374968-2-T

M. Carr
J. L. Volakis

Semi-annual Report

Doppler Shift Analysis and Scattering Phenomenology of Rotating Realistic Helicopter Blades Using the Adaptive Optimal-Kernel Method

Sikosky Aircraft Corp.
1201 South Ave
Bridgeport, CT. 06604

December 1997

374968-2-T = RL-2538

PROJECT INFORMATION

PROJECT TITLE:	Fast Integral Algorithms for Rotorcraft Blade Scattering in the VLF and UHF Bands
REPORT TITLE:	Doppler Shift Analysis and Scattering Phenomenology of Rotating Realistic Helicopter Blades Using the Adaptive Optimal-Kernel Method
U-M REPORT No.:	374968-2-T
CONTRACT START DATE:	1 January 1997
END DATE:	31 December 1997
DATE:	December 1997 Semiannual Report
SPONSOR:	Daniel C. Ross Sikorsky Aircraft Corp. Mail Stop B101A 1201 South Ave. Bridgeport, CT. 06604 Phone: (203) 384-7010 Fax: (203) 384-6701 Email: dross@sikorsky.com
SPONSOR CONTRACT No.:	P.O. N5600918
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) 647-2106 volakis@umich.edu http://www-personal.engin.umich.edu/~volakis/
PROJECT PEOPLE:	Michael Carr, Mikhael Smelianskiy, John Volakis

Doppler Shift Analysis and Scattering Phenomenology of Rotating Realistic Helicopter Blades Using the Adaptive Optimal-Kernel Method

Interim Report 4

Michael Carr (mcarr@umich.edu)
John Volakis (volakis@umich.edu)

Radiation Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan at Ann Arbor
Ann Arbor, Michigan 48109

December 10, 1997

I. Introduction

For this project we consider the Doppler analysis of helicopter blade radar backscatter patterns. In our previous interim report, we discussed the time-frequency analysis technique known as the adaptive optimal kernel (AOK). We then applied the AOK analysis to some sample patterns from flat, 2D blades. In this report, we examine more realistic 3D geometries and identify their scattering characteristics. We also describe some of our current work with the adaptive integral method (AIM), a method which can substantially reduce the necessary memory in a moment method solution.

II. Test Geometries

One step up from the flat plate blade approximation is the block blade shown in Figure 1 below. This blade is 6 meters long, .4 meters wide, and .1 meters thick. The blade is made of a PEC material.

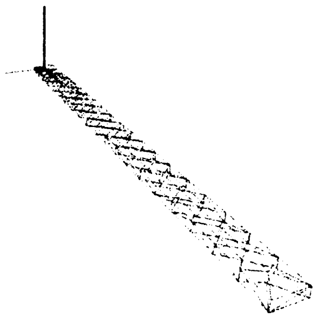


Figure 1: Block Blade Geometry

The blade shown in Figure 2 below approximates one found on a helicopter. The blade is 10.5 meters long, .7 meters wide, and .06 meters thick. The geometry includes a swept portion of the blade—we are most interested in the effects of this "bladelet" on the TFD. This blade is also made of a PEC material.

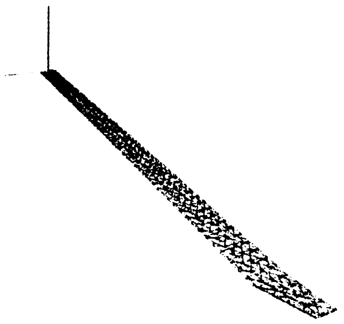


Figure 2: Realistic Blade Geometry

III. Simulation Results

A. Timing

Table 1 and Table 2 contain the timing and memory information for block- and real-blade simulations respectively.

Table 1: Timings for Block Blade

	Block 100MHz	Block 500MHz	Block 900MHz
Unknowns	366	6042	18234
CPU Time (min)	0.72	138	N/A
Memory (MB)	0.535	146	1330

Table 2: Timings for Realistic Blade

	Real 100MHz	Real 300MHz	Real 500MHz	Real 700MHz	Real 900MHz
Unknowns	270	1734	4506	9012	13836
CPU Time	0.5	7.4	61	369	N/A
Memory	0.291	12	81	324	766

B. Time-Frequency Distributions

First, we examine the TFDs of the block blade geometry as shown in Appendix A. Figure A1, showing the 100-MHz block blade TFD, contains little information because the blade is too small (only 2λ) to create a distinct TFD. The speed of the blade's tip is directly proportional to its length, and therefore the maximum frequency shift generated by Doppler analysis will be much lower.

Figure A2 reveals the effect of blade depth on the TFD. The strong specular reflection usually visible at normal incidence has been replaced by constant amplitude TFD because the scattering centers have become blurred along the blade. The usual sinusoidal pattern is again observed as this is a characteristic of the rotating tips. However, the image is no longer a "clean" sinusoid as we have come to expect from our studies of wires and flat plates. The TFD now contains a blurring of the scattering centers along the blade. These new scattering effects were quite puzzling at first until we correlated them to the current distributions on the blade itself.

Before going into further detail on this point, let us first look at the TFDs associated with the realistic blade given in Appendix B. Ignoring for a moment 100 MHz pattern, the TFDs each have characteristic traits that identify them with the blade geometry. But in general, they are now in close resemblance with the crisp sinusoidal behavior of the straight wire TFDs. One interesting behavior is the "bladelet" to the left and right of broadside (ϕ=180°).

Our first impressions were that these bladelets were due to specular reflections from the bent end. However, once we correlated these returns with the current distributions (Figure B5) using MeshView, we concluded that these "bladelets" were actually due to higher order diffraction mechanisms. More specifically, we can conclude that these hot spots are due to a wave that is bouncing between the two straight edges and then decays as it travels toward the other end of the blade.

Several years ago, these diffraction mechanisms were referred to as "edge waves" [Sikta and Peters]. In their study, edge-wave diffraction coefficients were heuristically introduced to match moment method data. These models were typically poor. We believe that propagation factors other than free space are associated with the "edge waves" to account for the wave delay and decay as it propagates down the blade. From the correlation of the current distributions and the TFDs, we can assert two reasons for the previously poor performance of Sikta's edge-wave model. The first is the actual phenomenology of the mechanism (i.e. being a "mode/waveguide" phenomenon different from that assumed in the edge-wave model) and the second reason stems from the narrow angular sector confining the scattering behavior of this mechanism. This mechanism should be easily suppressed by edge or surface treatments.

Now we return to the block blade analysis. The above phenomenon is indeed very crisp when seen in the realistic blade TFDs but becomes diffused for the block blade. Because of the blade's finite thickness of $\lambda/6$, leakage across the side edges prevents the crisp mode-like bouncing behavior of the currents.

The bladelet at the end of the realistic blade also appears to scatter energy away from the origin at all incidence angles. This is shown by the increased energy in the TFD occurring after just after normal incidence (see Figure B1).

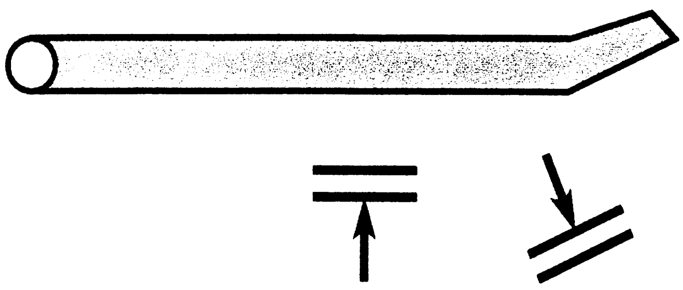


Figure 3: Scattering from Realistic Blade

IV. AIM Approach

We are well into AIM implementation using object-oriented code written in Fortran 90/95. Currently, we are examining the implementation of the CG-FFT portion of the code, where convolution involving the free-space Green's function is replaced by simple vector multiplication in the frequency domain. Once this portion of the research is completed, we must then consider the procedure of mapping the arbitrary blade geometry's moments to a regularly spaced grid. These are the two major components of the AIM implementation; we expect that once these hurdles are overcome, the remainder of the code will be straightforward.

V. Conclusions

The time-frequency distribution of realistic geometries prompts some consideration of the scattering centers' behaviors. Even a small change in the geometry may cause major changes in the TFD—notably, new specular reflections in the case of bladelets, and a spreading of energy in the case of a very thick blade.

APPENDIX A – BLOCK BLADE TIME-FREQUENCY DISTRIBUTIONS

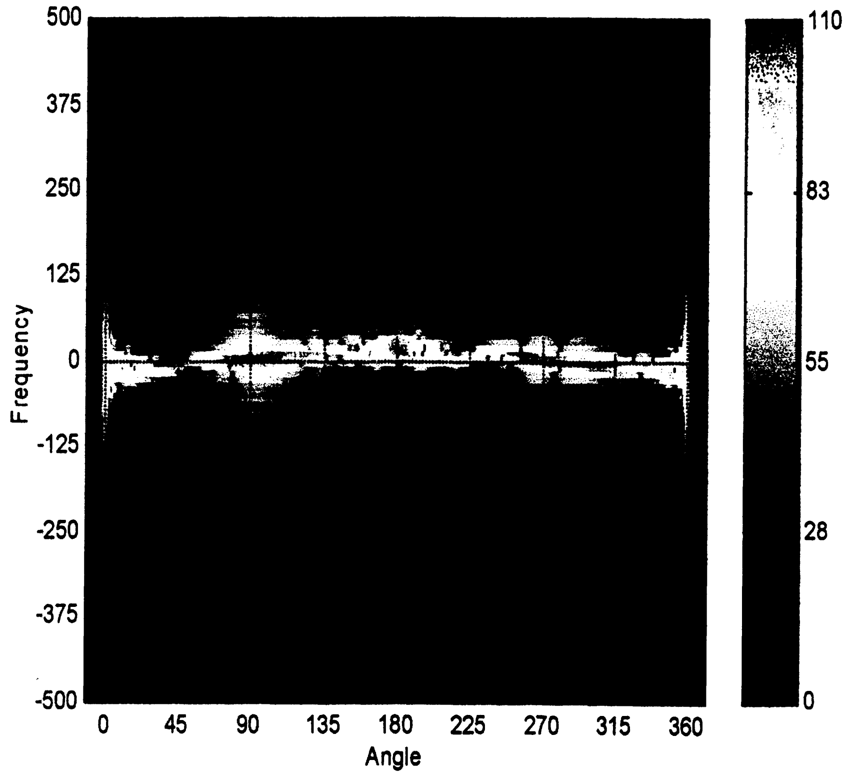


Figure A1: TFD of Block Blade at 100 MHz

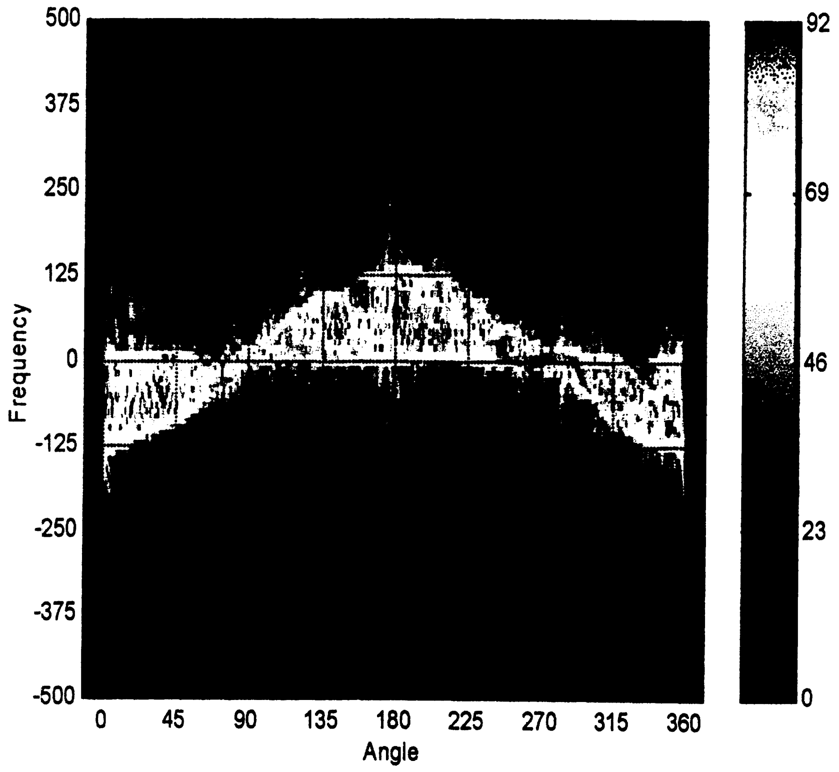


Figure A2: TFD of Block Blade at 500 MHz

APPENDIX B – REALISTIC BLADE TIME-FREQUENCY DISTRIBUTIONS

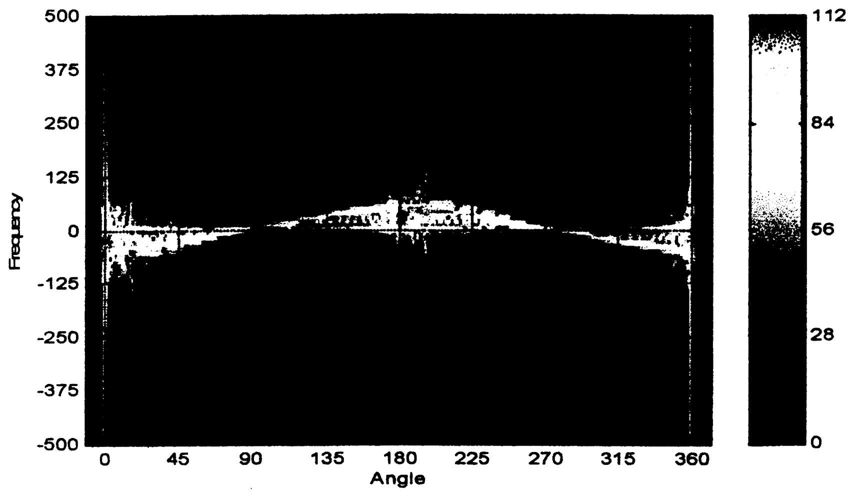


Figure B1: TFD of Realistic Blade at 100 MHz

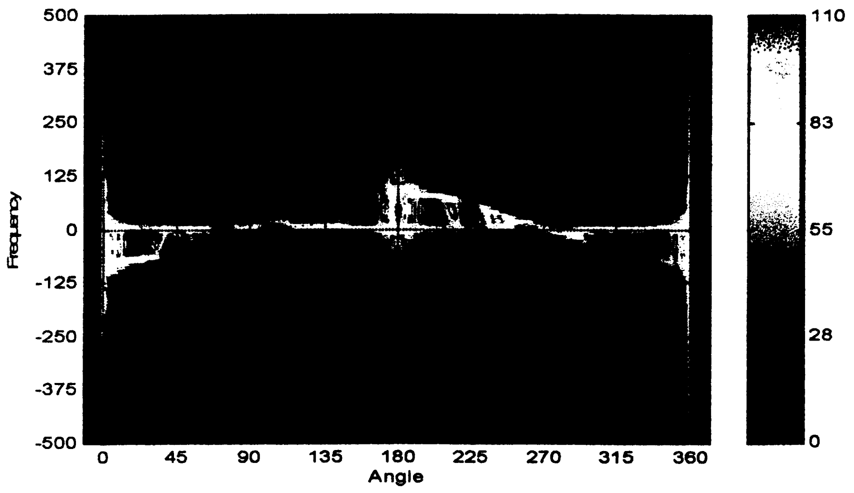


Figure B2: TFD of Realistic Blade at 300 MHz

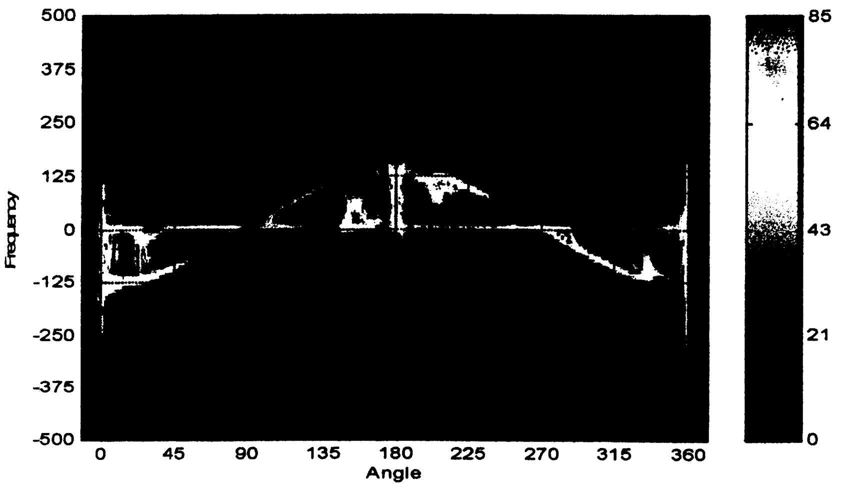


Figure B3: TFD of Realistic Blade at 500 MHz

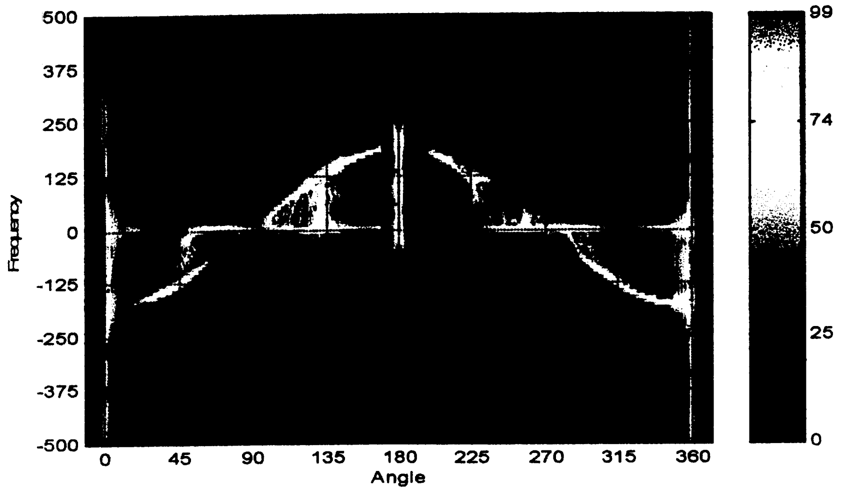


Figure B4: TFD of Realistic Blade at 700 MHz

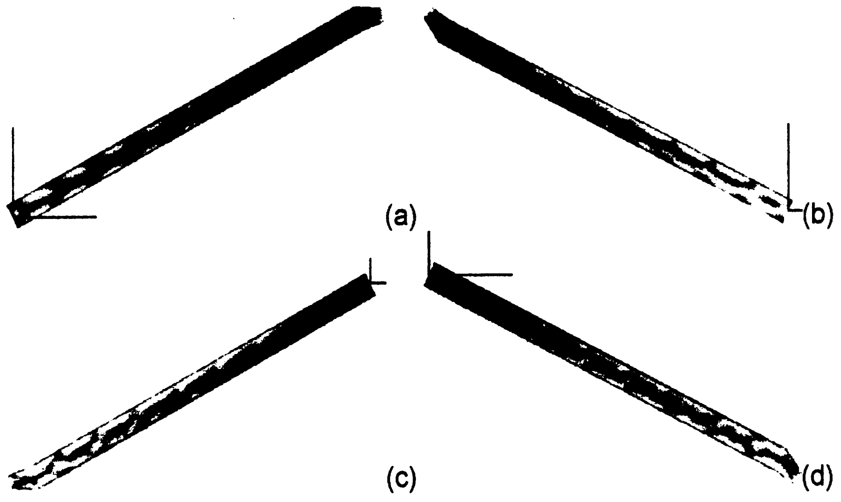


Figure B5: Currents on Realistic Blade at 500 MHz

APPENDIX C – PTRI AND AOK DOCUMENTATION

For the simulations in this project, a moment method solver was developed called **PTri**. **PTri** solves for the surface currents by applying the standard RWG basis functions. Then E-fields resulting from these currents are then calculated. The user has the option of storing the surface currents in a file—this file can be visualized using **MeshView**, a Radiation Laboratory visualization package.

The code has been parallelized and optimized to run on any MIPS (Silicon Graphics) machine, but can be easily converted to run on other architectures.

To begin a simulation, there are several initialization procedures which must be followed:

1. Prepare the geometry in Tricode file format (see Appendix D).
2. Obtain the number of nodes, edges, and faces present in the geometry. You can do this by running PTRI, and then breaking (Ctrl-C) immediately after the count is displayed.
3. Edit the file "dim.inc" and specify the number of nodes, edges, and faces.
   - Set **maxnod** greater than the number of nodes
   - Set **maxedg** greater than the number of edges
   - Set **maxtri** greater than the number of triangles
4. Create a file called "incident.dat" containing the incident angles in the following format:

```
<n=Number of look angles>
<phi[1] in degrees> <theta[1] in degrees>
<phi[2] in degrees> <theta[2] in degrees>
<phi[3] in degrees> <theta[3] in degrees>
        .
        .
        .
<phi[n] in degrees> <theta[n] in degrees>
```

5. Edit the file "dim.inc" and specify the number of look angles.
   - Set **maxla** greater than the number of look angles
6. Recompile the program by typing "make".
7. Run the program by typing "ptri".

```
Compiled with multiprocessor support.
 Running with           2 processors.

 Mesh file?
<mesh file>
 Output file?
<output file>
 Incident wave polarization (degrees)?
<incident wave polarization>
 Output currents (0=No, 1=Yes)?
<0 or 1>
 Reading incident angles from incident.dat
     # of nodes:          124
     # of edges:          366
 # of triangles:          244
```

8. After the program finishes, examine the output file for the results in the following format:

```
<phi[1]> <theta[1]> <E_phi_real[1]> <E_phi_complex[1]> <H_theta_real[1]>
       <H_theta_complex[1]> <rcs[1]>
<phi[2]> <theta[2]> <E_phi_real[2]> <E_phi_complex[2]> <H_theta_real[2]>
       <H_theta_complex[2]> <rcs[2]>
<phi[3]> <theta[3]> <E_phi_real[3]> <E_phi_complex[3]> <H_theta_real[3]>
       <H_theta_complex[3]> <rcs[3]>
    .

    .

    .
<phi[n]> <theta[n]> <E_phi_real[n]> <E_phi_complex[n]> <H_theta_real[n]>
       <H_theta_complex[n]> <rcs[4]>
```

9. If desired, run the perl script **stripang** to split the output file into its phi and theta components. The **stripang** script will read the input file(s) it is given and produce files with .phi and .theta extensions containing the components of the phi- and theta-polarized fields respectively.

   ```
   stripang <ptri output file>
   ```

Performing Time-Frequency Analysis

The TFD is generated by a code written at Rice University called **aok**. Follow this procedure to execute the code:

```
ADAPTIVE OPTIMAL-KERNEL (AOK) TIME-FREQUENCY REPRESENTATION

   Version 4.0

Name of input file?
<name of input file>
Name of output file?
<name of output file>
Number of signal samples in input file?
<number of look angles>
Length of sliding analysis window?  (power of two, no larger than 256)
  (Number of samples along each dimension of the STAF)
64
Number of output frequency samples per time-slice?  (power of two)
128
Time increment in samples between time-slice outputs?
1
Normalized volume of optimal kernel?
  (Typically between 1 and 5)
3
```

The values given above are only the most generally used. For more information, look in the file **aok4.GUIDE** included with the package.

The output file will contain the TFD of the signal--to plot the TFD using MATLAB, use the included MATLAB .m file, **plotlogaok**, which formats the TFD using a log color scale and attaches the correct axis labelling.

APPENDIX D – TRICODE FILE FORMAT

TRICODE file format – The following document lists the file format created by Pam Haddad and expected by TRICODE/MOMFREE, MOMJET, AIMFREE, AIMJET, and others based on TRICODE.

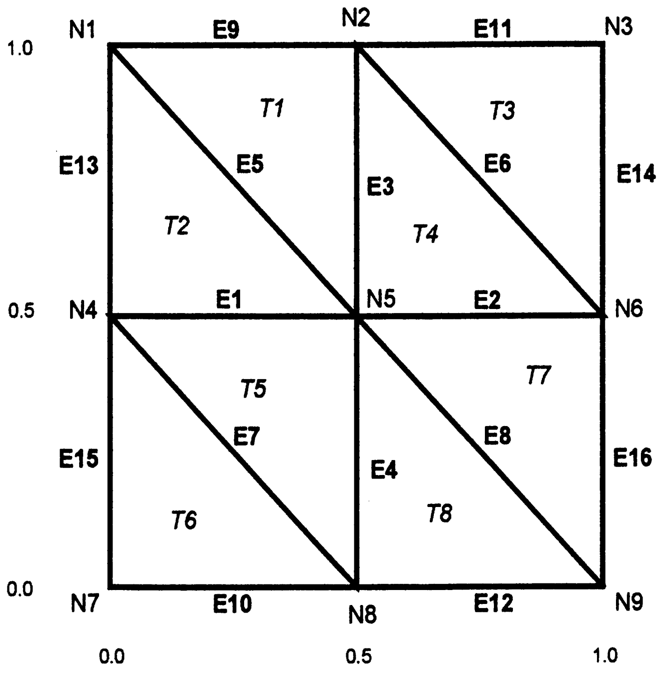
1. The first line of the file holds the number of nodes and triangles.
2. The following lines give the x, y, and z coordinates (in order) of each node.
3. The next line holds the number of edges and the number of exterior edges.
4. Following that line are the interior edges, made up of pointers to the nodes. Note that the first node is indexed as 1 (one) and not 0 (zero). This format is more compatible for Fortran. The order in which the nodes are listed is unimportant.
5. The exterior edges follow in the same format.
6. Next are the triangles, made up of pointers to the edges. Note that the first edge is indexed as 1 (one). The order in which the edges are listed is unimportant.

Following is the format displayed in a programmer-friendly manner.

```
<n=number of nodes> <t=number of triangles>
<node #1 x coord> <node #1 y coord> <node #1 z coord>
<node #2 x coord> <node #2 y coord> <node #2 z coord>
  .
  .
  .
<node #n x coord> <node #n y coord> <node #n z coord>
<e=number of edges>
<ex=number of exterior edges>
<interior edge #1 start node> <interior edge #1 end node>
<interior edge #2 start node> <interior edge #2 end node>
  .
  .
  .
<interior edge #(e-ex) start node> <interior edge #(e-ex) end node>
<exterior edge #(e-ex+1) start node> <exterior edge #(e-ex+1) end node>
<exterior edge #(e-ex+2) start node> <exterior edge #(e-ex+2) end node>
  .
  .
  .
<exterior edge #(e) start node> <exterior edge #(e) end node>
<triangle #1 edge1> <triangle #1 edge2> <triangle #1 edge3>
<triangle #2 edge1> <triangle #2 edge2> <triangle #2 edge3>
  .
  .
  .
<triangle #t node1> <triangle #t node2> <triangle #t node3>
```

EXAMPLE:

```
9 8
0.0 0.0 0.0
0.5 0.0 0.0
1.0 0.0 0.0
0.0 0.5 0.0
0.5 0.5 0.0
1.0 0.5 0.0
0.0 1.0 0.0
0.5 1.0 0.0
1.0 1.0 0.0
16 8
4 5
5 6
2 5
5 8
1 5
2 6
4 8
5 9
1 2
7 8
2 3
8 9
1 4
3 6
4 7
6 9
 9  3  5
 5  1 13
11 14  6
 6  2  3
 1  4  7
 7 10 15
 2 16  8
 8 12  4
```


N1
E9
N2
E11
N3
1.0
T1
T3
E13
E5
E3
E6
E14
T2
T4
0.5
N4
E1
N5
E2
N6
T5
T7
E15
E7
E4
E8
E16
T6
T8
0.0
N7
E10
N8
E12
N9
0.0
0.5
1.0