User Manual for EXCALIBUR: A FE-BI Numerical Laboratory for Cavity-Backed Antennas in a Circular Cylinder Version 1.2

Leo C. Kempel
Radiation Laboratory
University of Michigan
1301 Beal Ave.
Ann Arbor, MI 48109-2122

December 16, 1993

1 Introduction

The Finite Element-Boundary Integral (FE-BI) technique has been used to analyze the scattering and radiation properties of cavity-backed patch antennas recessed in a metallic groundplane. A program, CAVITY3D, was written and found to yield accurate results for large arrays without the usual high memory and computational demand associated with competing formulations. Recently, the FE-BI approach was extended to cavity-backed antennas recessed in an infinite, metallic circular cylinder. EXCALIBUR is a computer program written in the Radiation Laboratory of the University of Michigan which implements this formulation.

This user manual will give a brief introduction to EXCALIBUR and some hints as to its proper use. As with all computational electromagnetics programs (especially finite element programs), skilled use and best performance is only obtained through experience. However, we will comment on several important aspects of the program such as portability, geometry generation, interpretation of results and custom modification.

2 Formulation

EXCALIBUR implements the FE-BI formulation for cavity-backed antennas recessed in metallic circular cylinders. The formulation imposes some restrictions on the geometries which will be discussed. Principally, use of the BiConjugate Gradient-Fast Fourier Transform (BiCG-FFT) iterative solver requires uniform zoning on the aperture of the cavity. Thus, each surface patch has cylindrical-rectangular shape. The basis functions, dyadic Green's function and associated field formulas are given in a paper concerning scattering by these structures [1]. Modifications of this formulation for antenna analysis was given in another paper [2].

The uniform zoning requirement causes some difficulty in modeling; however, with some practice, these difficulties may be overcome. For example, the specification of the patch and cavity size must both be expressed by an integer number of edges (hence nodes). Thus, if the cavity is twice the size of the patch, one has no problem specifying the patch and the cavity with the same uniform grid. However, if the ratio of the patch and cavity sizes are not integers, discretization may not be possible. This is often the case with a continuous wraparound cavity. Such a cavity is shown in figure 1 along with an example of a discrete wraparound array. If the cavity size and patch size are not convenient, you must either change the cavity size as possible or change the radius of the cylinder. If the radius is changed slightly, it will not effect the electromagnetic properties of the structure, but it may allow uniform discretization. Although the restrictions imposed by the uniform zoning requirement seems rather stringent, with practice, an antenna designer will find that EXCALIBUR is quite flexible.

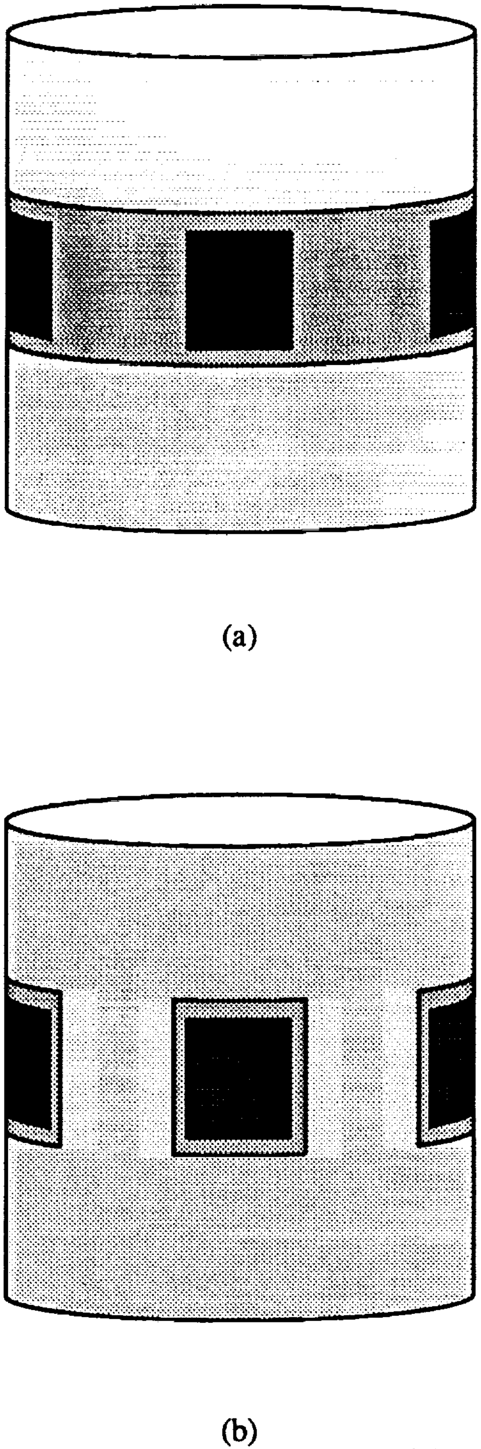
3 Compilation

The first task in utilizing EXCALIBUR is to compile and link the various files of the program. The following files are required to run EXCALIBUR and are given on the distribution diskette: *excalibur.f,dyadic.f*, *matrixGenerate.f*, *preProcessor.f*, *rcs.f*, *fft.f*, *gauss.inc*, *fft.inc* and *excalibur.inc.* These files perform the following functions:

- *excalibur.f*: Main program, BiCG solver, matrix building subroutines, FE-BI subroutine, impedance insert and various auxiliary subroutines.
- *dyadic.f*: Compute dyadic Green's function terms for admittance matrix, on-surface and far-field Fock functions and gamma function.
- *matrixGenerate.f*: Boundary integral and FE matrix terms.
- *preProcessor.f*: Geometry/mesh generator.
- *rcs.f*: Radar Cross Section, far-zone dyadic Green's function and plane wave excitation functions.
- *fft.f*: Forward and inverse Fast Fourier Transform subroutines.
- *gauss.inc*: Numerical integration parameters for gaussian quadrature.
- *fft.inc*: Include file for 2-D FFT subroutines.
- *excalibur.inc*: Main memory allocation file also contains variable dictionary.

The distribution disk also contains the following Fast Fourier Transform options:

- *fftCooley.f*: Classic Cooley-Tukey Radix-2 algorithm.
- *fftSplitRadix.f*: Split-Radix algorithm by Sorenson [3].



(a)

(b)

Figure 1: Illustration of two types of arrays: (a) wraparound array; (b) discrete cavity array

- fftCRAY.f: Calls the CRAY library vectorized, multitasked 2-D FFT routine.
- fftCONVEX.f: Calls the *veclib* vectorized 2-D FFT routine.

Prior to compilation, the user should copy one of these files into *fft.f.* The optimized routines for the CRAY and CONVEX architectures should be used when possible while in general the Split-Radix algorithm should give better performance than the Cooley-Tukey version. To enhance their efficiency, the Split-Radix and Cooley-Tukey files utilize a decimation-in-frequency forward transform and a decimation-in-time inverse transform in order to avoid the need to perform bit reversal.

Another file which is included on the distribution diskette is *convertToASCII.f.* To save disk space, the geometry information is stored in a binary format by *preprocessor.* The program *convertToASCII* is included to produce a human-readable file (ASCII). The nodes, elements, edges, unknowns and other useful information is provided in a easy to read (although disk space consuming) format. All the programs are compiled and linked by invoking the UNIX *make* utility. A *Makefile* has been provided on the distribution disk. To date, EXCALIBUR has been successfully compiled, linked and run on the following architectures/operating systems: SUN, DEC UNIX, HP 9000/7xx, IBM RS/6000, Silicon Graphics IRIS, CRAY and CONVEX. Three variable must be set within the *Makefile*:

- FF: The Fortran compiler name for the architecture.
- FOPT: The Fortran compiler options i.e. optimization, precision, etc.
- LOPT: The name of any libraries required for linking.

The user should uncomment these variables for the target architectures in *Makefile.* EXCALIBUR is constructed by simply typing *make* at the command line, while the binary-to-ASCII conversion program is constructed by entering *make convert.* EXCALIBUR is invoked by typing *excalibur* at the command line while the conversion program is run with the command *convertToASCII.* Finally, the directory may be cleaned up of all object and executable file by typing *make clean.*

4 Geometry Information

The binary geometry file created by *preprocessor* contains all the information concerning the physical structure under study except for the placement of any probe-feeds or lumped impedance posts. Therefore, it is important that the user be aware of the geometry entered into the FE-BI portion of the code. As previously mentions, the *convertToASCII* program creates a human-readable file from the machine-readable geometry file.

The first information provided in the resulting ASCII file is the header. which contains the number of nodes, number of edges, number of unknowns, etc. and an example of the header is shown in figure 2. The next field contains the node information. The information given is as follows (see figure 3):

- Column 1: Node number
- Column 2: Radial (ρ) coordinate in centimeters.
- Column 3: Angular (ϕ) coordinate in degrees.
- Column 3: Axial (z) coordinate in centimeters.
- Column 4: Layer number from top of the cylinder (aperture).
- Column 5: Row number from lowest axial coordinate.
- Column 6: Column number from smallest azimuthal coordinate.

Each node is associated with a physical location (ρ,ϕ,z) and a grid location (layer,row,column).

Grid points must be used in the discretization of a geometry since the BiCG-FFT solver requires that each node pair lie an integer number of units apart. Thus, the distance between two nodes (primed and unprimed) on the surface of the cylinder is given by

$$R(n,m;n',m') = \sqrt{(n-n')a\Delta\phi + (m-m')\Delta z} \qquad (1)$$

EXCALIBUR distinguishes between grid points and nodes. A grid point can be thought of as the intersection of two lines of a piece of graph paper which is placed on the surface of the cylinder. A node is a grid point which lies within a cavity. The row and column number associated with a node actually is the row and column number of the grid point which formed the node. The first grid point which corresponds to the lower-left corner of the grid has row

```
Binary filename:test.exc


ASCII filename:test.ascii


NODE statistics:

Total number of nodes:  50
Number of nodes on the surface:  25
Number of nodes along the metallic walls:  41
Number of nodes on surface metallic patches:  0
Number of nodes which are resistive:  0


EDGE statistics:

Total number of edges:  105
Interior edges:  9
Aperture edges:  24
         a) substrate edges:  24
         b) resistive edges:  0
Metal edges (NOT unknowns):  72

||||||||| UNKNOWNS |||||||||-->   33


ELEMENT statistics:

Total number of elements:  16
Surface Elements:  16
```

Figure 2: Geometry header.

```
Node        rho(cm)    phi(deg)      z(cm)  layer  row  column

      1    1.00000   -5.00000   -0.50000     0     0     0
      2    1.00000   -2.50000   -0.50000     0     0     1
      3    1.00000    0.00000   -0.50000     0     0     2
      4    1.00000    2.50000   -0.50000     0     0     3
      5    1.00000    5.00000   -0.50000     0     0     4
      6    1.00000   -5.00000   -0.25000     0     1     0
      7    1.00000   -2.50000   -0.25000     0     1     1
      8    1.00000    0.00000   -0.25000     0     1     2
      9    1.00000    2.50000   -0.25000     0     1     3
     10    1.00000    5.00000   -0.25000     0     1     4
     11    1.00000   -5.00000    0.00000     0     2     0
     12    1.00000   -2.50000    0.00000     0     2     1
     13    1.00000    0.00000    0.00000     0     2     2
     14    1.00000    2.50000    0.00000     0     2     3
     15    1.00000    5.00000    0.00000     0     2     4
     16    1.00000   -5.00000    0.25000     0     3     0
     17    1.00000   -2.50000    0.25000     0     3     1
     18    1.00000    0.00000    0.25000     0     3     2
     19    1.00000    2.50000    0.25000     0     3     3
     20    1.00000    5.00000    0.25000     0     3     4
     21    1.00000   -5.00000    0.50000     0     4     0
     22    1.00000   -2.50000    0.50000     0     4     1
     23    1.00000    0.00000    0.50000     0     4     2
     24    1.00000    2.50000    0.50000     0     4     3
     25    1.00000    5.00000    0.50000     0     4     4
     26    0.90000   -5.00000   -0.50000     1     0     0
     27    0.90000   -2.50000   -0.50000     1     0     1
     28    0.90000    0.00000   -0.50000     1     0     2
     29    0.90000    2.50000   -0.50000     1     0     3
     30    0.90000    5.00000   -0.50000     1     0     4

              << Remainder of nodes truncated >>
```

Figure 3: Node information.

= 0 and column = 0. For a wraparound grid, the first grid point is physically located at $\phi = -180^o$ and once again has row = 0 and column = 0.

The next set of information provided is the edges which form the cavities. The information given is as follows (see figure 4)

- Column 1: Edge number.
- Column 2: Left (lower) node forming the edge.
- Column 3: Right (upper) node forming edge.
- Column 3: Unknown number (zero indicates a fixed edge (e.g. metal)).
- Column 4: Orientation (ρ-, ϕ- or z-directed).
- Column 5: Type of edge (metal,substrate,resistive or interior).

Each edge is associated with two nodes and hence has an orientation in the cylindrical coordinate system. If an edge is metal, since EXCALIBUR uses a total field formulation, that edge's weight is fixed at zero. All other edges are unknowns which must be solved using the BiCG-FFT solver. A substrate or resistive edge is associated with the boundary integral while interior edges contribute only to the FE portion of the system. Currently, resistive cards are not implemented in EXCALIBUR.

The next set of information related the unknowns on the aperture of the cavities to their edge number. It also includes the row and column number of that edge in the discretization grid. Although this information is useful for understanding the mechanics of the BiCG-FFT solver, it is of little interest to the general user. The given information is (see figure 5)

- Column 1: Unknown number.
- Column 2: Associated edge number.
- Column 3: Row of this edge in the discretization.
- Column 3: Column of this edge in the discretization.

The edges which form each element of the mesh are given next. Each cylindrical shell element consists of eight nodes which form twelve edges. This information is useful in visualizing the mesh and could be hooked into a graphics package to generate a 3-D picture of the mesh. The prototype element is shown in figure 6 which displays the node numbering scheme. The information given by *convertToASCII* is (see figure 7)

- Row 1, Column 1: Element number.

```
Edge          Node 1     Node 2      Unknown   Orientation       Type

         1        26          1            0   rho-directed      metal
         2        27          2            0   rho-directed      metal
         3        31          6            0   rho-directed      metal
         4        32          7           25   rho-directed      interior
         5        26         27            0   phi-directed      metal
         6         1          2            0   phi-directed      metal
         7        31         32            0   phi-directed      metal
         8         6          7            1   phi-directed      substrate
         9        26         31            0   z-directed        metal
        10         1          6            0   z-directed        metal
        11        27         32            0   z-directed        metal
        12         2          7           13   z-directed        substrate
        13        28          3            0   rho-directed      metal
        14        33          8           26   rho-directed      interior
        15        27         28            0   phi-directed      metal
        16         2          3            0   phi-directed      metal
        17        32         33            0   phi-directed      metal
        18         7          8            2   phi-directed      substrate
        19        28         33            0   z-directed        metal
        20         3          8           14   z-directed        substrate
        21        29          4            0   rho-directed      metal
        22        34          9           27   rho-directed      interior
        23        28         29            0   phi-directed      metal
        24         3          4            0   phi-directed      metal
        25        33         34            0   phi-directed      metal
        26         8          9            3   phi-directed      substrate
        27        29         34            0   z-directed        metal
        28         4          9           15   z-directed        substrate
        29        30          5            0   rho-directed      metal
        30        35         10            0   rho-directed      metal

                      << Remaining edges truncated >>
```

Figure 4: Edge information.

Unknown	Edge	Row	Column
1	8	2	1
2	18	2	3
3	26	2	5
4	34	2	7
5	40	4	1
6	47	4	3
7	52	4	5
8	57	4	7
9	63	6	1
10	70	6	3
11	75	6	5
12	80	6	7
13	12	1	2
14	20	1	4
15	28	1	6
16	44	3	2
17	49	3	4
18	54	3	6
19	67	5	2
20	72	5	4
21	77	5	6
22	90	7	2
23	95	7	4
24	100	7	6

Figure 5: Relationship between unknown number and edge number on aperture.

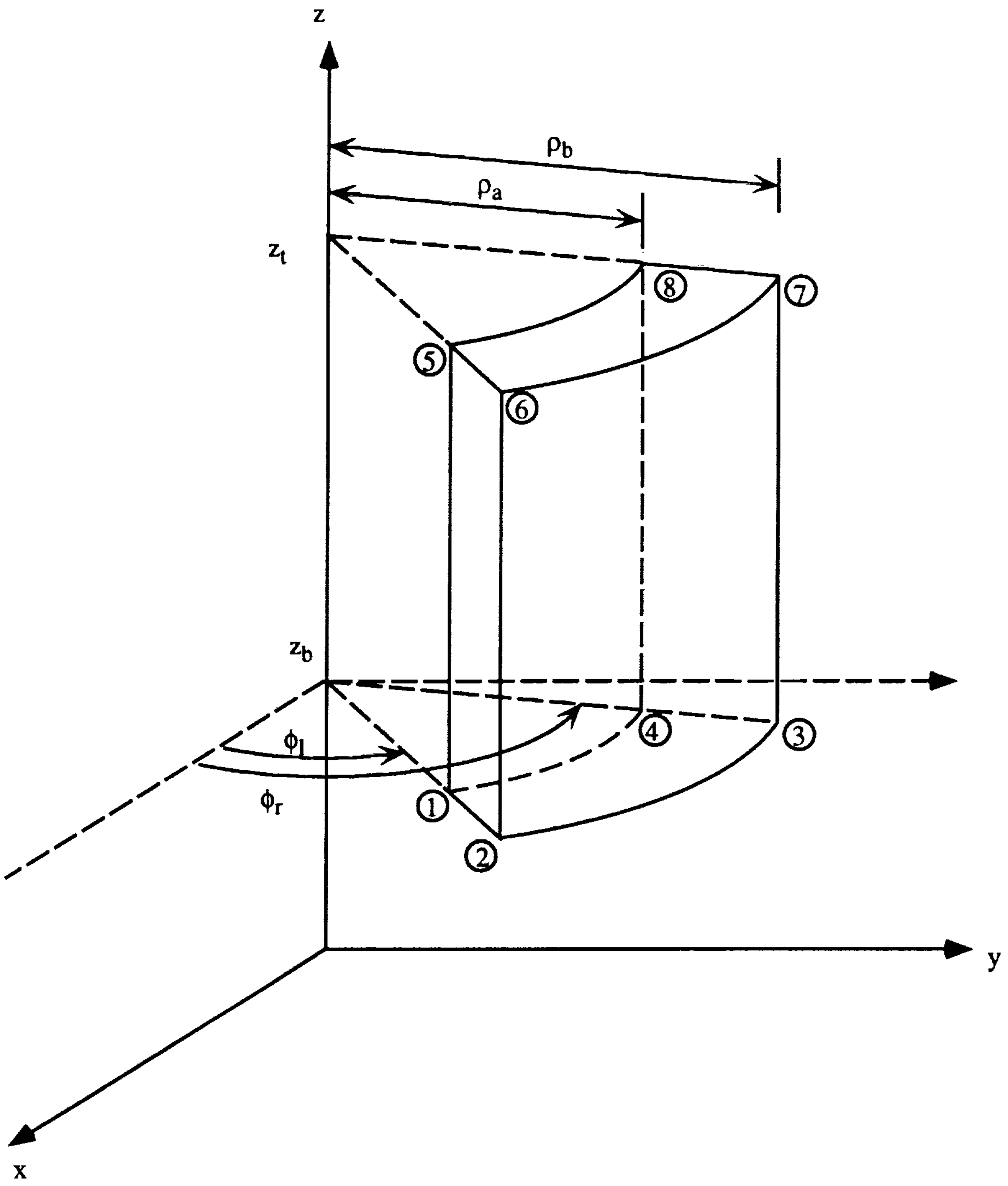


Figure 6: Cylindrical shell element.

- Row 1, Column 2-5: ρ-directed edges.
- Row 2, Column 2-5: ϕ-directed edges.
- Row 3, Column 2-5: z-directed edges.

The final set of information provided is the element parameters as shown in figure 6 which includes (see figure 8)

- Row 1: Element number.
- Row 2: ρ_a, ρ_b and t = $\rho_b - \rho_a$.
- Row 3: ϕ_l, ϕ_r and $\alpha = \phi_r - \phi_l$.
- Row 3: z_b, z_t and h = $z_t - z_b$.

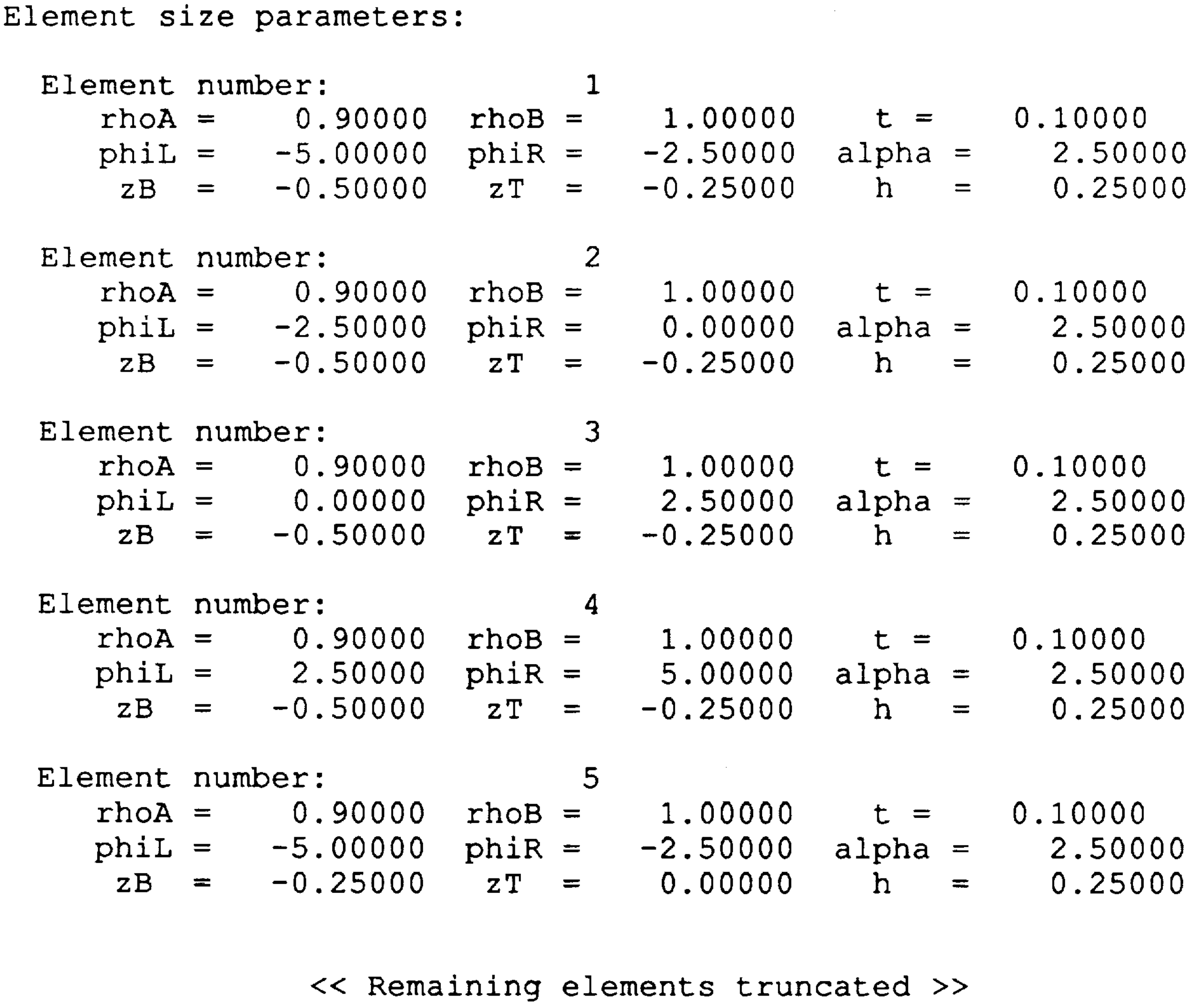
5 Geometry Generation

Having reviewed the geometry information provided by *preprocessor* through *convertToASCII*, we are prepared to generate some example geometries. Specifically, we shall look at radiation and scattering by a 2 cm × 3 cm patch antenna which is placed in a 5 cm × 6 cm cavity, a four element discrete array of such cavities and the same radiating array placed in a continuous wraparound cavity. These three variations of the same geometry exhibit the main classes of structures encountered in practice. Users will find it very helpful to check the entered geometry via *convertToASCII* prior to running the solver part of EXCALIBUR. In particular, it is useful to discretize the cavity without patches present and retain the node information since it will be necessary to specify the row and column of the lower-left corner of each patch as well as the number of edges along each side.

EXCALIBUR has a *preprocessor* module which generates the required mesh. It first generates the surface nodes which are then used to create the 3-D mesh. An in-house modeling package, such as SDRC IDEAS, may be used to create such a surface grid so long as the nodes are an integer number of units apart. It would be quite easy to interface such a package with EXCALIBUR by replacing the subroutine *simpleMesh* with a universal file reader. However, we have found that the custom mesh routine provided with EXCALIBUR (*simpleMesh*) is sufficient for most modeling tasks while

```
Element                             Edges

      1              1          2          3          4
                     5          6          7          8
                     9         10         11         12

      2              2         13          4         14
                    15         16         17         18
                    11         12         19         20

      3             13         21         14         22
                    23         24         25         26
                    19         20         27         28

      4             21         29         22         30
                    31         32         33         34
                    27         28         35         36

      5              3          4         37         38
                     7          8         39         40
                    41         42         43         44

              << Remaining elements truncated >>
```

Figure 7: Edges associated with each element.



```
Element size parameters:

  Element number:                 1
     rhoA =     0.90000  rhoB =    1.00000     t =     0.10000
     phiL =    -5.00000  phiR =   -2.50000   alpha =     2.50000
      zB  =    -0.50000   zT  =   -0.25000     h   =     0.25000

  Element number:                 2
     rhoA =     0.90000  rhoB =    1.00000     t =     0.10000
     phiL =    -2.50000  phiR =    0.00000   alpha =     2.50000
      zB  =    -0.50000   zT  =   -0.25000     h   =     0.25000

  Element number:                 3
     rhoA =     0.90000  rhoB =    1.00000     t =     0.10000
     phiL =     0.00000  phiR =    2.50000   alpha =     2.50000
      zB  =    -0.50000   zT  =   -0.25000     h   =     0.25000

  Element number:                 4
     rhoA =     0.90000  rhoB =    1.00000     t =     0.10000
     phiL =     2.50000  phiR =    5.00000   alpha =     2.50000
      zB  =    -0.50000   zT  =   -0.25000     h   =     0.25000

  Element number:                 5
     rhoA =     0.90000  rhoB =    1.00000     t =     0.10000
     phiL =    -5.00000  phiR =   -2.50000   alpha =     2.50000
      zB  =    -0.25000   zT  =    0.00000     h   =     0.25000

                 << Remaining elements truncated >>
```

Figure 8: Element parameters.

being quite efficient. For this manual, we will use this package for all meshing requirements.

The first example is a 2 cm × 3 cm patch antenna recessed in a 5 cm × 6 cm cavity which is centered at ($\phi = 0^\circ$, z = 0 cm). The required information is as follows

- Choose item 1 (Preprocessor) from main menu.
- Enter radius of the cylinder in centimeters.
- Enter angular and axial size of grid in degrees and centimeters.
- Enter center of grid in degrees and centimeters.
- Enter number of grid points in azimuthal and axial directions.
- Enter number of identical cavities in the azimuthal and axial directions.
- Enter number of nodes per cavity.
- Indicate whether all surface nodes are metallic or resistive.
- Indicate whether all surface nodes are on the substrate (0 means a patch is present).
- Enter number of patches.
- Enter row and column of lower-left hand node of the patch (see node section of the geometry file for this information).
- Enter number of edges in (ϕ,z) directions for this patch.
- Indicate any additional metallic nodes (-999 2 denotes no remaining nodes).
- Enter number of substrate layers.
- For each layer, enter its thickness in centimeters.
- Enter 1 to save this geometry.
- Enter filename for this geometry.

The transcript for this geometry is shown on the following pages. The next two examples are a discrete array of these elements and a continuous wraparound array. The main difference is that the number of grid points is not the same as the number of nodes on the cavity surface. Essentially, the same information is entered as above with the exception that these two arrays have a wraparound (360°) grid. See the attached transcripts for more details. Note for continuous wraparound cavities, if a patch crosses the branch cut ($\phi = \pm 180^\circ$), the nodes along the lower edge of the patch must be hand entered as shown in example 3. These nodes are obtained by running the *preprocessor* without specifying any patches and inspecting the human-readable geometry file. Note that for these three examples, the number of unknowns is dramatically different. For the single cavity of example 1, only 541 unknowns are required. The four identical discrete cavities requires 2164 unknowns which is of course 4 × 541. Finally, the continuous wraparound cavity has 12992 unknowns!

```
          ||||||||||||||||||||||||||||||||||||||||||
          |                                        |
          |               Example 1                |
          |     Single Discrete Cavity Example     |
          |                                        |
          ||||||||||||||||||||||||||||||||||||||||||

 Do you wish to run:
                     1) Preprocessor
                     2) FEM-BI
                     3) Impedance aperture
                     0) Exit
1

 Performing initialization, please wait....

 Enter radius of cylinder (cm):
15.27887
 Enter array size (phi,z) in (deg,cm):
18.75 6
 Enter center of array (phi,z) in (deg,cm):
0 0
 Enter number of grid points in (phi,z) direction:
11 25

 Discretization:
                  deltaPhi =  1.875 deg
                  deltaZ   =  .25 cm

 Enter number of cavities in phi,z directions
1 1

 Enter number of nodes per cavity (phi,z):
11 25


 275 surface nodes have been generated...
 The node numbering has the following pattern

 SINGLE CAVITY:

   ^
   |   21   22   23   24   25
   |   16   17   18   19   20
   z   11   12   13   14   15
   |    6    7    8    9   10
   |    1    2    3    4    5
   ---------- phi ---------->

 if for example 25 nodes were specified.
 where node 13 is the center point.


 Now specify which other nodes are either:
             metallic
                or
             resistive

 All surface nodes metal or resistive (1=yes):
0
 Are the nodes substrate nodes (1=yes):
0
 Enter number of metallic patches:
1
 Enter row,column of lower left corner of
 the patch on uniform grid for patch: 1
6 3
```

```
 Enter number of unknowns(edges) for each
 dimension (phi,z):
4 12

  << List of nodes constituting the patch deleted >>
  << Each of these nodes is specified by (phi,z) coord >>

 Now enter any individual nodes...
 Specify the class (metal or resistive) of all
 surface nodes which are NOT substrate nodes:
 Also indicate if 2 = metal or 3 = resistive...

 Enter node number (-999 if done):
-999 2

 0 resitive nodes entered...
 66 metal nodes entered...

 Enter number of substrate layers(INTEGER):
1

 Generating sub-surface nodes....
 Enter layer  1 thickness (cm):
0.07874

 All  550 NODES have now been created
 which form 240 elements...

 Generating edges...

 1303 edges generated....


 All element edges have been identified...

 Classifying edges....

 Number of elements: 240
 Total number of nodes:  550
 Total number of edges:  1303
 Number of interior edges:  207
 Number of metal edges:  762
 Number of aperture edges:  334
      Resistive edges:  0
      Substrate edges:  334
 Number of unknowns:  541

 Determining unknown order...

 Do you want to save this geometry (1=yes):
1
 Enter EXCALIBUR filename:
example1.exc

 Writing binary EXCALIBUR file:
 example1.exc

 Do you wish to run:
                     1) Preprocessor
                     2) FEM-BI
                     3) Impedance aperture
                     0) Exit
0
```

```
        |||||||||||||||||||||||||||||||||||||||||
        |                                       |
        |              Example 2                |
        |    Four Discrete Cavities Example     |
        |                                       |
        |||||||||||||||||||||||||||||||||||||||||

 Do you wish to run:
                    1) Preprocessor
                    2) FEM-BI
                    3) Impedance aperture
                    0) Exit
1

 Performing initialization, please wait....

 Enter radius of cylinder (cm):
15.27887
 Enter array size (phi,z) in (deg,cm):
360 6
 Wrap-around cavity entered...
 Enter center of array (z) in (cm):
0
 Enter number of grid points in (phi,z) direction:
192 25

 Discretization:
                 deltaPhi =  1.875 deg
                 deltaZ   =  .25 cm

 Enter 1 if ring element, 0 if rectangular array:
0
 Enter number of cavities in phi direction
4

 Enter number of nodes per cavity (phi,z):
11 25
 Enter lower left-hand (row,column): 1
0 43
 Enter lower left-hand (row,column): 2
0 91
 Enter lower left-hand (row,column): 3
0 139
 Enter lower left-hand (row,column): 4
0 187


 1100 surface nodes have been generated...
 The node numbering has the following pattern

 AZMUTHAL CAVITY ARRAY:

   ^
   |
   |  19  20  21     22  23  24
   z  13  14  15     16  17  18
   |   7   8   9     10  11  12
   |   1   2   3      4   5   6
   ---------- phi ------------->

 if for example two 3x4 cavities
 were specified


 Now specify which other nodes are either:
            metallic
               or
```

```
              resistive

 All surface nodes metal or resistive (1=yes):
0
 Are the nodes substrate nodes (1=yes):
0
 Enter number of metallic patches:
4
 Enter row,column of lower left corner of
 the patch on uniform grid for patch: 1
6 46
 Enter number of unknowns(edges) for each
 dimension (phi,z):
4 12
 << Node list deleted >>
 Enter row,column of lower left corner of
 the patch on uniform grid for patch: 2
6 94
 Enter number of unknowns(edges) for each
 dimension (phi,z):
4 12
 << Node list deleted >>
 Enter row,column of lower left corner of
 the patch on uniform grid for patch: 3
6 142
 Enter number of unknowns(edges) for each
 dimension (phi,z):
4 12
 << Node list deleted >>
 Enter row,column of lower left corner of
 the patch on uniform grid for patch: 4
6 190
 Enter number of unknowns(edges) for each
 dimension (phi,z):
4 12
 << Node list deleted >>
 Now enter any individual nodes...
 Specify the class (metal or resistive) of all
 surface nodes which are NOT substrate nodes:
 Also indicate if 2 = metal or 3 = resistive...

 Enter node number (-999 if done):
-999 2

 0 resitive nodes entered...
 261 metal nodes entered...

 Enter number of substrate layers(INTEGER):
1

 Generating sub-surface nodes....
 Enter layer  1 thickness (cm):
0.07874

 All  2200 NODES have now been created
 which form 960 elements...

 Generating edges...

 5212 edges generated....


 All element edges have been identified...
```

```
 Classifying edges....
 Number of elements: 960
 Total number of nodes:  2200
 Total number of edges:  5212
 Number of interior edges:  828
 Number of metal edges:  3048
 Number of aperture edges:  1336
      Resistive edges:  0
      Substrate edges:  1336
 Number of unknowns:  2164

 Determining unknown order...


 Do you want to save this geometry (1=yes):
1
 Enter EXCALIBUR filename:
example2.exc

 Writing binary EXCALIBUR file:
 example2.exc

 Do you wish to run:
                     1) Preprocessor
                     2) FEM-BI
                     3) Impedance aperture
                     0) Exit
0
```

```
          |||||||||||||||||||||||||||||||||||||||||
          |                                       |
          |               Example 3               |
          |      Four Patches on Continuous       |
          |      Wraparound Cavity Example        |
          |                                       |
          |||||||||||||||||||||||||||||||||||||||||

 Do you wish to run:
                        1) Preprocessor
                        2) FEM-BI
                        3) Impedance aperture
                        0) Exit
1

 Performing initialization, please wait....

 Enter radius of cylinder (cm):
15.27887
 Enter array size (phi,z) in (deg,cm):
360 6
 Wrap-around cavity entered...
 Enter center of array (z) in (cm):
0
 Enter number of grid points in (phi,z) direction:
192 25

 Discretization:
                  deltaPhi =  1.875 deg
                  deltaZ   =  .25 cm

 Enter 1 if ring element, 0 if rectangular array:
1
 Enter number of identical rings (>=1):
1 Enter number of nodes per cavity (phi,z):
192 25
 Enter lower left-hand (row,column): 1
0 0


 4800 surface nodes have been generated...
 The node numbering has the following pattern

 SINGLE CAVITY:

   ^
   |   21   22   23   24   25
   |   16   17   18   19   20
   z   11   12   13   14   15
   |    6    7    8    9   10
   |    1    2    3    4    5
   ---------- phi ---------->

 if for example 25 nodes were specified.
 where node 13 is the center point.


 Now specify which other nodes are either:
            metallic
               or
            resistive

 All surface nodes metal or resistive (1=yes):
0
 Are the nodes substrate nodes (1=yes):
0
 Enter number of metallic patches:
```

```
4
 Enter row,column of lower left corner of
 the patch on uniform grid for patch: 1
6 46
 Enter number of unknowns(edges) for each
 dimension (phi,z):
4 12
    << Metallic node list deleted >>
 Enter row,column of lower left corner of
 the patch on uniform grid for patch: 2
6 94
 Enter number of unknowns(edges) for each
 dimension (phi,z):
4 12
    << Metallic node list deleted >>
 Enter row,column of lower left corner of
 the patch on uniform grid for patch: 3
6 142
 Enter number of unknowns(edges) for each
 dimension (phi,z):
4 12
    << Metallic node list deleted >>
 Enter row,column of lower left corner of
 the patch on uniform grid for patch: 4
6 190
 Enter number of unknowns(edges) for each
 dimension (phi,z):
4 12
 This patch includes branch...
 Enter first row nodes:
1343
 1343 176.25 -1.5
1344
 1344 178.1249 -1.5
1153
 1153 -180.0 -1.5
1154
 1154 -178.125 -1.5
1155
    << Metallic node list deleted >>
 Now enter any individual nodes...
 Specify the class (metal or resistive) of all
 surface nodes which are NOT substrate nodes:
 Also indicate if 2 = metal or 3 = resistive...

 Enter node number (-999 if done):
-999 2

 0 resitive nodes entered...
 261 metal nodes entered...

 Enter number of substrate layers(INTEGER):
1

 Generating sub-surface nodes....
 Enter layer  1 thickness (cm):
0.07874

 All  9600 NODES have now been created
 which form 4584 elements...

 Generating edges...

 23616 edges generated....

 All element edges have been identified...
 Classifying edges....

 Number of elements: 4608
 Total number of nodes:  9600
 Total number of edges:  23616
 Number of interior edges:  4416
 Number of metal edges:  10624
 Number of aperture edges:  8576
      Resistive edges:  0
      Substrate edges:  8576
 Number of unknowns:  12992

 Determining unknown order...


 Do you want to save this geometry (1=yes):
1
 Enter EXCALIBUR filename:
example3.exc

 Writing binary EXCALIBUR file:
 example3.exc

 Do you wish to run:
                     1) Preprocessor
                     2) FEM-BI
                     3) Impedance aperture
                     0) Exit
0
```

6 Operation Modes

EXCALIBUR has three main operation modes for FE-BI calculations (option 2 from the main menu). They are: input impedance vs. frequency, radiation pattern or RCS vs. frequency and single frequency radiation and RCS pattern calculations. This section will describe each mode using example 1 above.

6.1 Input Impedance – Multiple Frequencies

The first option presented is calculation of a patch antenna's input impedance at multiple frequencies. This is most useful in determining the resonant frequency of a patch antenna. The following information is required

- Choose item 2 (FE-BI) from main menu.
- Enter the stored binary geometry file.
- Enter 1 if all elements have the same material parameters.
- Enter complex permittivity.
- Enter complex permeability.
- Enter BiCG convergence tolerance, minimum and maximum number of iterations.
- Enter 1 to monitor convergence.
- Enter 1 for diagonal preconditioning and 0 for no preconditioning.
- Enter 1 for frequency sweep of the input impedance.
- Enter name of file to store the input impedance.
- Enter number of probe feeds.
- Enter location of each feed in terms of (ϕ=degrees, z=cm).
- Enter which layer in which the feed is embedded.
- Enter complex current for this feed.
- Enter number of impedance post loads.
- Enter frequency range (in GHz) for this sweep.
- Return to main menu.

The input impedance for 3.1 GHz to 3.3 GHz computed every 10 MHz is shown in figure 9. The transcript for computing the input impedance of an axially polarized 2 cm × 3 cm patch in a 5 cm × 6 cm cavity is given on the following page.

```
 Do you wish to run:
                     1) Preprocessor
                     2) FEM-BI
                     3) Impedance aperture
                     0) Exit
2

 Performing initialization, please wait....

 Enter EXCALIBUR filename:
example1.exc
 Reading excalibur file: example1.exc

 Material Parameter Specification....
 Is the material filling constant (1=yes,0=no)?
1
 Enter relative permittivity [real,imaginary]:
2.17 0
 Enter relative permeability [real,imaginary]:
1 0

 Checking dimension allocations...

 Number of Boundary Integral Unknowns: 334
 Total number of UNKNOWNS: 541

Enter tolerance, minimum and maximum iterations:
0.01 2 500
 Do you wish to monitor convergence (0=no,1-yes)?
0
 Do you want:  0 = no preconditioning, 1 = diagonal?
1
 Do you want to compute:
      0 = RCS/Pattern, 1 = Zin, 2 = Freq. Sweep?
1
 Enter Zin filename:
example1.zin

 |||||||||||||||||||||||||||||||||||||||||
 |     PROBE FEED INFORMATION            |
 |||||||||||||||||||||||||||||||||||||||||

 Enter number of feeds:
1
 Enter feed location (phi,z) in (deg,cm):
0 -0.375
 Enter layer number of feed <=: 1
1
 Enter mag,phase of probe current (amp,deg):
1 0

 |||||||||||||||||||||||||||||||||||||||||
 |     IMPEDANCE LOAD INFORMATION        |
 |||||||||||||||||||||||||||||||||||||||||

 Enter number of loads:
0

 Enter start,stop and increment freq. (GHz):
3.1 3.3 0.01
 << Data shown in attached figure >>
```

6.2 Pattern – Multiple Frequencies

The next option presented is calculation of a radiation or RCS pattern at multiple frequencies. This is useful in computing the variation of gain or RCS with respect to frequency. Usually, a single observation angle is specified although multiple angles are allowed. This mode permits radiation, bistatic and backscatter computations. In addition, the input impedance as a function of frequency is stored if a probe feed is used for excitation. In this example, we compute the backscatter at normal incidence for an E_z-polarized plane wave as a function of frequency. The required information is

- Choose item 2 (FE-BI) from main menu.
- Enter the stored binary geometry file.
- Enter 1 if all elements have the same material parameters.
- Enter complex permittivity.
- Enter complex permeability.
- Enter BiCG convergence tolerance, minimum and maximum number of iterations.
- Enter 1 to monitor convergence.
- Enter 1 for diagonal preconditioning and 0 for no preconditioning.
- Enter 2 for frequency sweep of the far-zone fields.
- Enter name of file to store the input impedance.
- Enter name of file to store the RCS or Gain.
- Enter observation type (0 = backscatter).
- Enter start, stop and increment azimuth (ϕ) angles (in degrees).
- Enter start, stop and increment elevation (θ) angles (in degrees).
- Enter polarization angle (0 = E-pol, 90 = H-pol).
- Enter RCS filename.
- Enter number of probe feeds.
- Enter number of impedance post loads.
- Enter frequency range (in GHz) for this sweep.
- Return to main menu.

The RCS for 3.1 GHz to 3.3 GHz computed every 10 MHz is shown in figure 10. The transcript for backscatter calculations for backscatter calculations is shown on the next page.

```
 Do you wish to run:
                        1) Preprocessor
                        2) FEM-BI
                        3) Impedance aperture
                        0) Exit
2
 Performing initialization, please wait....

 Enter EXCALIBUR filename:
example1.exc
 Reading excalibur file: example1.exc

 Material Parameter Specification....
 Is the material filling constant (1=yes,0=no)?
1
 Enter relative permittivity [real,imaginary]:
2.17 0
 Enter relative permeability [real,imaginary]:
1 0
 Checking dimension allocations...
 Number of Boundary Integral Unknowns: 334
 Total number of UNKNOWNS: 541
Enter tolerance, minimum and maximum iterations:
0.01 2 500
 Do you wish to monitor convergence (0=no,1-yes)?
0
 Do you want:  0 = no preconditioning, 1 = diagonal?
1
 Do you want to compute:
      0 = RCS/Pattern, 1 = Zin, 2 = Freq. Sweep?
2
 Enter Zin filename:
test.zin
 Enter Frequency sweep data filename:
example1.fsw
 Enter observation type:
                             0) Backscatter
                             1) Bistatic
                             2) Radiation
0

 Enter start,stop, and increment azmuth angles [deg]:
0 0 1
 Enter start,stop,and increment elevation angles [deg]:
90 90 1
 Enter polarization angle [0 <= alpha <= 90 deg]:
           Ephi=0:  alpha = 0
           Etheta=0:  alpha = 90
0
 Enter ouput RCS filename [<= 40 characters]:
junk

 ||||||||||||||||||||||||||||||||||||||||
 |     PROBE FEED INFORMATION           |
 ||||||||||||||||||||||||||||||||||||||||

 Enter number of feeds:
0
 ||||||||||||||||||||||||||||||||||||||||
 |     IMPEDANCE LOAD INFORMATION       |
 ||||||||||||||||||||||||||||||||||||||||

 Enter number of loads:
0
 Enter start,stop and increment freq. (GHz):
3.1 3.3 0.01
```

6.3 Pattern – Single Frequency

The final operation mode is radiation and RCS pattern calculations at a single frequency. Of course, for the case of an antenna, the input impedance is also computed. This mode is generally used for multiple incident and observation angle applications. The only difference between this mode and the previous two modes is the option to save the admittance matrix in binary format. Since this matrix is excitation independent (though still frequency dependent), it can save time for multiple runs of a large geometry if this matrix is stored and then read in for each run. The required inputs are

- Choose item 2 (FE-BI) from main menu.
- Enter the stored binary geometry file.
- Enter 1 if all elements have the same material parameters.
- Enter complex permittivity.
- Enter complex permeability.
- Enter BiCG convergence tolerance, minimum and maximum number of iterations.
- Enter 1 to monitor convergence.
- Enter 1 for diagonal preconditioning and 0 for no preconditioning.
- Enter 0 for single frequency operation.
- Enter matrix storage/read option(1 to store, 2 to read, 0 to do nothing).
- Enter six (6) character filename for matrix storage/read.
- Enter observation type (1 = bistatic).
- Enter incidence angle (ϕ,θ).
- Enter start, stop and increment azimuth (ϕ) angles (in degrees).
- Enter start, stop and increment elevation (θ) angles (in degrees).
- Enter polarization angle (0 = E-pol, 90 = H-pol).
- Enter RCS filename.
- Enter number of probe feeds.
- Enter number of impedance post loads.
- Enter frequency (in GHz).
- Return to main menu.

This option is illustrated in the next session transcript for bistatic scattering and radiation pattern calculations.

```
 Do you wish to run:
                     1) Preprocessor
                     2) FEM-BI
                     3) Impedance aperture
                     0) Exit
2
 Performing initialization, please wait....
 Enter EXCALIBUR filename:
example1.exc
 Reading excalibur file: example1.exc
 Material Parameter Specification....
 Is the material filling constant (1=yes,0=no)?
1
 Enter relative permittivity [real,imaginary]:
2.17 0
 Enter relative permeability [real,imaginary]:
1 0
 Checking dimension allocations...
 Number of Boundary Integral Unknowns: 334
 Total number of UNKNOWNS: 541
Enter tolerance, minimum and maximum iterations:
0.01 2 500
 Do you wish to monitor convergence (0=no,1-yes)?
0
 Do you want:  0 = no preconditioning, 1 = diagonal?
1
 Do you want to compute:
      0 = RCS/Pattern, 1 = Zin, 2 = Freq. Sweep?
0
 Save the BI arrays in a file (0=no,1=yes,2=read)?
1
 All BI files will have the following form:
 <filename>.guu,<filename>.guv, etc.
 Enter BI binary filename(must be six(6) char):
exampl
 Enter observation type:
                          0) Backscatter
                          1) Bistatic
                          2) Radiation
1
 Enter incident phi,theta [deg]:
0 90
 Enter start,stop, and increment azmuth angles [deg]:
-180 180 1
 Enter start,stop,and increment elevation angles [deg]:
90 90 1
 Enter polarization angle [0 <= alpha <= 90 deg]:
            Ephi=0:  alpha = 0
            Etheta=0:  alpha = 90
0
 Enter ouput RCS filename [<= 40 characters]:
exampl.bi
 Enter frequency (GHz):
3.3
 |||||||||||||||||||||||||||||||||||||||||
 |     PROBE FEED INFORMATION            |
 |||||||||||||||||||||||||||||||||||||||||

 Enter number of feeds:
1
 Enter feed location (phi,z) in (deg,cm):
0 -0.375
 Enter layer number of feed <=: 1
1
 Enter mag,phase of probe current (amp,deg):
1 0
```

```
 ||||||||||||||||||||||||||||||||||||||||
 |     IMPEDANCE LOAD INFORMATION       |
 ||||||||||||||||||||||||||||||||||||||||

 Enter number of loads:
0
 Writing BI binary files...
 Generate bistatic excitation vector...
 Solve it...

  << Pattern deleted. >>

 Run anouther excitation? (1=yes,0=no)
1

 Enter observation type:
                          0) Backscatter
                          1) Bistatic
                          2) Radiation
2
 Enter start,stop, and increment azmuth angles [deg]:
-180 180 1
 Enter start,stop,and increment elevation angles [deg]:
90 90 1
 Enter ouput RCS filename [<= 40 characters]:
exampl.pat
 Enter normalized pattern filename:
exampl.norm
 Solve it...
  << Pattern deleted. >>
 Run anouther excitation? (1=yes,0=no)
0

 Do you wish to run:
                     1) Preprocessor
                     2) FEM-BI
                     3) Impedance aperture
                     0) Exit
0
```

The bistatic patterns taken at 3.3 GHz for the three geometry configurations presented in this manual are shown in figure 11 The corresponding antenna pattern comparisonat 3.3 GHz is shown in figure 12

7 Concluding Remarks

This user manual presented some basic operation information for the FE-BI code, EXCALIBUR. This presentation was only meant to get an initial user started. As one becomes experienced with the code, additional features such as 2-D patch array modeling, multiple feed arrays and use of lumped impedance loads may prove useful. Indeed, an experienced user will find that custom features may readily be added to EXCALIBUR. For example, currently EXCALIBUR allows entry of material parameters either for the entire substrate, each layer of the substrate or on an element-by-element basis. This subroutine *material* in file *excalibur.f* may readily be modified by the user to input a custom inhomogeneous substrate.

The code is fairly "dummy proof". If the user enters data which is not expected by EXCALIBUR such as a character when an integer is expected or an angle greater than 360^o, the code will prompt the user to re-enter the requested data. Additionally, the storage allocation parameters in *excalibur.inc* must be set by the user prior to compilation. If a particular parameter is too small for a given run, the code will halt and suggest a new value for the offensive parameter. The user must reset that parameter, recompile and run the code again. The program also estimates the amount of RAM required at the start of a run. This estimate is based on the storage required by the arrays in *excalibur.inc* plus some scratch arrays. Each complex number is assumed to require eight bytes and each integer and real number require four bytes. The user should consider this estimate to be a slightly lower than the actual consumed memory.

References

[1] L.C. Kempel and J.L. Volakis, "Scattering by cavity-backed antennas on a circular cylinder," submitted to *IEEE Trans. Antennas and Propagat.*.

[2] L.C. Kempel, J.L. Volakis and R. Sliva, "Radiation by cavity-backed antennas on a circular cylinder," to be submitted to *IEEE Trans. Antennas and Propagat.*.

[3] Sorenson *et al*, *IEEE Trans. Acoustics, Speech and Signal Processing*, pp. 152-156, Feb. 1986.

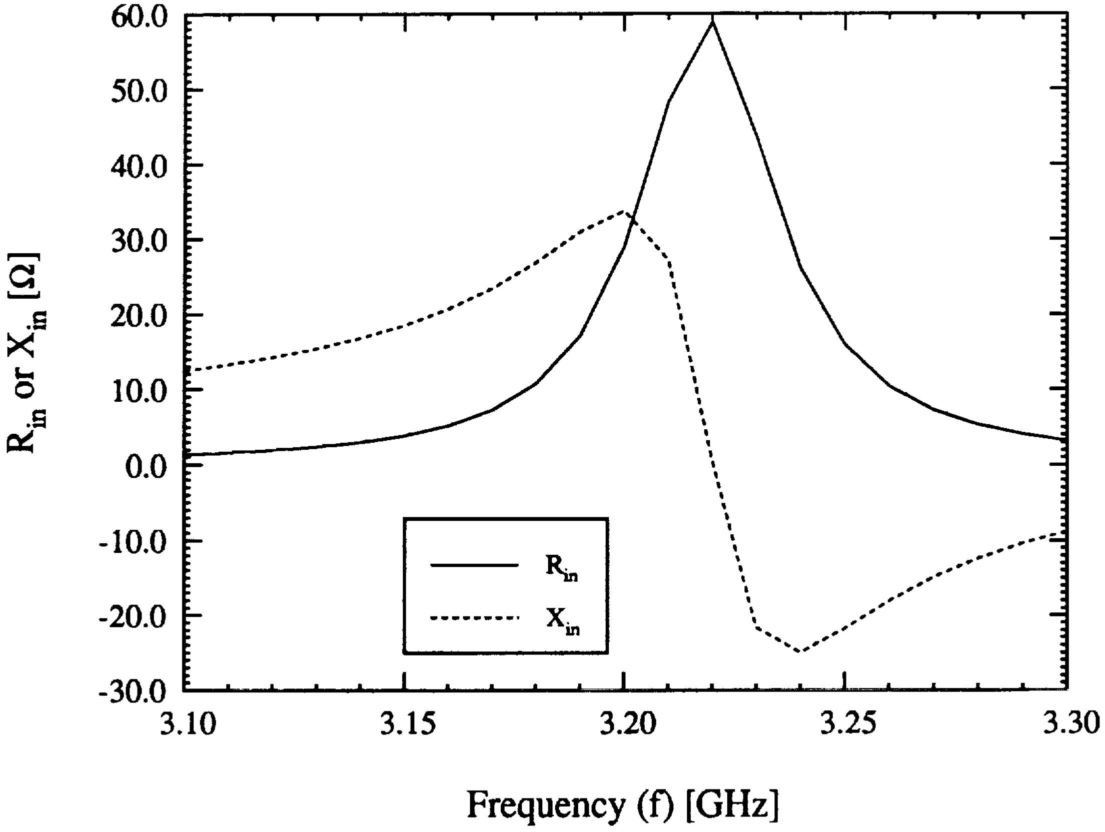


Figure 9: Input impedance vs. frequency for the axially polarized patch antenna which is 2 cm × 3 cm in a 5 cm × 6 cm cavity.

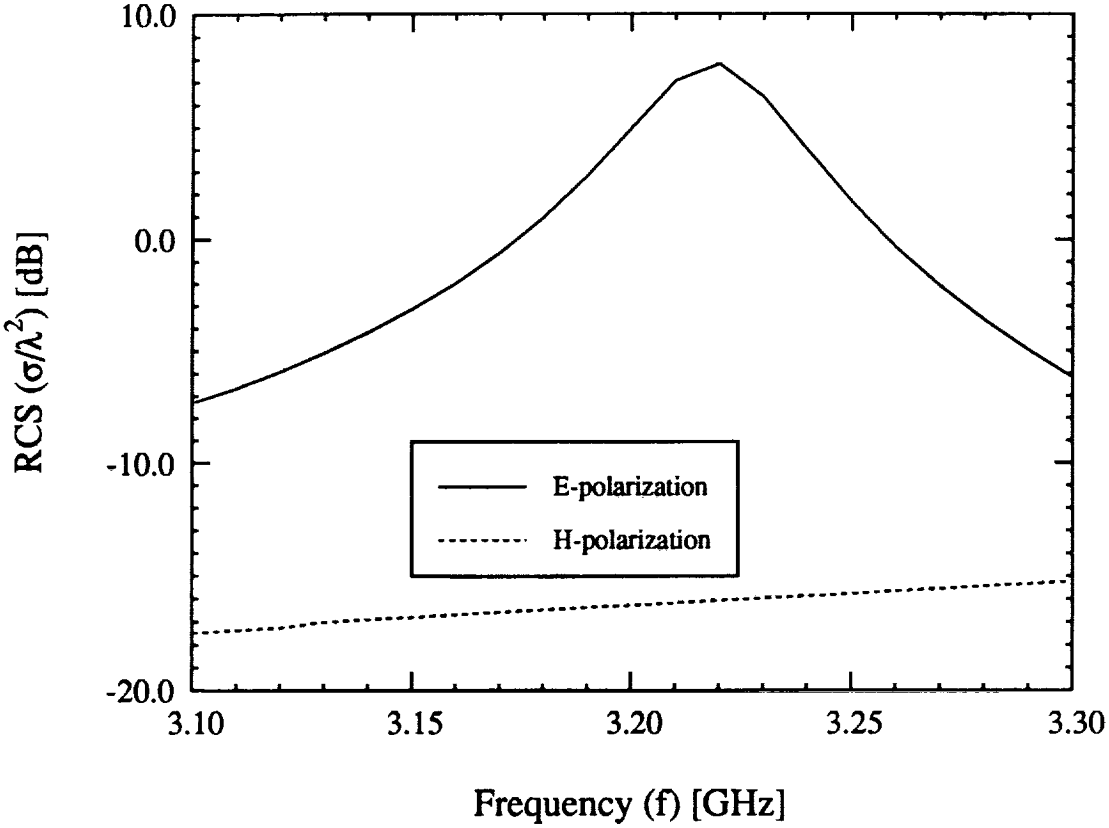


Figure 10: RCS vs. frequency for a patch antenna which is 2 cm × 3 cm in a 5 cm × 6 cm cavity.

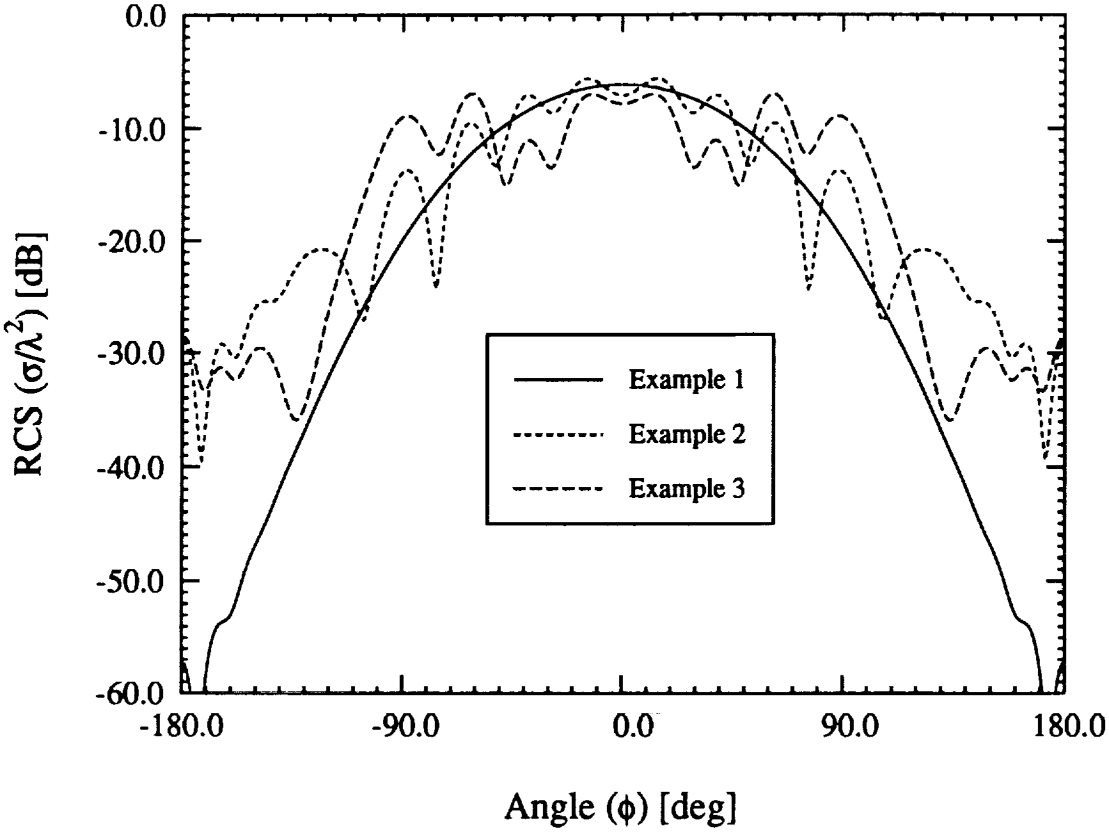


Figure 11: E-polarized bistatic patterns for a 2 cm × 3 cm embedded in the three different geometries presented in this manual. The incidence angle is $(0^o, 90^o)$ and observation is in the $\theta = 90^o$ plane.

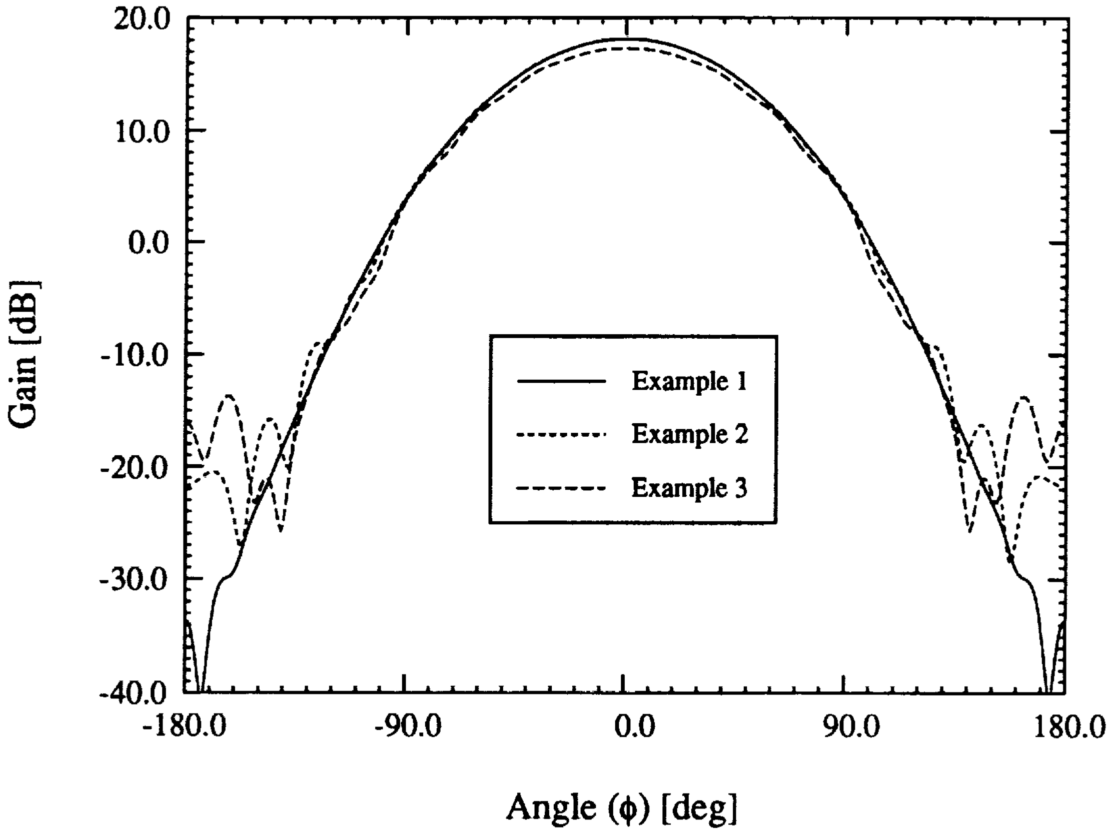


Figure 12: Axially polarized H-plane patterns for a 2 cm × 3 cm embedded in the three different geometries presented in this manual.

Appendix 1

A Note on Applying New Vector ABCs to Patch Antennas on a Circular Cylinder

Leo C. Kempel and John L. Volakis
Radiation Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109-2122

Recently, we developed a Finite Element-Boundary Integral (FE-BI) method which may be used to accurately model the scattering and radiation by conformal patch antennas. The use of a boundary integral to provide an exact mesh closure condition results in a rigorous formulation. When coupled with a Biconjugate Gradient-Fast Fourier Transform (BiCG-FFT) solver, the resulting computer code possesses both low memory and low computational demand. However, boundary integral closure involves a dyadic Green's function which necessarily limits its extension to coated geometries. Although it is possible to construct a suitable Green's function for multilayered superstrates, the resulting formulation is both complex and bulky. Furthermore, this approach cannot be extended in a rigorous manner to more general doubly curved structures.

Another hybrid FE method which has been used for very large 3-D scattering calculations at the University of Michigan is the finite element-absorbing boundary condition (FE-ABC) method. ABCs provide an approximate mesh closure which maintains a highly sparse matrix for the whole system. We propose utilizing a new conformal ABC (A. Chatterjee and J.L. Volakis, *Microwave and Optical Tech. Letters*, **6**, No. 16, pp. 886-889, Dec. 20 1993) which allows the closure surface to be brought quite close to the cavity aperture. As a result, the number of unknowns required is significantly less than the number required if a traditional spherical closure surface/boundary condition is used.

We have applied the second order ABC to radiation and scattering by cavity-backed patch antennas which are embedded in a metallic cylinder. Although extensive testing has not as of yet been accomplished, the preliminary results are encouraging. We looked at a 2 cm × 3 cm patch antenna which was printed on a 5 cm × 6 cm × 0.07874 cm substrate which had a dielectric constant of $\epsilon_r = 2.17$. The second order ABC was place 0.5λ from the cavity aperture as shown in figure 1. Two comparisons with the FE-BI formulation were performed. The first involved bistatic scattering with normal incidence ($\phi_i = 0^o$,$\theta_i = 90^o$) and observation in the $\theta = 90^o$ plane for an E_z-polarized plane wave which is shown in figure 2. A radiation pattern for the same antenna is shown in figure 3 where the resonant frequency is 3.3 GHz. Since this is an axially polarized element, figure 3 is an H-plane pattern.

The scattering and radiation example given herein involves a conformal antenna element without a covering superstrate. The FE-ABC formulation may be readily modified to account for such a covering which may consist of multiple inhomogeneous layers as shown in figure 1. A future report will develop the formulation for both covered and uncovered elements and it will provide suitable validation of the method.

In addition, we shall investigate whether the near-field performance of the FE-ABC method is sufficient for input and mutual impedance calculations. Having shown that the FE-ABC approach can indeed be used for conformal antennas on singly curved surfaces, we may in confidence develop a similar formulation for antennas mounted on doubly curved surfaces.

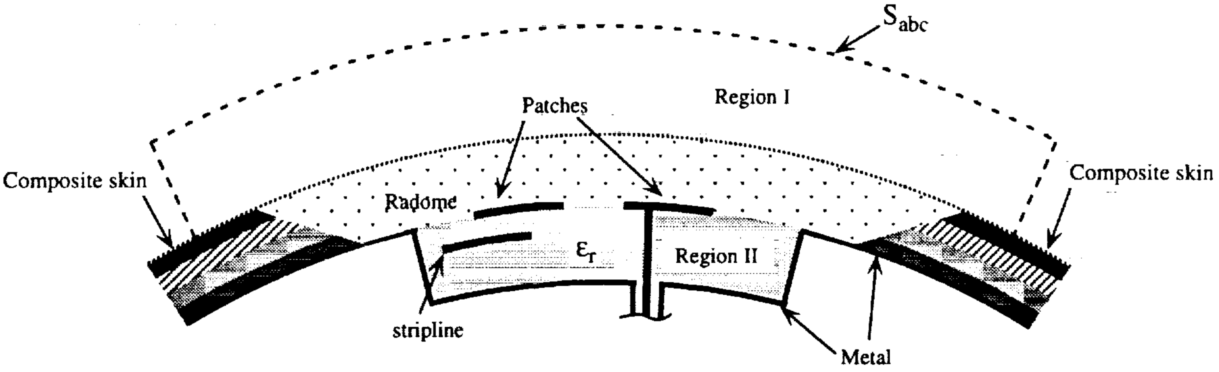


Figure 1: Coated cavity-backed patch antenna with ABC mesh termination.

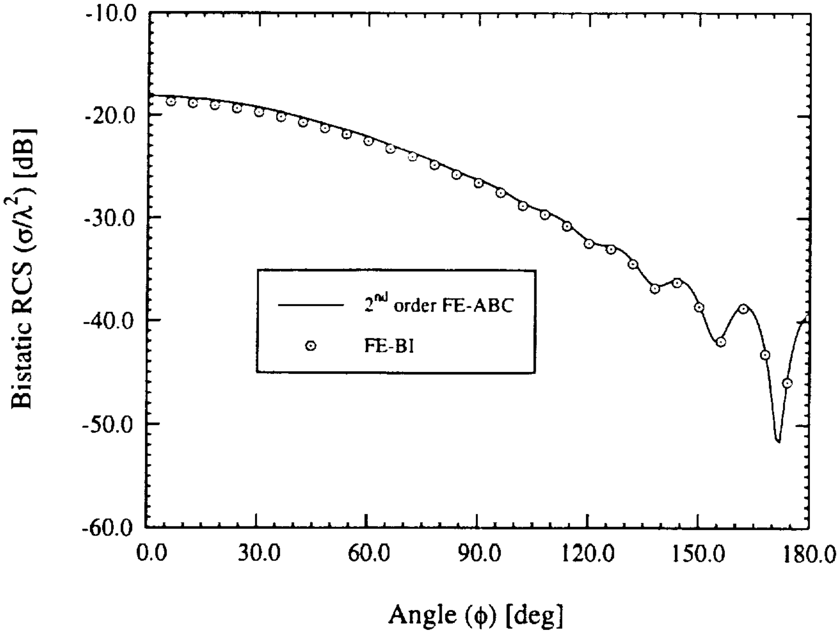


Figure 2: Comparison of 2^{nd} order conformal ABC mesh closure condition with the exact boundary integral condition for bistatic scattering. The patch was 2 cm × 3 cm printed on a 5 cm × 6 cm × 0.07874 cm dielectric substrate ($\epsilon_r = 2.17$). The ABC was placed 0.5λ from the aperture and the operating frequency was 3.0 GHz. The E_z-polarized plane wave was incident from ($\phi_i = 0^0$,$\theta_i = 90^o$) and observation is in the $\theta = 90^o$ plane.

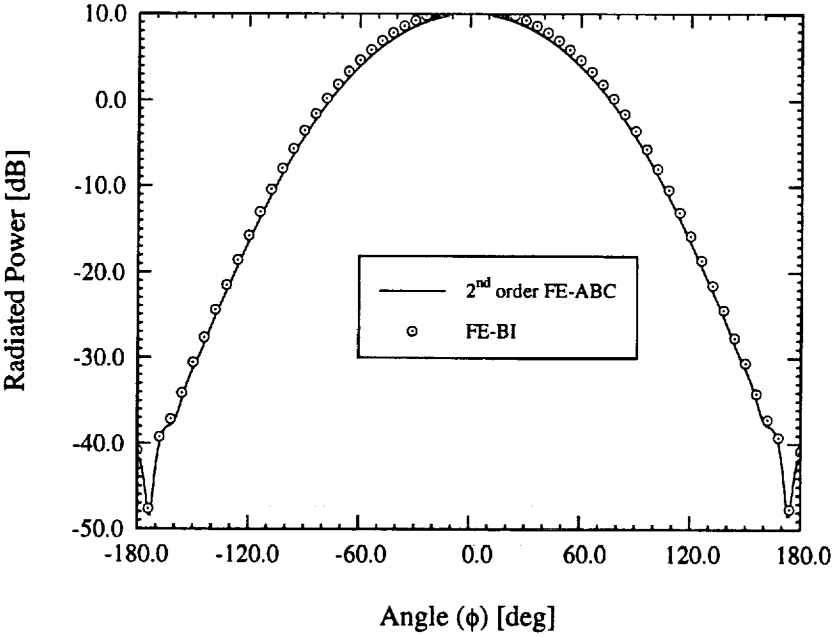


Figure 3: Comparison of 2^{nd} order conformal ABC mesh closure condition with the exact boundary integral condition for antenna pattern calculations. The geometry is identical to the one used in figure 2. The operating frequency was 3.3 GHz. The feed point was ($\phi_s = 0^o$,$z_s = -0.375$ cm) which results in axial polarization and observation is made in the H-plane.

APPENDIX 2

FE-BI implementation for coated planar platforms

Consider the cavity recessed in a coated ground plane as shown in Fig 1. The coating is assumed to be of thickness d and of relative permittivity ϵ_r. We will denote the free space region above the coating ($z > d$) as region 1, the region inside the coating ($0 < z < d$) as region 2 and that inside the cavity ($-c < z < 0$) as region 3. We will assume that the cavity is filled with an inhomogeneous material having a relative permittivity $\epsilon_{rc}(\mathbf{r})$ and relative permeability $\mu_{rc}(\mathbf{r})$. In accordance with the equivalence principle, the fields in regions 2 and 3 can be decoupled by closing the aperture with a perfect conductor and introduce the equivalent magnetic current

$$\mathbf{M} = \mathbf{E} \times \hat{z} \tag{1}$$

over the extent of the aperture, where $\mathbf{E}$ is the electric field at the aperture ($z = 0$). The magnetic field is then due to the radiation caused by the equivalent current $\mathbf{M}$ residing on the ground plane and is given by

$$\mathbf{H}(\mathbf{r}) \;=\; \mathbf{H}^i(\mathbf{r}) + \mathbf{H}^r(\mathbf{r}) - jkY\left[\bar{\mathbf{I}} + \frac{1}{k^2}\nabla\nabla\right] \cdot \mathbf{F} \tag{2}$$

where $\mathbf{F}$ is the magnetic vector potential and is given in terms of the dyadic Green's function of the coating as

$$\mathbf{F} = \iint_S \bar{\mathbf{G}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{M}(\mathbf{r}') dS' \tag{3}$$

and $\mathbf{H}^i$ is the incident field, $\mathbf{H}^r$ is that reflected by the coated ground plane without the aperture, S denotes the planar surface area of the aperture. If the conducting surfaces are restricted to horizontal planes, as is the situation here, we can find an equivalent scalar Green's function G_V such that [1]

$$\nabla \cdot \bar{\mathbf{G}} = \nabla' G_V(\mathbf{r}, \mathbf{r}') \tag{4}$$

where

$$G_V = \frac{\omega}{k_\rho^2 k_x} \frac{\partial G_E^{zx}}{\partial z} - \left(\frac{k}{k_\rho}\right)^2 \frac{G_H^{zx}}{jk_y\epsilon_r} \tag{5}$$

More specifically, the function **P** defined in [2] is identically zero. Now consider F_S as

$$F_S \;=\; -jkY\left[\bar{\mathbf{I}} + \frac{1}{k^2}\nabla\nabla\right]\cdot\mathbf{F} \tag{6}$$

Making use of (4) in (6) we can write F_S as

$$F_S \;=\; -jkY\iint_S \bar{\mathbf{G}}\cdot\mathbf{M}(\mathrm{r}')dS' + \frac{jY}{k}\nabla\iint_S \nabla'\cdot\mathbf{M}(\mathrm{r}')G_V(\mathrm{r},\mathrm{r}')dS' \tag{7}$$

The required components of the Green's function in (7) are given by

$$G_{xx} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\tilde{G_{xx}}e^{-jk_2(z-d)}e^{jk_x(x-x')}e^{jk_y(y-y')}dk_x dk_y \tag{8}$$

$$G_{yx} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\tilde{G_{yx}}e^{-jk_2(z-d)}e^{jk_x(x-x')}e^{jk_y(y-y')}dk_x dk_y \tag{9}$$

$$G_{yy} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\tilde{G_{yy}}e^{-jk_2(z-d)}e^{jk_x(x-x')}e^{jk_y(y-y')}dk_x dk_y \tag{10}$$

$$G_{xy} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\tilde{G_{xy}}e^{-jk_2(z-d)}e^{jk_x(x-x')}e^{jk_y(y-y')}dk_x dk_y \tag{11}$$

where

$$\tilde{G_{xx}} = \left(\epsilon_r k_0^2 k_y^2 \Gamma_1 - k_1^2 k_x^2 \Gamma_2\right) / \left(8\pi^2 k_0^2 k_1 \beta^2\right) \tag{12}$$

$$\tilde{G_{yy}} = \left(\epsilon_r k_0^2 k_x^2 \Gamma_1 - k_1^2 k_y^2 \Gamma_2\right) / \left(8\pi^2 k_0^2 k_1 \beta^2\right) \tag{13}$$

$$\begin{aligned}\tilde{G_{xy}} &= \tilde{G_{yx}} \\ &= -k_x k_y\left(k_1^2\Gamma_2 + \epsilon_r k_0^2 \Gamma_1\right) / \left(8\pi^2 k_0^2 k_1 \beta^2\right)\end{aligned} \tag{14}$$

with Γ_1 and Γ_2 given by

$$\Gamma_1 = \frac{\epsilon_r k_2 \sin(k_1 d) - jk_1\cos(k_1 d)}{\epsilon_r k_2\cos(k_1 d) + jk_1 \sin(k_1 d)} \tag{15}$$

$$\Gamma_2 = \frac{k_2\cos(k_1 d) + jk_1\sin(k_1 d)}{k_2\sin(k_1 d) - jk_1\cos(k_1 d)} \tag{16}$$

Also,

$$G_{zx}^E = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\tilde{G_{zx}^E}e^{jk_x(x-x')}e^{jk_y(y-y')}dk_x dk_y \tag{17}$$

$$G^H_{zx} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{G}^H_{zx} e^{jk_x(x-x')} e^{jk_y(y-y')} dk_x dk_y \tag{18}$$

with

$$\tilde{G}^E_{zx} = \frac{1}{4\pi^2}\left[C\cos(k_1 z) + D\sin(k_1 z)\right] \tag{19}$$

where

$$C = -\frac{k_y\left[k_1\cos(k_1 d) + j\epsilon_r k_2 \sin(k_1 d)\right]}{k_1\left[\epsilon_r k_2 \cos(k_1 d) + jk_1 \sin(k_1 d)\right]} \tag{20}$$

$$D = \frac{jk_y}{k_1} \tag{21}$$

and

$$\tilde{G}^H_{zx} = \frac{1}{4\pi^2}\left[E\sin(k_1 z) + F\cos(k_1 z)\right] \tag{22}$$

where

$$E = \frac{k_x\left[k_1\sin(k_1 d) - jk_2\cos(k_1 d)\right]}{k_0 Z_0\left[k_1\cos(k_1 d) + jk_2\sin(k_1 d)\right]} \tag{23}$$

$$F = \frac{k_x}{k_0 Z_0} \tag{24}$$

with $k_1^2 = \epsilon_r k_0^2 - \beta^2$, $k_2^2 = k_0^2 - \beta^2$ and $\beta^2 = k_x^2 + k_y^2$. After weighting (7) becomes

$$\begin{aligned} F_S \;=\; & -jkY \iint_S \mathbf{M}(\mathbf{r}) \cdot \iint_S \bar{\mathbf{G}} \cdot \mathbf{M}(\mathbf{r}') dS\, dS' \\ & + \frac{jY}{k} \iint_S \nabla \cdot \mathbf{M}(\mathbf{r}) \iint_S \nabla' \cdot \mathbf{M}(\mathbf{r}') G_V(\mathbf{r},\mathbf{r}') dS\, dS' \end{aligned} \tag{25}$$

The fields in region 3 (the cavity region) are formulated employing the finite element method [3].

Evaluation of the Sommerfeld integrals using the Chebyshev decomposition method

This is a new, fast and efficient method [4] of evaluating Sommerfeld integrals encountered, when we use a spectral-domain transformation to calculate fields of a dipole source in a homogeneous space or in a layered medium with planar boundaries. The integration is based on the decomposition of the integrand into Chebyshev polynomials, the oscillatory nature of which facilitates rapid integration of the fast oscillating integrands. In (8)-(18) we encounter Fourier transform pairs given by

$$\tilde{f}(k_x, k_y) = \frac{1}{2\pi} \int^{\infty} \int_{-\infty} f(x,y) e^{-jk_x x - jk_y y} \, dx dy \tag{26}$$

$$f(x,y) = \frac{1}{2\pi} \int^{\infty} \int_{-\infty} \tilde{f}(k_x, k_y) e^{jk_x x + jk_y y} \, dk_x dk_y \tag{27}$$

Introducing the Bessel function as

$$J_0(k_\rho \rho) = \frac{1}{2\pi} \int_0^{2\pi} e^{jk_\rho \rho \cos(\alpha - \phi)} d\phi \tag{28}$$

we have

$$\tilde{f}(k_\rho) = \int_0^{\infty} J_0(k_\rho \rho) f(\rho) \rho d\rho \tag{29}$$

$$f\rho = \int_0^{\infty} J_0(k_\rho \rho) \tilde{f}(k_\rho) k_\rho dk_\rho \tag{30}$$

(29) and (30) is the Fourier-Bessel or Hankel integral transform pair. Inverse Hankel tranforms such as (30) are the generic form of Sommerfeld integrals which we consider for the demonstration of this method. Transverse derivatives of $f(\rho)$ can be expressed as :

$$\frac{\partial f}{\partial x} = -\cos\phi \int_0^{\infty} J_1(k_\rho \rho) \tilde{f}(k_\rho) k_\rho^2 dk_\rho \tag{31}$$

Hence the most general Sommerfeld integral is written as

$$SI = \int_0^{\infty} J_m(k_\rho \rho) h(k_\rho) \, dk_\rho \tag{32}$$

where J_m is the Bessel function of the first kind of order m. Assuming that $h(k_\rho)$ has a pole at $k_\rho = k_P$ $(0 < k_P < \sqrt{\epsilon_r} k_0)$

$$\begin{aligned} SI &= \int_0^{\sqrt{\epsilon_r} k_0} (J_m(k_\rho \rho) h(k_\rho) - F(k_\rho)) \, dk_\rho \\ &\quad + \int_0^{\sqrt{\epsilon_r} k_0} F(k_\rho) \, dk_\rho \\ &\quad + \int_{\sqrt{\epsilon_r} k_0}^{\infty} J_m(k_\rho \rho) h(k_\rho) \, dk_\rho \end{aligned} \tag{33}$$

with

$$F(k_\rho) = \frac{A}{k_\rho - k_P} \tag{34}$$

where A is the residue of $J_m(k_\rho \rho)h(k_\rho)$ at $k_\rho = k_P$. The second integral in (33) is solved analytically. Next we make the behaviour of the integrands in (33) smoother by applying a change of variables, for the first and third integrands respectively

$$k_\rho = \sqrt{\epsilon_r} k_0 \sin \xi \tag{35}$$

and

$$k_\rho = \sqrt{\epsilon_r} k_0 \cosh \xi \tag{36}$$

Transforming the domain of integration to the domain of definition of the Chebyshev polynomials the first integral of (33) becomes

$$\frac{\pi}{4} \int_{-1}^{1} \frac{f_1(z_1)}{\sqrt{1-z_1^2}} dz_1 + j\frac{\pi}{4} \int_{-1}^{1} \frac{f_2(z_1)}{\sqrt{1-z_1^2}} dz_1 \tag{37}$$

with

$$f_{1,2}(z) = Re, Im \left((J_m(k_\rho \rho)h(k_\rho) - F(k_\rho))\sqrt{1-k_\rho^2}\sqrt{1-z^2} \right) \tag{38}$$

$$k_\rho = \sqrt{\epsilon_r} k_0 \sin \left(\frac{\pi}{4}(z+1) \right) \tag{39}$$

Similarly transforming the domain of integration for the third integral in (33) and truncating the upper limit at a and b for the real and imaginary parts respectively we have

$$\frac{a}{2} \int_{-1}^{1} \frac{f_3(z_2) f_5(z_2)}{\sqrt{1-z_2^2}} dz_2 + j\frac{b}{2} \int_{-1}^{1} \frac{f_4(z_3) f_6(z_3)}{\sqrt{1-z_3^2}} dz_3 \tag{40}$$

with

$$\begin{aligned} f_{3,4}(z) &= J_m \left(\sqrt{\epsilon_r} k_0 \cosh \left(\frac{a,b}{2}(z+1) \right) \rho \right) \\ &\quad \left(\sqrt{\epsilon_r} k_0 \sinh \left(\frac{a,b}{2}(z+1) \right) \right) \sqrt{1-z^2} \qquad (41) \end{aligned}$$

$$f_{5,6}(z) = Re, Im \left(h \left(\sqrt{\epsilon_r} k_0 \cosh \left(\frac{a,b}{2}(z+1) \right) \right) \right) \tag{42}$$

Now $f_1(z)$ to $f_6(z)$ are fiited with Chebyshev polynomials so that

$$f(z) \approx \left[\sum_{k=1}^{N} c_k T_{k-1}(z)\right] - \frac{1}{2}c_1 \tag{43}$$

with

$$c_j = \frac{2}{N}\sum_{k=1}^{N} f(z_k)T_{j-1}(z_k) \tag{44}$$

The orthogonal property of these polynomials is defined as

$$\int_{-1}^{1} \frac{T_i(z)T_j(z)}{\sqrt{1-z^2}}dz = \begin{cases} 0 & (i \neq j) \\ \frac{\pi}{2} & (i = j \neq 0) \\ \pi & (i = j = 0) \end{cases} \tag{45}$$

Applying the orthogonal property we evaluate the SI as

$$\begin{aligned} SI \approx & \frac{\pi}{2}\frac{\pi}{4}(c_{1,1} + jc_{2,1}) + \frac{\pi}{2}\left(\frac{a}{2}f(c_3, c_5) + j\frac{b}{2}f(c_4, c_6)\right) \\ & + A\ln\left(\frac{\sqrt{\epsilon_r}k_0 - k_P}{k_P}\right) - j\pi A \end{aligned} \tag{46}$$

with

$$f(c_i, c_j) = \sum_{k=0}^{min(N_i,N_j)-1} \left(\frac{2-\delta(k)}{2}\right) c_{i,k+1}c_{j,k+1} \tag{47}$$

Using the Chebyshev decomposition method, a typical Sommerfeld integral encountered while analysing a horizontal electric dipole in the x direction above a ground with relative pemittivity ϵ_r and conductivity σ, is evaluated. The integral is expressed as

$$A_x = 2\int_0^{\infty} J_0(k_\rho \rho)\left(\frac{e^{-u_0|z+h|}}{u_0 + u}\right) k_\rho dk_\rho \tag{48}$$

where

$$\rho = \sqrt{x^2 + y^2} \quad u_0 = \sqrt{k_\rho^2 - k^2} \tag{49}$$

$$u = \sqrt{k_\rho^2 - N^2k^2} \quad N^2 = \epsilon_r - j\frac{\sigma}{\omega\epsilon_0} \tag{50}$$

and k is the free space propagation constant. For this evaluation z (the observer height) to be equal to the dipole height h ($\lambda/50$). The results of this evaluation is shown in Fig 2.

References

[1] T. Itoh, ed., Numerical Techniques for Microwave and Millimeter- wave Passive Structures, Wiley-Interscience, 1989.

[2] K. A. Michlaski, "Electromagnetic Scattering and Radiation by Surfaces of Arbitary Shape in Layered Media, Part 1: Theory," *IEEE Trans. Antennas Propagat.*, vol.38, no. 3, pp.335-344, March 1990.

[3] J. M. Jin and J. L. Volakis, "A Finite Element-Boundary Integral Formulation for Scattering by Three-Dimensional Cavity-Backed Apertures, "*IEEE Trans. Antennas Propagat.*, vol.39, no.1, pp.97-104, Jan 1991.

[4] G. Annaert, "Evaluation of Sommerfeld Integrals using Chebyshev Decomposition," *IEEE Trans. Antennas Propagat.*, vol.41, no. 2, pp.159-164, Feb. 1993.

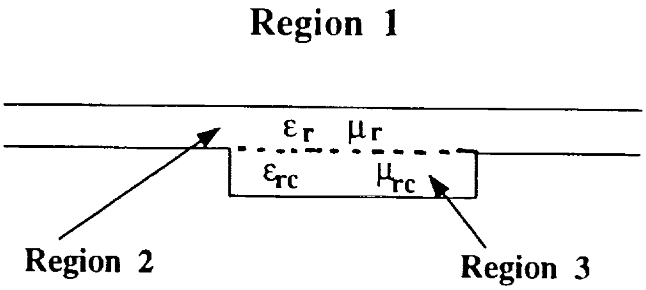


Figure 1: Geometry of a groove in a coated plane

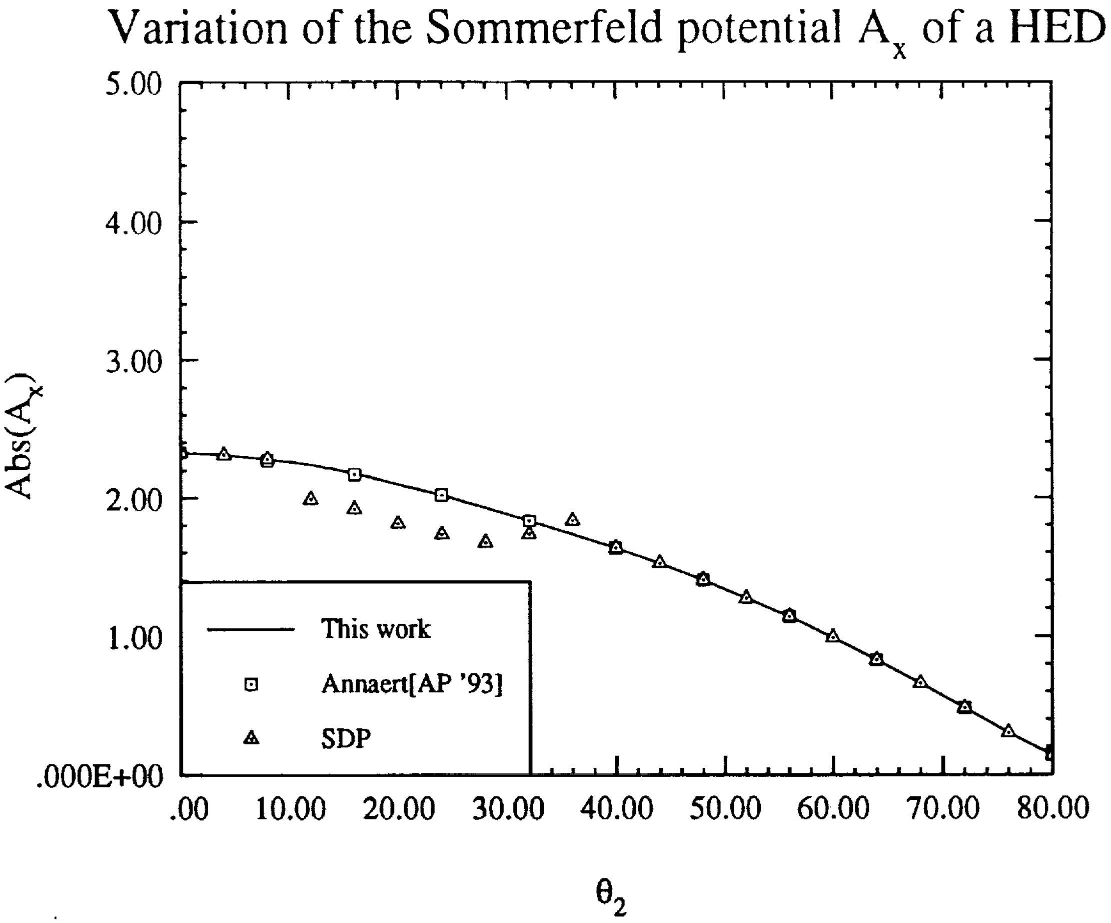


Figure 2: A typical Sommerfeld integral evaluation