031307-7-T

ANALYSIS OF THREE DIMENSIONAL PLANAR SCATTERERS WITH THE ADAPTIVE INTEGRAL METHOD

Sunil Bindiganavale
John Volakis

January 1997

31307-7-T = RL-2443

Analysis of Three Dimensional Planar Scatterers with the Adaptive Integral Method (AIM)

Sunil S. Bindiganavale and John L. Volakis
Radiation Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122

Abstract

Fast integral equation algorithms provide for a reduction of the computational complexity of moment method integral equation solutions. The adaptive integral method (AIM) is one such algorithm and has been demonstrated to reduce memory down to $O(N^{1.5})$ and computational time down to $O(N^{1.5}\log N)$ for a N unknown surface problem. For planar bodies, the reduction is even further and in this paper we demonstrate the suitability of AIM to model planar scatterers with intricate geometrical details. It is shown that AIM is extremely accurate while saving a significant amount of memory especially for bodies requiring high tessellation due to the inclusion of very small features. AIM is particularly attractive for modeling planar three-dimensional scatterers even on workstation platforms, eliminating the need for elaborate parallelization and domain decomposition procedures. The capability of AIM to model small and intricate perturbations on otherwise smooth bodies and its ability to predict near fields accurately (as reported in this paper) even for bodies which are not electrically very large makes it attractive for many scattering and radiation computations.

1 Introduction

Research on integral equation techniques has been revitalized since the early 1990s with the introduction and development of techniques which accelerate the computation of matrix-vector products employed in the iterative solvers. The Adaptive Integral Method (AIM) [1] and the Fast Multipole Method (FMM) [2] belong to this class of fast integral solution techniques. Both AIM and FMM simultaneously reduce the solution time and memory requirement of the moment method (MM) solutions. While most of the initial applications of these techniques focussed on electromagnetic scattering from large conducting bodies, recently, they have also been used successfully in hybrid methods [3],[4] to evaluate the scattering from composite structures. FMM achieves its CPU reduction by grouping the far-zone unknowns and interacting their weighted contributions. In the case of AIM the CPU reduction is achieved by mapping the original MM discretization to a rectangular grid and exploiting the Toeplitz property of the Green's function on this grid by invoking the Fast Fourier Transform (FFT) to compute the matrix-vector products. For an arbitrary three dimensional body, a three dimensional FFT is required and as can be understood this FFT calculation is more time consuming. However, for planar scatterers the dimensionality of the FFT is reduced by one thereby significantly accelerating the solution. In this paper, we also examine the benefits of AIM when the body is not electrically large, but is highly tessellated owing to its intricate construction, thus leading to a large unknown count.

2 AIM for Planar Scatterers

The AIM algorithm for arbitrary three dimensional bodies has been given in [1]. In this section, we describe its specialization to planar scatterers. We describe only the essential details required for implementation of the algorithm. We consider a planar, arbitrary, conducting body whose surface S is illuminated by an incident plane wave $\mathbf{E}^i$. The boundary condition enforced on S is

$$(\mathbf{E}^i + \mathbf{E}^s) \cdot \hat{t} = 0 \tag{1}$$

where the scattered field $\mathbf{E}^s$ is given by

$$\mathbf{E}^s = -j\omega\mathbf{A} - \nabla\phi \tag{2}$$

in which

$$\mathbf{A}(\mathbf{r}) = \frac{\mu}{4\pi}\int_S \mathbf{J}_s(\mathbf{r}')\frac{e^{-jkR}}{R}dS' \tag{3}$$

is the vector potential and

$$\phi(\mathbf{r}) = \frac{1}{4\pi\epsilon}\int_S \rho_s(\mathbf{r}'\frac{e^{-jkR}}{R}dS' \tag{4}$$

is the scalar potential. The continuity equation

$$\nabla_s \cdot \mathbf{J}_s = -j\omega\rho_s \tag{5}$$

provides the relation between charges and currents. Substituting (2)- (5) into (1) gives the necessary integral equation which must be discretized for the solution of $\mathbf{J}_s$. To do so, the equivalent electric current density $\mathbf{J}_s$ is expanded using linear basis functions [5] as

$$\mathbf{J}_s(\mathbf{r}') = \sum_N I_n \mathbf{f}_n(\mathbf{r}') \tag{6}$$

where I_n are the unknown coefficients. Application of Galerkin's technique leads to the linear system

$$[Z]\{I\} = \{V\} \tag{7}$$

with $[Z]$ being the element interaction matrix, whereas $\{I\}$ is the vector of the unknown coefficients and $\{V\}$ is the excitation vector. The matrix $[Z]$ is full and requires $O(N^2)$ storage. Also, each $[Z]\{I\}$ matrix-vector product requires $O(N^2)$ multiplications. Fast algorithms such as the FMM and AIM are used to reduce the operation count from N^2 to N^α where $\alpha \leq 1.5$. Both algorithms work on approximating the far zone interactions. In the case of AIM, the CPU reduction is achieved by first splitting the matrix as

$$[Z] = [Z^{near}] + [Z^{far}] \tag{8}$$

based on a threshold distance also referred to as the near-zone radius. The matrix $[Z^{near}]$ contains the interactions between elements with separation distance less than the threshold distance, while $[Z^{far}]$ contains the remaining interactions. The elements of $[Z^{near}]$ are evaluated with the exact MM while the elements of $[Z^{far}]$ and the product $[Z^{far}]\{I\}$ are evaluated in an approximate manner according to the AIM procedure. Application of AIM requires that the whole geometry be enclosed in a regular rectangular grid.

Each interior edge is expressed as a superposition of delta functions at the vertices of a square that surrounds its center as shown in Figure 1. The new basis functions ψ_m corresponding to the mth edge are given by

$$\psi_m = \sum_{q=1}^{M^2} \delta(x - x_{mq})\delta(y - y_{mq})[\Lambda^x_{mq}\hat{x} + \Lambda^y_{mq}\hat{y}] \tag{9}$$

where $\mathbf{r}_{mq}$ are locations of M^2 points on the square surrounding the center of the edge, and $\delta(\mathrm{x})$ is the usual Dirac delta function and $\Lambda^{x,y}_{mq}$ are suitable coefficients chosen so that the new expansion is equivalent to the original representation using triangular elements. A similar representation is used for the divergence of the basis functions

$$\psi^d_m = \sum_{q=1}^{M^2} \delta(x - x_{mq})\delta(y - y_{mq})\Lambda^d_{mq} \tag{10}$$

Equivalence is achieved by equating moments of the two expansions up to order M and this provides a relation between the Λ coefficients and I_n coefficients in (6). The moments are given by

$$\begin{aligned} M^m_{q_1 q_2} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \psi_{\mathbf{m}}(x - x_a)^{q_1}(y - y_a)^{q_2} dxdy \;\; for \;\; 0 \le q_1, q_2 \le M \\ &= \sum_{q=1}^{M^2}(x_{mq} - x_a)^{q_1}(y_{mq} - y_a)^{q_2}[\Lambda^x_{mq}\hat{x} + \Lambda^y_{mq}\hat{y}] \;\; with \;\; q = q_1 + q_2 \end{aligned} \tag{11}$$

$$\mathbf{F}^m_{q_1 q_2} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{f}_m(x - x_a)^{q_1}(y - y_a)^{q_2} dxdy \tag{12}$$

for the equivalent and original basis functions, respectively. We need to also consider the moments of the divergence expansions. Namely,

$$D^m_{q_1 q_2} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \psi^d_m(x - x_a)^{q_1}(y - y_a)^{q_2} dxdy = \sum_{q=1}^{M^2}(x_{mq} - x_a)^{q_1}(y_{mq} - y_a)^{q_2}\Lambda^d_{mq} \tag{13}$$

$$H^m_{q_1 q_2} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \nabla_s \cdot \mathbf{f}_m(x - x_a)^{q_1}(y - y_a)^{q_2} dxdy \tag{14}$$

Equating (11) and (12) and also (13) and (14) yields three $M^2 \times M^2$ systems yielding the equivalence coefficients as the solution. This process is depicted pictorially in Figure 1.

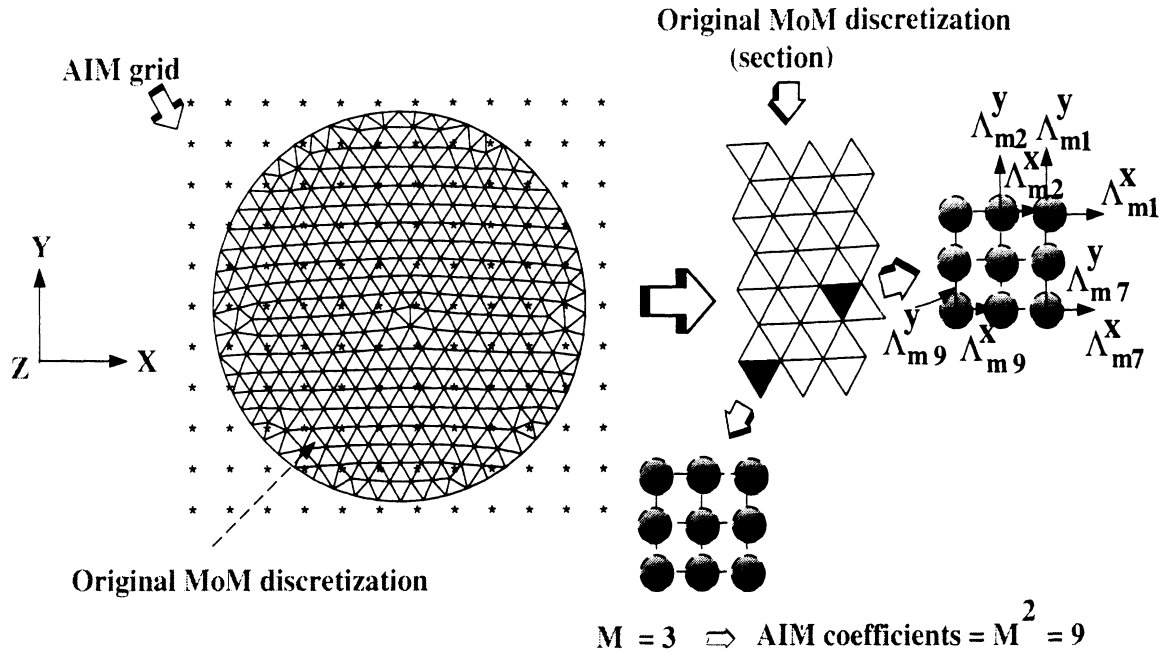

Figure 1: The process of transformation from the original MM grid onto the AIM grid

Were we to use the equivalent expansions to represent the currents everywhere, the resulting impedance matrix will be

$$[Z]_{AIM}^{total} = \sum_{i=1}^{3} [\Lambda]_i [G] [\Lambda]_i^T \tag{15}$$

In this $[\Lambda]_i$ are the sparse matrices containing the coefficients of the expansion (11) and (10) and $[G]$ is the Toeplitz matrix whose elements are the free space Green's function evaluated at the grid points. It has been shown in [1] that $[Z_{AIM}^{total}]$ is not of sufficient accuracy for modeling the interactions between the nearby current elements. To take advantage of the Toeplitz structure of $[G]$ and sparsity of $[\Lambda]$ we can still use $[Z_{AIM}^{total}]$ to represent the far element interactions. However we will retain the exact interaction matrix elements for the near element interactions. That is we rewrite $[Z_{AIM}^{total}]$ as

$$[Z]_{AIM}^{total} = [Z]_{AIM}^{near} + [Z]_{AIM}^{far} \tag{16}$$

From (8) and for the far field interactions $[Z]^{far} \simeq [Z]^{far}_{AIM}$ thus giving

$$[Z] \simeq [Z]^{near} - [Z]^{near}_{AIM} + [Z]^{total}_{AIM} \tag{17}$$

$$[Z] \simeq [S] + \sum_{i=1}^{3} [\Lambda]_i [G] [\Lambda]_i^T \tag{18}$$

where $[S]$ is a sparse matrix corresponding to the difference between the near field interactions computed by the MM and AIM. The Toeplitz property of the Green's function, defined on the regular grid, enables use of the FFT to accelerate the computation of the matrix-vector product. The sequence of operations involved in the construction of the coefficient and Green's function matrices are indicated in Figure 2(a) while those for the matrix-vector product execution are indicated in Figure 2(b). In the computation of the matrix-vector product, the initial step of transforming the currents from the original MM grid onto the uniform AIM grid is comparable to the grouping operation of the FMM. While the FMM relies on grouping to reduce the number of scattering centers, the sequence of operations in AIM can be interpreted as a realignment of scattering centers onto a regular grid. Although, this process does not reduce the number of scattering centers the regularity of their location enables the use of the FFT for fast computation of matrix-vector products.

3 Results

When examining the merits of a fast integral algorithm such as AIM, of importance is the memory and CPU requirements, both contrasted to the delivered accuracy. Although approximate analytical expressions have been derived in [1] for some of these parameters, our goal is to give quantitative comparisons for typical planar geometries. The near-zone radius or threshold distance has a dramatic impact on the CPU requirements since it controls the non-zero element population of the system matrix. Moreover, in the case of AIM, because of the inherent mapping to a constant grid, we are highly interested in examining its suitability in modeling small and fine details embedded in much larger scale structures. In fact, the calculations for the six plate configurations given below are intended to address this issue by examining the method's performance in a number of representative and practical situations. All of the included results were generated using single precision arithmetic on an HP9000/C-110 workstation with a rated peak flop rate of

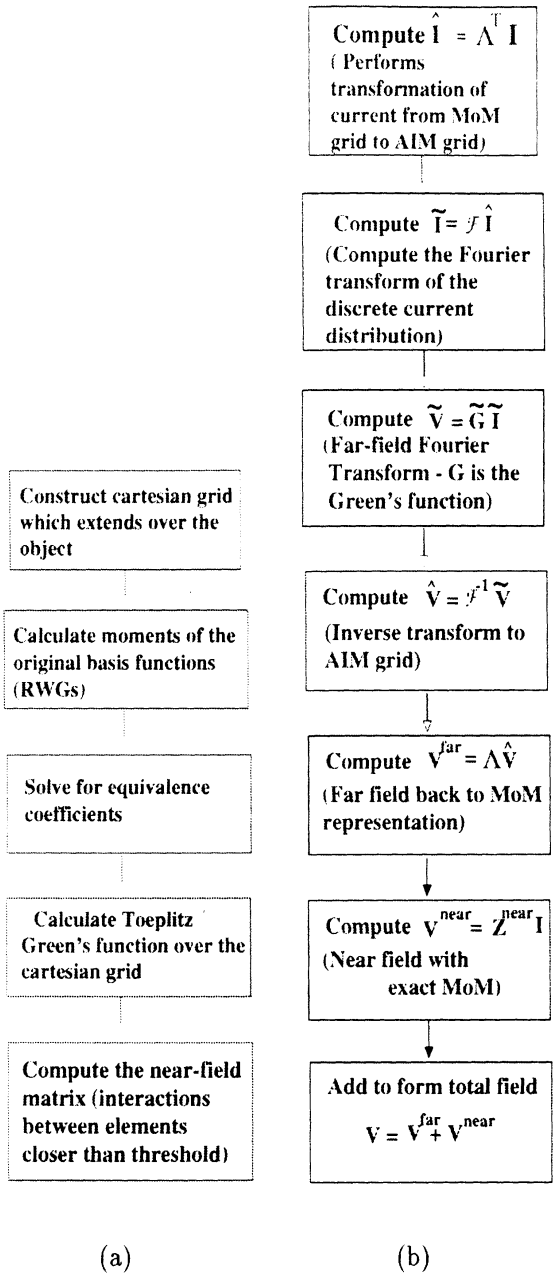

Figure 2: (a) Matrix build operations and (b) Matrix vector product computation in AIM

47 Mflops (level 4 optimization was also employed). Also, in all cases a third order (M=3) multipole expansion was used with a grid spacing of 0.05λ.

Figures 3-9 depict the $\hat{\theta}\hat{\theta}$ and $\hat{\phi}\hat{\phi}$ polarization radar cross section as calculated by the AIM method for the different threshold distances indicated. The pattern cuts are in the $\phi = 0^0$ principal plane of the plate structure. The first two plates (a rectangular and a circular) have no holes and were used to validate the method with the traditional MM. From the pattern comparisons, it is clear that AIM recovers the exact result very well with only minor differences in the deep nulls of the RCS pattern. As given in Table 1 and 2, AIM achieves this with at least a factor of five less memory than the traditional MM even though the geometries are still rather small to demonstrate the impact of AIM. Moreover, Table 2 shows that a near zone radius of 0.2λ is sufficient to maintain good accuracy (below one dB in RMS error [6]). To our surprise, the advantage of AIM is even more pronounced when holes are inserted into the plates surfaces. As depicted in Figures 5-7, AIM maintains its accuracy for the same threshold criterion even though the holes/slots have a dominant effect on the RCS pattern as shown in Figure 11. In the case of narrow slots (or thin ridges in the plates) of width 0.03λ, the memory requirements of the traditional MoM increase quickly due to the higher element density. However, in AIM we can still use a constant and uniform grid density of 0.05λ without accuracy deterioration, even though the cell width is larger than the smallest detail in the geometry. **The geometry in Figure 8 yields a saving in memory of 79% while the CPU time is cut by a factor of 12, retaining the monostatic pattern accuracy to a tenth of a dB.** Such an accurate estimation of the far field suggests that the near field should also be in very small error. Examining the currents for this configuration revealed an average error of 7.3% for a near zone radius of 0.2λ and only 6% for the threshold of 0.4λ for normal incidence. The currents for this geometry at key locations are indicated in Figure 10. This geometry demonstrates an important feature of AIM, that the near zone threshold criterion is not affected by the specific geometrical details, leading to tremendous memory savings. Moreover, the accuracy of the results demonstrate that AIM can handle highly irregular and resonant (i.e. antenna) geometries as well as smooth scatterers. The residual error in the Biconjugate gradient algorithm as a function of number of solution iterations is shown in Figure 12 for both MM and AIM. It is seen that though the convergence curve for the AIM shows some deviation, the number of iterations for convergence is approximately the same indicating that the condition

of the system is not appreciably altered. This is of paramount importance in fast iterative solutions, since an increase in iteration count caused by poor system conditioning would annul the faster computation of the matrix-vector product. Figure 9 shows the monostatic RCS pattern for a grating structure which acts as a "polarization filter". The thin ridges in the grating cause a high specular return for the $\hat{\phi}\hat{\phi}$ polarization at incidence angles close to normal and this is evident in the higher return (almost 10 dB) for the structure in Figure 9(a) when compared to the one in Figure 9(b). We also note that this structure is very densely meshed with element sizes in the ridge of 0.02λ while the other elements were larger (0.025λ). In this case we used two different grid spacings - 0.05 λ and 0.1 λ. From Table 3, with a grid spacing of 0.1λ the solution time is cut by a third at the expense of accuracy. While memory requirements for the two grid spacings are approximately the same, the FFT pad with a grid spacing of 0.1λ (32) is one half of that with a grid spacing of $0.05\lambda(64).Withagridspacingof0.05$ λ memory requirement over the exact MM is reduced by a factor of about **8**, solution time by a factor of **9** with loss of accuracy of only a **twentieth** of a dB for the RCS. Finally, it is seen in Figures 13 and 14 that the small size of the elements in the mesh and the small non-uniformity in the mesh does not seem to have a deleterious effect on the AIM system condition as is eveident from the number of iterations to convergence. It should be noted that when comparing solution times we used identical biconjugate gradient solvers without any form of preconditioning.

4 Summary

The performance is clearly much improved when using a 2D (rather than a 3D) FFT for calculating the RCS of flat complex scatterers. A memory reduction of 5 to 10 times over traditional MM was observed without compromise in accuracy when using a threshold radius of down to 0.2λ. There was appreciable savings in CPU time for the geometry in Figure 6 and the CPU time can be reduced even further by parallelization or by using optimized FFTs which has already been reported in literature. The algorithm is capable of modeling very small details in large bodies with a high degree of accuracy, while simultaneously saving considerable memory.

5 Acknowledgement

The first author would like to thank Mr. Hristos Anastassiu for helpful discussions on AIM.

Geometry	Discretization			MM memory (MB)	MM solution time $\hat{\theta}\hat{\theta}$ pol ($\theta = 0^0$ inc.)
	Facets	Edges	Unknowns		
Figure 3	240	403	317	0.76	6 secs
Figure 4	586	908	850	5.51	32 secs
Figure 5	710	1129	1001	7.64	1 min 32 secs
Figure 6	554	890	772	4.54	29 secs
Figure 7	1130	1806	1584	19.14	4 mins 50 secs
Figure 8	1036	1667	1441	15.84	4 mins
Figure 9	1038	1957	1157	10.21	2 mins 45 secs

Table 1: Solution CPU time and memory requirement of the moment method

References

[1] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz. AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems. *Radio Science*, 31(5):1225–1251, Sept-Oct 1996.

[2] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comput. Phys.*, 86(2):414–439, 1990.

[3] S. S. Bindiganavale and J. L. Volakis. A hybrid FEM-FMM technique for electromagnetic scattering. *IEEE Trans. Antennas and Propagat.*, 45(1):180–181, January 1997.

[4] N. Lu and J. M. Jin. Application of the fast multipole method to finite element-boundary integral solution of scattering problems. *IEEE Trans. Antennas and Propagat.*, 44(6):781–786, 1996.

[5] S.M. Rao, D.R. Wilton, and A.W. Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans. Antennas and Propagat.*, 30(3):409–418, May 1982.

Geometry	AIM Data					
	Threshold (λ)	Non-Zeros in Near Z	Memory (MB)	Solution time $\hat{\theta}\hat{\theta}$ pol ($\theta = 0^0$ inc.)	RMS Error(dB) $\hat{\theta}\hat{\theta}$ pol	$\hat{\phi}\hat{\phi}$ pol
Figure 3	0.3	13724	0.15	6 secs	3.5815	0.9878
	0.5	22996	0.26	7.2 secs	2.2636	0.7070
	0.7	31720	0.36	7.5 secs	1.2806	0.4477
Figure 4	0.3	59928	0.68	23 secs	0.1718	0.0755
	0.4	100182	1.14	25 secs	0.1490	0.0693
	0.7	257390	2.94	28 secs	0.0728	0.0490
Figure 5	0.7	178664	2.04	1 min 17 secs	1.4654	0.7501
Figure 6	0.4	79030	0.9	21 secs	0.0728	0.0583
	0.6	157994	1.8	27 secs	0.0721	0.0520
Figure 7	0.7	283774	3.24	3 mins 32 secs	0.8017	0.5185
Figure 8	0.2	296250	3.39	20 secs	0.1063	0.0949
	0.4	649556	7.43	31 secs	0.0548	0.0632
Figure 9	0.2	120220	1.37	18 secs	0.0469	0.0469

Table 2: Solution CPU time, memory requirement and RMS error of AIM (all entries in this table were computed with an AIM grid spacing of 0.05λ)

Grid spacing	Solution time $\hat{\theta}\hat{\theta}$ pol ($\theta = 0^0$ inc.)	RMS Error(dB) $\hat{\theta}\hat{\theta}$ pol	$\hat{\phi}\hat{\phi}$ pol
0.05	18 secs	0.0469	0.0469
0.1	12 secs	0.4150	0.3058

Table 3: Effect of AIM grid spacing on solution time and error

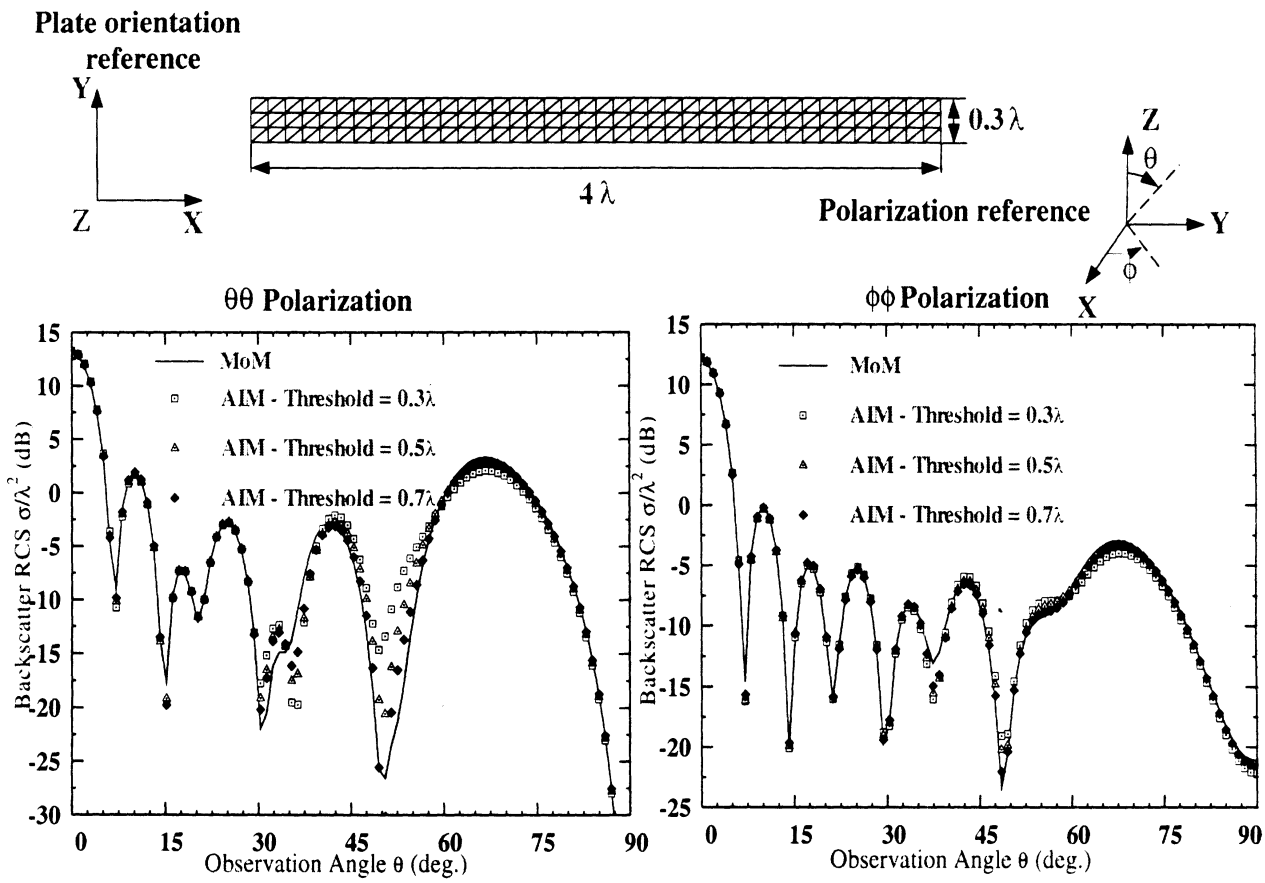

Figure 3: Monostatic RCS for a rectangular $4\lambda \times 0.3\lambda$ plate computed with standard MM & AIM

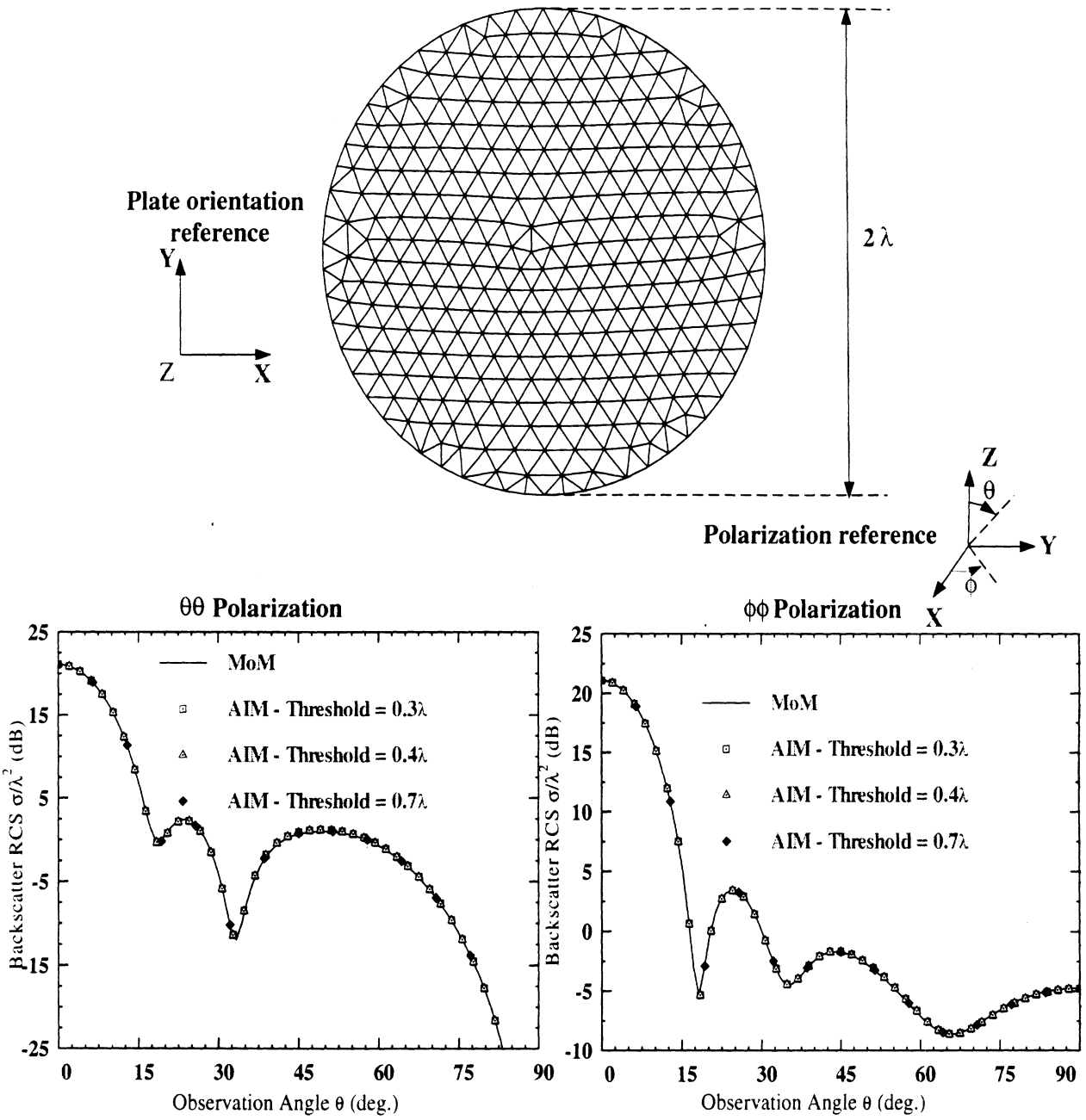

Figure 4: Monostatic RCS for a circular plate of diameter 2λ computed with standard MM & AIM

[6] S. S. Bindiganavale and J. L. Volakis. Guidelines for using the fast multipole method to calculate the RCS of large objects. *Micro. Opt. Tech. Lett.*, 11(4):190–194, March 1996.

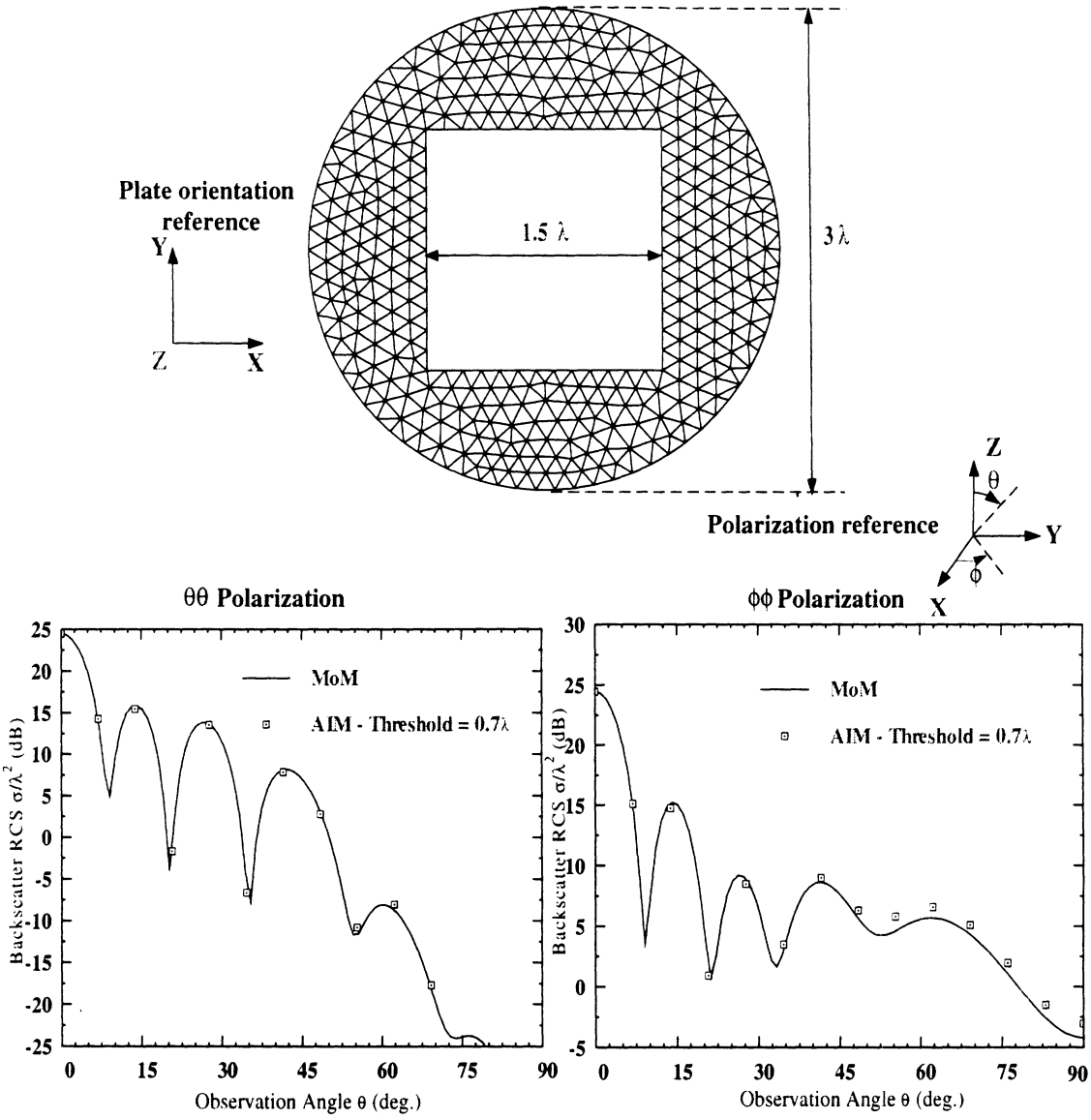

Figure 5: Monostatic RCS for a circular plate of diameter 3λ with a square hole of side 1.5λ computed with standard MM & AIM

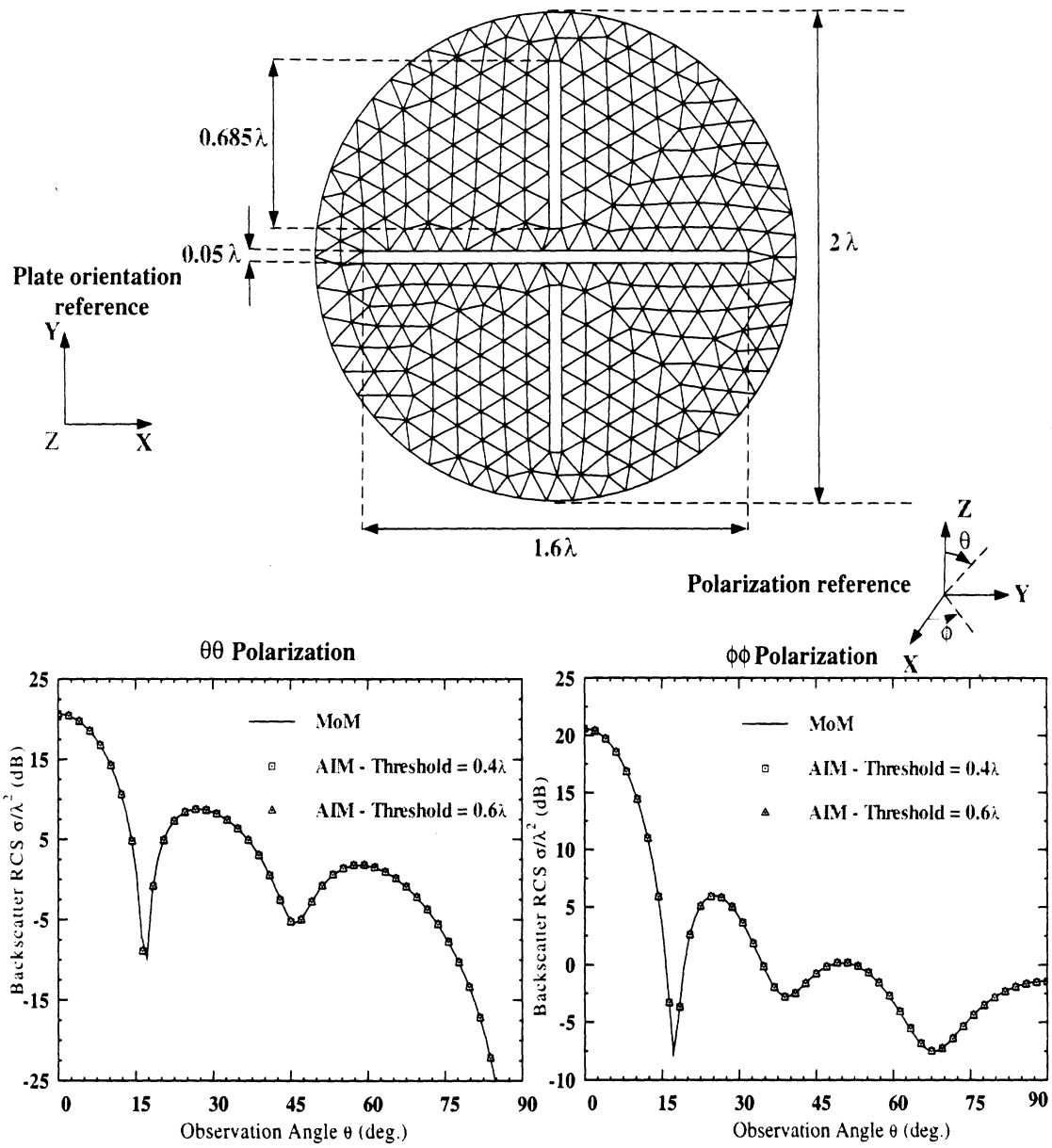

Figure 6: Monostatic RCS for a circular plate of diameter 2λ with three slots computed with standard MM & AIM

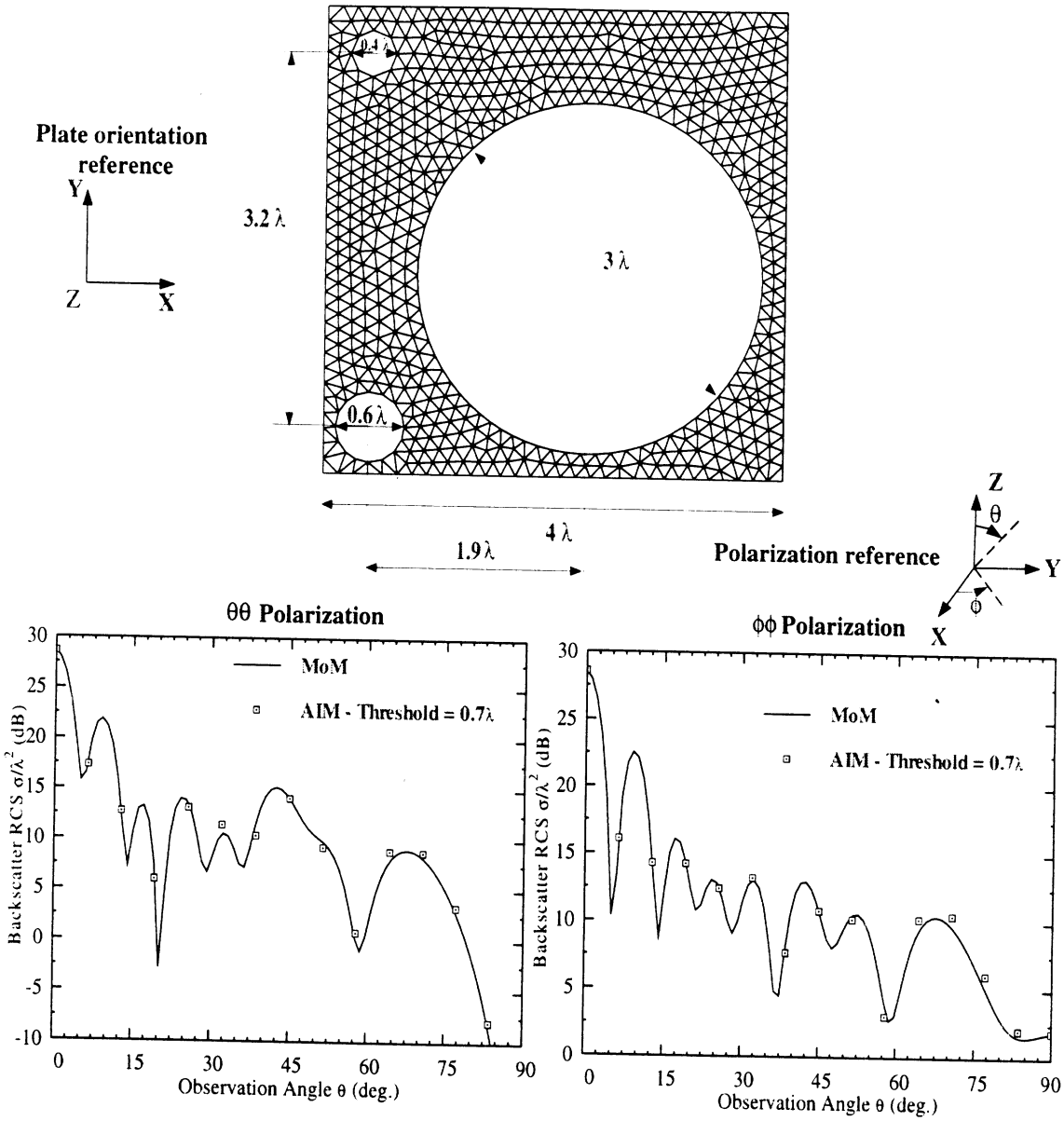

Figure 7: Monostatic RCS for a circular plate of diameter 3λ with three holes computed with standard MM & AIM

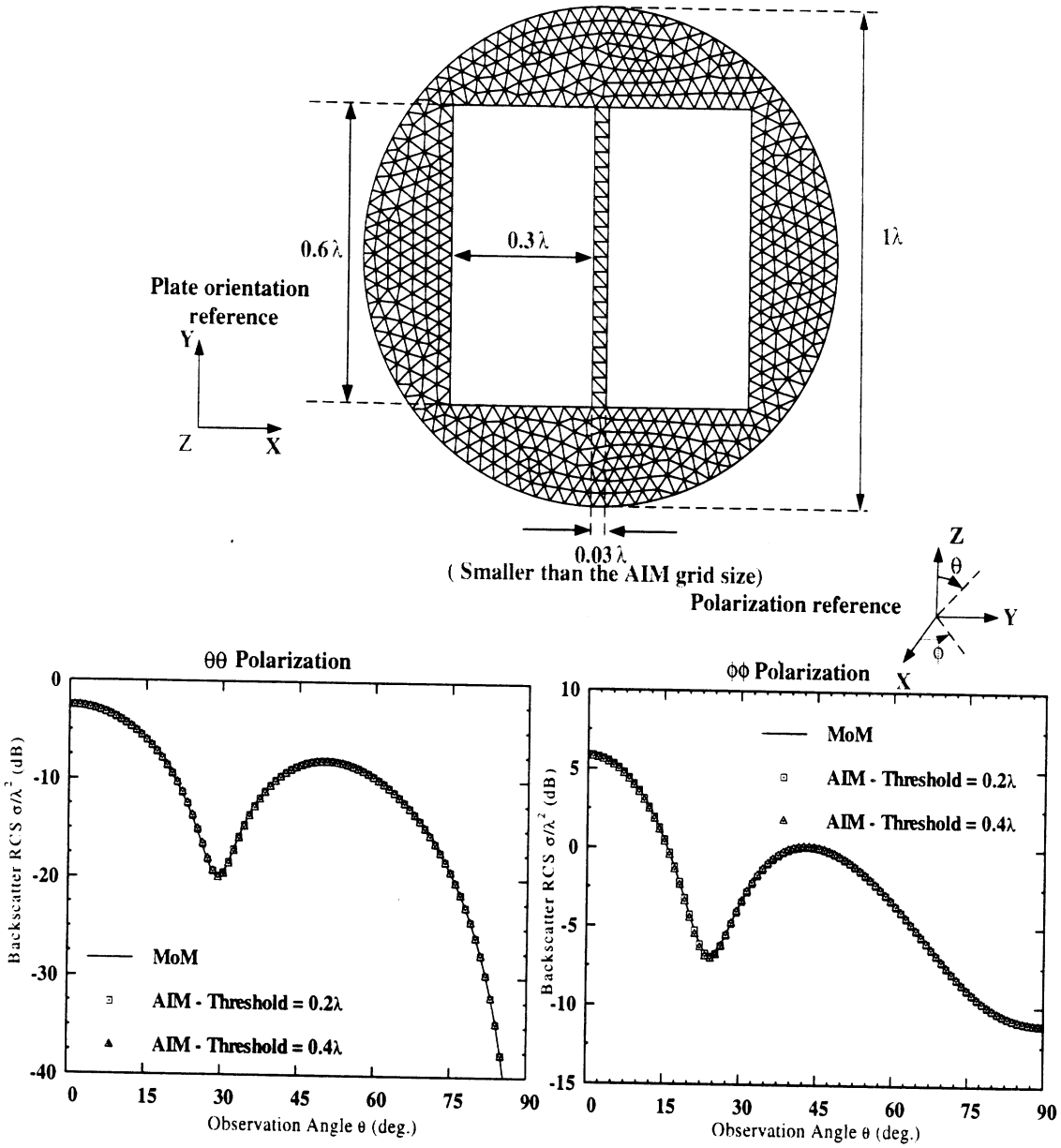

Figure 8: Monostatic RCS for a circular plate of diameter 1λ sampled at 0.03 λ (smaller than the AIM grid spacing) due to the narrow center ridge

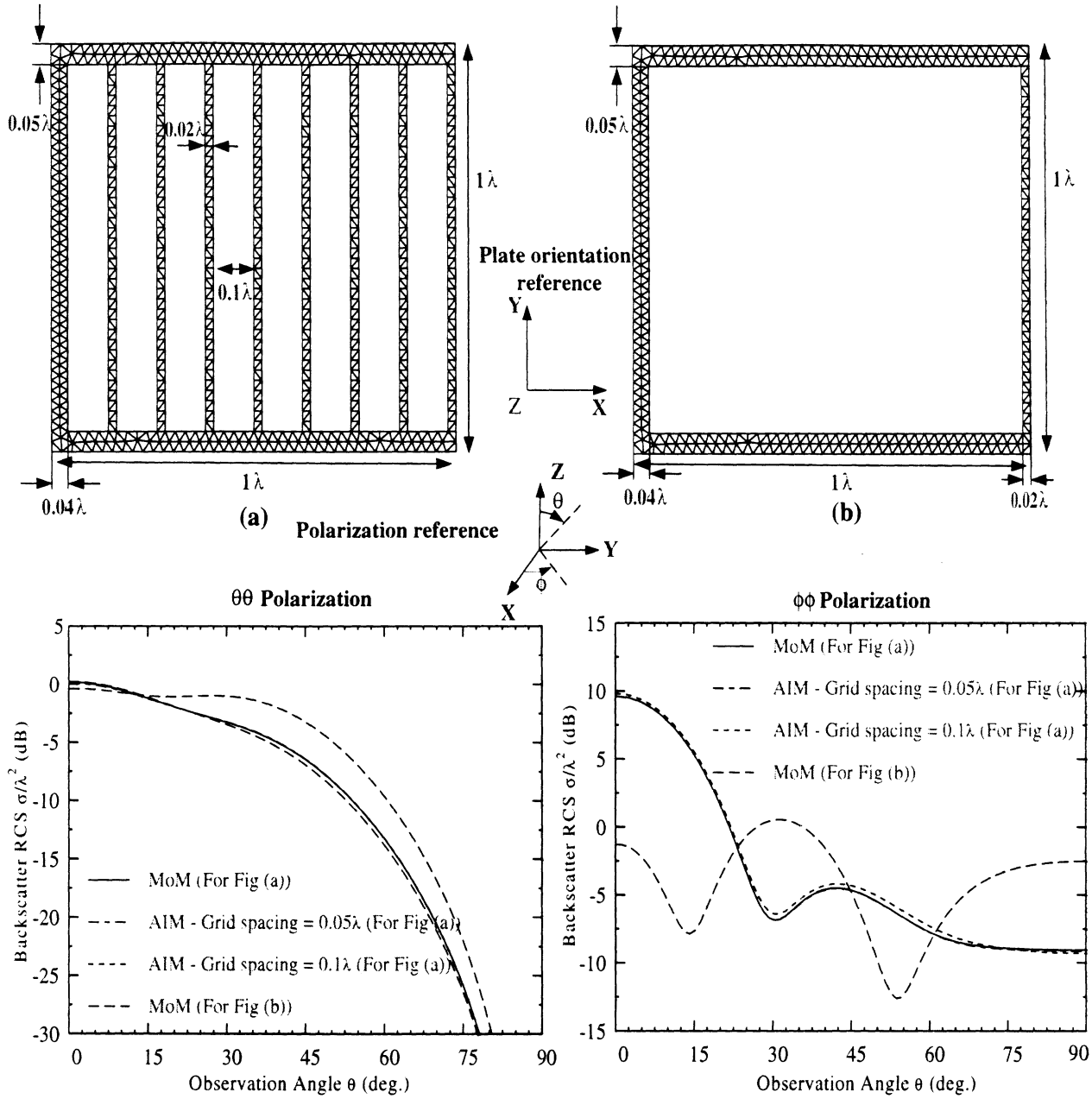

Figure 9: Monostatic RCS for a grating which acts like a "polarization filter". Note the use of an AIM grid spacing of 0.1λ which is *five* times the size of the smallest detail in the mesh

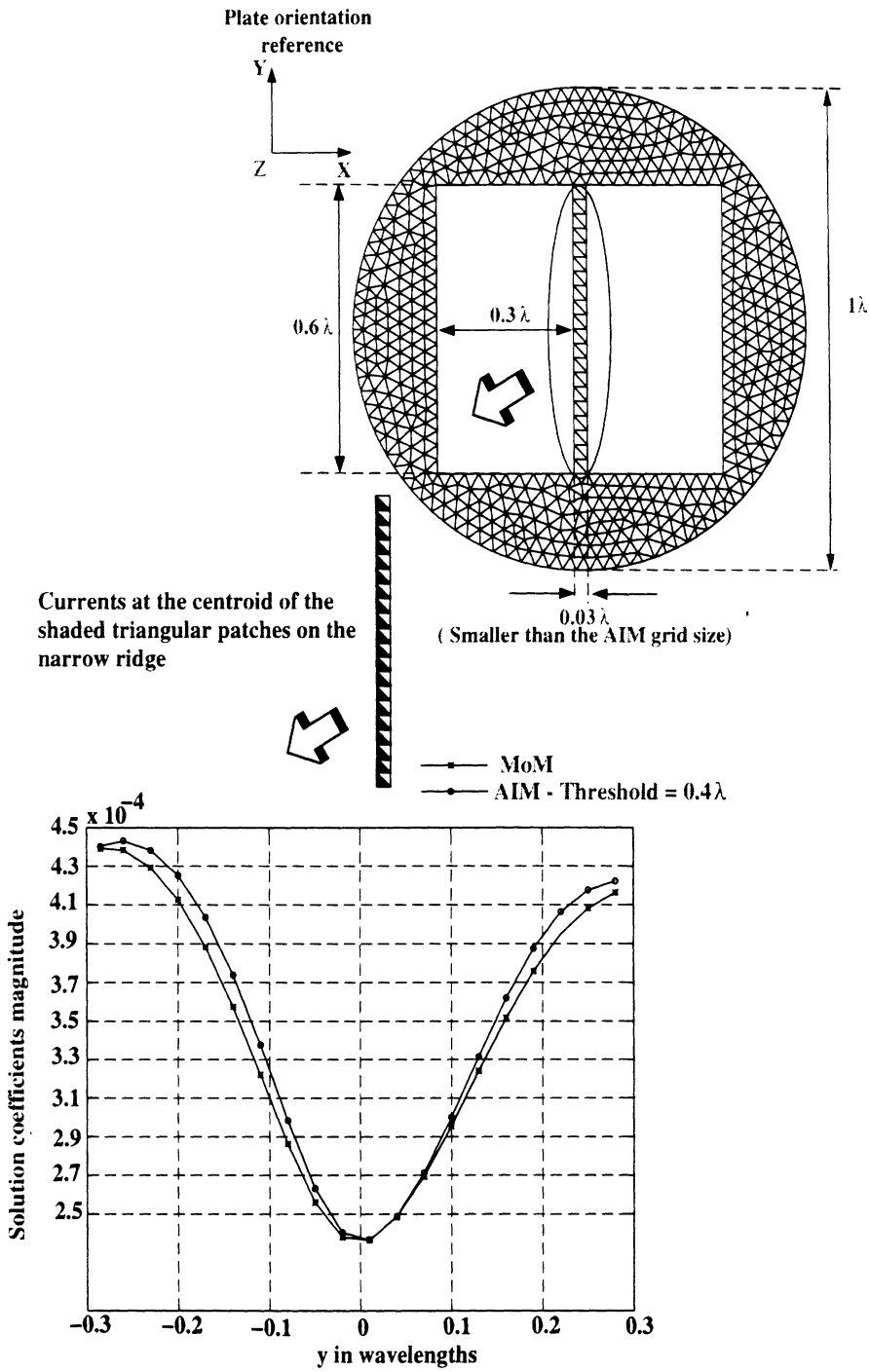

Figure 10: Electric currents (Solution coefficients) on the narrow ridge for the geometry of Figure 8

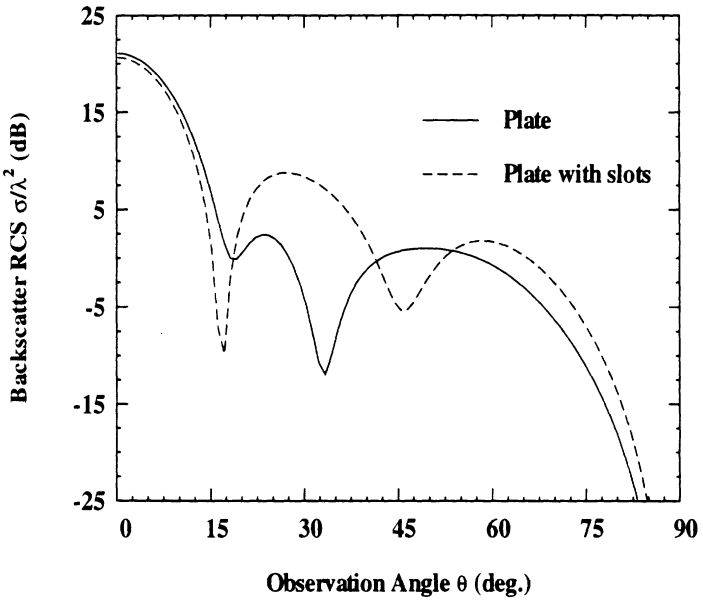

Figure 11: Effect of the narrow slots of Figure 6 on the Monostatic RCS of the 2λ diameter plate of Figure 4

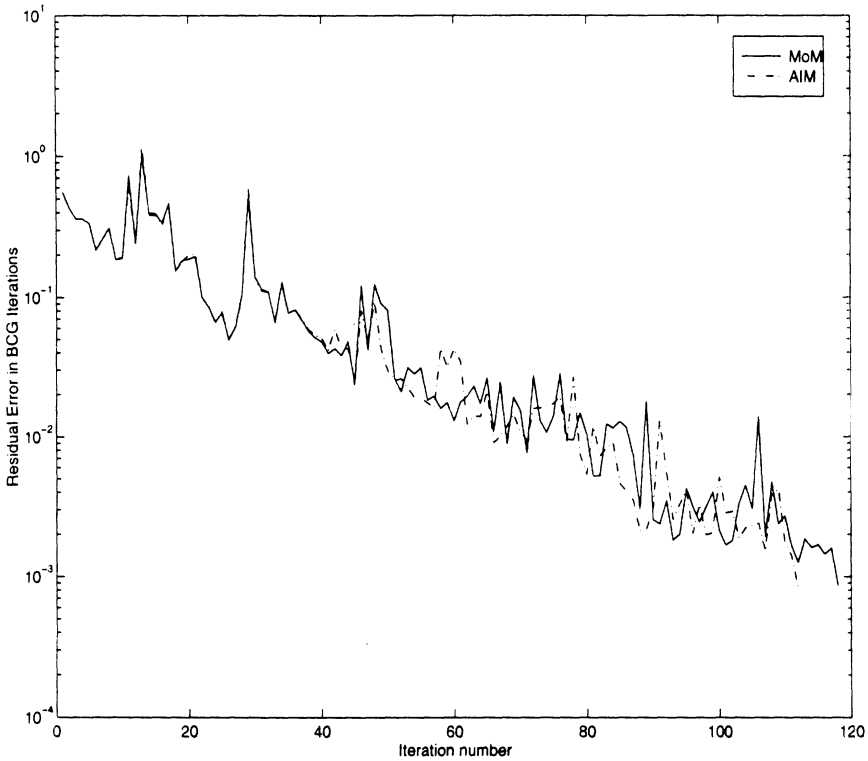

Figure 12: Biconjugate gradient solver convergence curves for the 2λ diameter circular plate at normal incidence

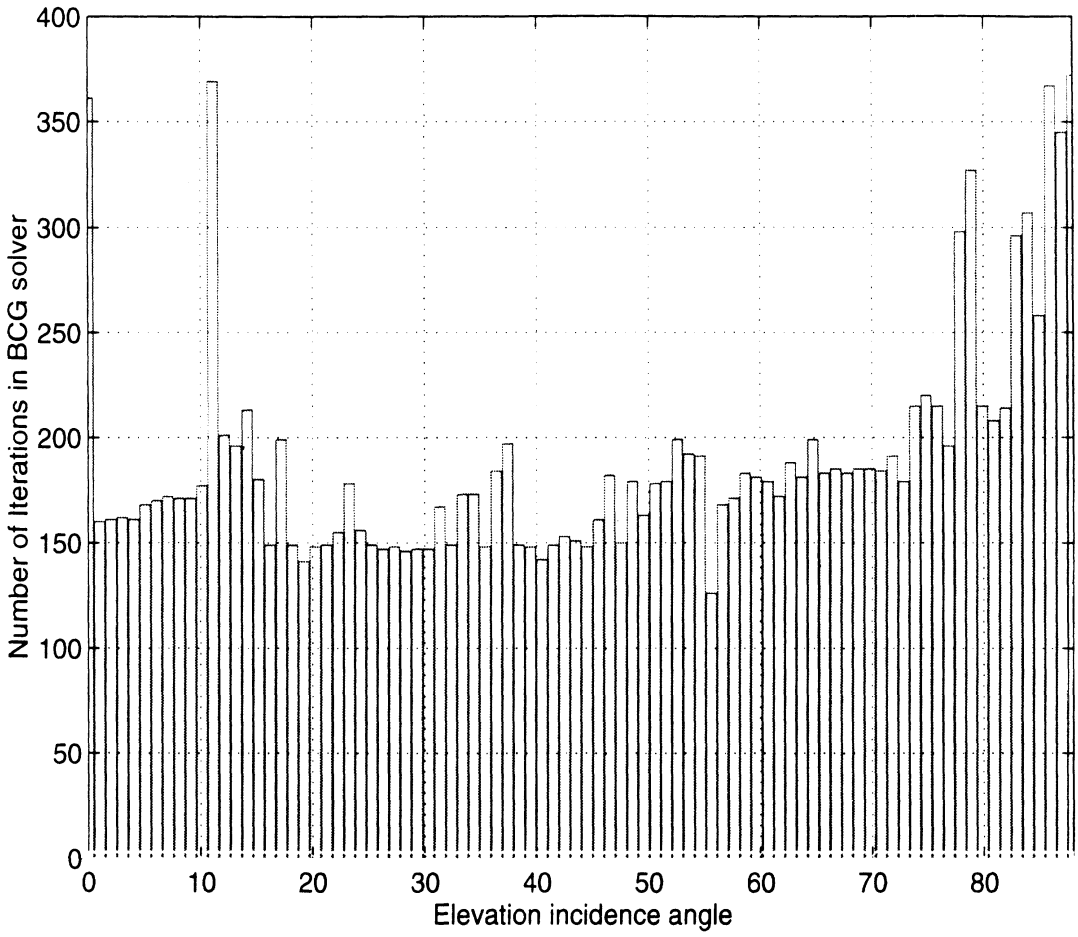

Figure 13: Number of iterations in the BCG solver as a function of incidence angle for $\hat{\theta}\hat{\theta}$ polarization for the geometry of Figure 9

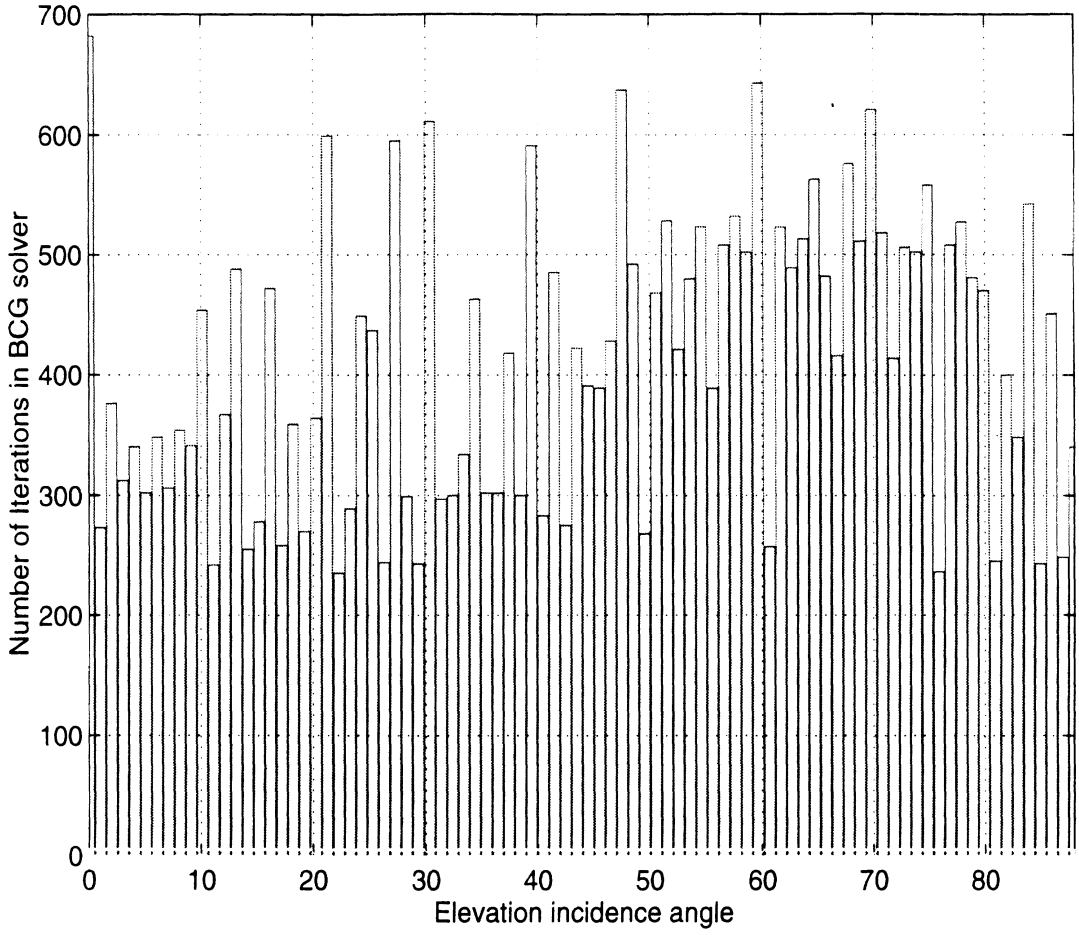

Figure 14: Number of iterations in the BCG solver as a function of incidence angle for $\hat{\phi}\hat{\phi}$ polarization for the geometry of Figure 9