RUFDRIVER User Manual

L.C. Kempel and J.L. Volakis

December 1990

Abstract

An automatic geometry generation package is described for use with **RUFCODE**. The capabilities of this mesh generation package is discussed and illustrated through several examples. This geometry generation package is easy to use and can provide accurate discretizations for electromagnetic applications. Visualization of the geometry on various machine architectures is made possible via an X-windows graphical interface.

mgn
MR0496

1 Introduction

The computation of the scattered fields by arbitrary composite scatterers requires a numerical solution of boundary and/or volume integral equations. For the solution of the internal scatterer fields or equivalent currents, these are first discretized to obtain a system of equations. To yield the discretized set of equations it is necessary that the scatter's volume (for three-dimensions) or cross section (for two-dimensions) be subdivided into small volume or surface elements. Although this is a well defined task, it is often tedious and difficult to accomplish without the aid of some programmable routine for discretizing a two-dimensional,composite cross section in a form compatible with the integral equations described in [1] for computing the scattering by the given cross section.

The integral equations employed in [1] involve one volume current density over the penerable portion of the scatterer and a surface current density at contours coinciding with discontinuities in material. Their numerical implementation comprises the code **RUFCODE** which requires that the geometry volume be entered as a collection of second order,small quadrilaterals, pie shells (could be large), or rectangles characterized with constant ϵ_r and μ_r or material parameters which vary in accordance with a prescribed variation (i.e. linear, Gaussian, etc.). The contours describing discontinuities in the material must be separately entered as a collection of linear or circular segments/arcs. All volume and contour elements must be entered in accordance with the format described in [1] and each of the elements (rectangular, pie shell, small quadrilaterals, line segment, or circular arc) requires at least one input line for its geometrical and material description. Obviously, in the case of volume sections not conforming to one pie shell or a rectangle, one must discretize or mesh it in terms of small quadrilaterals. This is, obviously, a tedious task requiring algorithmic implementation and in this report we describe the use of a mesh generation code based on a mapping algorithm for generating the mesh. The code will be referred to as **RUFDRIVER** and in addition to discretizing the cross section of the scatterer, it also provides an output file to be directly employed as the input of **RUFCODE**. Finally, the **RUFDRIVER** allows viewing of the discretized geometry on an X-window platform making all of its graphical features portable to other machine architectures.

2 Description

RUFDRIVER is a convenient software package for the discretization of a cross-sectional area in terms of second order quadrilaterals. The program contains several features which aid in the scattering analysis of two-dimensional structures. We will discuss here how one discretizes the geometry using **RUFDRIVER** in a manner compatable with the input format for **RUFCODE**. Next, we describe the features of **RUFDRIVER** in general terms and a specific example is deferred until the next section of this report.

RUFDRIVER requires the user to partition the geometry into several sections having a single taper specification chosen from those allowed by **RUFCODE**. The sections or blocks are four-sided and are described by four, eight, or twelve control points. Thus the blocks can be either rectangular, quadratic, or cubic quadralaterals. For example, four control points are required to provide a cubic polynomial fit for a side of the block and three control points are required for a quadratic fit. Figure 1 illustrates a strip which is subdivided into two blocks. Each block is described by four control points as shown.

The user must first layout the geometry for partitioning into as many blocks as required. This will also aid in the manual generation of contours since the same information is required to describe these contours. Initially, all control points are entered as prompted by **RUFDRIVER**. The blocks are then defined by specifying the control points on their periphery and this is accomplished through a graphical user-code interaction. **RUFDRIVER** performs the discretization of each block by mapping the block onto a rectangle which is then readily gridded in a uniform fashion. The samples are finally mapped back to the original domain yielding the appropriate discretization. For best results, the elements should be as close to square as possible. To achieve this, the blocks should not deviate substantially from a rectangle.

The final geometry should be viewed and the user must ensure that none of the elements are elongated. Since **RUFCODE** employs a constant basis expansion for the field over each element, inaccuracies will result if such elements are allowed. If neccessary, the user may need to subdivide the block which contains deformed cells into smaller partitions and repeat the discretization and mapping processes. This issue is addressed in the next section where a triangular section of a wedge is partitioned into three smaller blocks.

Once the blocks and control points are entered as prompted, the user will characterize each block by:

1) Material type
2) Material parameter taper
3) Two-dimensional sampling
4) Two-dimensional sampling gradient

The material type is a two digit number which corresponds to the material identifier as used by **RUFCODE** (see [1]). The material parameter taper is also the two digit identifier used by **RUFCODE**. The code next requests the number of samples in the x-direction (principal axis) and the y-direction. Finally, the user specifies the gradient of the sampling function. The first three items of the table above are familiar to **RUFCODE** users. The sampling gradient allows the sample interval to be smoothly tapered along the length and width of the block. The gradient may be any number greater than zero with the following characteristics:

Table 1
Gradient Parameter

Gradient	Effect
< 1.0	left interval is smaller
$= 1.0$	uniform sampling
> 1.0	right interval is smaller

The further the gradient is from unity, the larger the disparity of the sampling intervals. An illustration of the effect of a gradient is shown in figure 2. The geometry of figure 1 is tapered from the center. In this case, the left block has an x-gradient of 1.4 while the right block has an x-gradient of 0.5 . This type of sampling is desirable when the currents near the end of the block are expected to be rapidly changing. Thus, one avoids wasting unknowns by sampling higher near the ends than in the center of the block.

After all blocks have been completely specified, the mesh is generated and displayed. The user may choose to number the elements and nodes as shown in figure 2. The scale factor for the element/node numbers is proportional to the points of the font used. Thus, a smaller value results in smaller symbols. A value of unity has proven to be useful. The numbering of elements and nodes in the case of large geometries or high sampling is

not recommended since this will clutter the figure. Finally, **RUFDRIVER** will prompt the user to enter additional **RUFCODE** parameters such as material specifications, observation angles, etc.

Although **RUFDRIVER** is primarily used for the generation of volume elements, it can provide contour segments in some situations. If each node away from the outer boundary of the geometry is shared by at least three elements, the contours can be generated by **RUFDRIVER**. Specifically, if a node of an element falls on the link connecting two nodes of an adjacent element, the contour segments produced by **RUFDRIVER** are unreliable. The first three examples of the next section are sampled so that **RUFDRIVER** generates valid contour segments.

When **RUFDRIVER** is executed, the main menu allows the user to enter the parameters from the keyboard or a pre-saved,free formated geometry file. It also allows the creation of a geometry file as parameters are entered through the keyboard during the course of a session. The geometry file is appended upon keyboard entry and thus in case of an accidental session break, the entry of parameters may be continued upon restart. Since the geometry file will contain all keyboard entries including errors, it is advised that the geometry file be edited upon successful termination so that only the last set of input parameters is kept. The information contained in the geometry file is detailed below:

Table 2 Echo File Information
Description Number of blocks = N Number of control pts = M Control point 1 . . . Control point M Block 1 Material/Taper identifiers x and y sampling Number of control points for block = J Node 1 . . . Node J x- and y-gradient . . . Block N Element numbering flag Element number scale Node number flag Node number scale

In the next section we present a geometry generation example. This simple example is followed by several increasingly more complex mesh generations which illustrate the versitality of this geometry generation package. The first three examples employ **RUFDRIVER** to generate the contour segments. In the case of the fourth geometry, the contour segments are entered manually in the line entry format described in [1].

3 Examples

This section will present several examples of parameter entry for **RUFDRIVER**. The detailed example presented herein was chosen for it's simpicity. The chosen example is a two layer dielectric strip. This model consists of two different material blocks. Figure 3 is the geometry as generated by **RUFDRIVER**. The results for H-polarization compare favorably with those which are generated by a manually entered geometry. The second example refers to a dielectrically coated PEC cylinder. The associated geometry is shown in figure 4 along with a comparison to the eigenfunction solution for H-polarization. This model illustrates **RUFDRIVER**'s capability to model curved boundaries with quadratic or cubic polynomial boundaries. Once again, the computed data is in good agreement with the exact solution. The third example is a high-loss dielectric coated triangular cylinder. The associated geometry and a comparison of the E-polarization scattering pattern is shown in figure 5. The final example is a composite material wedge. The geometry as well as the E-polarization scattering pattern is shown in figure 6. The appendix contains the echo files for all the examples.

4 Summary

RUFCODE has been shown to be an accurate code to numerically solve for the scattering of two-dimensional,complex objects. With the addition of **RUFDRIVER** to aid in the generation of volume elements, this code can model more general,complex geometries. Since the graphical display of **RUFDRIVER** is written for X-window platform, the code may be run on many machine architectures. To date, the code has been run on Apollo, DEC, and Sun workstations.

This report gave a short description of how one discretizes a cross-section using **RUFDRIVER**. Some simple concepts on selecting the blocks and control points to use were discussed and some general observations were made. The required parameter entries for **RUFDRIVER** were discussed. Some examples were presented which illustrated the capability of the mesh generation package

References

[1] M.A. Ricoy, L.C. Kempel, and J.L. Volakis, Simple Integral Equations for Two-Dimensional Scattering with Further Reduction in Unknowns and Scattering Code User's Manual, University of Michigan Technical Report No. RL-857, Rev. 1, June 1990.

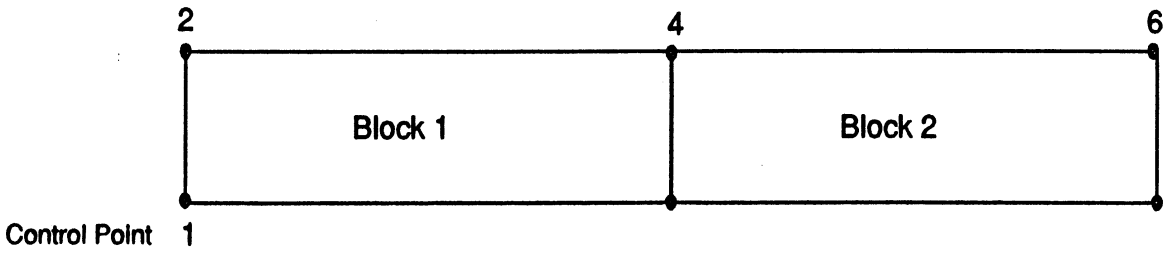


Figure 1. Illi

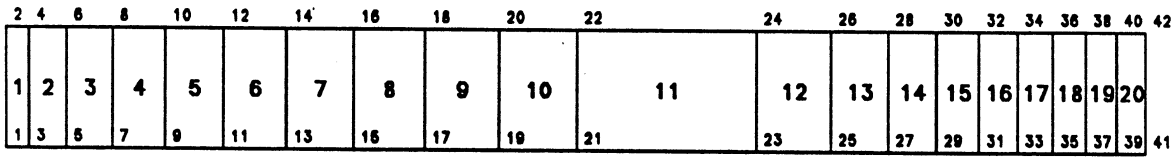


Figure 2. Effect of Gradient on Sampling Intervals

Block 1: x-gradient = 1.4, y-gradient = 1.0
Block 2: x-gradient = 0.5, y-gradient = 1.0

```
 E X E C U T I O N   B E G I N S


================================================

           --------------------------------
           :::::::::: RufDriver :::::::::
           --------------------------------

               Developed by CHONGYU HUA

             For the RADIATION LABORATORY
              THE UNIVERSITY OF MICHIGAN
                     ANN  ARBOR
                       1990

================================================


 -------- C U R R E N T   P A R A M E T E R S --------

        1. Input data device keyboard is : ON

        2. Echo mode of the program is   : OFF

      To alter any of the above parameters type
      the corresponding item number or 3 to exit
      this menu or 0 to stop the program

3


 >> :: GEOMETRY MODELING / MESH GENERATION ::


 >> Enter the number of blocks which
 >> are used to create the model

2

 >> Enter the number of control points which
 >> are used to define these blocks
 >> NOTE:  Total number for all blocks.

6

 >> Enter the coordinates for each control point

 ==> CONTROL POINT (   1)
-0.5 -0.1
 ==> CONTROL POINT (   2)
-0.5 0.0
 ==> CONTROL POINT (   3)
-0.5 0.1
 ==> CONTROL POINT (   4)
0.5 0.1
 ==> CONTROL POINT (   5)
0.5 0.0
 ==> CONTROL POINT (   6)
0.5 -0.1


 --- INPUT DATA FOR THE BLOCK   1 ---


 >> Enter an Identification Number for
 >> the material IMPEDANCE properties
 >>(epsilon,mu)

21

 >> Enter an Identification Number for
 >> the material TAPER properties
 >> (epsilon,mu)

11

 >> Enter the number of elements in
 >> the x & y directions

15 1

 >> Enter the number of control points which
 >> are use to define the current block
 >> (Enter 4 OR 8 OR 12 only)
```

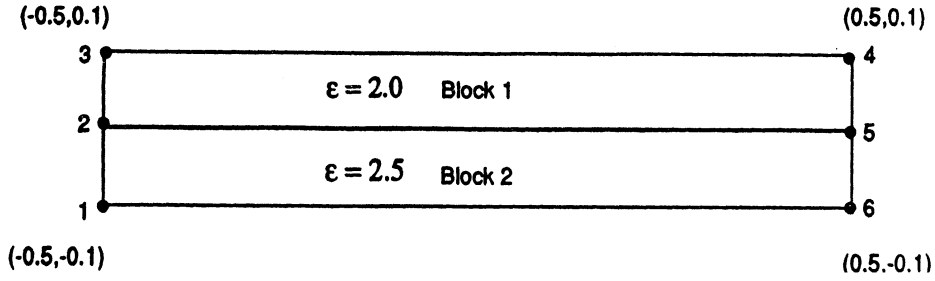
```
4

 >> Enter the block connectivity in
 >> the following block node order:

 >>       4 ------------ 3
 >>       |              |
 >>       |              |
 >>       |              |
 >>       |              |
 >>       |              |
 >>       1 ------------ 2

 ==> BLOCK NODE(  1 )
2
 ==> BLOCK NODE(  2 )
5
 ==> BLOCK NODE(  3 )
4
 ==> BLOCK NODE(  4 )
3

 >> Enter two real parameters for the mesh
 >> gradient in the 1-st and 2-nd direction

1.0 1.0

 >> Now, mesh generation. Please wait ...


 --- INPUT DATA FOR THE BLOCK   2 ---


 >> Enter an Identification Number for
 >> the material IMPEDANCE properties
 >>(epsilon,mu)

34

 >> Enter an Identification Number for
 >> the material TAPER properties
 >> (epsilon,mu)

11

 >> Enter the number of elements in
 >> the x & y directions

15 1

 >> Enter the number of control points which
 >> are use to define the current block
 >> (Enter 4 OR 8 OR 12 only)

4

 >> Enter the block connectivity in
 >> the following block node order:

 >>       4 ------------ 3
 >>       |              |
 >>       |              |
 >>       |              |
 >>       |              |
 >>       |              |
 >>       1 ------------ 2

 ==> BLOCK NODE(  1 )
1
 ==> BLOCK NODE(  2 )
6
 ==> BLOCK NODE(  3 )
5
 ==> BLOCK NODE(  4 )
2

 >> Enter two real parameters for the mesh
 >> gradient in the 1-st and 2-nd direction

1.0 1.0

 >> Now, mesh generation. Please wait ...

 >> Do you wish to add element numbers
 >> on the plot ? (YES/NO)
```

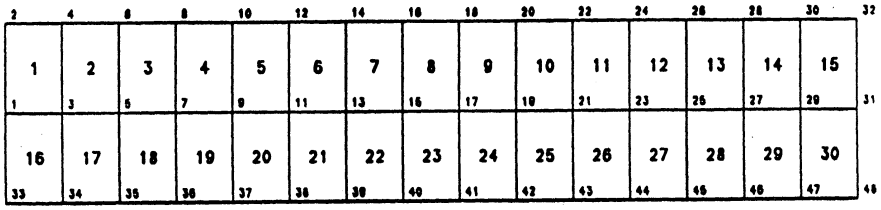
```
y

 >> Enter a scaling factor to specify
 >> the character size

1

 >> Do you wish to add node numbers
 >> on the plot ? (YES/NO)

y

 >> Enter a scaling factor to specify
 >> the character size

1

 >> Do you wish to save the plot ?
 >> (Yes/No)

n

 >> Do you wish to modify the plot ?
 >> (YES/NO)

n

 >> Are you satisfied with the generated mesh ?
 >> (Yes/No). If not, we'll do it again.

y

 >> Good. Let's continue ...


 >> Enter a file name to save data

strip.inr

 >> Enter a problem title (up to 72 characters)

Strip as done by RufDriver

 >> -------- Pattern Type Specification --------
 >>
 >> Enter 0 for bistatic scattering pattern, or
 >>       1 for backscatter scattering pattern

1

 >> ------ Scaling Ratio Specification ------
 >>
 >> Enter a scaling ratio which will be used
 >> to convert the geometric data coordinates
 >> in terms of wavelength

1

 >> ----- Polarization Specification -----
 >>
 >>    Enter 1 for E-polarization, or
 >>          2 for H-polarization

2

 >> Enter plot file name for output data
 >> (6 characters)

stripH

 >> Enter the initial observation angle (in degrees)

0

 >> Enter the final observation angle (in degrees)

90

 >> Enter a angular increment (in degrees)

1

 >> Enter an incident angle for bistatic
 >> computations

0
```

```
 >> --------- Output Request 1 ---------
 >> Enter 0 no option
 >>       1 to compute gap impedance, or
 >>       2 to include image wave

0

 >> --------- Output Request 2 ---------
 >>
 >> Enter 0 for no print option, or
 >>       3 to print absorber statistics, or
 >>       4 to print matrix elements (magnitude), or
 >>       5 to print matrix elements (complex), or
 >>       6 to print geometry plot file, or
 >>       7 to print parametric cell principle nodes, or
 >>       8 to print both geometry and parametric cellsc, or
 >>       9 = 3 and 8 above

3

 >> Enter the offset

0

 >> ::::: Material Impedance Specifications :::::

 >> How many sets of different material impedance
 >> properties are considered ?

4

 ==> Material Set No.(    1)

 >> Enter an identification number for
 >> the designated impedance properties

1

 >> Enter the RE and IM parts of the starting
 >> complex permittivity or permeability

1 0

 >> Enter the RE and IM parts of the ending
 >> complex permittivity or permeability

1 0

 ==> Material Set No.(    2)

 >> Enter an identification number for
 >> the designated impedance properties

2

 >> Enter the RE and IM parts of the starting
 >> complex permittivity or permeability

2 0

 >> Enter the RE and IM parts of the ending
 >> complex permittivity or permeability

2 0

 ==> Material Set No.(    3)

 >> Enter an identification number for
 >> the designated impedance properties

3

 >> Enter the RE and IM parts of the starting
 >> complex permittivity or permeability

2.5 .1

 >> Enter the RE and IM parts of the ending
 >> complex permittivity or permeability

2.5 .1

 ==> Material Set No.(    4)

 >> Enter an identification number for
 >> the designated impedance properties
```

```
4

 >> Enter the RE and IM parts of the starting
 >> complex permittivity or permeability

1 0

 >> Enter the RE and IM parts of the ending
 >> complex permittivity or permeability

1 0

 >> ::::: Material Taper Specifications :::::

 >> How many sets of different material taper
 >> properties are considered(>= 1) ?

1

 ==> Material Set No.(    1)

 >> Enter an identification number for
 >> the designated taper properties

1

 >> ----- Type of Material Taper -----
 >>
 >> Enter 0 for no taper, or
 >>       1 for linear type, or
 >>       2 for Gaussian type, or
 >>       3 for Hanning type, or
 >>       4 for Blackman-Harris type

0

 >> Please wait ...

Volume Elements generated ...

Do you want contours? (1=yes)
1
Segments ordered through:   2 out of   34
Segments ordered through:   3 out of   34
Segments ordered through:   4 out of   34
Segments ordered through:   5 out of   34
Segments ordered through:   6 out of   34
Segments ordered through:   7 out of   34
Segments ordered through:   8 out of   34
Segments ordered through:   9 out of   34
Segments ordered through:   10 out of   34
Segments ordered through:   11 out of   34
Segments ordered through:   12 out of   34
Segments ordered through:   13 out of   34
Segments ordered through:   14 out of   34
Segments ordered through:   15 out of   34
Segments ordered through:   16 out of   34
Segments ordered through:   17 out of   34
Segments ordered through:   18 out of   34
Segments ordered through:   19 out of   34
Segments ordered through:   20 out of   34
Segments ordered through:   21 out of   34
Segments ordered through:   22 out of   34
Segments ordered through:   23 out of   34
Segments ordered through:   24 out of   34
Segments ordered through:   25 out of   34
Segments ordered through:   26 out of   34
Segments ordered through:   27 out of   34
Segments ordered through:   28 out of   34
Segments ordered through:   29 out of   34
Segments ordered through:   30 out of   34
Segments ordered through:   31 out of   34
Segments ordered through:   32 out of   34
Segments ordered through:   33 out of   34
Segments ordered through:   34 out of   34

Number of VOLUME elements generated:   30
Number of CONTOUR segments generated:  49


  E X E C U T I O N   T E R M I N A T E D

  The following files have been created :

  test            - contains the RufCode
                    data generated by this run
```

```
################################################
#             NOTE:  RUFCODE Reminder          #
################################################
# If one of the following conditions exist in  #
# the model, please remove lines from the      #
# RUFCODE input file:                          #
# 1) PEC body w/ E-pol:  Remove all contours   #
# 2) PEC body w/ H-pol:  Remove all volumes    #
# 3) PEC + diel. w/ H-pol: Remove PEC volumes  #
#     Remove contours on inside of PEC if any  #
# 4) PEC + diel. w/ E-pol: Remove all contours #
#     between PEC and air                      #
################################################
```





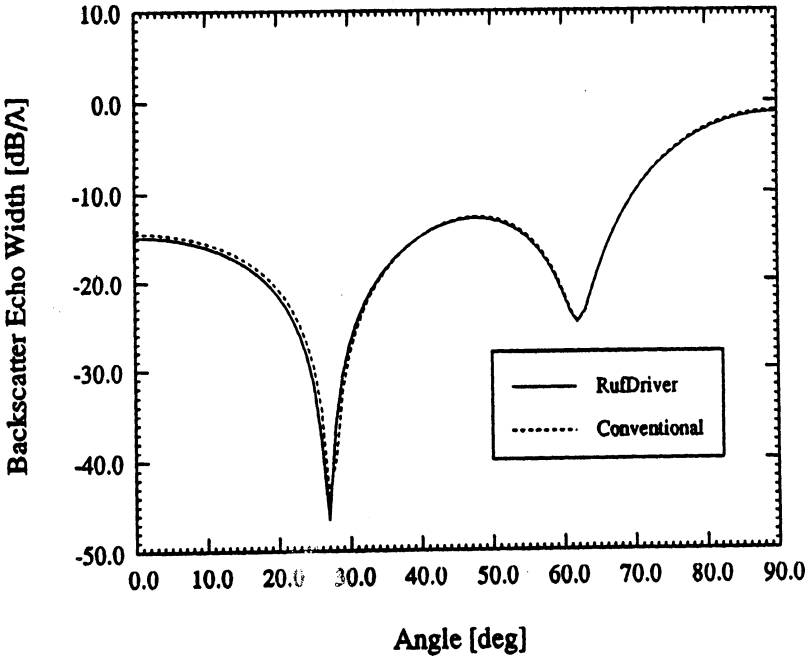
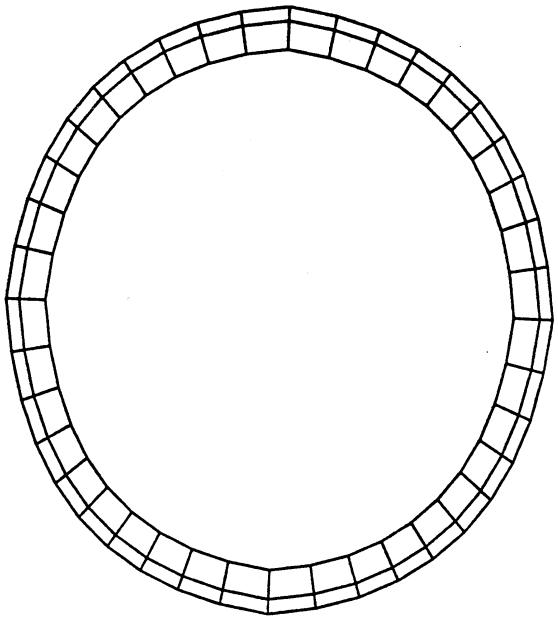


Figure 3. Two-layer dielectric strip



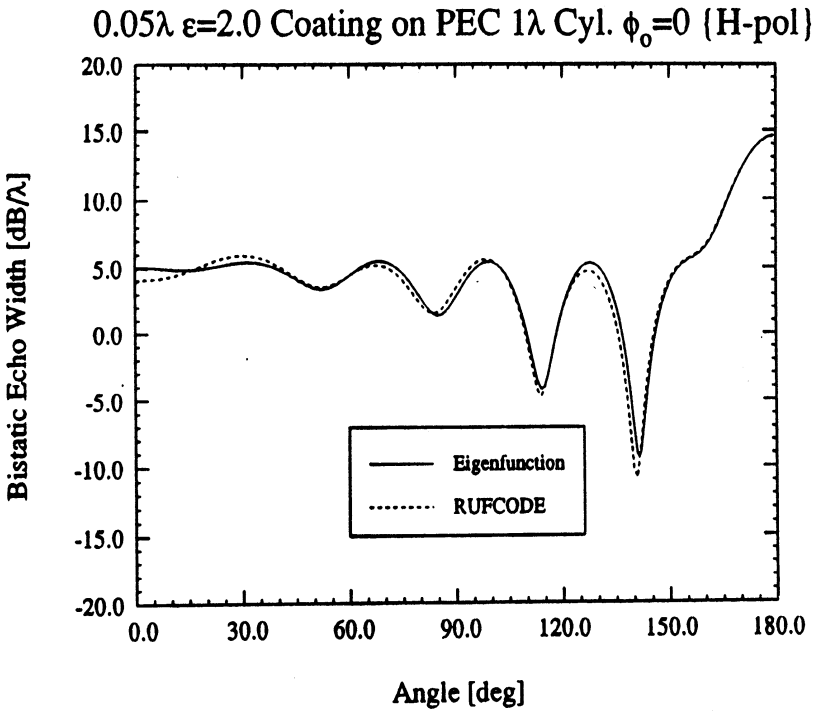
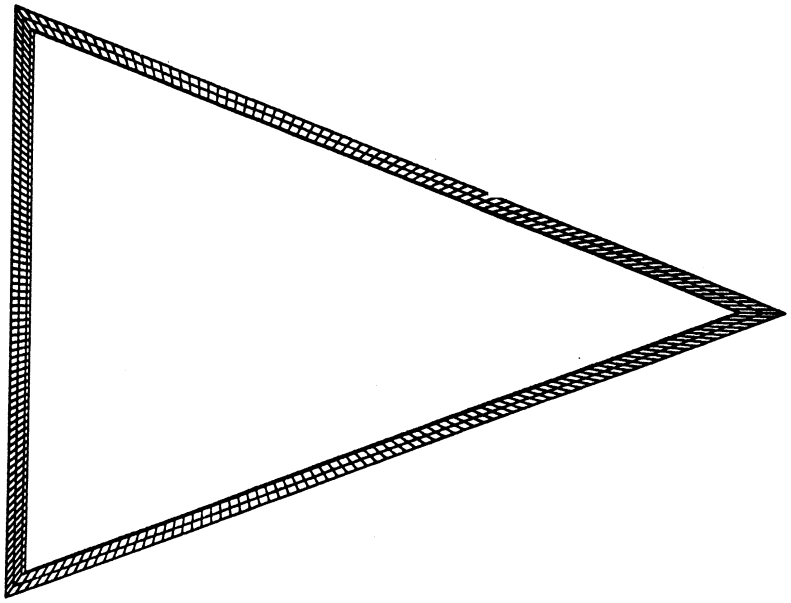
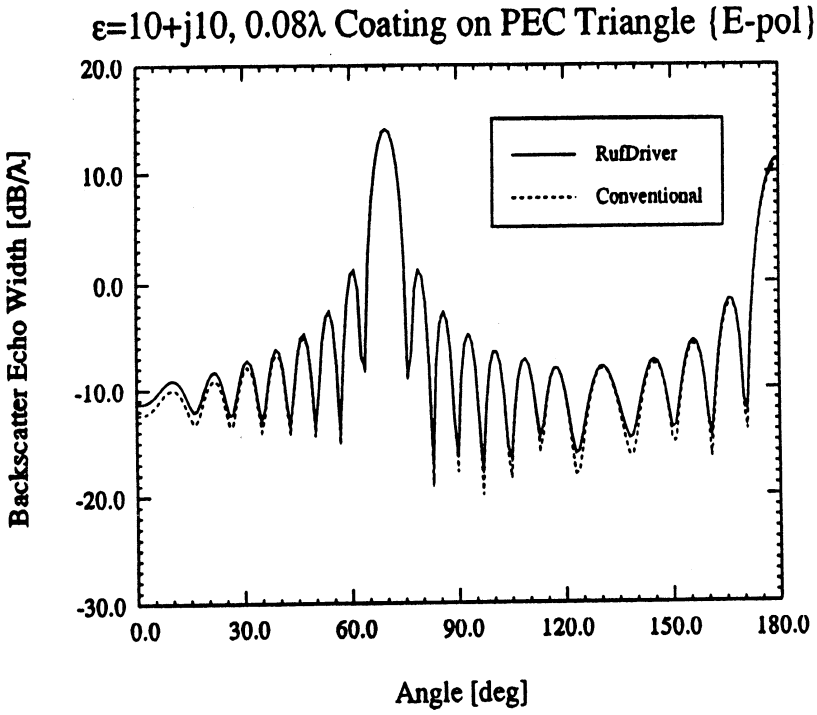
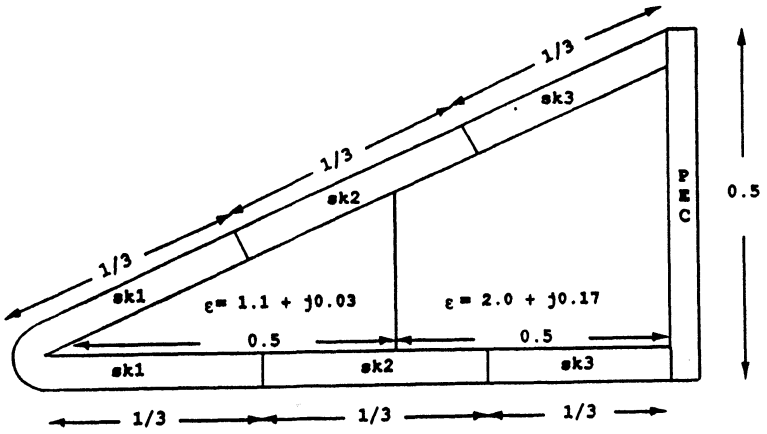


Figure 4. Dielectric coated PEC circular cylinder




ε=10+j10, 0.08λ Coating on PEC Triangle (E-pol)
Backscatter Echo Width [dBΛ]
20.0
10.0
0.0
-10.0
-20.0
-30.0
0.0
30.0
60.0
90.0
120.0
150.0
180.0
Angle [deg]
RufDriver
Conventional

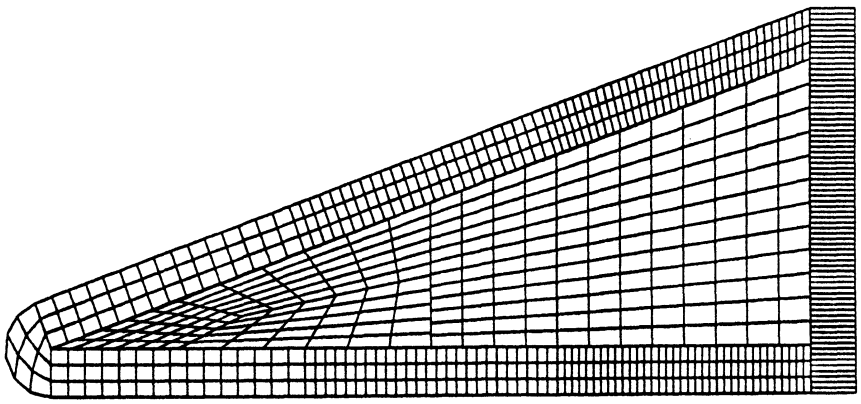


Note: All dimensions in wavelengths

sk1: ε= 4 + j1

sk2: ε= 8 + j3.3

sk3: ε= 16 + j11



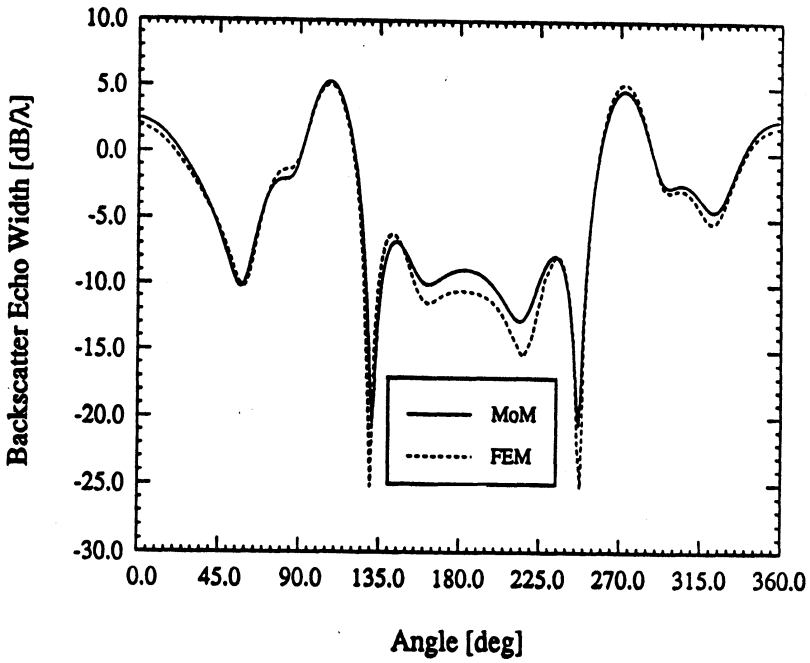


Figure 6. Composite material wedge

References

[1] M.A. Ricoy, L.C. Kempel, and J.L. Volakis, Simple Integral Equations for Two-Dimensional Scattering with Further Reduction in Unknowns and Scattering Code User's Manual, University of Michigan Technical Report No. RL-857, Rev. 1, June 1990.

APPENDIX

Strip Model Geometry File

```
2
6
-0.5000000 -0.1000000
-0.5000000 0.0000000E+00
-0.5000000 0.1000000
0.5000000 0.1000000
0.5000000 0.0000000E+00
0.5000000 -0.1000000
4
21 11
15 1
4
2
5
4
3
1.000000 1.000000
4
34 11
15 1
4
1
6
5
2
1.000000 1.000000
1
1.000000
1
1.000000
```

Coated Cylinder Geometry File

```
8
32
0.0000000E+00 0.9000000
0.6364000 0.6364000
0.9000000 0.0000000E+00
0.6364000 -0.6364000
0.0000000E+00 -0.9000000
-0.6364000 -0.6364000
-0.9000000 0.0000000E+00
-0.6364000 0.6364000
0.0000000E+00 0.9500000
0.9500000 0.0000000E+00
0.0000000E+00 -0.9500000
-0.9500000 0.0000000E+00
0.0000000E+00 1.000000
0.7071100 0.7071100
1.000000 0.0000000E+00
0.7071100 -0.7071100
0.0000000E+00 -1.000000
-0.7071100 -0.7071100
-1.000000 0.0000000E+00
-0.7071100 0.7071100
0.0000000E+00 1.025000
1.025000 0.0000000E+00
0.0000000E+00 -1.025000
-1.025000 0.0000000E+00
0.0000000E+00 1.050000
0.7424600 0.7424600
1.050000 0.0000000E+00
0.7424600 -0.7424600
0.0000000E+00 -1.050000
-0.7424600 -0.7424600
-1.050000 0.0000000E+00
-0.7424600 0.7424600
```

Coated Cylinder Geometry File Con't
4 41 11 10 1 8 1 3 15 13 2 10 14 9 1.000000 1.000000 4 41 11 10 1 8 3 5 17 15 4 11 16 10 1.000000 1.000000 4 41 11 10 1 8 5 7 19 17 6 12 18 11 1.000000 1.000000 4 41 11 10 1 8 7 1 13 19 8 9 20 12

Coated Cylinder Geometry File Con't
1.000000 1.000000 4 23 11 10 1 8 13 15 27 25 14 22 26 21 1.000000 1.000000 4 23 11 10 1 8 15 17 29 27 16 23 28 22 1.000000 1.000000 4 23 11 10 1 8 17 19 31 29 18 24 30 23 1.000000 1.000000

Coated Cylinder Geometry File Con't

```
4
23 11
10 1
8
19
13
25
31
20
21
32
24
1.000000 1.000000
-1
1.000000
-1
1.000000
```

Coated Triangle Geometry File
6
24
-3.758770 -1.368080
-3.758770 0.0000000E+00
-3.758770 1.368080
-1.879390 0.6840400
0.0000000E+00 0.0000000E+00
-1.879390 -0.6840400
-3.798770 -1.425210
-3.798770 1.425210
0.1169500 0.0000000E+00
-3.838770 -1.482330
-3.838770 0.0000000E+00
-3.838770 1.482330
-1.802440 0.7411700
0.2339000 0.0000000E+00
-1.802440 -0.7417700
-3.748770 -1.360800
-3.748770 1.360800
-9.9999998E-03 0.0000000E+00
-3.738770 -1.353520
-3.738770 0.0000000E+00
-3.738770 1.353520
-1.879390 0.6767600
-2.0000000E-02 0.0000000E+00
-1.879390 -0.6767600

Coated Triangle Geometry File Cont't
4
23 11
67 2
8
1
3
12
10
2
8
11
7
1.000000 1.000000
4
23 11
67 2
8
3
5
14
12
4
9
13
8
1.000000 1.000000
4
23 11
67 2
8
5
1
10
14
6
7
15
9
1.000000 1.000000

Coated Triangle Geometry File Cont't
4
41 11
60 1
8
19
21
3
1
20
17
2
16
1.000000 1.000000
4
41 11
60 1
8
21
23
5
3
22
18
4
17
1.000000 1.000000
4
41 11
60 1
8
23
19
1
5
24
16
6
18
1.000000 1.000000
0
5.7453237E-44
0
1.1210388E-43

Composite Wedge Geometry File
12
28
-1.06000 6.00000E-02
-1.08100 0.116300
-0.726670 0.181300
-0.746670 0.237810
-0.393900 0.302600
-0.413000 0.359000
-6.00000E-02 0.420400
-6.00000E-02 0.483970
0. 0.483970
0. 0.
-6.00000E-02 0.
-6.00000E-02 6.00000E-02
-0.393330 6.00000E-02
-0.393330 0.
-0.726670 0.
-0.726670 6.00000E-02
-1.06000 0.
-0.560000 6.00000E-02
-0.560000 0.241985
-1.07026 8.81900E-02
-1.06000 3.00000E-02
-1.11909 4.95800E-02
-1.11196 9.00000E-02
-1.09857 1.40400E-02
-0.810000 0.100000
-0.893340 0.118710
-0.893340 6.00000E-02
-0.560000 0.150000

Composite Wedge Geometry File Con't

```
4
21 11
15 2
4
1
3
4
2
1.00000 1.00000
4
31 11
25 2
4
3
5
6
4
1.00000 1.00000
4
41 11
35 3
4
5
7
8
6
1.00000 1.00000
4
71 11
1 70
4
11
10
9
8
1.00000 1.00000
```

Composite Wedge Geometry File Con't
4
41 11
35 3
4
14
11
12
13
1.00000 1.00000
4
31 11
25 2
4
15
14
13
16
1.00000 1.00000
4
21 11
15 2
4
17
15
16
1
1.00000 1.00000
4
61 11
12 12
4
18
12
7
19
1.00000 1.00000

Composite Wedge Geometry File Con't
4
21 11
3 3
8
22
17
1
2
24
21
20
23
1.00000 1.00000
4
51 11
8 7
4
25
28
19
26
1.00000 1.00000
4
51 11
6 5
4
27
18
28
25
1.00000 1.00000
4
51 11
6 5
4
1
27
25
26
1.00000 1.00000
0
0.
0
0.