

J. L. Volakis
K. Sertel
M. A. Carr

Users Manual,
Test Case Manual,
and
Mathematical Background for
Single Level SWITCH-FMM

Version 1

April 2000

Users Manual for SWITCH-FMM Version 1.0

John L. Volakis, Kubilay Sertel, and Michael A. Carr
Radiation Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2212, USA

April 5, 2000

Contents

1	Introduction	3
2	Input and Output Files	4
2.1	Dimension File	4
2.2	Geometry Files	5
2.2.1	PATRAN Geometry Output	5
2.2.2	SWITCH Geometry Output	7
2.3	Input File	8
2.4	Output Files	10
2.4.1	Farfield Radar-Cross-Section (RCS) Data	10
2.4.2	Induced Surface Currents	11
2.4.3	Solution Coefficients	12
2.4.4	Clustering Information	12
3	Executing the Solver	13
4	Data Visualization	14
4.0.5	RCS Data Visualization	14
4.0.6	Surface Currents Visualization	14
4.0.7	Cluster Visualization	15
5	Quick Reference	16
5.1	Compilation of the Solver	16
5.2	Executing the Solver	16
5.3	Checking the Output	17
5.4	Visualizing the Outputs	17

1 Introduction

This document is intended to describe the compilation and execution of the code SWITCH-FMM developed at the Radiation Laboratory of the University of Michigan Electrical Engineering and Computer Science Department.

SWITCH-FMM is a Method of Moments (MoM) based Computational Electromagnetics (CEM) tool to simulate electromagnetic (EM) scattering scenarios involving arbitrarily curved perfect electric conductor (PEC) targets. It is implemented in Fortran 77. The MoM formulation used in the implementation of the code is based on the Electric-Field-Integral-Equation (EFIE), the Magnetic-Field-Integral-Equation (MFIE), and the Combined-Field-Integral-Equation (CFIE) formulations of EM scattering from PEC targets. To overcome the computation time length and memory requirements of the conventional MoM implementations ($O(N^3)$ and $O(N^2)$ respectively), the Fast Multipole Method (FMM) is implemented. The FMM reduces the computation complexity to $O(N^{1.5})$ per iteration (in iterative solution of the resulting linear system) and the memory to $O(N^{1.5})$. The reader is referred to the references [1, 2, 3] for further information on the formulations used in the implementation of the code.

The target geometry is represented in the code using curved quadrilateral surface patches which are defined by 9 points in space located on a topologically rectangular 3×3 grid [1]. The generation of target geometry is discussed in Section 2.2.

The input parameters related to the simulation should be given in the input file as outlined in Section 2.3.

Section 2.4 describes the output files generated by the code. RCS and induced surface current density data produced at the output can be visualized using the MATLAB and PATRAN visualization files as described in Section 4.

There are two versions of the code. The “non-symmetric” version located in **SWITCH/Non_Symmetric/** uses the whole target geometry, whereas the “symmetric” version in **SWITCH/Symmetric/** makes use of symmetry to model half of the target. This manual is based on the “non-symmetric” version. The difference between the two versions are mentioned whenever necessary.

2 Input and Output Files

The following subsections outline the I/O and visualization files used by the solver SWITCH-FMM.

2.1 Dimension File

Before compiling the solver, the dimensions of the data arrays used in the code must be approximately specified in the dimension file **main.dim** located in **SWITCH/Non_Symmetric/src** directory. The structure of the file is

```
parameter (  
  
& NNodes = 5600,  
& Nntri = 3600,  
& Nunknowns = 4500,  
& Ncluster = 60,  
& NL = 15,  
& Nnzbimat = 1500000  
&  
& )  
  
C NNodes : Number of NODES in target geometry  
C Nntri : Number of ELEMENTS in target geometry  
C Nunknowns : Number of UNKNOWNS in target geometry  
C Ncluster : Number of CLUSTERS in target geometry  
C (Ncluster=sqrt(Nunknowns))  
C NL : Number of MULTIPOLES in target geometry  
C Nnzbimat : Number of NONZEROES+1 in the near-field matrix
```

The number of NODES and ELEMENTS can be found in the first few lines of the geometry files. Number of UNKNOWNS is equal to the number of edges shared by two quad elements. It is equal to twice the number of ELEMENTS for a closed geometry. Number of CLUSTERS should be set to be $Ncluster = \sqrt{Nunknowns}$. Number of MULTIPOLES is computed by the code and it varies between 6 and 20. Number of NONZEROES in the

near-field matrix is also problem dependent and an estimate to this number can be found as $Nnzbimat=(10 \text{ to } 20)*Ncluster*Nunknowns$.

The whole code needs to be recompiled whenever this file is modified. The compilation is machine dependent and must be done using proper “make-files”.

The dimension file for the “symmetric” version is located in **SWITCH/Non-Symmetric/src** and has the same structure as that of the “non-symmetric” version as explained above.

2.2 Geometry Files

The geometry files used in the code are the neutral PATRAN output files and the original SWITCH geometry files. Two separate geometry files are required. One file contains the coordinates of the nodes of the quad elements forming the mesh, and the other contains the nodes of the elements. The geometry file containing the quad element nodes must be placed in the directory **SWITCH/Non-Symmetric /geom/elements** and the other geometry file must be placed in the directory **SWITCH/Non-Symmetric /geom/nodes**. These two geometry files must have the same name. It is crucial that the element normals be pointing outward (necessary when the MFIE or CFIE option is used). This can easily be checked in commercial mesh generation packages such as PATRAN. Example element and node files for PATRAN and IDEAS outputs are given below for the “non-symmetric” version.

For the “symmetric” version, the same structure is used. However, the geometry files contain only half of the symmetric geometry. The symmetry plane is specified in the input file. The assumption used is that the symmetric geometry lies on one side of the symmetry plane. The latter is assumed to be on one of the principle planes in cartesian coordinate system.

2.2.1 PATRAN Geometry Output

After the target mesh is generated using PATRAN, it is then exported in Neutral File format to two separate files (one for nodes and the other for node connectivity as noted above). The format of the output node file from PATRAN is like the one below.

```

25 0 0 1 0 0 0 0 0
P3/PATRAN Neutral File from: /tmp/PATRAN/dart2.db
26 0 0 1 8906 0 0 0 -1
05-Mar-99 16:01:00 3.0
1 1 0 2 0 0 0 0 0
-1.000000000E+0 -7.814555225E-9 3.184821606E-1
1G 6 0 0 000000
1 2 0 2 0 0 0 0 0
-1.000000000E+0 5.538074672E-2 3.129804432E-1
1G 6 0 0 000000
1 3 0 2 0 0 0 0 0
-1.000000477E+0 1.105192080E-1 3.056425154E-1
1G 6 0 0 000000
:
1 8906 0 2 0 0 0 0 0
-4.162685776E+1 -5.847183704E+0 1.057478547E+0
1G 6 0 0 000000
99 0 0 1 0 0 0 0 0

```

Not all of the lines above are needed by the SWITCH-FMM code. Clearly the line including the 3 decimal numbers refer to the node coordinates. The actual node number is given as the second number in the line above the coordinates. The format of the element connectivity file exported from PATRAN is

```

25 0 0 1 0 0 0 0 0
P3/PATRAN Neutral File from: /tmp/PATRAN/dart2.db
26 0 0 1 8906 2226 0 0 -1
05-Mar-99 16:01:12 3.0
2 1 4 2 0 0 0 0 0
9 0 0 0 .000000000E+00 .000000000E+00 .000000000E+00
219 216 185 189 217 218 188 221 220
2 2 4 2 0 0 0 0 0
9 0 0 0 .000000000E+00 .000000000E+00 .000000000E+00
178 184 189 185 182 187 188 183 186
:

```

```

2 2226 4 2 0 0 0 0 0
9 0 0 0 .000000000E+00 .000000000E+00 .000000000E+00
1710 6229 6231 1712 6221 6230 6223 1711 6222
99 0 0 1 0 0 0 0 0

```

Again, not all the lines are used by FMM-SWITCH. Here, the lines containing the 9 integers gives the node numbers of the element. As noted above, the node file should be placed in the directory **SWITCH/Non_Symmetric/geom/nodes** and the element file should be placed in the directory **SWITCH/Non_Symmetric/geom/elements** with the same name for proper operation of the code. Also the geometry type in the main input file should be specified as “patran”. This file should be placed in the directory **SWITCH/Non_Symmetric/input**. For the “symmetric” version, the node file should be placed in the directory **SWITCH/Symmetric/geom/nodes** and the element file should be placed in the directory **SWITCH/Symmetric/geom/elements** with the same name. Also the geometry type in the input file in **SWITCH/Symmetric/input** directory should be specified as “PATRAN”.

2.2.2 SWITCH Geometry Output

The geometry files that are in the format of the original Northrop SWITCH code can also be read into the SWITCH-FMM solver. In this case, the node file should have the format

```

962
1 0.4999994E+01 0.2838807E-13 -0.6494433E-06
2 0.1192099E-05 0.9999973E+00 0.0000000E+00
3 0.4566187E+01 0.1716829E+00 0.0000000E+00
:
962 -0.3427966E+01 -0.2119954E+00 -0.4965332E+00

```

Clearly, the first column in the above listing is the node number followed by its coordinates. The first line gives the total number of nodes (962 here). The element file should have the format

```
P3/PATRAN Neutral File /ogive480.db small 240 quad EMCC test
```



```

case, 11/24/97
240 0 240 240 0 0
0 480 480
0 0
0 0
1 9 7 0
3 72 27 45 118 71 1 70 26
0 0 1 2 0 3 4 0 0 0 0 0
2 9 7 0
4 74 28 46 119 73 3 72 27
0 0 2 5 0 6 7 0 0 0 0 0
:
240 9 7 0
754 783 755 932 962 931 870 892 871
0 0 -440 -386 0 480 479 0 0 0 0 0

```

The node file should be placed in the directory **SWITCH/Non_Symmetric/geom/nodes** and the element file should be placed in the directory **SWITCH/Non_Symmetric/geom/elements** with the same name for proper operation of the code. Also the geometry type in the input file in **SWITCH/Non_Symmetric/input** directory should be specified as “switch”.

For the “symmetric” version, the node file should be placed in the directory **SWITCH/Symmetric/geom/nodes** and the element file should be placed in the directory **SWITCH/Symmetric/geom/elements** with the same name. Also the geometry type in the input file to be placed in the directory **SWITCH/Symmetric/input** should be specified as “switch”.

2.3 Input File

The parameters (frequency, pattern cuts, etc) related to the specific simulation runs must be specified in **filename.input**. As noted earlier, this should be placed in **SWITCH/Non_Symmetric/input**. The file structure is

```

"switch"
"inches"
5.91
1.0

```

```
"bistatic"  
2  
90.0 90.0 0.0  
0.0 180.0 0.5  
0.001  
500
```

The first line in this file specifies the format of the geometry file. Valid options are “switch” and “PATRAN”. Others can be added in the future.

The second line gives the units of the geometry. Valid options are “inches” and “meters”. When the unit is specified as “inches” the solver automatically scales the frequency so that the read geometry data are treated in units of “meters”. The RCS results are unaffected.

The solution frequency is entered in GHz and is specified in the third line. This version of the solver only works at a single frequency.

The α parameter of the CFIE is specified in the fourth line as a real number. Valid values are $0.0 \leq \alpha \leq 1.0$ for closed targets. For open structures α must be set to 1.0.

The RCS pattern type is specified in line 5. Valid entries are “bistatic” and “monostatic”.

The polarization of the incident plane electromagnetic wave is specified in line 6. Valid entries are 1 and 2 (1 for vertical polarization, 2 for horizontal polarization).

Line 7 specifies the range of observation angle θ measured from the vertical axis. The first value is the beginning value and the second is the ending value; the third value is the angle increment.

Line 8 specifies the range of observation angles ϕ with reference to the x-axis. Again, the first value is the beginning value and the second is the ending value; the third value is the angle increment.

Line 9 specifies the convergence tolerance for the iterative solver. The Conjugate Gradient Squared (CGS) is implemented this time but others can be used.

Line 10 specifies the maximum number of iterations prior to terminating the iterative solver. If the solver does not converge in the specified number of iterations, it will stop. Output data will be generated, but depending on the error in the last step it may or may not reflect a correct solution.

For the “symmetric” version, the input file must contain the symmetry

plane information, so its structure is slightly different from the above. It should be placed in the directory **SWITCH/Symmetric/input** and has the structure

```
"PATRAN"  
"x-z"  
"inches"  
5.91  
1.0  
"bistatic"  
2  
90.0 90.0 0.0  
0.0 180.0 0.5  
0.001  
500
```

The symmetry-plane is specified in the second line, the other information is the same as in the “non-symmetric” version. Valid entries for the symmetry-plane specification are

```
"x-y"  
"y-x"  
"x-z"  
"z-x"  
"y-z"  
"z-y"
```

2.4 Output Files

The output files for the “symmetric” and “non-symmetric” versions are the same. The following output files are generated.

2.4.1 Farfield Radar-Cross-Section (RCS) Data

After solving the problem, the code computes the scattered field by the target and generates the RCS data in the following format

```
90.00000 0.00000 -52.49841 120.04202 -65.76239 108.26443
```

```

90.00000 0.50000 -52.52096 119.97476 -65.92950 108.37727
90.00000 1.00000 -52.56872 119.91295 -66.09913 108.49729
90.00000 1.50000 -52.64211 119.85621 -66.27121 108.62270
:
90.00000 179.50000 -19.82681 -30.93680 -63.67008 84.47012
90.00000 180.00000 -19.82989 -30.90826 -63.61202 83.64088

```

This is the specific output for the **ogive480** geometry for VV pol. The first column gives the θ -angle value (in deg.) and the second column gives the ϕ -angle value (in deg.) corresponding to the computed RCS values given in the remaining columns to the right. The third and fifth columns give the RCS results for vertical and horizontal field components in dBs. The fourth and sixth columns are the corresponding phases of the vertical and horizontal field components in degrees.

2.4.2 Induced Surface Currents

For visualization purposes, the values of the induced surface current are also output. The file structure is

```

bistatic
1 1 1 0 962 240 1 0 0 0
bistatic
1 0.00000 0.00000 0 962 1 1 1 1 0 0 0 0 0 0
bistatic
1 1
6.0863093E-072.7060833E-035.1328743E-016.0863093E-07 ...
2 1
9.7520167E-035.8606099E-101.1953557E-019.7520167E-03 ...
:
962 1
1.4312185E+009.0374618E-023.6556860E-011.4312185E+00 ...

```

This is an auxiliary file to be used as an input to the converter **restxt**. The file contains the magnitudes of the induced surface current on the target at each node of the mesh. It can be converted into binary format using an auxiliary PATRAN file **restxt**. It can then be imported into PATRAN to

visualize the surface currents induced on the target. This will be addressed in Section 4.

2.4.3 Solution Coefficients

This file is generated for completeness. It contains the solution vector (the unknown coefficient in the basis function expansion of the induced surface current) for the problem being analyzed. The structure is

```
1 1.58523E-03 5.14229E-04
2 1.55354E-02 5.81232E-02
3 -0.257247 -0.200015
4 5.44715E-02 0.349153
:
479 -0.523232 -0.763163
480 -0.623706 0.479390
```

The values in the first column are the associated unknown-number of the basis functions used (here they refer to the edges shared by two quad elements in the mesh). Second and the third columns are the real and imaginary parts of the coefficient. It should be noted that the basis functions are not numbered as they appear in the element file.

2.4.4 Clustering Information

To check the quality of the clustering for FMM implementation, the file **ccenters.xyz.filename** is output after clustering is done. The file structure is

```
1.874762 0.318430 0.675829 0.800198
3.048203 -0.124327 0.438653 0.733757
0.400361 0.680715 0.552857 0.829164
:
-0.462368 -0.557853 -0.695336 0.794405
-2.523491 -0.546767 -0.092770 0.851855
```

Each line refers to one of the clusters. The first column is the values of the maximum radii of the cluster; the second, third and fourth columns are

simply the x, y, and z coordinates of the cluster centers.

3 Executing the Solver

The following directories must be available/created prior to compiling the code

SWITCH /Non_Symmetric/geom/nodes

SWITCH /Non_Symmetric/geom/elements

SWITCH /Non_Symmetric/input

SWITCH /Non_Symmetric/src

SWITCH /Non_Symmetric/workspace

SWITCH /Non_Symmetric/bin

SWITCH /Non_Symmetric/datavis

SWITCH /Non_Symmetric/output

The explanation to execute the solver given below is based on the “non-symmetric” version of the code. The same instructions also apply to the “symmetric” version if the steps are carried out in the directory of the “symmetric” version (**SWITCH** /Symmetric/).

Once the geometry and input files are ready for the simulation, they must be put in the proper directories. The source files are located in the directory **SWITCH** /Non_Symmetric/src. After the dimension file **main.dim** is designed for the specific problem, the user can proceed with its compilation. The code is compiled using the **make** utility with a proper **makefile** depending on the machine being used. After compilation the executable is automatically placed in the directory **SWITCH**/Non_Symmetric/bin. This directory must therefore be available prior to compilation.

To run the code, the user must be in the directory **SWITCH**/Non_Symmetric. Executing the available batch file

runsw filename

will execute the solver.

If the dimensions in the file **main.dim** are less than needed, the code will prompt the information and stop. The dimensions should be fixed, the code should be recompiled and rerun. The sparse solver library file need not be recompiled unless the operating platform is changed.

The code uses the directory **SWITCH/Non_Symmetric/workspace** to write auxiliary files while it is running. The output files are written in this directory, and after the run is complete those files are moved to **SWITCH/Non_Symmetric/output**.

4 Data Visualization

The output files can be imported in e.g. Matlab for visualization. The following subsections outline the visualization tools for Matlab and PATRAN.

4.0.5 RCS Data Visualization

The Matlab file **ffphi.m** can be used to generate RCS plots on constant θ cuts using the data in the RCS output file generated and similarly **fftheta.m** can be used to generate RCS plots on constant ϕ cuts. Both files are located in the directory **SWITCH/Non_Symmetric/datavis**. The RCS data file should be copied in **SWITCH/Non_Symmetric/datavis** under the name **farfield.output**.

4.0.6 Surface Currents Visualization

The output file **current.txt.filename** can be converted into PATRAN binary results file using an auxiliary PATRAN file **PATRAN/bin/restxt**. Extensive information about the auxiliary files of PATRAN can be found in PATRAN's Users Manual. The output file can then be imported into PATRAN and the magnitude of the induced surface currents can be plotted in PATRAN. To do so, the output file, e.g. **current.txt.dart2** should first be copied into **current.txt**. When the auxiliary file **restxt** is executed, it prompts for the type of conversion. The "txt-to-res" conversion option should be selected and the name of the file should be input as *current* (i.e. without the .txt extension). The output will create the file **current.res**. In the PATRAN Analysis menu, this file should be read into PATRAN. In the Results menu in PATRAN the surface currents can then be plotted.

4.0.7 Cluster Visualization

The quality of clustering can be checked using the Matlab file **plcntr.m** located in **SWITCH/Non_Symmetric/datavis**. The clustering file (e.g. **ccenters.ogive**) should be copied in **SWITCH/Non_Symmetric/datavis** with the name **ccenters.xyz**. The Matlab script **plcntr.m** can be used to plot the circles representing the clusters. This version only plots 2-D circles representing the clusters.

5 Quick Reference

This section is a step-by-step reference to execute the code SWITCH-FMM. It is assumed that the directory structure of the code is already created on user's account. To use the "symmetric" version of the solver, users should execute the steps below in the directory **SWITCH/Symmetric/** as opposed to executing them in **SWITCH/Non_Symmetric/**. It should be kept in mind that the geometry and input files of the "symmetric" version are different than those of the "non-symmetric" version.

5.1 Compilation of the Solver

- Go to directory **SWITCH/Non_Symmetric/src**
- Design the parameter file **main.dim** subject to the geometry size.
- Type *touch *.f* so that each file includes the updated **main.dim**
- Compile the code by executing *make makefile_XXX* using the proper *makefile_XXX* where "XXX" refers to the machine type such as "sun", "hp", or "sgi".
- Now the executable **sw** is generated and moved in the directory **SWITCH/Non_Symmetric/bin**

5.2 Executing the Solver

- Put the geometry files in **SWITCH/Non_Symmetric/geom**. See the **Users Manual** for more information.
- Select the run parameters and put the input file in **SWITCH/Non_Symmetric/input**
- Go to directory **SWITCH/Non_Symmetric**
- Execute the solver by typing *runsw filename*(type *runsw filename* to run the symmetric version in the directory **SWITCH/Symmetric**). The name of the geometry files and the input file should all be the same.

5.3 Checking the Output

- Go to directory **SWITCH/Non_Symmetric/output**
- Check that all four output files are generated after the simulation is completed. These are
 1. `ccenters.filename`
 2. `coeff.filename`
 3. `current.txt.filename`
 4. `farfield.filename`

5.4 Visualizing the Outputs

- Copy `farfield.filename` to **SWITCH/Non_Symmetric/datavis/farfield.output**.
- Copy `ccenters.filename` to **SWITCH/Non_Symmetric/datavis/ccenters.xyz**.
- Copy `current.txt.filename` to **SWITCH/Non_Symmetric/datavis/current.txt**.
- Go to **SWITCH/Non_Symmetric/datavis/**.
- Start up Matlab by typing *matlab*.
- In Matlab run the RCS visualization files to plot the RCS data. See the **Users Manual** for more information.
- In Matlab run the clustering visualization files to plot the clustering data. See the **Users Manual** for more information.
- Convert the file `current.txt` into `current.res` using `restxt` utility of PATRAN. See the **Users Manual** for more information.
- Start up PATRAN by typing *PATRAN*.
- Import the geometry files into PATRAN.

- Import **current.res** into PATRAN using the Analysis menu.
- Choose the PATRAN “Results” menu to display the surface currents.

References

- [1] G. E. Antilla and N. G. Alexopoulos, "Scattering from Complex 3D Geometries by a Curvilinear Hybrid Finite Element-Integral Equation Approach", *J. Opt. Soc. Am. A*, Vol. 11, No. 4, pp. 1445-1457, 1994
- [2] J. M. Song and W. C. Chew, "Multilevel Fast Multipole Algorithm for solving Combined Field Integral Equations of Electromagnetic Scattering", *Microwave and Opt. Tech. Letters*, Vol. 10, No. 1, pp. 14-19, September 1995
- [3] R. Coifman, V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription", *IEEE Antennas and Propagation Magazine*, Vol. 35, pp. 7-12, June 1993

Test Case Manual for SWITCH-FMM Version 1.0

John L. Volakis, Kubilay Sertel, and Michael A. Carr
Radiation Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2212, USA

April 5, 2000

Contents

1	Example Runs	3
1.1	The Ogive	3
1.2	Sphere	4
1.3	Dart	7
2	Example: Sphere	10
3	Quick Reference	14
3.1	Compilation of the Solver	14
3.2	Executing the Solver	14
3.3	Checking the Output	15
3.4	Visualizing the Outputs	15

1 Example Runs

This section is intended to demonstrate typical runs and outputs of the solver SWITCH-FMM.

1.1 The Ogive

In this example a PEC ogive is analyzed at 5.91 GHz. The geometry is the same as that used in the original SWITCH manual. The two geometry files are located in the directory **SWITCH/Non-Symmetric/geom** with the name **ogive480**. The dimension file for this problem given in **main.dim** is

```
parameter (  
  
& NNodes = 962,  
& Nntri = 240,  
& Nunknowns = 480,  
& Ncluster = 21,  
& NL = 8,  
& Nnzbimat = 113923  
&  
& )
```

The input file for the ogive depicted in Fig 1 is

```
"switch"  
"inches"  
5.91  
0.5  
"bistatic"  
2  
90.0 90.0 0.0  
0.0 180.0 0.5  
0.001  
500
```

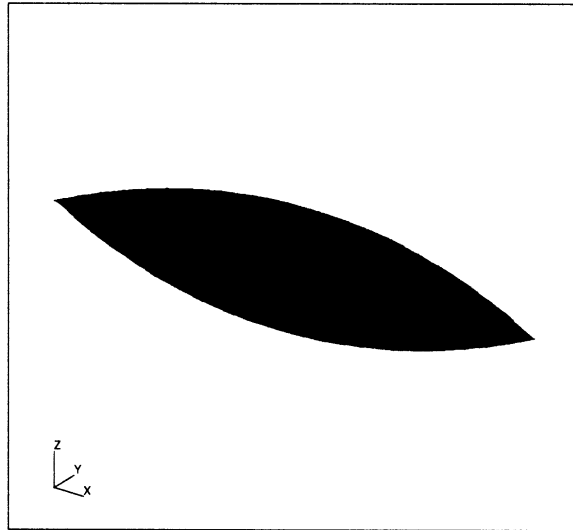



Figure 1: The Ogive Mesh

Fig 2 depicts the RCS result for the ogive480 geometry.

1.2 Sphere

In this example a PEC sphere of radius 1m. is analyzed at 0.3 GHz. The geometry is generated in PATRAN. The two geometry files are located in the directory **SWITCH/Non_Symmetric/geom** with the name **sphere05**. The dimension file for this problem given in **main.dim** is

```
parameter (  
  
& NNodes = 5494,  
& Nntri = 1373,  
& Nunknowns = 2746,  
& Ncluster = 52,  
& NL = 7,  
& Nnzbimat = 2448423  
&  
& )
```

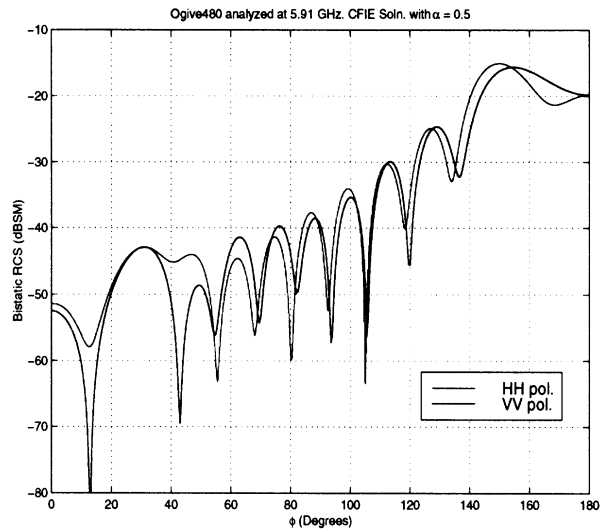


Figure 2: The RCS of the ogive at 5.91 GHz

The input file of the sphere depicted in Fig 3 is

```
"PATRAN"
"meters"
0.3
0.5
"bistatic"
2
90.0 90.0 0.0
0.0 180.0 0.5
0.001
500
```

Fig 4 depicts the RCS result for the sphere05 geometry and Fig 5 depicts the magnitude plot of the induced surface currents.

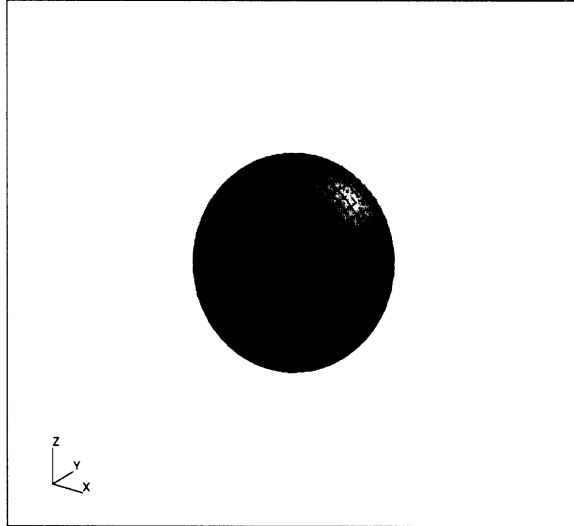


Figure 3: The Sphere Mesh

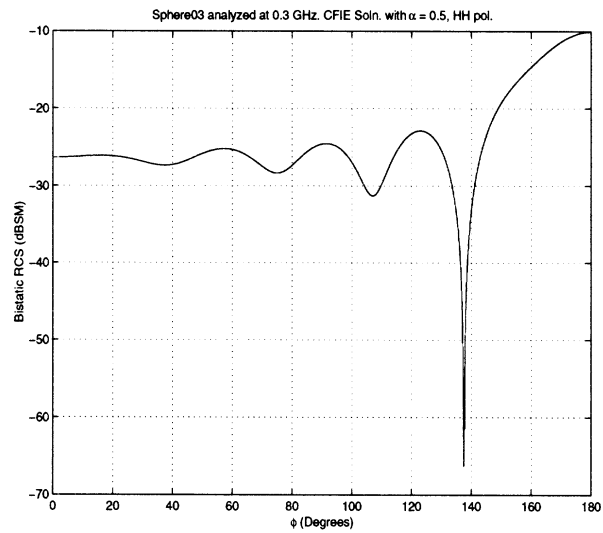


Figure 4: Bistatic RCS of the sphere at 0.3 GHz

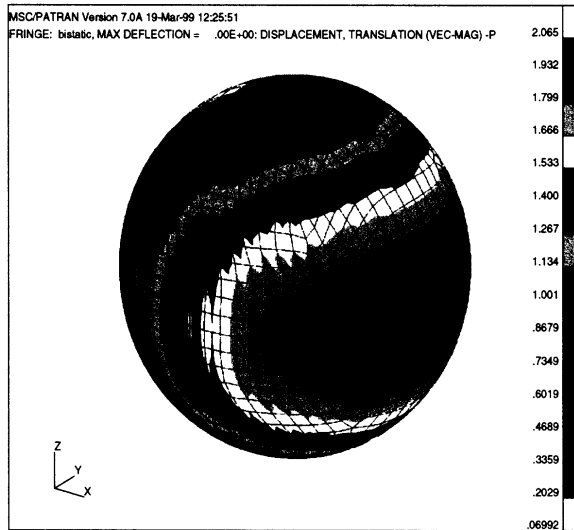


Figure 5: Induced surface currents on the sphere at 0.3 GHz

1.3 Dart

In this example the PEC dart is analyzed at 2 GHz. The geometry is generated using PATRAN. The two geometry files are located in the directory **geom** with the name **dart2**. The dimension file for this problem is given in **main.dim**. The input file of the dart depicted in Fig 6 is

```
parameter (
& NNodes = 8906,
& Nntri = 2226,
& Nunknowns = 4452,
& Ncluster = 66,
& NL = 11,
& Nnzbimat = 4248015
&
& )
```

The input file of the dart depicted in Fig 6 is

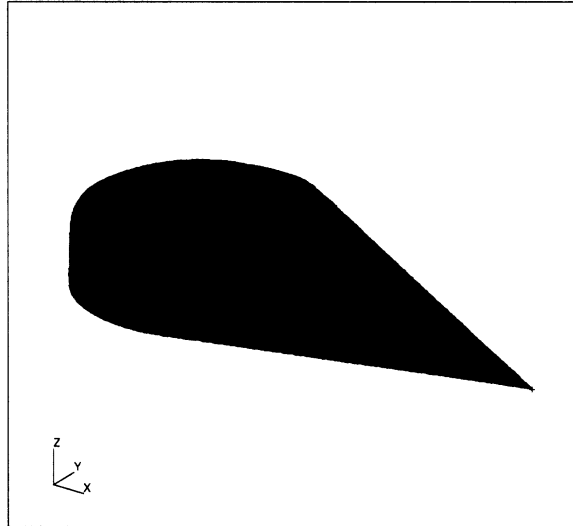


Figure 6: The Dart Mesh

```
"PATRAN"  
"inches"  
2.0  
0.5  
"bistatic"  
2  
90.0 90.0 0.0  
0.0 180.0 0.5  
0.001  
500
```

Fig 7 depicts the RCS result for the dart2 and Fig 8 depicts the magnitude plot of the induced surface currents.

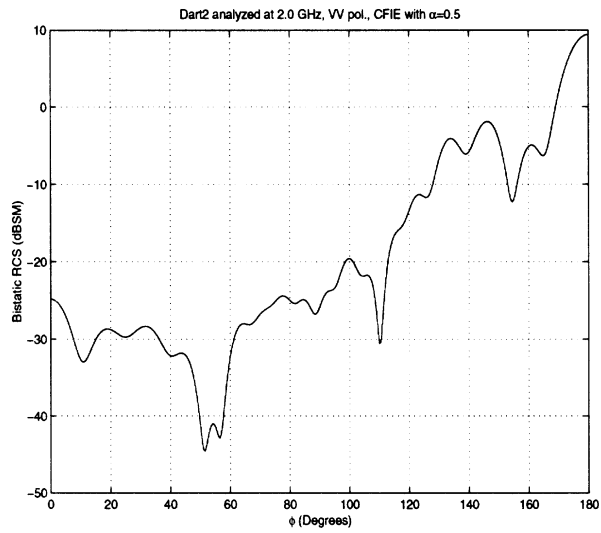


Figure 7: Bistatic RCS of the dart at 2.0 GHz

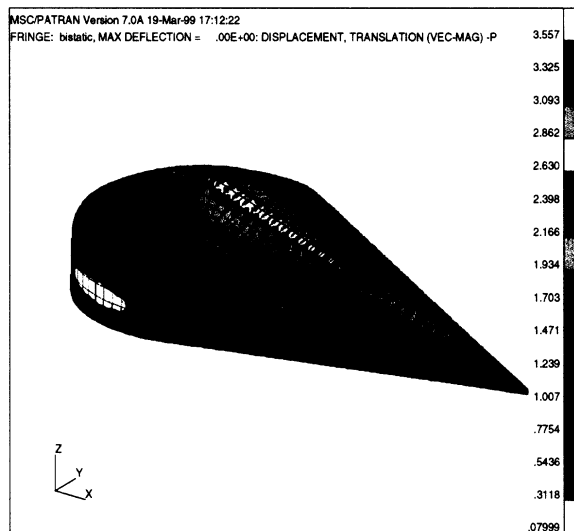


Figure 8: Induced surface currents the dart at 2.0 GHz, VV pol.

2 Example: Sphere

Given in this section is an example run to demonstrate the usage of the code. The sphere geometry is generated using PATRAN. The radius of the sphere is 1m. Two files are generated one of which contains the element nodes and the other contains the node coordinates. The file containing the element nodes reads as follows:

```
25 0 0 1 0 0 0 0 0
P3/PATRAN Neutral File from: /tmp/PATRAN/sphere.db
26 0 0 1 5494 1373 0 0 -1
18-Mar-99 20:05:39 3.0
2 1 4 2 0 0 0 0 0
9 0 0 0 .000000000E+00 .000000000E+00 .000000000E+00
3 63 62 1 443 442 441 380 1105
2 2 4 2 0 0 0 0 0
9 0 0 0 .000000000E+00 .000000000E+00 .000000000E+00
4 64 63 3 445 444 443 381 1106
:
2 1373 4 2 0 0 0 0 0
9 0 0 0 .000000000E+00 .000000000E+00 .000000000E+00
4478 4477 4508 4479 5077 5155 5154 5079 5494
99 0 0 1 0 0 0 0 0
```

This file is copied to **SWITCH/Non_Symmetric/geom/elements/**. The file containing the node coordinates reads as follows:

```
25 0 0 1 0 0 0 0 0
P3/PATRAN Neutral File from: /tmp/PATRAN/sphere.db
26 0 0 1 5494 0 0 0 -1
18-Mar-99 20:05:25 3.0
1 1 0 2 0 0 0 0 0
-1.000000000E+0-3.797475040E-20 8.687610093E-13
1G 6 0 0 000000
1 2 0 2 0 0 0 0 0
9.999999404E-1 -8.741933044E-8 0.000000000E+0
```

```

1G 6 0 0 000000
:
1 5494 0 2 0 0 0 0 0
-5.068947077E-1 7.210024595E-1 -4.724544585E-1
1G 6 0 0 000000
99 0 0 1 0 0 0 0 0

```

This file is copied to **SWITCH/Non_Symmetric/geom/nodes/**. For this problem the bistatic RCS of the sphere is computed at 0.3 GHz. for VV polarization. So the input file is as follows:

```

"PATRAN"
"meters"
0.3
0.5
"bistatic"
1
90.0 90.0 0.0
0.0 180.0 0.5
0.001
500

```

This file is copied to **SWITCH/Non_Symmetric/input/**. The dimension file **main.dim** in **SWITCH/Non_Symmetric/src/** reads as:

```

parameter (

& NNodes = 5494,
& Nntri = 1373,
& Nunknowns = 2746,
& Ncluster = 52,
& NL = 7,
& Nnzbimat = 2448423
&
& )

```

C NNodes : Number of NODES in target geometry

C Nntri : Number of ELEMENTS in target geometry
C Nunknowns : Number of UNKNOWNNS in target geometry
C Ncluster : Number of CLUSTERS in target geometry
C (Ncluster=sqrt(Nunknowns))
C NL : Number of MULTIPOLES in target geometry
C Nnzbimat : Number of NONZEROES in the near-field matrix

The solver is then compiled in **SWITCH/Non_Symmetric/src/** using

```
make f makefile.sun
```

Now the solver is ready to run. User must be in **SWITCH/Non_Symmetric/**. Type

```
runsw sphere05
```

to execute the solver. The following is output on the screen on a single processor SUN Ultra:

```
wavelength= 1.000000000000000 m.  
nnode,nquad= 5494 , 1373  
Geometry file read in = 0.710684  
Elements paired in = 3.40323  
nbasis= 2746  
nclus= 52  
Clusters formed in = 0.183546  
Clusters paired in = 5.19966E-02  
Translations computed in = 2.04550  
Aggregations computed in = 39.1651  
alpha= 0.500000000000000  
Sparse matrix structure formed in = 0.493427  
10% done.  
20% done.  
30% done.  
40% done.  
50% done.  
60% done.
```

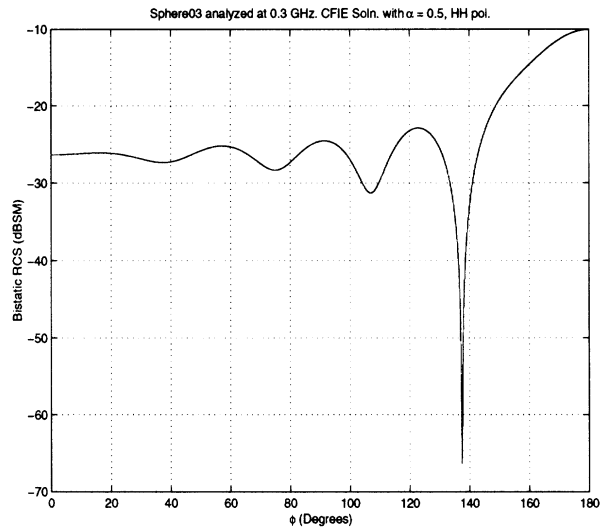


Figure 9: Bistatic RCS of the sphere at 0.3 GHz

70% done.

80% done.

90% done.

100% done.

Near-field matrix filled in = 1318.00

Non-zeros in preconditioner= 80354

Near-field matrix factored in = 204.301

1 of 1 is being processed.

ITER= 1 ERROR= 0.29340707366055

ITER= 2 ERROR= 8.9859981758242D-02

ITER= 3 ERROR= 8.4873548373478D-02

ITER= 4 ERROR= 5.5020817083269D-03

ITER= 5 ERROR= 3.2430174817679D-04

ITER= 5 ERROR= 3.2430174817679D-04

Solution computed in = 15.1542

Time per Iteration = 3.03085

After a successful run, the output files are moved in the directory **SWITCH/Non_Symmetric/output**. Fig 9 is generated using the visualization tool **ffphi.m** in **SWITCH/Non_Symmetric/datavis/**.

3 Quick Reference

This section is a step-by-step reference to execute the code SWITCH-FMM. It is assumed that the directory structure of the code is already created on user's account. To use the "symmetric" version of the solver, users should execute the steps below in the directory **SWITCH/Symmetric/** as opposed to executing them in **SWITCH/Non_Symmetric/**. It should be kept in mind that the geometry and input files of the "symmetric" version are different than those of the "non-symmetric" version.

3.1 Compilation of the Solver

- Design the parameter file **main.dim** subject to the geometry size.
- Type *touch*.f* so that each file includes the updated **main.dim**
- Compile the code by executing *make makefile_XXX* using the proper *makefile_XXX* where "XXX" refers to the machine type such as "sun", "hp", or "sgi".
- Now the executable **sw** is generated and moved in the directory **SWITCH/Non_Symmetric/bin**

3.2 Executing the Solver

- Put the geometry files in **SWITCH/Non_Symmetric/geom**. See the **User's Manual** for more information.
- Select the run parameters and put the input file in **SWITCH/Non_Symmetric/input**
- Go to directory **SWITCH/Non_Symmetric**
- Execute the solver by typing *runsw filename*. The name of the geometry files and the input file should all be the same.

3.3 Checking the Output

- Go to directory **SWITCH/Non_Symmetric/output**
- Check that all four output files are generated after the simulation is completed. These are
 1. `ccenters.filename`
 2. `coeff.filename`
 3. `current.txt.filename`
 4. `farfield.filename`

3.4 Visualizing the Outputs

- Copy `farfield.filename` to **SWITCH/Non_Symmetric/datavis/farfield.output**.
- Copy `ccenters.filename` to **SWITCH/Non_Symmetric/datavis/ccenters.xyz**.
- Copy `current.txt.filename` to **SWITCH/Non_Symmetric/datavis/current.txt**.
- Go to **SWITCH/Non_Symmetric/datavis/**.
- Start up Matlab by typing *matlab*.
- In Matlab run the RCS visualization files to plot the RCS data. See the **User's Manual** for more information.
- In Matlab run the clustering visualization files to plot the clustering data. See the **User's Manual** for more information.
- Convert the file `current.txt` into `current.res` using `restxt` utility of PATRAN. See the **User's Manual** for more information.
- Start up PATRAN by typing *PATRAN*.
- Import the geometry files into PATRAN.

- Import **current.res** into PATRAN using the Analysis menu.
- Choose the PATRAN “Results” menu to display the surface currents.

Mathematical Background for SWITCH-FMM

Version 1.0

John L. Volakis, Kubilay Sertel, and Michael A. Carr
Radiation Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2212, USA

April 5, 2000

Contents

1	Introduction	3
2	Geometry Modeling and Basis Functions	3
3	Integral Equation Formulation and Method of Moments	5
4	Fast Multipole Method	6
5	Iterative Solver and Preconditioning	7

1 Introduction

SWITCH-FMM is a Method of Moments (MoM) based Computational Electromagnetics (CEM) tool to simulate electromagnetic (EM) scattering scenarios involving arbitrarily curved perfect electric conductor (PEC) targets. The MoM formulation used in the implementation of the code is based on the Electric-Field-Integral-Equation (EFIE), the Magnetic-Field-Integral-Equation (MFIE), and the Combined-Field-Integral-Equation (CFIE) formulations of EM scattering from PEC targets. To overcome the computation time length and memory requirements of the conventional MoM implementations ($O(N^3)$ and $O(N^2)$ respectively), the Fast Multipole Method (FMM) is implemented. The FMM reduces the computation complexity to $O(N^{1.5})$ per iteration (in iterative solution of the resulting linear system) and the memory to $O(N^{1.5})$.

The target geometry is represented in the code using curved quadrilateral surface patches which are defined by 9 points in space located on a topologically rectangular 3×3 grid [1].

2 Geometry Modeling and Basis Functions

SWITCH-FMM uses curved quadrilateral biquadratic surface elements to model the PEC target geometry under investigation. Surface modeling using curved surface elements has the advantage of decreasing the geometry modeling error substantially [4]. The implementation in SWITCH-FMM can easily be generalized for higher order surface representations.

A biquadratic surface patch is defined by 9 points in space (on a topologically rectangular 3×3 grid) as depicted in Fig 1. The mathematical representation in terms of two parameters (u and v) of the patch is given by

$$\mathbf{r}(u, v) = \sum_{n=0}^2 \sum_{m=0}^2 \mathbf{a}_{nm} u^n v^m \quad (1)$$

with the restriction $0 < u < 1$ and $0 < v < 1$. In (1), \mathbf{a}_{nm} are constant vector coefficients related to the nine \mathbf{r}_{nm} points defining the curved element.

The basis functions approximating the unknown surface current density induced on the PEC surface by the incident EM field needs to be defined on these quadrilateral elements, conformal to the surface model. Hence, they are

defined in terms of the surface tangent vectors on the quadrilateral element. Each element pair supports a *curved rooftop* basis function, and each edge of a patch may support one half of the basis function. Four half rooftop basis functions are used to represent the current density on a single surface patch. These are

$$\mathbf{f}_u^{(1)} = \frac{u}{\sqrt{g}} \mathbf{t}_u, \quad \mathbf{f}_u^{(2)} = \frac{(1-u)}{\sqrt{g}} \mathbf{t}_u, \quad \mathbf{f}_v^{(1)} = \frac{v}{\sqrt{g}} \mathbf{t}_v, \quad \mathbf{f}_v^{(2)} = \frac{(1-v)}{\sqrt{g}} \mathbf{t}_v, \quad (2)$$

in which g is the determinant of the metric tensor of the parametric transformation given in (1) and \mathbf{t}_u and \mathbf{t}_v are the two principle surface tangent vectors

$$\mathbf{t}_u = \frac{\partial \mathbf{r}}{\partial u}, \quad \mathbf{t}_v = \frac{\partial \mathbf{r}}{\partial v} \quad (3)$$

on the curved element. Note that in each of the above expansion components (2), g serves as a normalization factor so that current continuity across neighboring patches is ensured.

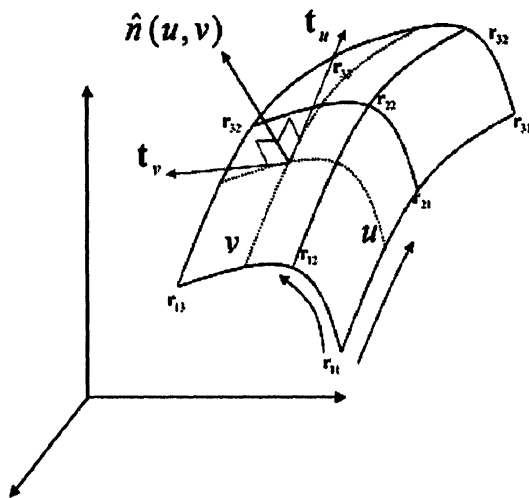


Figure 1: Geometrical construction of the surface elements.

3 Integral Equation Formulation and Method of Moments

For conducting objects, the EFIE is given by

$$\hat{t} \cdot \int_s \left[\mathbf{J}(\mathbf{r}') + \frac{1}{k^2} \nabla' \cdot \mathbf{J}(\mathbf{r}') \nabla \right] \frac{e^{ikR}}{R} ds' = \frac{4\pi i}{k\eta} \hat{t} \cdot \mathbf{E}^i(\mathbf{r}), \quad (4)$$

where

$$R = |\mathbf{r} - \mathbf{r}'|. \quad (5)$$

Equation (4) is the statement of the boundary condition on the tangential component of the electric field on a conducting surface. The vector denoted by \hat{t} is any unit tangent vector on the surface s of the scatterer, and $\mathbf{E}^i(\mathbf{r})$ is an impressed field which excites the system.

The EFIE for the unknown electric current density $\mathbf{J}(\mathbf{r})$ on the conducting surface induced by an incident wave is discretized using the MoM technique. The induced surface current is approximated by a sum of N known basis functions $\{\mathbf{j}_n(\mathbf{r})\}$ as

$$\mathbf{J}(\mathbf{r}) \approx \sum_{n=1}^N a_n \mathbf{j}_n(\mathbf{r}). \quad (6)$$

The EFIE thus becomes

$$\sum_{n=1}^N a_n \hat{t} \cdot \int_s \left[\mathbf{j}_n(\mathbf{r}') + \frac{1}{k^2} \nabla' \cdot \mathbf{j}_n(\mathbf{r}') \nabla \right] \frac{e^{ikR}}{R} ds' - \frac{4\pi i}{k\eta} \hat{t} \cdot \mathbf{E}^i(\mathbf{r}) \approx 0 \quad (7)$$

Using standart MoM procedure with Galerkin's testing, the system of equations obtained is,

$$\sum_{n=1}^N Z_{mn} a_n = F_m, \quad m = 1, 2, \dots, N, \quad (8)$$

where

$$Z_{mn} = \int_s ds \mathbf{t}_m(\mathbf{r}) \cdot \int_{s'} ds' \left[\mathbf{j}_n(\mathbf{r}') + \frac{1}{k^2} \nabla' \cdot \mathbf{j}_n(\mathbf{r}') \nabla \right] \frac{e^{ikR}}{R} \quad (9)$$

and

$$F_m = \frac{4\pi i}{k\eta} \int_s ds \mathbf{t}_m(\mathbf{r}) \cdot \mathbf{E}^i(\mathbf{r}). \quad (10)$$

Direct application of the MoM requires the computation of N^2 double surface integrals appearing in Eq. (9) as the elements of the resulting MoM matrix. Solution of this system of equations by Gaussian elimination requires $O(N^3)$ operations. Iterative solvers require $O(N^2)$ operations per iteration. The memory requirement of the MoM is also $O(N^2)$. This large order for storage limits the size of the problem that can be solved on a given hardware, and the high operation cost poses a limit to the size of problems that can be solved in a practically acceptable period of time. For these reasons, the FMM is proposed [3, 2], which requires less memory and CPU time for the solution of large problems.

4 Fast Multipole Method

Spherical (multipole) wave expansion of the free-space Green's function appearing in the MoM formulation of the electric/magnetic field integral equation (E/MFIE) forms the basis of the FMM algorithm. This enables the modified representation of the MoM matrix elements (for EFIE) as [3]

$$Z_{ij} \approx \frac{ik}{4\pi} \int d^2\hat{k} \mathbf{V}_{fmi}(\hat{k}) \cdot T_L(kr_{mm'}, \hat{k} \cdot \hat{r}_{mm'}) \mathbf{V}_{sm'j}^*(\hat{k}), \quad (11)$$

where

$$\begin{aligned} \mathbf{V}_{sm'j}(\hat{k}) &= \int_S ds' e^{i\mathbf{k} \cdot \mathbf{r}_{jm'}} [\bar{\mathbf{I}} - \hat{k}\hat{k}] \cdot \mathbf{j}_j(\mathbf{r}_j), \\ \mathbf{V}_{fmi}(\hat{k}) &= \int_S ds e^{i\mathbf{k} \cdot \mathbf{r}_{im}} [\bar{\mathbf{I}} - \hat{k}\hat{k}] \cdot \mathbf{t}_i(\mathbf{r}_i) \end{aligned} \quad (12)$$

and

$$T_L(kr, \hat{k} \cdot \hat{r}) = \sum_{l=0}^L i^l (2l+1) h_l^{(1)}(kr) P_l(\hat{k} \cdot \hat{r}). \quad (13)$$

The translation operator, being only dependent on the prechosen vector $\mathbf{r}_{mm'}$ between the points m and m' , is key element to reducing the $O(N^2)$ complexity of the MoM matrix-vector product in the iterative solution of the resulting matrix equations. By grouping the basis functions into a prespecified number of clusters and by reusing the translation operator to compute the interactions of the basis and testing functions in far clusters, the $O(N^2)$ complexity of MoM can be reduced down to $O(N^{1.5})$. This is the conventional FMM algorithm [3].

For clustering the k -means algorithm [6] is implemented to form the clusters. This algorithm aims to minimize a defined error function for the purpose of achieving a fairly uniform set of groups. The specific error function used by the k -means algorithm minimizes the sum of the distances between the cluster centers and the centers of the basis functions in each cluster.

5 Iterative Solver and Preconditioning

FMM achieves its speed-up by using an indirect fast computation of the matrix vector product in the context of an iterative solution of the MoM matrix. Various iterative solvers [7] are available including the conjugate gradient (CG), biconjugate gradient (BiCG), conjugate gradient squared (CGS), quasi-minimal residual (QMR), and generalized minimal residual (GMRES). The convergence of all these iterative solvers is, of course, dictated by the matrix condition which typically deteriorates as the matrix size increases. Therefore, iterative solutions of such large, fully populated matrix systems inevitably require some kind of preconditioning. Otherwise, propagation of numerical errors during the execution of the iterative solution may lead to convergence failures.

Various preconditioning techniques for improving the condition of the system can be found in [7]. Diagonal and block diagonal preconditioners for multilevel FMM implementations were reported in [5].

Among other solvers and preconditioners implemented, the CGS solver and the ILU preconditioner have provided the best performance, for all EFIE, MFIE, and CFIE formulations. By its nature, the ILU preconditioner is constructed using the near-field FMM matrix and significantly reduces the number of iterations required for convergence [8].

References

- [1] G. E. Antilla and N. G. Alexopoulos, "Scattering from Complex 3D Geometries by a Curvilinear Hybrid Finite Element-Integral Equation Approach", *J. Opt. Soc. Am. A*, Vol. 11, No. 4, pp. 1445-1457, 1994
- [2] J. M. Song and W. C. Chew, "Multilevel Fast Multipole Algorithm for solving Combined Field Integral Equations of Electromagnetic Scattering", *Microwave and Opt. Tech. Letters*, Vol. 10, No. 1, pp. 14-19, September 1995
- [3] R. Coifman, V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription", *IEEE Antennas and Propagation Magazine*, Vol. 35, pp. 7-12, June 1993
- [4] K. Sertel and L. Gürel, "Comparison of Surface Modeling Techniques", *IEEE AP-S Conference Digest*, July 1997.
- [5] J. M. Song, C. C. Lu, and W. C. Chew, "Multilevel Fast Multipole Algorithm for Electromagnetic Scattering by Large Complex Objects", *IEEE Trans. Antennas Propagat.*, vol. AP-45, no. 10, pp. 1488-1493, 1997.
- [6] Griffiths, P. and Hill, I. D., *Applied Statistics Algorithms*, West Sussex, England, Ellis Horwood, 1985
- [7] Saad, Y., *Iterative Methods for Sparse Linear Systems*, Boston: PWS Pub. Co., 1996.
- [8] K. Sertel and J. Volakis, "Incomplete LU Preconditioner for FMM Implementation", *ACES Conference Digest*, March 2000.

On the Selection of FMM Parameters

K. Sertel* and J. L. Volakis

Radiation Laboratory

Department of Electrical Engineering and Computer Science

University of Michigan

Ann Arbor, MI 48109-2212, USA

Abstract

This paper investigates the effects of the fast multipole method (FMM) parameters on the radar cross section (RCS) computations of different target geometries. In our formulation, the integral equation is solved for perfect electric conductor targets using biquadratic parametric surface modeling and curved rooftop basis functions to minimize the geometry modeling error induced on the solution. Guidelines for the selection of FMM parameters are revisited and numerical examples are provided to demonstrate the dependence of RCS error on FMM parameters.

1 Introduction

The fast multipole method (FMM) [1, 2] and other fast algorithms such as the adaptive integral method (AIM) [3, 4, 5, 6, 7] has been successfully applied to speed-up moment method (MoM) solutions of scattering and radiation problems. Fast algorithms have also been incorporated within the context of hybrid finite element-boundary integral formulations for solving complex electromagnetic media problems [6, 7, 8, 9, 10]. In the case of FMM

*K. Sertel was funded by NPACI. The computer code used for this work, FMM-SWITCH, was based on the SWITCH code originally developed at Northrop Grumman Corp.

a number of variations have been reported including the fast far field approximation (FAFFA), windowed FMM, and multilevel FMM [2, 5, 11, 12, 13, 14, 15, 16]. Standard FMM reduces the computation and memory costs of conventional MoM from $O(N^2)$ to $O(N^{1.5})$ per iteration and as a result large scale problems have been carried out involving over a million unknowns.

The FMM implementation attains its speed-up through a number of approximations. However these approximations lead to errors which must be understood and controlled for its reliable utilization. Previous studies assessed FMM errors by examining the approximation of the pertinent Green's function used for evaluating the MoM matrix elements. Practice, however, dictates that the error between FMM-MoM and uncompressed MoM solutions also depends on the excitation, the specific structure under analysis, and the quantity of interest.

In this paper, we consider the effect of FMM parameters on RCS computations for various targets. Namely, our study focuses on the computation of the observable parameter (the RCS for our case) and not just on the matrix elements. Choices relating to the number of multipoles used in the expansion of the Green's function and the near-field threshold parameter are both examined to obtain an integrated measure of the error in FMM reflected on RCS computations. To minimize the geometry modeling error, which can be significant for doubly curved complex target geometries, biquadratic parametric surface patches are used [17, 18]. The paper concludes with specific recommendations for optimum values of the FMM parameters for minimizing RCS error.

2 FMM Formulation and Parameters

This section outlines the MoM and FMM formulations on the basis of biquadratic surface patch models of PEC scatterers. The FMM parameters under investigation are also defined in this section.

2.1 Moment Method for Curvilinear Elements

Since our goal is to evaluate errors associated with the FMM formulation, a key decision in our modeling approach was to reduce error associated with geometry modeling. By using 2^{nd} order curvilinear quadrilateral elements [18]

for surface modeling instead of the usual triangular facets [19], we can better suppress errors associated with geometry modeling. More specifically, the FMM implementation used herewith was built upon that in [18] and, as such, linear basis functions were employed to approximate the current density on each of the quadrilateral patches forming the geometry (see Fig. 1 for a view of the quad-tessellated geometries to be considered).

Briefly, for the electric field integral equation (EFIE) the resulting linear system after Galerkin's testing is of the form

$$\sum_{n=1}^N Z_{mn} a_n = V_m, \quad m = 1, 2, \dots, N \quad (1)$$

where

$$Z_{mn} = \int_s ds \mathbf{f}_m(\mathbf{r}) \cdot \int_{s'} ds' \left[\mathbf{f}_n(\mathbf{r}') + \frac{1}{k^2} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') \nabla \right] \frac{e^{ikR}}{R} \quad (2)$$

and

$$V_m = \frac{4\pi i}{k\eta} \int_s ds \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}^i(\mathbf{r}). \quad (3)$$

An $e^{-i\omega t}$ time convention has been assumed and suppressed and $\{a_n\}$ refers to the column containing unknown coefficients of the surface current expansion

$$\mathbf{J}(\mathbf{r}) \approx \sum_{n=1}^N a_n \mathbf{j}_n(\mathbf{r}). \quad (4)$$

In the above expressions \mathbf{r} and \mathbf{r}' denote the usual observation and source point locations, $\mathbf{E}^i(\mathbf{r})$ is the incident excitation plane wave at \mathbf{r} , $\hat{\mathbf{t}}$ is the vector tangent to the surface at \mathbf{r} , $\eta = 120\pi$ denotes the free space impedance, and $k = 2\pi/\lambda$ is the free space wavenumber.

The linear expansion functions $\mathbf{f}_n(\mathbf{r})$ must be defined on the quadrilateral elements. Other than being defined on a curved surface, they are very similar to the roof-top basis functions traditionally used for representing currents on a rectangular or square flat patch. In fact, their representation is best done by first introducing the biquadratic parametric representation (see Fig. 1)

$$\mathbf{r}(u, v) = \sum_{p=0}^2 \sum_{q=0}^2 \mathbf{a}_{pq} u^p v^q \quad (5)$$

where \mathbf{a}_{pq} are the transformation coefficients determined from the original 9-points used to specify the quad in the (x,y,z) coordinate-space. In this expansion $0 < u < 1$ and $0 < v < 1$ with (u=0,v=0) corresponding to the left bottom corner of the patch.

Based on (5), the surface tangent vectors at any point on the biquadratic patch are given by (these are not normalized)

$$\mathbf{t}_u = \frac{\partial \mathbf{r}}{\partial u}, \quad \mathbf{t}_v = \frac{\partial \mathbf{r}}{\partial v} \quad (6)$$

and to ensure continuity of the currents across the patches, we introduce the following representation

$$\mathbf{f}_u(u, v) = \sum_{i=1}^4 \mathbf{f}_u^i(u, v) \quad (7)$$

where

$$\mathbf{f}_u^{(1)} = \frac{u}{\sqrt{g}} \mathbf{t}_u, \quad \mathbf{f}_u^{(2)} = \frac{(1-u)}{\sqrt{g}} \mathbf{t}_u, \quad \mathbf{f}_v^{(1)} = \frac{v}{\sqrt{g}} \mathbf{t}_v, \quad \mathbf{f}_v^{(2)} = \frac{(1-v)}{\sqrt{g}} \mathbf{t}_v, \quad (8)$$

in which g is the determinant of the metric tensor of the parametric transformation given in (5). Note that in each of the above expansion components (8), g serves as a normalization factor so that the functions are unity or zero at the appropriate boundary edge of the patch. Thus, they ensure current continuity across neighboring patches.

We can define each of (8) using the generic form

$$\mathbf{j}_w = \frac{1}{\sqrt{g}} \tilde{w} \mathbf{t}_w, \quad (9)$$

to obtain the generic form for the matrix entries as,

$$Z_{mn} = \int_s dudv \tilde{w} \mathbf{t}_w \cdot \int_{s'} du' dv' \left[\tilde{w}' \mathbf{t}'_w + \frac{1}{k^2} \nabla' \cdot \tilde{w}' \mathbf{t}'_w \nabla \right] \frac{e^{ikR(u,v)}}{R(u,v)}. \quad (10)$$

Note that for the matrix elements in (10) \sqrt{g} is not present due to its cancellation by the Jacobian of the parametric transformation (since $ds = \sqrt{g} dudv$).

The general rule of thumb for RCS computation is to discretize the scatterer surface using an average edge length of about $\lambda/10$. This rule is not violated in this study. Basically, great care was used to ensure excellent geometrical fidelity so that error in the solution can be primarily attributed to the numerical FMM solution.

2.2 FMM Implementation

The key aspects of the FMM implementation are

- Iterative solution of the MoM system of equations $[Z]\{a\} = \{V\}$
- Fast evaluation of the matrix vector product $[Z]\{a\}$.

Fast evaluation of the matrix vector product (using $O(N^{1.5})$ or less resources, instead of $O(N^2)$) is attained by introducing the approximation [1]

$$\frac{e^{ik|\mathbf{r}+\mathbf{d}|}}{|\mathbf{r}+\mathbf{d}|} = \frac{ik}{4\pi} \int d^2\hat{k} e^{i\mathbf{k}\cdot\mathbf{d}} T_L(kr, \hat{k} \cdot \hat{r}), \quad (11)$$

where

$$T_L(kr, \hat{k} \cdot \hat{r}) = \sum_{l=0}^L (2l+1) h_l^{(1)}(kr) P_l(\hat{k} \cdot \hat{r}) \quad (12)$$

which is referred to as the translation operator. To take advantage of this expansion, the quadrilateral elements are clustered into M groups where r refers to the center-to-center distances and $\mathbf{d} = \mathbf{r}_{no} + \mathbf{r}_{mo}$ is the sum of the distances between the centers of the m th/ n th observation/integration points. Because of the grouping, $T_L(kr, \hat{k} \cdot \hat{r})$ is an expansion independent of the specific observation/integration point. It only depends on the inter-distances between the a priori chosen groups and is precomputed prior to the iterative solver execution. Also, \mathbf{d} is relatively small and thus the integral in (11) can be numerically evaluated quickly using small number of integration points. It can be shown [1] that if the number of element clusters is chosen as $M = \sqrt{N}$, then the CPU time per iteration/matrix vector product is $O(N^{1.5})$ instead of $O(N^2)$. Multilevel FMM can further reduce the CPU time down to $O(N \log N)$.

The above CPU estimates are asymptotic in the sense that they represent values which are approached for very large N . The actual efficiency of the implementation depends on the choice of various parameters as described below. These choices control the constant in front of the asymptotic behavior of the CPU requirements. However, choices of the FMM parameters for faster implementation inevitably lead to less accurate answers. This is the subject of the next section. At this point we will restrict ourselves to a qualitative description of the parameters.

2.2.1 Multipole Sum Computation

A most important parameter in the FMM implementation is the number of terms kept for the evaluation of T_L in (12). This parameter, commonly denoted as L [1], is semi-empirically chosen as

$$L = kD_{max} + \alpha \ln(kD_{max} + \pi) \quad (13)$$

where D_{max} is the maximum diameter of the clusters and α is an accuracy control parameter. For $\alpha = 5$, the resulting value of L gives single (4-digit precision) in evaluating T_L ; for $\alpha = 10$, double precision is obtained in evaluating T_L . Typically, however, it has been reported [12] that setting $\alpha = 1$ gives acceptable accuracy.

Clearly, keeping L as small as possible reduces the CPU time but not the order $CN^{1.5}$ for the matrix vector product computation ($C = constant$). Small L simply leads to smaller value for C and consequently, the CPU time crossover point between standard LU and FMM implementations occurs for lower values of N .

2.2.2 Aggregation/Disaggregation integrals

The spectral integral in (11) is discretized using L points over the θ angular sector (Gaussian integration) and $2L$ points in the ϕ sector (trapezoidal rule of integration). These choices of integration points and associated rules will be maintained throughout our study.

2.2.3 Clustering

Another key aspect of the FMM is element grouping. The associated element groups are called clusters and a grouping algorithm must be used. Here, we used the k -means algorithm [20] to form the clusters. This algorithm aims to minimize a defined error function for the purpose of achieving a fairly uniform set of groups. The specific error function used by the k -means algorithm minimizes the sum of the distances between the cluster centers and the centers of the basis functions in each cluster.

If the resulting clustering is not uniform, the FMM implementation will be inefficient due to the presence of very large or very small clusters. This affects the accuracy of the spectral integral since the number of terms used in the translation sum will depend on the size of the largest cluster.

2.2.4 Near and Far Clusters

In the FMM implementation, the approximation (11) is only used for computing the interactions of elements between clusters. Within a simple cluster, the elements are interacted without the FMM approximation. For greater accuracy, elements between adjacent or distant clusters can be computed without approximation as well. Since our interest in this paper relates to the FMM error evaluation, we introduce an additional parameter Δ for controlling the transition distance between the exact and the FMM approximation. Specifically, if d_{ij} is the distance between the i th and j th clusters, the FMM approximation was involved only if

$$d_{ij} > \Delta(r_i + r_j) \quad \text{and} \quad kd_{ij} > L \quad (14)$$

where $r_{i,j}$ denote the radii of the i th and j th clusters. Clearly, if the quantity $\Delta = 1$, then even neighboring clusters will be considered as far zone (and thus FMM will be used for their interactions). For $\Delta = 1.5$, the elements in the neighboring (touching) clusters will be evaluated without approximation. Basically, higher Δ implies that more interactions will be carried out without approximation, thus increasing the near-field matrix size.

3 Error Study

Based on the above formulations and defined FMM parameters, in this section we present error evaluation of the FMM implementation for a number of geometries. We then summarize the results.

3.1 Target Geometries

Three representative geometries of different geometric and electromagnetic properties were investigated for a more complete and unbiased study. The first geometry, depicted in Fig. 2 (a) is a 1 m radius PEC sphere. The sphere is a canonical geometry which has an analytical solution and is typically used for code benchmarking. For our study the standard MoM solution is used to benchmark the FMM-MoM approximation.

The second geometry considered is an ogive (see Fig. 2 (b)). Unlike the sphere, it has two sharp tips that cause strong scattering contributions near

edge-on incidences. In contrast, a large sphere is dominated by specular scattering.

The third geometry analyzed is an elongated strip. This strip is simply a flat plate having an aspect ratio of 40/1 as depicted in Fig. 2 (c). This target is chosen because it consists of a localized geometry and has sharp edges which cause substantial diffraction. When discretized, the clusters are also localized and are more non-uniformly distributed compared to the sphere or the ogive.

3.2 Error Definition

For each of these three geometries tip-on incidence is assumed with horizontal polarization (see Fig. 3). We choose this incidence because it is more stressing case for scattering computations. The scattered far-field is sampled on 332 points on a half sphere covering the entire $z < 0$ space. As noted earlier, the error curves were generated using the MoM solution as reference. The error values were computed by the l_∞ norm of the error vector at the sample points on the half sphere, and are normalized to the maximum value of the error vector to get the percent error.

3.3 Iterative Solver and Preconditioning

The solution of the matrix system is carried out using a preconditioned conjugate gradient squared (CGS) iterative solver is used. The CGS requires 2 matrix-vector products in each iteration, but usually converges twice as fast as the conjugate gradient (CG) solver which requires one matrix-vector product. A relative convergence criterion of 10^{-3} was used for terminating the CGS routine.

The FMM approximation deteriorates the matrix condition and preconditioning was found essential, and resulted in significant reduction of the CPU time to achieve convergence. The near field matrix (containing the interactions computed without FMM approximation) itself is a good candidate for the preconditioner. However, the necessity of inverting (or LU decomposing) the near field matrix to obtain the preconditioner [9] using less than $O(N^{1.5})$ prompted us to use a reduced version of the near field matrix. For CFIE formulations we can obtain a reduced band matrix by filtering out the elements which are smaller than 10 percent in magnitude than the largest

element (usually a diagonal element). This provides for a convergence rate similar to that obtained when the entire near-field matrix is kept.

3.4 Number of Multipoles, L

Beginning with the first target geometry, Fig. 4 shows the FMM solution error for a 1 m radius sphere at 0.3 GHz. The discretization employed biquadratic surface patches which resulted in 2746 unknowns. We observe clearly that the error decreases as more multipole terms are and this was to be expected. Some oscillations are observed (i.e. convergence is not monotonic) and these are likely to be due to the asymptotic nature of the multipole expansion. It should be noted that the grouping and the discretization are fairly uniform and thus these oscillations are not due to clustering variations. For this example the near-field threshold was set to $\Delta = 1.5$.

The FMM solution error curves for the ogive are given in Fig. 5 (edge-on incidence). Its width was 2 m (1 wavelength), i.e. the frequency was 150 MHz. The discretization resulted in 480 unknowns. Similar to the sphere results, the error is decreasing as a function of L and some oscillations in the error curve are again observed. These oscillations are more pronounced for the strip geometry for which the error curve is given in Fig. 6. The frequency of the incident plane wave was 300 MHz, the strip was 20 m (20 wavelengths) long and 0.5 m (0.5 wavelengths) wide, and the discretization resulted in 1795 unknowns. Clearly, the most pronounced difference among the error curves in Figs. 4 – 6 is the error for the error for the strip. This is possibly due to the stronger multiple diffraction mechanisms characterizing its RCS at edge-on incidence.

Although the error analysis based on the matrix elements forms a basis for the implementation of the FMM, the error in matrix entries affects the convergence of the solver and causes unexpected error behavior in the final solution as observed in these plots. The above error curves indicate that, care should be taken when implementing and optimizing FMM to yield a faster solution of MoM problems. The faster solution should also be an accurate one especially for scatterers where multiple interactions or resonant behaviors are present. Basically, the number of multipoles should be chosen judiciously for each geometry prior to making design runs.

3.5 Near-field Threshold, Δ

Keeping L small is certainly advantageous to attain maximum speed-up by using the least memory for the FMM implementation. However, speed and accuracy are also dependent on the near-field threshold. Thus, it is essential to look at the overall RCS accuracy when L and Δ are both varied.

Fig. 7 shows the effect of the near field threshold parameter Δ on the accuracy of the computed bistatic RCS for a sphere. It is important to note here that regardless of the choice of L , the curves nearly overlay for $\Delta > 3.5$. This important observation means that the RCS error is the same regardless of the L parameter choice. This is also observed for the ogive geometry as shown in Fig. 8. Again beyond $\Delta \approx 3.5$, the error becomes independent of L . The conclusion that can be drawn from these observations is one can use a smaller number of multipoles and demand the same level of accuracy by choosing the near field threshold larger. This is consistent with previous error analysis based on isolated Green's function errors [1] where it was noted that fewer number of multipoles give accurate approximations for far clusters. A key conclusion of our work is that choosing $\Delta \approx 3.5$ gives an optimum choice for the near field threshold parameter.

When the strip geometry is considered we could no longer observe a consistent error behavior as a function of the near field threshold parameter for edge on incidence (Fig. 9). This is due to the very low level of the RCS. This observation is in contrast to earlier comments that the parameters can be chosen regardless of geometry and RCS level to attain given error criterion [1, 17]. When we consider broadside incidence for the strip the error curves (see Fig. 10) are again consistent with those for the sphere and the ogive.

3.6 General Observations

Numerical error results for bistatic RCS computations using the FMM for three different representative geometries were presented. The effects of the number of multipoles and the near-field threshold parameter were examined. In contrast to earlier observations it was demonstrated that care should be taken when choosing FMM parameters for fast MoM implementations and that the parameter choices are geometry dependent for a given error criterion.

Our observations can be summarized as follows:

- Error of the FMM solution depends on the target geometry under investigation.
- Error also depends on the excitation (incidence direction and polarization).
- Overall RCS error behavior does not correlate with error analysis based solely on the expansion of the Green's function. In that case the geometry and excitation do not enter into the analysis.
- For the three representative geometries examined, it was shown that choosing the near-field threshold parameter as $\Delta \approx 3.5$ produces optimum error performance (almost independent of the number of multipoles used for $L > kD_{max} + \ln(kD_{max} + \pi)$). This increases the near field computations, hence the optimization of the FMM algorithm should be done accordingly.

The given error curves in Figs. 8 – 10 can be used to guide the implementation of the FMM algorithm. It should be emphasized that the near-field parameter Δ does not affect the $O(N^{1.5})$ complexity of the FMM algorithm. Thus, choosing a larger Δ for large scale simulation is advantageous since it yields more accurate results with little compromise in speed.

4 Acknowledgements

We would like to acknowledge Mr. Michael Carr for his help in carrying out the parallelization and some of the computations mentioned in this paper. The computing resources were provided by the ASC MSRC High Performance Computing, Dayton, OH. and National Partnership for Advanced Computing Infrastructure (NPACI).

FIGURE CAPTIONS

- Fig. 1** Biquadratic quadrilateral surface element.
- Fig. 2** Test target geometries: (a) Sphere, (b) Ogive, and (c) Strip.
- Fig. 3** Test simulation setup, depicted for the ogive.
- Fig. 4** Error in the scattered field as a function of the number of multipoles used for 1λ radius sphere for $\Delta = 1.5$ (2476 unknowns at 0.3 GHz).
- Fig. 5** Error in the scattered field as a function of the number of multipoles used for 1λ radius sphere for $\Delta = 1.5$ (2476 unknowns at 0.3 GHz).
- Fig. 6** Error in the scattered field as a function of the number of multipoles used for the 20λ long, 0.5λ wide strip for $\Delta = 1.5$ and $\Delta = 3.5$ (1795 unknowns at 0.3 GHz).
- Fig. 7** Error in the scattered field as a function of the near-field threshold for the sphere for two different values of L .
- Fig. 8** Error in the scattered field as a function of the near-field threshold for the ogive for three different values of L .
- Fig. 9** Error in the scattered field as a function of the near-field threshold for the strip for four different values of L .
- Fig. 10** Error in the scattered field as a function of the near-field threshold for the strip geometry for broadside incidence and for $L = 6$.

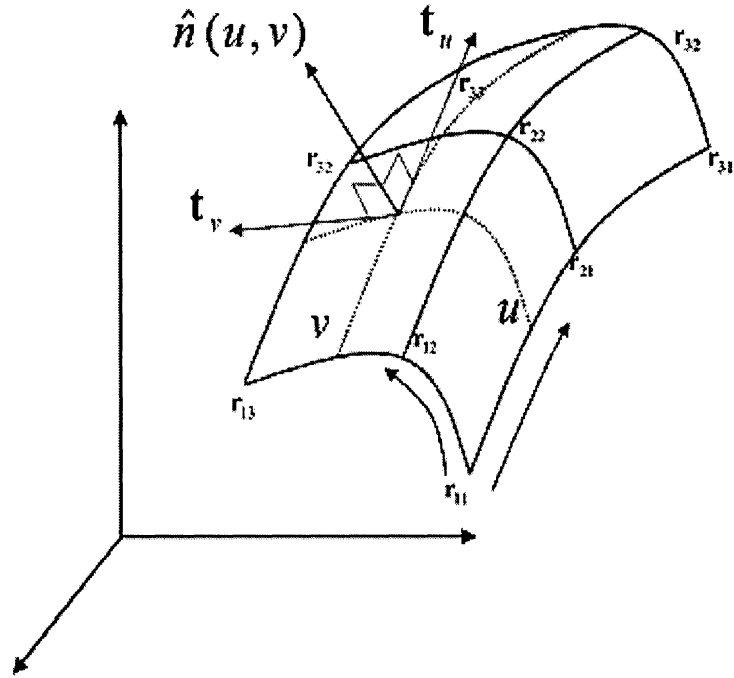


Figure 1: Biquadratic quadrilateral surface element.

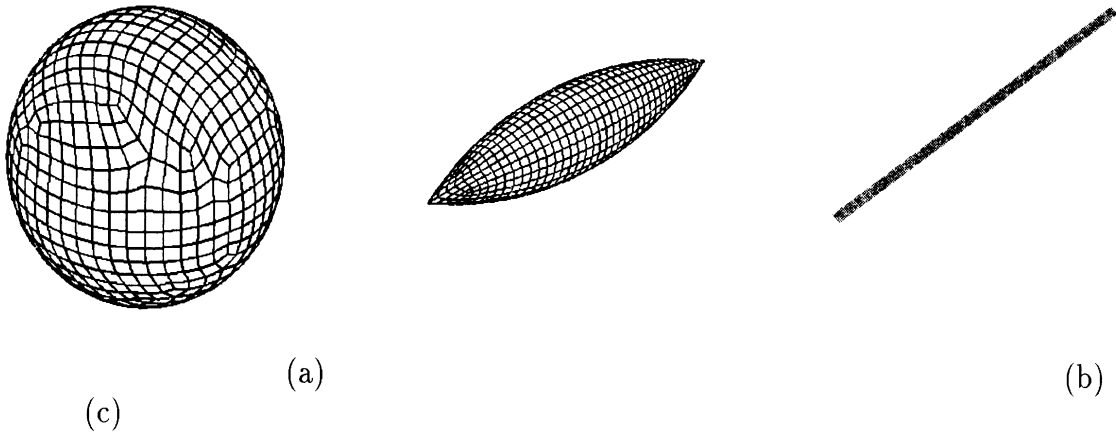


Figure 2: Test target geometries: (a) Sphere, (b) Ogive, and (c) Strip.

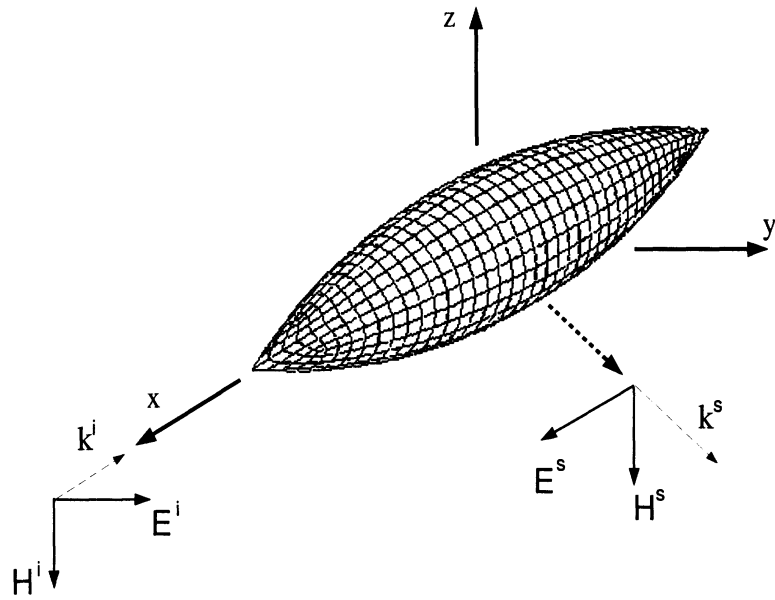


Figure 3: Test simulation setup, depicted for the ogive.

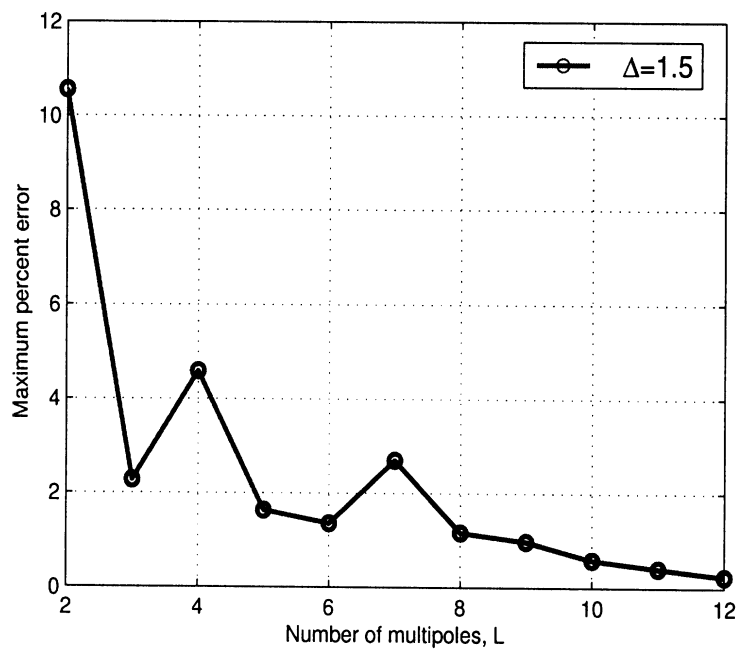


Figure 4: Error in the scattered field as a function of the number of multipoles used for 1λ radius sphere for $\Delta = 1.5$ (2476 unknowns at 0.3 GHz).

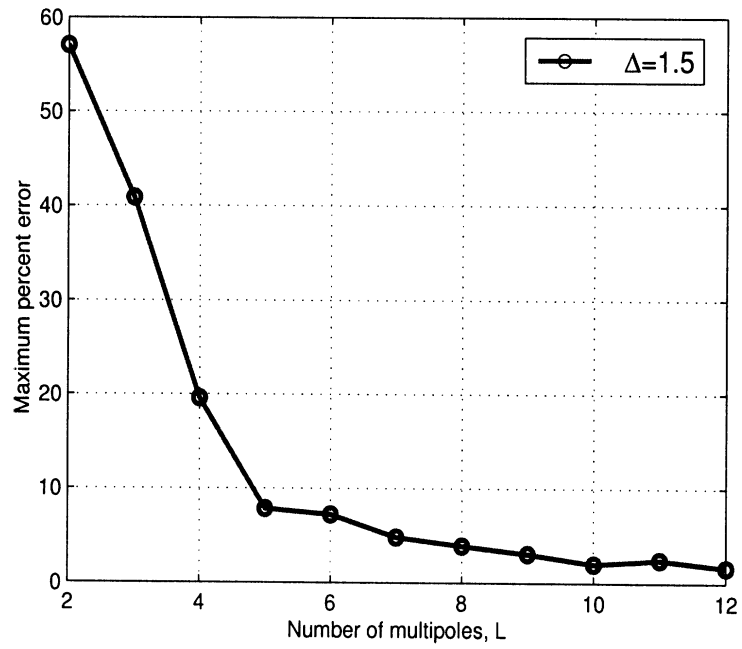


Figure 5: Error in the scattered field as a function of the number of multipoles used for the 5λ long, 1λ wide ogive for $\Delta = 1.5$ (480 unknowns at 0.15 GHz).

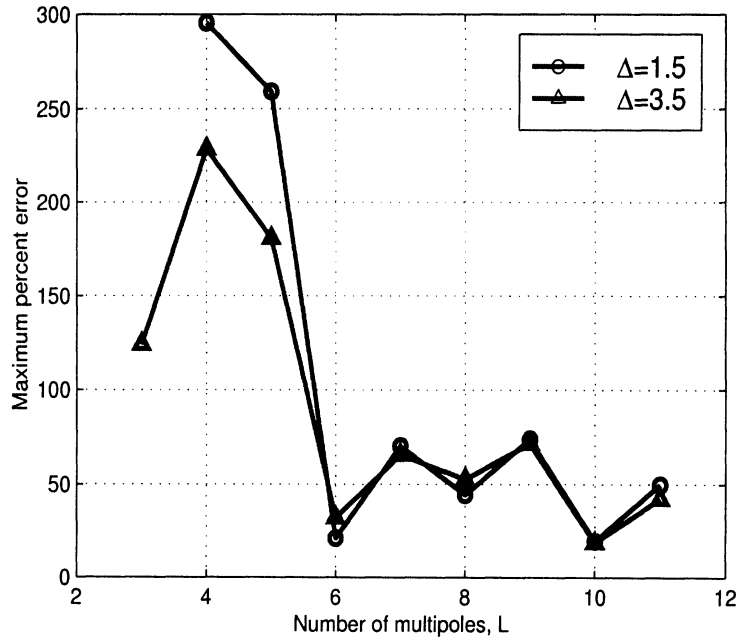


Figure 6: Error in the scattered field as a function of the number of multipoles used for the 20λ long, 0.5λ wide strip for $\Delta = 1.5$ and $\Delta = 3.5$ (1795 unknowns at 0.3 GHz).

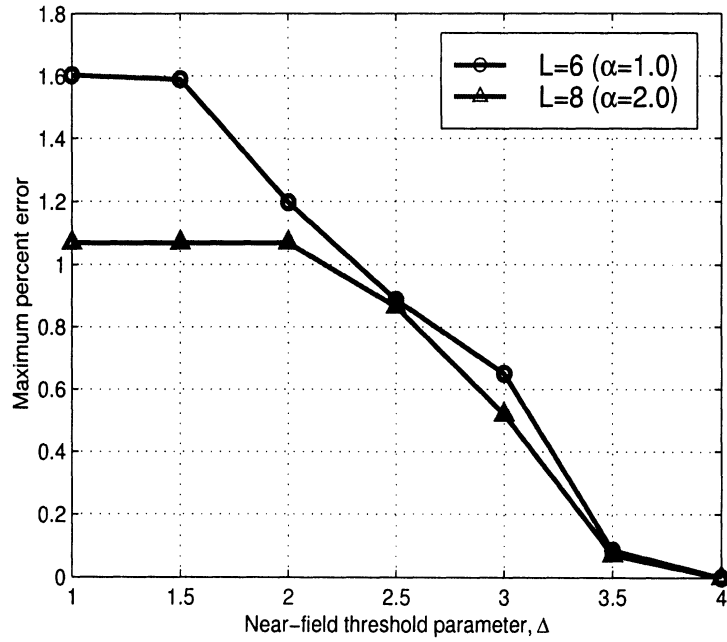


Figure 7: Error in the scattered field as a function of the near-field threshold for the sphere for two different values of L .

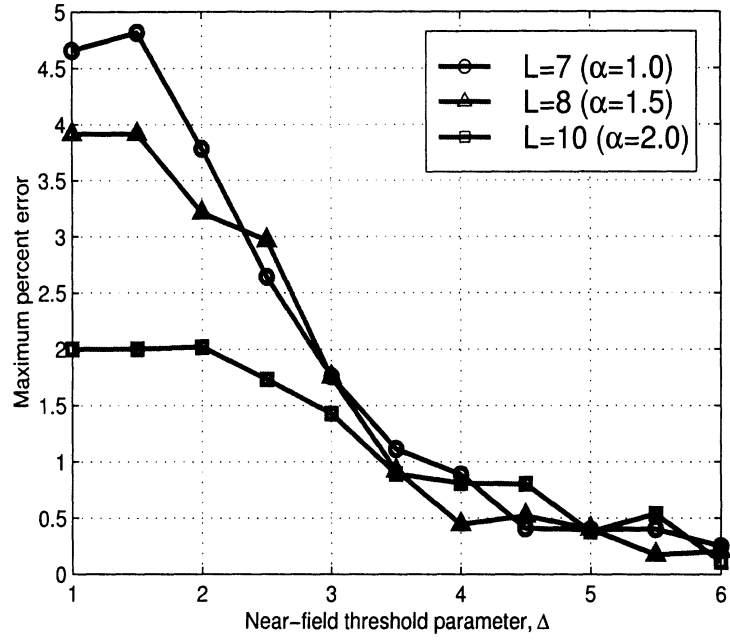


Figure 8: Error in the scattered field as a function of the near-field threshold for the ogive for three different values of L .

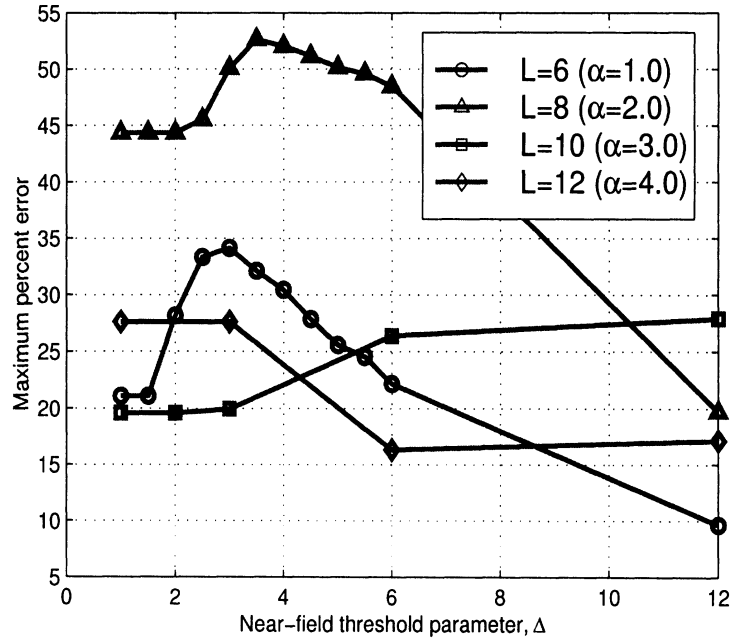


Figure 9: Error in the scattered field as a function of the near-field threshold for the strip for four different values of L .

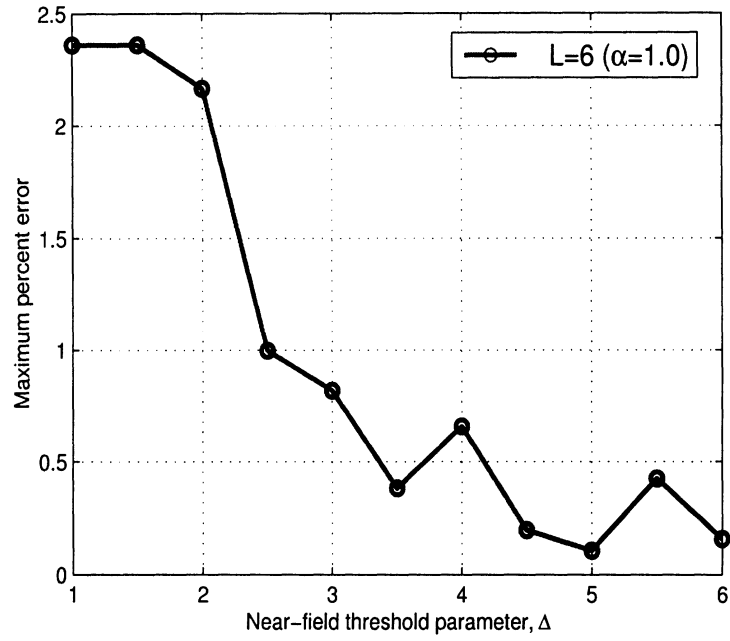


Figure 10: Error in the scattered field as a function of the near-field threshold for the strip geometry for broadside incidence and for $L = 6$.

References

- [1] Coifman, R., V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription," *IEEE Antennas and Propagat. Mag.*, vol. 35, pp. 7–12, Jun. 1993.
- [2] Chew, W. C., J. Jin, C. Lu, E. Michielssen, and J. M. Song, "Fast Solution Methods in Electromagnetics," *IEEE Trans. Antennas Propagat.*, vol. AP-45, no. 3, pp. 533–543, Mar. 1997.
- [3] Bleszynski, E., M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive Integral Method for Solving Large-Scale Electromagnetic Scattering and Radiation Problems," *Radio Sci.*, vol. 31, no. 5, pp. 1225–1251, Sep./Oct. 1996.
- [4] Anastassiou, H. T., M. Smelyanskiy, S. S. Bindiganavale, and J. L. Volakis, "Scattering from Relatively Flat Surfaces Using the Adaptive Integral Method," *Radio Sci.*, vol. 33, no. 1, pp. 7–16, 1998.
- [5] Bindiganavale, S. S. and J. L. Volakis, "Scattering and Radiation from Planar Structures Containing Small Features Using a Finite Element Fast Integral Method," *Electron. Lett.*, vol. 34, no. 21, Oct. 15, 1998.
- [6] Eibert, T. F. and J. L. Volakis, "Adaptive Integral Method for Hybrid FE/BI Modeling of 3D Doubly Periodic Structures," *IEE Proc. Microwaves, Antennas and Propagation*, vol. 146, pp. 17–22, Feb. 1999.
- [7] Lu, N. and J.-M. Jin, "Application of the Fast Multipole Method to Finite-Element Boundary-Integral Solution of Scattering Problems," *IEEE Trans. Antennas Propagat.*, vol. AP-44, no. 6, pp. 781–786, 1996.
- [8] Volakis, J. L., T. Özdemir, and J. Gong, "Hybrid Finite Element Methodologies for Antennas and Scattering," *IEEE Trans. Antennas Propagat.*, pp. 493–507, Mar. 1997.
- [9] Volakis, J. L., A. Chatterjee, and L. C. Kempel, *Finite Element Methods for Electromagnetics*, Piscataway, NJ:IEEE Press, 1998.

- [10] Eibert, T. F., J. L. Volakis, D. R. Wilton, and D. R. Jackson, "Hybrid FE/BI Modeling of 3D Doubly Periodic Structures Utilizing Triangular Prismatic Elements and a MPIE Formulation Accelerated by the Ewald Transformation," *IEEE Trans. Antennas Propagat.*, accepted for publication, 1998.
- [11] Burkholder, R. J. and D. H. Kwon, "High frequency asymptotic acceleration of the fast multipole method," *Radio Sci.*, vol. 31, pp. 1199-1206, 1996.
- [12] Song, J. M., C. C. Lu, and W. C. Chew, "Multilevel Fast Multipole Algorithm for Electromagnetic Scattering by Large Complex Objects," *IEEE Trans. Antennas Propagat.*, vol. AP-45, no. 10, pp. 1488-1493, 1997.
- [13] Bindiganavale, S. S. and J. L. Volakis, "Comparison of Three FMM Techniques for Solving Hybrid FE/BI Systems," *IEEE Trans. Antennas Propagat.*, vol. AP-39, no. 4, pp. 47-59, Aug. 1997.
- [14] R. L. Wagner and C. W. Chew, "A Ray-Propagation Fast Multipole Algorithm," *Microwave Opt. Tech. Lett.*, vol. 7, no. 7, pp. 435-438, Jul. 1994.
- [15] Lu, N. and W. C. Chew, "Fast Far-Field Approximation for Calculating the RCS of Large Objects," *Microwave Opt. Tech. Lett.*, vol. 8, no. 5, pp. 238-241, 1995.
- [16] Song, J. M. and W. C. Chew, "Multilevel Fast Multipole Algorithm for Solving Combined Field Integral Equations of Electromagnetic Scattering," *Microwave Opt. Tech. Lett.*, vol. 10, no. 1, pp. 14-19, Sep. 1995.
- [17] Song, J. M. and W. C. Chew, "Fast Multipole Method Solution Using Parametric Geometry," *Microwave Opt. Tech. Lett.*, vol. 7, no. 16, pp. 760-765, Nov. 1994.
- [18] G. E. Antilla and N. G. Alexopoulos, "Scattering from Complex 3D Geometries by a Curvilinear Hybrid Finite Element-Integral Equation Approach", *J. Opt. Soc. Am. A*, Vol. 11, No. 4, pp. 1445-1457, 1994

- [19] Rao, S. M., D. R. Wilton, and A. W. Glisson, "Electromagnetic Scattering by Surfaces of Arbitrary Shape," *IEEE Trans. Antennas Propagat.*, vol. AP-30, no. 3, pp. 409–418, May 1982.
- [20] Griffiths, P. and Hill, I. D., *Applied Statistics Algorithms*, West Sussex, England, Ellis Horwood, 1985

ERROR EVALUATION OF FAST INTEGRAL METHODS FOR RCS COMPUTATIONS

K. Sertel*, M. Carr and J. L. Volakis

Radiation Laboratory

Department of Electrical Engineering and Computer Science

University of Michigan

Ann Arbor, MI 48109-2212, USA

Email: ksertel,mcarr,volakis@eecs.umich.edu

INTRODUCTION

The fast multipole method (FMM) [1] and the adaptive integral method (AIM) [2] has been successfully applied to speed-up moment method (MoM) solutions of scattering and radiation problems. Both the FMM and the AIM implementations attain their speed-up through a number of approximations. However these approximations lead to errors which must be understood and controlled for its reliable utilization. Previous studies assessed FMM errors by examining the approximation of the pertinent Green's function used for evaluating the MoM matrix elements. Practice, however, dictates that the error between FMM-MoM and uncompressed MoM solutions also depends on the excitation, the specific structure under analysis, and the quantity of interest.

In this paper, we consider the effect of FMM and AIM parameters on RCS computations for various targets. Namely, our study focuses on the computation of the observable parameter (the RCS for our case) and not just on the matrix elements.

FMM PARAMETERS

This section outlines the key step in FMM formulations. The FMM parameters under investigation are also defined in this section. The FMM implementation used herewith was built upon that in [3].

The CPU time per iteration/matrix vector product of FMM is $O(CN^{1.5})$ instead of $O(N^2)$ [1]. However, choices of the FMM parameters to achieve a faster implementation (i.e. a smaller C) inevitably lead to less accurate answers.

A most important parameter in the FMM implementation is the number of terms kept for the evaluation of the translation operator T_L , commonly denoted as L [1]. Clearly, keeping L as small as possible reduces the CPU time but not the order $CN^{1.5}$ for the matrix vector product computation. Small L simply leads to smaller value for C and consequently, the CPU time crossover point between standard LU and FMM implementations occurs for lower values of N .

Another key aspect of the FMM is element grouping. The associated element groups are called clusters and a grouping algorithm must be used. Here, we used the k -means algorithm [4] to form the clusters. This algorithm aims to minimize a defined error function for the purpose of achieving a fairly uniform set of groups. The specific error function used by the k -means algorithm minimizes the sum of the distances between the cluster centers and the centers of the basis functions in each cluster. If the resulting clustering is not uniform, the FMM implementation will be inefficient due to the presence of very large or very small clusters. This effects the accuracy of the spectral integral since the number of terms used in the translation sum will depend on the size of the largest cluster.

In the FMM implementation, the indirect evaluation of the matrix-vector product is only used for computing the interactions of elements between clusters. Within a simple cluster, the elements are interacted without the FMM approximation. For greater accuracy, elements between adjacent or distant clusters can be computed without approximation as well. Here, we introduce an additional parameter Δ for controlling the transition

*K. Sertel was funded by NPACI. The computer code used for this work, FMM-SWITCH, was based on the SWITCH code originally developed at Northrop Grumman Corp.

distance between the exact and the FMM approximation. Specifically, if d_{ij} is the distance between the i th and j th clusters, the FMM approximation was involved only if $d_{ij} > \Delta(r_i + r_j)$ and $kd_{ij} > L$ where $r_{i,j}$ denote the radii of the i th and j th clusters.

AIM PARAMETERS

The Adaptive Integral Method (AIM) achieves memory and complexity of $O(N \log N)$ through the usage of the FFT to accelerate the matrix vector product [2]. The AIM implementation used here is based on the MoM formulation given in [5].

The impedance matrix terms are divided into Z_{near} and Z_{far} matrices using a threshold parameter. For each pair of edges, the distance between the midpoints of the edges is measured. If this distance is greater than the threshold, the pair is included in Z_{far} matrix. Otherwise, they are included in Z_{near} . In this paper, we investigate the effects of the Z_{near} threshold upon the error in RCS for AIM computations.

In AIM, the original basis functions are remapped onto a regularly spaced grid. There is no requirement that the grid must have the same spacing in the x-, y-, and z-direction, but it is normally taken to be the same, and a nominal value for the grid spacing is 0.05λ .

ERROR STUDY

Three representative geometries of different geometric and electromagnetic properties were investigated for a more complete and unbiased study. The first geometry, depicted in Fig. 1 (a) is a PEC sphere. For our study the standard MoM solution is used to benchmark the FMM-MoM approximation. The second geometry considered is an ogive (see Fig. 1 (b)). Unlike the sphere, it has two sharp tips that cause strong scattering contributions near edge-on incidences. The third geometry analyzed is an elongated strip. This strip is simply a flat plate having an aspect ratio of 40/1 as depicted in Fig. 1 (c). This target is chosen because it consists of a localized geometry and has sharp edges which cause substantial diffraction.

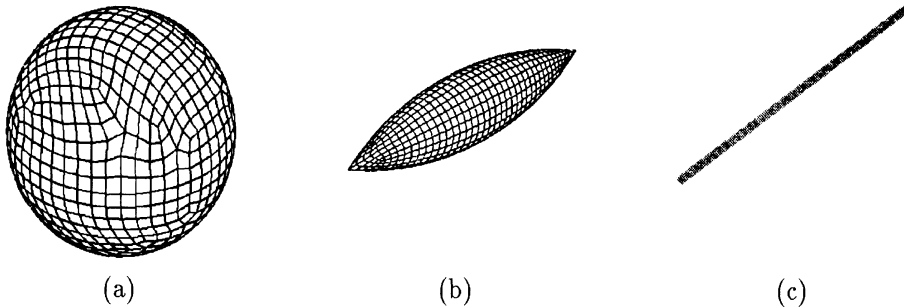


Figure 1: Test target geometries: (a) Sphere, (b) Ogive, and (c) Strip.

Error Definition

For each of these three geometries, tip-on incidence is assumed with horizontal polarization. We choose this incidence because it is the more stressing case for scattering computations. The scattered far-field is sampled on 332 points on a half sphere covering the entire $z < 0$ space. As noted earlier, the error curves were generated using the MoM solution as reference. The error values were computed using the l_∞ norm of the error vector at the sample points on the half sphere, and are normalized to the maximum value of the reference solution vector to get the percent error.

Number of Multipoles, L

Beginning with the first target geometry, Fig. 2 (a) shows the FMM solution error for a 1 m radius sphere at 0.3 GHz. The discretization resulted in 2746 unknowns. We observe clearly that the error decreases as more multipole terms are and this was to be expected. Some oscillations are observed (i.e. convergence is not monotonic) and these are likely to be due to the asymptotic nature of the multipole expansion. It should be noted that the grouping and the discretization are fairly uniform and thus these oscillations are not due to

clustering variations. For this example the near-field threshold was set to $\Delta = 1.5$. The FMM solution error curves for the ogive are given in Fig. 2 (b). Its width was 2 m (1 wavelength), i.e. the frequency was 150 MHz. The discretization resulted in 480 unknowns. Similar to the sphere results, the error is decreasing as a function of L and some oscillations in the error curve are again observed. These oscillations are more pronounced for the strip geometry for which the error curve is given in Fig. 2 (c). The frequency of the incident plane wave was 300 MHz, the strip was 20 m (20 wavelengths) long and 0.5 m (0.5 wavelengths) wide, and the discretization resulted in 1795 unknowns. Clearly, the most pronounced difference among the error curves in Figs. 2 (a) –(c) is the error for the error for the strip. This is possibly due to the stronger multiple diffraction mechanisms characterizing its RCS at edge-on incidence.

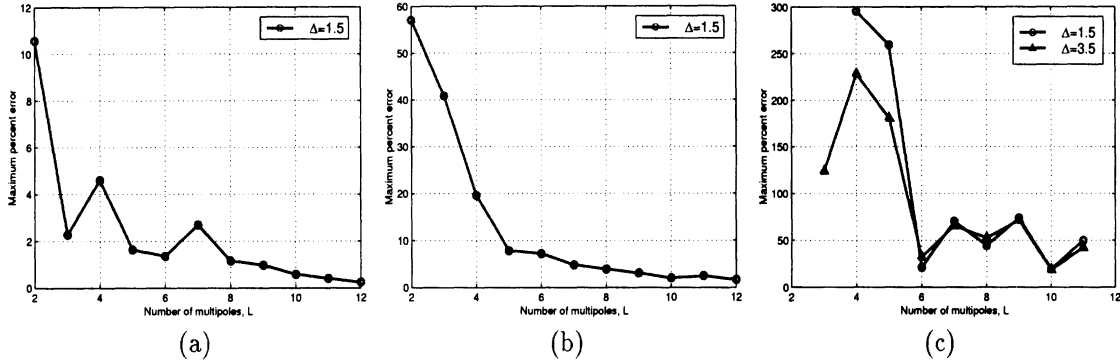


Figure 2: Error in the scattered field as a function of the number of multipoles for: (a) Sphere (b) Ogive (c) Strip

Near-field Threshold, Δ

Keeping L small is certainly advantageous to attain maximum speed-up by using the least memory for the FMM implementation. However, speed and accuracy are also dependent on the near-field threshold. Thus, it is essential to look at the overall RCS accuracy when L and Δ are both varied.

Fig. 3 (a) shows the effect of the near field threshold parameter Δ on the accuracy of the computed bistatic RCS for a sphere. It is important to note here that regardless of the choice of L , the curves nearly overlay for $\Delta > 3.5$. This is also observed for the ogive geometry as shown in Fig. 3 (b). When the strip geometry is considered we could no longer observe a consistent error behavior as a function of the near field threshold parameter for edge on incidence (Fig. 3 (c)). This is due to the very low level of the RCS and this observation is in contrast to earlier comments that the parameters can be chosen regardless of geometry and RCS level to attain given error criterion.

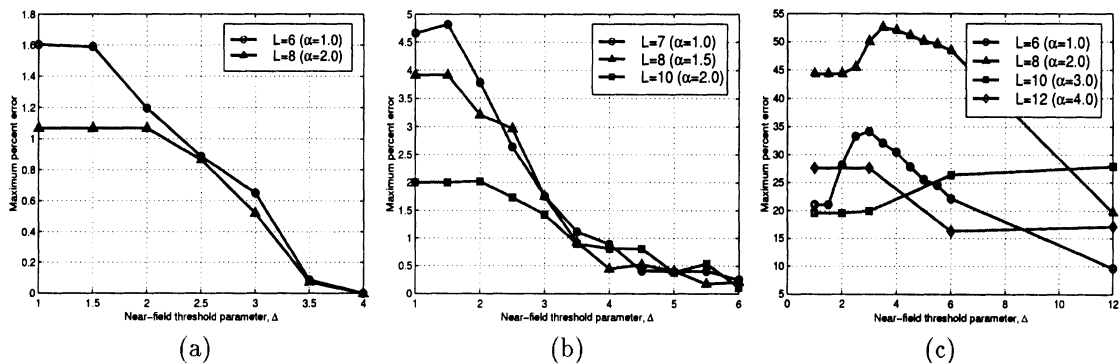


Figure 3: Error in the scattered field as a function of the FMM near-field threshold for: (a) Sphere (b) Ogive (c) Strip

In Fig. 4 we plot the AIM error analysis results for the same strip geometry. The strip geometry, being planar, is very well suited for AIM application. As observed, the error in RCS drops dramatically for a threshold greater than 0.2λ . We note that this AIM error analysis may be highly dependent upon the geometry and the direction

of incidence and should not be generalized. Nonetheless, a threshold of 0.2λ for near field AIM matrix is a good starting point. We are currently investigating the error behavior for the other two geometries.

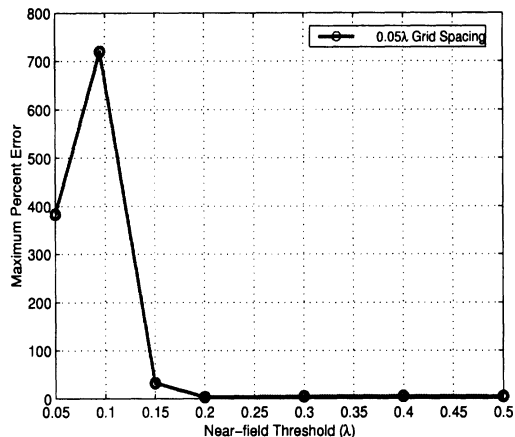


Figure 4: Error in the scattered field as a function of the AIM near-field threshold parameter.

General Observations

In contrast to earlier observations it was demonstrated that care should be taken when choosing FMM parameters for fast MoM implementations and that the parameter choices are geometry dependent for a given error criterion. This is expected since the matrix condition is also geometry dependent. To summarize:

- Error of the FMM solution depends on the target geometry under investigation (equivalently on the matrix condition).
- Error also depends on the excitation (incidence direction and polarization).
- Overall RCS error behavior does not correlate with error analysis based solely on the expansion of the Green's function. In that case the geometry and excitation do not enter into the analysis.
- For the examined three representative geometries, it was shown that choosing the near-field threshold parameter as $\Delta \approx 3.5$ produces optimum error performance (almost independent of the number of multipoles used for $L > kD_{max} + \ln(kD_{max} + \pi)$). This increases the near field computations, hence the optimization of the FMM algorithm should be done accordingly.
- For the strip geometry, for which the FMM performs poorly, the AIM algorithm is easy to implement and provides very small RCS error when the near field threshold is chosen to be $\geq 0.2\lambda$

References

- [1] Coifman, R., V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription," *IEEE Antennas and Propagat. Mag.*, vol. 35, pp. 7–12, Jun. 1993.
- [2] Bleszynski, E., M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive Integral Method for Solving Large-Scale Electromagnetic Scattering and Radiation Problems," *Radio Sci.*, vol. 31, no. 5, pp. 1225–1251, Sep./Oct. 1996.
- [3] G. E. Antilla and N. G. Alexopoulos, "Scattering from Complex 3D Geometries by a Curvilinear Hybrid Finite Element-Integral Equation Approach", *J. Opt. Soc. Am. A*, Vol. 11, No. 4, pp. 1445-1457, 1994
- [4] Griffiths, P. and Hill, I. D., *Applied Statistics Algorithms*, West Sussex, England, Ellis Horwood, 1985
- [5] Rao, S. M., D. R. Wilton, and A. W. Glisson, "Electromagnetic Scattering by Surfaces of Arbitrary Shape," *IEEE Trans. Antennas Propagat.*, vol. AP-30, no. 3, pp. 409–418, May 1982.

The Effect of FMM Parameters on the Solution of PEC Scattering Problems

K. Sertel* and J. L. Volakis

Radiation Laboratory

Department of Electrical Engineering and Computer Science

The University of Michigan

Ann Arbor, MI 48109-2122 *

1 Introduction

The Method of Moments (MoM) has been successfully applied for the solution of electromagnetic scattering and radiation problems as the main full-wave analysis method. Due to the convolutional nature of the Green's functions governing these problems, conventional MoM requires the solution of a fully populated interaction matrix. The resulting system needs to be solved by either LU decomposition or with the aid of an iterative solution algorithm such as conjugate gradient method (CG). This aspect limits the electrical size of the problems that can be solved on a given computer with limited memory. Recently proposed fast methods such as the adaptive integral method (AIM), fast multipole method (FMM), and its variations promise to provide the solution of the problem using less computer memory and less computation time. In this paper, the FMM has been analyzed on the basis of two selected PEC scatterers in terms of its parameters to be mentioned below.

2 MoM and FMM Formulations for PEC Scattering Problems

The electric field integral equation (EFIE) for electromagnetic scattering problems involving open or closed conducting surfaces is given by

$$\hat{t} \cdot \int_s \left[\mathbf{J}(\mathbf{r}') + \frac{1}{k^2} \nabla' \cdot \mathbf{J}(\mathbf{r}') \nabla \right] \frac{e^{ikR}}{R} ds' = \frac{4\pi i}{k\eta} \hat{t} \cdot \mathbf{E}^i(\mathbf{r}), \quad (1)$$

where $e^{i\omega t}$ time convention is adopted and $R = |\mathbf{r} - \mathbf{r}'|$.

The EFIE for the unknown electric current density $\mathbf{J}(\mathbf{r})$ on the conducting surface induced by an incident wave is discretized using the MoM technique. The induced surface current is approximated by a sum of N known basis functions $\{\mathbf{j}_n(\mathbf{r})\}$ and the EFIE is converted into a linear system of equations in the unknown coefficients of the current expansion using Galerkin's testing and the resulting matrix elements and excitation vector elements are given by [1]

$$Z_{ij} = \int_s ds \mathbf{t}_i(\mathbf{r}) \cdot \int_{s'} ds' \left[\mathbf{j}_j(\mathbf{r}') + \frac{1}{k^2} \nabla' \cdot \mathbf{j}_j(\mathbf{r}') \nabla \right] \frac{e^{ikR}}{R} \quad (2)$$

and

$$F_i = \frac{4\pi i}{k\eta} \int_s ds \mathbf{t}_i(\mathbf{r}) \cdot \mathbf{E}^i(\mathbf{r}). \quad (3)$$

*Thanks go to Northrop-Grumman

The basis functions used in this work are the curved rooftop basis functions (CRT) defined on the biquadratic surface elements forming the mesh of the scatterer are slightly different versions of those presented in [1].

Direct application of the MoM requires the computation of N^2 double surface integrals appearing in Eq. (2) as the elements of the resulting MoM matrix. Solution of this system of equations by Gaussian elimination requires $O(N^3)$ operations. Iterative solvers require $O(N^2)$ operations per iteration. The memory requirement of the MoM is also $O(N^2)$. This large order for storage limits the size of the problem that can be solved on a given hardware, and the high operation cost poses a limit to the size of problems that can be solved in a practically acceptable period of time. For these reasons, the FMM is proposed [2, 3, 4], which requires less memory and CPU time for the solution of large problems.

2.1 Fast Multipole Method

The fast multipole method (FMM) is based on a spherical multipole and subsequent plane wave expansion of the free-space Green's function appearing in Eq (2) as the kernel in the matrix elements of the discretized EFIE. The resulting representation of the BI matrix elements can be written as [3, 4, 5]

$$Z_{ij} \approx \frac{ik}{4\pi} \int d^2\hat{k} \mathbf{V}_{fmi}(\hat{k}) \cdot T_L(kr_{mm'}, \hat{k} \cdot \hat{r}_{mm'}) \mathbf{V}_{sm'j}^*(\hat{k}), \quad (4)$$

with

$$\begin{aligned} \mathbf{V}_{sm'j}(\hat{k}) &= \int_S ds' e^{i\mathbf{k} \cdot \mathbf{r}_{jm'}} [\bar{\mathbf{I}} - \hat{k}\hat{k}] \cdot \mathbf{j}_j(\mathbf{r}_j), \\ \mathbf{V}_{fmi}(\hat{k}) &= \int_S ds e^{i\mathbf{k} \cdot \mathbf{r}_{im}} [\bar{\mathbf{I}} - \hat{k}\hat{k}] \cdot \mathbf{t}_i(\mathbf{r}_i) \end{aligned} \quad (5)$$

and

$$T_L(kr, \hat{k} \cdot \hat{r}) = \sum_{l=0}^L i^l (2l+1) h_l^{(1)}(kr) P_l(\hat{k} \cdot \hat{r}). \quad (6)$$

It has been shown [3, 4, 5] that the computational complexity of the FMM matrix-vector product is $\mathcal{O}(NM) + \mathcal{O}(N^2/M)$ where M is the number of basis function clusters. This complexity is minimized by choosing $M = \sqrt{N}$ resulting in $\mathcal{O}(N^{1.5})$ complexity.

An important parameter which affects the accuracy of the FMM implementation is the number of terms kept in the multipole expansion of the free-space Green's function. A semi-empirical formula to formula for the choice of L is given in [3]

$$L = kD_{max} + \alpha \log(kD_{max} + \pi) \quad (7)$$

where k is the wavenumber and D_{max} is the maximum physical size of the clusters and α is a parameter affecting the accuracy of the approximation. Since large values of L correspond to large CPU requirements for the FMM, it is desirable to keep L as small as possible. Clearly for small L the numerical errors in evaluating the translation operators between the clusters may be unacceptable. The near-field matrix involves the interactions of elements within a single cluster and additional element-to-element interactions which belong in different nearby clusters satisfying the criteria

$$d_{ij} < 1.5(r_i + r_j) \quad (8)$$

or

$$kd_{ij} < L \quad (9)$$

where d_{ij} is the separation between the clusters i and j and r_i, r_j denote the radii of the clusters. It should be noted that although keeping L small decreases the FMM computations in the matrix-vector product, this also results in increased near-field matrix computations and storage as dictated by (9). Hence the designed FMM solver should be properly optimized to achieve minimum computational complexity within tolerable solution accuracy.

3 Results and Comparisons

Two representative geometries are studied using the FMM code developed based on the given formulation. The effect of the number of multipoles kept in the FMM expansion on the computed RCS signatures of the scatterers is investigated based on these geometries. The scatterers are illuminated by plane wave and the bistatic and monostatic radar cross section (RCS) of the scatterers are observed. The first geometry is a PEC sphere of radius $2'$. The effect of L on the bistatic RCS of the sphere is plotted in Fig 1 (a). Choosing $\alpha = 1.0$, which in turn leads to an $L = 8$ provides a good agreement in the RCS data. Nevertheless this good agreement can not be used to conclude that for RCS computations an $\alpha = 1.0$ provides reasonable accuracy as demonstrated by the next example.

The second geometry examined is the PEC ogive geometry depicted in Fig 1 (b). The ogive is also illuminated by a plane wave incident on one of the tips. Due to its sharp tips, the biquadratic surface mesh of the ogive is irregular around these tips. Since the current is most significant around the tips, this fact leads to an ill conditioned system. Also the existence of traveling surface waves around the ogive deteriorates the condition of the resulting matrix. The bistatic RCS of the ogive is also plotted in Fig 1 (b). Clearly for a scatterer with strong scattering centers, the choice $\alpha = 1.0$ does not provide an accurate FMM solution. The parameter L should hence be chosen to be larger than that value governed by Eq (7) for an accurate FMM solution. Fig 2 (a) depicts the monostatic RCS of the ogive at 9.0GHz for different values of the parameter L . The problem size for this geometry is 1776. The effect of numerical error in the multipole expansion is clearly observed for low values of the RCS. This effect is more pronounced at lower frequencies as observed in Fig 2 (b) which shows the monostatic RCS of the ogive at 5.9GHz. The resulting size of the matrix is 480.

For scatterers with strong scattering centers like the tips of the ogive and scatterers represented by ill-conditioned surface meshes, the resulting system is also ill-conditioned. In the iterative solution of such systems preconditioning improves the convergence. A good preconditioner should bear in it most of the information that the matrix itself carries. In FMM formulations the near-field matrix provides an excellent preconditioner. In numerical practice using the near-field matrix provided a speed-up in convergence larger than a factor of 10 in the iterative solution. To avoid the memory requirement of the preconditioner, a less dense matrix formed by keeping only a percentage of the elements of the near-field matrix that are large in absolute value can be used. This provides 5-10 times less number of iterations in the matrix solution. It should also be noticed that the iterative solution time is increased by the preconditioner solution step added to the solution algorithm hence the number of non-zeros in the LU decomposed preconditioner affect the solution time. This is observed in the timings below, the LU decomposition time providing an indicator to the number of non-zeros in the LU decomposed preconditioner. In numerical practice the order of the LU decomposition of the near-field matrix was observed to be less than $\mathcal{O}(N_{nz}^{1.5})$ where N_{nz} is the number of non-zeros in the preconditioner. Remembering that $N_{nz} = \mathcal{O}(N^{1.5})$, the computational complexity of the LU decomposition of the near-field matrix is less than $\mathcal{O}(N^{2.25})$, and since the iterative solvers are of $\mathcal{O}(N)$, the total complexity of the FMM solution, which is $\mathcal{O}(N^{2.5})$, is not

increased by utilizing the near-field matrix as a preconditioner.

The timings for the 1776 unknown ogive problem the following data is recorded on a SUN-Ultra workstation:

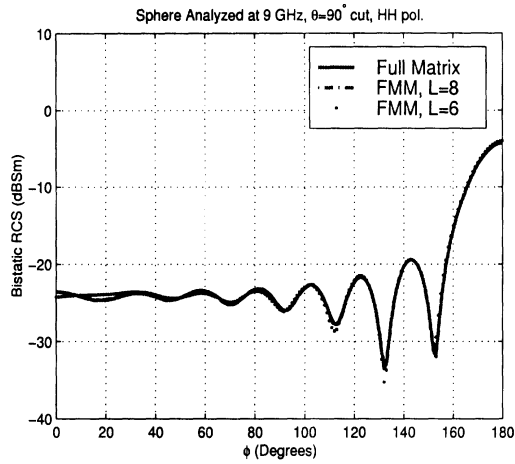
L	Matrix Fill Time(s)	Precond. LU Time(s)	Number of Iter.	Total Solution Time(s)	Time per Iteration(s)
Full	1258	510	24	29	1.18
10	462	591	34	81	2.37
8	485	473	25	36	1.42
7	401	522	19	30	1.57

Table 1: Timings obtained for the ogive geometry at 9.0GHz.

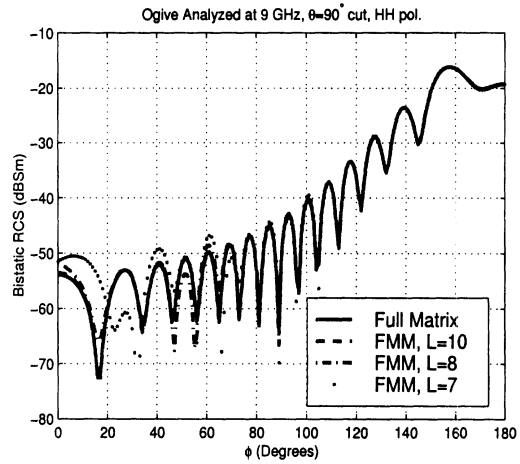
To conclude, in employing the FMM care should be taken for the choice of number of multipoles in problems involving scatterer geometries with strong scattering centers. For such scatterers that generate ill-conditioned systems of equations, preconditioning, using the near-field matrix or a part of the near-field matrix, improves the solution time without sacrificing the $O(N^{1.5})$ complexity of the FMM.

References

- [1] J. M. Song and W. C. Chew, "Moment Method Solution Using Parametric Geometry", Journal of Electromagnetic Waves and Applications, Vol. 9, No. 1/2, pp. 71-83, 1995
- [2] N. Engheta, W. D. Murphy, V. Rokhlin, and M. S. Vassiliou, "The Fast Multipole Method (FMM) for Electromagnetic Scattering Problems", IEEE Trans. on Antennas and Propagation, Vol. 40, No. 6, pp. 634-641, June 1992
- [3] R. Coifman, V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription", IEEE Antennas and Propagation Magazine, Vol. 35, pp. 7-12, June 1993
- [4] J. M. Song and W. C. Chew, "Fast Multipole Method Solution Using Parametric Geometry", Microwave and Optical Technology Letters, Vol. 7, No. 16, pp. 760-765, November 1994
- [5] S. S. Bindiganavale and J.L. Volakis, "Comparison of Three FMM Techniques for Solving Hybrid FE-BI Systems", IEEE Antennas and Propagation Magazine, Vol. 39, No. 4, pp. 47-59, August 1993
- [6] W. C. Chew, J. Jin, C. Lu, E. Michielssen, and J. M. Song, "Fast Solution Methods in Electromagnetics", IEEE Trans. Antenna Propagat., Vol. 45, No. 3, pp. 533-543, March 1997

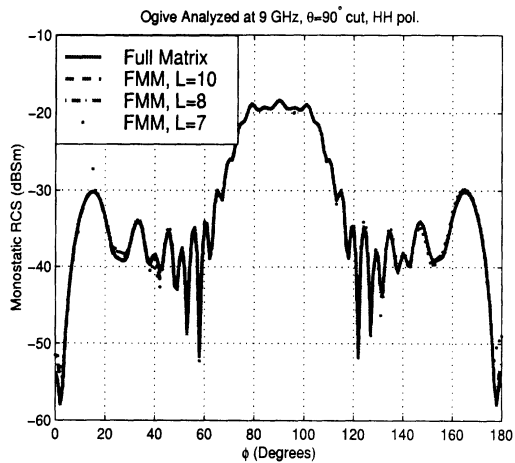


(a)

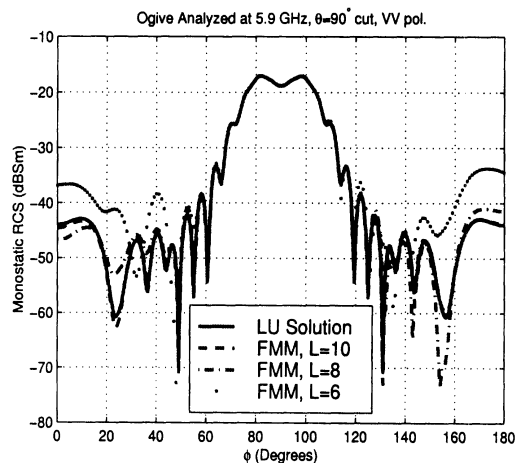


(b)

Figure 1: Effect of number of multipoles on the bistatic RCS of (a) Sphere at 9.0 GHz, HH pol., (b) Ogive at 9.0 GHz, HH pol.



(a)



(b)

Figure 2: Effect of number of multipoles on the monostatic RCS of the ogive, (a) HH polarization, at 9.0 GHz, (b) VV polarization, at 5.9 GHz.

Incomplete LU preconditioner for FMM implementation

Kubilay Sertel and John L. Volakis
Radiation Laboratory, EECS Department, University of Michigan
Ann Arbor, MI 48109-2122, USA
E-mail : {ksertel, volakis}@umich.edu

Abstract

Incomplete LU (ILU) preconditioner using the near-field matrix of the fast multipole method (FMM) is investigated to increase the efficiency of the iterative conjugate gradient squared (CGS) solver. Unlike the conventional LU, ILU requires no fill-ins hence no extra memory and CPU time in computing the LU decomposed preconditioner. It is shown that due to the nature of the near-field matrix, ILU preconditioning decreases the number of iterations dramatically.

Introduction

The FMM has been proposed and successfully applied to increase the efficiency of the method of moments (MoM) solution of large scale electromagnetic scattering and radiation problems [1, 2, 4, 5, 6, 7, 8, 9, 10, 11]. The FMM is based on the iterative solution of the MoM matrix equation using an indirect fast computation of the matrix vector product. The convergence behavior is strongly dependent on the condition of the constructed matrix as well as the type of the iterative solver used. Various options to choose from exist for the iterative solver, such as the conjugate gradient (CG), bi-conjugate gradient (BiCG), conjugate gradient squared (CGS), quasi minimal residual (QMR), and generalized minimal residual (GMRES). First three belong to the family of conjugate gradient solvers and the last two are in the minimal residual family of iterative solvers.

The convergence of all iterative solvers are dictated by the conditioning of the matrix equation and the condition of the matrix deteriorates as the matrix size increases. For large scale electromagnetics simulations the resulting matrix equation usually has a size larger than 100,000. Iterative solution of such large, fully populated matrix systems inevitably requires some kind of a preconditioning. Otherwise, propagation of the numerical error at each iteration renders the convergence of the solver impossible.

Various preconditioning techniques to improve the condition of the system can be found in [12]. Diagonal and block diagonal preconditioners for multilevel FMM implementations were reported in [6]. An ILU preconditioner for MoM and FMM implementations is presented for a CGS solver. Due to its nature the ILU preconditioner constructed using the near-field FMM matrix significantly reduces the number of iterations to get convergence for a given tolerance.

Method of Moments and the Fast Multipole Method

The MoM formulation of electromagnetic scattering problems using 2^{nd} order curvilinear quadrilateral elements for surface modeling is given in [13]. The FMM implementation used herewith was built upon that in [13].

Briefly, for the electric field integral equation (EFIE) the resulting linear system after Galerkin's testing is of the form

$$\sum_{n=1}^N Z_{mn} a_n = V_m, \quad m = 1, 2, \dots, N \quad (1)$$

where

$$Z_{mn} = \int_s ds \mathbf{f}_m(\mathbf{r}) \cdot \int_{s'} ds' \left[\mathbf{f}_n(\mathbf{r}') + \frac{1}{k^2} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') \nabla \right] \frac{e^{ikR}}{R} \quad (2)$$

and

$$V_m = \frac{4\pi i}{k\eta} \int_s ds \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}^i(\mathbf{r}). \quad (3)$$

An $e^{-i\omega t}$ time convention has been assumed and suppressed and $\{a_n\}$ refers to the column containing unknown coefficients of the surface current expansion

$$\mathbf{J}(\mathbf{r}) \approx \sum_{n=1}^N a_n \mathbf{j}_n(\mathbf{r}). \quad (4)$$

In the above expressions \mathbf{r} and \mathbf{r}' denote the usual observation and source point locations, $\mathbf{E}^i(\mathbf{r})$ is the incident excitation plane wave at \mathbf{r} , \hat{t} is the vector tangent to the surface at \mathbf{r} , $\eta = 120\pi$ denotes the free space impedance, and $k = 2\pi/\lambda$ is the free space wavenumber.

The key aspects of the FMM implementation are

- Iterative solution of the MoM system of equations $[Z]\{a\} = \{V\}$
- Fast evaluation of the matrix vector product $[Z]\{a\}$.

Fast evaluation of the matrix vector product (using $O(N^{1.5})$ or less resources, instead of $O(N^2)$) is attained by introducing an approximation to the pertinent Green's function [1]. It can be shown [1] that if that the CPU time per iteration/matrix vector product for FMM is $O(N^{1.5})$ instead of $O(N^2)$.

The above CPU estimates are asymptotic in the sense that they represent values which are approached for very large N . The actual efficiency of the implementation depends on the choice of various parameters. These choices control the constant in front of the asymptotic behavior of the CPU requirements. However, choices of the FMM parameters for faster implementation inevitably lead to error in the matrix vector product inducing erroneous minima for the iterative solver. This aspect may lead to unacceptable error since it may cause the solver to converge to one of these minimums, especially when non-monotonic solvers (e.g. CG, BiCG, or CGS) are employed.

Along with the EFIE formulation, the same problem can also be formulated in terms of the boundary condition on the magnetic field on the scatterer surface (i.e. MFIE formulation) for closed scatterers [10]. Yet another formulation, the CFIE [10] has the advantage of eliminating the problems due to the internal resonances of the scatterer, and producing much better conditioned systems compared to EFIE or MFIE.

Necessity of Preconditioning

For large scale simulations and/or for geometries with small detailed regions of interest (e.g. antenna arrays on aircraft), the density of the surface mesh cannot be controlled to be a uniform mesh. A nonuniform mesh is well known to produce an ill-conditioned MoM matrix equation. Also, different formulations of the same electromagnetic problem results in different systems with different levels of condition, e.g. the CFIE formulation results in much better conditioned systems than EFIE or MFIE. Also, as noted above, using the FMM implementation for a faster solution may change the matrix condition and introduce erroneous minima to the solution domain, that can render a non-monotonic solver converge to a minimum far from the actual global minimum. It becomes inevitable to improve the convergence behavior of the solver by using a preconditioner.

Although it is very simple to apply the diagonal preconditioner, for systems that are not diagonally dominant diagonal preconditioner does not improve the solution time. To use block diagonal preconditioner, the unknowns need to be numbered such that the larger magnitude matrix elements (i.e. interaction terms of physically close unknowns) are clustered around the diagonal. This can be easily applied to 2 dimensional problems, but finding such a numbering for 3 dimensional problems is not possible for arbitrary geometries. So, with an arbitrary numbering, block diagonal preconditioning cannot guarantee a faster solution. On the other hand, due to the nature of the Green's function in MoM formulations, the near field matrix of FMM is a very good candidate for a preconditioner. The MoM matrix elements with larger magnitudes are kept in the near-field matrix, hence it has a significant portion of the information in the full MoM matrix. The elements of the MoM matrix that are not in the near-field matrix (hence computed using FMM) are smaller in magnitude compared to the near-field matrix elements. It can be concluded that the near-field matrix of FMM is a very attractive preconditioner candidate.

The near-field matrix can directly be LU decomposed to be used as a preconditioner. But, depending on the sparsity pattern of the near-field matrix, this usually requires a significant amount of fill-ins (additions of more elements in the LU decomposed matrix). For large scale simulations, these fill-ins (hence the increase in memory used) becomes a bottleneck in memory utilization. It is also worth noting here that the LU decomposed near-field matrix is a preconditioner and is by no means the exact LU decomposition of the original matrix.

The fill-in requirement of direct LU can be resolved by using the ILU factorization. The ILU algorithm is the same as a direct LU algorithm except it does not fill-in any elements in the decomposed LU matrices, this also results in less CPU utilization.

ILU Preconditioner for FMM

The algorithm for the ILU can be found in [12]. The pseudocode is repeated below for completeness:

```

for i = 2, ..., n, do:
  for k = 1, ..., i-1 and for (i,k) in NZ(Z) do:
    compute  $z_{ik} = z_{ik}/z_{kk}$ 

```

```

    for j = k+1,...,n and for (i,j) in NZ(Z) do:
        compute  $z_{ij} = z_{ij} - z_{ik}z_{kj}$ 
    enddo
enddo
enddo

```

Here $\text{NZ}(\mathbf{Z})$ is the sparsity pattern of the near-field matrix \mathbf{Z} , and the conventional LU decomposition algorithm is only applied to the non-zero entries of the matrix, hence the memory used is not affected and the sparsity pattern of the stored ILU matrix is the same as the original matrix which further saves memory in the sparse store of the ILU matrix.

Performance of Preconditioned CGS Solver

The scattering from a perfectly electrically conducting (PEC) ogive geometry (depicted in Fig. 1) is considered for the performance evaluation of the ILU preconditioner for EFIE, MFIE, and CFIE formulations. The FMM near field matrix is used as the preconditioner. The solution is computed at 5.91 GHz (480 unknowns) for horizontal polarization using a CGS iterative solver.

Fig. 2 shows the residual error as a function of iteration number for the EFIE matrix. Due to the irregular mesh around the sharp tips of the ogive, the CGS solver does not easily converge without preconditioning. Nevertheless, when the ILU preconditioner is used, it is observed to dramatically reduce the number of iterations for convergence.

Non-preconditioned and preconditioned solution data for MFIE formulation is given in Fig. 3. Since the MFIE formulation produces better conditioned systems, the residual error behavior of MFIE is better than that of EFIE, both with and without preconditioning, but dramatically improved by the utilization of the preconditioner.

In Fig. 4, we present the corresponding results for the CFIE formulation. It is important to note the superior effect of the preconditioner in decreasing the number of iterations. Actually, the problem converged in 8 iterations with a residual error less than 10^{-13} .

Table 1 summarizes the performance of ILU preconditioner for a larger problem. The scatterer in this simulation has sharp edges and tips as well as smooth sections. The mesh is quite distorted and non-uniform around

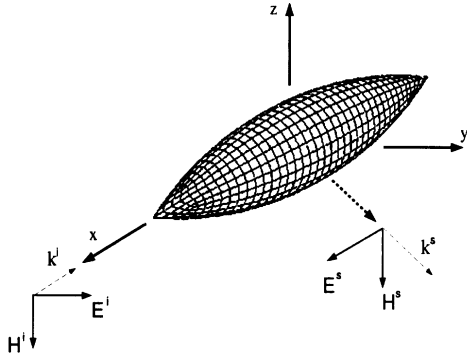


Figure 1: Ogive geometry and the simulation setup

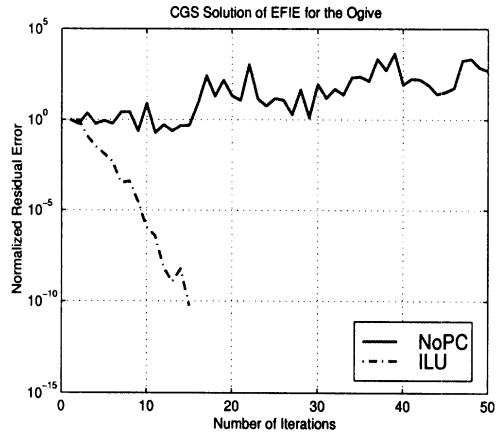


Figure 2: Performance of ILU in conjunction with EFIE formulation

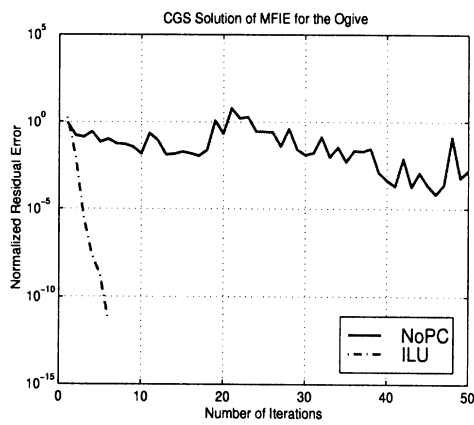


Figure 3: Performance of ILU in conjunction with MFIE formulation

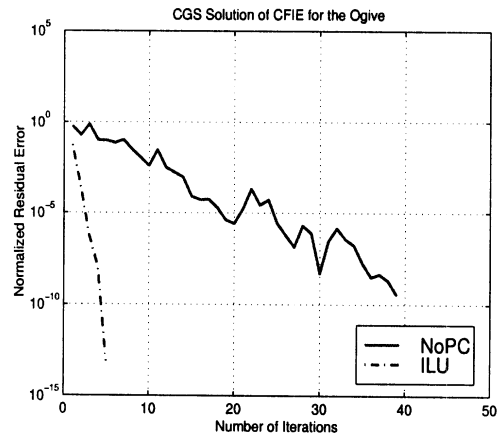


Figure 4: Performance of ILU in conjunction with CFIE($\alpha = 0.5$) formulation

the edges. Despite of these drawbacks, the performance of the ILU preconditioner is quite impressive.

Number of Unknowns	Matrix Fill Time(min) (8 proc.)	Precond. LU Time(min) (1 proc.)	Number of Iter.	Residual Error	Time per Solution(min) (8 proc.)
53000	77	81	5	0.001	5

Table 1: Performance of ILU for a large-scale complex target with sharp edges an tips on 8 processor SGI Origin 2000.

With the above performance evaluations, we conclude that the ILU preconditioner can be used to improve the performance of iterative solvers in FMM implementations without increasing the memory utilized for the preconditioner matrix. Here, the near-field matrix inherent to FMM algorithm is used as the preconditioner resulting in a much faster convergence.

References

- [1] Coifman, R., V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription," *IEEE Antennas and Propagat. Mag.*, vol. 35, pp. 7–12, Jun. 1993.
- [2] Chew, W. C., J. Jin, C. Lu, E. Michielssen, and J. M. Song, "Fast Solution Methods in Electromagnetics," *IEEE Trans. Antennas Propagat.*, vol. AP-45, no. 3, pp. 533–543, Mar. 1997.
- [3] Lu, N. and J.-M. Jin, "Application of the Fast Multipole Method to Finite-Element Boundary-Integral Solution of Scattering Problems," *IEEE Trans. Antennas Propagat.*, vol. AP-44, no. 6, pp. 781–786, 1996.
- [4] Volakis, J. L., A. Chatterjee, and L. C. Kempel, *Finite Element Methods for Electromagnetics*, Piscataway, NJ:IEEE Press, 1998.
- [5] Burkholder, R. J. and D. H. Kwon, "High frequency asymptotic acceleration of the fast multipole method," *Radio Sci.*, vol. 31, pp. 1199-1206, 1996.

- [6] Song, J. M., C. C. Lu, and W. C. Chew, "Multilevel Fast Multipole Algorithm for Electromagnetic Scattering by Large Complex Objects," *IEEE Trans. Antennas Propagat.*, vol. AP-45, no. 10, pp. 1488–1493, 1997.
- [7] Bindiganavale, S. S. and J. L. Volakis, "Comparison of Three FMM Techniques for Solving Hybrid FE/BI Systems," *IEEE Trans. Antennas Propagat.*, vol. AP-39, no. 4, pp. 47–59, Aug. 1997.
- [8] R. L. Wagner and C. W. Chew, "A Ray-Propagation Fast Multipole Algorithm," *Microwave Opt. Tech. Lett.*, vol. 7, no. 7, pp. 435–438, Jul. 1994.
- [9] Lu, N. and W. C. Chew, "Fast Far-Field Approximation for Calculating the RCS of Large Objects," *Microwave Opt. Tech. Lett.*, vol. 8, no. 5, pp. 238–241, 1995.
- [10] Song, J. M. and W. C. Chew, "Multilevel Fast Multipole Algorithm for Solving Combined Field Integral Equations of Electromagnetic Scattering," *Microwave Opt. Tech. Lett.*, vol. 10, no. 1, pp. 14–19, Sep. 1995.
- [11] Song, J. M. and W. C. Chew, "Fast Multipole Method Solution Using Parametric Geometry," *Microwave Opt. Tech. Lett.*, vol. 7, no. 16, pp. 760–765, Nov. 1994.
- [12] Saad, Y., *Iterative Methods for Sparse Linear Systems*, Boston: PWS Pub. Co., 1996.
- [13] G. E. Antilla and N. G. Alexopoulos, "Scattering from Complex 3D Geometries by a Curvilinear Hybrid Finite Element-Integral Equation Approach", *J. Opt. Soc. Am. A*, Vol. 11, No. 4, pp. 1445-1457, 1994