 9.78 dB. A new array geometry is generated with 30 cm spacings (1λ) and TICMARW run to generate the edit file for the TOR file.

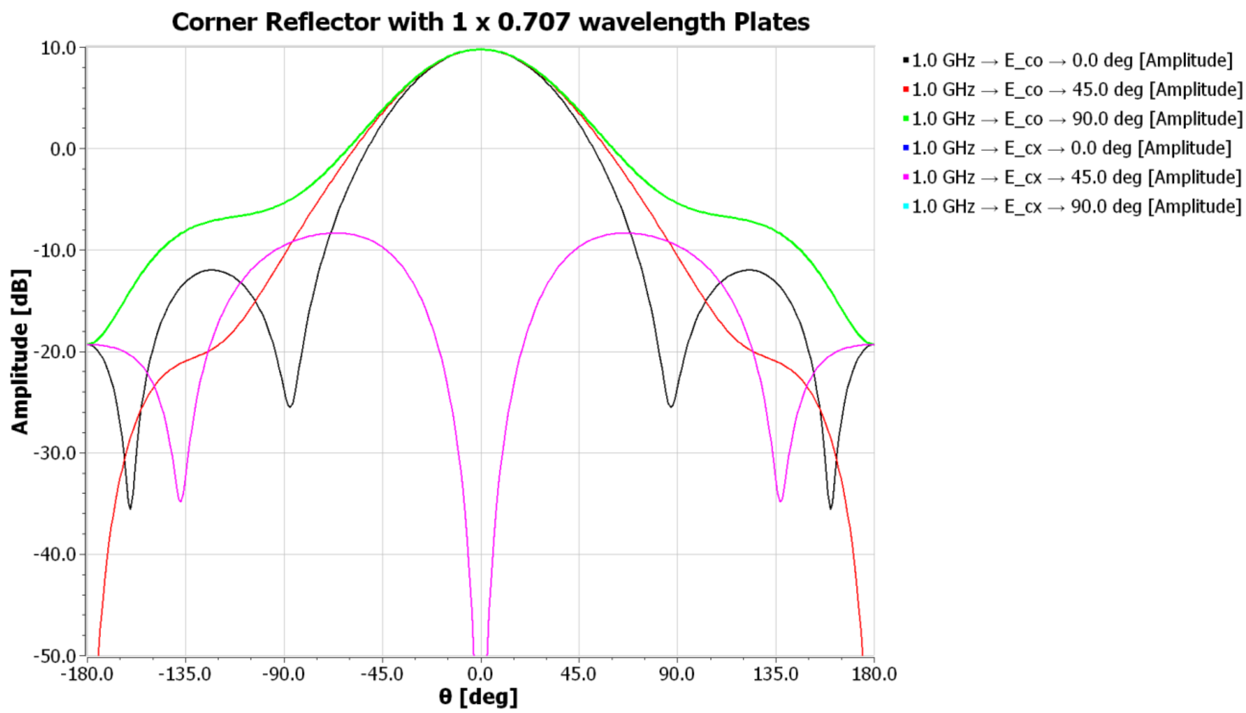


Figure 40 Corner reflector with 0.707λ long plates and 1λ width (9.78 dB gain)

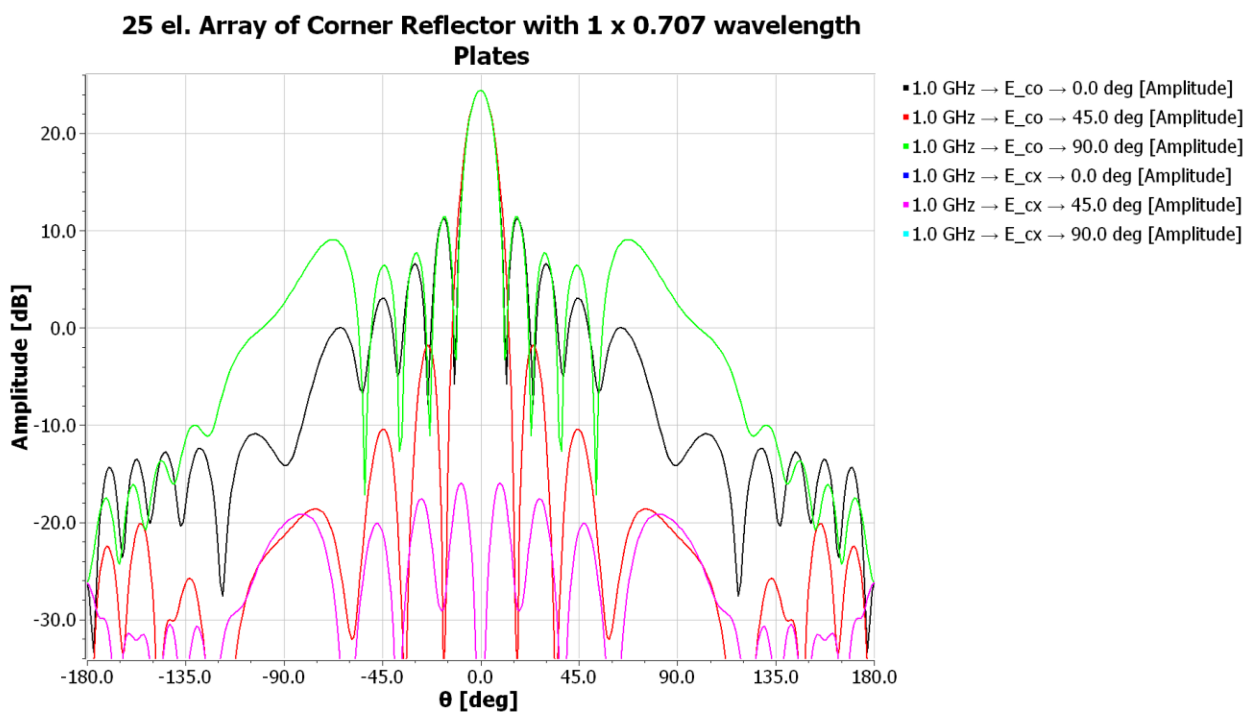


Figure 41 25-element Corner Reflector Array with 1λ x 1λ spacings pattern 24.42 dB gain.

The (element pattern gain) x (number of elements) gives a gain of 23.76 dB or a 0.66 dB difference from the actual gain. The H-plane grating lobes have been reduced by 7.6 dB and shifted to larger angle due to the closer element spacing.

Arrays of Helical Wire Antennas

Single Element Design

The design program generates the GRASP objects and gathers them into a scatter cluster so that it can be easily arrayed. This means an element position file (*.isp) and excitation file (*.exi) must be written even for a single element placed at the origin. Each element scatter cluster includes an individual coordinate system which the arraying program alters to place and orientate the element in the array. The program GMHELIX generates the GRASP objects for a MoM solution of an axial mode helix. The program can be run either by hand answering questions when running the program or by a designated input test file. Here is an example input text file (with explanations not in file):

4	units: cm
cup_helix_8.txt	output file with TOR file elements
cup_helix8.msh	TOR mesh file of ground plane
4.77	helix diameter 1 wavel. Circumference 2 GHz
.09	wire radius
.10	vertical probe length
13,6	turns, points/turn
2	pitch of initial impedance matching helix
.25,2	turns, pts. in initial impedance matching helix
14,1	helix pitch angle, sense: 1 RH
60,1.5	final taper (%), turns
15,15	ground plane Y-axis, X-axis (top)
15	X-axis center width (square)
yes	include cup
10,4.	cup diameter, height
helix_coord	object file of element coordinate system incremented for each element
cup_helix	tabulated mesh object name incremented
feed_probe	Straight wire element incremented
helix_curve	Curved wire object of helix incremented

The input file specifies a helical curved wire fed from a vertical probe in the ground plane of length 0.10 cm. The total helix has 13 turns modeled using 6 points around each turn. The number of points around the turn of the helix should be no less than 4. The helix can be impedance matched by starting with a helical section close to the ground plane (2° helix) over a length of ¼ turn. This design uses a tapered section over the last 1 ½ turn which reduces the diameter by 60%. The tapered end reduces the traveling wave current along the wire reflection from the end of the helix which generates cross polarization. The ground plane is 15- by 15-cm. If the center width does not equal the top width, a hexangular ground plane can be generated. This design includes a cup around the helix to reduce side and back radiation. If the answer is no cup, then the line specifying diameter and height is removed. The last lines four specify the names given objects in the TOR file incremented for each element.

The output file of GMHELIX: cup_helix_8.txt requires specifying the number of lines in each object and adding the numbers shown in red below when used in the arraying program TICMARW. The number of points in the curved wire element is an output of GMHELIX; normally cup_helix has 6 lines in the TOR file and the feed_probe 11 lines. The program TICMARW uses the modified output of GMHELIX to generate the TOR file of the full array.

```
6
cup_helix  tabulated_mesh
(
  coor_sys        : ref(helix_coord)
  file_name       : cup_helix8.msh
)

11
feed_probe  piecewise_straight_wire
(
  coor_sys        : ref(helix_coord),
  nodes           : sequence
    (
     struct(x:  2.385000E+00 cm, y:  0.000000E+00 cm, z:  0.000000E+00 cm),
     struct(x:  2.385000E+00 cm, y:  0.000000E+00 cm, z:  1.000000E-01 cm),
    ),
  radius          :  9.000000E-02 cm
)

97
helix_curve  curved_wire
(
  coor_sys        : ref(helix_coord),
  nodes           : sequence
    (
     struct(x:  2.385000E+00 cm, y:  0.000000E+00 cm, z:  1.000000E-01 cm, radius:  9.000000E-02 cm),
     struct(x:  1.686450E+00 cm, y:  1.686450E+00 cm, z:  1.654127E-01 cm, radius:  9.000000E-02 cm),
     struct(x: -1.042517E-07 cm, y:  2.385000E+00 cm, z:  2.308254E-01 cm, radius:  9.000000E-02 cm),
.
.
.
     struct(x:  1.431000E+00 cm, y:  3.935006E-06 cm, z:  4.668523E+01 cm, radius:  9.000000E-02 cm),
    )
)
```

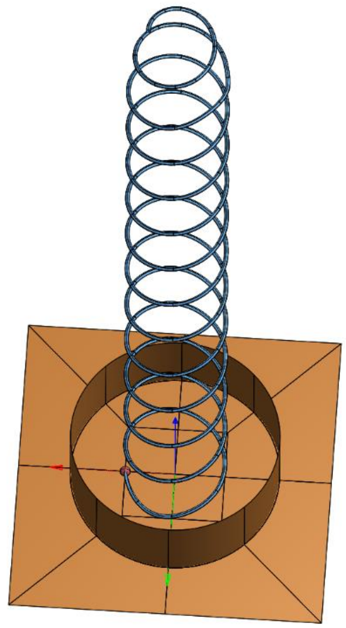


Figure 42 13 turn helix antenna in cup specified above and generated by GRASP from the TOR file.

An existing TOR is modified to add the new design by using the output from TICMARW. A new GRASP project is generated from an old one ("save as"). Both the output file of TICMARW and the TOR file are opened in a text editor. The complete output of the TICMARW file is copied and replaces the upper portion of the TOR file where the frequency and pattern specification objects are retained. The new GRASP project can be modified in the GRASP GUI to change the frequency and pattern angles.

Figure 43 shows the pattern response computed by GRASP MoM at 2 GHz. The ground plane is 1λ x 1λ with a cup 2λ/3 in diameter and 0.266λ high. The peak gain is 12.88 dB. The UV contour plot of Figure 44 illustrates the axisymmetric nature of the pattern including the sidelobes.

The combination of the ground plane and cup increases the Front/Back (F/B) to 26.66 dB for the LHC cross polarization. An analysis below without the cup (Figure 45) shows decreased F/B of 16.53 dB (Figure 46). The cup also reduces the pattern near 90° from boresight. The ground plane cup shields part of the radiation from the helix and reduces boresight gain by 0.36 dB.

The helix end taper reduces cross polarization. Figure 43 of the tapered end helix has a boresight cross polarization of -34.2 dB while a helix without the taper (Figure 48) has -28.3 dB cross polarization. The difference is minor for this long helix. Tapering the helix end has decreased gain by 0.14 dB.

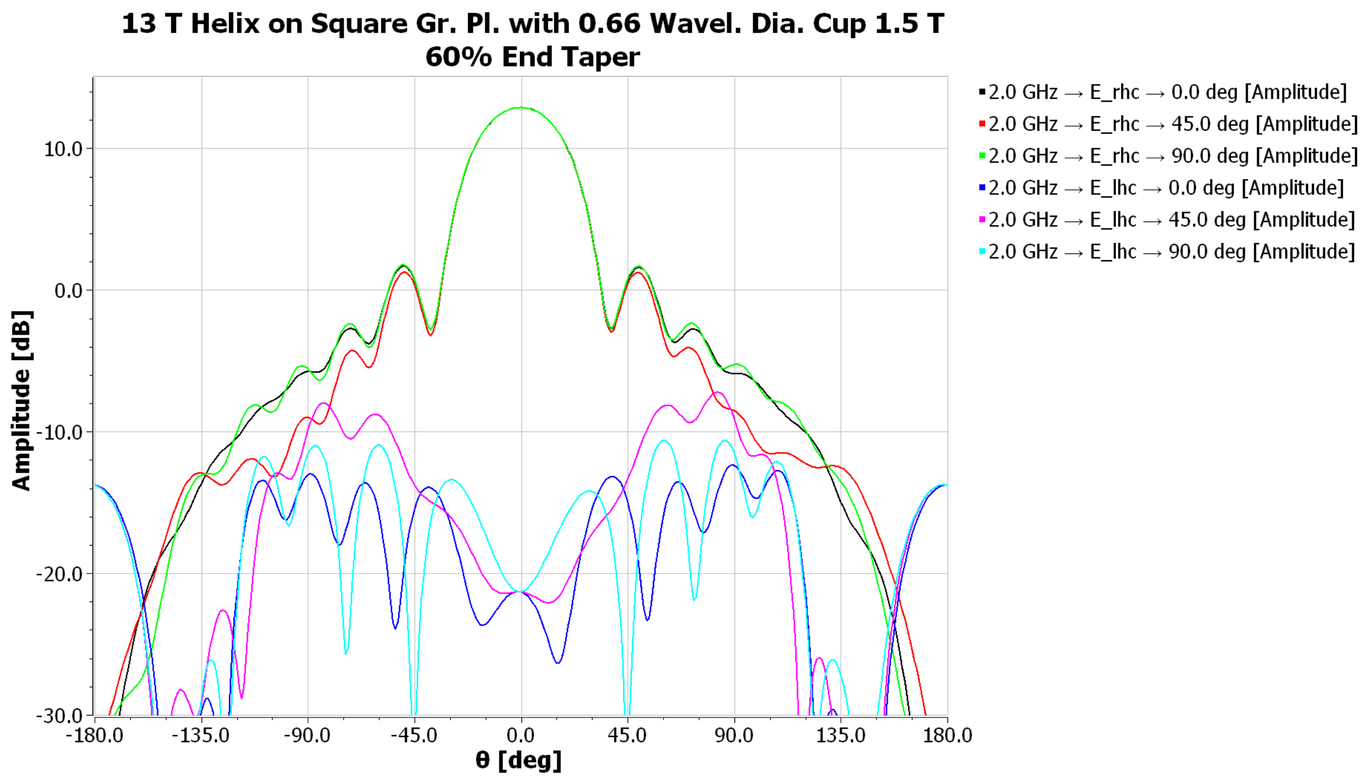


Figure 43 Pattern of 13 turn helix antenna in cup specified above and generated by GRASP MoM

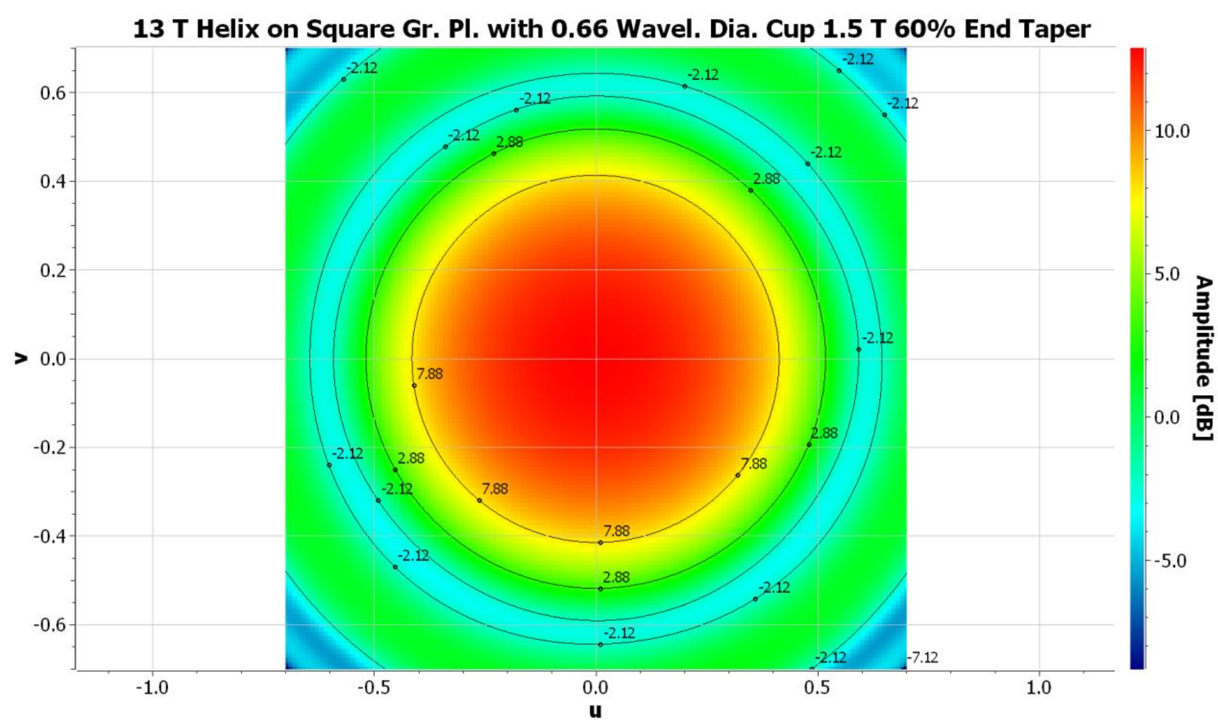


Figure 44 UV-contour pattern of 13 turn helix antenna in cup specified above and generated by GRASP MoM

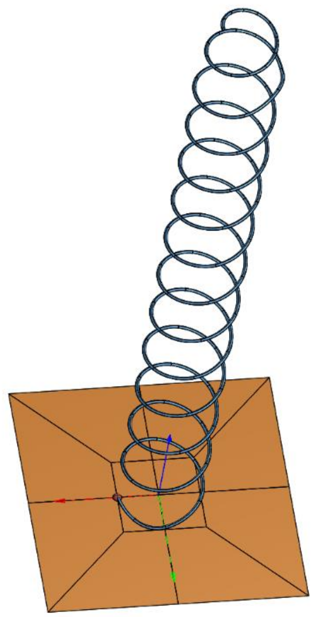


Figure 45 13 turn helix antenna without cup and the same as specified above; generated by GRASP MoM TOR file

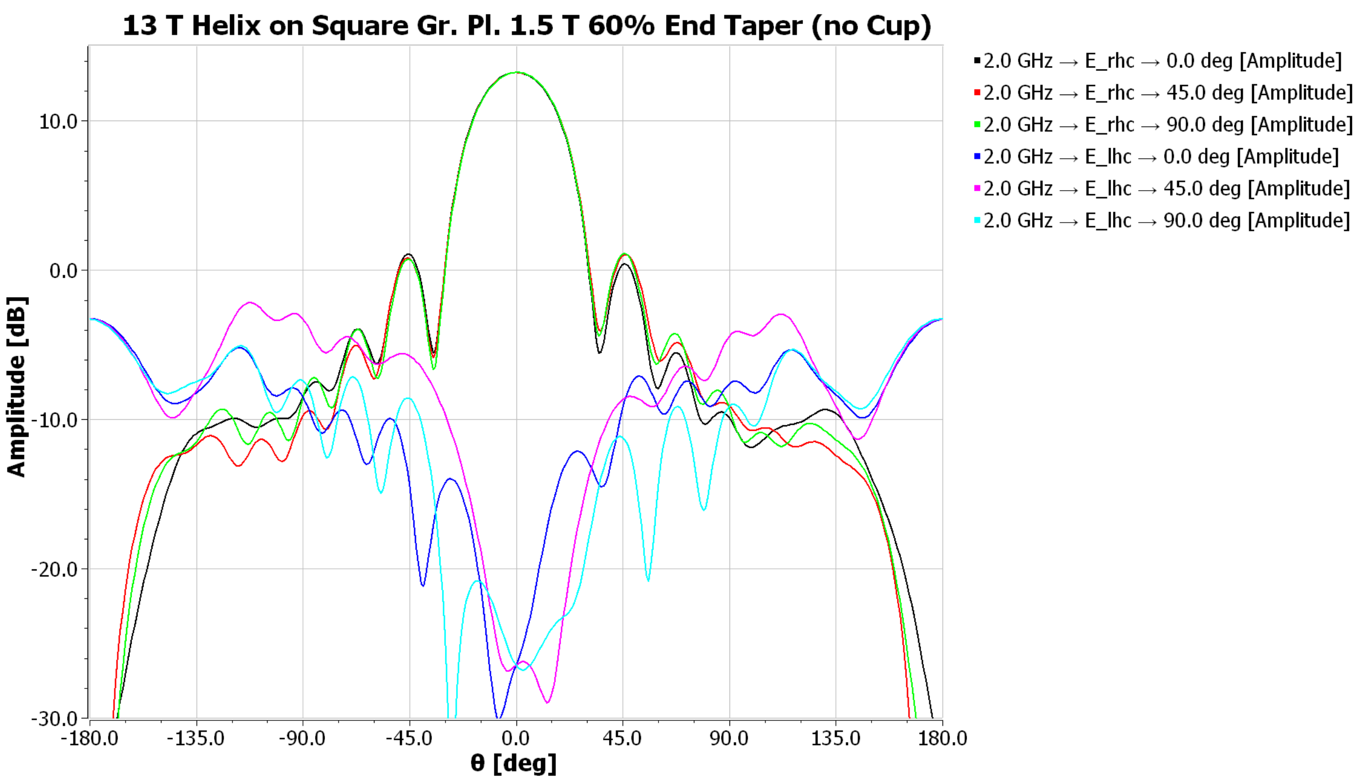


Figure 46 Pattern of 13 turn helix antenna without cup specified above and generated by GRASP MoM

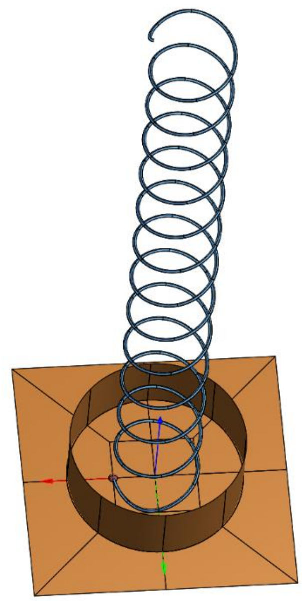


Figure 47 13 turn helix antenna in cup specified above without helix taper and generated by GRASP from the TOR file.

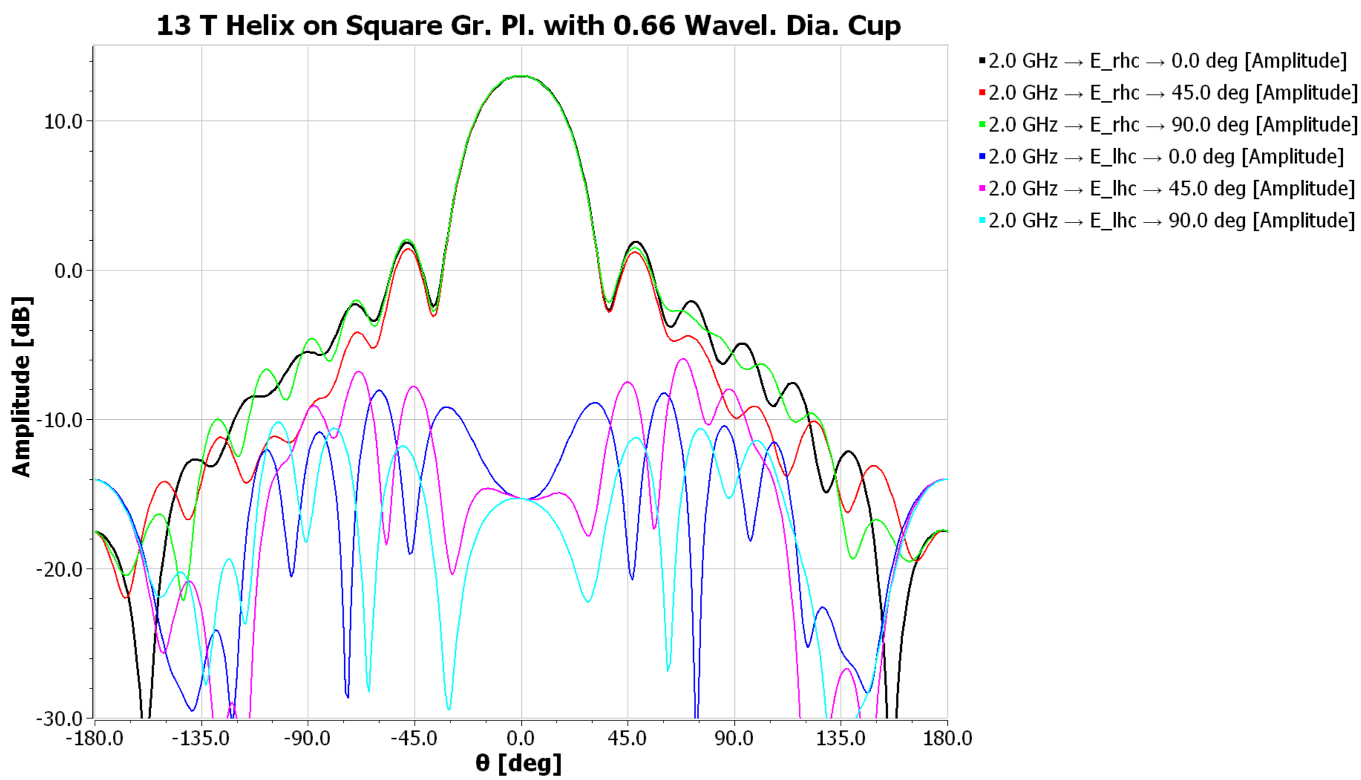


Figure 48 Pattern of 13 turn helix antenna in cup specified above with no helix end taper and generated by GRASP MoM

Helix Array

The program XADEF generates the position of elements including pointing direction. By using the program TICARR the GRASP input file of array positions (*.isp) is created along with multiple excitation files (*.exi) for various array scan directions. The program TICMARW generates the TOR file additions to specify the array and its excitations for GRASP. TICMARW can be run using an input text file with answers to program inputs or it can be run by hand inputs. The following input text file are the inputs for a planar array of helices with 25 elements. Comments in red have been added to denote inputs which would be confused as added to the name and not present in the input text file.

```
ticmarwh25f.txt      Output text file of TOR file edits
helix25a.isp         Array position and orientation input file to GRASP
25                   # array elements
helix_elem           Scatter cluster name for array elements
3                    # of scatterers in each element
cup_helix            1st name of scatter element (*. msh, mesh element)
feed_probe           2nd name of scatter element: straight wire feed probe
helix_curve          3rd name of scatter element: curved wire element
cup_helix_8.txt      Output of GMHELIX containing TOR elements
1                    add voltage generators
helix25a.exi         Excitation input file
2.385,0.,0.          feed position on element
Generator            Name of voltage generator
array_spar           Name of S-parameter output
1                    add voltage generators
helix25s.exi         Excitation file feeding only central element
2.385,0.,0.          feed position on element
generator_sing       Name of voltage generator
array_spar           Name of S-parameter output
0                    no other voltages
```

The ticmarwh25f.txt has the following elements:

The initial lines generate the scatter cluster with all the array elements

```
helix_elem         scatterer_cluster
(
  scatterers     :
sequence(ref(elem1),ref(elem2),ref(elem3),ref(elem4),ref(elem5)
,ref(elem6),ref(elem7),ref(elem8),ref(elem9),ref(elem10),ref(elem11)
,ref(elem12),ref(elem13),ref(elem14),ref(elem15),ref(elem16),ref(elem17)
,ref(elem18),ref(elem19),ref(elem20),ref(elem21),ref(elem22),ref(elem23)
,ref(elem24),ref(elem25))
)

elem1  scatterer_cluster
(
```

```
 scatterers     :
sequence(ref(cup_helix1),ref(feed_probe1),ref(helix_curve1))
)

elem2  scatterer_cluster
(
 scatterers     :
sequence(ref(cup_helix2),ref(feed_probe2),ref(helix_curve2))
)
```

Two of the 25 element scatter clusters are shown which includes the cup ground plane mesh file, the straight wire probe element, and the curved wire helix element.

Each element has its own coordinate system which locates and rotates its pointing, if necessary. Starting with the total array coordinate system as the base, the first two elements coordinate systems are (followed by the others):

```
array_coor_sys   coor_sys
(
 origin         : struct(x: 0.0 m, y: 0.0 m, z: 0.0 m),
)

coor_sys_elem1  coor_sys_grasp_angles
(
 origin        :struct(x: -3.00000E+01 cm, y: -3.00000E+01 cm, z:  0.00000E+00 cm),
 theta         :    0.000,
 phi           :    0.000,
 psi           :    0.000,
 base          : ref(array_coor_sys)
)

coor_sys_elem2  coor_sys_grasp_angles
(
 origin        :struct(x: -1.50000E+01 cm, y: -3.00000E+01 cm, z:  0.00000E+00 cm),
 theta         :    0.000,
 phi           :    0.000,
 psi           :    0.000,
 base          : ref(array_coor_sys)
)
```

Each element has its own MoM elements centered on an individual coordinate system. Note the mesh file is the same for all elements. The first two elements are given as:

```
cup_helix1  tabulated_mesh
(
 coor_sys       : ref(coor_sys_elem1),
 file_name       : cup_helix8.msh
```

```
)

feed_probe1  piecewise_straight_wire
(
  coor_sys        : ref(coor_sys_elem1),
  nodes           : sequence
    (
     struct(x:  2.385000E+00 cm, y:  0.000000E+00 cm, z:  0.000000E+00 cm),
     struct(x:  2.385000E+00 cm, y:  0.000000E+00 cm, z:  1.000000E-01 cm),
    ),
  radius          :  9.000000E-02 cm
)

helix_curve1  curved_wire
(
  coor_sys        : ref(coor_sys_elem1),
  nodes           : sequence
    (
     struct(x:  2.385000E+00 cm, y:  0.000000E+00 cm, z:  1.000000E-01 cm, radius:  9.000000E-02 cm),
.
.
.
     struct(x:  1.431000E+00 cm, y:  3.935006E-06 cm, z:  4.668523E+01 cm, radius:  9.000000E-02 cm),
    )
)

cup_helix2  tabulated_mesh
(
  coor_sys        : ref(coor_sys_elem2),
  file_name       : cup_helix8.msh
)

feed_probe2  piecewise_straight_wire
(
  coor_sys        : ref(coor_sys_elem2),
  nodes           : sequence
    (
     struct(x:  2.385000E+00 cm, y:  0.000000E+00 cm, z:  0.000000E+00 cm),
     struct(x:  2.385000E+00 cm, y:  0.000000E+00 cm, z:  1.000000E-01 cm),
    ),
  radius          :  9.000000E-02 cm
)

helix_curve2  curved_wire
(
  coor_sys        : ref(coor_sys_elem2),
  nodes           : sequence
```

```
   (
    struct(x:  2.385000E+00 cm, y:  0.000000E+00 cm, z:  1.000000E-01 cm, radius:  9.000000E-02 cm),
.
.
.
    struct(x:  1.431000E+00 cm, y:  3.935006E-06 cm, z:  4.668523E+01 cm, radius:  9.000000E-02 cm),
   )
)
```

The geometry is followed with the voltage generator objects. The first one is a uniform excitation of the array that has been normalized by TICARR. The first and last voltage generator elements shown have been moved to the actual position of the elements after being moved by individual element coordinate systems. This TOR file input is only in the base coordinates.

```
generator          voltage_generator
(
 generators      :sequence
(
   struct(x: -2.76150E+01 cm, y: -3.00000E+01 cm, z:  0.00000E+00 cm,
   amplitude:  2.00009E-01 V, phase:    0.000),
.
.
.
   struct(x:  3.23850E+01 cm, y:  3.00000E+01 cm, z:  0.00000E+00 cm,
   amplitude:  2.00009E-01 V, phase:    0.000),
  ),
 parameter_selection : s_parameters,
 parameter_file   : array_spar,
 coor_sys        : ref(array_coor_sys)
)
```

The second set of voltage generator specify only feeding the central element. The array file specified by the XADEF was modified by editing the central element to be 100 dB followed with TICARR generating the excitation file: helix25s.exi used in TICMARW. This will used when generating a UV-contour to determine scan blindness problems where currents generated by coupling to nearby elements are included in the scan element (active element) pattern.

```
generator_sing      voltage_generator
(
 generators      :sequence
(
   struct(x: -2.76150E+01 cm, y: -3.00000E+01 cm, z:  0.00000E+00 cm,
   amplitude:  1.00000E-05 V, phase:    0.000),
.
.
.
   struct(x:  2.38500E+00 cm, y:  0.00000E+00 cm, z:  0.00000E+00 cm, element 13
```

```
   amplitude:  1.00000E+00 V, phase:    0.000),
.
.
.
   struct(x:  3.23850E+01 cm, y:  3.00000E+01 cm, z:  0.00000E+00 cm,
   amplitude:  1.00000E-05 V, phase:    0.000),
  ),
 parameter_selection : s_parameters,
 parameter_file   : array_spar,
 coor_sys        : ref(array_coor_sys)
)
```

The rest of the TOR file contains essential elements to the GRASP project and must retained when adding the output to TICMARW.

```
wavelength  frequency_range
 (
 frequency_range  : struct(start_frequency: 2.0 GHz, end_frequency: 2.0 GHz, number_of_frequencies: 1)
)

mom  mom
(
 frequency       : ref(wavelength),
//  scatterer       : ref(elem1),
 scatterer       : ref(helix_elem),
 relative_geom_tolerance : 0.1E-03,
 iterative_solution : struct(use_mlfmm: automatic, relative_error: 0.1E-02, mlfmm_use_disk: allow, mlfmm_precision: normal, preconditioner_accuracy: normal, obsolete_group_size: 4.0),
 keep_matrix      : on,
 polynomial_precision : 1,
 integration_precision : 2,
 advanced_polynomial : struct(edge: 1, wedge: 0, junction: 0, pec: 0, dielectric: 0, wire: 0),
 file_name        : array.cur,
 colour_plot_file : array.cpf
)

array_pattern  spherical_cut
(
 coor_sys        : ref(array_coor_sys),
 theta_range      : struct(start: -180.0, end: 180.0, np: 361),
 phi_range        : struct(start: 0.0, end: 90.0, np: 3),
 polarisation     : circular,
 file_name        : array01.cut,
 frequency        : ref(wavelength)
)
```

```
spherical_grid_01  spherical_grid
(
  coor_sys        : ref(array_coor_sys),
  x_range         : struct(start: -0.7, end: 0.7, np: 141),
  y_range         : struct(start: -0.7, end: 0.7, np: 141),
  polarisation    : circular,
  file_name       : pattern.grd,
  frequency       : ref(wavelength)
)

//DO NOT MODIFY OBJECTS BELOW THIS LINE.
//THESE OBJECTS ARE CREATED AND MANAGED BY THE
//GRAPHICAL USER INTERFACE AND SHOULD NOT BE
//MODIFIED MANUALLY!
view_1  view
(
  objects         :
sequence(ref(view_1_mom_plot),ref(view_1_mom_source_plot),ref(view_1_output_points_plot),
ref(view_1_coor_sys_plot),ref(view_1_tabulated_mesh_plot),ref(view_1_wires_plot),ref(view_1_box_pl
ot))
)

view_1_mom_plot  mom_plot
(
)

view_1_mom_source_plot  mom_source_plot
(
)

view_1_output_points_plot  output_points_plot
(
)

view_1_coor_sys_plot  coor_sys_plot
(
)

view_1_tabulated_mesh_plot  tabulated_mesh_plot
(
)

view_1_wires_plot  wires_plot
(
)

view_1_box_plot  box_plot
```

```
(
)

 //$$ Saved at 14:15:49 on 06.12.2017 by GRASP ver. 10.6.0 SN=005400
```

This GRASP project contains commands to generate the pattern of the uniformly feed array and another to generate a UV-contour when only the central element is fed. The TCI file contains these:

```
COMMAND OBJECT mom get_currents ( source : sequence(ref(generator)))
#COMMAND OBJECT D3_PLOT get_all_ogl_plot () Cmd_1
#
COMMAND OBJECT array_pattern get_field ( source : sequence(ref(mom)))
COMMAND OBJECT mom get_currents ( source : sequence(ref(generator_sing)))
COMMAND OBJECT spherical_grid_01 get_field ( source : sequence(ref(mom)))
QUIT
```

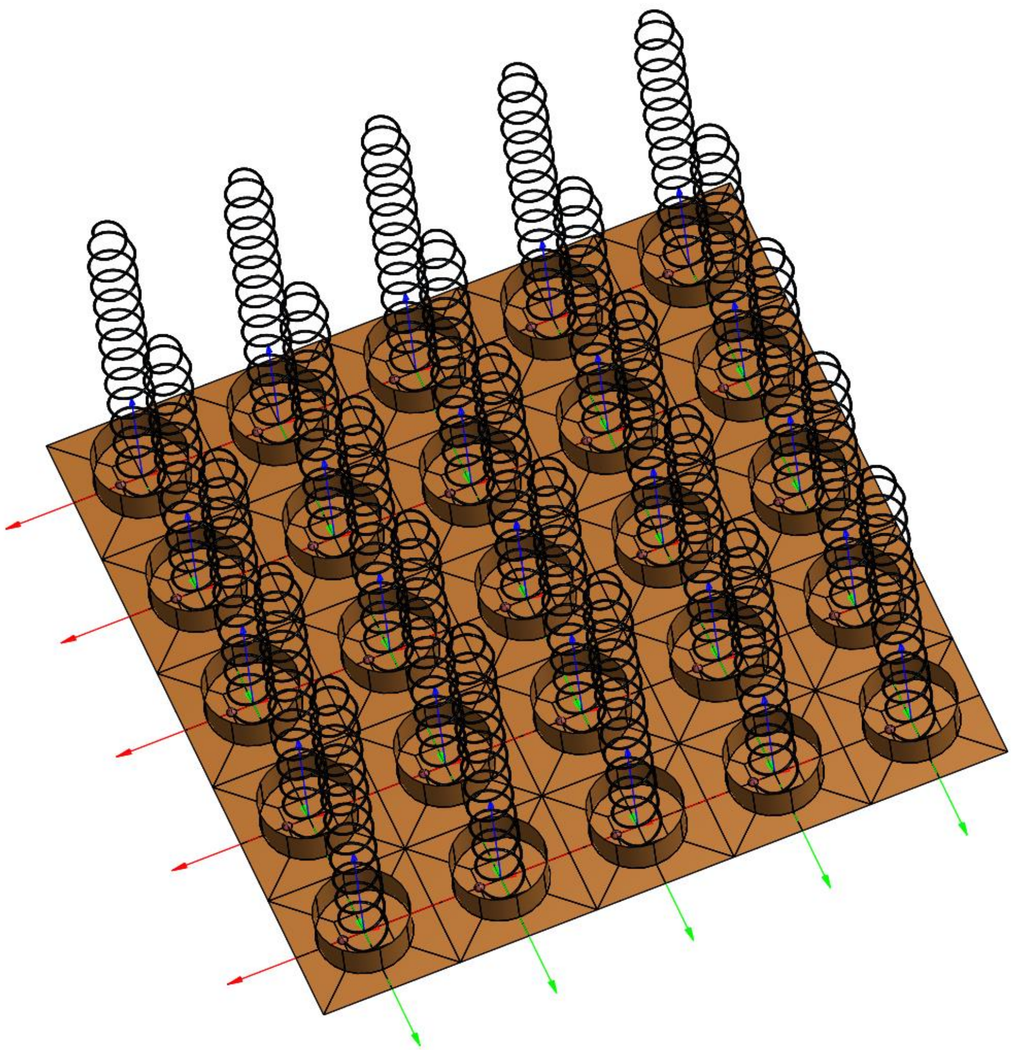


Figure 49 25-element array of 13 turn helix antenna in cup generated by GRASP from the TOR file.

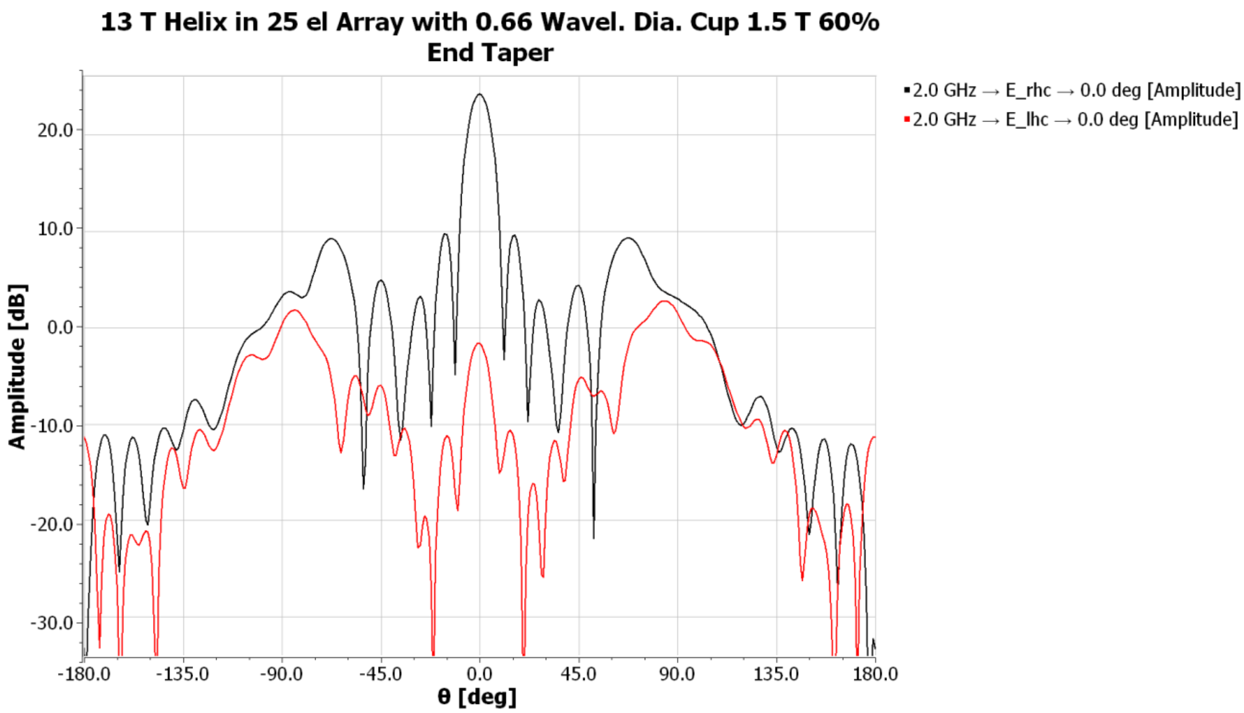


Figure 50 25-element array of 13 turn helix antenna in cup GRASP MoM pattern

The pattern grating lobes near 70° because elements are spaced 1λ. The lobes are reduced first order by the element pattern (Figure 43). However, the close coupling of the elements alters the effective element pattern (scan or active element pattern) and complicates the effect.

Figure 50 has a gain of 24.21 dB. A single element outside the array has a gain of 12.88 dB. The simple analysis of element-gain times the number of elements predicts a gain of 26.86 dB. The gain of single helix greatly exceeds the allotted area in the array and causes 2.65 dB loss. Losses like this are predicted in Section 4-28. When the beamwidth of a single helix computed by MoM is used with the coupling model of Section 4-28, the predicted loss is 2.26 dB (24.60 dB array gain). Figure 51 of the PO summation GRASP analysis has lower grating lobes in both the co- and cross-polarization patterns than the MoM analysis and accounts for the 0.39 dB gain difference between the two methods. MoM accounts for the change in current in the helical wires due to direct coupling between them. A comparison between the radiation patterns of figures 50 and 51 shows that the effect is small. The thermodynamic analysis of section 4-28 predicts most of the gain reduction due to overlapping equivalent area of the elements. The changes of the current distribution due to coupling is small and shows why a physical optics solution predicts the proper pattern shape even in this pathological case of trying to place high gain array elements too close together.

Coupling between elements has little effect on element input impedance. The single helix has an impedance of 54.1 + j13.2. The central element has an input impedance of 52.4 + j13.2 when all elements are fed equally and an impedance 52.1 + j13.1 when only the central element is fed. Noted in Section 4-28 all other elements in the array appear the same to particular element whether fed or not provided they have a load on their inputs. They are only loads to the central element. What applies to the central element applies to all elements. The only difference is whether an element is close to the edge of the array and surrounded by fewer elements.

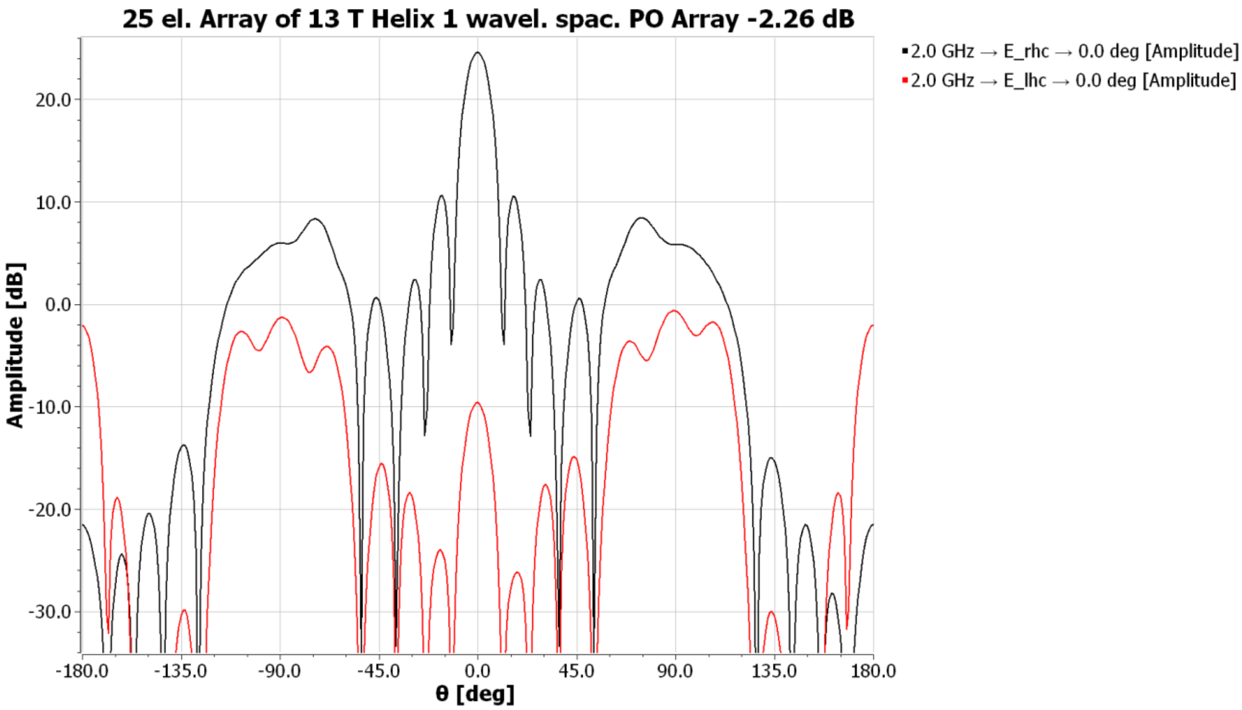


Figure 51 25-element array of 13 turn helix antenna in cup GRASP PO array pattern

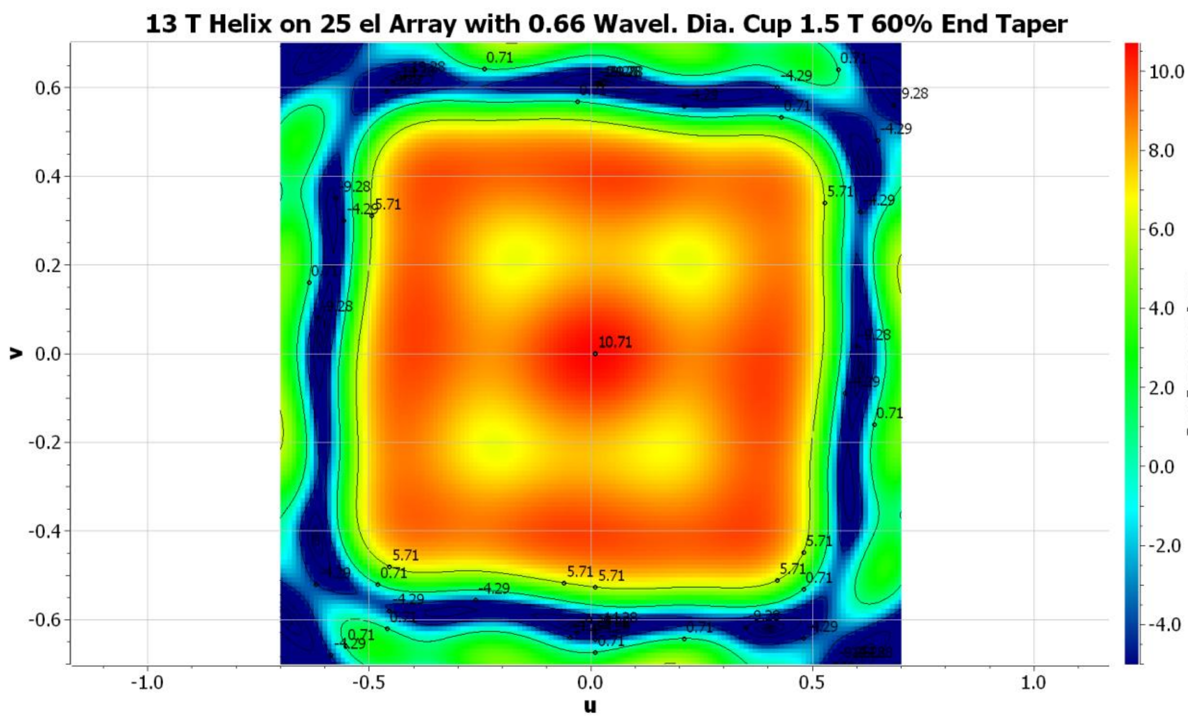


Figure 52 UV-pattern contour with only the central element fed (scan- or active-element pattern) of the 25-element helix element array

Scan blindness can be determined by looking at the pattern of a central element in a small array, a measurement method to check for the problem. A larger array magnifies the effect seen in the small array contour. Figure 52 gives the predicted pattern using GRASP MoM. The pattern dip at u = 0.2 and v = 0.2 shows that the array will have scan blindness near this scan direction. Of course, the axial helix antenna has too much gain to useful in this array of 1λ spaced elements. Scanning the array to a direction of scan blindness usually causes a large power reflection into the feed network which may not be built to handle the surge of power. It can cause severe overheating. Figure 52 predicts scan blindness at about 18.1° when scanned in the diagonal plane.

Hexagonal Array

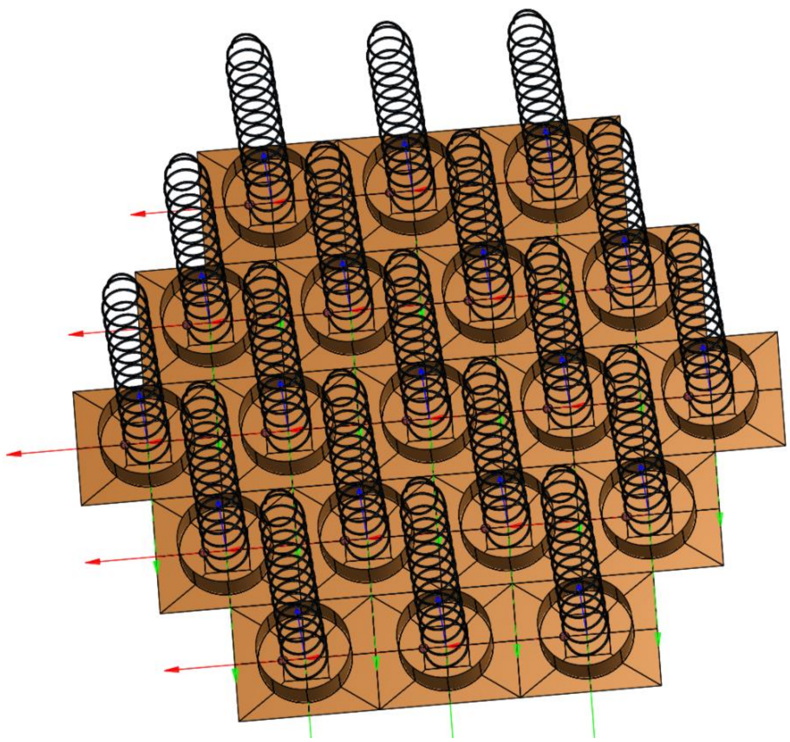


Figure 53 19-element Hexagonal array of 13 turn helix antenna in cup generated by GRASP from the TOR file.

Figure 53 illustrates a 19-element array with 1λ element spacing. XADEF is used to design a hexagonal array and TICARR to generate the *.isp and *.exi files. While the x-axis width remains 15 cm, the y-axis width becomes 15 sin(60°) cm. Unlike the rectangular array with 1λ element spacing which has a grating lobe at 90° with no scan, the hexagonal array with 1λ does not have grating lobes without beam scan. When the hexagonal array is scanned to 10.58° along the x-axis direction, two grating lobes enter visible space along ϕ = ± 145°. Figure 54 shows the pattern response for the un-scanned phased array which contains no grating lobes. Compared to element-gain times (number of elements), the array has 2.28 dB loss. The center element is fed with all others loaded to generate the UV-pattern, Figure 55. The nearly circular pattern dip, starting at u (or v) = 0.22 (12.7°), says that scan blindness occurs over a range of θ and any ϕ scan plane.

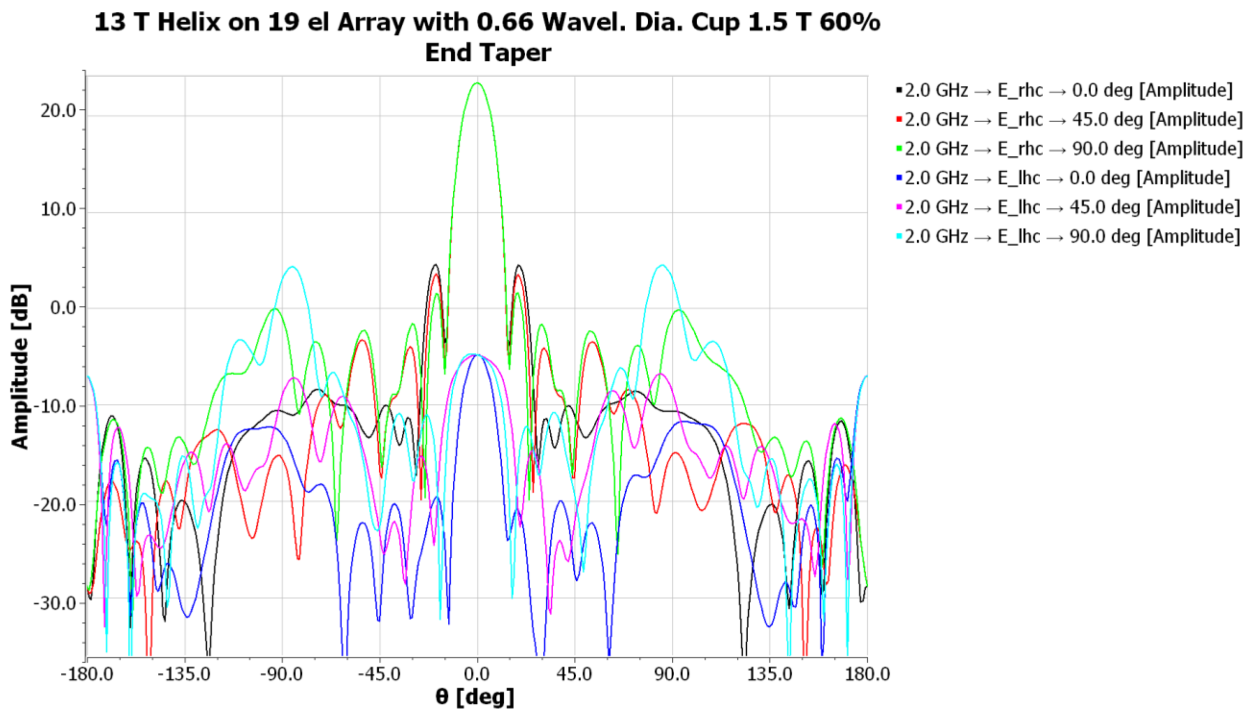


Figure 54 19-element Hexagonal array of 13 turn helix antenna in cup GRASP MoM pattern

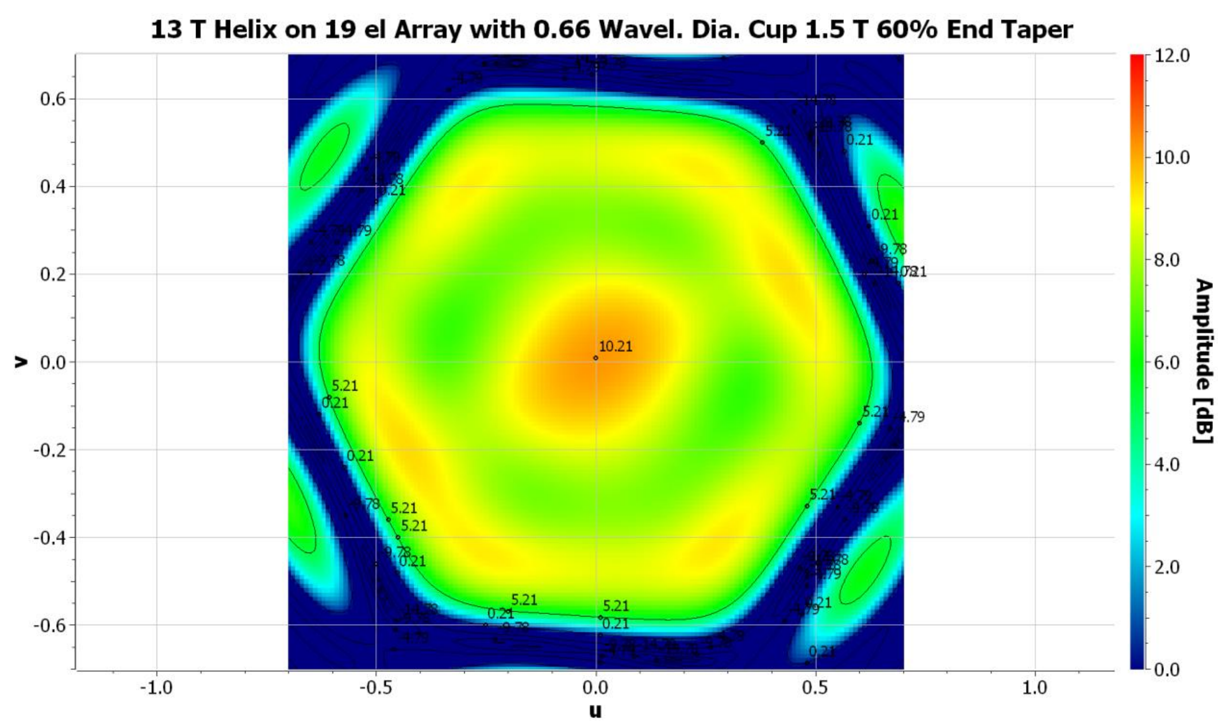


Figure 55 UV-pattern contour with only the central element fed (scan- or active-element pattern) of the 19-element helix element hexagonal array

6 Turn Helix 25-Element Array

Reducing the gain of the array elements lowers arraying loss for elements spaced 1λ. Figure 15 shows construction of 6 turn helix including a cup shield in the ground plane. The few turns generate a wider beamwidth response, figure 16 analyzed by using GRASP MoM.

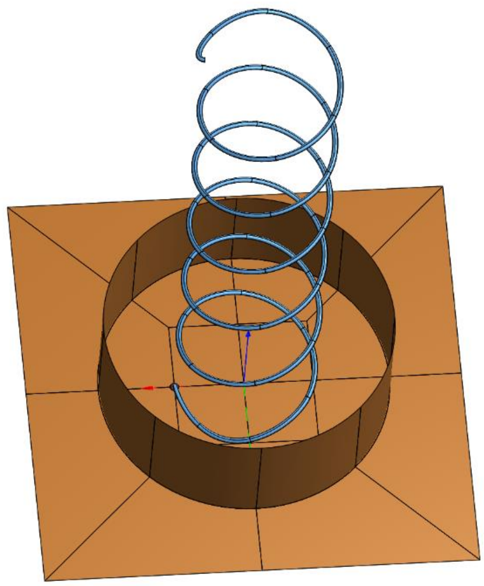


Figure 56 6 Turn Un-tapered Helix located on a Ground Plane with Cup Shield

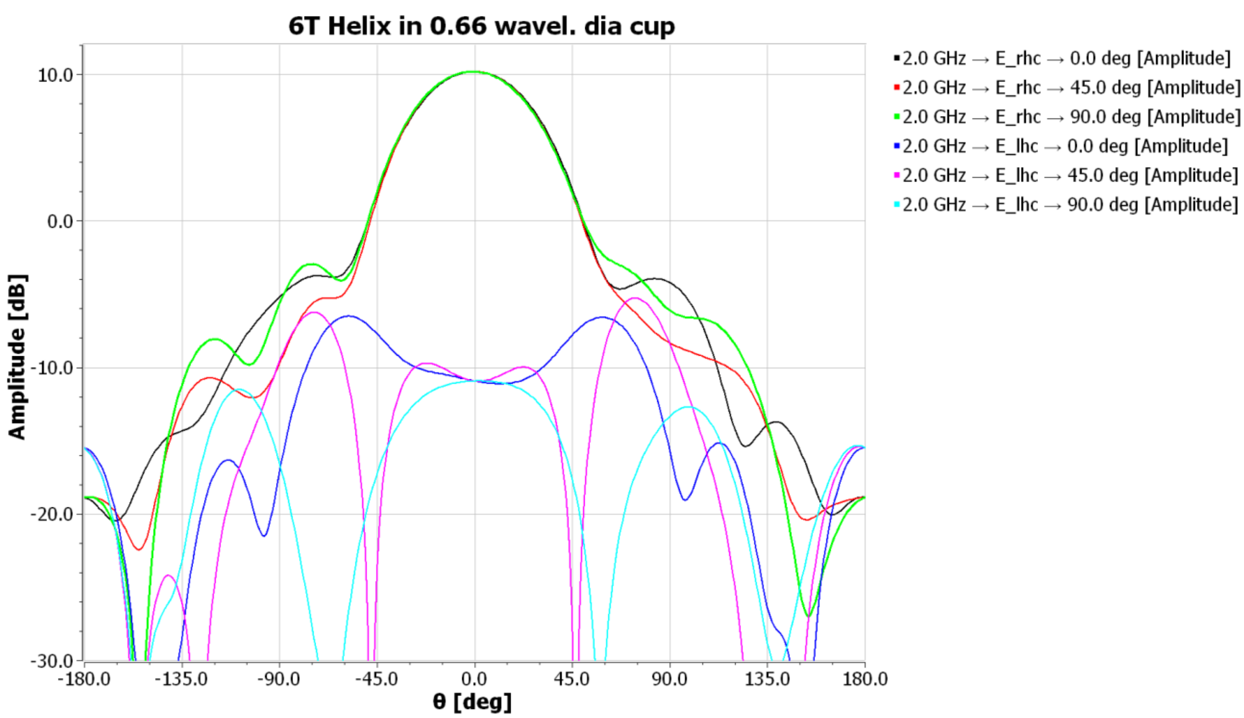


Figure 57 GRASP MoM pattern analysis for 6 turn helix in cup shielded ground plane

When arrayed in a 25-element antenna, GRASP MoM predicts the pattern, Figure 58.

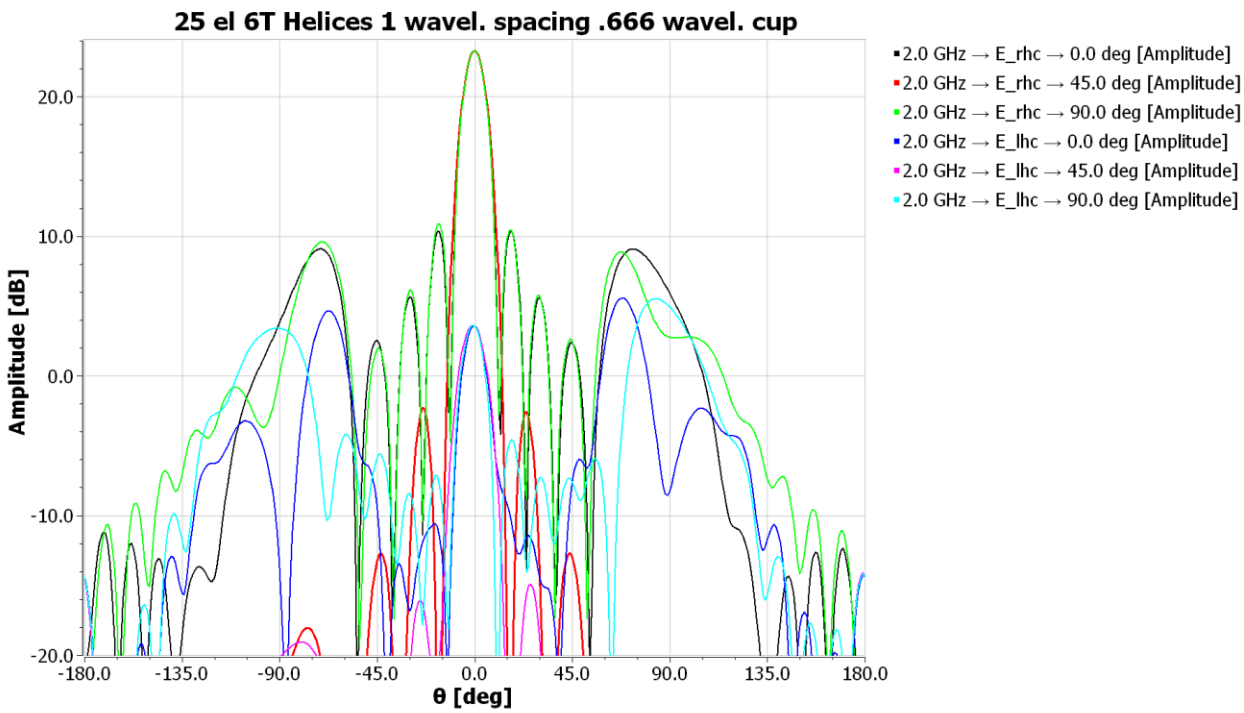


Figure 58 25-element 6 Turn Helix antennas in 2/3λ diameter cup spaced 1λ using GRASP MoM

The difference between the peak of Figure 58 and 25 times the peak of Figure 57 of the single element is 1 dB loss. The shorter, low gain, element gain is closer to matching the gain the equivalent area allotted to each element. We still see the grating lobe response near 70° on figure 58. By using the element beamwidth and the simple antenna model as in Section 4-28, we calculate 0.76 dB loss. If the GRASP MOM analysis generates a cut file output that covers the radiation sphere, the program CUTMUT (section 4-28) can be used to compute mutual resistance between elements versus spacing. The program GPDIRU uses the array geometry (XADEF file) and predicts a loss of 0.75 dB nearly the same as the simple beamwidth model. In this case the coupling between the fewer turns produces less current distortions shown in the 0.24 dB difference between the two methods. The PO GRASP model matches better with these lower gain elements when there is a shorter region of coupling between elements.

Figure 59 plots the scan-element pattern of the 6-turn helix antenna placed in 25-element array with 1λ spacings. This shows that scan blindness maybe a problem even without scanning as the array size grows. Of course, the pattern loss due to arraying is smaller and will have less effect and grow slower.

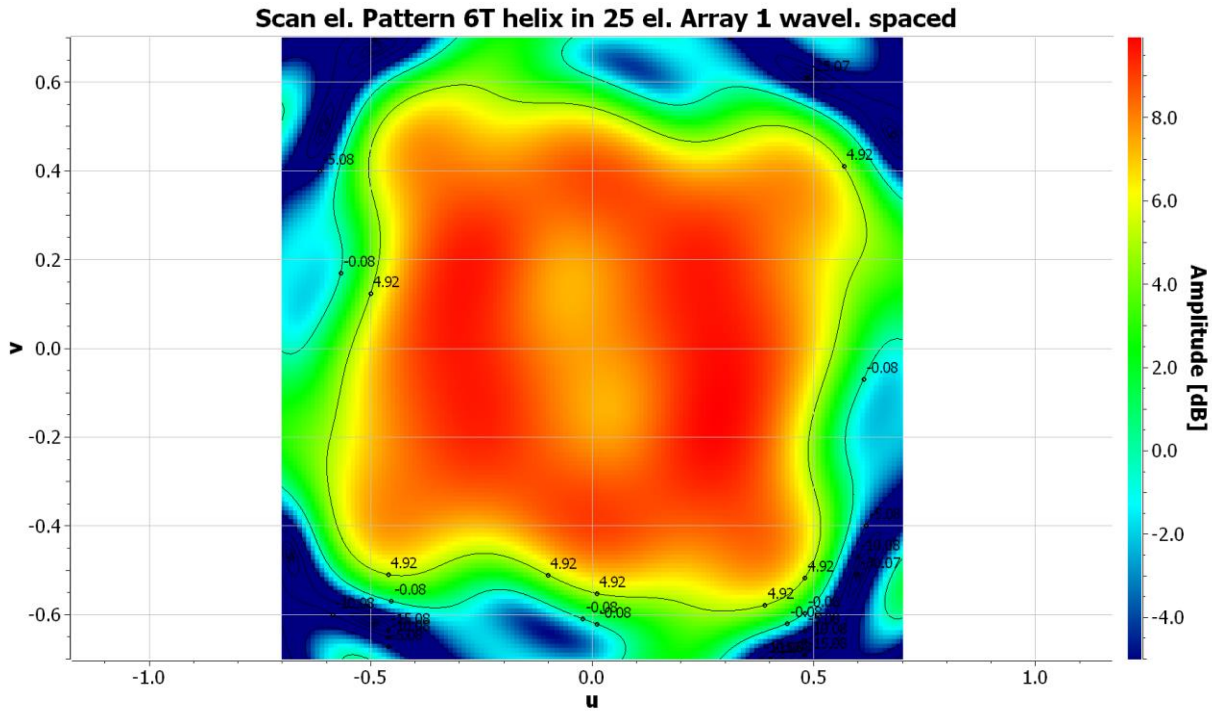


Figure 59 Scan-element pattern of 6 Turn Helix in 25 element array

Conformal Array of Helix elements on a Cylinder

The program XADEF is able to roll an array around a cylinder. The distance between elements is along the arc length of the cylinder. This causes a geometric problem because we use flat ground plane elements that produces ground gaps if we do not account for this effect. If done incorrectly, it can also cause overlap or intersection of the pieces of ground plane in a non-physical manner. Figure 22 shows half of two plates joined outside the rolling cylinder where the distance between elements is 2*arclength.

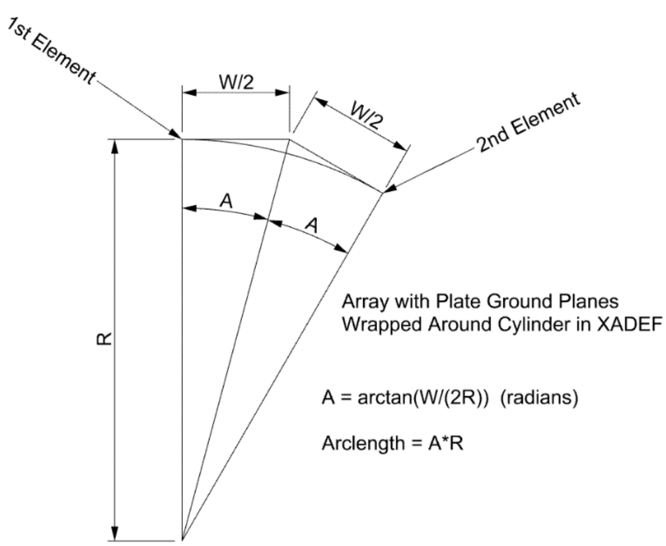


Figure 22 Geometry of construction of Flat Plates rolled in a Cylinder

Consider an array of elements with 15 cm ground planes rolled on a cylinder of radius = 70 cm. The distance between elements needs to be 2*70*arctan(15/140) = 14.94299 cm in XADEF in the roll direction. We use XADEF to generate the array geometry using the following steps:

```
Command Prompt - xadef
C:\arrays>xadef
Enter input 0 keyboard, 1 file 0
Enter File Name cyl_helix_25.arr
Enter units: 1 in., 2 ft, 3 cm, 4 m 3
New File? y
Enter Frequency (GHz) 2
xadef: ad,re,25
Enter Ampl (dB), Phase 0,0
Enter Number of Elements along X-axis 5
Enter initial spacings in X,Y axes cm  14.94299,15
Array along X-axis
Enter Axis Spacing: 1 Uniform, 2 Bratkovic  3 Geometric, 4 Uneven Taylor 1
Enter 1) Uniform Amplitude Distribution
      2) Area Sampling of Taylor Distribution
      3) Point Sampling of Taylor Distribution
      4) Zero Sampled Taylor Distribution
      5) Point Sampled Bayliss Distribution
      6) Zero Sampled Bayliss Distribution
      7) Chebyshev array 1
Enter Quadratic Phase Factor, S 0
Array along Y-axis
Enter Axis Spacing: 1 Uniform, 2 Bratkovic  3 Geometric, 4 Uneven Taylor 1
Enter 1) Uniform Amplitude Distribution
      2) Area Sampling of Taylor Distribution
      3) Point Sampling of Taylor Distribution
      4) Zero Sampled Taylor Distribution
      5) Point Sampled Bayliss Distribution
      6) Zero Sampled Bayliss Distribution
      7) Chebyshev array 1
```

```
Enter Quadratic Phase Factor, S 0
move: pr,co
Projection of Array from X-Y plane to Cone
Is the array orientated so that the X-axis will be around
the cone and Y axis along the cone axis? y
Enter Radius of Cone at Center of Array  cm  70
Enter Cone Angle (0 for cylinder) 0
Enter Rotation from X-axis of Center of array on Cone 0
Enter Orientation of Cone Axis: Theta, Phi 0,0
move: ro,ax
Array must orientated in X-Y plane before it is Rotated, ready? y
Enter Lower End Point (X,Y,Z) of Axis of Rotation 0,0,0
Enter Upper End Point (X,Y,Z) of Axis of Rotation 0,1,0
Enter Rotation Angle about Axis -90
Rotate element pointing? y
move: tr,ge
Enter Translation X,Y,Z  cm  0,0,-70
move: ex
Enter Final New Z axis Rotation of Antennas (array) 0
xadef:
```

```
xadef: li,fi

File:cyl_helix_25.arr

                  Location                          Element
No        X          Y          Z      Ampl(dB)   Phase      Euler Angles
   1     30.000    -28.986     -6.283     0.00     0.00   -90.00    24.46    90.00
   2     30.000    -14.830     -1.589     0.00     0.00   -90.00    12.23    90.00
   3     30.000      0.000      0.000     0.00     0.00     0.00     0.00     0.00
   4     30.000     14.830     -1.589     0.00     0.00    90.00    12.23   -90.00
   5     30.000     28.986     -6.283     0.00     0.00    90.00    24.46   -90.00
   6     15.000    -28.986     -6.283     0.00     0.00   -90.00    24.46    90.00
   7     15.000    -14.830     -1.589     0.00     0.00   -90.00    12.23    90.00
   8     15.000      0.000      0.000     0.00     0.00     0.00     0.00     0.00
   9     15.000     14.830     -1.589     0.00     0.00    90.00    12.23   -90.00
  10     15.000     28.986     -6.283     0.00     0.00    90.00    24.46   -90.00
  11      0.000    -28.986     -6.283     0.00     0.00   -90.00    24.46    90.00
  12      0.000    -14.830     -1.589     0.00     0.00   -90.00    12.23    90.00
  13      0.000      0.000      0.000     0.00     0.00     0.00     0.00     0.00
  14      0.000     14.830     -1.589     0.00     0.00    90.00    12.23   -90.00
  15      0.000     28.986     -6.283     0.00     0.00    90.00    24.46   -90.00
  16    -15.000    -28.986     -6.283     0.00     0.00   -90.00    24.46    90.00
  17    -15.000    -14.830     -1.589     0.00     0.00   -90.00    12.23    90.00
  18    -15.000      0.000      0.000     0.00     0.00     0.00     0.00     0.00
  19    -15.000     14.830     -1.589     0.00     0.00    90.00    12.23   -90.00
  20    -15.000     28.986     -6.283     0.00     0.00    90.00    24.46   -90.00
  21    -30.000    -28.986     -6.283     0.00     0.00   -90.00    24.46    90.00
  22    -30.000    -14.830     -1.589     0.00     0.00   -90.00    12.23    90.00
  23    -30.000      0.000      0.000     0.00     0.00     0.00     0.00     0.00
  24    -30.000     14.830     -1.589     0.00     0.00    90.00    12.23   -90.00
  25    -30.000     28.986     -6.283     0.00     0.00    90.00    24.46   -90.00
xadef: _
```

The array is orientated on a cylinder with its axis along the X-axis. The final translation of the move command places the central element (13) at the origin and pointed along z-axis.

We use TICARR to generate GRASP input files.

```
Enter input 0 keyboard, 1 file 0
Enter XADEF array File Name cyl_helix_25.arr
Number of Elements:      25
Frequency:              2.000GHz
Enter Ticra element position output filename (.isp) helix25c.isp
Enter units: 1 in, 2 ft, 3 mm, 4 cm, 5 m 4
Enter label
25-element 13T helix array on 70 cm radius cylinder
Enter TICRA source object name for element pattern helix_elem
Enter Ticra excitation output filename (.exi) helix25c.exi
Enter quadratic phase taper (deg) X,Y, Max Radius X,Y 0,0,10,10
Enter array efficiency dB 0
Enter 1 to form beam of array 1
Enter scan direction of array Theta, Phi 0,0
Enter beam pointing frequency (GHz) 2
Quantize Phase? n
Enter Feed Error (1 Sigma) Ampl(dB), Phase(deg) 0,0
Another beam excitation file? n
Process another array file? n
Stop - Program terminated.
```

To generate the scan-element pattern of the central element in the array on the cylinder, we use XADEF to edit the amplitude of element 13 (at origin).

```
C:\arrays>xadef
Enter input 0 keyboard, 1 file 0
Enter File Name cyl_helix_25.arr
Enter units: 1 in., 2 ft, 3 cm, 4 m 3
New File? n
Number of Elements:     25
xadef: ed,am,13
Element No:     13
          Location: X =      0.000   Y =      0.000   Z =      0.000
          Ampl(dB) =    0.00  Phase =    0.00
Euler Angles =      0.00      0.00      0.00
Enter New Amplitude (dB) 100
xadef: ex
Another file? n
Stop - Program terminated.
```

We re-run TICARR to generate the GRASP excitation file of the single element in the loaded array.

Run GRASP starting with the MoM solution of a single helix element and generate a new project for the 25-element array. The input to TICMARW for a 25-element array is modified to replace the GRASP array geometry file (*.ISP) with the new array and run the program to generate the TOR file edit file. Open the output of TICMARW in a text editor and copy the entire file. Open the TOR file in the working directory which will contain the GRASP objects for a single helix element. Replace the upper portion with the output from TICMARW as described above and save the file. Re-open the GRASP project and it will now contain an array (Figure 60).

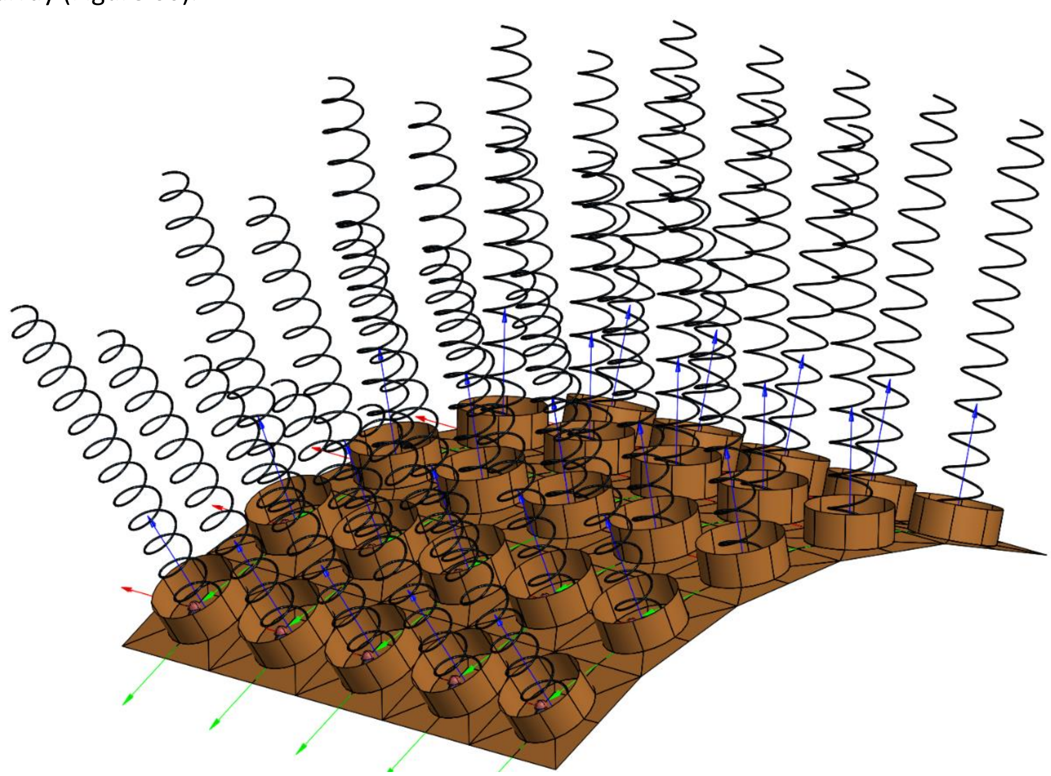


Figure 60 25-element 13 T Helix Array on Cylinder with axis along x-axis GRASP MoM Model

Figure 61 of a close-up of the corner illustrates that application of figure 19 to the element separation before the array is rolled over the cylinder leads to a model without gaps between flat plate ground planes.

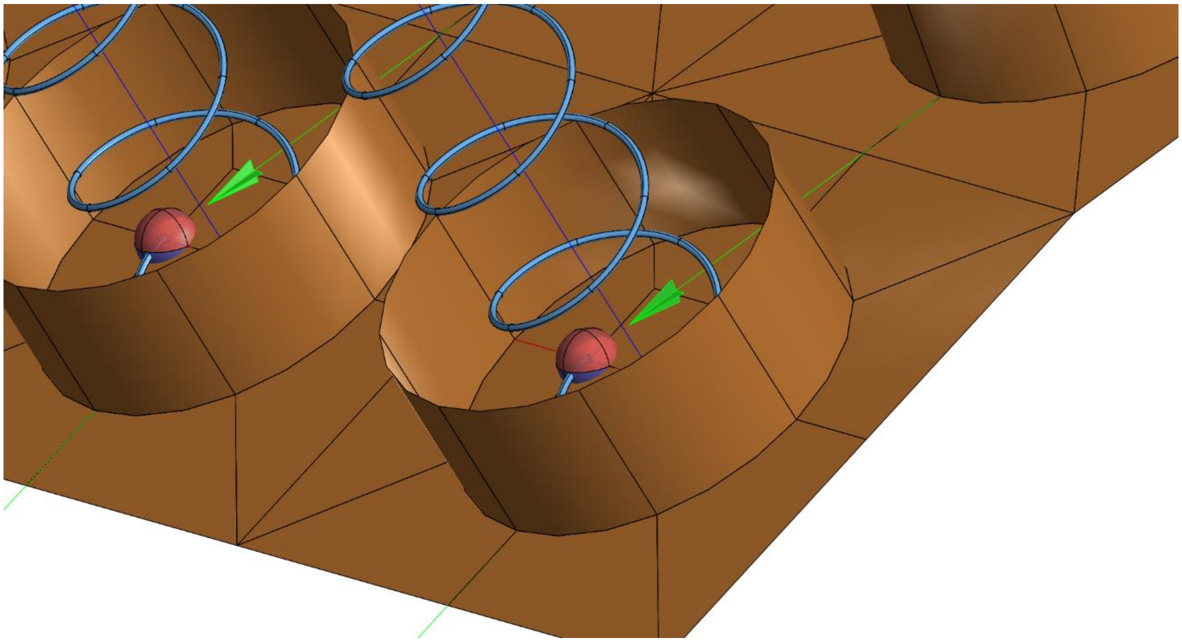


Figure 61 Corner of 25-element 13 T Helix Array on Cylinder with axis along x-axis GRASP MoM Model

The actual array is not practical because its elements are too long and it produces grating lobes in the roll plane (ϕ = 90°), figure 62, but illustrates a method of generating a conformal array in GRASP MoM.

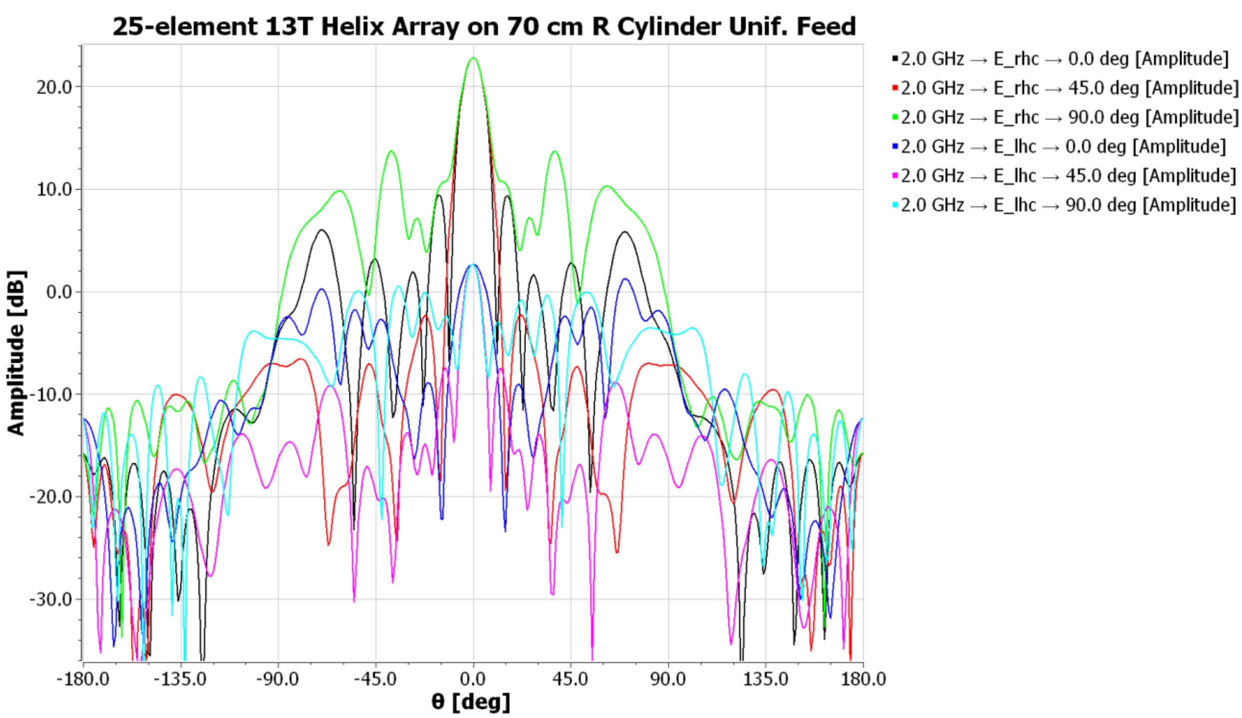


Figure 62 Patterns: 25-element 13 T Helix Array on Cylinder with axis along x-axis GRASP MoM Model

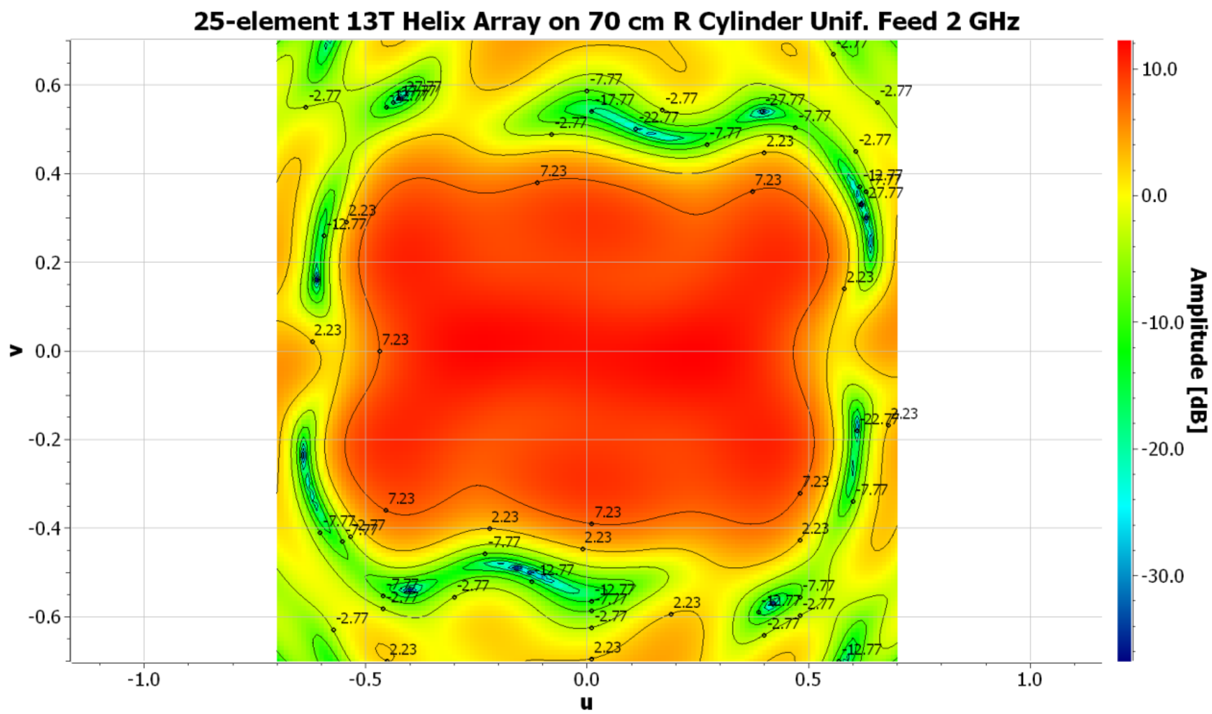
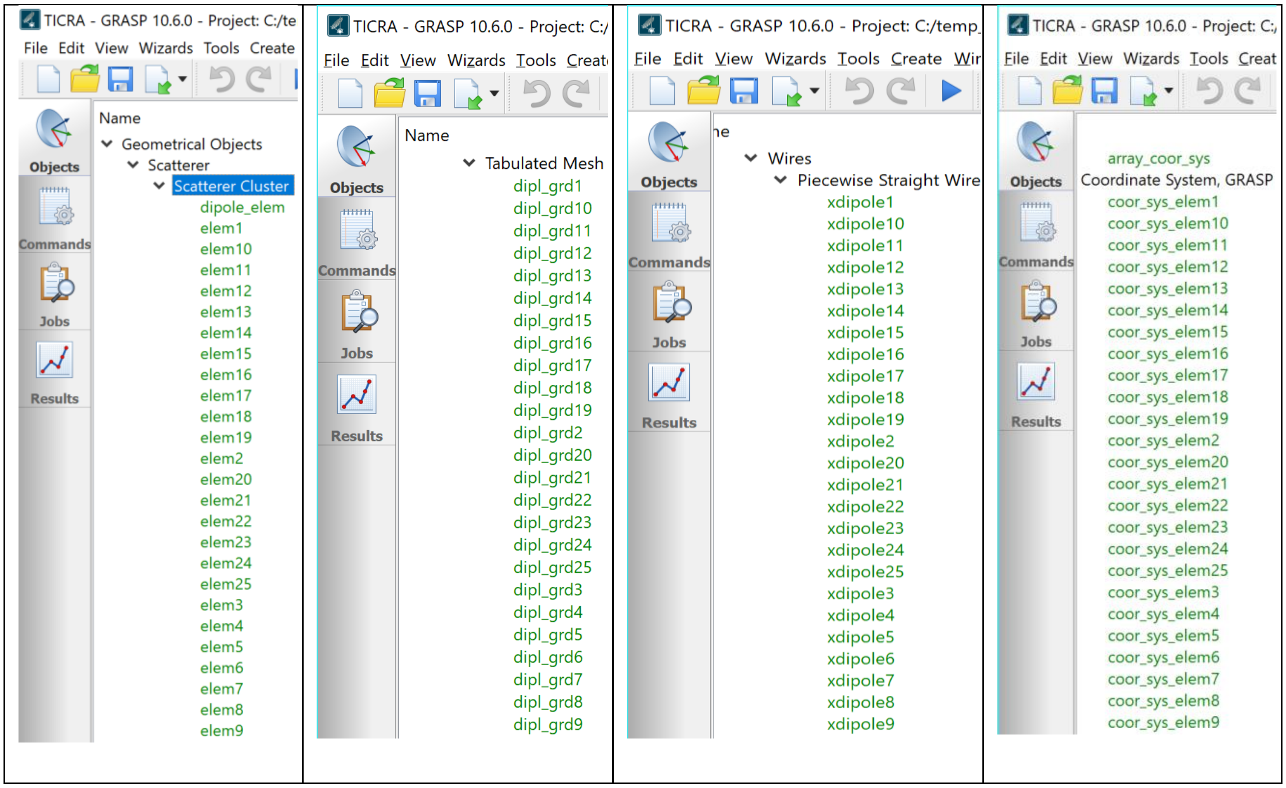


Figure 63 Central element UV-pattern with other elements loaded in 25-element 13 T Helix Array on Cylinder with axis along x-axis GRASP MoM Model

Alternative Approaches

Each antenna element presented above has a special program to generate the geometry of a single element which may not be practical for the arbitrary antenna element. GRASP MoM accepts CAD input where the single element and its ground plane can be designed using a CAD program and imported into GRASP in place of the mesh file shown in the examples above. Because the elements are duplicated in the array, GRASP will run faster if the CAD input of a single element is converted by GRASP into a mesh file. GRASP will not have to convert every CAD element of the array into a mesh file for each element and for each analysis.

It is not practical to use the GRASP GUI to generate all the objects required for an array analysis and I recommend writing a procedure which writes edits to the TOR to generate the many objects and gather them into scatterer clusters. For an example, consider the 25-element dipole array of Figure 4. The program TICMARW generates a scatter cluster for each element, a tabulated mesh object of each ground plane, a piecewise-straight-wire object of each dipole, and a separate coordinate system for each element: 100 GRASP objects for a 25-element array.



When we look at the TOR file, we can identify various elements of the GRASP objects.

Scatterer clusters:

```
dipole_elem  scatterer_cluster
(
  scatterers      : sequence(ref(elem1),ref(elem2),ref(elem3),ref(elem4),ref(elem5),ref(elem6),ref(elem7),
ref(elem8),ref(elem9),ref(elem10),ref(elem11),ref(elem12),ref(elem13),ref(elem14),ref(elem15),
ref(elem16),ref(elem17),ref(elem18),ref(elem19),ref(elem20),ref(elem21),ref(elem22),ref(elem23),
ref(elem24),ref(elem25))
)

elem1  scatterer_cluster
(
  scatterers      : sequence(ref(dipl_grd1),ref(xdipole1))
)

elem2  scatterer_cluster
(
  scatterers      : sequence(ref(dipl_grd2),ref(xdipole2))
)
 …
elem25  scatterer_cluster
(
  scatterers      : sequence(ref(dipl_grd25),ref(xdipole25))
)
```

Tabulated Mesh objects:

```
dipl_grd1  tabulated_mesh
(
  coor_sys        : ref(coor_sys_elem1),
  file_name       : dipgrd1.msh
)

dipl_grd2  tabulated_mesh
(
  coor_sys        : ref(coor_sys_elem2),
  file_name       : dipgrd1.msh
)
…
dipl_grd25  tabulated_mesh
(
  coor_sys        : ref(coor_sys_elem25),
  file_name       : dipgrd1.msh
)
```

Piecewise_straight_wire (Dipoles):

```
xdipole1  piecewise_straight_wire
(
  coor_sys        : ref(coor_sys_elem1),
  nodes          : sequence
    (    struct(x: -6.69531 cm, y: 0.0 cm, z: 5.563107 cm),
    struct(x: 0.0 cm, y: 0.0 cm, z: 8.0 cm),
    struct(x: 6.69531 cm, y: 0.0 cm, z: 5.563107 cm)
    ),
  radius         : 0.125 cm
)

xdipole2  piecewise_straight_wire
(
  coor_sys        : ref(coor_sys_elem2),
  nodes          : sequence
    (    struct(x: -6.69531 cm, y: 0.0 cm, z: 5.563107 cm),
    struct(x: 0.0 cm, y: 0.0 cm, z: 8.0 cm),
    struct(x: 6.69531 cm, y: 0.0 cm, z: 5.563107 cm)
    ),
  radius         : 0.125 cm
)
…
xdipole25  piecewise_straight_wire
(
  coor_sys        : ref(coor_sys_elem25),
  nodes          : sequence
    (    struct(x: -6.69531 cm, y: 0.0 cm, z: 5.563107 cm),
```

```
  struct(x: 0.0 cm, y: 0.0 cm, z: 8.0 cm),
  struct(x: 6.69531 cm, y: 0.0 cm, z: 5.563107 cm)
  ),
 radius        : 0.125 cm
)
```

Element Coordinate Systems:

```
coor_sys_elem1  coor_sys_grasp_angles
(
 origin        : struct(x: -30.0 cm, y: -30.0 cm, z: 0.0 cm),
 base          : ref(array_coor_sys)
)

coor_sys_elem2  coor_sys_grasp_angles
(
 origin        : struct(x: -15.0 cm, y: -30.0 cm, z: 0.0 cm),
 base          : ref(array_coor_sys)
)
…
coor_sys_elem13  coor_sys_grasp_angles  (center element)
(
 origin        : struct(x: 0.0 cm, y: 0.0 cm, z: 0.0 cm),
 base          : ref(array_coor_sys)
)
…
coor_sys_elem25  coor_sys_grasp_angles
(
 origin        : struct(x: 30.0 cm, y: 30.0 cm, z: 0.0 cm),
 base          : ref(array_coor_sys)
)
```

Each array excitation has a voltage generator: (normalized uniform amplitude and phase)
This must be generated from the EXI excitation file and ISP geometry file relative to the feed point on each element

```
generator  voltage_generator
(
 generators      : sequence
  (   struct(x: -30.0 cm, y: -30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: -15.0 cm, y: -30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 0.0 cm, y: -30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 15.0 cm, y: -30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 30.0 cm, y: -30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: -30.0 cm, y: -15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: -15.0 cm, y: -15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 0.0 cm, y: -15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 15.0 cm, y: -15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 30.0 cm, y: -15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
```

```
  struct(x: -30.0 cm, y: 0.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: -15.0 cm, y: 0.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 0.0 cm, y: 0.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 15.0 cm, y: 0.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 30.0 cm, y: 0.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: -30.0 cm, y: 15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: -15.0 cm, y: 15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 0.0 cm, y: 15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 15.0 cm, y: 15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 30.0 cm, y: 15.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: -30.0 cm, y: 30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: -15.0 cm, y: 30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 0.0 cm, y: 30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 15.0 cm, y: 30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0),
  struct(x: 30.0 cm, y: 30.0 cm, z: 8.0 cm, amplitude: 0.200009 V, phase: 0.0)
  ),
 parameter_selection : s_parameters,
 parameter_file   : array_spar,
 coor_sys        : ref(array_coor_sys)
)
```

The MoM analysis is performed on the scatterer cluster of all elements of the array.