USER'S MANUAL FOR AIMJET

Hristos T. Anastassiu, Mikhail Smelyanskiy,
and John L. Volakis
Radiation Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
1301 Beal Avenue
Ann Arbor MI 48109-2122
FAX: (313) 647-2106
hristosa@engin.umich.edu
msmelyan@engin.umich.edu
volakis@engin.umich.edu

35675-1-T = RL-2494

1 Introduction

Aimjet is a FORTRAN77 code which uses the Adaptive Integral Method (AIM) [1] to model electromagnetic scattering from perfectly conducting jet engine models. It is based on a free space Moment Method code written by Pamela Haddad at the Radiation Laboratory in 1991, on a free space AIM code developed by Hristos Anastassiu in 1995 and a jet engine MoM code written by Hristos Anastassiu in 1997.

Section 2 of this manual gives a brief description of AIM applied to the engine problem. Section 3 describes the code operation, and section 4 presents a few examples of Radar Cross Section calculations.

2 AIM applied to the jet engine problem

A major reason for an emphasis on the EM characterization of jet engines is their critical importance in determining the airplanes's reflectivity and for target identification. Even though the engine is a small component of the overall structure, its complexity and visibility in the front sector of the aircraft make its contribution a large component of the overall airframe scattering. Moreover, the rotating engine blades cause unavoidable doppler shifts which are important for target identification purposes. The latter can be an important discriminator for both civilian and military applications and more reliable than other methods.

As applied to electromagnetic scattering problems, the Moment Method (MoM) has proven to be a very accurate integral equation technique. However, its brute force application to the jet engine problem is not feasible due to excessive electrical size of the structure. To reduce the CPU time and storage requirements down to manageable levels, the inherent blade periodicity of the jet engine was exploited to show that the computational domain can be reduced down to a single blade or engine "slice" [2, 3].

In free space we have demonstrated [2] that for large scatterers the Adaptive Integral Method (AIM) is much more efficient than MoM in terms of memory and CPU time, yet it retains MoM accuracy. Therefore, it is expected that application of AIM to the jet engine

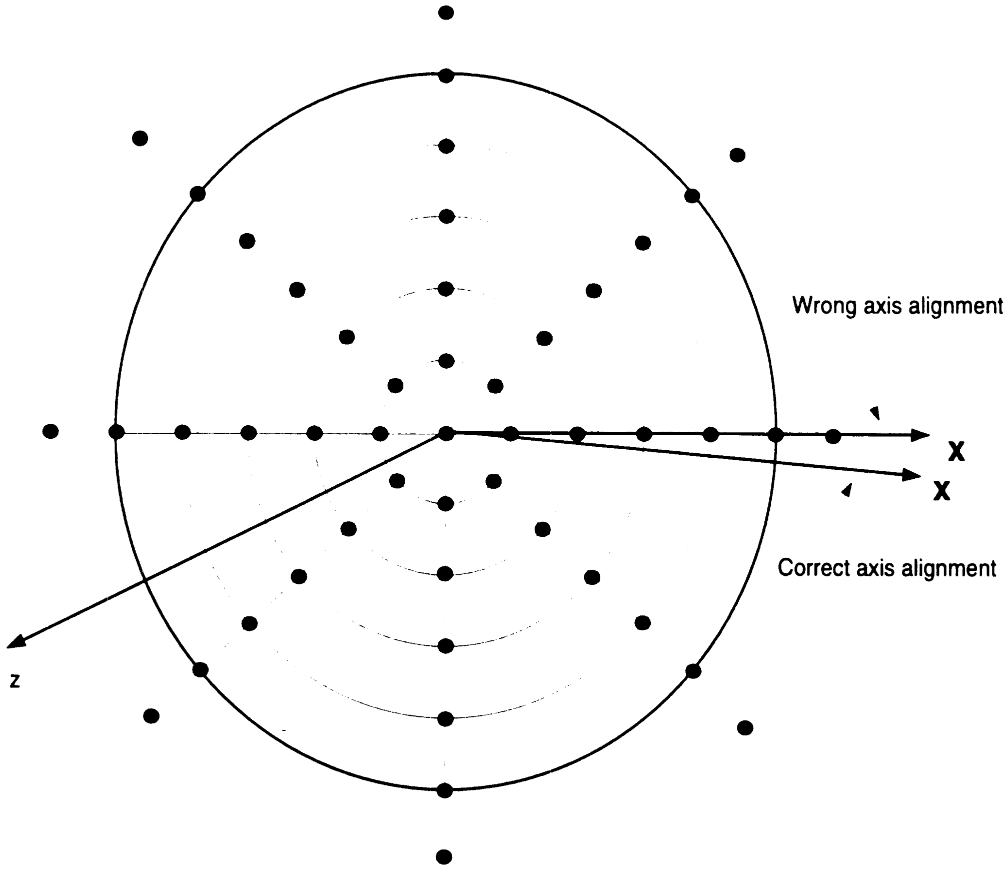

Figure 1: Cylindrical grid (front view).

problem will demonstrate similar merits. Although the application of the slicing scheme greatly enhances the efficiency of MoM, the benefits for AIM are not that impressive without substantial modifications of the method for its application to this structure. The main reason is that the Toeplitz property of the Green's function matrix can only be retained by extending the rectangular grid over the *whole* scatterer, and not over a single slice. The details of this problem are explained in [2].

To avoid using a global rectangular grid and, hence, large dimensions for the FFT, the only possible solution is to construct a grid that still retains the Toeplitz property for the Green's function matrix [G]. Such a grid is not rectangular, but cylindrical, as depicted in Fig. 1.

In cylindrical coordinates the Green's tensor is not Toeplitz along the ρ direction, therefore the FFT can only be used along the ϕ and z directions, meaning that matrix-vector product calculations are slower than in the case of the rectangular grid. This drawback of the cylindrical grid is balanced by the smaller size of the FFT dimensions, since the computational domain is reduced down to a single slice.

Tha mathematical details of the implementation are out of the scope of this manual, but all essential information can be found in [2].

3 Code Operation

3.1 Code structure

Aimjet models the scattering from metallic cylindrical jet engine inlets terminated by an arbitrary, PEC, cylindrically periodic structure. A simplified geometry is illustrated in Fig. 2, where the termination is a cylindrical hub. A plane wave is assumed to illuminate the open end and couple to waveguide modes in Region 1, propagating towards the termination (Region 2). Each of these modes is used as an excitation for AIM applied to one periodic sector of the termination (see Figs. 3, 4). Solving the linear system via AIM yields the equivalent currents on the termination, and in turn the scattered field within the inlet. To extract the coefficients of the scattered modes propagating towards the open end an integration [2, 4] is performed at a hand-off plane, which is perpendicular to the inlet axis and lies at a fraction of a wavelength in front of the termination (see Fig. 5). Each of the scattered modes propagates to the open end and radiates in free space. The overall modal contribution yields the Radar Cross Section (RCS) of the engine.

To exploit the inherent periodicity of the engine, the waveguide modes must have exponential, and not trigonometric dependence on ϕ. Immediate consequence of this is that the modal orders may be negative, positive or zero, and there is no distinction between "odd" and "even" behavior, as opposed to the Mode Matching code that has been developed in the past [6].

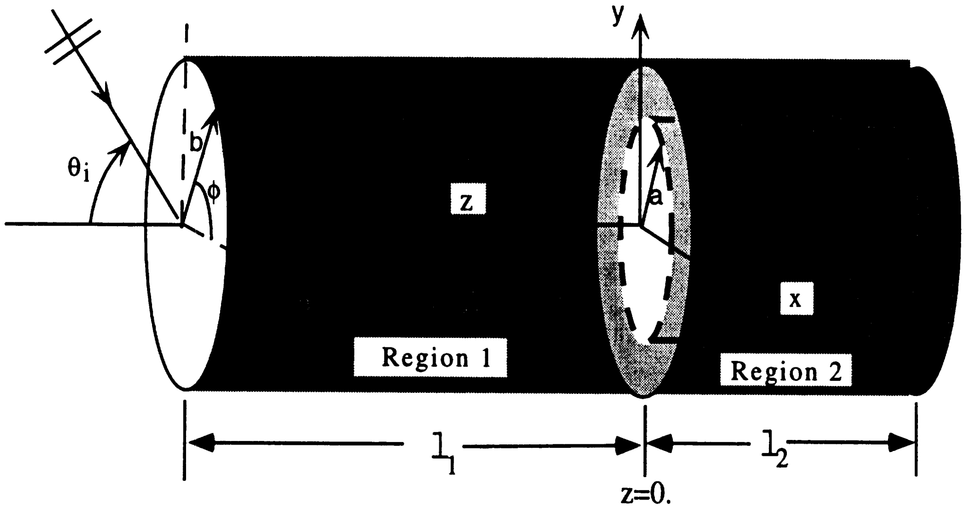

Figure 2: Cylindrical inlet terminated by a cylindrical hub

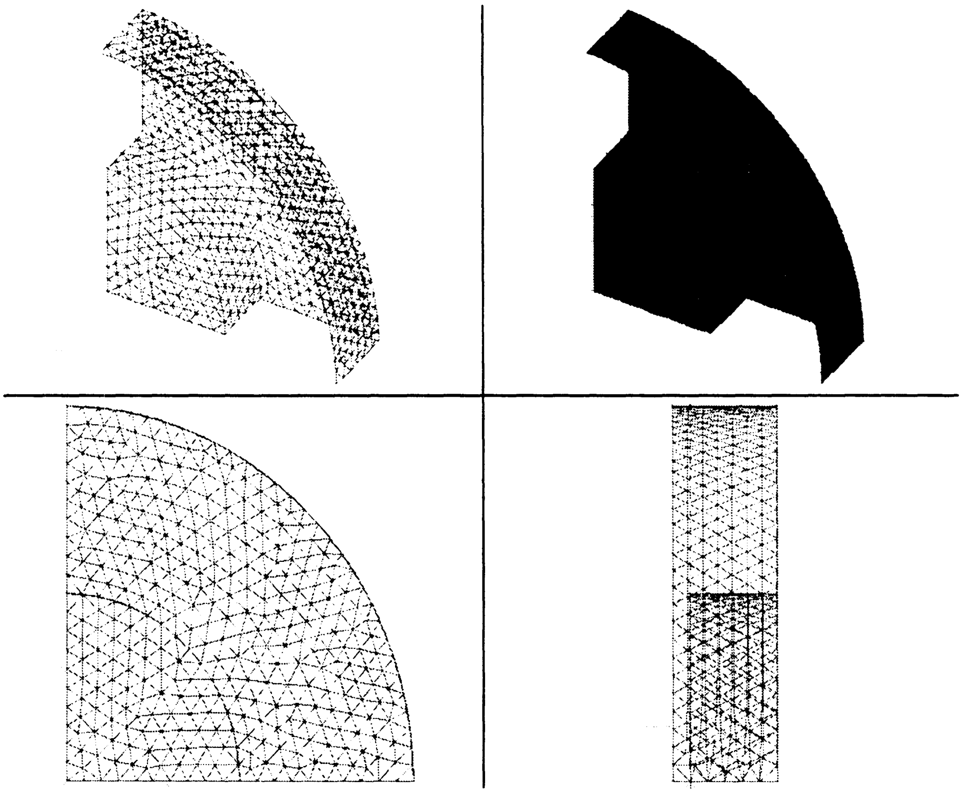

Figure 3: A quarter of a hub geometry.

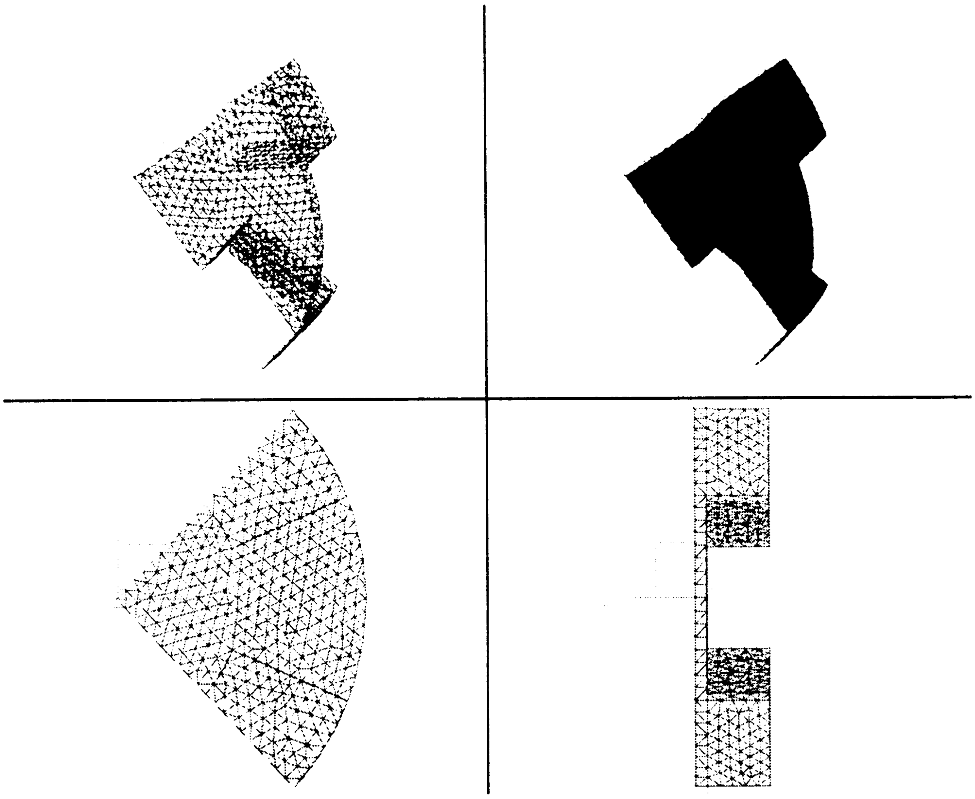

Figure 4: A quarter of a blade geometry.

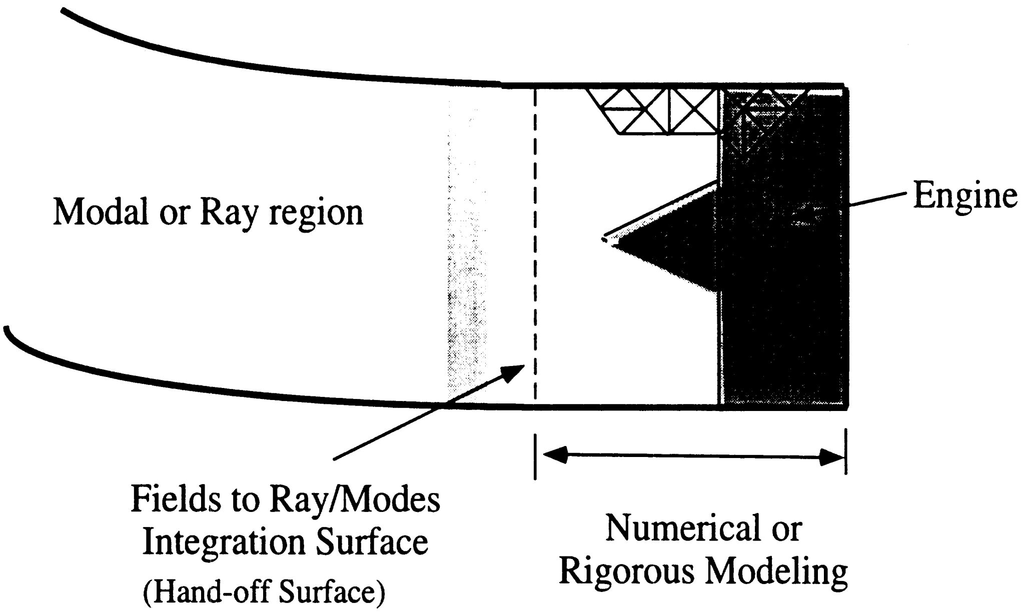

Figure 5: Decomposition of the computational domain.

3.2 Input file

The input file comprises of a discretized sector of the termination, including the waveguide wall up to the hand-off plane. Typical input geometries are illustrated in Figs. 3, 4. The current version of Momjet requires an edge-based format of the geometry file. To illustrate the file format we consider a $0.2\lambda \times 0.2\lambda$ flat plate (see Fig. 6), lying on the xy plane and centered at the origin. This geometry is not typical for the code, but its simplicity is exploited to clarify the meshing format of the input file. The geometry file has the form

9 8	($N_{nodes} = 9$, $N_{triangles} = 8$)
.1 .1 .0	(the next N_{nodes} entries are
.1 -.1 .0	the cartesian coordinates of the nodes)
-.1 .1 .0	
-.1 -.1 .0	
.1 .0 .0	
.0 .1 .0	
-.1 .0 .0	
.0 -.1 .0	
.0 .0 .0	
16	(total number of edges N_{edges})
8	(number of exterior edges N_{ext})
7 8	(next $N_{edges} - N_{ext}$ entries
5 6	give the pair of nodes defining the edges)
5 9	
6 9	
3 9	
7 9	
8 9	
2 9	
4 8	
4 7	
1 6	
1 5	
3 7	
3 6	
2 8	
2 5	
1 9 10	(the remaining entries give the
2 11 12	triangle connectivity. Each line
2 3 4	entry contains 3 numbers corresponding
5 6 13	to the edge numbers forming the triangle
4 5 14	in a clockwise order)
1 6 7	
8 15 7	
16 8 3	

To obtain this format:

• Create the mesh geometry using a meshing package such as IDEAS and export it to a universal file called, for instance, geo-file.unv.

• Convert geo-file.unv to a converter file, called cnv, by running program u2c-new2.f.

• Convert cnv to the desired format geo-file by running program c2p-fast.f. Both u2c-new2.f and c2p-fast.f are available in the same directory with the main code.

3.3 Preprocessing

• In module dim.inc set maxnod greater or equal to the number of nodes in geo-file.

• In module dim.inc set nx, ny, nz in accordance to the cylindrical grid structure. nx is the number of grid points along the ρ direction, ny is the number of grid points along the ϕ direction and nz is the number of grid points along the z direction. The number of grid points is determined by the grid steps and the dimensions of the structure. A typical maximum grid step is chosen to be about 0.05λ.

• Compile the preprocessing code by typing **make -f Makestartjet**.

• Run the preprocessing code by typing **startj**.

• While running **startj** enter the mesh file name (geo-file) and the AIM near field threshold in wavelengths. A typical value of the threshold is 0.6 wavelengths, but for most geometries decent accuracy can be retained for thresholds as low as 0.3 wavelengths.

• The output of **startj** is the number of near field element pairs (nonzero) and also the FFT orders orx2, ory2, orz2 along the three axes ρ, ϕ, z.

• It module dim.inc set nonzero, orx2, ory2, orz2 according to the output of **startj**.

• In module param.inc set

* in=number of *nonnegative* modal orders in hollow section (Region 1).

* ii=number of modes per order. Chosen to include at least all propagating modes.

* ngas=number of points per subinterval for hand-off plane integration.

* nsym=number of periodic sectors (slices) of engine termination.

* nangle=number of different incidence angles (θ_{inc}).

* bgas=Gaussian integration tolerance (at the hand-off plane).

* exrho=2^{intrho}=number of integration subintervals along the ρ direction.

* exphi=2^{intphi}=number of integration subintervals along the ϕ direction.

* zhandoff=z location of the hand-off plane (the tip of the termination is assumed at $z = 0$.

*str=grid step along ρ direction (in wavelengths). Usually set to 0.05λ.

*stp=grid step along ϕ direction (in radians). Chosen so that the maximum distance between any two grid points is less than a given threshold (e.g. 0.05λ).

*stz=grid step along z direction (in wavelengths). Usually set to 0.05λ.

3.4 Compilation and running

- Compile the code by typing **make -f Makeaimjet**.
- Run the code by typing **aimj**. The interaction is self-explanatory. See section 4 for further details.

3.5 List of subroutines

1. Initialization codes:

u2c-new2.f: converts geo-file.unv to a converter file, called cnv.

c2p-fast: converts cnv to the desired format geo-file.

2. Subroutines in Makefile:

Aimjet.f: main code.

ac-intcyl2.f: performs a double surface integration over two triangles of a function in local area coordinates.

an-int1.f: performs an analytical double surface integration over the singular part of the integrand.

edge-len.f: updates the edge length table given the edge numbers of the source and observation triangles.

excitcyl.f: computes elements of excitation vector (right hand side of linear system).

prepfill.f: Includes several subroutines associated with AIM calculations. Specifically it includes prepaimcyl.f which calculates the mesh to grid mapping coefficients, green.f which computes the Green's function matrix and its 2D FFT, fillzaimcyl.f which calculates elements of the impedance matrix, nearaim.f which calculates the near field interactions, farraim.f which calculates the far field interactions, and several auxiliary subroutines, i.e. multi1, multi2, discrete, regmom, regmomdiv, prodmix, vemult and FOURN. See code listing for further information on the function of each subroutine. FOURN is the Numerical Recipes FFT subroutine, which must be replaced by a parallel scheme for improvement of the code efficiency.

bcgcyl.f: Solves the AIM linear system using the Biconjugate Gradient Method (BiCG). The currenrt BiCG subroutine is valid only for symmetric systems, which is not the general case when the periodicity "slicing" scheme is exploited. For general terminations this subroutine must be replaced by a non-symmetric iterative scheme.

get-R.f: reads in triangle resistivity values from file specified by user. According to Pamela Haddad, author of the basic MoM code, the algorithm has not been validated for non-metallic scatterers.

get-mesh1.f: reads in mesh data from file specified by user.

get-verts.f: returns the vertices of a triangle.

normal.f: constructs normal vector to a triangle.

ops.f: various small functions and subroutines performing auxiliary mathematical calculations.

pot-ints-ss.f: computes the potential integrals for uniform and linearly varying surface sources distributed on a planar polygon S. Dependent subroutine in the same module: i-params.

pwcyl5.f: returns the value of $(\mathbf{r}-\mathbf{r}_m)\cdot\mathbf{E}_i$, where $\mathbf{E}_i$ is the incident cylindrical mode and $\mathbf{r}_m$ is the position vector to the m^{th} vertex of the observation triangle. Dependent subroutines in the same module: integration, discreteap, fieldap, vectintcyl, ffargcyl, dyadfree, cGsldx, GAULEGX, cGsldy, GAULEGY. See code listing for more information on the function of each subroutine.

resfnc.f: returns the value of the integrand of the resistive term $(\mathbf{r}-\mathbf{r}_m)\cdot(\mathbf{r}-\mathbf{r}_n)$, where $\mathbf{r}_m, \mathbf{r}_n$ are the position vectors to the m^{th}/n^{th} vertices of the given observation triangle.

xsintcyl.f: performs a single surface integration of a function over a triangle in local area coordinates.

solve2.f: linear system solvers.

update-signs.f: updates the triangle sign and edge use tables for an input observation interior edge and an input source interior edge.

usr-datacyl.f: prompts user for, and reads in, data pertaining to file names, incident field, rcs observation cut via standard I/O.

vectint.f: performs a single surface integration of vector function over a triangle.

inverd.f, invers.f: sets of subroutines for the inversion of linear systems.

cylinsubr2.f: includes several subroutines and functions related to the modal analysis of the hollow section (duct). Subroutines included: excitvec, radcrosec, vemult, modesl. See code listing for more information on the function of each subroutine.

blockdata.f: initializes common block which contains the local area coordinate integration points and weights used in all integration routines.

4. Auxiliary files:

const.inc: includes various constants.

dim.inc: includes uniform array dimensioning.

param.inc: includes modal and integration parameters.

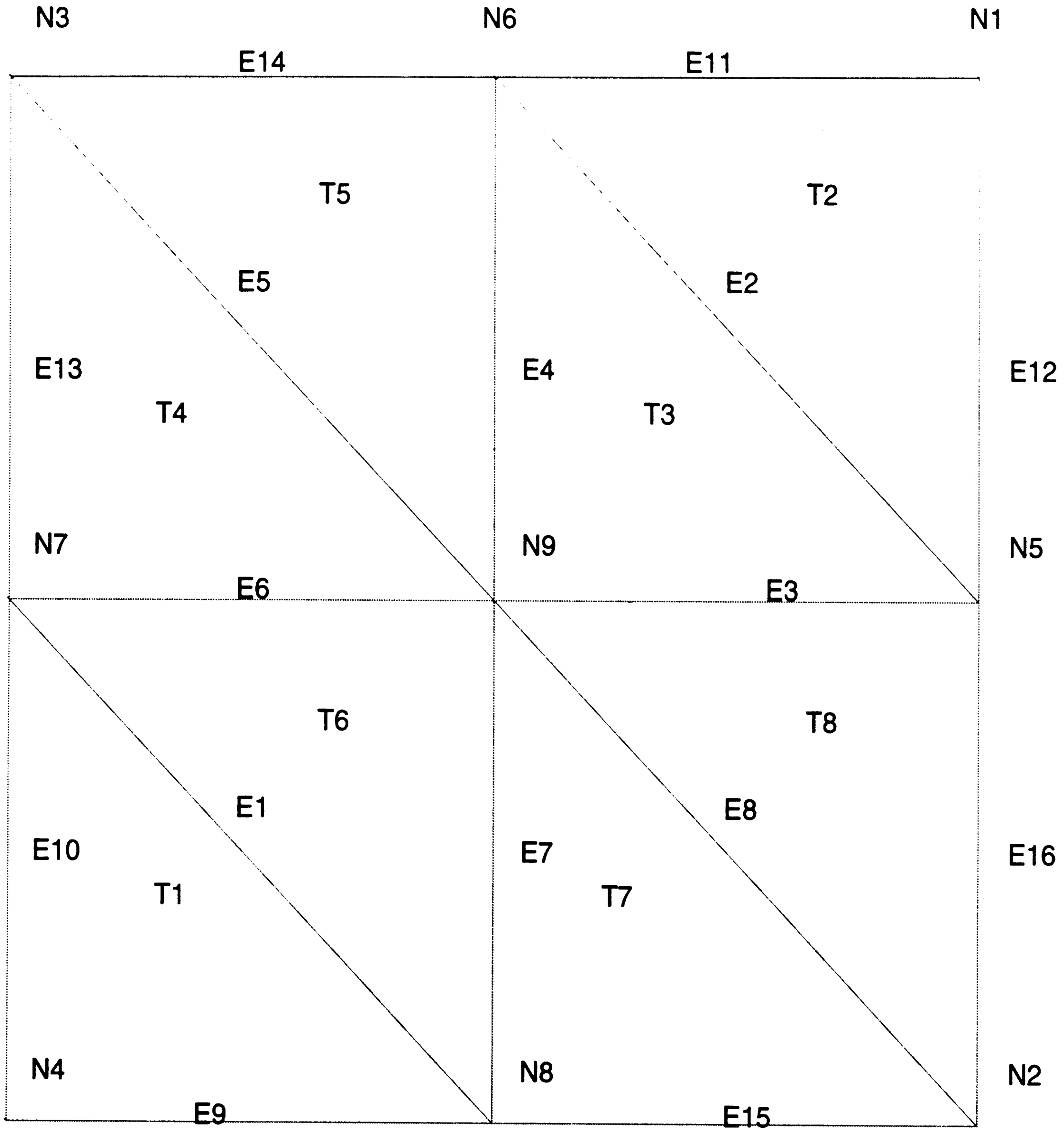

Figure 6: Mesh with node, edge and triangle numbering.

4 Examples

This example is of a flat plate termination of radius λ. To apply the periodicity slicing scheme only half of the termination is meshed. The length of the hollow region is λ and the hand-off plane is located 0.1λ in front of the termination. The maximum radial, azimuth or axial distance among grid points is 0.05λ, the expansion order is $M = 3$ and the near field threshold is set equal to $r_{thr} = 0.7\lambda$. For testing purposes, only propagating modes of order 0 are used in the waveguide. To account for a 0.05λ grid step along all threee directions, in dim.inc set nx=26, ny=70 and nz=7 (grid point numbers in the ρ, ϕ and z directions respectively). Running **startj** with a 0.7 wavelength threshold yields nonzero=191355, orx2=64, ory2=256, orz2=16. These values must be set in module dim.inc before code operation. The param.inc file should read

```
integer ii,in,iis,ii2,ii4
integer ngas,nsym,nangle
integer intrho,intphi,exrho,exphi
real*8 bgas,zhandoff
parameter(in=1,ii=3)
parameter(iis=in*ii)
parameter(ii2=2*iis)
parameter(ii4=2*ii2)
parameter(ngas=3)
Parameter (nsym=2)
Parameter (nangle=121)
parameter(bgas=1.0d-2)
parameter(intrho=2)
```

```
    parameter(intphi=2)
    parameter(exrho=2**intrho)
    parameter(exphi=2**intphi)
    parameter(zhandoff=0.1d0)
    parameter(pifix=3.141592654d0)
    parameter(str=0.05d0,strh=0.5d0*str)
    parameter(stp=pifix/31.0d0,stph=0.5d0*stp)
    parameter(stz=0.05d0,stzh=0.5d0*stz)
```

We emphasize that in general in and ii are chosen so that all propagating modes are included. In our case we set in and ii very low only for testing purposes. The code output yields the propagation characteristics of each mode, hence running the code itself for a few test cases eventually provides the correct values of in and ii.

For this geometry the input reads:

```
    Enter inlet radius in wavelengths
    1.0d0
    Enter mesh file name:
    disk-1w-2
    Enter length of hollow section in wavelengths
    1.0d0
    Enter 1 for inclusion of rim
contribution and 2 for non-inclusion
    2                   !(not including Ufimtsef currents on the rim)
    Enter 1 for inclusion of evanescent modes
```

and 2 for exclusion

2

Enter measurement frequency in GHz

0.3d0 !(to obtain RCS in dB/λ^2)

Enter near field threshold in wavelengths

0.7d0

The output reads:

TM modes (n,m,krho,prop./evan.)

---

0 1 2.40482565831768 1 mode indices, eigenvalues in λ^{-1}

0 2 5.52007838590681 1 1 for propagation and -1 for evanescence

0 3 8.65372785443314 -1

TE modes (n,m,krho,prop./evan.)

---

0 1 3.83170597025498 1

0 2 7.01558679227743 -1

0 3 10.17346810959348 -1

Incident mode order= 0 !(working with mode order=0)

Order of system = 575

No. of triangular elements = 401

Slice 0 !(working with slice 0)

```
...

Phi-phi Polarization
1.000000000000000E-02 -121.047319669989
```

RCS vs. θ_{inc} in degrees

```
.51 -52.7482157063922
1.01 -40.8889918329867
1.51 -33.9207128371362
2.01 -28.9770105579115
2.51 -25.150081887174
3.01 -22.0336614785254
3.51 -19.4104360759395
4.01 -17.150449664679
4.51 -15.1697659720945
5.01 -13.4110150873241
...
Theta-theta Polarization
1.000000000000000E-02 -114.692693414436
.51 -46.3946089608898
1.01 -34.5383648650336
1.51 -27.5750249452714
2.01 -22.6382194275045
2.51 -18.8201429361788
3.01 -15.7145271891473
3.51 -13.1040553080986
4.01 -10.8587669470665
4.51 -8.89472076651106
```

```
5.01 -7.15454102186346
...
```

References

[1] E. Bleszynski, M. Bleszynski and T. Jaroszewicz, "AIM-Adaptive integral method compression algorithm for solving large scale electromagnetic scattering and radiation problems" *Radio Science*, vol. 31, no. 5, pp. 1225-1251, Sept.-Oct. 1996.

[2] H.T. Anastassiu, *Electromagnetic Scattering from Jet Engine Inlets Using Analytical and Fast Integral Equation Methods*, Ph. D. Thesis, University of Michigan, 1997.

[3] H.T. Anastassiu, *User's Manual for MOMJET*, University of Michigan, 1997.

[4] D. C. Ross, J. L. Volakis and H. T. Anastassiu, "Hybrid Finite Element-Modal analysis of jet engine inlet scattering" *IEEE Trans. Antennas and Propagation*, vol. 43, no. 3, pp. 277-285, March 1995.

[5] D. C. Ross, J. L. Volakis and H. T. Anastassiu, "Overlapping modal and geometric symmetries for computing jet engine inlet scattering" *IEEE Trans. Antennas and Propagation*, vol. 43, no. 10, pp. 1159-1163, Oct. 1995.

[6] H. T. Anastassiu, J. L. Volakis and D. C. Ross, "The Mode Matching Technique for electromagnetic scattering by cylindrical waveguides with canonical terminations" *Journal of Electromagnetic Waves and Applications*, vol. 9, no. 11/12, pp. 1363-1391, Nov./Dec. 1995.

USER'S MANUAL FOR MOMJET

Hristos T. Anastassiu, Mikhail Smelyanskiy,
and John L. Volakis
Radiation Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
1301 Beal Avenue
Ann Arbor MI 48109-2122
FAX: (313) 647-2106
hristosa@engin.umich.edu
msmelyan@engin.umich.edu
volakis@engin.umich.edu

June 18, 1997

1 Introduction

Momjet is a FORTRAN77 code which uses the Moment Method (MoM) to model electromagnetic scattering from perfectly conducting jet engine models. It is based on a free space Moment Method code written by Pamela Haddad at the Radiation Laboratory in 1991.

Section 2 of this manual gives a brief description of MoM applied to the engine problem. Section 3 describes the code operation, and section 4 presents a few examples of Radar Cross Section calculations.

2 MoM applied to the jet engine problem

A major reason for an emphasis on the EM characterization of jet engines is their critical importance in determining the airplanes's reflectivity and for target identification. Even though the engine is a small component of the overall structure, its complexity and visibility in the front sector of the aircraft make its contribution a large component of the overall airframe scattering. Moreover, the rotating engine blades cause unavoidable doppler shifts which are important for target identification purposes. The latter can be an important discriminator for both civilian and military applications and more reliable than other methods.

As applied to electromagnetic scattering problems, the Moment Method (MoM) has proven to be a very accurate integral equation technique. However, its brute force application to the jet engine problem is not feasible due to excessive electrical size of the structure. To reduce the CPU time and storage requirements down to manageable levels, the inherent blade periodicity of the jet engine was exploited to show that the computational domain can be reduced down to a single blade or engine "slice". This discrete body of revolution (DBOR) approach reduces the number of unknowns and storage requirements by a factor equal to the number of blades. In [1] we applied the DBOR concept in the context of integral equation methods for modeling the complex jet engine configurations. By considering mode by mode excitation [1, 2, 3] it is shown that the analysis over

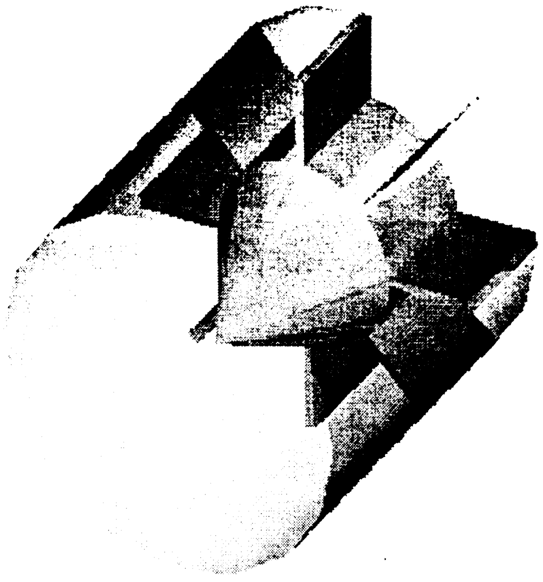

Figure 1: Simplified model of a jet engine.

the entire engine can be reduced down to a surface integral equation over a single blade. It is further demonstrated that the resulting modal scattering matrix is sparse, leading to additional storage and CPU time reductions.

To demonstrate the reduction of the computational domain we consider a cylindrically periodic scatterer (Fig. 1). We define the current column vector $\{\mathbf{J}\}^{(m_s)}$ of the m_s^{th} slice by

$$\{\mathbf{J}\}^{(m_s)} \equiv \left[I_1^{(m_s)}, I_2^{(m_s)}, ..., I_Q^{(m_s)}\right]^T \tag{1}$$

where $I_p^{(m_s)}$ are the elementary currents on the m_s^{th} slice. Brute force MoM yields the linear system

$$
\begin{aligned}
&[\mathbf{A}]^{(00)}\{\mathbf{J}\}^{(0)} + [\mathbf{A}]^{(01)}\{\mathbf{J}\}^{(1)} + ... + [\mathbf{A}]^{(0,N_s-1)}\{\mathbf{J}\}^{(N_s-1)} = \{\mathbf{b}\}^{(0)} \\
&[\mathbf{A}]^{(10)}\{\mathbf{J}\}^{(0)} + [\mathbf{A}]^{(11)}\{\mathbf{J}\}^{(1)} + ... + [\mathbf{A}]^{(1,N_s-1)}\{\mathbf{J}\}^{(N_s-1)} = \{\mathbf{b}\}^{(1)} \\
&\cdots \\
&[\mathbf{A}]^{(N_s-1,0)}\{\mathbf{J}\}^{(0)} + ... + [\mathbf{A}]^{(N_s-1,N_s-1)}\{\mathbf{J}\}^{(N_s-1)} = \{\mathbf{b}\}^{(N_s-1)}
\end{aligned}
\tag{2}
$$

where

$$
A_{pq}^{(n_s m_s)} \equiv jkZ \int_{\Sigma_0}\int_{\Sigma_0} \mathbf{f}_p(\mathbf{r}) \cdot \bar{\bar{\mathbf{R}}}^{-n_s} \cdot \bar{\bar{\mathbf{G}}}(\bar{\bar{\mathbf{R}}}^{n_s} \cdot \mathbf{r}, \bar{\bar{\mathbf{R}}}^{m_s} \cdot \mathbf{r}') \cdot \bar{\bar{\mathbf{R}}}^{m_s} \cdot \mathbf{f}_q(\mathbf{r}') d^2S d^2S' \tag{3}
$$

are the entries of the submatrices $[\mathbf{A}]^{(n_s m_s)}$ representing the interactions between the $\mathbf{f}_p$ and $\mathbf{f}_q$ elementary currents located on the m_s^{th} and the n_s^{th} slices, $\bar{\bar{\mathbf{G}}}$ is the cylindrical waveguide dyadic Green's function and $\bar{\bar{\mathbf{R}}}$ is the rotation dyadic [1]. Also, $\{\mathbf{b}\}^{(n_s)}$ is the excitation column vector of the n_s^{th} slice defined by

$$
\{\mathbf{b}\}^{(n_s)} \equiv \left[V_1^{(n_s)}, V_2^{(n_s)}, ..., V_Q^{(n_s)}\right]^T \tag{4}
$$

where

$$
V_p^{(n_s)} = \int_{\Sigma_0} \mathbf{E}^i \left(\bar{\bar{\mathbf{R}}}^{n_s} \cdot \mathbf{r}\right) \cdot \bar{\bar{\mathbf{R}}}^{n_s} \cdot \mathbf{f}_p(\mathbf{r})\, d^2S \tag{5}
$$

In the latter equation, $\mathbf{E}^i$ is the incident field, Σ_0 is the *outer* surface of the slice within the volume V_0. Clearly, the index m_s indicates the slice where the source point is located, whereas n_s is the index representing the slice containing the observation point. Both indices run from 0 to $N_s - 1$.

Following the analysis in [1] it can be shown that it is sufficient to determine only the currents on the basic slice by solving

$$
\left[[\mathbf{A}]^{(00)} + [\mathbf{A}]^{(01)} e^{jn_i\phi_s} + ... + [\mathbf{A}]^{0,(N_s-1)} e^{j(N_s-1)n_i\phi_s}\right] \{\mathbf{J}\}^{(0)} = \{\mathbf{b}\}^{(0)} \tag{6}
$$

If the cylindrical waveguide dyadic Green's function is to be used, (6) can be further simplified, but that procedure is mainly of theoretical importance; the cylindrical waveguide dyadic Green's function is very difficult to handle computationally, and therefore the free space Green's function is used throughout the code. Additional elementary currents are placed on the waveguide walls to account for the appropriate boundary conditions. An additional result of [1] is that for a given order of the incident mode, only a limited set of scattered modes is excited. Namely, only the scattered modes with orders n that satisfy $n = n_i + \nu N_s \quad , \quad \nu \in \mathbf{Z}$ are reflected back, and this is in agreement with the result given by the FEM analysis of a similar problem [3].

The importance of the periodicity analysis cannot be exaggerated. Since the problem essentially reduces to modeling only one slice of the scatterer, the number of equations or unknowns is reduced by a factor of N_s. For a typical $N_s = 40$, this implies a CPU time and memory reduction each by a factor of 1600. For large scatterers with large periodicity numbers N_s, such as jet engines, the problem can thus be scaled to a tractable size. Moreover, the limited coupling among the scattered modes results in sparse modal scattering matrices which are much easier to store and handle.

3 Code Operation

3.1 Code structure

Momjet models the scattering from metallic cylindrical jet engine inlets terminated by an arbitrary, PEC, cylindrically periodic structure. A simplified geometry is illustrated in Fig. 2, where the termination is a cylindrical hub. A plane wave is assumed to illuminate the open end and couple to waveguide modes in Region 1, propagating towards the termination (Region 2). Each of these modes is used as an excitation for the MoM applied to one periodic sector of the termination (see Figs. 3, 4). Solving the MoM linear system yields the equivalent currents on the termination, and in turn the scattered field within the inlet. To extract the coefficients of the scattered modes propagating towards the open end an integration [1, 2] is performed at a hand-off plane, which is perpendicular to the inlet axis and lies at a fraction of a wavelength in front of the termination (see Fig. 5). Each of the scattered modes propagates to the open end and radiates in free space. The overall modal contribution yields the Radar Cross Section (RCS) of the engine.

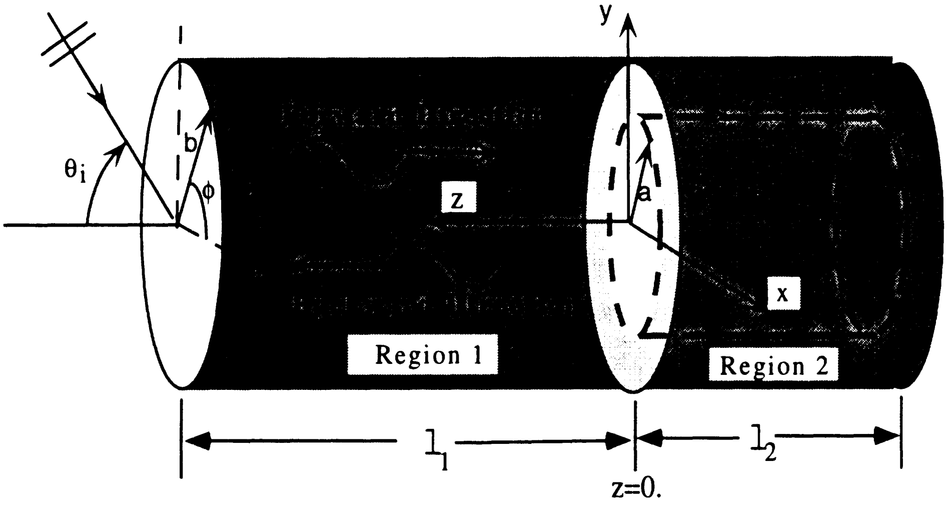

Figure 2: Cylindrical inlet terminated by a cylindrical hub

To exploit the inherent periodicity of the engine, the waveguide modes must have exponential, and not trigonometric dependence on ϕ. Immediate consequence of this is that the modal orders may be negative, positive or zero, and there is no distinction between "odd" and "even" behavior, as opposed to the Mode Matching code that has been developed in the past [4].

Additionally, this representation enables calculation of RCS modulation (caused by blades rotation) in a simple manner as a postprocessing job [5].

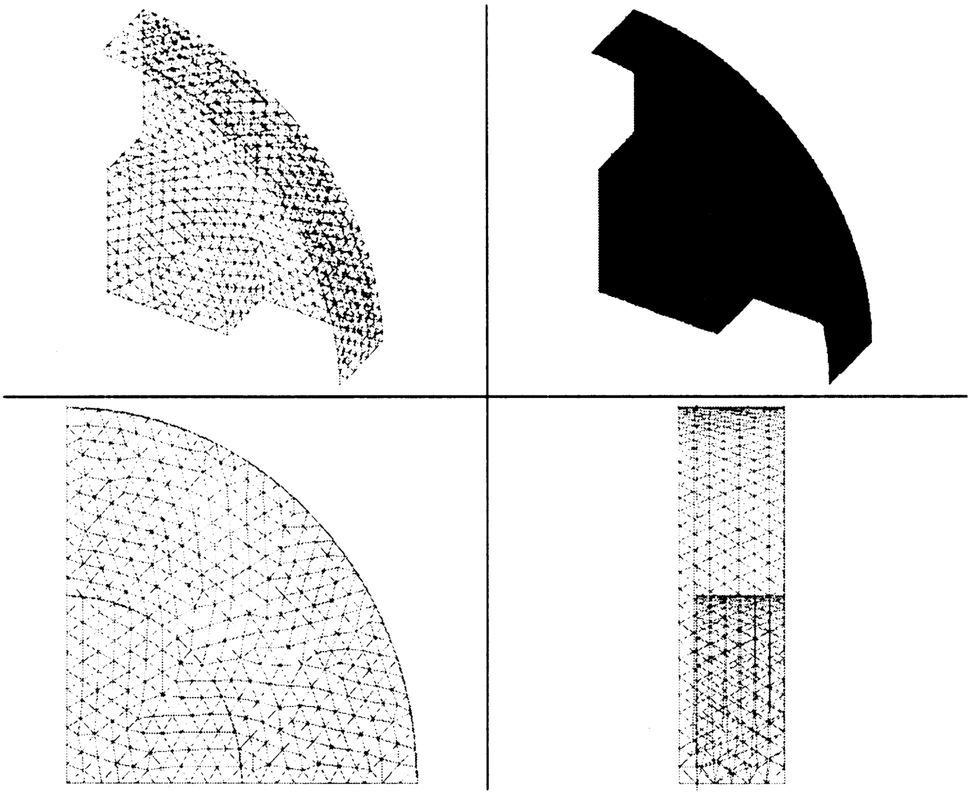

Figure 3: A quarter of a hub geometry.

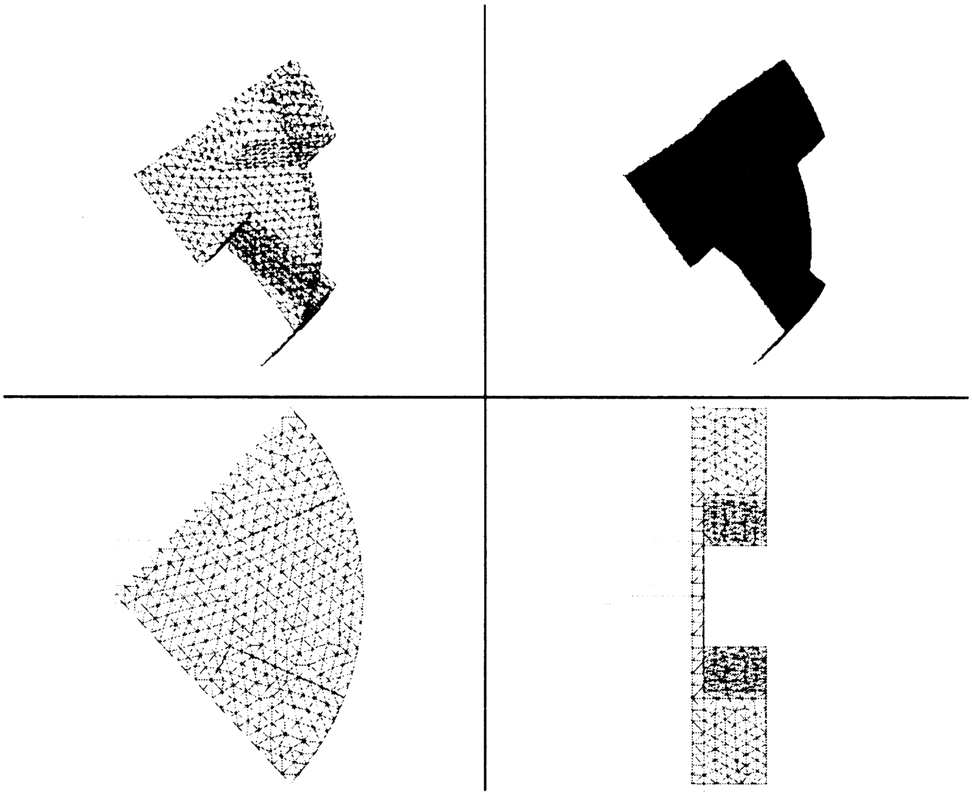

Figure 4: A quarter of a blade geometry.

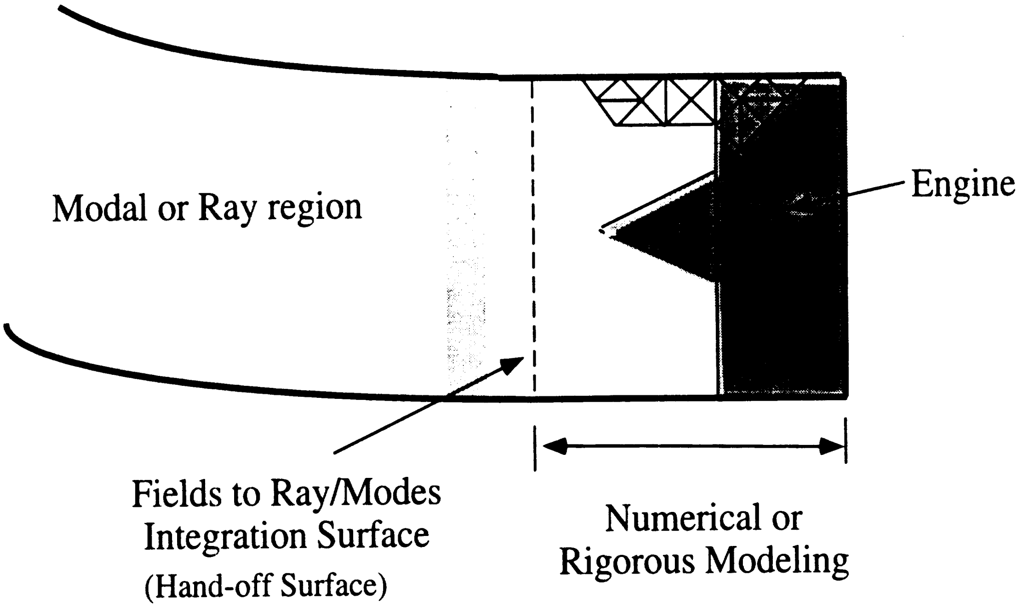

Figure 5: Decomposition of the computational domain.

3.2 Input file

The input file comprises of a discretized sector of the termination, including the waveguide wall up to the hand-off plane. Typical input geometries are illustrated in Figs. 3, 4. The current version of Momjet requires an edge-based format of the geometry file (interior nodes and corresponding edges are listed first). To illustrate the file format we consider a $0.2\lambda \times 0.2\lambda$ flat plate (see Fig. 6), lying on the xy plane and centered at the origin. This geometry is not typical for the code, but its simplicity is exploited to clarify the meshing format of the input file. The geometry file has the form

9 8	($N_{nodes} = 9$, $N_{triangles} = 8$)
.1 .1 .0	(the next N_{nodes} entries are
.1 -.1 .0	the cartesian coordinates of the nodes-
-.1 .1 .0	inner nodes first followed by outer
-.1 -.1 .0	
.1 .0 .0	
.0 .1 .0	
-.1 .0 .0	
.0 -.1 .0	
.0 .0 .0	
16	(total number of edges N_{edges})
8	(number of exterior edges N_{ext})
7 8	(next $N_{edges} - N_{ext}$ entries
5 6	give the pair of nodes defining the edges)
5 9	
6 9	
3 9	
7 9	
8 9	
2 9	
4 8	
4 7	
1 6	
1 5	
3 7	
3 6	
2 8	
2 5	
1 9 10	(the remaining entries give the
2 11 12	triangle connectivity. Each line
2 3 4	entry contains 3 numbers corresponding
5 6 13	to the edge numbers forming the triangle
4 5 14	in a clockwise order)
1 6 7	
8 15 7	
16 8 3	

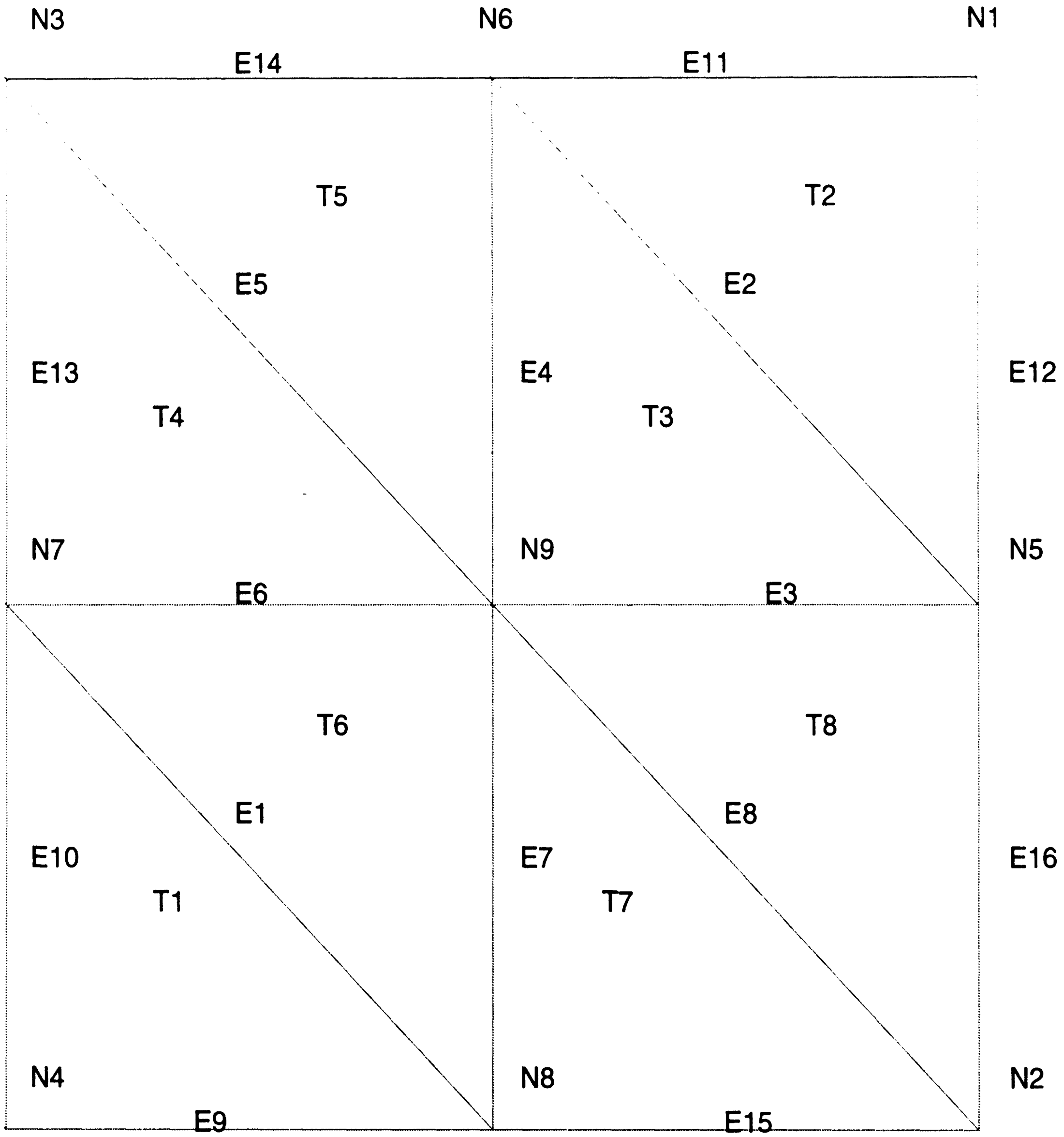

Figure 6: Mesh with node, edge and triangle numbering.

To obtain this format:

• Create the mesh geometry using a meshing package such as IDEAS and export it to a universal file called, for instance, geo-file.unv.

• Convert geo-file.unv to a converter file, called cnv, by running program u2c-new2.f.

• Convert cnv to the desired format geo-file by running program c2p-fast.f. Both u2c-new2.f and c2p-fast.f are available in the same directory with the main code.

Momjet also prompts user for outer radius (in terms of λ), inlet length up to the hand-off plane (in terms of λ), measurement frequency (choose $0.3d0$ if want RCS in $dB/(\lambda^2)$). Inclusion of Ufimtsef currents and evanescent modes is optional. All input parameters may be specified in the input file.

3.3 Preprocessing

Prior to running the code, several parameters must be set appropriately:

• In module dim.inc set maxnod greater or equal to the number of nodes in geo-file.

• In module param.inc set

* in=number of *nonnegative* modal orders in hollow section (Region 1).

* ii=number of modes per order. Chosen to include at least all propagating modes.

* ngas=number of points per subinterval for hand-off plane integration.

* nsym=number of periodic sectors (slices) of engine termination.

* nangle=number of different incidence angles (θ_{inc}).

* bgas=Gaussian integration tolerance (at the hand-off plane).

* exrho=2^{intrho}=number of integration subintervals along the ρ direction.

* exphi=2^{intphi}=number of integration subintervals along the ϕ direction.

* zhandoff=z location of the hand-off plane (the tip of the termination is assumed at $z = 0$.

*irotmax=max rotational angle ϕ (sugested integer value).

*irotst=rotation step (sugested integer value).

*irott=number of rotation steps.

*kstepp=θ step for RCS calculation.

*kfica=number of sempling points on the hand-off plane for unique engine ID representation.

In order to define accurate number of propagating modes (in and ii), two runs of the code have to be done, according to following steps:

1) inicialize in and ii sufficiently high.

2) uncomment line in the Makefile for compilation without optimisation, and comment the line with optimization includded.

3) compile and run the code.

4) check max in and ii when modes are propagating, stop execution, and change parameters in the param.inc file.

5) uncomment line with included optimization and comment line without optimization.

6) compile and run the code.

3.4 Compilation and running

- Compile the code by typing **make** (there is only one Makefile in the directory).
- Run the code by typing **momj**. The interaction is self-explanatory. See section 4 for further details. If you want to use input parameter redirection, then type **momj < input.dat**.

3.5 List of subroutines

1. Initialization codes:

u2c-new2.f: converts geo-file.unv to a converter file, called cnv.

c2p-fast: converts cnv to the desired format geo-file.

2. Subroutines in Makefile:

Momjet.f: main code.

ac-intcyl2.f: performs a double surface integration over two triangles of a function in local area coordinates.

an-int1.f: performs an analytical double surface integration over the singular part of the integrand.

back-subst.f: solves the MoM linear system using LU decomposition.

edge-len.f: updates the edge length table given the edge numbers of the source and observation triangles.

excitcyl.f: computes elements of excitation vector (right hand side of linear system).

fillzcyl2.f: calculates elements of impedance matrix Z corresponding to incident modal order ninc .

get-R.f: reads in triangle resistivity values from file specified by user. According to Pamela Haddad, author of the basic MoM code, the algorithm has not been validated for non-metallic scatterers.

get-mesh1.f: reads in mesh data from file specified by user.

get-verts.f: returns the vertices of a triangle.

normal.f: constructs normal vector to a triangle.

ops.f: various small functions and subroutines performing auxiliary mathematical calculations.

pot-ints-ss.f: computes the potential integrals for uniform and linearly varying surface sources distributed on a planar polygon S. Dependent subroutine in the same module: i-params.

pwcyl5.f: returns the value of $(\mathbf{r}-\mathbf{r}_m)\cdot\mathbf{E}_i$, where $\mathbf{E}_i$ is the incident cylindrical mode and

$\mathbf{r}_m$ is the position vector to the m^{th} vertex of the observation triangle. Dependent subroutines in the same module: integration, discreteap, fieldap, vectintcyl, ffargcyl, dyadfree, cGsldx, GAULEGX, cGsldy, GAULEGY. See code listing for more information on the function of each subroutine.

resfnc.f: returns the value of the integrand of the resistive term $(\mathbf{r}-\mathbf{r}_m)\cdot(\mathbf{r}-\mathbf{r}_n)$, where $\mathbf{r}_m, \mathbf{r}_n$ are the position vectors to the m^{th}/n^{th} vertices of the given observation triangle.

xsintcyl.f: performs a single surface integration of a function over a triangle in local area coordinates.

solve2.f: linear system solvers.

update-signs.f: updates the triangle sign and edge use tables for an input observation interior edge and an input source interior edge.

usr-datacyl.f: prompts user for, and reads in, data pertaining to file names, incident field, rcs observation cut via standard I/O.

vectint.f: performs a single surface integration of vector function over a triangle.

cylinsubr2.f: includes several subroutines and functions related to the modal analysis of the hollow section (duct). Subroutines included: excitvec, radcrosec, vemult, modesl. See code listing for more information on the function of each subroutine.

blockdata.f: initializes common block which contains the local area coordinate integration points and weights used in all integration routines.

4. Auxiliary files:

const.inc: includes various constants.

dim.inc: includes uniform array dimensioning.

param.inc: includes modal and integration parameters.

5. Output files:

rcsphi.txt: $RCS = RCS(\theta_{inc})$ for ϕ polarization.

rcsth.txt: $RCS = RCS(\theta_{inc})$ for θ polarization.

modul.txt: $RCS = RCS(\varphi_{rot})|_{\theta_{inc}=const}$ modulation file.

etotal.txt: total unweighted scattered field on the hand-off plane (ID for every engine termination). Need to be weighted in the future. Number and position of sempling points is determined by the parameters ngas, exrho and exphi (see param.inc file and figure 7).

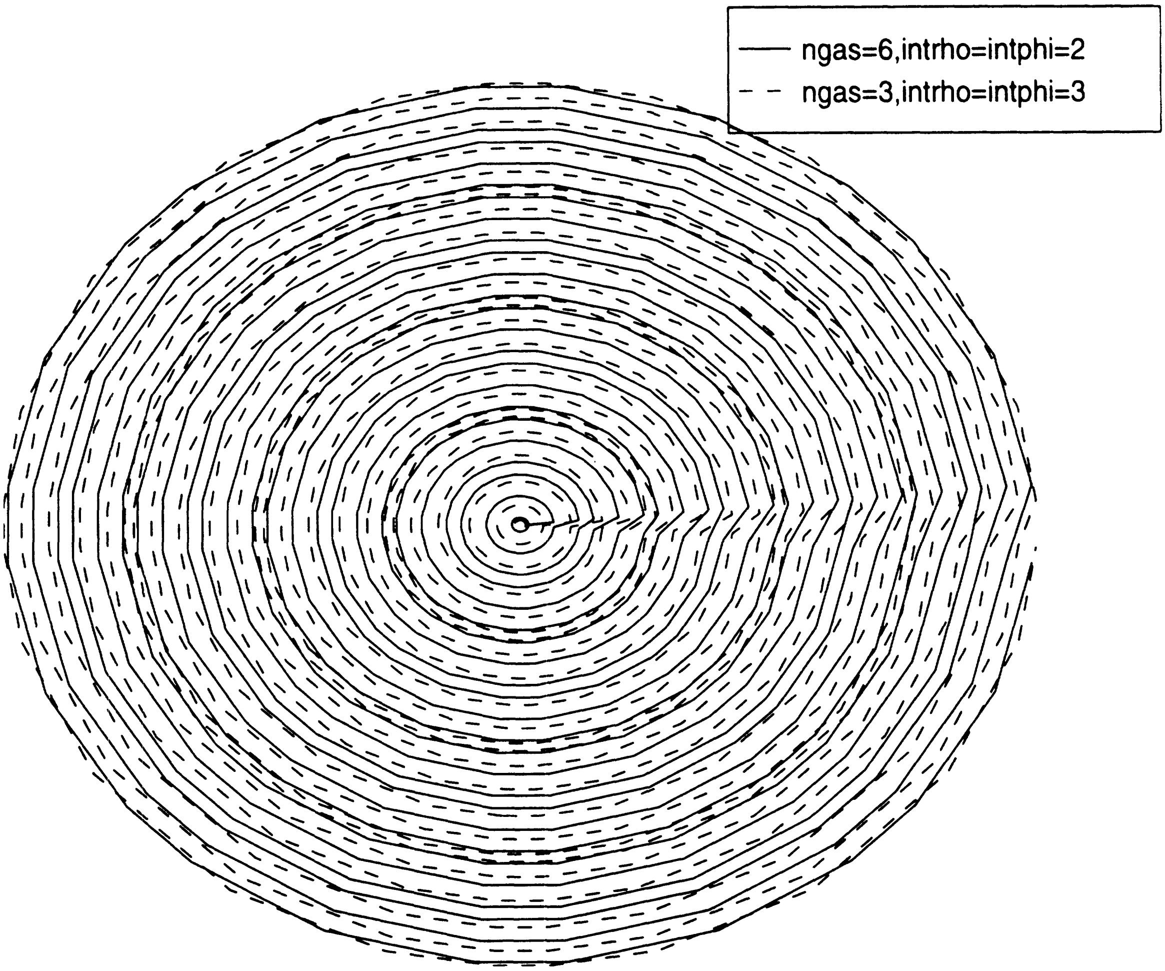

Figure 7: Sampling points on the hand-off plane.

4 Examples

This example is of the 4 blade termination described in Fig. 4. The outer radius is 2λ, the hub radius is λ, the termination is 0.5λ deep, the length of the hollow region is 3λ and the hand-off plane is located 0.1λ in front of the termination. Since the number of nodes for this geometry is 656, the dim.inc file should read

```
Integer nmax,maxnod,maxedg,maxtri,maxz,maxrcs
Parameter (nmax=4)
Parameter (maxnod=660)
Parameter (maxtri=2*maxnod)
Parameter (maxedg=3*maxnod)
Parameter (maxz=3*maxnod)
Parameter (maxrcs=200)
```

and the param.inc file should read

```
integer ii,in,iis,ii2,ii4
integer ngas,nsym,nangle,kfica,irott,irotmax,itotst
integer intrho,intphi,exrho,exphi,kstepp
real*8 bgas,zhandoff
parameter(in=11,ii=4)
parameter(iis=in*ii)
parameter(ii2=2*iis)
parameter(ii4=2*ii2)
parameter(ngas=3)
```

```
parameter (nsym=4)
parameter (nangle=121)
parameter(bgas=1.0d-2)
parameter(intrho=2)
parameter(intphi=2)
parameter(exrho=2**intrho)
parameter(exphi=2**intphi)
parameter(zhandoff=0.1d0)
parameter(irotmax=91)
parameter(irotst=1)
parameter(irott=irotmax/irotst + 1)
parameter(kstepp=1)
parameter(kfica=exrho*exphi*ngas*ngas)
```

We emphasize that in and ii are chosen so that all propagating modes are included. The code output yields the propagation characteristics of each mode, hence running the code itself for a few test cases eventually provides the correct values of in and ii.

For this geometry the input reads:

```
Enter inlet radius in wavelengths
2.0d0
Enter mesh file name:
blade_2w_4
Enter length of hollow section in wavelengths
3.0d0
```

Enter 1 for inclusion of rim

contribution and 2 for non-inclusion

2 !(not including Ufimtsef currents on the rim)

Enter 1 for inclusion of evanescent modes

and 2 for exclusion

2

Enter measurement frequency in GHz

0.3d0 !(to obtain RCS in dB/λ^2)

For redirection running, the input file is:

```
2.0d0
blade_2w_4
3.0d0
2
2
0.3d0
```

The output reads:

```
____________________________
TM modes (n,m,krho,prop./evan.)
____________________________
0 1 1.20241288350811 1
0 2 2.760039056538 1
```

0 3 4.32686395109136 1

0 4 5.89576721702858 1

1 1 1.91585298512173 1

1 2 3.50779348764358 1

1 3 5.08673402355608 1

1 4 6.66184574789692 -1 mode indices, eigenvalues in λ^{-1},

2 1 2.56781132204712 1 1 for propagation and -1 for evanescence

...

TE modes (n,m,krho,prop./evan.)

---

0 1 1.91585298512173 1

0 2 3.50779348764358 1

...

Incident mode order= -10 !(working with mode order=-10)

Order of system = 1777

No. of triangular elements = 1216

Slice 0 !(working with slice 0)

...

Output file rcsphi.txt is:

Phi-phi Polarization

1.000000000000000E-02 29.63130309987938 RCS vs. θ_{inc} in degrees

.51 29.61263172062007

1.01 29.55830862102566

1.51 29.4690832676185

```
2.01 29.34619755314773
2.51 29.19138128112484
3.01 29.00683892922878
3.51 28.7952212610019
4.01 28.55957452047508
4.51 28.30326037655948
5.01 28.02984213788905
5.51 27.74293755636333
...
```

Output file rcsth.txt is:

```
Theta-theta Polarization
1.000000000000000E-02 29.63128937395119
.51 29.57655924653133
1.01 29.41605365137127
1.51 29.14821273587283
2.01 28.77040147133607
2.51 28.27883999812705
3.01 27.66851579327062
3.51 26.93309717096325
4.01 26.06489475433391
4.51 25.0549787304352
5.01 23.89369678494705
5.51 22.57214244083259
...
```

Output file modul.txt is:

```
Phi-phi Polarization
```

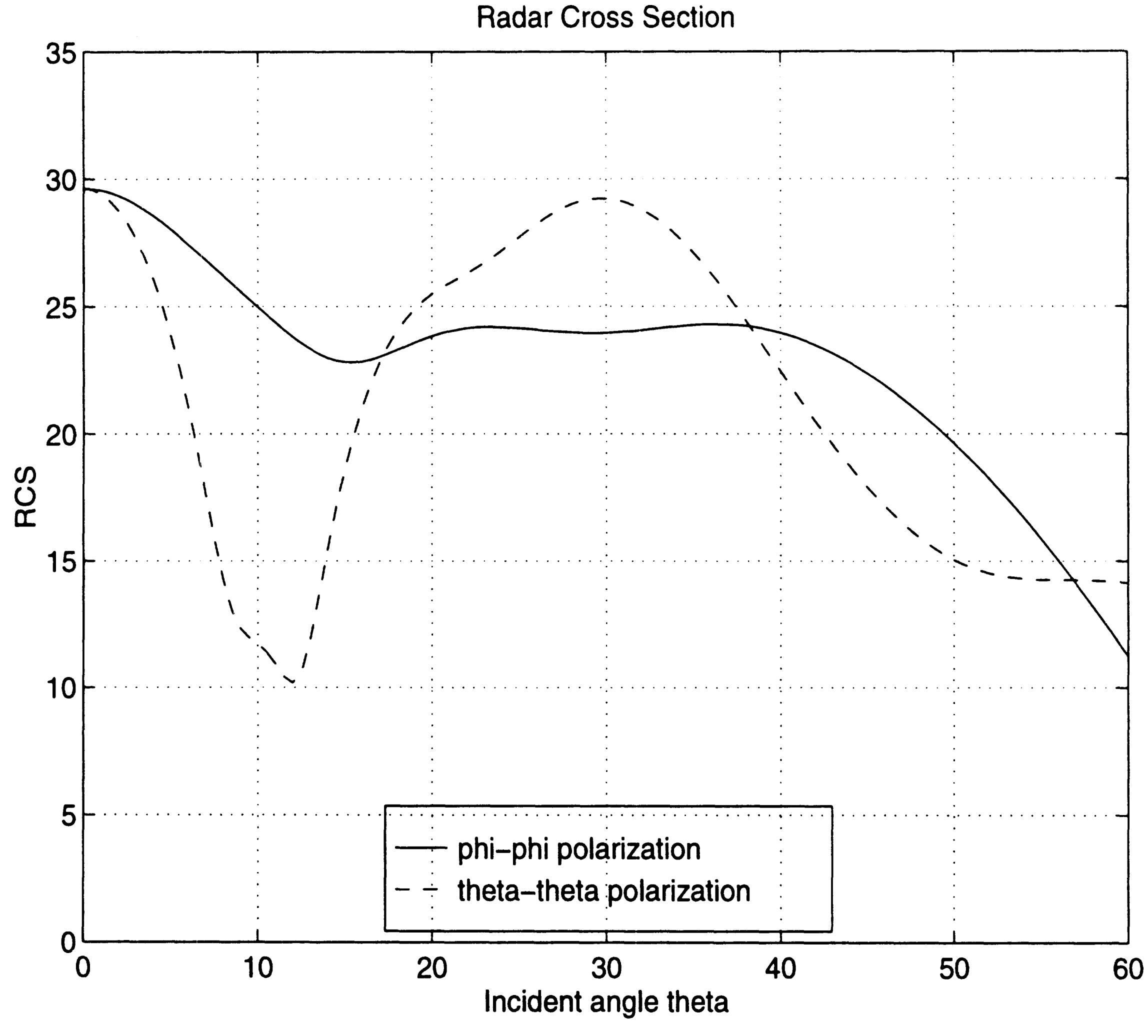

Figure 8: RCS for both polarizations.

```
theta= 1.000000000000000E-02
1 29.63130309269711
2 29.67204477665604
3 29.67223584446394
4 29.67223669880159
5 29.67223664691986
...
```

Theta-theta Polarization

```
theta= 1.000000000000000E-02
1 29.6312893811335
2 29.67203110417029
3 29.67222223582765
4 29.67222317841145
5 29.67222323874154
...
```

Output file etotal.txt is:

```
X,Y,Z,Unwaited E-Field at the Handoff Plane
5.641662856629827E-02 1.015140670462751E-02 .1 81.08614154147517
4.049268616987463E-02 4.057375264402420E-02 .1 80.9713012643719
1.003855320833504E-02 5.643681853501174E-02 .1 80.26172131340697
-1.015140671619875E-02 5.641662856421617E-02 .1 79.68034874500992
-4.057375265232938E-02 4.049268616155283E-02 .1 78.92981225914005
...
```

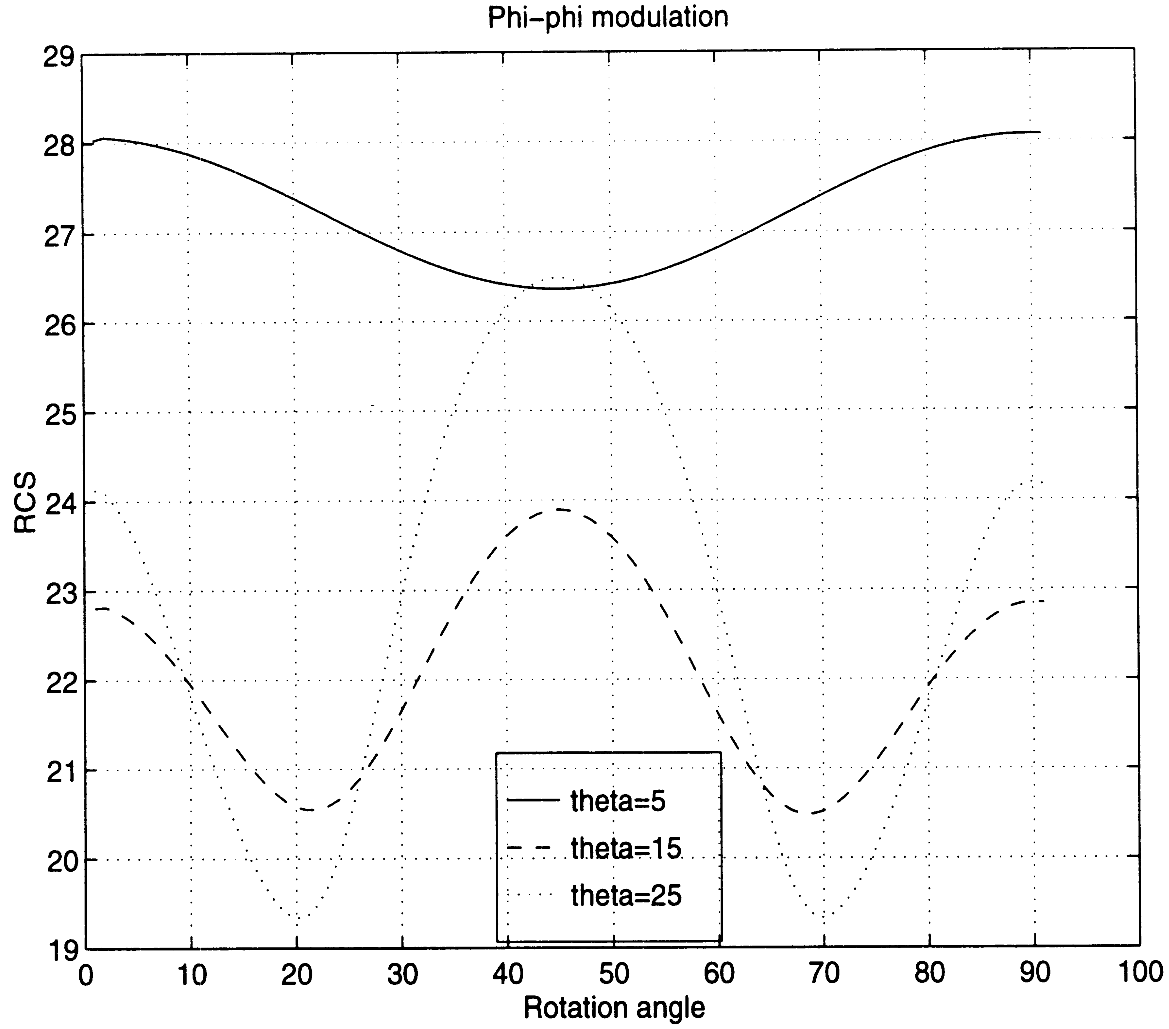

Figure 9: Modulation pattern for rotating blades-phi polarization.

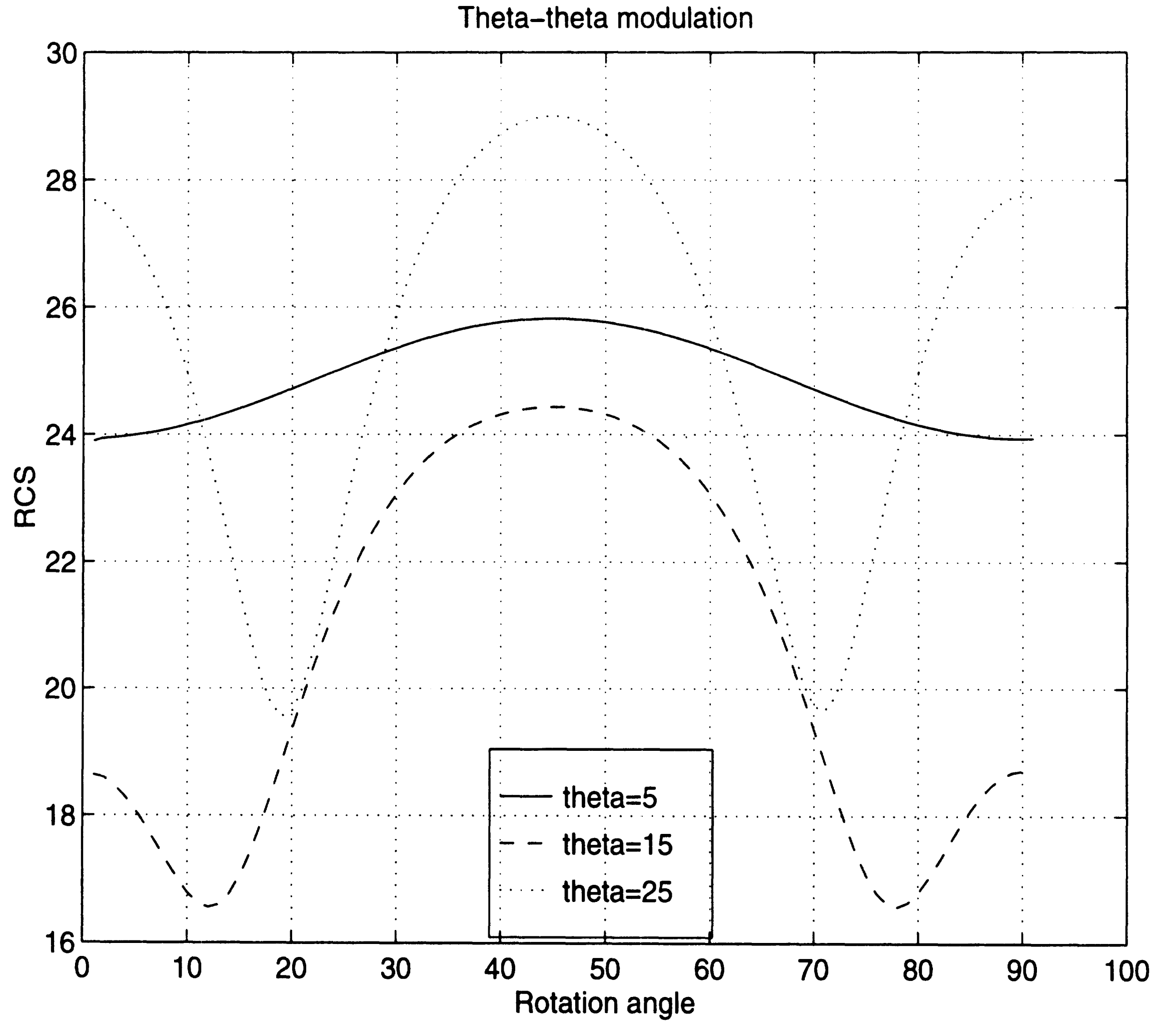

Figure 10: Modulation pattern for rotating blades-theta polarization.

References

[1] H.T. Anastassiu, *Electromagnetic Scattering from Jet Engine Inlets Using Analytical and Fast Integral Equation Methods*, Ph. D. Thesis, University of Michigan, 1997.

[2] D. C. Ross, J. L. Volakis and H. T. Anastassiu, "Hybrid Finite Element-Modal analysis of jet engine inlet scattering" *IEEE Trans. Antennas and Propagation*, vol. 43, no. 3, pp. 277-285, March 1995.

[3] D. C. Ross, J. L. Volakis and H. T. Anastassiu, "Overlapping modal and geometric symmetries for computing jet engine inlet scattering" *IEEE Trans. Antennas and Propagation*, vol. 43, no. 10, pp. 1159-1163, Oct. 1995.

[4] H. T. Anastassiu, J. L. Volakis and D. C. Ross, "The Mode Matching Technique for electromagnetic scattering by cylindrical waveguides with canonical terminations" *Journal of Electromagnetic Waves and Applications*, vol. 9, no. 11/12, pp. 1363-1391, Nov./Dec. 1995.

[5] C. Ross, J. L. Volakis and H. T. Anastassiu, "Efficient Computation of Radar Scattering Modulation from Jet Engines"*Radio Science*, vol. 31, no. 4, pp. 991-997, July-Aug., 1996.