

036307-7-T

**STRUCTURED 2D AND 3D MESHING OF ANTENNA
SURFACES AND VOLUMES: USERS MANUAL VERSION 2**

**Zhigang 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**

July 1999

36307-7-T = RL-2508

PROJECT INFORMATION

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

REPORT TITLE: STRUCTURED 2D AND 3D MESHING OF ANTENNA SURFACES AND VOLUMES: USERS MANUAL VERSION 2

U-M REPORT No.: 036307-7-T

CONTRACT

START DATE: July 1998

END DATE: June 1999

DATE: July 1999

SPONSOR:

S. McCool and K. Trott
Mission Research Corp.
147 John Sims Parkway
Valparaiso, FL. 32580
l.kempel@ieee.org
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: Zhifang Li and J.L. Volakis

Structured 2-D and 3-D Meshing of Antenna Surfaces and Volumes: Users Manual Version 2

Zhifang Li and John L. Volakis
Radiation Laboratory

Department of Electrical Engineering and Computer Science

University of Michigan, Ann Arbor, MI 48109-2122

Email: zhifang@umich.edu, volakis@umich.edu

July 1999

Contents

1	Introduction	3
1.1	New Features for Version 2	3
1.2	Features of the Mesher	3
2	How to Use the Mesher	5
2.1	User-specified geometry	5
2.1.1	Single rectangular patch	5
2.1.2	Single circular patch	5
2.1.3	Rectangular arrays	6
2.2	SurfMesh file pre-supplied	7
2.3	Viewing the surface mesh in MatLab	8
2.4	Output files	8
3	Appendix. Disk Contents	9

1 Introduction

This manual is a supplement to a previous version (see “Theory, Mesher and Users Manual for a Ferrite Finite Element-Boundary Integral Tetrahedral Code” by Arik Brown, Zhifang Li and John L. Volakis, Rad Lab Report 036307-2-T, University of Michigan, May 1998) but can also be used independently. It describes a complete 2-D and 3-D mesher package for antenna patches or slots. All codes are written in FORTRAN 77.

1.1 New Features for Version 2

Because boundary integral (BI) truncation is more popular than the artificial absorber (AA) truncation, this new version considerably facilitates the meshing process for BI. The user specifies the geometric and electric parameters only once, and the codes will automatically run through the surface and volume mesh to generate an IDEAS-compatible universal file. New features of version 2 include:

- For BI truncation, one-step run of surface and volume mesh is available for rectangular and circular geometries;
- The mesher can start either from simple geometry inputs or an outside-supplied SurfMesh file and generate a volume mesh accordingly;
- Infinite/finite array geometry accepted.

1.2 Features of the Mesher

- The meshers can mesh patch or slot antennas of rectangular, circular, or log-periodic geometry;
- The mesh is terminated by either boundary integral (BI) or artificial absorber (AA) truncation;
- For BI truncation, non-uniform surface mesh is available of rectangular or circular geometry (log-periodic is intrinsically non-uniform);
- For AA truncation, meshing is done in two steps: surface first and then growing to volume; all rectangular, circular and log-periodic geometries are accepted;
- For AA truncation, cavity or microstrip configuration are both available; while for BI truncation only cavity configuration is available;
- In meshing the 3-D structure, either prismatic or tetrahedral elements can be chosen;
- Surface mesh can be viewed with Matlab; and Volume mesh is in the form of universal file, which is compatible with commercial software I-DEAS;

- The 3-D mesher is additionally capable of meshing non-planar platforms for circular antennas with BI termination;
- Multiple layers of substrates and/or superstrates can be dealt with;
- Multiple probe feeds coinciding with edges can be specified within the mesher.

In this manual, we will only provide brief usage when the geometry is terminated by BI truncation. For more illustrated examples and detailed instructions on AA truncation, please refer to the manual of Version 1: “Theory, Mesher and Users Manual for a Ferrite Finite Element-Boundary Integral Tetrahedral Code” by Arik Brown, Zhifang Li and John L. Volakis, Rad Lab Report 036307-2-T, University of Michigan, May 1998.

2 How to Use the Mesher

If you do not have any file at hand and would like to specify the geometry from scratch, please read section 2.1. If you already have a surface mesh and would like to expand it to volume mesh, please go directly to section 2.2.

2.1 User-specified geometry

2.1.1 Single rectangular patch

If you would mesh a single rectangular patch, you need to:

1. Compile the code ‘meshrect.f’:
f77 meshrect.f -o meshrect.e
2. Run the mesher by typing: *meshrect.e* and then hit return. Input your geometry as prompted by the code, which is self-explained. A sample file is shown in Table 1.

0	1 = cavity only; 0 = patch/slot
1	1 = patch; 0 = slot
0.5 0.5	Element size (cm) between antenna and cavity wall (x,y)
2 2	Number of elements between antenna and cavity wall (x,y)
0.25 0.25	Element size (cm) of antenna (x,y)
6 4	Number of antenna elements (x,y)
1	1 = Tetra; 0 = Prism
2	Number of substrate layers
0	1 = same thickness; 0 = no
0.1	Thickness (cm) of substrate layer 1 (from bottom up)
0.15	Thickness (cm) of substrate layer 2
1	Material code of substrate layer 1
2	Material code of substrate layer 2

Table 1: Sample input file for mesh of a single rectangular patch.

2.1.2 Single circular patch

If you would mesh a single circular patch, you need to:

1. Compile the code ‘meshcirc.f’:
f77 meshcirc.f -o meshcirc.e
2. Run the mesher by typing: *meshcirc.e* and then hit return. Input your geometry as prompted by the code, which is self-explained. A sample file is shown in Table 2.

```

0      1 = cavity only; 0 = patch/slot
1      1 = patch; 0 = slot
0.75  Element size (cm) of rings between antenna and cavity wall
4      Number of rings between antenna and cavity wall
1.     Element size (cm) of antenna rings
3      Number of antenna rings
1      1 = Tetra; 0 = Prism
1      Number of substrate layers
0.1    Thickness (cm) of substrate layer 1 (from bottom up)
        Material code automatically assigned to be 1 for single layer

```

Table 2: Sample input file for mesh of a single circular patch.

2.1.3 Rectangular arrays

If you would mesh a rectangular array, you need to:

1. Compile the code 'meshrect_arr.f':
`f77 meshrect_arr.f -o meshrect_arr.e`
2. Run the mesher by typing: `meshrect_arr.e` and then hit return. Input your geometry as prompted by the code, which is self-explained. A sample file is shown in Table 3.

This code can also be used to mesh a single rectangular patch.

```

0      all surface nodes resistive? (1 = yes, 0 = no)
1      Patch (= 1) or Slot (= 0)
3 2    Number of patches in x- and y-direction
0.2 0.2 Element size (cm) dX and dY
4 4    Number of edges for every patch (x, y)
2 2    Number of edges between LL patch and cavity wall (x, y)
2      Number of edges between neighboring patches in x direction(>=2)
1      Number of edges between neighboring patches in y direction(>=2)
1      1 = Tetra; 0 = Prism
1      Number of substrate layers
0.1    Thickness (cm) of substrate layers

```

Table 3: Sample input file for mesh of a rectangular array.

2.2 SurfMesh file pre-supplied

If the user already has a surface mesh and would like to grow it to a 3-D volume mesh with prismatic or tetrahedral elements, he needs to have a file called 'SurfMesh' before running the volume mesh. The 'SurfMesh' file should have the format shown in Table 4.

40 278 278 164 164 0	no. of antenna tri, cavity tri, total tri, cavity nodes, total nodes, and absorber tri (for BI, 2nd entry = 3rd entry, 4th entry = 5th entry, and 6th entry = 0)
1 2 19 14	triangle number and the global numbering of its three nodes
...	...
1 .0 -2.0 .0	node number and its (x,y,z) coordinates
...	...

Table 4: Sample SurfMesh file.

Note: antenna triangles should be listed first, followed by non-antenna triangles.

After the SurfMesh file is generated, the user is ready to grow it to 3-D volume mesh. The steps are very simple:

1. Compile the code 'surfproc.f':
f77 surfproc.f -o surfproc.e
2. Run *surfproc.e*. This code reads in the 'SurfMesh' file and modifies it to get ready for the volume mesh.
3. Compile the code 'volproc.f':
f77 volproc.f -o volproc.e
4. Run *volproc.e*. This code conducts the volume mesh.

2.3 Viewing the surface mesh in MatLab

After the mesh has been generated, the 2-D surface mesh can be viewed in Matlab. There are several m files in the directory which can do different tasks. They are:

Setup sets up the screen and reads in the mesh data. Must be issued first prior to other MatLab commands.

Mesh displays the mesh.

GloNod numbers the nodes globally.

LocNod numbers the nodes locally with respect to each triangle.

TriNum numbers the triangular patches.

AngDst plots on a second window the distribution of internal angles in the mesh (this is useful for assessing the quality of the mesh).

Bound displays geometry with mesh in the background.

After the creation of the surface mesh, the user can start MatLab and issue the above commands as needed. Among the commands listed above, the most important one is the **GloNod**. This is a must for specifying feed location in the 'MainIn' file for the input of the 3-D mesher. Another important note is that the user must run 'Setup' first before running the command 'Mesh'.

2.4 Output files

Several auxiliary files will be created after the above procedures. Among them 'SurfMesh' is the 2-D surface file which could be used by Matlab to display the mesh. 'mesh.unv' is the 3-D universal file which is compatible for commercial software I-DEAS.

3 Appendix. Disk Contents

This is a list of directories and all files in this delivery.

DIRECTORY *src* contains all codes and files for the 2-D and 3-D meshers.

- AngDst.m** MatLab code to plot the internal angle distribution in the mesh.
- AngInt** contains information of distribution of internal angles created by surface mesher.
- AntEdg** contains information of antenna edges created by surface mesher.
- ArrayDetector.f** code needed in array meshing.
- Attr** file needed for surface mesh display.
- Bound.m** MatLab code to display surface geometry with mesh in the background.
- CavEdg** contains information of cavity edges created by surface mesher.
- GloNod.m** MatLab code to number nodes globally.
- LocNod.m** MatLab code to number nodes locally with respect to each triangle.
- MainIn** input file for VolumeMesher.f.
- Mesh.m** MatLab code to display the surface mesh.
- MeshDs** file generated for surface mesh display.
- MesherScript** a script file used internally by the mesher.
- Plot.aux** contains the statement 'axis('equal')' in its first line.
- Setup.m** MatLab code to set up the screen and read in the surface mesh data.
Must be issued prior to other MatLab commands.
- SrfEdg** contains information of surface edges created by surface mesher.
- SurfChecker.f** code to check the user-supplied 'SurfMesh' file and modify it to be used by volume mesher.
- SurfIn** input file for SurfMesher.f.
- SurfMesh** surface mesh file created by surface mesher.
- SurfMesh.old** if the SurfMesh file supplied by the user has been modified during the volume mesh, the original SurfMesh file will be saved in SurfMesh.old file.
- SurfMesherBI.f** FORTRAN code of the surface mesher.
- SurfReader.f** code to read in the 'SurfMesh' supplied by user and make necessary modifications.
- TriNum.m** MatLab code to number the triangles in surface mesh.

VolumeMesherBI.f FORTRAN code of the volume mesher.

dm.f FORTRAN code to read in the SurfMesh file and create a decent 'fema.dm1' file before running volume mesh.

fema.dm1 contains the parameter statement for memory allocation.

fema.dm2 contains the parameter statement for memory allocation of circular patch.

mesh.unv universal file generated by volume mesher to view the mesh in I-DEAS.

meshcirc.f code to conduct BI mesh of single circular patch.

meshrect.f code to conduct BI mesh of single rectangular patch.

meshrect_arr.f code to conduct BI mesh of rectangular array.

surfproc.f code to read in a user-supplied 'SurfMesh' file and modify it to be used by volproc.f.

volproc.f code to conduct volume mesh with a pre-supplied surface mesh.

DIRECTORY *document* contains all the manuals and documents for the 2-D and 3-D mesher package.

manual.tex User's manual for the structured 2-D and 3-D mesher package (in LaTeX format).

manual.ps Postscript file of the manual.

example_array example input file for rectangular array mesh.

example_circ example input file for circular patch mesh.

example_rect example input file for rectangular patch mesh.