036307-5-T

USER MANUAL FOR FEMTETRA.F: A TETRAHEDRAL FINITE ELEMENT-BOUNDARY INTEGRAL (FE-BI) CODE

Arik Brown
Zhifang Li
John L. Volakis

Mission Research Corp.
147 John Sims Parkway
Valparaiso, FL. 32580

Air Force Office of Scientific Research
AFOSR/NM
Bolling AFB, DC 20332-0001

December 1998

36307-5-T = RL-2500

PROJECT INFORMATION

PROJECT TITLE: Focused Application Software for Design of Ferrite Patch Antennas

REPORT TITLE: USER MANUAL FOR **FEMTETRA.F**: A TETRAHEDRAL FINITE ELEMENT-BOUNDARY INTEGRAL (FE-BI) CODE

U-M REPORT No.: 036307-5-T

CONTRACT
START DATE: July 1997
END DATE: June 1999

DATE: December 1998

SPONSOR:

S. McCool and K. Trott
Mission Research Corp.
147 John Sims Parkway
Valparaiso, FL. 32580
trott@fwb.gulf.net
Phone: 904-729-4494
Fax: 904-729-2443

Dr. Arje Nachman
Air Force Office of Scientific Research
AFOSR/NM
Bolling AFB, DC 20332-0001
nachman@afosr.af.mil

SPONSOR
CONTRACT No.: SC-0022-97-001
(Air Force Prime Contract no. F49620-97-C-0022)

U-M PRINCIPAL
INVESTIGATOR:

John L. Volakis
EECS Dept.
University of Michigan
1301 Beal Ave
Ann Arbor, MI 48109-2122
Phone: (313) 764-0500 FAX: (313) 747-2106
volakis@umich.edu
http://www-personal.engin.umich.edu/~volakis/

CONTRIBUTORS
TO THIS REPORT: Arik Brown, Zhifang Li and J.L. Volakis

User Manual for *femtetra.f*: A Tetrahedral Finite Element-Boundary Integral(FE-BI) Code

Arik D. Brown
Zhifang Li
John L. Volakis

University of Michigan Radiation Laboratory
Department of Electrical Engineering and Computer Science
1301 Beal Ave.
Ann Arbor, MI 48109-2122

December 15, 1998

Contents

List of Figures

1 Introduction

1.1 Code Description

The program *femtetra.f* is written in Fortran77 and has been verified on Hewlett Packard(HP) and Sun workstations. The code has an internal preprocessor that uses a Universal file mesh(.unv) for input. The mesh can be generated using Ideas. The University of Michigan has developed an automatic mesher which can be used in conjunction with *femtetra.f*. The mesher allows the user to bypass generating a mesh manually and facilitates a simple and convenient interface between the mesher and *femtetra.f*.

femtetra.f is used to analyze printed antennas on dielectric or anisotropic substrates. The code employs a hybrid Finite Element Method - Boundary Integral(FE-BI) approach using an Electric Field Integral Equation(EFIE) formulation. The resulting FE-BI system of equations is solved iteratively using a Bi-Conjugate Gradient(BiCG) solver. An additional feature of the code is the Adaptive Integral Method(AIM), which is a fast integral method used specifically in the BI portion of the formulation for improved speed.

Figure 1 gives an overview of the steps needed to run *femtetra.f* in conjunction with the automatic mesher. First, *meshrect.f* (rectangular geometries) or *meshcirc.f* (circular geometries) is run, to produce a Universal file. This Universal file is then given as input to *femtetra.f* which processes the file, and produces the necessary tetrahedral information and dimensioning information for the code. The code is then compiled using a **make** file. After compiling is completed, *femtetra.f* is then used to compute either scattering or radiation data.

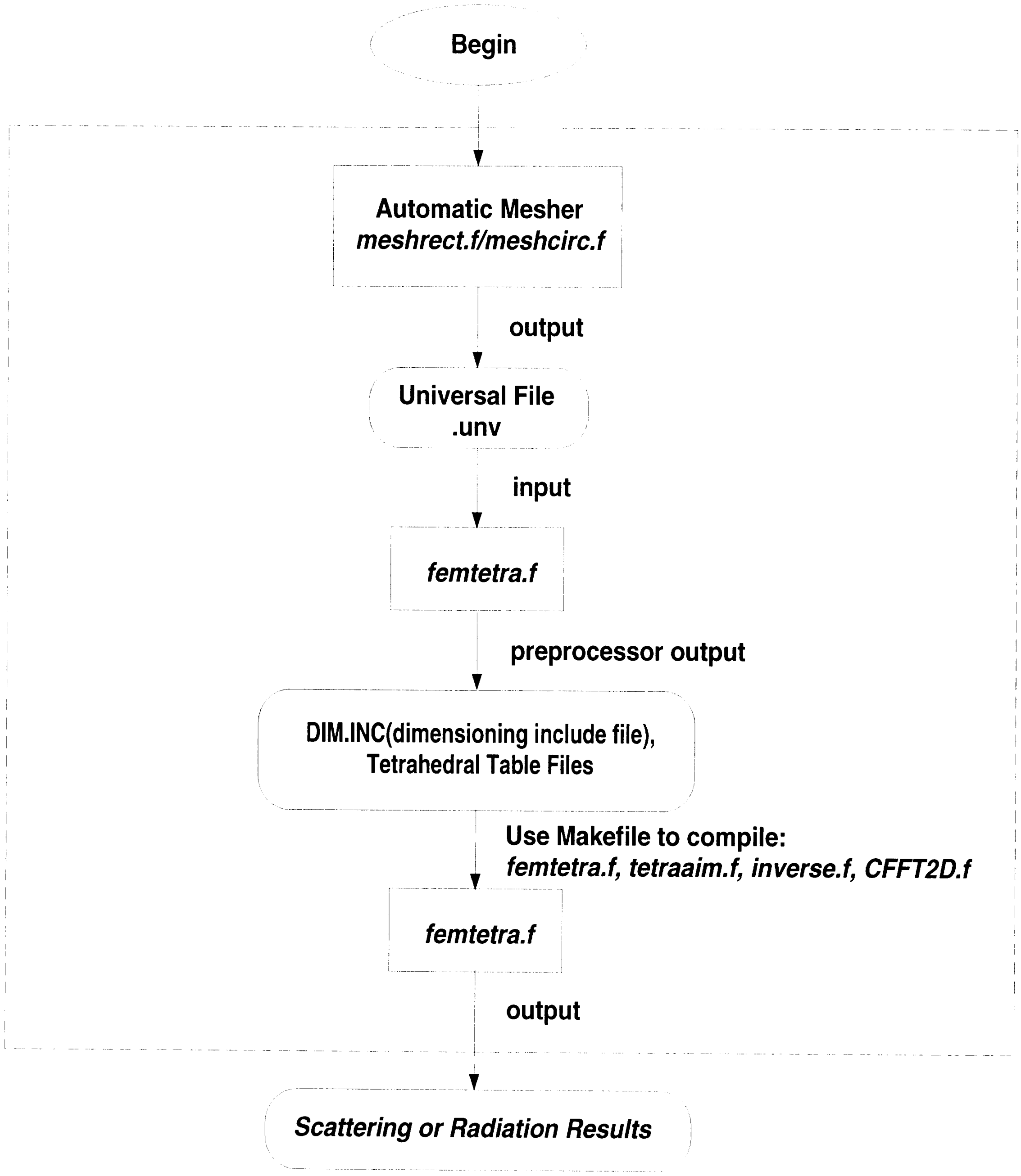

Figure 1: Diagram of the procedure for running *femtetra.f* in conjunction with the automatic mesher.

```
0
1
.4166667 .425
3 2
.4166667 .425
12 8
1
1
.08779
```

Figure 2: Input file for *meshrect.f.*

2 Mesher

2.1 Running the Automatic Mesher

A Universal file is needed to run *femtetra.f.* The Universal file can be generated using the automatic mesher. Depending on whether the problem geometry is rectangular or circular, *meshrect.f* or *meshcirc.f* is the file needed for mesh generation.

2.1.1 Rectangular Meshes

Figures 2 and 3 show a sample input file for *meshrect.f*, and the corresponding surface mesh. An explanation of each line of the sample file, *inputrect*, is shown in Figure 4.

2.1.2 Circular Meshes

Figures 5 and 6 show a sample input file for *meshcirc.f*, and the corresponding surface mesh. The description of the file *inputcirc* is similar in structure to *inputrect* and will not be discussed.

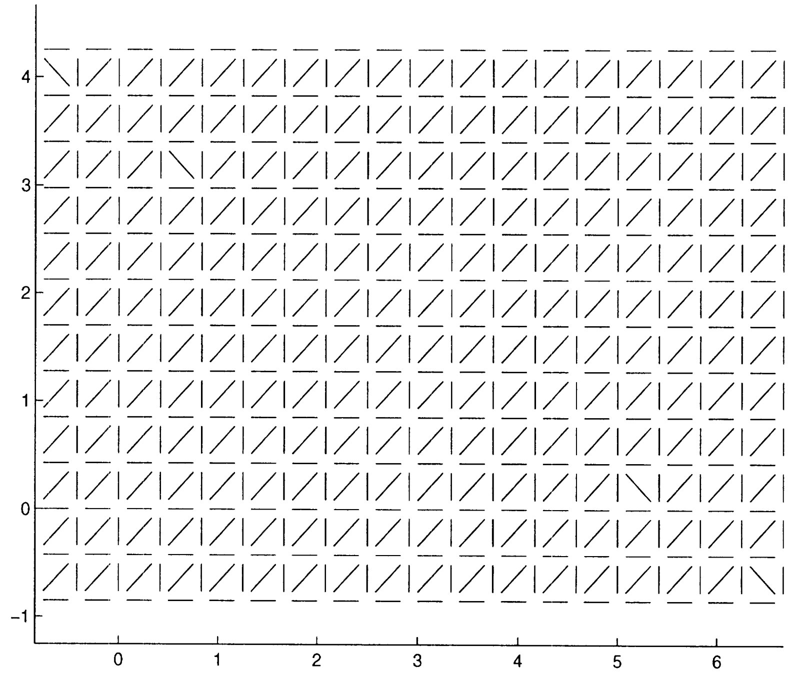

Figure 3: Surface mesh for the sample input file in Figure 2.

0;	line1, Specifies whether surface nodes are all substrate(1=yes, 0=no).
1;	line2, Specifies whether patch or slot(1=patch, 0=slot).
.4166667 .425;	line3, Specifies the **element size**(cm) in the x and y directions, respectively, between the *antenna* and *cavity wall.*
3 2;	line4, Specifies the **number of elements** between the *antenna* and *cavity wall* in the x and y directions, respectively.
.4166667 .425;	line5, Specifies the **element size**(cm) in the x and y directions, of the *antenna* only.
12 8;	line6, Specifies the **number of antenna elements** in the x and y directions, respectively.
1;	line7, Specifies element type(1=tetras, 0=prisms).
1;	line8, Specifies the number of substrate layers.
.08779;	line9, Specifies the height of the substrate layer.

Figure 4: Description of input file in Figure 2.

```
0
1
.0825
8
.082
5
1
1
.0508
```

Figure 5: Input file for *meshcirc.f.*

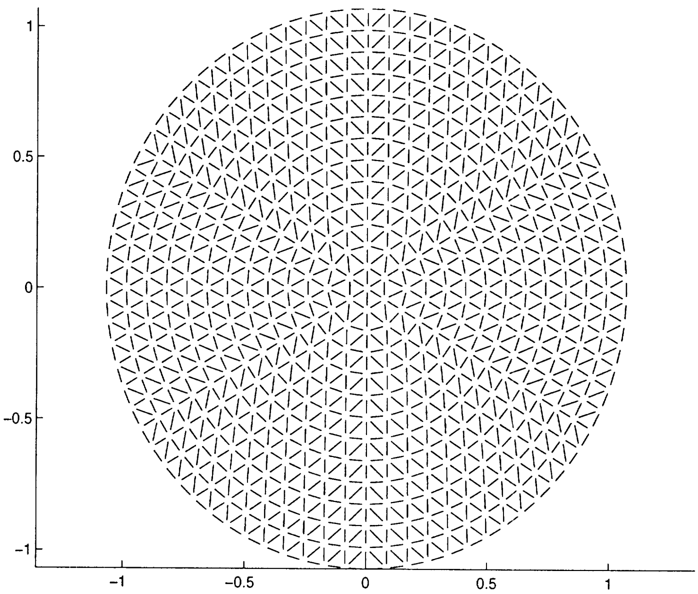

Figure 6: Surface mesh for the sample input file in Figure 5.

3 Running *femtetra.f*

3.1 Preprocessor

Before running *femtetra.f*, it is necessary to process the Universal input file generated by the automatic mesher. When running the code initially, the user will be prompted with 3 menu options:

1. Run Preprocessor
2. Run Tetrahedral FEM-BI Code
3. Exit Code

In order to process the mesh file, the user must choose option '1'. The user will then be prompted with a series of questions about the mesh geometry. Since the preprocessor prompts the user at each entry, the user can easily apply the following example to other mesh geometries(i.e. open cavity, circular patch, circular cavity, etc...) with a clear understanding of how the preprocessor works. Figure 7 shows a description of the input needed for the preprocessor. For this sample input file, it is assumed that the cavity is rectangular and that there is one pec patch on the aperture, and one substrate layer.

3.1.1 Additional Key Information on Running the Preprocessor

There are several important things the user should be cognizant of when running the preprocessor in *femtetra.f*. Initially, the user should have a dummy *DIM.INC* file in order to compile the code for the first time. See[1] for the structure of *DIM.INC*. In actuality, the user will be supplied with several demonstration runs which will include *DIM.INC* files that can be used to compile *femtetra.f* for the first time. In addition, if the user is modeling a ferrite material, the user should enter $(1.0, 0.0)$ for the value of μ_r.

When running the preprocessor for a rectangular patch geometry, the user will need to **first** ascertain the maximum/minimum x-y coordinates of the patch, using either Ideas or the mesh viewer in Matlab, which is supplied with the automatic mesher. This is because the automatic mesher does not center the patch at the coordinates (0.0,0.0), thus the values for $xmax_{patch}$, $xmin_{patch}$, $ymax_{patch}$, and $ymin_{patch}$ can not be predetermined. This is not necessary for circular patches because the automatic mesher centers the patch at the coordinates (0.0,0.0)

meshinput.unv(user supplied)	line1,	Specify input mesh file name.
1	line2,	Specify if there is a patch (1=yes, 0=no).
1	line3,	Specify cavity shape (1=rectangular, 0=circular).
1	line4,	Specify number of pec patches.
$xmax_{patch}$	line4,	Specify maximum x coordinate for the patch.
$xmin_{patch}$	line5,	Specify minimum x coordinate for the patch.
$ymax_{patch}$	line4,	Specify maximum y coordinate for the patch.
$ymin_{patch}$	line5,	Specify minimum y coordinate for the patch.
1	line6,	Specify number of cavity layers.
$(\epsilon_{r_{real}},\epsilon_{r_{imag}})$ $(\mu_{r_{real}},\mu_{r_{imag}})$	line7,	Specify ϵ_r and μ_r for each layer (in this case there is one substrate layer).

Figure 7: Sample preprocessor file.

3.2 Running *femtetra.f*

There are 3 files separate from *femtetra.f* which are needed in order to run the code. The files are:

- *tetraaim.f*
- *inverse.f*
- *CFFT2D.f*

These files are included with the code. In order to compile *femtetra.f* and it's corresponding files, a **make** file is also included with the code. The name of this file is *Makefile*[1].

After compiling *femtetra.f*, the user will then be able to run the code. When running the code, the user will once again be prompted with 3 menu options:

1. Run Preprocessor
2. Run Tetrahedral FEM-BI Code

[1]When compiling the code on a Sun workstation, the options **+E1** and **+U77** should be omitted.

3. Exit Code

The user should choose option '2' to generate the particular scattering or radiation information needed. Similar to the automatic mesher, and the preprocessor, when running *femtetra.f* the user will be prompted for the information needed for each line of input. In the next section, several demonstration examples are shown giving the user a chance to become familiar with the code.

3.2.1 Additonal Information for Running *femtetra.f*

Running *femtetra.f* is straightforward, although an additional comment will be made regarding geometries that include a probe-fed patch. When running *femtetra.f* for a problem with a probe feed, the user will be prompted to specify the global node numbers for the corresponding probe edges within the FEM volume. The global node numbers can be found in two ways. The first approach is to view the mesh file in Ideas before running the code, and locate the node numbers for the probe feed edges. The second and easier approach is to view the surface mesh using Matlab via the automatic mesher. See[1] for more information on using the mesh viewing option in the automatic mesher package. The mesh viewer displays the global node numbers for the surface mesh on the **bottom** of the cavity. To obtain the global node number for the node directly above the corresponding node in the surface mesh, simply add the number of aperture nodes to the corresponding node number. For example, if the bottom of the probe has the global node number 10, and there are 250 aperture nodes, then the node directly above node 10 would be global node number 260. In this manner the global node numbers for the probe edge(s) can be found easily.

4 Demonstration Examples

This section contains three different demonstration examples for *femtetra.f.* The examples will consist of the following:

- Single layer rectangular cavity with a rectangular patch, input impedance(Z_{in}).
- Single layer circular cavity with a circular patch, backscatter.
- Single layer rectangular ferrite cavity with a rectangular patch, backscatter.

Each individual example will include the following items:

- Input file for the automatic mesher.
- Surface mesh figure.
- Input file for the preprocessor in *femtetra.f*.
- Input file for running *femtetra.f*.
- Output data.

These examples allow the user to become familiar with the procedure for running *femtetra.f.*

4.1 Rectangular Patch: Z_{in}

The rectangular patch shown in Figure 8 can be found in [2]. It consists of a rectangular patch residing in a cavity, with a single probe feed. The patch size is 5.0 cm x 3.4 cm and the cavity size is 7.5 cm x 5.1 cm. The substrate has a thickness of 0.08779 cm, $\epsilon_r = 2.17$, and a loss tangent of 0.0015. The probe is located at the coordinates (x_p = 1.22cm, y_p = 0.85cm).

4.1.1 Mesh Generation

In order to generate the Z_{in} for this geometry, the mesh must be generated. To accomplish this using the automatic mesher, the appropriate file to use is *meshrect.f.* The corresponding input file *inputmeshZin* is shown on page 13 with a description for each line of input. The surface mesh generated by the automatic mesher is shown in Figure 10.

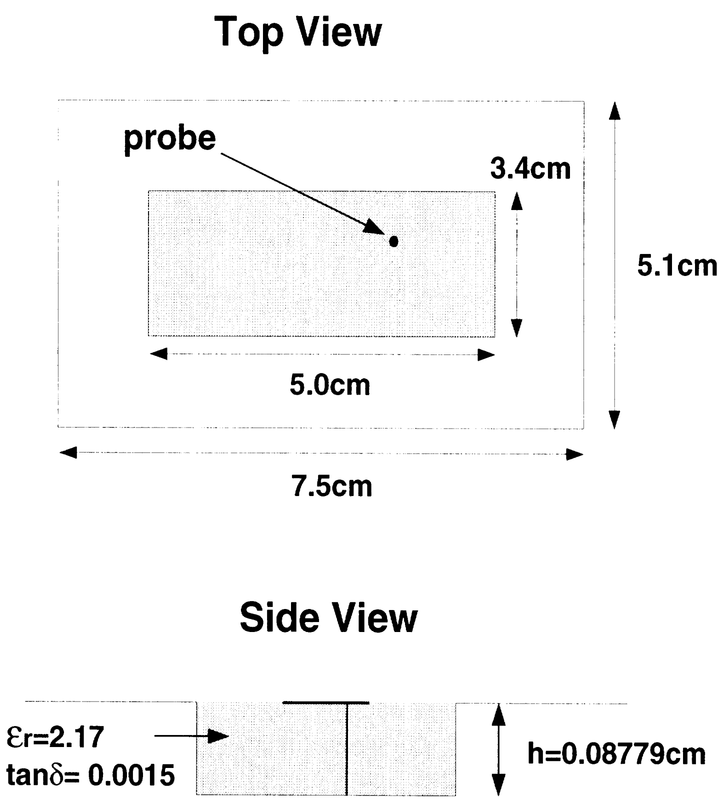

Figure 8: Rectangular patch geometry.

0	Specifies that the surface substrate nodes are not all dielectric.
1	Specifies a patch geometry.
.4166667 .425	Size(cm) of the elements(x,y) between the patch and cavity wall.
3 2	Number of elements in the x-y direction between the patch and cavity wall.
.4166667 .425	Size(cm) of the elements(x,y) on the patch.
12 8	Number of elements in the x-y direction on the patch.
1	Specifies tetra elements.
1	Specifies one substrate layer.
.08779	Specifies the substrate height.

Figure 9: Input file *inputmeshZin*.

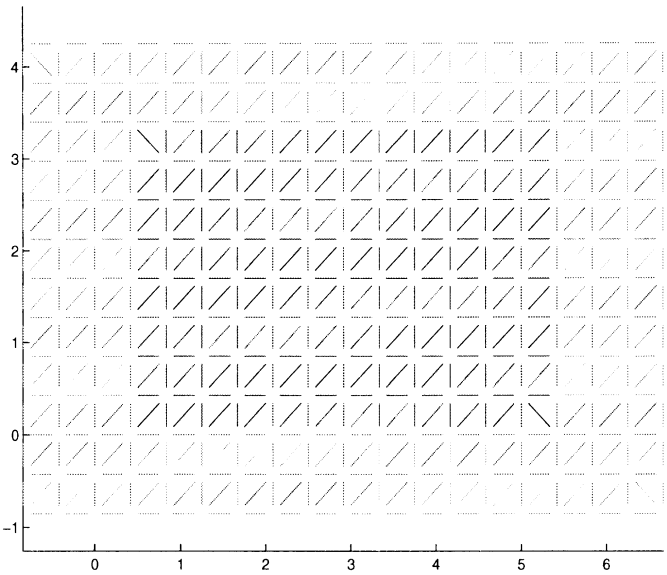

Figure 10: Surface mesh for the rectangular patch.

1	Preprocessor menu option.
Zin.unv	Name of Universal file(mesh).
1	Specifies a patch geometry.
1	Shape of cavity(1=rectangular).
1	Specifies number of patches.
5.41667	Maximum x value for patch.[2]
0.4166667	Minimum x value for patch.[2]
3.4	Maximum y value for patch.[2]
0.0	Minimum y value for patch.[2]
1	Specifies number of material layers.
(2.17,-0.003255) (1.,0.)	$(\epsilon_{r_{real}}, \epsilon_{r_{imag}}), (\mu_{r_{real}}, \mu_{r_{imag}})$.

Figure 11: Z_{in} preprocessor file

4.1.2 Preprocessor

Once the mesh has been created, it must then be processed by *femtetra.f*. The input file is shown on page 15 with a description for each line of input. Once the mesh has been processed, *femtetra.f* will automatically exit from the code. The user must then recompile *femtetra.f* using the *Makefile*. The preprocessor automatically creates the include file *DIM.INC* which contains the proper dimensioning. Thus the user does not have to change or adjust the dimensioning for the code at all.

[2]See Section 3.1.1.

2	Option to run the code.
3	Specifies output data desired(3=Radiation).
1.8, 1.87, 9	Specifies frequency range: f_{start},f_{stop}, number of frequency points.
2	Specifies radiation pattern, Z_{in}, or gain(2=Z_{in}).
Zin	Output file name.
1	Number of current probes.
101	Global node #1 for current probe edge[3].
348	Global node #2 for current probe edge[3].
0	Specifies desired solver(0=Symmetric BiCG).
.03	Specifies tolerance.
1	Option to implement AIM(1=yes)[4].
6,6	x-y integer threshold values for determining Z^{BI}_{far} and Z^{BI}_{near}.
21,42	Size of FFT grid(NXFFT) and FFT pad(NXFFTP); NXFFTP=2*NXFFT[5].
1	Check to see if input data is correct to this point(1=yes).
0.	Specifies phase for probe feed(Assume unit amplitude).
0	Specifies whether there are any ferrite layers(0=no).

Figure 12: Z_{in} input file.

4.1.3 Running the Code and Code Output

After the mesh has been processed and *femtetra.f* has been recompiled, the code is then ready to run. Figure 12 shows the input file for the rectangular patch, and a description for each line of input. The ouput file Z_{in} is shown in Figure 13. A plot of the file is shown in Figure 14. The first column in Figure 13 is frequency(GHz), the second column is the input resistance R_{in}, the third column is the input reactance X_{in}, the fouth column is the residual error of the iterative solver, the fifth column is the time it takes to solve for the unknown electric fields, and the sixth column is the number of iterations needed for convergence.

4.2 Circular Patch: Backscattering

The circular patch shown in Figure 15, can be found in [4]. It consists of a circular patch on an infinite dielectric substrate. The patch is modeled in a finite sized

[3]See Section 3.2.1.

[4]See [3].

[5]After compiling, the user must look at the file*DIM.INC* in order to ascertain the value needed for input for **NXFFTP**. This value should be set to the value of **MAXFFTP** in *DIM.INC*.

Freq(GHz)	Rin	Xin	Res.	Time(s)	Iter.
1.800	2.44997	29.70910	.03	497.7800	794
1.840	23.52436	77.13712	.03	517.1900	864
1.845	41.05225	96.10212	.03	547.3000	628
1.850	82.92909	119.73116	.03	441.6300	899
1.855	183.08553	106.96385	.03	562.7600	625
1.856	196.88599	96.64947	.01	517.1400	1012
1.860	223.11853	-52.38531	.03	440.4600	752
1.865	111.11250	-112.74718	.03	497.2300	1063
1.870	52.95052	-92.74758	.03	635.9600	884
1.911	3.46754	-21.77493	.03	618.9000	807
1.967	.92698	-7.42587	.03	520.4500	667
2.022	.46880	-2.39746	.03	458.2000	837
2.078	.31412	.37886	.03	534.6700	663
2.133	.25608	2.30123	.03	456.6200	795
2.189	.23824	3.85179	.03	515.8199	657
2.244	.25143	5.26132	.02	454.0700	657
2.300	.29810	6.68483	.02	454.2800	641

Figure 13: Z_{in} output file.

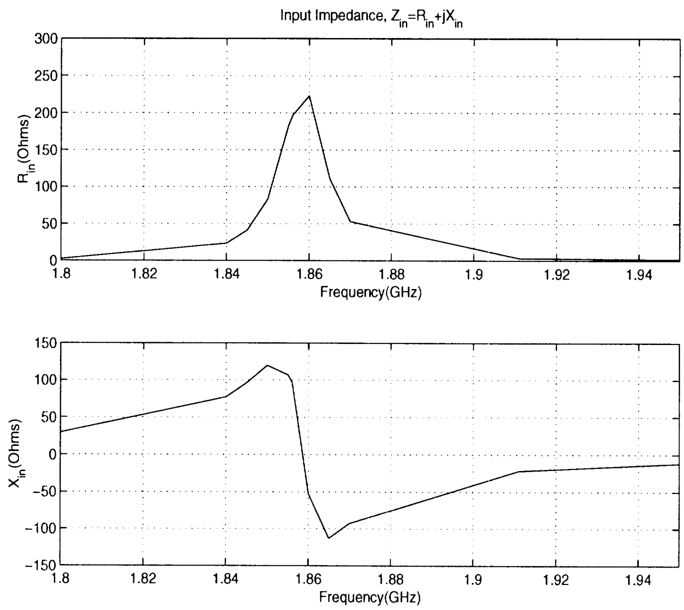

Figure 14: Plot of $Z_{in} = R_{in} + jX_{in}$.

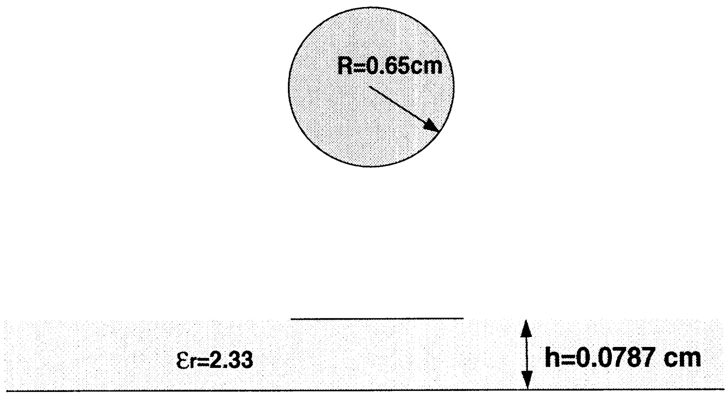

Figure 15: Circular patch geometry.

0	Specifies that the surface substrate nodes are not all dielectric.
1	Specifies a patch geometry.
.175625	Size(cm) of the radial elements between the patch and cavity wall.
8	Number of elements in the radial direction between the patch and cavity wall.
.1625	Size(cm) of the radial elements on the patch.
4	Number of elements in the x-y direction on the patch.
1	Specifies tetra elements.
1	Specifies one substrate layer.
.0787	Specifies the substrate height.

Figure 16: Input file to *meshcirc.f*.

cavity, that is large enough to simulate the infinite substrate. The patch radius is 0.65cm, the substrate height is 0.0787cm, the distance from the cavity to the patch is 1.405cm, and the substrate has an $\epsilon_r = 2.33$.

4.2.1 Mesh Generation

In order to calculate the backscattering for this geometry, the mesh must be generated. To accomplish this using the automatic mesher, the appropriate file to use is *meshcirc.f*. The corresponding input file *inputmeshcircpatch* is shown on page 18 with a description for each line of input. The surface mesh generated by the automatic mesher is shown in Figure 17.

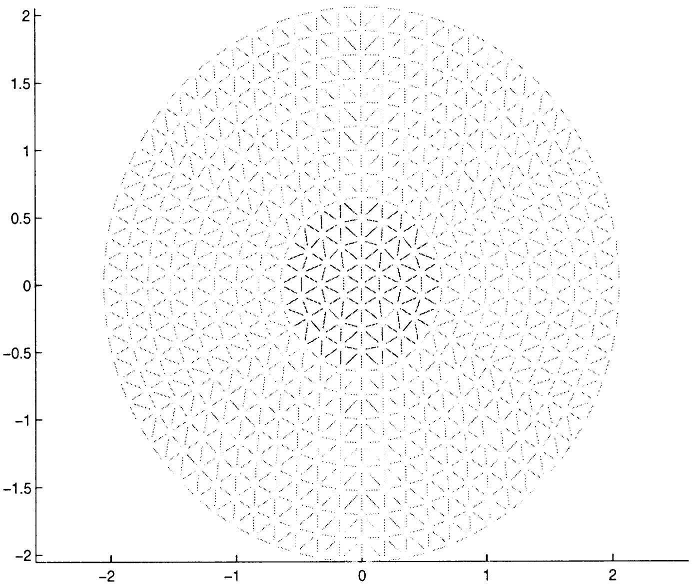

Figure 17: Surface mesh for circular patch.

1	Preprocessor menu option.
circpatch.unv	Name of Universal file(mesh).
1	Specifies a patch geometry.
0	Shape of cavity(0=circular).
2.005	Radius of the *entire* cavity.
1	Specifies number of patches.
0.0	x coordinate for the center of the patch.
0.0	y coordinate for the center of the patch.
0.65	Radius of the circular patch.
1	Specifies number of material layers.
(2.33,0.0) (1.,0.)	$(\epsilon_{r_{real}}, \epsilon_{r_{imag}})$, $(\mu_{r_{real}}, \mu_{r_{imag}})$.

Figure 18: Circular patch preprocessor input file.

4.2.2 Preprocessor

Once the mesh has been created, it must then be processed by *femtetra.f.* The input file is shown in Figure 18. Once the mesh has been processed, *femtetra.f* will automatically exit from the code. The user must then recompile *femtetra.f.* The preprocessor automatically creates the include file *DIM.INC* which contains the proper dimensioning. Thus the user does not have to change or adjust the dimensioning for the code at all.

2	Option to run the code.
2	Specifies output data desired(2=Backscatter).
7., 9., 1	Specifies frequency range: f_{start},f_{stop}, number of frequency points.
rcscirc	Output file name.
0	Option to normalize the Radar Cross Section(RCS).
0	Choose E field polarization angle, α (degrees) (0=$\theta - pol$).
2	Specifies cut(2=θ).
180.	*Fixed* start observation angle ϕ(degrees).
180.	*Fixed* start observation angle ϕ(degrees).
60.	Start observation angle θ(degrees).
60.	Start observation angle θ(degrees).
1	Number of observation points.
1	Number of *fixed* observation points.
0	Specifies desired solver(0=Symmetric BiCG).
.03	Specifies tolerance.
1	Option to implement AIM(1=yes)[6].
6,6	x-y integer threshold values for determining Z^{BI}_{far} and Z^{BI}_{near}.
36,72	Size of FFT grid(NXFFT) and FFT pad(NXFFTP); NXFFTP=2*NXFFT[7].
1	Check to see if input data is correct to this point(1=yes).
0	Specifies whether there are any ferrite layers(0=no).

Figure 19: Circular patch input file.

4.2.3 Running the Code and Code Output

After the mesh has been processed and *femtetra.f* has been recompiled, the code is then ready to run. On page 21 is the input file for the circular patch, and a description for each line of input. The ouput file *rcscirc* is shown in Figure 20. A plot of the file is shown in Figure 21. The first column in Figure 20 is the backscatter angle θ (degrees), the second column is the backscatter angle ϕ (degrees), the third column is the frequency(GHz), the fourth column is σ (dBsm), the fifth column is $\sigma_{\theta\phi}$ (dBsm), the sixth column is $\sigma_{\theta\theta}$ (dBsm), the seventh column is the residual error, the eighth column is the time(sec.) it takes to solve for the unknown electric fields, and the ninth column is the number of iterations needed for convergence.

[6]See [3].

[7]After compiling, the user must look at the file*DIM.INC* in order to ascertain the value needed for input for **NXFFTP**. This value should be set to the value of **MAXFFTP** in *DIM.INC*.

ϕ	θ	Freq(GHz)	σ	$\sigma_{\theta\phi}$	$\sigma_{\theta\theta}$	Res.	Time(s)	Iter.
180.000	60.000	7.000	-49.514	-80.000	-49.514	.0210	1060.87	156
180.000	60.000	7.250	-38.948	-80.000	-38.948	.0279	1122.71	225
180.000	60.000	7.500	-26.992	-98.964	-26.992	.0263	1081.23	179
180.000	60.000	7.750	-31.237	-103.721	-31.237	.0255	1071.01	166
180.000	60.000	8.000	-35.036	-107.449	-35.036	.0246	1109.24	210
180.000	60.000	8.250	-37.151	-108.184	-37.151	.0278	1074.67	171
180.000	60.000	8.500	-38.670	-108.764	-38.670	.0284	1047.25	141
180.000	60.000	8.750	-40.019	-108.835	-40.019	.0290	1046.36	140
180.000	60.000	9.000	-41.366	-108.713	-41.366	.0267	1046.63	140

Figure 20: Backscattering output file.

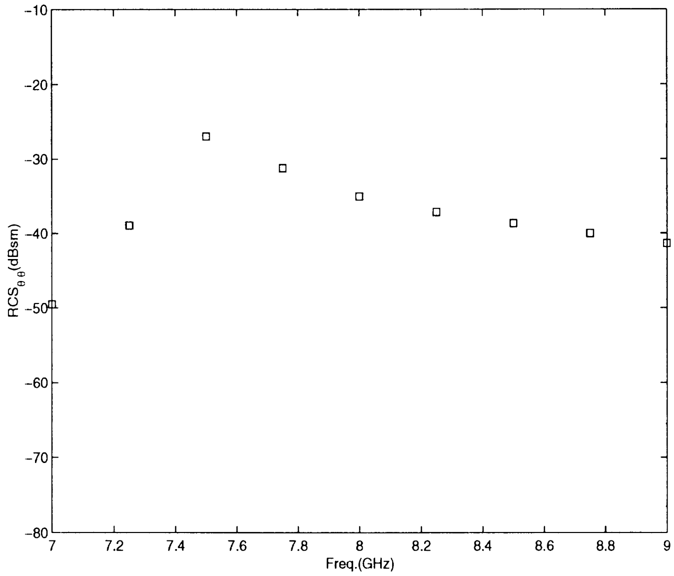

Figure 21: Plot of $\sigma_{\theta\theta}$.

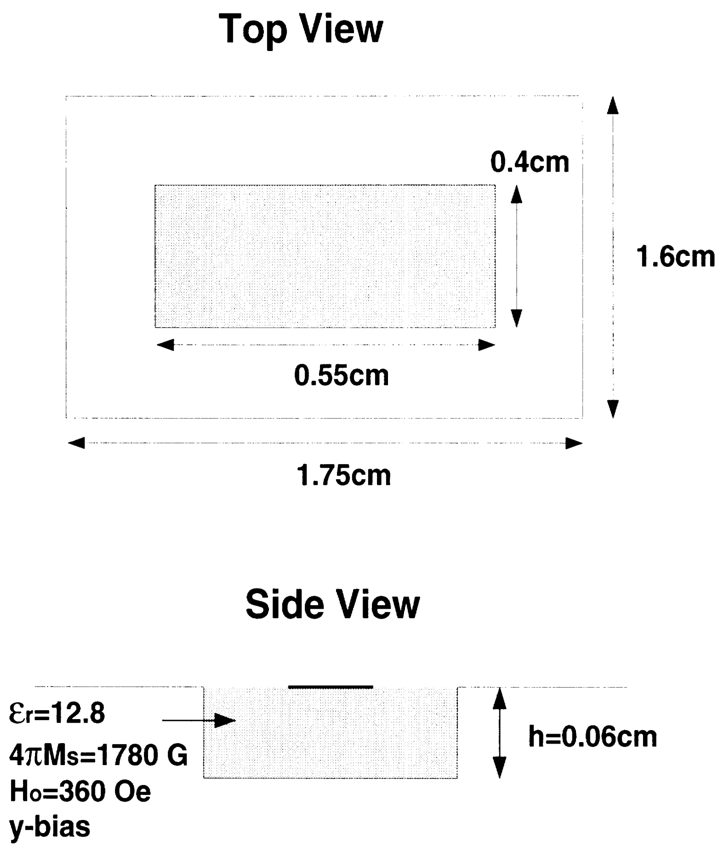

Figure 22: Rectangular patch geometry.

4.3 Rectangular Ferrite Patch: Backscattering

The rectangular patch shown in Figure 22 can be found in [5]. It consists of a rectangular patch residing on an infinite y-biased ferrite substrate. In order to model this a finite cavity was used. The patch size is 0.55 cm x 0.4 cm and the cavity size is 1.75 cm x 1.6 cm. The substrate has a thickness of 0.06 cm, $\epsilon_r = 12.8$, saturation magnetization($4\pi M_s$) = 1780 G and a y-bias magnetic field(H_o) = 360 Oe.

0	Specifies that the surface substrate nodes are not all dielectric.
1	Specifies a patch geometry.
.06 .06	Size(cm) of the elements(x,y) between the patch and cavity wall.
10 10	Number of elements in the x-y direction between the patch and cavity wall.
.06875 .06667	Size(cm) of the elements(x,y) on the patch.
8 6	Number of elements in the x-y direction on the patch.
1	Specifies tetra elements.
1	Specifies one substrate layer.
.06	Specifies the substrate height.

Figure 23: Input file *inputmeshyang*.

4.3.1 Mesh Generation

In order to generate the backscattering data for this geometry, the mesh must be generated. To accomplish this using the automatic mesher, the appropriate file to use is *meshrect.f*. The corresponding input file *inputmeshyang* is shown in Figure 23 with a description for each line of input. The surface mesh generated by the automatic mesher is shown in Figure 24.

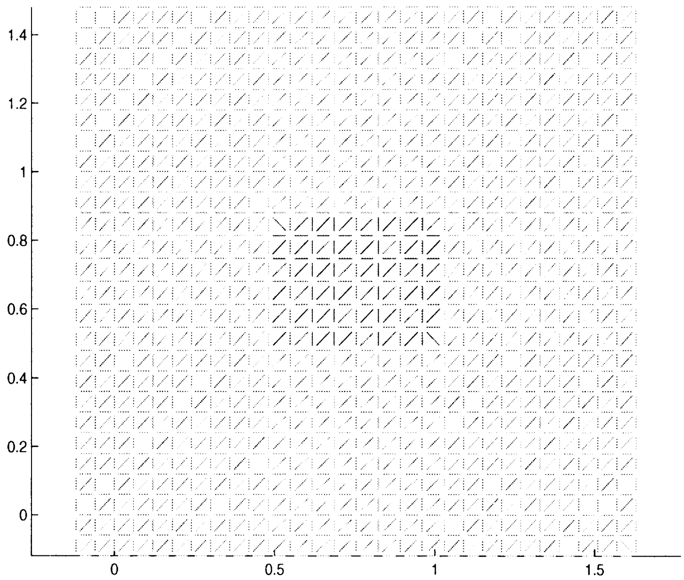

Figure 24: Surface mesh for the rectangular patch.

1	Preprocessor menu option.
yang.unv	Name of Universal file(mesh).
1	Specifies a patch geometry.
1	Shape of cavity(1=rectangular).
1	Specifies number of patches.
1.03	Maximum x value for patch[8].
0.48	Minimum x value for patch.[8]
0.88002	Maximum y value for patch.[8]
0.48	Minimum y value for patch.[8]
1	Specifies number of material layers.
(12.8,0.) (1.,0.)	$(\epsilon_{r_{real}}, \epsilon_{r_{imag}})$, $(\mu_{r_{real}}, \mu_{r_{imag}})$.

Figure 25: Ferrite patch preprocessor file

4.3.2 Preprocessor

Once the mesh has been created, it must then be processed by *femtetra.f.* The input file is shown on page 26 with a description for each line of input. Once the mesh has been processed, *femtetra.f* will automatically exit from the code. The user must then recompile *femtetra.f* with the proper dimensioning. The preprocessor automatically creates the include file *DIM.INC* which contains the proper dimensioning. Thus the user does not have to change or adjust the dimensioning for the code at all.

[8]See Section 3.1.1.

4.3.3 Running the Code and Code Output

After the mesh has been processed and *femtetra.f* has been recompiled, the code is then ready to run. On page 27 is the input file for the rectangular patch, and a description for each line of input. The ouput file *yangferrircsy* is shown in Figure 27 for the y-biased patch. A plot of the file is shown in Figure 28. Both biased and unbiased($4\pi M_s = 0$ G and $H_o = 0$ Oe) data are shown in the plot.

2	Option to run the code.
2	Specifies output data desired(2=Backscatter).
6.4,20.,35	Specifies frequency range: f_{start},f_{stop}, number of frequency points.
rcscirc	Output file name.
0	Option to normalize the Radar Cross Section(RCS).
0	Choose E field polarization angle, α (degrees) (0=$\theta - pol$).
2	Specifies cut(2=θ).
45.	*Fixed* start observation angle ϕ(degrees).
45.	*Fixed* start observation angle ϕ(degrees).
60.	Start observation angle θ(degrees).
60.	Start observation angle θ(degrees).
1	Number of observation points.
1	Number of *fixed* observation points.
0	Specifies desired solver(0=Symmetric BiCG).
.03	Specifies tolerance.
1	Option to implement AIM(1=yes)[9].
6,6	x-y integer threshold values for determining Z^{BI}_{far} and Z^{BI}_{near}.
47,94	Size of FFT grid(NXFFT) and FFT pad(NXFFTP); NXFFTP=2*NXFFT[10].
1	Check to see if input data is correct to this point(1=yes).
1	Specifies whether there are any ferrite layers(1=yes).
1	Specifies ferrite or isotropic for each layer(1=ferrite).
1780.	$4\pi M_s$ value.
360.	H_o value.
0	Specifies whether bias is non-uniform or not(0=no).
0.	Linewidth(ΔH) value.
1	Direction of bias(1 = y-bias).

Figure 26: Ferrite patch input file.

[9] See [3].

[10] After compiling, the user must look at the file*DIM.INC* in order to ascertain the value needed for input for **NXFFTP**. This value should be set to the value of **MAXFFTP** in *DIM.INC*.

φ	θ	Freq(GHz)	σ	σφθ	σθθ	Res.	Tiime(sec.)	Iter.
45.000	60.000	6.400	-55.004	-63.503	-55.665	0.0263	5050.96	420
45.000	60.000	6.800	-47.834	-55.703	-48.609	0.0281	4020.92	286
45.000	60.000	7.200	-52.078	-56.610	-53.964	0.0212	4667.75	389
45.000	60.000	7.600	-66.251	-66.377	-81.703	0.0292	4165.59	299
45.000	60.000	8.000	-63.838	-69.537	-65.200	0.0284	3652.52	204
45.000	60.000	8.400	-61.424	-69.846	-62.099	0.0294	3799.83	207
45.000	60.000	8.800	-59.671	-68.418	-60.293	0.0297	3595.98	198
45.000	60.000	9.200	-57.619	-65.654	-58.362	0.0279	4244.96	275
45.000	60.000	9.600	-53.928	-61.069	-54.861	0.0291	5104.06	414
45.000	60.000	10.000	-43.544	-50.302	-44.572	0.0284	4453.91	300
45.000	60.000	10.400	-47.287	-54.831	-48.128	0.0286	4013.00	244
45.000	60.000	10.800	-53.797	-63.756	-54.259	0.0295	3925.36	234
45.000	60.000	11.200	-54.914	-68.964	-55.088	0.0247	3755.71	220
45.000	60.000	11.600	-53.402	-72.828	-53.451	0.0288	4204.11	308
45.000	60.000	12.000	-45.721	-67.884	-45.748	0.0287	4375.55	337
45.000	60.000	12.400	-62.062	-71.537	-62.582	0.0288	4868.86	414
45.000	60.000	12.800	-52.454	-65.255	-52.688	0.0225	4774.47	397
45.000	60.000	13.200	-37.413	-44.516	-38.354	0.0278	5583.75	525
45.000	60.000	13.600	-62.659	-64.600	-67.090	0.0285	3913.35	242
45.000	60.000	14.000	-57.752	-69.638	-58.043	0.0208	3861.77	225
45.000	60.000	14.400	-55.497	-72.225	-55.590	0.0296	3933.65	232
45.000	60.000	14.800	-54.167	-73.392	-54.219	0.0292	3913.48	229
45.000	60.000	15.200	-53.253	-73.352	-53.296	0.0287	3898.74	230
45.000	60.000	15.600	-52.574	-72.129	-52.623	0.0224	3944.39	255
45.000	60.000	16.000	-52.068	-69.457	-52.148	0.0295	4825.23	412
45.000	60.000	16.400	-52.393	-61.862	-52.914	0.0262	4337.27	328
45.000	60.000	16.800	-49.088	-68.792	-49.135	0.0251	4111.01	279
45.000	60.000	17.200	-45.704	-57.832	-45.979	0.0243	4143.86	289
45.000	60.000	17.600	-43.158	-49.604	-44.275	0.0275	5153.78	450
45.000	60.000	18.000	-56.564	-60.900	-58.559	0.0291	4267.26	300
45.000	60.000	18.400	-57.988	-66.222	-58.694	0.0276	4555.44	335
45.000	60.000	18.800	-55.211	-72.497	-55.293	0.0276	4514.57	345
45.000	60.000	19.200	-44.825	-54.051	-45.377	0.0272	5622.29	531
45.000	60.000	19.600	-42.380	-54.069	-42.685	0.0249	6569.80	713
45.000	60.000	20.000	-49.478	-59.978	-49.884	0.0259	4592.91	340

Figure 27: Backscattering output file.

The first column in Figure 27 is the backscatter angle θ (degrees), the second column is the backscatter angle ϕ (degrees), the third column is the frequency(GHz), the fourth column is σ (dBsm), the fifth column is $\sigma_{\theta\phi}$ (dBsm), the sixth column is $\sigma_{\theta\theta}$ (dBsm), the seventh column is the residual error, the eighth column is the time(sec.) it takes to solve for the unknown electric fields, and the ninth column is the number of iterations needed for convergence.

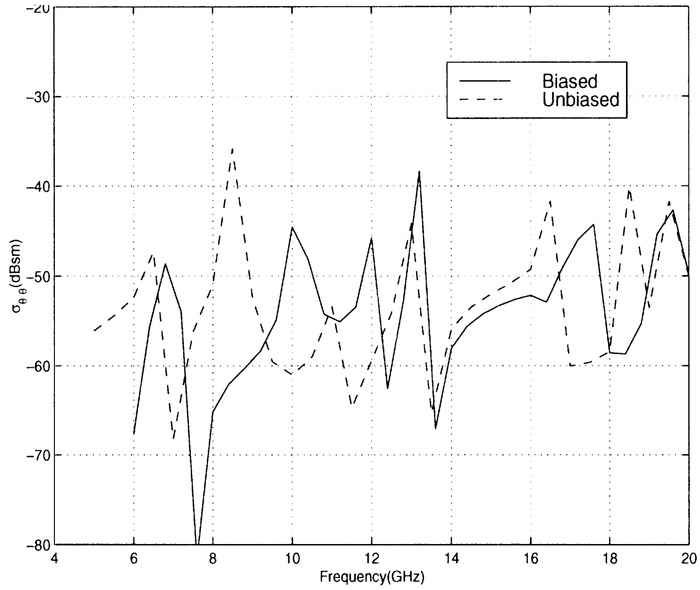

Figure 28: Plot of $\sigma_{\theta\theta}$.

References

[1] A. Brown, Z. Li and J. Volakis, "Theory, Mesher and Users Manual for a Ferrite Finite Element-Boundary Integral Tetrahedral Code," University of Michigan Radiation Laboratory, Department of Electrical Engineering and Computer Science, Ann Arbor, MI, 1998

[2] J. Jin and J.L. Volakis,"A hybrid finite element method for scattering and radiation by microstrip patch antennas and arrays residing in a cavity," *IEEE Trans. Antennas Propagat.*, vol. 39, no. 11, pp.1598-1604, 1991.

[3] T. Eibert and J.L.Volakis,"Fast Spectral Domain Method for Hybrid Finite Element - Boundary Integral Modeling of Doubly Periodic Structures," University of Michigan Report #375458-3-T, Ann Arbor, MI, 1998

[4] J.L.Volakis, J. Gong and A. Alexanian,"Electromagnetic scattering from microstrip patch antennas and spirals residing in a cavity," *Electromagnetics*, vol. 14, no. 1, pp.63-85, 1994.

[5] H.Y. Yang, J.A. Castaneda, and N.G. Alexopoulos,"The RCS of a microstrip patch on an arbitrarily biased ferrite substrate,"*IEEE Trans. Antennas Propagat.*, vol. AP-41, no. 12, pp. 1610-1614, 1993.