032248-1-T

Error and Execution Time Analysis of the Fast Multipole Method for Calculating the RCS of Large Objects

S. Bindiganavale and J. L. Volakis

Naval Air Warfare Center
Weapons Division
China Lake, CA 93555

July 1995

32248-1-T = RL-2451

Report #032248-1-T

Project Title: Nose Antenna-Radome Analysis

Report Title: Error and Execution Time Analysis of the Fast Multipole Method for Calculating the RCS of Large Objects

Report Authors: S. Bindiganavale and J.L. Volakis

Date: July 1995

Navy Coordinator: Dr. Helen Wang
Code C2951
Naval Air Warfare Center
Weapons Division
China Lake, CA 93555

University Project Director: John L. Volakis
Radiation Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
1301 Beal Avenue
Ann Arbor, MI 48109-2122
e-mail: volakis@umich.edu
phone: (313) 764-0500
fax: (313) 747-2106

Error and execution time analysis of the fast multipole method for calculating the RCS of large objects

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

In this report we consider with particular emphasis to error and execution time the implementation of a version of the fast multipole method for scattering by large objects. In contrast to the traditional moment method, the fast multipole method has $O(N^{1.5})$ CPU requirement per incidence angle. This substantially reduced CPU time is achieved by subdividing the far zone elements into groups whose weighted contribution is then interacted with the test element. The size of the groups and the various approximations used in the interaction of the groups play an important role on the solution accuracy, but so far no error analysis has been performed. In this report we make a thorough study of the solution error as a function of all parameters controlling the accuracy of the fast multipole method. Our study is carried out with reference to a version of the FMM referred to as the Fast Far Field Approximation.

1 Introduction

Integral equation methods and in particular the moment method were introduced first in the 1960s [1]. In the ensuing years the moment method achieved spectacular success and was applied to model a wide variety of complicated electromagnetic phenomena. Among its chief attractions were the exactness of the solution technique and the ability to model complicated geometries with ease. However, these advantages come at a price. The price to be paid for the exactness of the solution was the full matrix systems, a necessary part of the technique. Since full matrices need a storage of $O(N^2)$, the MoM suffers severe problems of scalability, especially for 3D and for large 2D problems. Recently, researchers have been trying to develop IE techniques which make the matrix system sparse by using special basis functions [2] or by making use of filtering techniques to isolate the dominant elements of the moment-method matrix [3]. Another technique relies on clumping the matrix elements in the far-field [4],[5]. The technique which we have adopted is based on the latter principle and was introduced in [6]. This technique reduces the operation count in the iterative solution of moment method problems from $O(N^2)$ to $O(N^{1.5})$ and is based on computing the interactions between different elements using different schemes depending on the electrical distance between the elements. For small interaction distances, the exact kernel is employed while for large distances an approximate kernel is used. The memory requirement, with the preclusion of matrix factorization, is reduced to $O(N)$, thus enabling the solution of larger problems.

2 Formulation

The computation of the scattered field from two-dimensional structures is traditionally accomplished via a direct numerical solution of the appropriate integral equations. For two-dimensional resistive and metallic structures, the appropriate integral equations are the electric field integral equation (EFIE) and the magnetic field integral equation (MFIE).

2.1 TM incidence

For TM incidence, the EFIE is given by ($e^{-i\omega t}$ convention)

$$\frac{k_0}{4}\int_C J_z(l')H_0^{(1)}\left(k_0|\overline{\rho}-\overline{\rho'}|\right)\,dl' + R(l)J_z(l) = Y_0 E_z^i(l) \tag{1}$$

with the geometry as depicted in Figure 1. A pulse-basis point-matching discretization of this integral equation results in an impedance matrix whose elements are given by

$$\begin{aligned} Z_{ij} &= R_i + \frac{\delta_j}{\lambda}\left[\frac{\pi}{2} + \mathrm{i}\left(\ln\frac{\delta_j}{\lambda} + 0.02879837\right)\right], \quad i = j && (2)\\ &= \frac{\pi}{2}\int_{S_j} H_0^{(1)}\left(k_0|\overline{\rho_j}-\overline{\rho_i}|\right) d\left(\frac{\delta_j}{\lambda}\right), \quad i \neq j && (3)\end{aligned}$$

The latter integral is evaluated using 3-point Simpson's rule,

$$\int_{a_j}^{b_j} f(x)dx = \frac{\delta_j}{6}(f(1) + 4f(2) + f(3)) \tag{4}$$

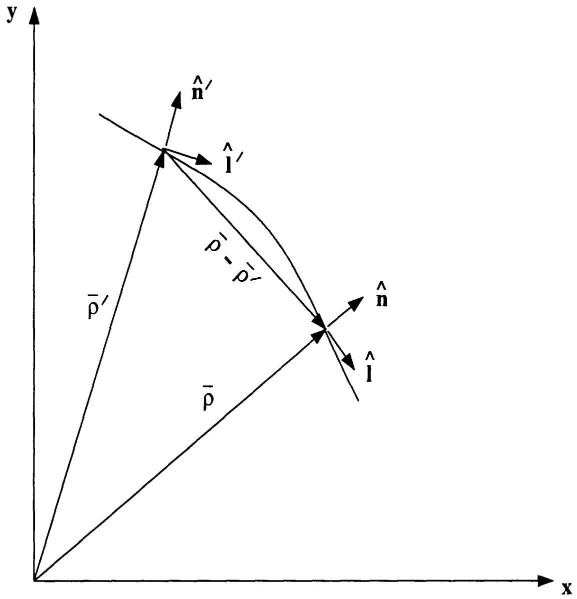

Figure 1: General two-dimensional scatterer configuration and associated notation

or 5-point Simpson's rule

$$\int_{a_j}^{b_j} f(x)dx = \frac{\delta_j}{90}(7f(1) + 32f(12) + 12f(2) + 32f(23) + 7f(3)) \tag{5}$$

for the elements adjacent to the self cell. The quantity δ_j is the cell length, $\delta_j = b_j - a_j$, and the argument 1,2,3 of the f function designate the start, center and the end points on the cell respectively. In (5) 12 and 23 represent points halfway between 1 and 2, and 2 and 3, respectively.

In a typical MoM code, all the matrix elements would be computed using (2) and (3). However in the Fast Far Field Algorithm (FAFFA) a different procedure is adopted. In the first step, the N unknowns are divided into groups comprised of adjacent elements, and with the premise that the number of unknowns in each group are roughly the same. The grouping process is depicted in Fig 2. If each group has M unknowns, the number of groups would thus be $\frac{N}{M}$. Depending on the distance between the groups, they can be classified as being near or far from each other. If two groups are in the near field of each other, their interaction is calculated exactly by using (2) and (3). In the limiting case of very large bodies, the number of near field groups for each test group is much less than the total number of groups and thus the near field operation count for the matrix-vector product is of *O(NM)*. The near field matrix elements are stored in compact form in the row-indexed sparse storage mode [7]. For far field groups, the Hankel function in the kernel is approximated as

$$H_0^{(1)}(k_0\rho_{ji}) \sim \sqrt{\frac{2}{\pi \mathrm{i}}}\frac{e^{\mathrm{i}k_0\rho_{ji}}}{\sqrt{k_0\rho_{ji}}}, \quad k_0\rho_{ji} \gg 1 \tag{6}$$

where $\rho_{ji} = |\overline{\rho_i} - \overline{\rho_j}|$. We also observe that the distance vector can be written as

$$\overline{\rho_{ji}} = \overline{\rho_{jl'}} + \overline{\rho_{l'l}} + \overline{\rho_{li}} \tag{7}$$

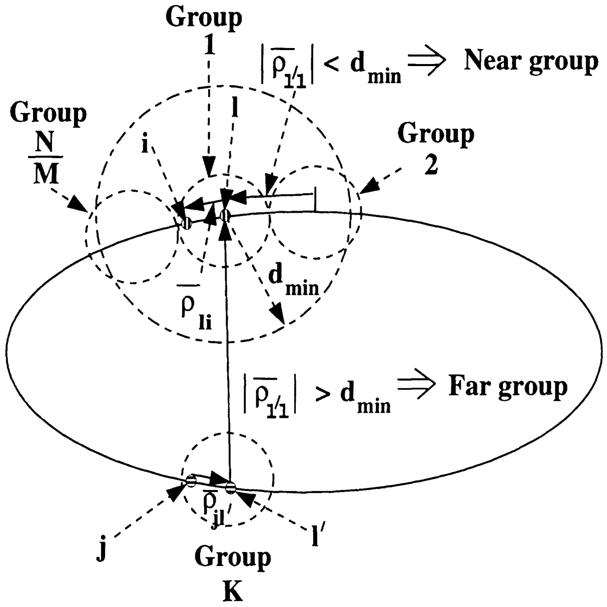
Group 1
$|\overline{\rho_{l'l}}| < d_{min} \Rightarrow$ Near group
Group $\frac{N}{M}$
i
l
Group 2
d_{min}
$\overline{\rho_{li}}$
$|\overline{\rho_{l'l}}| > d_{min} \Rightarrow$ Far group
$\overline{\rho_{jl'}}$
j
l'
Group K

Figure 2: The process of grouping unknowns - two groups are in the near field of each other if the distance between their centers $\rho_{l'l}$ is less than d_{min}

where $\overline{\rho_{jl'}}$ is the distance vector from the jth source element contained in the l'th group to the center of the group, $\overline{\rho_{l'l}}$ is the distance vector from the center of the l'th group to the center of the lth group and $\overline{\rho_{li}}$ is the distance vector from the center of the lth group to the ith test element. For far field groups, since $|\overline{\rho_{l'l}}| \gg |\overline{\rho_{jl'}}|$ and $|\overline{\rho_{l'l}}| \gg |\overline{\rho_{li}}|$, ρ_{ji} can be approximated as

$$\rho_{ji} \sim \rho_{l'l} + \hat{\rho_{l'l}} \cdot \overline{\rho_{li}} + \hat{\rho_{l'l}} \cdot \overline{\rho_{jl'}} \tag{8}$$

On substituting this into (6) we get

$$H_0^{(1)}(k_0\rho_{ji}) \sim e^{\mathrm{i}k_0\hat{\rho_{l'l}}\cdot\overline{\rho_{li}}}\sqrt{\frac{2}{\pi\mathrm{i}}}\frac{e^{\mathrm{i}k_0\rho_{l'l}}}{\sqrt{k_0\rho_{l'l}}}e^{\mathrm{i}k_0\hat{\rho_{l'l}}\cdot\overline{\rho_{jl'}}} \tag{9}$$

The decoupling of test-source element interactions in the kernel as in (9) enables the computation of the matrix-vector product for far-field groups with a reduced operation count as detailed in the following sequence.

1. For each test group, the aggregation of source elements in a single source group involves M operations, corresponding to the number of elements in the source group. The aggregation operation corresponds to

$$b_{l'l} = \sum_{j=1}^{M} J_j e^{\mathrm{i}k_0\hat{\rho_{l'l}}\cdot\overline{\rho_{jl'}}} \tag{10}$$

2. Since the above aggregation operation needs to be done for all source groups the operation count becomes $O(\frac{N}{M}M) \sim O(N)$, where $\frac{N}{M}$ represents the total number of groups. Also this operation, being dependent only on the test *group* rather than the test *element*, needs to be repeated for $\frac{N}{M}$ test groups leading to a total operation count of $O(\frac{N^2}{M})$ for aggregation.

3. The next step would be a translation operation corresponding to

$$c_{l'l} = \sqrt{\frac{2}{\pi \mathbf{i}}} \frac{e^{\mathbf{i}k_0 \rho_{l'l}}}{\sqrt{k_0 \rho_{l'l}}} b_{l'l} \tag{11}$$

Since this needs to be done only at the group level, it involves $O(\frac{N^2}{M^2})$ operations for all possible test and source group combinations and is the least computationally intensive step.

4. The final step in the sequence would be the process of disaggregation corresponding to the operation

$$I_{li} = \sum_{l'=1}^{N/M} c_{l'l} e^{\mathbf{i}k_0 \hat{\rho_{l'l}} \cdot \overline{\rho_{li}}} \tag{12}$$

Conceptually, this process is the converse of aggregation. Since this operation involves only the source *group* instead of the source *element* it needs to be done for each source group thus implying an $O(\frac{N}{M})$ operation to generate a single row of the matrix-vector product. To generate M rows corresponding to a test group the operation count would be $O(N)$. With $\frac{N}{M}$ test groups, the operation count would be of $O(\frac{N^2}{M})$.

5. The near field operation count being of $O(NM)$ and the far field being $O(\frac{N^2}{M})$ gives a total operation count of

$$Op.count \sim C_1 NM + C_2 \frac{N^2}{M} \tag{13}$$

Typically, we can set the elements in each group, $M = \sqrt{N}$ and as a result the total operation count is $O \sim N^{1.5}$.

2.2 TE incidence

For TE incidence, the MFIE is given by

$$Y_0 E_l^i(l) = R(l) J_l(l) + \lim_{\rho \to \rho'} \frac{1}{4} \frac{\partial}{\partial n} \int_C J_l(l') \left(\hat{n'} \cdot \frac{(\overline{\rho} - \overline{\rho'})}{\left| \overline{\rho} - \overline{\rho'} \right|} \right) H_1^{(1)} \left(k_0 \left| \overline{\rho} - \overline{\rho'} \right| \right) dl' \tag{14}$$

Again a pulse-basis point matching discretization of this integral equation results in an impedance matrix whose elements are given by

$$\begin{aligned} Z_{ij} &= R_i + \frac{\pi}{4} \frac{\delta_j}{\lambda} + \frac{\mathbf{i}}{2} \left\{ \frac{2}{\pi^2 \frac{\delta_j}{\lambda}} + \frac{\delta_j}{\lambda} (\ln \frac{\delta_j}{\lambda} - 0.47120163) \right\}, \quad i = j \\ &= \frac{\pi}{2} \int_{C_j} (\hat{n}_j \cdot \hat{n}_i) H_0^{(1)} (k_0 |\overline{\rho_j} - \overline{\rho_i}|) \, d\left(\frac{\delta_j}{\lambda} \right) \\ &+ \frac{1}{4} \left(\hat{l}_i \cdot \frac{(\overline{\rho} - \overline{\rho'})}{\left| \overline{\rho} - \overline{\rho'} \right|} \right) H_1^{(1)} \left(k_0 \left| \overline{\rho} - \overline{\rho'} \right| \right) \Bigg|_{a_i}^{b_j} \quad i \neq j \end{aligned} \tag{15}$$

As before, the integral in (15) is evaluated using (4) and (5) for the near group elements. When $|\overline{\rho_j} - \overline{\rho_i}| \gg \delta$, the far zone approximation (6) of $H_0^{(1)}$ is introduced and

$$H_1^{(1)}(k_0\rho_{ji}) \sim \sqrt{\frac{2}{\pi \mathrm{i}}} \frac{1}{\mathrm{i}} \frac{e^{\mathrm{i}k_0\rho_{ji}}}{\sqrt{k_0\rho_{ji}}}, \quad k_0\rho_{ji} \gg 1 \tag{16}$$

The implementation of the FAFFA for H-pol parallels that for the E-pol, with the only difference that the element normals in (15) are replaced by the group normals.

3 Results

3.1 Geometry and objective for error analysis

The objective of this work is to implement the FAFFA and study the parameters effecting the CPU performance and accuracy. Specifically, we examined the issues and accuracy on the basis of

- the near-group radius,d_{min}
- the sampling rate
- group size, and
- memory requirements

Our benchmark for accuracy was the RMS error given by

$$ERROR_{RMS} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}[RCS_{EX}(i) - RCS_{FF}(i)]^2} \tag{17}$$

where RCS_{EX} denotes the exact radar cross sections as computed by the standard moment method approach without grouping and RCS_{FF} is the RCS calculated using the FAFFA; M being the number of points at which the RCS is computed. We note that this error formula is among several that have been considered, but has been found to represent a reasonable measure of the accuracy. It has been employed by Schuh and Woo [8] for a study on the accuracy of RCS computations by various codes. In the following we will show error curves as a function of the aforementioned parameters for two cylinders. One is a PEC isosceles triangle having a base 2.5λ in length and a height of 17.95λ. The other is a rectangular PEC cylinder $25\lambda \times 4\lambda$ in size. These geometries (see Figure 3) are analyzed at grazing (0 degrees) incidence since this excitation is most stressing for RCS calculations, particularly so for the triangular cylinder. In fact, the two geometries were selected to have different characteristics in the backscatter region.

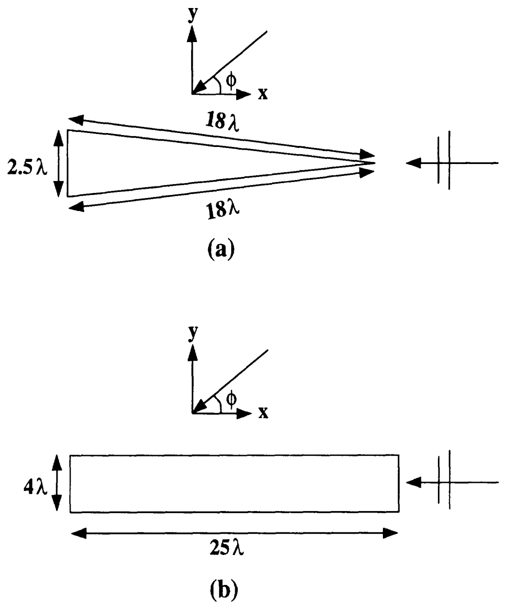

Figure 3: (a) An isosceles triangular metallic cylinder - 17.95λ high with a 2.5λ base (b) A rectangular metallic cylinder - 25λ × 4λ

3.2 RMS Error vs. near-group separation distance and sampling rate

First we look at the RMS error of the bistatic RCS pattern as a function of the near-group radius, d_{min}. Such error curves are shown in Figures 4 and 5 for the isosceles and rectangular cylinders, respectively. Each of these figures depicts several curves, each corresponding to a different sampling rate ($\lambda/10$, $\lambda/15$, $\lambda/20$, $\lambda/30$ or $\lambda/40$ as designated in the figure). Surprisingly, the error curves for each geometry give rise to the same conclusion. Basically, Figures 4 and 5 demonstrate that the sampling rate has a profound effect on the accuracy of the solution and the value of the error is strongly dependent on the near-group radius. For example, if d_{min} is set to 1.7λ, a tessellation rate of 10 segments per wavelength yields an RMS error of 4.79 dB whereas a tessellation rate of 30 segments per wavelength leads to an RMS error down to 1.19 dB. As expected, a higher sampling rate leads to smaller errors. However, these errors are different and much larger than those resulting from a standard moment method implementation. Specifically, a tessellation rate of 10 segments per wavelength yields an RMS error of 0.1615 dB when employing the standard unreduced moment method solution but increases to 4.79 dB when FAFFA is employed with $d_{min} = 1.7\lambda$. As depicted in Figures 4 and 5, the values of d_{min} plays an equally important role in achieving a given RMS error. For example, when the segment length is $\lambda/10$, the error is over 4 dB when $d_{min} = 2\lambda$ but the error decreases to less than 2 dB when d_{min} reaches 3λ for the triangular cylinder and 5λ for the rectangular cylinder. This trend is, of course, logical since for larger values of d_{min} the FAFFA implementation resembles more and more the unreduced moment method.

Having found such a strong dependence of the RMS error on the tessellation rate and the near-group distance, to proceed further with our study, it is essential to look at what may be an acceptable RMS error. This error value can then be used to determine acceptable relationships between sampling rates and d_{min} on the basis of a certain RMS error. We therefore refer to Figure 6 which shows the bistatic RCS pattern of the isosceles triangle as computed using FAFFA for different sampling rates. The shown (exact) pattern has a dynamic range of 30 dB and we observe that an RMS error over 4 dB is associated with large and unacceptable RCS deviations (over 10 dB) from the reference RCS values. However, when the RMS error is about 1 dB or so, it is clear that the FAFFA results have only small deviations from the exact curve which is actually tracked very well by the FAFFA results even in low RCS regions. Consequently, we may select the RMS error value of 1 dB as the threshold level in determining whether a given choice of FAFFA parameters lead to acceptable accuracy or not.

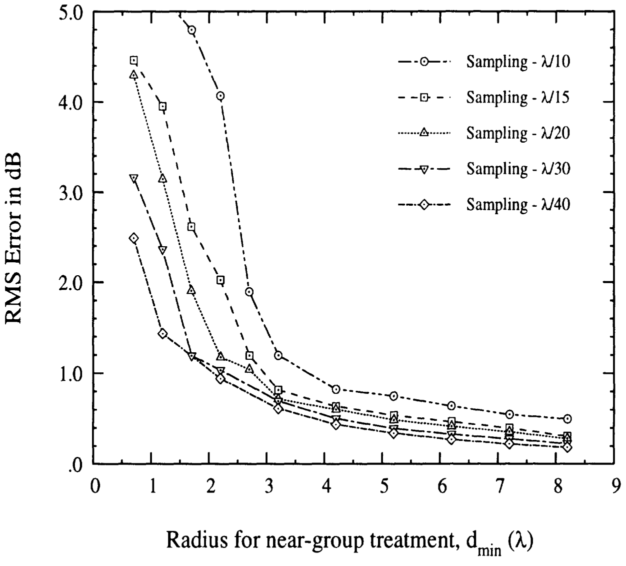

Figure 4: Error curves for the FAFFA for a isosceles triangular metallic cylinder - 17.95λ high with a 2.5λ base

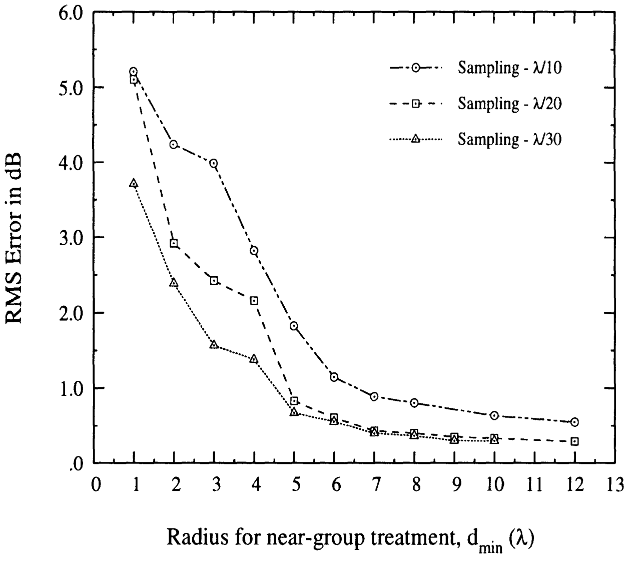

Figure 5: Error curves for the FAFFA for a rectangular pec cylinder - 25λ × 4λ

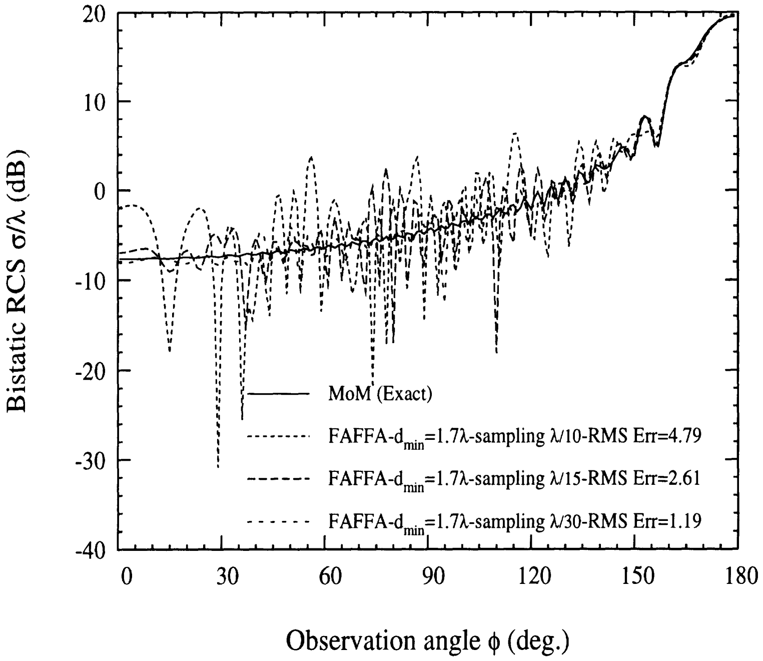

Figure 6: Comparison of bistatic patterns of an isosceles triangular pec cylinder - 17.95λ high with a 2.5λ base computed using the FAFFA at different sampling rates

3.3 CPU requirements

Returning now to Figures 4 and 5, we observe that an RMS error of 1 dB or less is obtained for the triangular cylinder if $d_{min} > 4\lambda$ when the sampling rate is 10 segments per wavelength, but d_{min} must be at least 7λ for the rectangular cylinder with the same sampling rate. Unfortunately, these values for d_{min} are much larger than 1λ which is the value often suggested by other users of the fast multipole method and its variations. It is of course important to keep the near-group radius as small as possible for CPU and memory reduction. However, the sampling rate must be increased to 30 segments per wavelength (for the isosceles triangle) if we are to reduce d_{min} to 2λ, but clearly the increased sampling is counter to the benefits from a smaller near-group distance. Thus in Figures 7 and 8, we show curves of the CPU time as a function of the near-group distance for different sampling rates. Figure 7 refers to the triangular cylinder whereas Figure 8 corresponds to the rectangular cylinder, and both show a similar dependence on d_{min} and the sampling rate. For both cylinders the CPU time increases almost linearly with d_{min} and is, of course, higher for denser tessellations. These curves can be used in conjunction with those in Figures 4 and 5 to obtain the CPU time to determine d_{min} for a given RMS error and sampling rate. Alternatively, Figures 4 and 5 can be consolidated with the data in Figures 7 and 8 to obtain a new set of curves which explicitly give the CPU time as a function of a desired/given RMS error for the selected sampling rate. These CPU curves as a function of a specified RMS error are shown in Figures 9 and 10, corresponding to the triangular and rectangular cylinders, respectively. The trends in each of these Figures are identical, demonstrating that the dependence between CPU time, sampling, near group-distance and RMS error are likely the same regardless of geometry considerations.

Basically, Figures 9 and 10 show that the CPU time increases quadratically with sampling rate. Clearly, a lower sampling rate is more attractive in terms of CPU time since it leads to smaller systems and therefore faster convergence rate. That is, it is better to use a larger d_{min} with lower sampling rates rather than a higher sampling rate with a lower near-group window radius. For example, to achieve an RMS error of 1 dB for the triangular cylinder, the CPU time is about 10 sec when using tessellation of 10 segments per wavelength and from Figure 4 the corresponding d_{min} is 3.75λ. However, to achieve the same error using a tessellation of 30 segments per wavelength, the corresponding CPU time is more than five times higher but the required near-group window radius is less than 2.5 wavelengths. It may seem that the higher number of degrees of freedom associated with higher segmentation rates leads to unequivocally higher memory requirements as well. However, this is not always the case with the FAFFA implementation because the value of d_{min} plays a major role on the bandwidth of the dense section of the matrix. For example, in the case of the rectangular cylinder, sampling at $\lambda/10$ would imply 580 unknowns while sampling at $\lambda/30$ would imply 1740 unknowns. To achieve a RMS error of 3.75 dB, it would require a near group window of 3.5λ for sampling at $\lambda/10$ and a window of 1λ for sampling at $\lambda/30$. Execution time considerations are of the order of 20 seconds for the first case and 75 seconds for the latter case. An examination of the grouping of the unknowns would reveal that the memory requirement for the latter case is smaller. Sampling at $\lambda/30$ and employing a near group radius of 1λ would mean a storage requirement of $\sim$ 42 elements per row. Sampling at $\lambda/10$ and employing a near group radius of 3.5λ would mean a storage requirement of $\sim$ 72 elements per row, which is higher than the previous case. Consequently, although Figures 9 and 10 suggest that a lower sampling rate has the least CPU requirements in satisfying a given error criterion, the memory requirements will be higher when compared to those associated with a higher sampling error and the same error criterion. Memory considerations will of course be an issue for very large simulations and the choice of using a high sampling or not will depend on the available computing resources.

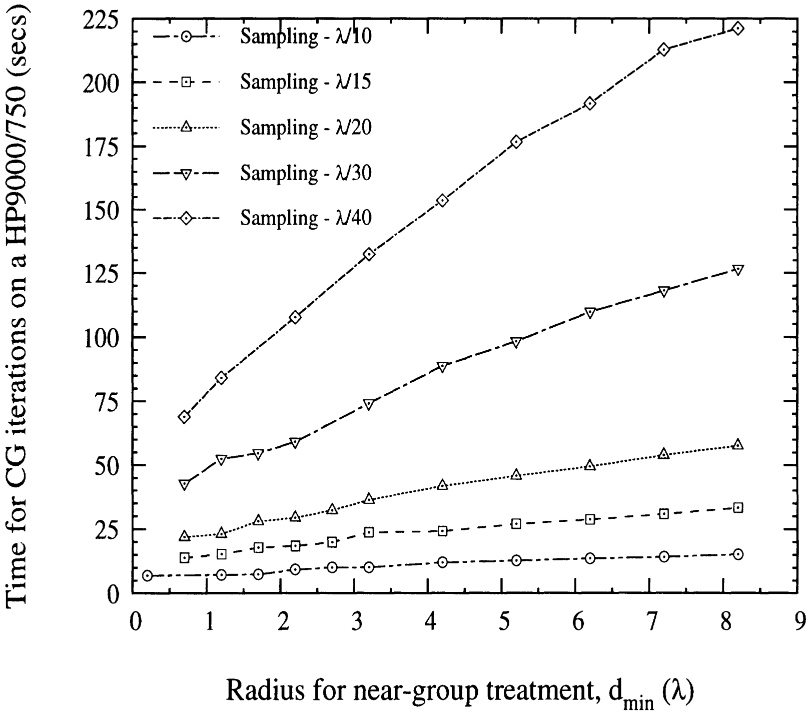

Figure 7: Time curves for the FAFFA for a isosceles triangular pec cylinder - 17.95λ high with a 2.5λ base

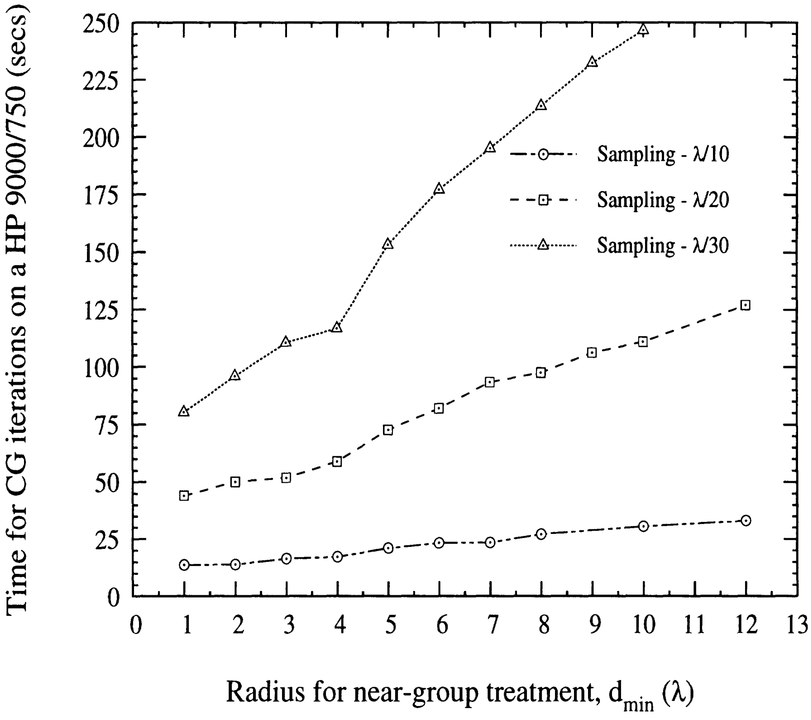

Figure 8: Time curves for the FAFFA for a rectangular pec cylinder - 25λ × 4λ

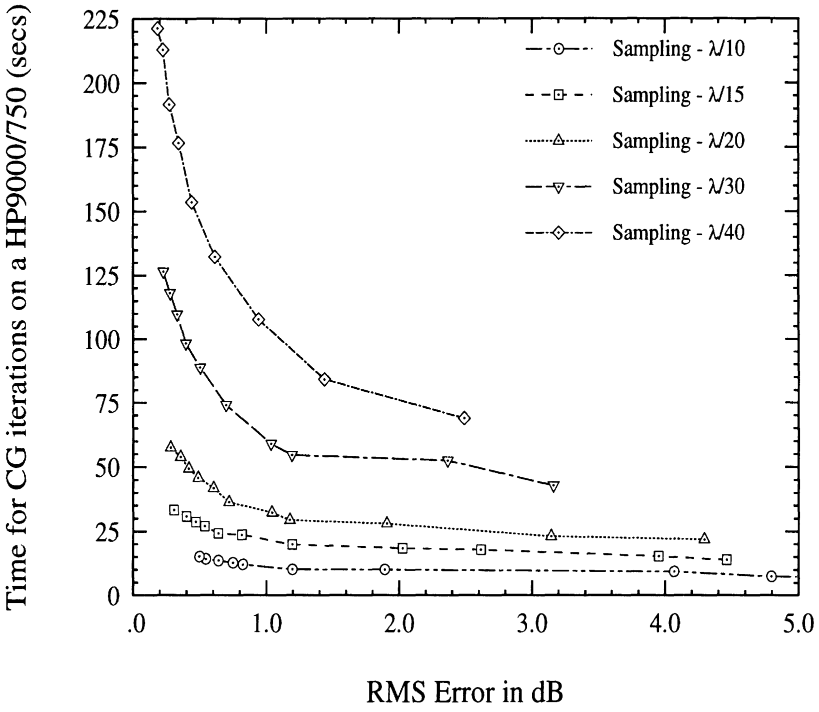

Figure 9: Time - error curves for the FAFFA for a isosceles triangular pec cylinder - 17.95λ high with a 2.5λ base

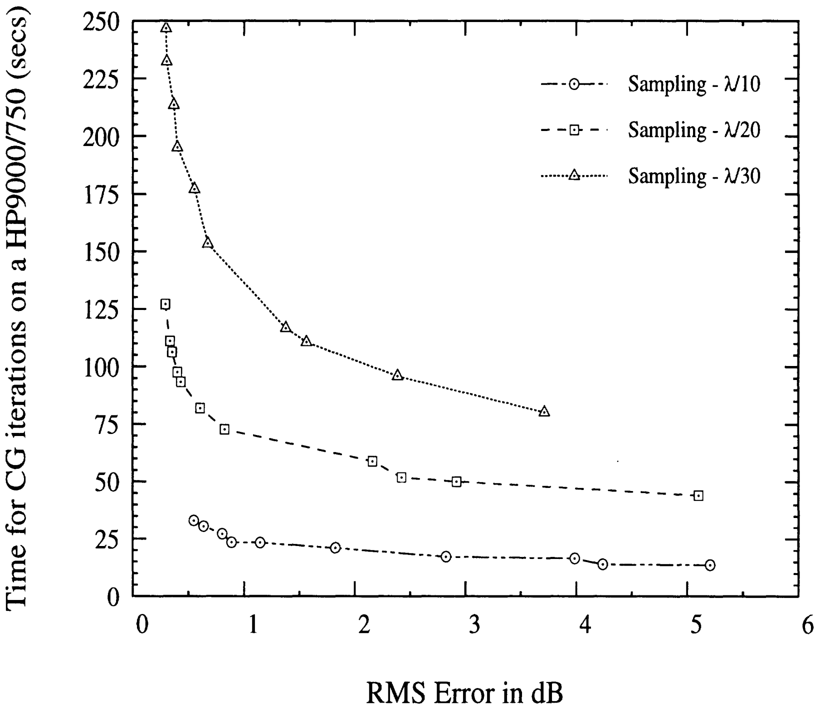

Figure 10: Time - error curves for the FAFFA for a rectangular pec cylinder - 25λ × 4λ

4 Conclusion

In this report we looked at the CPU requirements of the FAFFA as a function of the different parameters affecting its performance, including the near-group window radius, sampling rate and error. We presented curves which show the required sampling rates, near-group window radius and CPU time as a function of a given error criterion. Based on these curves, one concludes that

- The error is strongly dependent on d_{min} and the sampling rate. This error dependence on the sampling is particularly inherent to the FAFFA solution method and should not be confused with similar errors associated with the unreduced moment method implementations. With higher sampling, the spatial extent of each group reduces and thus the process of aggregation and disaggregation of group elements into group centers induces less error.
- In lieu of CPU time efficiency, it is best to employ lower sampling and larger near-group window radius.
- Stringent memory requirements behooves the adoption of high sampling and small near group radius but at the expense of high execution time.

References

[1] R.F. Harrington, *Field Computation by moment methods*, IEEE press, New York, 1993.

[2] F.X. Canning, Fast integral equation solutions using GTD-like matrices, *Radio Sc.*, 29(4):993-1008, Jul-Aug 1994.

[3] J.L. Volakis and S.V. Krestianinov, ISOMOM: A new method for reducing storage and execution time of moment-method solutions in electromagnetics, *Microwave and Optical Techn. Letters*, 8(4)184-186, March 1995.

[4] N. Engheta, W.D. Murphy, V.Rokhlin and M.S. Vassiliou, The fast multipole method for electromagnetic scattering problems, *IEEE Trans. Antennas Propagat.*, 40(4):634-641, June 1992.

[5] J.M. Song and W.C. Chew, Fast multipole method solution of combined field integral equation, *ACES Digest*, pp 629-636, March 1995.

[6] C.C Lu and W.C. Chew, Fast Far Field Approximation for Calculating the RCS of Large Objects, *ACES Digest*, pp 576-583, March 1995.

[7] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes*, pp 71-76, Cambridge University Press, 1992.

[8] M.J. Schuh and A.C. Woo, Code Scaling, *To appear in IEEE Antennas and Propagat. Magazine*, August 1995.