User Manual for FEMA-CYLA: A FE-ABC Numerical Program for Cavity Backed Antennas in a Circular Cylinder

L.C. Kempel and J.L. Volakis

Radiation Laboratory
University of Michigan
1301 Beal Avenue
Ann Arbor, MI 48109-2122

October 7, 1994

Contents

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. FEMA-CYL is a computer program written in the Radiation Laboratory of the University of Michigan which implements this formulation. Both of these codes exploit a BiCG-FFT linear system solver to achieve impressive economies in terms of memory and computational requirements. Although both CAVITY3D and FEMA-CYL have proven quite capable for determining the radiation, input impedance and scattering properties of large antenna arrays, the well-known restrictions of an BICG-FFT implementation of a FE-BI code prompted the investigation of new mesh truncation schemes. The result is FEMA-CYLA which uses an absorbing boundary condition (ABC) for mesh closure.

This user manual will give a brief introduction to FEMA-CYLA 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

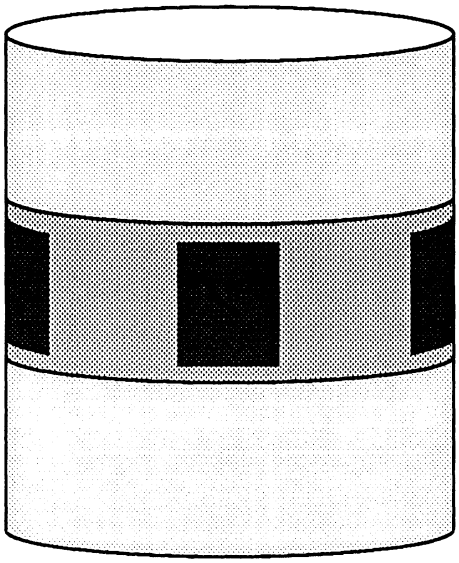
FEMA-CYLA uses a FE-ABC formulation to determine the secondary electric field in the region outside the cylinder and the total electric field within the underlying cavity region. The underlying formulation is given in detail in a previous Radiation Laboratory Technical Report [1]. FEMA-CYLA is an offshoot of FEMA-CYL and accordingly there will be strong similarities between the two codes. One such similarity involves the use of uniform zoning in the geometry discretization. Although uniform zoning is not required for this ABC implementation, its use simplified the development of the custom mesh generator provided with the package.

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. 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 FEMA-CYLA is quite flexible.

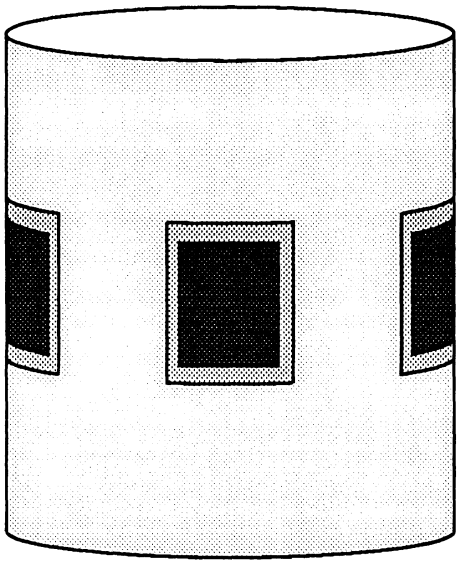
3 Compilation

The first task in utilizing FEMA-CYLA is to compile and link the various files of the program. The following files are required to run FEMA-CYLA and are given on the distribution diskette: *fema-cyla.f*, *integrate.f*, *matrixGenerate.f*, *preProcessor.f*, *rcs.f*, *coat.f*, *gauss.inc*, *hankel.inc* and *fema-cyla.inc*. These files perform the following functions:

- *fema-cyla.f*: Main program, BiCG and CGS solver, matrix building subroutines, FE-ABC subroutine, impedance insert and various auxiliary subroutines.
- *integrate.f*: Provide numerical integration codes, far-field Fock functions and gamma function.
- *matrixGenerate.f*: Finite element and ABC matrix terms.
- *preProcessor.f*: Geometry/mesh generator.
- *rcs.f*: Radar Cross Section, far-zone dyadic Green's function and plane wave excitation functions.
- *coat.f*: Generalized scattering codes which allow the inclusion of coatings.



(a)



(b)

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

- *gauss.inc*: Numerical integration parameters for gaussian quadrature.
- *hankel.inc*: Include file for Hankel function calculation.
- *fema-cyla.inc*: Main memory allocation file also contains variable dictionary.

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, FEMA-CYLA 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*. FEMA-CYLA is constructed by simply typing *make* at the command line, while the binary-to-ASCII conversion program is constructed by entering *make convert*. FEMA-CYLA is invoked by typing *fema-cyla* 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-ABC 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

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

FEMA-CYLA 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 the computational domain (cavity and superstrate). 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 = 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.

```
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.

- 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, ABC 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 FEMA-CYLA uses a total field formulation within the cavity, 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 FEMA-CYLA.

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.
- 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.

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.

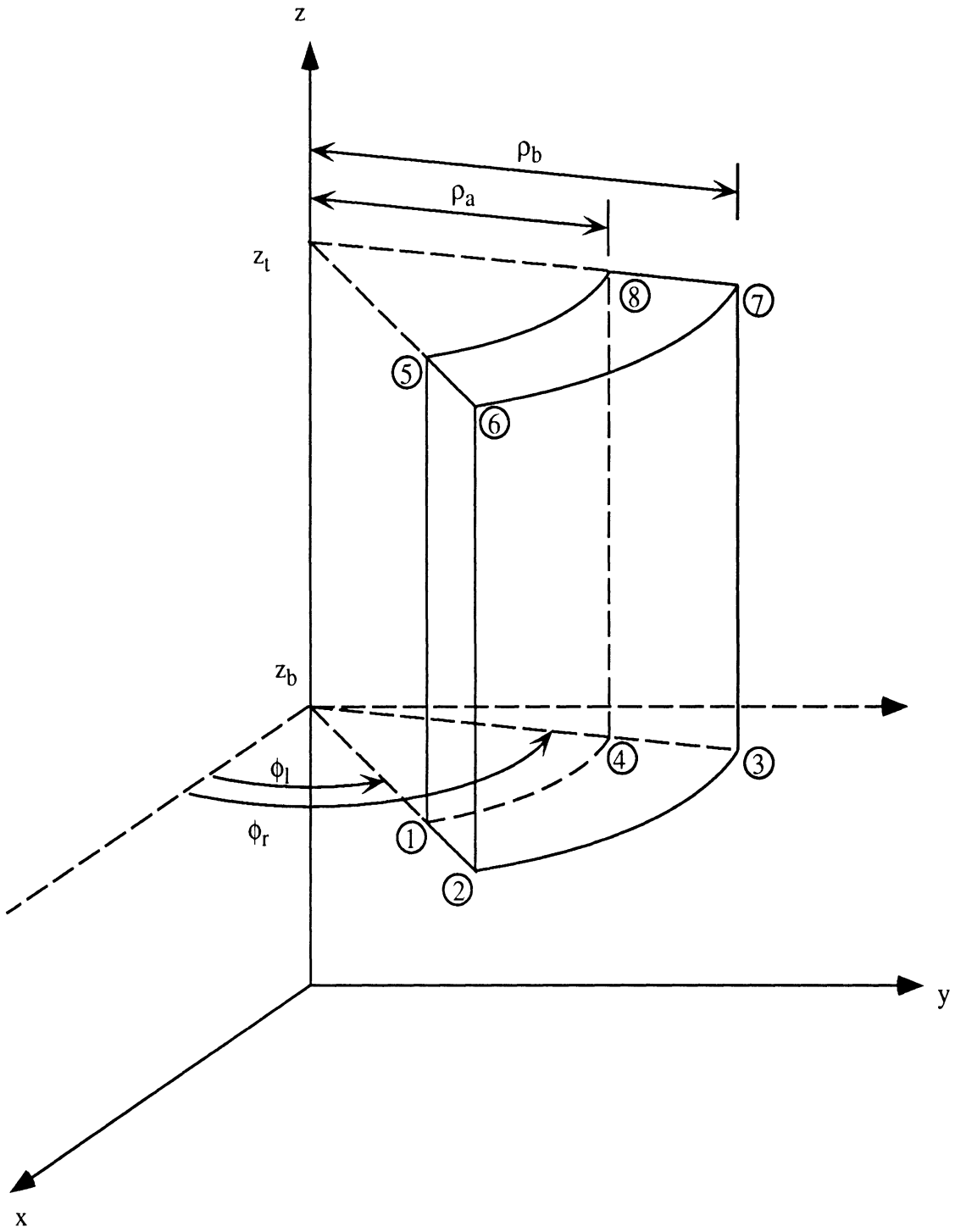


Figure 6: Cylindrical shell element.

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.

- 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 $\times$ 3 cm patch antenna which is placed in a 5 cm $\times$ 6 cm cavity. This antenna may either be placed in a discrete cavity or upon a wraparound substrate. Users will find it very helpful to check the entered geometry via *convertToASCII* prior to running the solver part of FEMA-CYLA. 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.

FEMA-CYLA 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 FEMA-CYLA by replacing the subroutine *simpleMesh* with a universal file reader. However, we have found that the custom mesh routine provided with FEMA-CYLA (*simpleMesh*) is sufficient for most modeling tasks while being quite efficient. For this manual, we will use this package for all meshing requirements.

FEMA-CYLA differs from FEMA-CYL primarily in its ability to include material coatings or protrusions outside the cylinder. Thus, FEMA-CYLA requires the specification of the superstrate materials. As is the case with the substrate, the superstrate may be specified as homogeneous, layered or inhomogeneous with individual element parameters (either by keypad input or by an existing ASCII file). In addition, since an ABC is used for mesh truncation, the user must specify the distance between the antenna and the

```
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.

ABC surface. Typically, at least 0.3λ is required for accurate RCS and radiation pattern calculations.

Consider a 2 cm $\times$ 3 cm patch antenna recessed in a 5 cm $\times$ 6 cm cavity which is centered at ($\phi = 0^o$, 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 ABC.
- Enter number of nodes per cavity.
- Enter lower-left corner of cavity in surface grid coordinates.
- Enter number superstrate layers.
- Enter number substrate layers.
- 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).
- For each layer, enter its thickness in centimeters (layers in reverse order for superstrate).
- Enter 1 to save this geometry.
- Enter filename for this geometry.

As mentioned previously, FEMA-CYLA also allows for a continuous wraparound array. A wraparound discretization of the exterior region may be required to properly model the physics associated with a superstrate. Note for continuous wraparound cavities, if a patch crosses the branch cut ($\phi = \pm 180^{o}$), 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.

An example geometry build is given in file *example.build* which is included in FEMA-CYLA's distribution disk. Note that in this example, a thin superstrate layer is placed just above the surface of the cylinder. It has been observed that such a layer is critical for accurate input impedance calculation especially around resonance. Evidently, the rapid near-field variation just above the surface of the radiators requires such fine sampling.

6 Operation Modes

FEMA-CYLA has three main operation modes for FE-ABC 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. This option is very similar to the second option (see Sec. 6.2); however, it does not compute the far-zone field which can be very time consuming. The following information is required for input impedance calculations

- Choose item 2 (FE-ABC) from main menu.
- Enter the stored binary geometry file.
- Indicate if this geometry is quasi-planar.
- Enter 1 if all superstrate elements have the same material parameters.
- Enter 1 if all substrate elements have the same material parameters.

- Enter complex permittivity.
- Enter complex permeability.
- Enter the layer number of the radiation integral.
- 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 solver type (symmetric or asymmetric matrix).
- If asymmetric, partially or fully (see [1]).
- 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 feed in degrees and centimeters.
- Enter layer which contains the feed.
- Enter driving point current (mag,phase).
- Enter number of impedance post loads.
- Enter frequency range (in GHz) for this sweep.
- Return to main menu.

The output file generated by this mode contains the following information:

- Column 1: Frequency (GHz).
- Column 2: Driving point resistance (Ω).
- Column 3: Driving point reactance (Ω).
- Column 4: Reflection coefficient at driving point (Γ).
- Column 5: Phase of the reflection coefficient.

An example input file for this mode of operation is given in *input.Zin* which is included on the distribution disk. The input impedance for 3.0 GHz to 3.2 GHz computed every 5 MHz is shown in figure 9 and compared with data generated using the FE-BI formulation (FEMA-CYL).

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-ABC) from main menu.
- Enter the stored binary geometry file.
- Indicate if this geometry is quasi-planar.
- Enter 1 if all superstrate elements have the same material parameters.
- Enter 1 if all substrate elements have the same material parameters.
- Enter complex permittivity.
- Enter complex permeability.
- Enter the layer number of the radiation integral.
- 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 solver type (symmetric or asymmetric matrix).
- If asymmetric, partially or fully (see [1]).
- 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 (monostatic, bistatic or radiation).
- Enter start, stop and increment azimuth (ϕ) angles (in degrees).
- Enter start, stop and increment elevation (θ) angles (in degrees).
- If scattering, enter polarization angle (0 = E-pol, 90 = H-pol).
- Enter RCS filename.
- Enter number of probe feeds.

- Enter location of feed in degrees and centimeters.
- Enter layer which contains the feed.
- Enter driving point current (mag,phase).
- Enter number of impedance post loads.
- Enter frequency range (in GHz) for this sweep.
- Return to main menu.

The far-zone field output file generated by this mode contains the following information:

- Column 1: Frequency (GHz).
- Column 2: Observation angle (θ) in degrees.
- Column 3: Observation angle (ϕ) in degrees.
- Column 4: Radiation pattern due to E_θ.
- Column 5: Radiation pattern due to E_ϕ.
- Column 6: Radiation pattern due to E_θ plus E_ϕ.

The radiation pattern at normal observation for 3.0 GHz to 3.2 GHz computed every 5 MHz is shown in figure 10 and the associated inputs are given in *input.Fsw* on the distribution disk.

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 required inputs are

- Choose item 2 (FE-ABC) from main menu.
- Enter the stored binary geometry file.
- Indicate if this geometry is quasi-planar.
- Enter 1 if all superstrate elements have the same material parameters.
- Enter 1 if all substrate elements have the same material parameters.
- Enter complex permittivity.
- Enter complex permeability.
- Enter the layer number of the radiation integral.

- 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 solver type (symmetric or asymmetric matrix).
- If asymmetric, partially or fully (see [1]).
- Enter 0 for monotone computation of the far-zone fields.
- Enter observation type (monostatic, bistatic or radiation).
- Enter start, stop and increment azimuth (ϕ) angles (in degrees).
- Enter start, stop and increment elevation (θ) angles (in degrees).
- If scattering, enter polarization angle (0 = E-pol, 90 = H-pol).
- Enter RCS/radiation pattern filename.
- If radiation, enter normalized pattern filename.
- Enter frequency of operation.
- Enter number of probe feeds.
- Enter location of feed in degrees and centimeters.
- Enter layer which contains the feed.
- Enter driving point current (mag,phase).
- Enter number of impedance post loads.
- Indicate whether additional observation cut is desired.
- Return to main menu.

The far-zone field output file generated by this mode contains the following information:

- Column 1: Polarization angle in degrees (if used).
- Column 2: Incidence angle (θ_i) in degrees (if used).
- Column 3: Incidence angle (ϕ_i) in degrees (if used).
- Column 4: Observation angle (θ) in degrees.
- Column 5: Observation angle (ϕ) in degrees.
- Column 6: Radar Cross Section due to E_θ.
- Column 7: Radar Cross Section due to E_ϕ.
- Column 8: Radar Cross Section due to E_θ plus E_ϕ.

Figure 11 illustrates the comparison between the FE-ABC method and the FE-BI formulation for the example antenna operated at resonance (3.11 GHz). The required inputs for this example are given in *input.Monotone* on the distribution disk.

7 Concluding Remarks

This user manual presented some basic operation information for the FE-ABC code, FEMA-CYLA. 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 FEMA-CYLA. For example, currently FEMA-CYLA 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 *fema-cyla.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 FEMA-CYLA 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 *fema-cyla.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 *fema-cyla.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] Leo C. Kempel and John L. Volakis, "Radiation and scattering fro cylindrically conformal printed antennas," University of Michigan Radiation Laboratory Technical Report, 031173-2-T, April 1994.

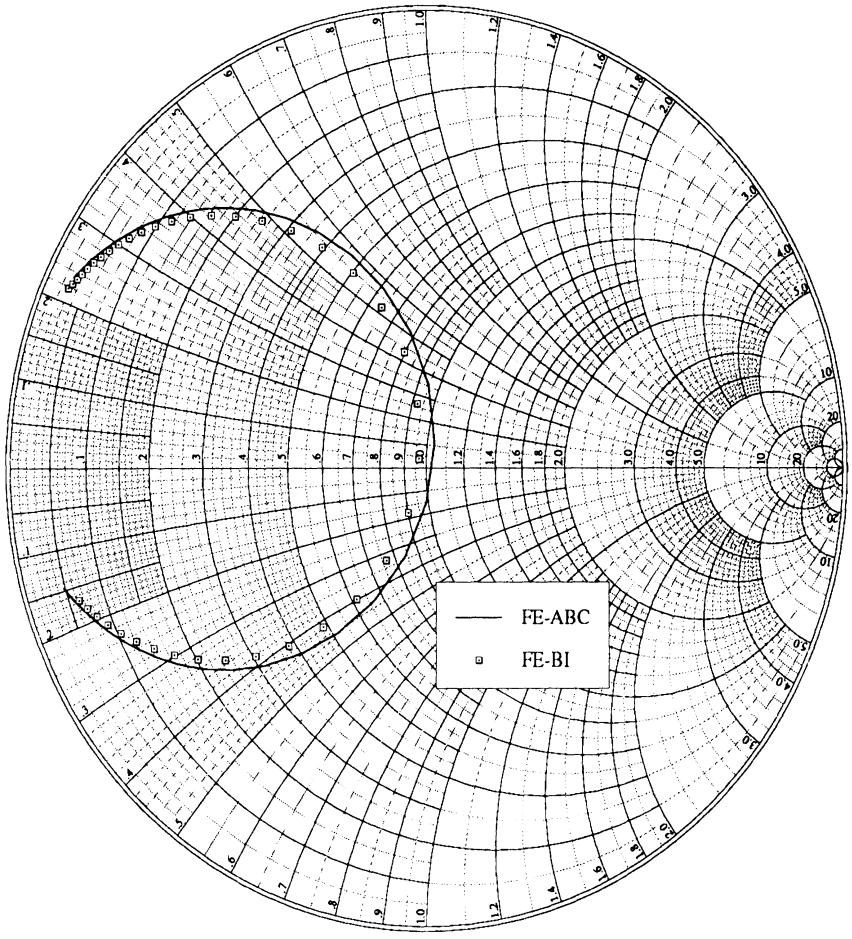


Figure 9: Input impedance for the axially polarized patch antenna which is 2 cm $\times$ 3 cm in a 5 cm $\times$ 6 cm cavity. The frequency range is 3.0 to 3.2 GHz with data taken every 5 MHz.

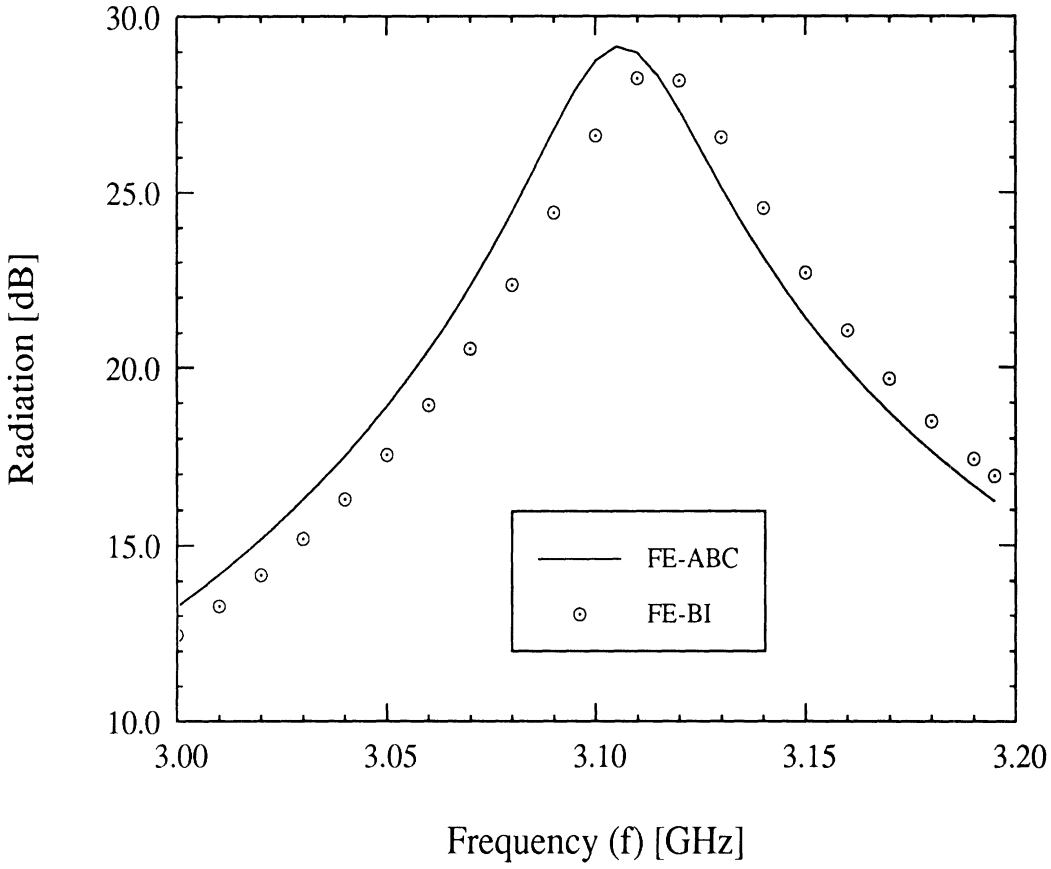


Figure 10: Radiation pattern vs. frequency for a patch antenna which is 2 cm $\times$ 3 cm in a 5 cm $\times$ 6 cm cavity.

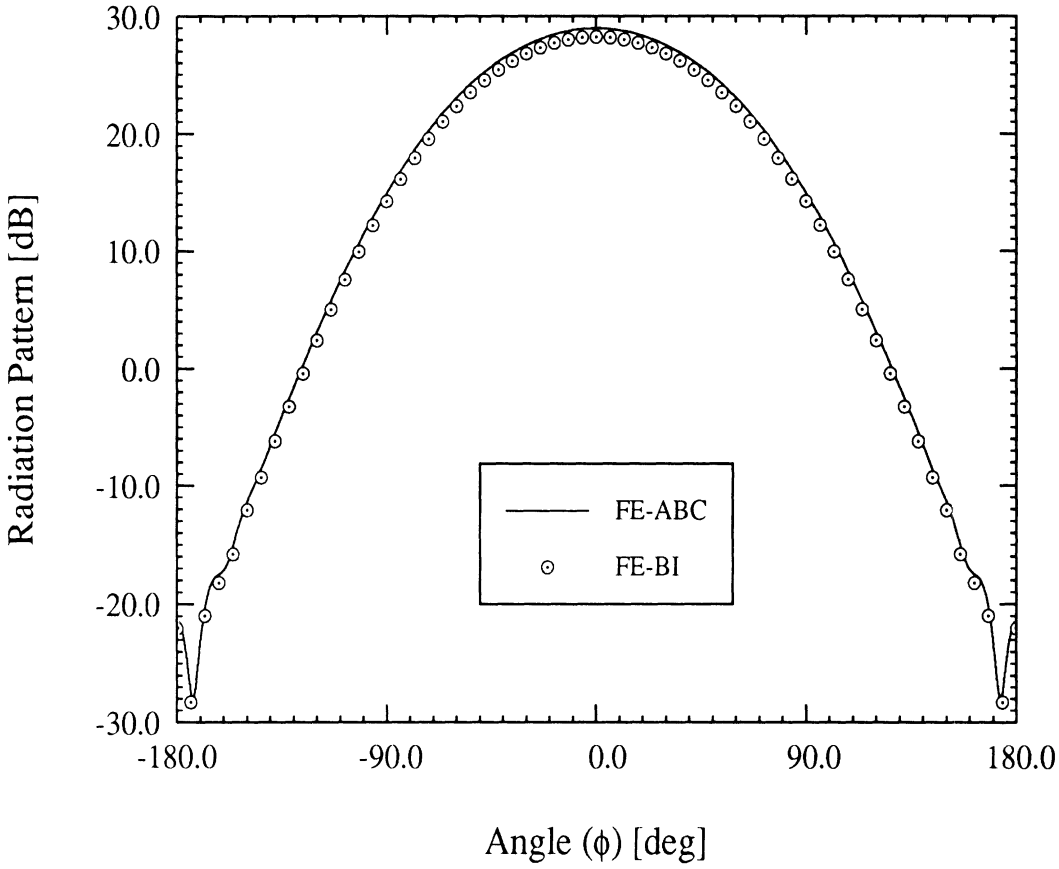


Figure 11: Radiation pattern for a patch antenna which is 2 cm $\times$ 3 cm in a 5 cm $\times$ 6 cm cavity operated at 3.11 GHz.

```
#######################################################
#C-SHELL
#
# Makefile for FEMA_CYLA
#
# Leo C. Kempel
# University of Michigan
# Radiation Laboratory
# Modification Date:  29 October 1993
#
#
# NOTE:  Modify these variables as needed for your system
#
# BINDIR= directory where executable file should
#         be placed
# BINS=   list of object files to be linked to main
# FF= Fortran compiler name
# FOPT= Fortran compilation options
# LOPT= Fortran linker options
#######################################################
#######################################################
#             Fema_cyla                               #
BINS= rcs.o integrate.o matrixGenerate.o preProcessor.o coat.o
BINDIR=
#######################################################
#Sun-4     ###############
#FF= f77
#FOPT=
#LOPT=
#######################################################
#DEC       ###############
#FF= f77
#FOPT= -O
#LOPT=
#######################################################
#HP 9000/7XX #############
#FF= f77
#FOPT= -O
#LOPT=
#######################################################
#IBM RS/6000 #############
#FF=xlf
#FOPT= -O
#LOPT=
#######################################################
#CRAY        #############
#FF=cf77
#FOPT= -Wf"-dp"
#LOPT=
#######################################################
#CONVEX      #############
#FF= fc
#FOPT= -or none -O2
#LOPT= -lveclib
#######################################################
#Iris        ###############
FF= f77
FOPT=
LOPT=
#######################################################
#######################################################
# Now build Fema_cyla....

fema_cyla: fema_cyla.o $(BINS);\
      $(FF) $(FOPT) fema_cyla.o $(BINS) -o $(BINDIR)fema_cyla $(LOPT)
```

```
rcs.o:  rcs.f;\
      $(FF) $(FOPT) -c rcs.f

integrate.o:  integrate.f;\
      $(FF) $(FOPT) -c integrate.f

matrixGenerate.o:  matrixGenerate.f;\
      $(FF) $(FOPT) -c matrixGenerate.f

preProcessor.o:  preProcessor.f;\
      $(FF) $(FOPT) -c preProcessor.f

fema_cyla.o: fema_cyla.f;\
      $(FF) $(FOPT) -c fema_cyla.f

coat.o: coat.f;\
      $(FF) $(FOPT) -c coat.f

clean:
        rm -f $(BINS) fema_cyla.o fema_cyla
```