The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription

Ronald Coifman[1], Vladimir Rokhlin[1], and Stephen Wandzura[2]

[1]Fast Mathematical Algorithms and Hardware Corporation
1020 Sherman Avenue
Hamden, CT 06514

and

[2]Hughes Research Labs
3011 Malibu Canyon Road
Malibu, CA 90265

1. Introduction

The purpose of this article is to give a practical and complete, but not rigorous, exposition of the Fast Multipole Method (FMM). The aim is to give the computational physicist or engineer a sufficiently clear understanding of the method that he or she will be able to implement it with a minimum of difficulty. For mathematical background and rigor, we refer the reader to Rokhlin's papers [1, 2].

The FMM provides an efficient mechanism for the numerical convolution of the Green's function for the Helmholtz equation with a source distribution. It can be used to radically accelerate the iterative solution of boundary-integral equations. In the simple single-stage form presented here, it reduces the computational complexity of the convolution from $O(N^2)$ to $O(N^{3/2})$, where N is the dimensionality of the problem's discretization. By implementing a multistage FMM [1,2], the complexity can be further reduced to $O(N \log N)$. However, even for problems that have an order of magnitude more variables than those currently tractable using dense-matrix techniques ($N \approx 10^5$), we estimate that the performance of the single-stage algorithm should be near optimal.

Our development is given in terms of the method of moments [3,4] (MoM), rather than the Nyström method [5]. We do this because

- Electrical engineers are more familiar with the MoM, and may therefore be more comfortable with the development.
- The prescription we present is sufficiently simple that it can be easily retrofitted to existing MoM codes.
- When used in the MoM, detailed comparisons to verify that results are identical to dense-matrix techniques are immediately available.
- We avoid all questions of singularity subtraction, as it is required only for matrix elements representing nearby interactions, and the computation of these is unchanged when the FMM is employed.
- The presentation demonstrates the independence of the FMM from the choice of discretization method, boundary-surface model, basis functions, etc.

The reader is cautioned not to interpret our choice of presentation as representing a preference toward the MoM. On the contrary, we think that the Nyström method is the appropriate tool for efficient and accurate boundary-integral-equation solvers.

For the purposes of demonstration, we first consider the MoM for the scalar wave equation, with Dirichlet boundary conditions on the surface of a scatterer. This is done for notational convenience only, the (naive) equivalent application to the electric-field integral equation (EFIE) being straightforward. (One can simply apply the scalar prescription to each Cartesian component of the vector expansion functions, and to their divergences; a more efficient method is described in Section 5.)

If the structure of this article seems somewhat confusing at an initial reading, it is because some considerations are intentionally delayed. We hope that the reasons for this become clear upon subsequent readings. In Section 2, we define notation, introduce the discretization of the scattering problem, relate the FMM to a more familiar fast algorithm, and introduce the fundamental analytic apparatus of the FMM. A detailed prescription for FMM implementation, except for the choice of some important parameters of the algorithm, is given in Section 3. After the structure of the method is exhibited, these parameters (the number of terms used in the multipole expansion, and the directions at which far-field quantities are tabulated) are analyzed in Section 4. The algorithm for the scalar problem then having been being completely defined, we exhibit the minor modifications necessary for application to vector (electromagnetic) scattering in Section 5. Before concluding, a physical interpretation of the analysis behind the FMM is given in Section 6.

2. Basics

2.1 Notation

Vectors in three-dimensional space are represented by boldface type ($\mathbf{x}$). The magnitude of a vector $\mathbf{x}$ is written as $x \equiv |\mathbf{x}|$, unit vectors are written as $\hat{x} \equiv \mathbf{x} / x$, and integrals over the unit sphere are written as $\int d^2\hat{x}$. The imaginary unit is denoted by i.

 1045-9243/93/$03.00©1993 IEEE

2.2 Time-independent scattering and the Method of Moments

A scattering problem [6, 7] can be defined by the scalar wave equation

$$\left(\nabla^2 + k^2\right)\psi = 0, \tag{1}$$

a Dirichlet boundary condition

$$\psi(\mathbf{x}) = 0; \quad \mathbf{x} \text{ on } S, \tag{2}$$

on the surface, S, of a bounded scatterer, and a radiation boundary condition. The method of moments [8] provides a discretization of the first-kind integral equation associated with this problem, giving a set of linear equations with a dense coefficient (impedance) matrix:

$$Z_{nn'} = -i\int_S d^2\mathbf{x}\int_S d^2\mathbf{x}' f_n(\mathbf{x})\frac{e^{ik|\mathbf{x}-\mathbf{x}'|}}{4\pi|\mathbf{x}-\mathbf{x}'|}f_{n'}(\mathbf{x}'). \tag{3}$$

We assume that the basis functions, f_n, are real, and supported on local subdomains. The FMM provides a prescription for the rapid computation of the matrix-vector product

$$B_n = \sum_{n'=1}^{N} Z_{nn'}I_{n'}, \tag{4}$$

for an arbitrary vector I. This rapid computation can then be used in an iterative (e.g., conjugate-gradient) solution of the discretized integral equation $Z \cdot I = V$, where, for an incident wave with wave vector $\mathbf{k}$,

$$V_n(k) = \int_S d^2\mathbf{x} f_n(\mathbf{x})e^{i\mathbf{k}\cdot\mathbf{x}}. \tag{5}$$

Note that we have chosen to use the same functions for expansion and testing (the Galerkin method). Not only does this simplify the development somewhat, but it also results in superconvergence of the scattering amplitude [9, 10].

2.3 Comparison with the Fast Fourier Transform

A discrete Fourier transform consists of multiplication by a dense $N \times N$ matrix F, with matrix elements

$$F_{kl} = \exp\frac{2\pi ikl}{N}. \tag{6}$$

The fast Fourier transform (FFT) works by using algebraic properties of F to construct a sparse factorization,

$$F = F^{(1)}F^{(2)}\ldots F^{(\log_2 N)}, \tag{7}$$

and applying the sparse factors, $F^{(\alpha)}$, one by one to the vector to be transformed, in lieu of a single multiplication by the matrix F. Because each of the factors has only $O(N)$ non-zero elements, this results in an algorithm that requires $O(N\log N)$ operations. The single-stage FMM works by a similar decomposition of the matrix Z:

$$Z = Z' + VTV^{\dagger}, \tag{8}$$

where Z', V, and T are all sparse. As described in detail in this article, this allows computation of the product of Z with an arbitrary vector (corresponding physically to the determination of the fields radiated by a known source distribution), with $O\left(N^{3/2}\right)$ operations. The complexity can be further reduced to $N^{4/3}, N^{5/4}, \ldots$, by recursive decomposition of Z' and V:

$$Z' = Z'' + V'T'V'^{\dagger} \tag{9}$$

$$V = V'S. \tag{10}$$

This is entirely analogous to the FFT: if one factors F into only two factors (independent of N), the result would be an $O\left(N^{3/2}\right)$ algorithm. We do not exhibit the details of the multi-stage FMM in this article.

In contrast to the FFT, the FMM decomposition is made possible by analytic rather than algebraic properties of the linear operator. Thus, while the FFT factorization is exact, the FMM decomposition is approximate. However, this does *not* constitute a practical limitation, as it is easy to control the FMM to achieve any desired level of precision (all the way to machine precision).

2.4 Identities

The FMM, as presented here, rests on two elementary identities. They, or formulas from which they may be easily derived, are found in many texts and handbooks on mathematical methods, such as Arfken [11] and Abramowitz and Stegun [12]. The first, an expansion of the kernel in the integral, Equation (3), for the impedance-matrix elements, is a form of Gegenbauer's addition theorem,

$$\frac{e^{ik|\mathbf{X}+\mathbf{d}|}}{|\mathbf{X}+\mathbf{d}|} = ik\sum_{l=0}^{\infty}(-1)^l(2l+1)j_l(kd)h_l^{(1)}(kX)P_l\left(\hat{d}\cdot\hat{X}\right), \tag{11}$$

where j_l is a spherical Bessel function of the first kind, $h_l^{(1)}$ is a spherical Hankel function of the first kind, P_l is a Legendre polynomial, and $d < X$. When using this expansion to compute the field at $\mathbf{x}$ from a source at $\mathbf{x}'$, $\mathbf{X}$ will be chosen to be close to $\mathbf{x}-\mathbf{x}'$, so that d will be small. This relationship of the various vectors is sketched in Figure 1. The special functions are as defined in [12]. The second is an expansion of the product j_lP_l in propagating plane waves:

$$4\pi i^l j_l(kd)P_l\left(\hat{d}\cdot\hat{X}\right) = \int d^2\hat{k}e^{i\mathbf{k}\cdot\mathbf{d}}P_l\left(\hat{k}\cdot\hat{X}\right). \tag{12}$$

Substituting Equation (12) into Equation (11), we get

$$\frac{e^{ik|\mathbf{X}+\mathbf{d}|}}{|\mathbf{X}+\mathbf{d}|} = \frac{ik}{4\pi}\int d^2\hat{k}e^{i\mathbf{k}\cdot\mathbf{d}}\sum_{l=0} i^l(2l+1)h_l^{(1)}(kX)P_l\left(\hat{k}\cdot\hat{X}\right), \tag{13}$$

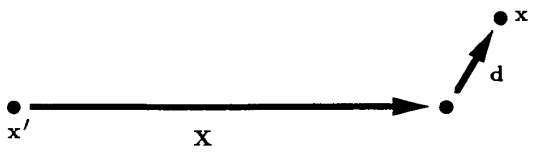


Figure 1. The basic geometry, illustrating the relationship between the locations $\mathbf{x}, \mathbf{x}'$ and the displacements $\mathbf{X}, \mathbf{x}$.

where we have performed the illegitimate but expedient interchange of summation and integration. The key point is that we intend to precompute the function

$$\mathcal{T}_L(\kappa,\cos\theta) \equiv \sum_{l=0}^{L} i^l(2l+1)h_l^{(1)}(\kappa)P_l(\cos\theta), \tag{14}$$

for various values of κ. This is not a function in the $L\to\infty$, but that need not concern us, as we obviously intend to truncate the sum in numerical practice. The number of kept terms, $L+1$, will depend on the maximum allowed value of kd, as well as the desired accuracy. The choice of L is discussed in Section 4. It suffices, for the present, to note that, in order to obtain accuracy from Equation (11), it must be slightly greater than κD, where D is the maximum value of d for which the expansion will be used. Ignoring this question for now (except for noting that the required number of terms becomes small as $D\to 0$), we have

$$\frac{e^{ik|\mathbf{X}+\mathbf{d}|}}{|\mathbf{X}+\mathbf{d}|} \approx \frac{ik}{4\pi}\int d^2\hat{k}e^{i\mathbf{k}\cdot\mathbf{d}}\mathcal{T}_L\left(kX,\hat{k}\bullet\hat{X}\right) . \tag{15}$$

Using this, the impedance-matrix element, Equation (3), is given by

$$Z_{nn'} \approx \frac{k}{(4\pi)^2}\int_S d^2\mathbf{x}f_n(\mathbf{x})\int_S d^2\mathbf{x}'f_{n'}(\mathbf{x}')\int d^2\hat{k}e^{i\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}'-\mathbf{X})}\mathcal{T}_L\left(kX,\hat{k}\bullet\hat{X}\right) \tag{16}$$

In infinite-precision arithmetic, and in the limit of large L, this result would be independent of the choice of $\mathbf{X}$ (for $X>|\mathbf{x}-\mathbf{x}'-\mathbf{X}|$). In practice, one chooses $\mathbf{X}$ to make $\mathbf{x}-\mathbf{x}'-\mathbf{X}$ relatively small, so that excellent accuracy can be obtained with a modest value of L. (That this can be done by the grouping scheme described below is a consequence of the local support of the basis functions.) Notice that Equation (16) gives the impedance-matrix element (for well-separated interactions) in terms of the Fourier transforms *with wave number k* of the basis functions, i.e. the basis functions' far fields. The acceleration provided by the FMM comes from the fact that these far fields can be grouped together *before* the integral over $\hat{k}$ is performed.

3. Algorithmic prescription

3.1 Setup

1. Divide the N basis functions into M localized groups, labeled by an index m, each supporting about N/M basis functions. (For now, M is a free parameter. Later it will be seen that the best choice will be $M\sim\sqrt{N}$.) Thus, establish a correspondence between the basis-function index, n, and a pair of indices (m,α), where α labels the particular basis function within the mth group. Denote the center of the smallest sphere enclosing each group as $\mathbf{X}_m$. The grouping and index correspondence is shown, for a simple case, in Figure 2.

2. For group pairs (m,m') that contain "nearby" basis functions [defined for now as those whose regions of support are separated by a distance comparable to or smaller than a wavelength, $2\pi/k$, so that Equation (16) is valid], construct the sparse matrix Z', with matrix elements

$$Z'_{m\alpha m'\alpha'} = Z_{n(m,\alpha)n'(m',\alpha')}, \tag{17}$$

by direct numerical computation of the matrix elements, Equation (3). For all other pairs, $Z'_{m\alpha m'\alpha'}=0$.

This part of the matrix computation is identical to what is conventionally done. All matrix elements, the computation of which requires subtraction of singularities, belong to Z'. If the large-N limit is taken with a fixed discretization interval and nearness criterion, this step would require $O(N)$ computations. In Section 4, we define nearby regions precisely, and it turns out that their volume increases as $\sqrt{N}$, so that this step requires $O\left(N^{3/2}\right)$ computations.

3. For K directions $\hat{k}$, compute the "excitation vectors" (Fourier transforms of the basis functions)

$$V_{m\alpha}\left(\hat{k}\right) = \int_S d^2\mathbf{x}\, e^{i\mathbf{k}\cdot(\mathbf{x}-\mathbf{X}_m)}f_{n(m,\alpha)}(\mathbf{x}), \tag{18}$$

where k is considered to be a parameter of the problem, not a variable. Because K needs to be chosen to give accurate numerical quadrature for all harmonics to some order $\propto L\sim kD$, $K\propto L^2\sim(kD)^2$, and because (from geometrical considerations) $kD\propto\sqrt{N/M}$, this step requires $O\left(N^2/M\right)$ computations.

4. For each pair (m,m') for which $Z'_{m\alpha m'\alpha'}=0$ (regions that are not nearby), compute the matrix elements

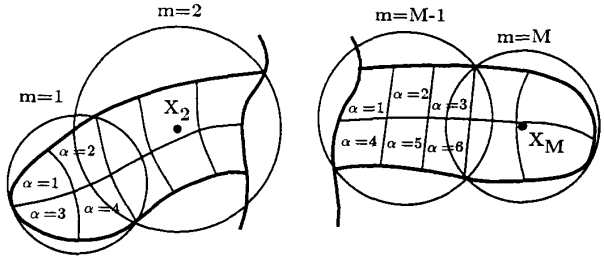


Figure 2. The grouping for a simple surface. It is assumed, for purposes of illustration only, that each patch supports only one basis function. The correspondence $n(m,\alpha)$ is abbreviated in Table 1.

Table 1. The abbreviated correspondence $n\leftrightarrow(m,\alpha)$ for the grouping shown in Figure 2.

m	α	$n(m,\alpha)$
1	1	1
1	2	2
1	3	3
1	4	4
2	1	5
2	2	6
2	3	7
...		

$$T_{mm'}(\hat{k}) = \frac{k}{(4\pi)^2}\sum_{l=0}^{L} i^l(2l+1)h_l^{(1)}(kX_{mm'})P_l(\hat{k}\cdot\hat{X}_{mm'}) \tag{19}$$

for the same K directions $\hat{k}$ as the previous step, where $L \propto \sqrt{K}$. If done in a naive manner, this computation requires $O(KLM^2 \sim M^{1/2}N^{3/2})$ operations. However, it can be accomplished more rapidly in a number of ways, the most elegant being the fast Legendre expansion [13].

3.2 Fast Matrix-Vector Multiplication

Rapid computation of the vector elements

$$B_{m\alpha} = \sum_{m'\alpha'} Z_{m\alpha m'\alpha'} I_{m'\alpha'} \tag{20}$$

is accomplished by the following steps:

1. Compute the KM quantities

$$s_m(\hat{k}) = \sum_{\alpha} V^*_{m\alpha}(\hat{k}) I_{m\alpha}, \tag{21}$$

which represent the far fields of each group m. This step requires $O(KN \sim N^2/M)$ operations.

2. Compute the KM quantities

$$g_m(\hat{k}) = \sum_{m'} T_{mm'}(\hat{k}) s_{m'}(\hat{k}). \tag{22}$$

These represent the Fourier components of the field in the neighborhood of group m, generated by the sources in the groups that are not nearby. This step requires $O(KM^2 \sim MN)$ operations.

3. Finally, compute

$$B_{m\alpha} = \sum_{m'\alpha'} Z'_{m\alpha m'\alpha'} I_{m'\alpha'} + \int d^2\hat{k} V_{m\alpha}(\hat{k}) g_m(\hat{k}). \tag{23}$$

The first term is the standard MoM computation of near interactions, and the second term gives the far interactions, in terms of the far fields generated by each group. This step requires $O(KN \sim N^2/M)$ operations.

Straightforward substitution of Equations (18), (19), (21), and (22) into Equation (23), and of Equations (14)-(16) into Equation (20), shows that the two expressions for the vector B, Equations (23) and (20), give equal results. Thus, computation of the vector B requires $aNM + bN^2/M$ operations, where a and b are machine and implementation dependent. The total operation count is minimized by choosing $M = \sqrt{bN/a}$; the result is an $O(N^{3/2})$ algorithm.

4. Required number of multipoles and directions

In this section:

- We show how to choose the summation limit in the transfer function $T_{mm'}(\hat{k})$, Equation (19), to achieve the desired accuracy (in the process, giving a precise definition of nearby regions).
- We discuss how to choose the K directions $\hat{k}$, for the tabulation of angular functions.

One must choose L large enough that the multipole expansion of the Green's function, Equation (11), converges to the desired accuracy. As a function of l, the Bessel functions $j_l(z)$ and $h_l^{(1)}(z)$ are of roughly constant magnitude for $l < z$. For $l > z$, $j_l(z)$ decays rapidly and $h_l^{(1)}(z)$ grows rapidly. While one must choose $L > kd = k|\mathbf{x} - \mathbf{x}' - \mathbf{X}_{mm'}|$ (so that the partial-wave expansion has converged), L cannot be taken to be much larger than $kX_{mm'}$, because the transfer function, Equation (14), will oscillate wildly, causing inaccuracies in the numerical angular integrations of Equations (15) and (23). This condition is a consequence of the interchange of summation and integration in Equation (13). An excellent semi-empirical fit to the number of multipoles required for single precision (32-bit reals) is

$$L_s(kD) = kD + 5\ln(kD + \pi), \tag{24}$$

where $D \geq 1/k$ is the maximum d which will be required (the "diameter" of the basis-function groups). For double precision (64-bit reals), our estimate is

$$L_d(kD) = kD + 10\ln(kD + \pi). \tag{25}$$

If the L dictated by the appropriate formula exceeds $kX_{mm'}$, then the groups are too close to use the FMM, and their interaction must be represented in the sparse matrix Z'.

The K directions $\hat{k}$, at which the angular functions are tabulated, must be sufficient to give a quadrature rule that is exact for all spherical harmonics of order $l < 2L$. A simple method [2] for accomplishing this is to pick polar angles θ such that they are zeros of $P_L(\cos\theta)$, and azimuthal angles ϕ to be $2L$ equally spaced points. Thus, for this choice of $\hat{k} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$, $K = 2L^2$. If more-efficient quadrature rules for the sphere (of the type described by McLaren [14]) are used, then $K \approx (4/3)L^2$. Since $kD \propto \sqrt{N/M}$, this justifies the assertion made in Section 3.1 that $K \propto N/M$.

5. Application to electromagnetic fields

In the solution of the electric-field integral equation, the impedance-matrix elements take the form [15]

$$Z_{nn'} = -i\sum_{j,j'=1}^{3}\int_s d^2\mathbf{x}\int_s d^2\mathbf{x}' f_{nj}(\mathbf{x}) G_{jj'}(\mathbf{x}-\mathbf{x}') f_{n'j'}(\mathbf{x}'), \tag{26}$$

where

$$G_{jj'}(\mathbf{x}-\mathbf{x}') = \left(\delta_{jj'} - \frac{1}{k^2}\frac{\partial}{\partial x_j}\frac{\partial}{\partial x'_{j'}}\right)\frac{e^{ik|\mathbf{x}-\mathbf{x}'|}}{4\pi|\mathbf{x}-\mathbf{x}'|}, \tag{27}$$

and the indices j, j' label Cartesian components. As implied in Section 1, one can integrate by parts, and simply use the scalar prescription, given above, on the three components of $\mathbf{f}$ and the scalar $\nabla \cdot \mathbf{f}$. This is not, however, the most economical procedure. By differentiating with respect to $\mathbf{d}$ under the integral in Equation (15), we get

$$G_{jj'}(\mathbf{X}+\mathbf{d}) \approx \frac{ik}{4\pi}\int d^2\hat{k}\left(\delta_{jj'}-\hat{k}_j\hat{k}_{j'}\right)e^{ik\cdot\mathbf{d}}\mathcal{T}_L\left(kX,\hat{k}\cdot\hat{X}\right) \quad (28)$$

Now it can be easily seen that the scalar prescription presented in Section 3 can be modified to an electromagnetic one, by promoting the quantities $V_{m\alpha}$, $s_m(\hat{k})$, and $g_m(\hat{k})$ to three-dimensional vectors, with

$$\mathbf{V}_{m\alpha}(\hat{k}) = \int d^2\mathbf{x}\, e^{i\mathbf{k}\cdot\mathbf{x}}\left[\mathbf{f}_{n(m,\alpha)}(\mathbf{x}) - \hat{k}\,\hat{k}\cdot\mathbf{f}_{n(m,\alpha)}(\mathbf{x})\right], \quad (29)$$

and using a dot product in the $\int d^2\hat{k}$ term of $B_{m\alpha}$, Equation (23). This method can be implemented using about half the storage of the four-fold use of the scalar formula, because the vector $\mathbf{V}_{m\alpha}$ has only two independent components: $\left[\hat{k}\cdot\mathbf{V}(\hat{k})=0\right]$.

6. Physical interpretation

The physics of the FMM rests on the following fact: given a field, $\psi(\mathbf{x})$, which satisfies the wave equation

$$\left(\nabla^2 + k^2\right)\psi(\mathbf{x}) = 0, \quad (30)$$

for all $\mathbf{x}$ outside a given sphere, the field can be reconstructed everywhere outside that sphere from its far field [16, 17]. This means that if the field is radiated by a source density, $\rho(\mathbf{x})$, supported only within a sphere of radius R centered at the origin,

$$\phi(\mathbf{x}) = \int\limits_{x'<R} d^3\mathbf{x}'\frac{e^{ik|\mathbf{x}-\mathbf{x}'|}}{4\pi|\mathbf{x}-\mathbf{x}'|}\rho(\mathbf{x}'), \quad (31)$$

then the contribution of the "off-shell" ($q^2 \neq k^2$) components in the Fourier expansion of the Green's function [11],

$$\frac{e^{ik|\mathbf{x}-\mathbf{x}'|}}{4\pi|\mathbf{x}-\mathbf{x}'|} = \int\frac{d^3\mathbf{q}}{(2\pi)^3}\frac{e^{i\mathbf{q}\cdot(\mathbf{x}-\mathbf{x}')}}{(2\pi)^3 q^2 - k^2 - i\varepsilon}, \quad (32)$$

(where ε is a infinitesimal positive number, prescribing the correct treatment of the singularity at $q^2 = k^2$) are determined for $x > R$ (after integration over $d^3\mathbf{x}'$) by the radiation condition and the "on-shell" components. The on-shell components, coming from the residue of the pole at $q^2 = k^2$, give the imaginary part of the Green's function, and the off-shell components give the real part. It is important that the off-shell part is *not* determined by the on-shell part for $x' < R$. This is related to the divergence of the series in Equation (11) for $d > X$. This interpretation explains why the far interactions can be computed [Equation (23)]from the radiation pattern $s_m(\hat{k})$ of the mth group. It also clarifies why one only need keep two components in $\mathbf{V}$, $\mathbf{g}$, and $\mathbf{s}$ for the electromagnetic case: the electromagnetic far field is transverse, and has only two independent components.

7. Conclusion

Present methods for computing radar and other scattering cross sections are limited by computer-processing and memory requirements. The significance of the increase in problem size made possible by the FMM can be illustrated by considering the calculation of RCS for X-band radar. With current methods, the size of the largest body that can be accurately modeled is a few feet. With the same computing resources, the techniques that we have described will increase this by at least an order of magnitude. Such computational capabilities would significantly reduce the technological risk of expensive projects employing stealth technology. They may likewise revolutionize other applications of scattering computations, such as high-frequency circuit modeling, sonar, and geophysical applications.

Because the FMM accelerates computation of the matrix-vector product $Z \cdot I$, and thus only indirectly solution of $Z \cdot I = V$, we are frequently asked about the relative merits of direct and iterative solutions, and techniques to reduce the iterations required in a conjugate-gradient type of solution. These are important questions, and are under study by us as well as many others. We consider them to be mostly beyond the scope of this article, but note that the FMM is compatible with "complexification," and with preconditioning by a sparse matrix.

Although we have only demonstrated the use of the FMM for surface-scattering problems, its application to volume-integral equations (necessary for the analysis of penetrable inhomogeneous scatterers) is obvious. When comparison to other techniques for computing the fields of volume source distributions is made, it should be noted that in this case the matrix T in Equation (8) is a strict convolution, and as such can be applied by FFT, resulting immediately in an $O(N \log N)$ algorithm, without further decomposition.

8. Acknowledgment

This research was supported by the Advanced Research Projects Agency of the US Department of Defense, and was monitored by the Air Force Office of Scientific Research under Contracts No. F49620-91-C-0064 and F49620-91-C-0084. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

9. References

1. V. Rokhlin, "Rapid solution of integral equations of scattering theory in two dimensions," *Journal of Computational Physics*, **86**, 2, pp. 414-439, 1990.

2. V. Rokhlin, "Diagonal form of translation operators for the Helmholtz equation in three dimensions," Technical Report YALEU/DCS/RR-894, Yale University, Department of Computer Science, March, 1992, to be published in *Applied and Computational Harmonic Analysis*.

3. R. F. Harrington, *Field Computation by Moment Methods*, New York, Macmillan, 1968.

4. R. F. Harrington, "Origin and development of the method of moments for field computation," *IEEE Antennas and Propagation Society Magazine*, June, 1990, pages 31-35.

5. V. Rokhlin, "Solution of acoustic scattering problems by means of second kind integral equations," *Wave Motion*, **5**, pp. 257-272, 1983.

6. A. W. Maue, "Toward formulation of a general diffraction problem via an integral equation," *Zeitschrift für Physik*, **126**, pp. 601-618, 1949.

7. A. W. Maue, "Toward formulation of a general diffraction problem via an integral equation," in E. K. Miller, L. Medgyesi-Mitschang, and E. H. Newman (eds.), *Computational Electromagnetics*, New York, IEEE Press, pp. 7-14, 1992, (translated into English from [6]).

8. E. K. Miller, L. Medgyesi-Mitschang, and E. H. Newman (eds.), *Computational Electromagnetics, second edition*, New York, IEEE Press, 1992.

9. J. H. Richmond, "On the variational aspects of the moment method," *IEEE Transactions on Antennas and Propagation*, **39**, 4, pp. 473-479, April, 1991.

10. S. M. Wandzura, "Optimality of Galerkin method for scattering computations," *Microwave and Optical Technology Letters*, **4**, 5, pp. 199-200, April, 1991.

11. G. Arfken, *Mathematical Methods for Physicists, second edition*, New York, Academic Press, 1970.

12. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, (Applied Mathematics Series), Cambridge, MA, National Bureau of Standards, 1972.

13. B. K. Alpert and V. Rokhlin, "A fast algorithm for the evaluation of Legendre expansions," *SIAM Journal of Scientific and Statistical Computing*, **12**, pp. 158-179, 1991.

14. A. D. McLaren, "Optimal numerical integration on a sphere," *Mathematics of Computation*, **17**, pp. 361-383, 1963.

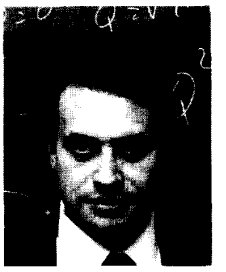
15. S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Transactions on Antennas and Propagation*, **AP-30**, 3, pp. 409-418, May, 1982.

16. A. J. Devaney and E. Wolf, "Radiating and nonradiating classical current distributions and the fields they generate," *Physical Review*, **D8**, 4, pp. 1044-1047, August, 1973.

17. R. Peierls, *Surprises in Theoretical Physics* (Princeton Series in Physics), Princeton, New Jersey, Princeton University Press, 1979.

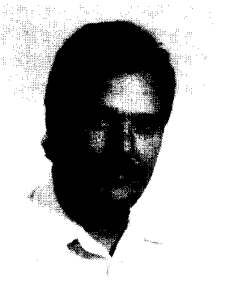
Introducing Feature Article Authors

Ronald Coifman is a professor of mathematics at Yale University. He received his PhD from the Universityof Geneva, in 1965. He received his License es Sciences Mathematiques in 1962. Prior to coming to Yale, Prof. Coifman was a professor at Washington University, a visiting professor at Tel-Aviv University, and a visiting assistant professor at the University of Chicago. Prof. Coifman's recent publications have been in the areas of scattering and inverse scattering, nonlinear harmonic analysis, and wavelet theory. He was chairman of theYale Mathematics Department from 1986 to 1989.



Vladimir Rokhlin

Vladimir Rokhlin has been a professor of computer science and mathematics at Yale University since 1985. He received a PhD in Applied Mathematics from Rice University, in 1983, and a MS in mathematics from Vilnius University, Vilnius, Lithuania, in 1973. Prior to joining the faculty at Yale, Prof. Rokhlin worked as a Senior Research Specialist at Exxon; a partner in Livshitz and Associates, Houston, TX; a consultant at Computer Systems, Houston, TX; and as a mathematician at the Institute of Arctic Geology, Leningrad, Russia. Prof. Rokhlin's research has been in the areas of numerical-scattering theory, elliptic partial-differential equations, numerical solution of integral equations, quadrature formulas for singular functions, and numerical complex analysis.



Stephen Wandzura

Stephen Wandzura is a Senior Scientist, Optical Physics Labortory, Hughes Research Laboratories. He received his BS in music from UCLA, in 1971, and his PhD in physics from Princeton University, in 1977. His research has been in diverse areas of scattering and propagation. His thesis research could be correctly, if somewhat whimsically, characterized as the study of the "impedance matrix of the proton." As a National Research Fellow with the NOAA, he studied scattering of light by atmospheric turbulence, and the occurrence of mountain lee waves. At HRL, he has worked on classical and quantum optics, especially the theoretical and numerical study of stimulated scattering. For the last three years, he has been studying improvements to moment-method techniques. He has published articles in *Physical Review*, *Physical Review Letters*, *Physics Letters*, *Optics Letters*, *Nuclear Physics*, *Wave Motion*, and other journals.

Fast Solution Methods in Electromagnetics

Weng Cho Chew, *Fellow, IEEE*, Jian-Ming Jin, *Senior Member, IEEE*, Cai-Cheng Lu, *Member IEEE*, Eric Michielssen, *Member, IEEE*, and Jiming M. Song, *Member, IEEE*

Invited Paper

***Abstract*— Various methods for efficiently solving electromagnetic problems are presented. Electromagnetic scattering problems can be roughly classified into surface and volume problems, while fast methods are either differential or integral equation based. The resultant systems of linear equations are either solved directly or iteratively. A review of various differential equation solvers, their complexities, and memory requirements is given. The issues of grid dispersion and hybridization with integral equation solvers are discussed. Several fast integral equation solvers for surface and volume scatterers are presented. These solvers have reduced computational complexities and memory requirements.**

***Index Terms*— Numerical methods.**

I. Introduction

COMPUTATIONAL electromagnetics is a fascinating discipline that has drawn the attention of mathematicians, engineers, physicists, and computer scientists alike. It is a discipline that creates a symbiotic marriage between mathematics, physics, computer science, and various application fields. Computational techniques for solving electromagnetic wave scattering problems involving large complex bodies and for analyzing wave propagation through inhomogeneous media have been intensely studied by many researchers in the past [1]–[12]. This is due to the importance of this research in many practical applications, such as the prediction of the radar cross section (RCS) of complex objects like aircraft, the interaction of antenna elements with aircraft and ships, the environmental effects of vegetation, clouds, and aerosols on electromagnetic wave propagation, the interaction of electromagnetic waves with biological media, and the propagation of signals in high-speed and millimeter wave circuits.

Due to the large electrical dimensions of typical aircraft, past efforts to ascertain their scattering cross section and their interaction with antennas have exploited approximate high-frequency techniques such as the shooting and bouncing ray method [13]. However, the recent phenomenal growth in computer technology, coupled with the development of fast algorithms with reduced computational complexity and memory requirements, have made a rigorous numerical solution of the problem of scattering from electrically large objects feasible. These numerical techniques involve either solving partial-differential equations with the finite-difference method (FDM) [6]–[9] or the finite-element method (FEM) [10]–[12] which result in sparse matrices, or integral equations which are converted into dense matrix equations using the method of moments (MoM) [1]–[5].

In a previous paper [14], we underscored the importance of reducing the computational complexity of computational electromagnetics techniques, especially for large-scale electromagnetic problems. We reviewed several direct solvers with reduced computational complexity whereby the solution is sought for all right-hand sides. These direct solvers are the recursive aggregate T-matrix algorithm (RATMA) [15], [16], and the nested equivalence principle algorithm (NEPAL) [16], [17]. In this paper, we will first review differential equation solvers, and discuss their computational complexities. We next focus on recent work in integral equation solvers, and contrast their complexities with those of differential equation solvers. Throughout, we will focus primarily on iterative solvers, which are used ubiquitously for solving both differential and integral equations. Iterative solvers, in general, require less memory storage, and exhibit reduced computational complexities when compared to direct solvers. Hence, they portend an important method for large scale computing.

II. Differential Equation Solvers

A popular way to solve electromagnetic problems is to solve the associated partial differential equation directly. These methods can be considered as the first fast solution methods in electromagnetics because one can solve an N unknown problem with computational complexity less than $O(N^3)$ and memory requirement less than $O(N^2)$. Differential equation solvers usually involve either the FEM [10]–[12] or the FDM [6]–[9]. The pertinent matrix equation is sparse with $O(N)$ nonzero elements. Consequently, a matrix–vector multiply can be performed in $O(N)$ operations. By properly ordering the elements, the bandwidth of the pertinent matrix equation can be compressed and inverted very efficiently [18]. Differential equation solvers are usually applied to volumetric problems and, hence, the following discussion is pertinent to volumetric cases.

Partial differential equations (PDE's) for electromagnetics can be roughly categorized into elliptic type (static or Laplacian-like) for low frequencies, hyperbolic type (wave-like) for high frequencies, and parabolic type (diffusion like) for intermediate frequencies and lossy media. Elliptic PDE's

Manuscript received April 4, 1996; revised September 30, 1996. This work was supported in part by AFOSR under Grant F49620-96-1-0025, as well as by from ONR and NSF.

The authors are with the Center for Computational Electromagnetics, Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801 USA.

Publisher Item Identifier S 0018-926X(97)02302-8.

shave the advantage of positive definiteness [18]; hence, when iterative methods are used to solve the associated matrix equation, a definitive statement can be made about their convergence rates. For instance, when the conjugate gradient (CG) method [19], [20] is used to solve the Poisson equation, it converges in $O(N^{0.5})$steps in two dimensions, and in $O(N^{0.33})$ in three dimensions. When the multigrid method is used to solve the same equation, the number of iterations is independent of the size of the problem [18]. As a consequence, the total computational labor associated with the conjugate gradient method to solve such problems scales as $N^{1.5}$ in two dimesnions and $N^{1.33}$ in three dimensions, while it scales as $N$for multigrid methods.

For hyperbolic (wavelike) problems which are indefinite, these computational complexities can only be regarded as lower bounds, because with the change of geometry, resonance can occur, and the number of iterations needed for convergence in an iterative solver can diverge. Multigrid solvers exploit the scale-invariant nature of an elliptic (Laplacian-like) equation to reduce the computational complexity. But when applied to the Helmholtz wave equation (hyperbolic), the computational complexity is not reduced, because the Helmholtz wave equation is not scale invariant.

When the finite-difference time-domain (FDTD) method is used to solve the wave equation directly in the time domain, the computational complexity is the same as CG ($N^{1.5}$ in two dimensions and $N^{1.33}$ in three dimensions) where they are lower bounds [21] except that FDTD generates the solution for all time and, hence, all frequencies at once. It is also an optimal algorithm in the sense that it generates $O(N^{\alpha})$ numbers in $O(N^{\alpha})$ operations.

Of interest also is the spectral Lanczos decomposition method (SLDM) [22], [23]. While it does not reduce the computational complexity compared to CG, it offers an advantage when there are large regions of homogeneity. Also, it can generate the solution for all frequencies without additional computational cost [23]. For numerical simulation of waveguides where a large section of uniformity exists, the method of lines [24] and the numerical mode-matching method [25], [26] offer an advantage over other differential equation methods in terms of speed.

When applied to a scattering problem, a PDE solver requires absorbing boundary conditions (ABC's) [21] to truncate the simulation region. Many ABC's have been proposed so that the sparsity of the matrix can be maintained. However, these ABC's are approximate and have to be imposed at a substantial distance from the scatterer to reduce the errors incurred by them. Recently, an absorbing material boundary condition (AMBC), called perfectly matched layer (PML), has been suggested by Berenger [27] and worked on intensely by a number of workers [28]–[35]. This AMBC is particularly well-suited for the parallel implementation of FDTD solvers because it permits parallel computers to operate in a single-instruction-multiple-data (SIMD) mode [28].

Another approach to truncate the simulation region is to employ the eigen function expansion of the scattered field outside a separable boundary. This separable boundary can either be circular or elliptical in two dimensions and spherical or spheroidal in three dimensions. The resultant method is often referred to as the unimoment method [36]–[39]. Through the use of coupled basis functions for the separable boundary, the differential equation can be effectively decoupled from the dense matrix generated from the eigen function expansion. The dimension of this dense matrix is, however, rather small. In two dimensions, it is about kd where d is the largest linear dimension of the scatterer.

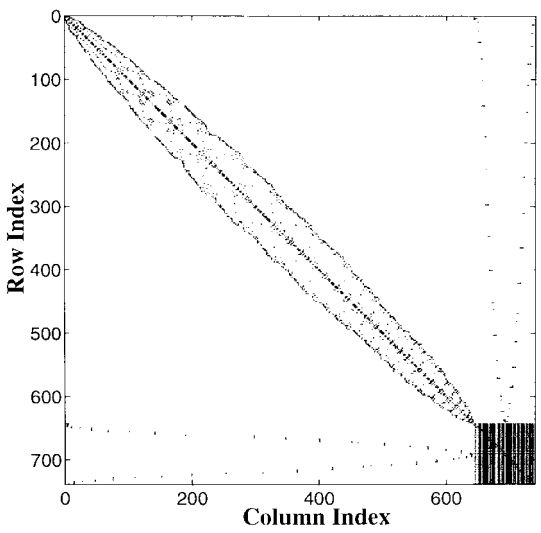


Fig. 1. Proper ordering of the elements of an FEM matrix that is coupled to a surface integral equation causes the dense matrix to reside at the bottom right-hand corner, making the system matrix suitable for the matrix-partitioning method.

Alternatively, surface integral equations (which can be considered to be numerically exact ABC's) can be used to truncate the mesh of the differential equation solvers [40], [41]. By so doing, the boundary of the simulation region can be brought much closer to the surface of the scatterer, thereby reducing the size of the simulation region and the number of associated unknowns. However, such a method of "absorbing" the outgoing wave results in a partially dense matrix in the final matrix system for the problem.

By a proper ordering of the nodes in FEM [41], [42], the dense matrix will reside only at the bottom right-hand corner of the matrix system as shown in Fig. 1. In this manner, the inverse of the matrix system can be found by the matrix-partitioning method. When nested-dissection ordering [43] is applied to the sparse part, and LU decomposition is applied to the dense part, the overall computational complexity is of $O(N^{1.5})$ in two dimensions, and of $O(N^2)$ in three dimensions. The memory requirements are $O(N \log N)$ in two dimensions and $O(N^{4/3})$ in three dimensions [18].

When iterative methods are used to solve the matrix system as shown in Fig. 1, the matrix–vector multiply from the dense submatrix could become a bottleneck in three dimensions or for thinly coated metallic scatterers. However, with the use of fast integral equation solvers [44], [45] this bottleneck could be removed. The example of hybridizing a fast integral equation solver and FEM has been illustrated [46]. Fig. 2 shows the comparison of such a calculation with experiments [47] when applied to an elliptically contoured crack in a ground plane.

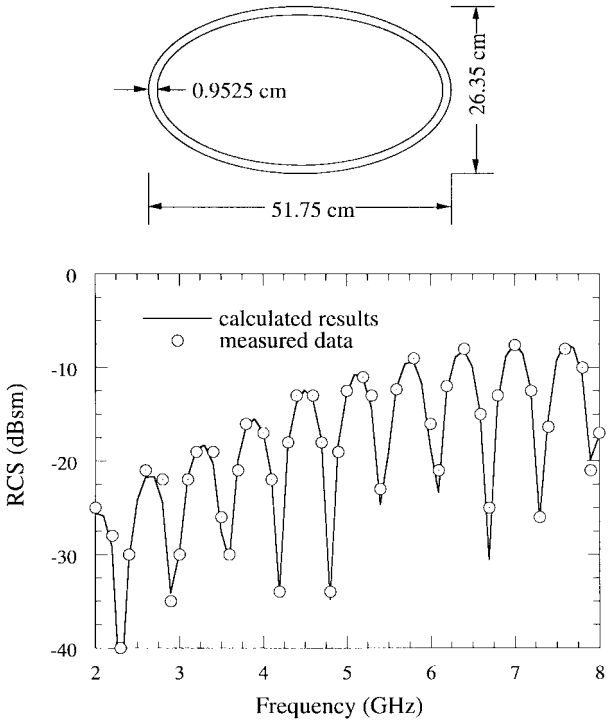


Fig. 2. Backscatter RCS of an elliptic contoured crack in a ground plane as a function of frequency at the incidence angle $\theta = 70°$ (20° from grazing incidence) and $\phi = 85°$ (5° from the minor axis). The crack is 0.635 cm deep and filled with air.

One of the drawbacks of differential equation solvers is the grid dispersion error incurred [48]–[50]. The grid dispersion error causes a wave to have a different phase velocity on a grid compared to the exact solution. This error can be suppressed by using a higher grid density, but at the expense of increased computational labor. Because the error is cumulative, it is particularly pronounced for simulation over a large region or for large scatterers. To suppress the error, the grid density has to increase as the simulation region increases in size. For second-order accurate schemes, the grid density in one dimension (number of points per wavelength) has to grow as $(kd)^{0.5}$ where d is the "diameter" of the simulation region and k is the wave number of the wave [49]. Therefore, the number of unknowns grows as $(kd)^{1.5}$ in one dimension. Hence, in two dimensions, the number of unknowns scales as $(kd)^3$ while in three dimensions, it scales as $(kd)^{4.5}$. A remedy for this is to use a higher order accurate differential equation solver [51], [52] or to couple the differential equation solver to an integral equation solver when large homogeneous regions exist.

III. Integral Equation Solvers

Alternatively, a scattering problem can be cast into an integral equation. Integral equation solvers usually involve a smaller number of unknowns than differential equation solvers because only the induced sources are unknowns, whereas in a differential equation, the field is the unknown. For example, for a metallic scatterer, the induced current resides only on the surface of the scatterer. Hence, for a scatterer in a three-dimensional (3-D) space, the induced current exists in a space of smaller dimensions, greatly reducing the number of unknowns required to accurately represent the solution, except for inhomogeneous scatterers. However, integral equation solvers result in dense matrices. If the matrix equation is then solved by LU decomposition (Gaussian elimination) or alternatively by an iterative technique such as the CG or related methods [19], [20], the computational labor may be excessive. LU decomposition requires $O(N^3)$ operations and $O(N^2)$ memory storage and provides a solution for all excitations of the scatterer. CG requires $O(N^2)$ operations per iteration for dense matrices, because the most costly step in a CG iteration is in the matrix–vector multiply. In general, the number of iterations grows with the electrical size of the scatterer. A straightforward implementation of CG requires $O(N^2)$ memory storage, providing a solution that is valid for only one excitation. However, it is possible to iteratively solve the pertinent equation concurrently for multiple right-hand sides, thereby exploiting as much as possible the redundancies in the right-hand sides [53].

The high-computational complexity of the aforementioned solution schemes precludes their application to the analysis of scattering from large structures. Many researchers have attempted to reduce the complexity of the traditional MoM algorithm by reducing the computational labor of the pertinent matrix–vector multiplies. For surface scatterers, Rokhlin [44] proposed the fast-multipole method (FMM) to reduce the computational complexity of matrix–vector multiplies in an iterative method. Canning [54] has developed the impedance-matrix localization (IML) method, which uses basis functions that produce directed beams. This results in a sparse MoM matrix, which in turn, expedites a matrix–vector multiply. The IML works well for smooth surfaces, but not for nonsmooth surfaces. In a similar spirit, the complex multipole beam approach has been introduced [55], but it again works only for smooth surfaces.

Wavelet transforms [56]–[61] have also been used to yield sparse matrices that can be solved rapidly. Since wavelets are scale invariant, they are well suited for solving static or low-frequency (elliptic) problems. When wavelets are used to sparsify matrices resulting from an integral equation of static, they sparsify the matrices to $O(N \log N)$ elements, reducing the operation count of a matrix–vector multiply to $O(N \log N)$. For wavelike problems, even though these methods expedite matrix–vector multiplies, they do not reduce the computational complexity [61] when the scatterer size grows with respect to wavelength. Many methods have been proposed in the past which, even though will reduce solution time, do not reduce the computational complexity [62]–[65].

For volumetric scatterers, several recursive and nesting algorithms have been developed to directly obtain the solutions of integral equations for all right-hand sides [14]–[17]. Also, in an iterative method, the FFT can be used to expedite the matrix–vector multiply and reduce the computational complexity and memory requirement for solving such scattering problems [66]–[77].

Here, we will first discuss fast methods to solve volume integral equations rapidly using FFT [66]–[72]. Then, for surface integral equations, we will first discuss the use of wavelet transforms to expedite matrix–vector multiplies in an

iteration solution method. Finally, we will discuss the use of the FMM related methods, and various multilevel algorithms to accelerate matrix–vector multiplies in an iterative solver.

IV. Iterative Methods for Volume Scattering

Scattering from a volumetric object can be analyzed efficiently using iterative methods where the bottleneck is the matrix–vector multiply involving a dense matrix. The matrix–vector multiply represents the action of the Green's operator on induced currents in the scatterer. Since the Green's operator is translationally invariant, this action can be written as a convolutional integral

$$\int_V d\mathbf{r}' g(\mathbf{r}-\mathbf{r}')j(\mathbf{r}') \tag{1}$$

where $g(\mathbf{r})$ is the Green's function and $j(\mathbf{r})$ is the induced current. Such action can be expedited using an FFT, with a complexity of $O(N\log N)$ [66]–[70]. However, in three dimensions, the Green's function for electromagnetic scattering is highly singular (as in the dyadic Green's function). Therefore, a high sampling rate is needed to perform the above convolution accurately. To mitigate the singularity of the dyadic Green's function, a difference operator is used to approximate the differential operator in the dyadic Green's function in [67]. In a similar vein, [68] proposed the use of a weak formulation of the integral equation plus a spherical mean approximation. In [69], the induced current is expanded in terms of a continuous function, even though the induced current, which is proportional to $(\epsilon-\epsilon_0)\mathbf{E}$, should be a discontinuous function.

Alternatively, we can discretize the above integral by projecting it on to a smaller subspace using the MoM [1], converting the integral operator into a matrix operator. The singularity of the Green's operator is being mitigated by this projection. When the subspace is spanned by subdomain basis functions, and the mesh used is rectilinear, the pertinent matrix is (block) Toeplitz [70]. Consequently, the matrix–vector multiply can be performed exactly by using an FFT requiring $O(N\log N)$ operations. Fig. 3 shows the bistatic RCS of a layered sphere computed using such a method. The sphere is modeled by 90 000 unknowns, and a matrix–vector multiply can be performed in several minutes on a 10 MFLOPS machine.

Alternatively, we can decompose the inhomogeneous scatterer into N subscatterers, whose scattering is characterized by a T matrix. Then a set of linear algebraic equations accounting for the multiple scattering between the subscatterers is derived. When the scatterers reside on a regular array, the pertinent matrix equation has a Toeplitz structure, and the FFT can be used to compute the matrix–vector multiply in $O(N\log N)$ operations [71], [72].

Both this method and the MoM method avoid the singularity of the Green's function, and only a low sampling rate is needed to perform the FFT accurately. Fig. 4 shows the bistatic RCS of a dielectric sphere computed using such a method. This method does not have low-frequency instability problems as opposed to some FEM formulations as discussed in [78], [79].

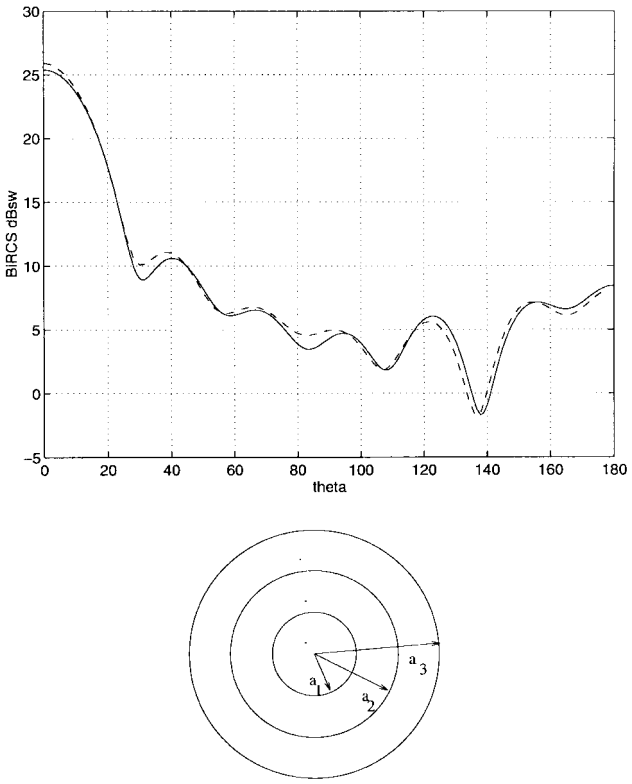


Fig. 3. The bistatic RCS of a three-layered spherical scatterer. The solid line is from the Mie series solution; the dashed line is for the numerical solution using BiCG–FFT. $a_1 = 0.169$ m, $\epsilon_{r,1} = 1.2$, $a_2 = 0.339$ m, $\epsilon_{r,2} = 2.0$, and $a_3 = 0.508$ m, $\epsilon_{r,3} = 2.4$; with $31 \times 31 \times 31$ grids. This problem has about 90 000 unknowns and the frequency is 590 MHz ($\lambda_0 = 0.508$ m).

Comparison of the efficiency of the CG–FFT method and recursive aggregate T-matrix algorithm (RATMA) has been presented in [73], [74]. When a scatterer is lossless, RATMA is superior to CG–FFT. But when the scatterer is lossy, the number of iterations required is small, and CG–FFT is more efficient than RATMA.

The CG–FFT method can also be used to expedite the solution of the scattering from a cluster of randomly distributed spheres and randomly distributed cylinders. When the subscatterers do not reside on a regular array, a precorrected method can be used to derive a Toeplitz matrix structure, and FFT can again be used to accelerate the matrix–vector multiply [71], [76]. The precorrected FFT method has also been used in the adaptive integral method (AIM) [77], which will be discussed in greater detail in Section XI.

V. Wavelets

There have been many attempts at using wavelets to solve scattering problems [56]–[61]. Such approaches have met with some success at lower frequencies due to the elliptic nature of the electrostatic problem. For instance, wavelets can be used to sparsify the boundary integral equation of electrostatics. The originally dense matrix resulting from discretizing this integral equation reaches a sparsity of $O(N\log N)$ after applying a wavelet transform [80]. This sparsity occurs because the integral operator belongs to the class of Calderon–Zygmund operators [80], [81], where the kernel is infinitely smooth.

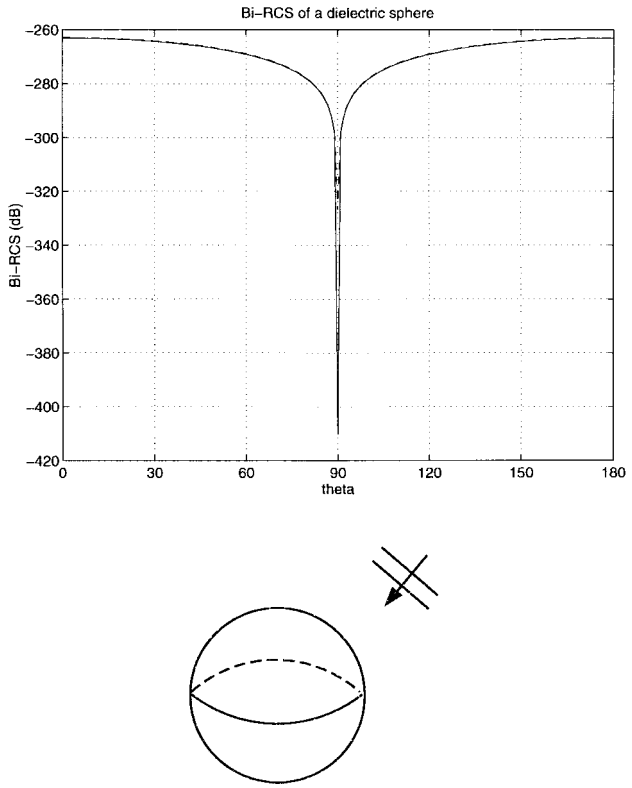
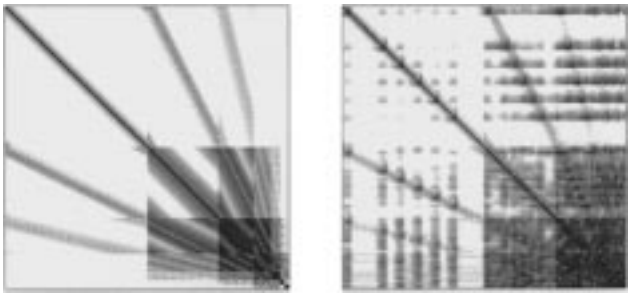


Fig. 4. The bistatic RCS of a dielectric sphere computed using the BiCG–FFT T-matrix method. The solid line is from the Mie series solution; the dashed line is the numerical solution. Here, the radius equals $10^{-5}\lambda_0$ and $\epsilon_r = 4.0$. A $16 \times 16 \times 16$ grid is used. The number of iterations needed to solve this problem is independent of the number of unknowns at such a low frequency.

A physical explanation is that at low frequencies, wavelet function currents generate only localized fields. In other words, in electrostatics the interaction between wavelet sources is mainly short range. This is particularly so for interactions due to the "fine features" of the sources. In addition, electrostatic problems are scale invariant, as are the wavelet bases.

For PDE's, the associated matrix is already sparse. Hence, there is no apparent advantage to applying a wavelet transform to such a matrix. However, for elliptic PDE's (static), the wavelet transform generates a matrix that can be easily preconditioned so that the resultant condition number of the matrix is of order one, irrespective of the size of the scatterer [82]. As a result, when an iterative solver is used, the number of iterations is independent of the problem size and it can be solved in $O(N)$ operations. Therefore, wavelets for elliptic PDE's offer advantages similar to those of multigrid.

Unfortunately, for wavelike problems the associated integral equation has an oscillatory kernel. In other words, wavelike problems are not scale invariant. Hence, there is no clear advantage to using a wavelet transform on such a kernel, as one can show that the sparsity of the matrix cannot be reduced to less than $O(N^2)$, the lower bound being related to the Nyquist sampling rate in Fourier analysis [61]. The physical explanation is that when the length scale of a wavelet equals or exceeds the wavelength, it becomes an efficient radiator. Hence, strong long-range interactions exist between these basis functions irrespective of the problem size. The long range interaction in electrodynamics falls off as $1/r$; this decay cannot be ignored even over large distances.



(a) (b)

Fig. 5. The matrices after wavelet transform appear sparse but are dense at the bottom right-hand corner. The left one is for a circular cylinder, while the right one is for an L-shaped cylinder. Daubechies wavelets with eight vanishing moments are used.

Local cosine transforms have been suggested as a remedy to this problem [83]. Local cosine current functions radiate a field that has a sharply directed beam pattern as in IML [54]. As a result, the matrix becomes sparse when the scatterer has a smooth surface. However, when the surface is rough, the local cosine current function loses its sharply directed beam pattern, and the matrix loses its sparsity.

Given a matrix equation resulting from discretizing an integral equation using the method of moments with a pulse basis

$$\overline{\mathbf{A}} \cdot \mathbf{x} = \mathbf{b} \tag{2}$$

the corresponding wavelet basis representation can be related to the pulse basis representation by a matrix transform

$$\mathbf{x} = \overline{\mathbf{U}} \cdot \mathbf{w} \tag{3}$$

where $\overline{\mathbf{U}}$ is unitary when the wavelet basis is orthonormal (though nonorthogonal wavelets are also used). By using (3) in (2), we have

$$\overline{\mathbf{U}}^t \cdot \overline{\mathbf{A}} \cdot \overline{\mathbf{U}} \cdot \mathbf{w} = \overline{\mathbf{U}}^t \cdot \mathbf{b} \tag{4}$$

or

$$\tilde{\overline{\mathbf{A}}} \cdot \mathbf{w} = \tilde{\mathbf{b}} \tag{5}$$

where

$$\tilde{\overline{\mathbf{A}}} = \overline{\mathbf{U}}^t \cdot \overline{\mathbf{A}} \cdot \overline{\mathbf{U}} \tag{6a}$$

$$\tilde{\mathbf{b}} = \overline{\mathbf{U}}^t \cdot \mathbf{b}. \tag{6b}$$

The matrix $\tilde{\overline{\mathbf{A}}}$ is the moment-method matrix represented in the wavelet basis. Fig. 5 [61] shows two matrices from a two-dimensional (2-D) electrodynamic boundary integral equation for a circular scatterer and an L-shaped scatterer after wavelet transform using Daubechies wavelets [84]. It is seen that the bottom right-hand corner of the matrix remains dense.

Fig. 6 [61] shows the matrix sparsity as a function of the number of unknowns for the circular scatterer and the L-shaped scatterer. It is clear that the fraction of nonzero elements remains a constant after the scatterer has increased to a certain size. Here, a discretization density of ten points per wavelength is used throughout the study. Hence, as the size of the scatterer increases, its dimension increases with

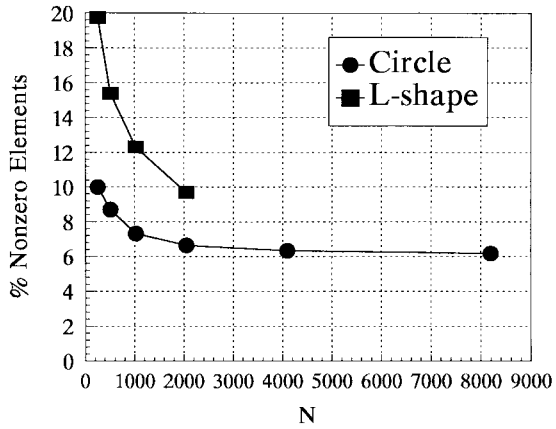


Fig. 6. The percentage of nonzero elements as a function of the number of unknowns. It is seen that the percentage does not go down after awhile because of the long-range interaction of wavelets in electrodynamics.

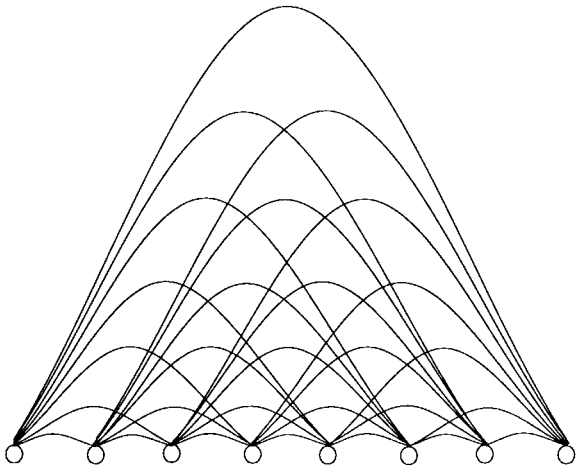


Fig. 7. A one-level matrix–vector multiply where all current elements talk directly to each other. The number of "links" is proportional to N^2 where N is the number of current elements.

respect to the wavelength. However, if we keep the size of the scatterer constant with respect to the wavelength and increase the discretization density to increase the number of unknowns, then the sparsity of the matrix will increase as expected. The wavelet transform removes oversampling of the unknowns beyond the Nyquist rate.

VI. Fast Multipole Method

For surface structures, there exists no direct solver with reduced computational complexity for efficiently solving the integral equation of scattering. Therefore, one resorts to an iterative solver whereby the computational complexity of a matrix–vector multiply can be reduced. Many methods for expediting matrix–vector multiplies have been proposed, but the FMM and its variants [44], [45], [85]–[92] hold most promise in providing a fast method that applies to scatterers of arbitrary geometry. The detailed mathematical description of the FMM for electromagnetic problems can be found in the aforementioned references. Therefore, we will describe this method from a heuristic viewpoint.

A matrix–vector multiply involving a dense matrix and a dense vector requiresN^2 operations. This is illustrated by Fig. 7. In essence, every element of a vector communicates with every other element directly. Clearly, N^2 operations are needed. The above is like connecting N cities with direct flight routes. The number of flight routes increases as N^2. However, if "hubs" are introduced in the flight routes, then their number can be reduced, as shown in Fig. 8, where the number of flight routes becomes less than N^2. Since Fig. 8 represents a two-level structure, a matrix–vector multiply would have to be effected in three stages. Therefore, a matrix element has to be factored as a product of three terms. In other words, a matrix–vector multiply can be expressed as

$$\sum_{j=1}^{N} A_{ij}\, x_j = \mathbf{V}_{il}^{t} \cdot \sum_{l'=1}^{N/M} \overline{\alpha}_{ll'} \cdot \sum_{j \epsilon \mathcal{G}_{l'}} \mathbf{V}_{l'j}\, x_j \qquad \begin{cases} i \in \mathcal{G}_l \\ l = 1, \cdots, \dfrac{N}{M}. \end{cases} \tag{7}$$

In the above, we assume that the N elements in the vector are divided in groups with M elements each. Therefore, there are a total of N/M groups $\mathcal{G}_l$. Moreover, it implies that a matrix A_{ij} derived from the integral equation of scattering can be factored as

$$A_{ij} = \mathbf{V}_{il}^{t} \cdot \overline{\alpha}_{ll'} \cdot \mathbf{V}_{l'j}. \tag{8}$$

The above factorization is achievable by using the addition theorem, where l' corresponds to the center of the l'th group, which contains the jth element. This is possible for i and j belonging to different nonoverlapping groups. Unfortunately, in the above, a scalar number $A_{ij}$is converted into a product of a vector, a matrix and a vector. Therefore, even though the number of "routes" diminishes as shown in Fig. 8, the number of operations is not reduced; it is still of $O(N^2)$. It can be shown that the dimensions of $\mathbf{V}$ and $\overline{\alpha}$ in (7) are proportional to M, the number of elements in the group it represents. Fortunately, a change of basis to the plane-wave basis diagonalizes the matrix $\overline{\alpha}_{ll'}$. This diagonalization was first achieved by Rokhlin [44]. Hence, one can write

$$A_{ij} = \tilde{\mathbf{V}}_{il}^{t} \cdot \tilde{\overline{\alpha}}_{ll'} \cdot \tilde{\mathbf{V}}_{l'j} \tag{9}$$

where $\tilde{\overline{\alpha}}_{ll'}$ is now a diagonal matrix. By so doing, the number of operations for a matrix–vector multiply as expressed by (7) can be reduced for the nonnearest neighbor (nonoverlapping) groups. Choosing the group size $M \sim \sqrt{N}$, the matrix–vector multiply can be effected in $O(N^{1.5})$ operations [44], [45], [85], [86]. Fig. 9 shows the use of the FMM to calculate the electromagnetic scattering of a NASA almond [86].

VII. Ray-Propagation Fast Multiple Algorithm (RPFMA)

In the FMM, a matrix–vector multiply is expressed as

$$\sum_{j=1}^{N} A_{ij} x_j = \tilde{\mathbf{V}}_{il}^{t} \cdot \sum_{l'=1}^{N/M} \tilde{\overline{\alpha}}_{ll'} \cdot \sum_{j \epsilon \mathcal{G}_{l'}} \tilde{\mathbf{V}}_{l'j} x_j \qquad \begin{cases} i \in \mathcal{G}_l \\ l = 1, \cdots, \dfrac{N}{M}. \end{cases} \tag{10}$$

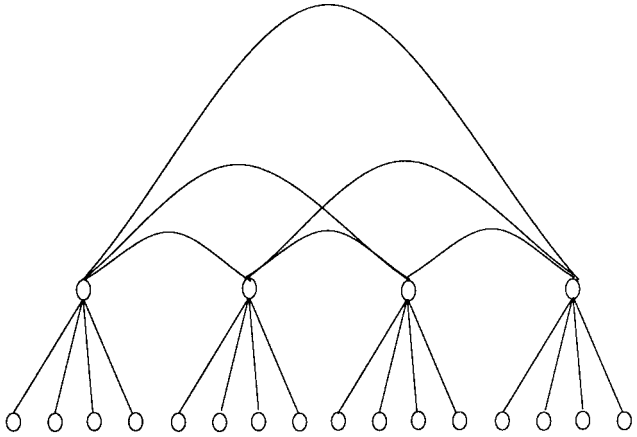


Fig. 8. A two-level matrix–vector multiply where "hubs" are established to reduce the number of direct "links" between the current elements. This could potentially reduce the complexity of a matrix–vector multiply.

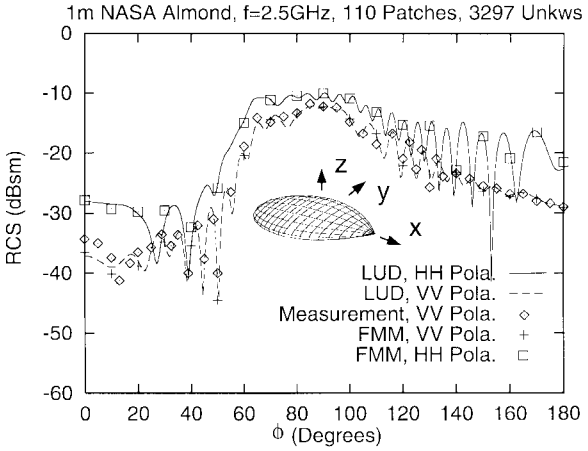


Fig. 9. Monostatic RCS of a 1-m long NASA almond at 2.5 GHz in the xy plane with $\theta = 90^\circ$. Five unknowns are used per wavelength. The results are computed with LU decomposition, and partially with FMM. The experimental measurement by Ohio State University [3] is also given for comparison.

The first step

$$\mathbf{b}_{l'} = \sum_{j \in \mathcal{G}_{l'}} \tilde{\mathbf{V}}_{l'j} x_j \tag{11}$$

calculates the plane waves with k-vectors on a sphere (or a circle in two dimensions) radiated by the sources x_j in group $\mathcal{G}_{l'}$. Then the second step

$$\mathbf{c}_l = \sum_{l'} \tilde{\overline{\alpha}}_{ll'} \cdot \mathbf{b}_{l'} \tag{12}$$

calculates the plane waves with different k vectors on a sphere received by group $\mathcal{G}_l$ after the plane waves have been translated through the space separating the centers of groups $\mathcal{G}_l$ and $\mathcal{G}_{l'}$. Then, the last multiply

$$d_i = \tilde{\mathbf{V}}^t_{il} \cdot \mathbf{c}_l \tag{13}$$

redistributes the plane waves received by group $\mathcal{G}_l$ to the ith element of the group.

If the groups $\mathcal{G}_l$ and $\mathcal{G}_{l'}$ are far apart, it is clear that not all plane waves on a sphere will participate in the interaction between the elements of the two groups [87], [88]. In fact,

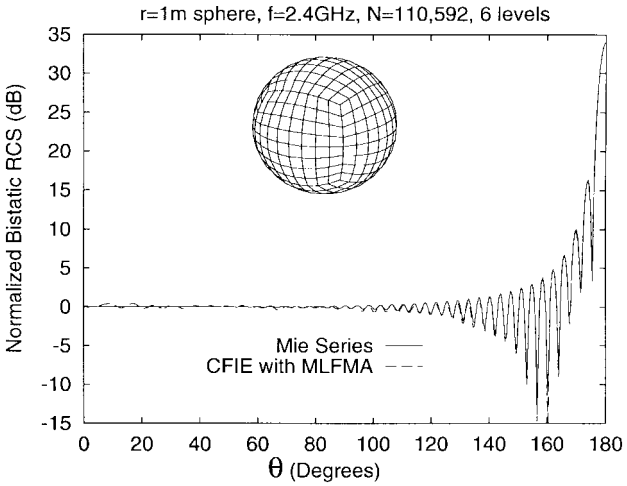


Fig. 10. Validations of CFIE [94] with MLFMA against the Mie series of the bistatic RCS of a metallic sphere of radius 1 m at 2.4 GHz ($\lambda_0 = 12.5$ cm) for VV polarization. 110 592 unknowns with a six-level FMM are used. The RCS is normalized by πa^2. The computation takes 24 h on a 10-MFLOPS machine.

ray physics dictates that only a small fraction of the plane waves accounts for the interaction between the two groups. Therefore, the dimension of the matrix $\tilde{\overline{\alpha}}_{ll'}$ can be reduced to manifest this ray physics. Then the cost of the operation in (10) can be further reduced. Because of this, more intragroup calculation is desired. In this case, we choose $M \sim N^{1/3}$, and the complexity of a matrix–vector multiply can be further reduced to $O(N^{4/3})$.

A simplification of ray-propagation fast multiple algorithm (RPFMA) is the fast far-field approximation (FAFFA) [89]. This method greatly simplifies the matrix elements for the far interactions between the elements; hence, they can be computed as needed. Therefore, an algorithm with $O(N)$ memory requirement is possible in this case.

VIII. Multilevel Algorithms

A logical extension of the two-level FMM is a multilevel algorithm [90]–[93]. In this case, the number of levels is proportional to $\log N$. If only N operations are needed at each level, this becomes an $N \log N$ algorithm for matrix–vector multiplies. Order N operations can be maintained at each level by interpolation and anterpolation [89], [90]. Fig. 10 shows the use of the multilevel fast multipole algorithm (MLFMA) to solve a 110 592 unknown problem on a workstation using the combined field integral equation [94]. The memory requirement of this algorithm is $O(N \log N)$, allowing large problems to be solved on a small computer.

The matrix decomposition algorithm (MDA) and its multilevel cousin (MLMDA) [95], [96] accelerate the iterative solution of electromagnetic scattering problems involving large scatterers. Unlike the FMM, which relies on an analytical diagonalization of the translation operator, the MDA and MLMDA decompose MoM matrices using commonly available linear algebraic techniques. The MDA and MLMDA directly exploit the limited number of degrees of freedom (DoF) [97] that characterize a field observed over a domain that is "well separated" from a source domain.

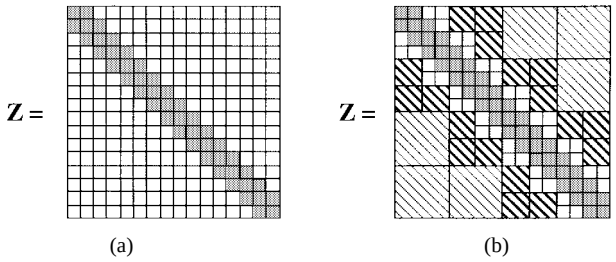


Fig. 11. (a) The MoM matrix. (b) Decomposed MoM matrix. Blocks representing near-field interactions are stored classically, while blocks representing far-field interactions are stored as products of low-rank matrices (MDA) or are aggregated and stored using an FFT-like scheme.

The MLMDA differs from the MDA in that off-diagonal blocks of the MoM matrix are aggregated into larger entities and decomposed using a multilevel algorithm that resembles an FFT as shown in Fig. 11. The memory requirements and the computational complexity of the MDA are $O(N^{3/2})$ while those for the MLMDA asymptotically approach $N \log^2 N$. The MLMDA can easily be incorporated into existing MoM programs. The MLMDA has been applied to the solution of scattering problems involving large 2-D scatterers with 50 000 unknowns.

IX. Fast Steepest Descent Path Algorithm (FASDPA)

The fast steepest descent path algorithm (FASDPA) [98], [99] constitutes a novel two-level algorithm that hybridizes the FMM and the MDA. Not unlike the FMM and the MDA, the FASDPA starts from a spatial decomposition of the scatterer into a large number of subscatterers, and interactions between nearby subscatterers are accounted for directly. Interactions between distant subscatterers are expressed in terms of a small set of equivalent sources that exhaust the degrees of freedom of the interaction field, as in the MDA. To permit the algorithm to "recycle" information in a manner similar to the FMM, the field radiated by each group is represented in terms of a set of homogeneous plane waves. Equivalent source amplitudes are obtained from the plane wave spectrum. More specifically, the FASDPA expresses the interaction field between distant groups as (8). However, in contrast to the FMM, where the $\overline{\alpha}_{ll'}$ matrix represents a diagonal translation matrix for homogeneous plane waves emanating from the source, the $\overline{\alpha}_{ll'}$ matrix for the FASDPA is empty except for a small translation block, appearing on the diagonal. The computational complexity of the FASDPA is $O(N^{4/3})$ per iteration without proceeding to a multilevel scheme. Fig. 12 compares the RCS of a corrugated semicircular structure computed using the FASDPA with results obtained using the MLMDA.

X. Fast Algorithm for Elongated Structures

Numerical algorithms for analyzing electromagnetic scattering from elongated objects, i.e., structures whose dimensions extend primarily along one spatial axis and which are uniform or of limited extent along the other two axes, are of great practical interest. A nonexclusive list of potential applications includes the analysis of scattering from rough surfaces, wing-

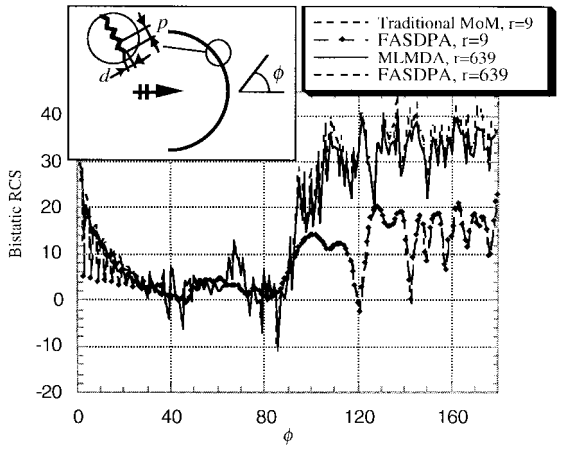


Fig. 12. Bistatic RCS of corrugated semicircular structure $p = 1.57\lambda$, their depth is $d = 0.5\lambda$, and the structure is illuminated by a TM_z plane wave traveling along the direction $r = 9\lambda$ ($N = 400$) and $r = 639\lambda$ ($N = 25\,600$).

like structures, truncated and quasiperiodic structures, as well as the analysis of radiation from large phased-array antennas. Several methods have been proposed to seek an efficient solution to such problems [100], [101].

The equivalent source algorithm for elongated structures (ESAES) [102] is a fast direct solver for analyzing scattering from such structures. ESAES is conceptually similar to RATMA for planar structures [100]—both are based on a recursive characterization of increasingly larger subscatterers using scattering matrices of reduced dimensions and both algorithms have $N \log^2 N$ computational complexity and $O(N \log N)$ memory requirements. The ESAE S abandons the plane wave representation of the 2-D Green's function employed in [100] in favor of a reduced spatial representation of the fields that are scattered by an elongated object. This reduced spatial representation permits the computation of the fields radiated by N quasi-aligned sources and observed over an elongated domain in terms of that radiated by $O(N \log N)$ equivalent sources. The concept of a reduced field representation is directly related to that of the limited number of degrees of freedom that characterize fields radiated by electromagnetic sources [97]. This reduced field representation can be obtained by augmenting an existing MoM code with purely algebraic techniques, e.g., a singular value or a rank revealing QR decomposition. We have applied the ESAES to 2-D structures that measure several thousand wavelengths in length. Fig. 13 shows the bistatic RCS of a finite periodic structure computed using the ESAES and compares the results to those obtained using the MLMDA.

XI. Adaptive Integral Method

Even though precorrected FFT methods have been used in the past to solve electrodynamic [71], [76] and electrostatic problems [103], a note is in order on a related technique developed by Bleszynski *et al.* [77], termed the adaptive integral method (AIM), which has been successfully applied to the analysis of scattering from very complex structures.

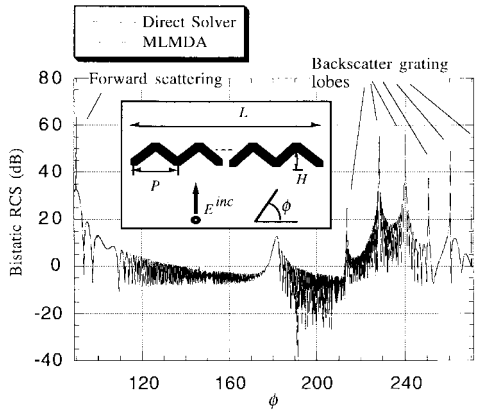


Fig. 13. Bistatic RCS of a finite triangular grating of length $L = 594\lambda$, $P = 6\lambda$, and $H = 1.1\lambda$ for normal TM plane wave incidence. Backscatter grating lobes can be observed.

As in the FMM, the AIM separately considers near- and far-field interactions when evaluating a matrix–vector multiply. To compute far-field interactions, sources supported by the scatterer are projected onto a regular grid by matching their multipole moments (up to a certain order) to guarantee the approximate equality of their far fields. Next, the fields at other grid locations produced by these grid-projected currents are evaluated by a 3-D convolution. Knowledge of these fields permits the computation of fields on the scatterer through interpolation. The projection and interpolation operators are represented by sparse matrices, while the convolution can be effected using an FFT. Unfortunately, the near fields radiated by these grid currents do not match those radiated by the original sources. Therefore, near-field interactions are evaluated directly, and corrected for errors introduced by the far-field operator.

For volumetric scatterers, the computational and memory costs associated with the AIM scale as $O(N \log N)$ and $O(N)$, respectively. For surface scatterers, its computational complexity scales as $O(N^{1.5} \log N)$ and its memory requirements as $N^{1.5}$. The computational complexity and memory requirements of the MLFMA are $O(N \log N)$ and, hence, asymptotically scale more favorably than those of the AIM. Nonetheless, the AIM competes with the MLFMA because the FFT butterfly tree is devoid of the complex interpolation and anterpolation operators inherent in MLFMA. Also, the AIM concept is applicable to all problems exhibiting convolutional structure and is easier to implement than MLFMA. As a result, the AIM has been successfully applied to the analysis of scattering from very large three dimensional structures.

XII. Conclusion

We have reviewed fast solution methods for efficiently solving electromagnetic scattering problems. Fast solution methods for electromagnetic scattering problems will have a definite impact in the area of computer-aided design of many technologies that rely on Maxwell's equations.

Even though a matrix–vector multiply for scattering problems only requires $O(N \log N)$ operations both for volume scattering and surface scattering problems, the number of iterations needed remains unpredictable. Therefore, preconditioning techniques for reducing the required number of iterations in iterative methods are urgently needed in solving electromagnetic wave scattering problems. Finally, even though direct solvers with reduced computational complexities are available for volumetric scattering problems, no such solvers exist for surface scatterers, except for colinear (or almost coplanar) structures. Hence, this remains an open problem.

Acknowledgment

The authors would like to thank the National Center for Supercomputing Applications (NCSA) at the University of Illinois, Urbana-Champaign, for the computer time provided.

References

[1] R. F. Harrington, *Field Computation by Moment Method.* Malabar, FL: Krieger Publ., 1982.
[2] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propagat.*, vol. AP-30, pp. 409–418, Mar. 1982.
[3] M. I. Sancer, R. L. McClary, and K. J. Glover, "Electromagnetic computation using parametric geometry," *Electromagne.*, vol. 10, pp. 85–103, 1990.
[4] D. L. Wilkes and C.-C. Cha, "Method of moments solution with parametric curved triangular patches," in *IEEE APS Int. Symp. Dig.*, London, Canada, July 1991, pp. 1512–1515.
[5] E. H. Newman, "Polygonal plate modeling," *Electromagn.*, vol. 10, pp. 65–83, 1990.
[6] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. AP-14, pp. 302–307, May 1966.
[7] S. S. Zivanovic, K. S. Yee, and K. K. Mei, "A subgridding method for the time-domain finite-difference method to solve Maxwell's equations," *IEEE Trans. Microwave Theory Tech.*, vol. 39, pp. 471–479, Mar. 1991.
[8] K. S. Kunz and R. J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics.* Boca Raton, FL: CRC, 1993.
[9] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method.* Boston, MA: Artech House, 1995.
[10] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1990.
[11] J. M. Jin, J. L. Volakis, and J. D. Collins, "A finite element-boundary integral method for scattering and radiation by two- and three-dimensional structures," *IEEE Antennas Propagat. Mag.*, vol. 33, pp. 22–32, June 1991.
[12] J. M. Jin, *The Finite Element Method in Electromagnetics.* New York: Wiley, 1993.
[13] S. W. Lee, H. Ling, and R. C. Chou, "Ray tube integration in shooting and bouncing ray method," *Microwave Opt. Tech. Lett.*, vol. 1, pp. 285–289, Oct. 1988.
[14] W. C. Chew, "Fast algorithms for wave scattering developed at the Electromagnetics Laboratory, University of Illinois," *IEEE Antennas Propagat. Mag.*, vol. 35, pp. 22–32, Aug. 1993.
[15] Y. M. Wang and W. C. Chew, "A recursive T-matrix approach for the solution of electromagnetic scattering by many spheres," *IEEE Trans. Antennas Propagat.*, vol. 41, pp. 1633–1639, Dec. 1993.
[16] W. C. Chew, C. C. Lu, and Y. M. Wang, "Review of efficient computation of three-dimensional scattering of vector electromagnetic waves," *J. Opt. Soc. Amer. A*, vol. 11, pp. 1528–1537, 1994.
[17] C. C. Lu and W. C. Chew, "The use of Huygens' equivalence principle for solving 3-D volume integral equation of scattering," *IEEE Trans. Antennas Propagat.*, vol. 43, pp. 500–507, May 1995.
[18] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation.* New York: Academic, 1984.
[19] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bur. Standards*, Sect. B, vol. 49, pp. 409–436, 1952.
[20] T. K. Sarkar, "On the application of the generalized biconjugate gradient method," *J. Electromagn. Waves Applicat.*, vol. 1, no. 3, pp. 223–242, 1987.

[21] W. C. Chew, *Waves and Fields in Inhomogeneous Media.* New York: Van Nostrand Reinhold, 1990; Piscataway, NJ: IEEE Press, 1995 (reprint).
[22] V. Druskin and L. Knizhnerman, "Two polynomial methods of calculating functions of symmetric matrices," *USSR Comput. Mathem. Phys.*, vol. 29, no. 6, pp. 112–121, 1989.
[23] ———, "A spectral semidiscrete method for the numerical solution of 3-D nonstationary problems in electrical prospecting," *Izv. Acad. Sci. USSR: Phys. Solid Earth*, no. 8. pp. 63–74, 1988.
[24] U. Schulz and R. Pregla, "A new technique for the analysis of the dispersion characteristics of planar waveguides," *Arch. Elek. Ubertragung, (AEÜ)* vol. 34, pp. 169–173, 1980.
[25] W. C. Chew, S. Barone, C. Hennessy, and B. Anderson, "Diffraction of axisymmetric waves in a bore hole by bed boundary discontinuities," *Geophys.*, vol. 49, no. 10, pp. 1586–1595, 1984.
[26] Q. H. Liu and W. C. Chew, "Analysis of discontinuities in planar dielectric waveguides: Results and applications," *IEEE Trans. Microwave Theory Tech.*, vol. 39, pp. 422–430, Mar. 1991.
[27] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comp. Phys.*, vol. 114, pp. 185–200, 1994.
[28] W. C. Chew and W. H. Weedon, "A 3-D perfectly matched medium from modified Maxwell's equations with stretched coordinates," *Microwave Opt. Tech. Lett.*, vol. 7, no. 13, pp. 599–604, 1994.
[29] D. S. Katz, E. T. Thiele, and A. Taflove, "Validation and extension to three dimensions of the Berenger PML absorbing boundary condition for FD-TD meshes," *IEEE Microwave Guided Wave Lett.*, vol. 4, pp. 268–270, Aug. 1994.
[30] R. Mittra and U. Pekel, "A new look at the perfectly matched layer (PML) concept for the reflection less absorption of electromagnetic waves," *IEEE Microwave Guided Wave Lett.*, vol. 5, pp. 84–86, Mar. 1995.
[31] E. A. Navarro, C. Wu, P. Y. Chung, and J. Litva, "Application of PML super absorbing boundary condition to nonorthogonal FDTD method," *Electron. Lett.*, vol. 30, no. 20, pp. 1654–1656, 1994.
[32] Z. S. Sacks, D. M. Kingsland, R. Lee, and J.-F. Lee, "A perfectly matched anisotropic absorber for use as an absorbing boundary condition," *IEEE Trans. Antennas Propagat.*, vol. 43, pp. 1460–1463, Dec. 1995.
[33] M. Gribbons, S. K. Lee, and A. C. Cangellaris, "Modification of Berenger's perfectly matched layer for the absorption of electromagnetic waves in layered media," in *11th Annu. Rev. Progress ACES*, Monterey, CA, Mar. 1995, vol. 1, pp. 498–503.
[34] W. V. Andrew, C. A. Balanis, and P. A. Tirkas, "A comparison of the Berenger perfectly matched layer and the Lindman higher-order ABC's for the FDTD method," *IEEE Microwave Guided Wave Lett.*, vol. 5, pp. 192–194, June 1995.
[35] J. Fang and Z. Wu, "Generalized perfectly matched layer—An extension of Berenger's perfectly matched layer boundary condition," *IEEE Microwave Guided Wave Lett.*, vol. 5, pp. 451–453, Dec. 1995.
[36] K. K. Mei, "Unimoment method of solving antenna and scattering problems," *IEEE Trans. Antennas Propagat.*, vol. AP-22, pp. 760–766, Nov. 1974.
[37] S. K. Chang and K. K. Mei, "Application of the unimoment method to electromagnetic scattering of dielectric cylinders," *IEEE Trans. Antennas Propagat.*, vol. AP-24, pp. 35–42, Jan. 1976.
[38] M. A. Morgan and K. K. Mei, "Finite-element computation of scattering by inhomogeneous penetrable bodies of revolution," *IEEE Trans. Antennas Propagat.*, vol. 27, pp. 202–214, Mar. 1979.
[39] J. M. Jin and N. Lu, "The unimoment method applied to elliptical boundaries," *IEEE Trans. Antennas Propagat.*, to be published.
[40] B. McDonald and A. Wexler, "Finite element solution of unbounded field problems," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-20, pp. 841–847, Dec. 1972.
[41] J. M. Jin and V. V. Liepa, "A note on hybrid finite element method for solving scattering problems," *IEEE Trans. Antennas Propagat.*, vol. 36, pp. 1486–1489, Oct. 1988.
[42] M. A. Nasir, W. C. Chew, P. Raghaven, and M. T. Heath, "$O(N^{1.5})$ solution of hybrid FEM problems," in *IEEE AP-S Int. Symp. Dig.*, Seattle, WA, June 1994, pp. 447–450.
[43] A. George, "Nested dissection of a regular finite element mesh," *SIAM J. Numer. Anal.*, vol. 10, pp. 345–363, 1973.
[44] V. Rokhlin, "Rapid solution of integral equations of scattering theory in two dimensions," *J. Comput. Phys.*, vol. 36, no. 2, pp. 414–439, 1990.
[45] C. C. Lu and W. C. Chew, "A fast algorithm for solving hybrid integral equation," *Inst. Elect. Eng. Proc. Pt. H*, vol. 140, no. 6, pp. 455–460, Dec. 1993.
[46] N. Lu and J. M. Jin, "Application of FMM to finite element-boundary integral solution of scattering problems," *IEEE Trans. Antennas Propagat.*, vol. 44, pp. 781–786, June 1996.
[47] A. K. Dominek, H. T. Shamansky, and N. Wang, "Scattering from three-dimensional cracks," *IEEE. Trans. Antennas Propagat.*, vol. 37, pp. 586–591, May 1989.
[48] A. Bayliss, C. I. Goldstein, and E. Turkel, "On accuracy conditions for the numerical computation of waves," *J. Computat. Phys.*, vol. 59, pp. 396–404, 1985.
[49] R. Lee and A. C. Cangellaris, "A study of discretization error in the finite element approximation of wave solution," *IEEE Trans. Antennas Propagat.*, vol. 40, pp. 542–549, May 1992.
[50] W. R. Scott, Jr., "Errors due to spatial discretization and numerical precision in the finite-element method," *IEEE Trans. Antennas Propagat.*, vol. 42, pp. 1565–1569, Nov. 1994.
[51] T. Deveze, L. Beaulieu, and W. Tabbara, "A fourth-order scheme for the FDTD algorithm applied to Maxwell's equations," in *IEEE AP-S Int. Symp. Dig.*, Chicago, IL, July 1992, pp. 346–349.
[52] C. W. Manry, S. L. Broschat, and J. B. Schneider,"Higher-order FDTD methods for large problems," *J. Appl. Comp. Electromagn. Soc.*, vol. 10, no. 2, pp. 17–29, 1995.
[53] V. Simoncini and E. Gallapoulos, "An iterative method for nonsymmetric systems with multiple right-hand sides," *SIAM J. Sci. Comput.*, vol. 16, pp. 917–933, 1995.
[54] F. X. Canning, "Transformations that produce a sparse moment matrix," *J. Electromagn. Waves Applicat.*, vol. 4, pp. 983–993, 1990.
[55] A. Boag and R. Mittra, "Complex multipole beam approach to electromagnetic scattering problems," *IEEE Trans. Antennas Propagat.*, vol. 42, pp. 366–372, Mar. 1994.
[56] H. Kim and H. Ling, "On the application of fast wavelet transform to the integral-equation solution of electromagnetic scattering problems," *Microwave Opt. Tech. Lett.*, vol. 6, no. 3, pp. 168–173, 1993.
[57] B. Z. Steinberg and Y. Leviatan, "On the use of wavelet expansions in the method of moments," *IEEE Trans. Antennas Propagat.*, vol. 41, pp. 610–619, May 1993.
[58] R. L. Wagner, G. P. Otto, and W. C. Chew, "Fast waveguide mode computation using wavelet-like basis functions," *IEEE Microwave Guided Wave Lett.*, vol. 3, pp. 208–210, July 1993.
[59] K. Sabetfakhri and L. P. B. Katehi, "Analysis of integrated millimeter-wave and submillimeter-wave waveguides using orthonormal wavelet expansion," *IEEE Trans. Microwave Theory Tech.*, vol. 42, pp. 2412–2422, Dec. 1994.
[60] G. Wang, G. Pan, and B. K. Gilbert, "A hybrid wavelet expansion and boundary element analysis for multiconductor transmission lines in multilayered dielectric media," *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 664–674, Mar. 1995.
[61] R. L. Wagner and W. C. Chew, "A study of wavelets for the solution of electromagnetic integral equations," *IEEE Trans. Antennas Propagat.*, vol. 43, pp. 802–810, Aug. 1995.
[62] Y. Leviatan and A. Boag, "Analysis of electromagnetic scattering from dielectric cylinders using a multifilament current model," *IEEE Trans. Antennas Propagat.*, vol. 35, pp. 1119–1127, Oct. 1987.
[63] R. Kastner, "On matrix partitioning, the SMW technique and the 'add-on' method," in *IEEE AP-S Int. Symp. Dig.*, San Jose, CA, June 1989, pp. 15–18.
[64] C. Hafner, *The Generalized Multipole Techniques for Computational Electromagnetics.* London, U.K.: Artech House, 1990.
[65] J. Zheng and S. Kiener, "Computation of big bodies with GMT: comparison between 2-D and 3-D cylinders," in *IEEE APS Int. Symp. Dig.*, London, Canada, July 1991, pp. 894–897.
[66] D. T. Borup and O. P. Gandhi, "Fast-Fourier transform method for calculation of SAR distributions in finely discretized inhomogeneous models of biological bodies," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-32, pp. 355–360, Apr. 1984.
[67] C. Y. Shen, K. J. Glover, M. I. Sancer, and A. D. Varvatsis, "The discrete Fourier transform method of solving differential-integral equations in scattering theory," *IEEE Trans. Antennas Propagat.*, vol. 37, pp. 1032–1041, Aug. 1989.
[68] P. Zwamborn and P. M. van den Berg, "The three-dimensional weak form of the conjugate gradient FFT method for solving scattering problems," *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 1757–1766, Sept. 1992.
[69] M. F. Catedra, E. Gago, and L. Nuno, "A numerical scheme to obtain the RCS of three-dimensional bodies of size using the conjugate gradient method and the fast Fourier transform," *IEEE Trans. Antennas Propagat.*, vol. 37, pp. 528–537, May 1989.
[70] H. Gan and W. C. Chew, "A discrete BCG–FFT algorithm for solving 3-D inhomogeneous scatterer problems," *J. Electromagn. Waves Applicat.*, vol. 9, no. 10, pp. 1339–1357, 1995.
[71] W. C. Chew, J. H. Lin, and X. G. Yang, "An FFT T-matrix method

for 3D microwave scattering solution from random discrete scatterers," *Microwave Opt. Tech. Lett.*, vol. 9, no. 4, pp. 194–196, July 1995.
[72] J. H. Lin and W. C. Chew, "BiCG–FFT T-matrix method for solving for the scattering solution from inhomogeneous bodies," *IEEE Trans. Microwave Theory Tech.*, to be published.
[73] ——, "A comparison of the CG–FFT method and the recursive T-matrix algorithm," in *IEEE AP-S Int. Symp. Dig.*, Chicago, IL, July 1992, pp. 1591–1594.
[74] J. J. Mallorqui and M. Rodrigeuz, "GRATMA method for biomedical applications: Comparison with the CGM-FFT," in *IEEE AP-S Int. Symp. Dig.*, Newport Beach, CA, June 1995, pp. 1581–1583.
[75] K. Lumme and J. Rahola, "Light scattering by porous dust particles in the discrete-dipole approximation," *Astrophys. J.*, vol. 425, pp. 653–667, 1994.
[76] C. H. Chan and L. Tsang, "A sparse-matrix canonical-grid method for scattering by many scatterers," *Microwave Opt. Tech. Lett.*, vol. 8, no. 2, pp. 114–118, Feb. 1995.
[77] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "A fast integral-equation solver for electromagnetic scattering problems," in *IEEE AP-S Int. Symp. Dig.*, Seattle, WA, June 1994, pp. 416–419.
[78] K. D. Paulsen and D. R. Lynch, "Elimination of vector parasites in finite element Maxwell solutions," *IEEE Trans. Microwave Theory Tech.*, vol. 39, pp. 395–404, Mar. 1991.
[79] I. D. Mayergorz and J. D'Angelo, "A new point of view on the mathematical structure of Maxwell's equations," *IEEE Trans. Magn.*, vol. 29, pp. 1315–1320, Mar. 1993.
[80] B. Alpert, G. Beylkin, R. R. Coifman, and V. Rokhlin, "Wavelet-like bases for the fast solution of second-kind integral equations," *SIAM J. Sci. Comput.*, vol. 14, no. 1, pp. 159–184, 1993.
[81] J. J. Benedetto and M. W. Frazier, *Wavelets: Mathematics and Applications.* Boca Raton, FL: CRC, 1994.
[82] S. Jaffard, "Wavelet methods for fast resolution of elliptic problems," *SIAM J. Numer. Anal.*, vol. 29, no. 4, pp. 965–986, 1992.
[83] R. R. Coifman and Y. Meyer, "Remarques sur l'analyze de Fourier \'a fenêtre," *C. R. Acad. Sci. Paris*, t. 312, série I, pp. 259–261, 1991.
[84] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communicat. Pure Appl. Mathem.*, vol. 41, pp. 909–996, 1988.
[85] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propagat. Mag.*, vol. 35, pp. 7–12, June 1993.
[86] J. M. Song and W. C. Chew, "Fast multipole method solution using parametric geometry," *Microwave Opt. Tech. Lett.*, vol. 7, no. 16, pp. 760–765, Nov. 1994.
[87] R. L. Wagner and W. C. Chew, "A ray-propagation fast multipole algorithm," *Microwave Opt. Tech. Lett.*, vol. 7, no. 10, pp. 435–438, July 1994.
[88] R. Coifman, V. Rokhlin, and S. Wandzura, "Faster single-stage multipole method for the wave equation," in *10th Annu. Rev. Progress ACES*, Monterey, CA, Mar. 1994, vol. 1, pp. 19–24.
[89] C. C. Lu and W. C. Chew, "Fast far-field approximation for calculating the RCS of large objects," *Microwave Opt. Tech. Lett.*, vol. 8, no. 5, Apr. 1995.
[90] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernels," *Comp. Phys. Communicat.*, vol. 65, pp. 24–38, 1991.
[91] C. C. Lu and W. C. Chew, "A multilevel algorithm for solving boundary-value scattering," *Microwave Opt. Tech. Lett.*, vol. 7, no. 10, pp. 466–470, July 1994.
[92] B. Dembart and E. Yip, "A 3-D moment method code based on fast multipole," in *URSI Radio Sci. Meet. Dig.*, Seattle, WA, June 1994, p. 23.
[93] J. M. Song 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, Sept. 1995.
[94] J. R. Mautz and R. F. Harrington, "H-field, E-field, and combined field solution for conducting bodies of revolution," *Arch. Elektron. Übertragungstech (AEÜ)*, vol. 32, no. 4, pp. 157–164, 1978.
[95] E. Michielssen and A. Boag, "Multilevel evaluation of electromagnetic fields for the rapid solution of scattering problems," *Microwave Opt. Tech. Lett.*, vol. 7, no. 17, pp. 790–795, Dec. 1994.
[96] ——, "A multilevel matrix decomposition algorithm for analyzing scattering from large structures," in *11th Annu. Rev. Progress ACES*, Monterey, CA, Mar. 1995, pp. 614–620.
[97] O. M. Bucci and G. Franceschetti, "On the degrees of freedom of scattered fields," *IEEE Trans. Antennas Propagat.*, vol. 37, pp. 918–926, July 1989.
[98] E. Michielssen and W. C. Chew, "Fast integral equation solver using plane-wave basis representation along the steepest descent path," in *URSI Radio Sci. Meet. Dig.*, Newport Beach, CA, June 1995, p. 301.
[99] ——, "The fast steepest descent path algorithm for analyzing scattering from two-dimensional scatterers," *Radio Sci.*, to be published.
[100] W. C. Chew and C. C. Lu, "A fast recursive algorithm to compute the wave scattering solution of a large strip," *J. Comput. Phys.*, vol. 107, pp. 378–387, 1993.
[101] K. Pak, C. H. Chan, and L. Tsang, "A SMFSIA method for the electromagnetic scattering from a two-dimensional (3-D scattering problem) perfectly conducting random rough surface," in *IEEE AP-S Int. Symp. Dig.*, Seattle, WA, June 1994, pp. 451–453.
[102] E. Michielssen, A. Boag, and W. Chew, "Scattering from large elongated objects: Direct solution in $O(N \log^2 N)$ operations," in *Symp. Electromagn. Theory*, St. Petersburg, Russia, May 1995, pp. 464–467.
[103] J. R. Phillips and J. K. White, "Efficient capacitance computation of 3-D structures using generalized pre-corrected FFT methods," in *Proc. 3rd Topical Meet. Elect. Performance Elect. Packaging*, Monterey, CA, Nov. 2–4, 1994, pp. 253–256.

Weng Cho Chew (S'79–M'80–SM'86–F'93), for photograph and biography, see p. 245 of the February 1997 issue of this TRANSACTIONS.

Jian-Ming Jin (S'87–M'89–SM'94) received the B.S. and M.S. degrees in applied physics from Nanjing University, China, in 1982 and 1984, respectively, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1989.

He joined the faculty of the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign in 1993, after working as a Senior Scientist at Otsuka Electronics, Inc., Fort Collins, CO. He has published more than 40 articles in refereed journals, authored the book, *The Finite Element Method in Electromagnetics* (New York: Wiley, 1993), and co-authored another book, *Computation of Special Functions* (New York: Wiley, 1996). He is currently an Associate Editor of the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION. His current research interests include computational electromagnetics, scattering and antenna analysis, electromagnetic compatibility, and magnetic resonance imaging.

Dr. Jin is a member of Commission B or USNC/URSI and Tau Beta Pi. He is a recipient of the 1994 National Science Foundation Young Investigator Award and the 1995 Office of Naval Research Young Investigator Award.

Cai-Cheng Lu (S'93–M'95) was born in Hubei, China, on October 12, 1962. He received the B.S. and M.S. degrees, both in electrical engineering, from Beijing University of Aeronautics and Astronautics, China, in 1983 and 1986, respectively, and the Ph.D. degree from University of Illinois at Urbana-Champaign, in 1995.

Currently, he is working as a Research Scientist at the Center for Computational Electromagnetics at the University of Illinois at Urbana-Champaign. He has been working on developing efficient algorithms for the simulation of wave interactions with complex structures. His interests are in the computational electromagnetics, scattering, and inverse scattering.

Dr. Lu is a member of Phi Kappa Phi.

Eric Michielssen (M'95), for photograph and biography, see this issue, p. 353.

Jiming M. Song (S'92–M'95), for photograph and biography, see p. 245 of the February 1997 issue of this TRANSACTIONS.

CHAPTER 2

FAST MULTIPOLE METHOD

2.1 Introduction

This chapter is designed to give the reader a better understanding of some of the intricate details involved in the fast multipole method (FMM). FMM has been compared to a telephone switching network [5] in which hubs are used to transmit information rather than direct lines connecting each individual telephone. This greatly reduces the number of wires that must connect each telephone. The method can also be compared to a postal service process in which letters are passed between individual mailboxes. By creating a complex process including stations and special transportation carriers, multidestinational letters can be sent from one box to multiple boxes very efficiently compared to a brute force method of hand delivering to individual boxes. The process involves collection, sorting, bulk transportation, resorting, and distribution. The structure of the process allows for the efficiency. In FMM the structural process includes aggregation, translation, and disaggregation. When FMM is applied to a scattering problem the purpose is to reduce the computational cost of a matrix vector product. A scattering problem is formed with basis functions whose coefficients are unknowns. Each of these basis functions interacts with the others based on a known incident wave. If there are N basis functions, or unknowns (as we call their coefficients), an NxN matrix will describe the interactions between each basis function. The computational load of a matrix vector product is $O(N^2)$ and FMM allows this to be reduced to $O(N^{1.5})$. The critical step in FMM is the diagonalization of the translation matrix which Rokhlin [6] first realized. This translation step is similar to the mass transportation of grouped postal letters between post offices. This chapter will describe the terminology of FMM.

2.2 Basic Process

In the method of moments (MoM), an integral equation is discretized and can be written as a matrix-vector product. The matrix must be filled with the proper elements and then inverted to yield the solution vector. Direct inversion is an N^3 process, but by using a method like the conjugate gradient method it can be reduced to $O(N^2)$. Of course, for large-scale problems, this order is too high and must be reduced. The iterative approach of the conjugate gradient method in solving for the unknowns also requires many matrix-vector products. FMM is an enabling technology to reduce the computational workload of the matrix-vector product associated with these electromagnetic problems. As will be discussed in Chapter 3, MLFMA is the miracle algorithm that extends FMM and creates a truly enabling technology for large-scale EM problems.

Since the idea of MoM is to compute the interaction of each source with itself and every other source, FMM promises to streamline this process by allowing faster mathematical interactions. To understand how the basic building block of FMM differs from MoM, it is useful to visualize two collections of particles, each enclosed by a circle (in 2-D) as shown in Figure 2.1. In the spirit of MoM, each particle must interact with every other particle and itself. This is the well-known two-step process of filling and solving the impedance matrix. In the illustration, MoM would fill the matrix by interacting each point with itself and all the other points. This is done by starting with a point (like i_1) and computing the interaction with $i_1, \ldots, j_1, j_2, j_3, \ldots$, where the first three interactions with j-points in the right side box are illustrated with dotted arrows pointing to j_1, j_2, and j_3. The self-interaction terms fall along the diagonal of the matrix. In FMM, great savings are realized by eliminating the need to calculate all of these interactions directly.

The process to achieve massive computational savings requires three steps: aggregation, translation, and disaggregation. Let us call the left box of Figure 2.1, containing nine points, the source group and the right one containing ten points the field group. Aggregation involves the collection of all the source particles to the center of the source group, and this collection can be used to represent outgoing waves that appear to emanate from the center of the group. These waves are valid outside a circle enclosing all the aggregated sources. The smallest circle guaranteed to enclose all particles in a box is the one that intersects each of the box corners.

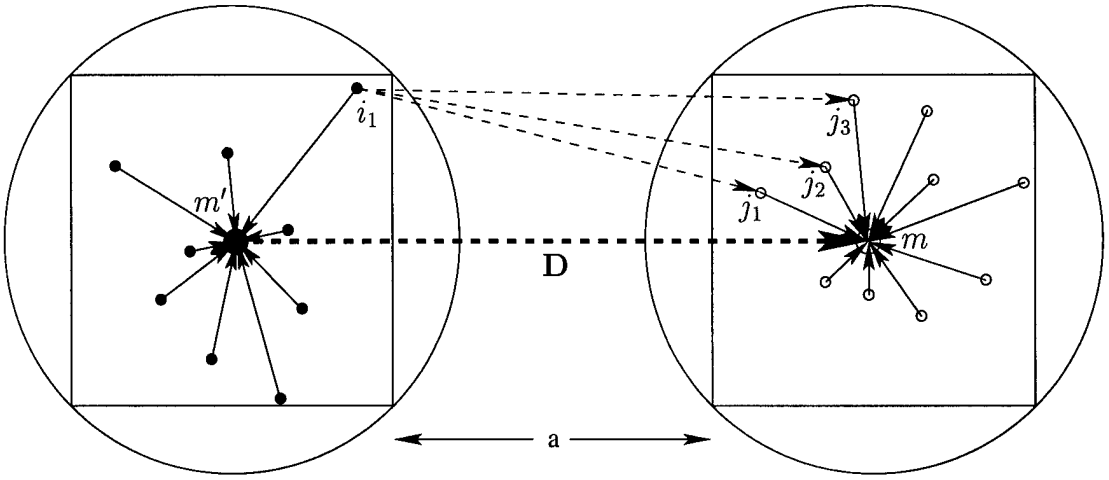


Figure 2.1 Two-dimensional source points and field points in aggregation, translations, and disaggregation. The square on the left contains the source points while the square on the right contains the field points. The three-step process includes the aggregation of nine sources from the left, translation of their multipole representation, and disaggregation to ten field points on the right. A single buffer square separates these two squares. The translation goes between the centers of the source groups and field groups and is shown with the vector **D**.

The outgoing waves are then translated to a field group center and distributed to each field point. Translation takes the radiation pattern of the sources and converts it to incoming waves valid inside a field group. This incoming pattern is used to represent the effect of each source on each field point. The distribution of the information to field points is the last step of disaggregation.

The addition theorem is fundamental in the upcoming mathematics. In order not to violate the addition theorem, the circle enclosing the source points cannot intersect the one around the field points. If it did, any particles found in the intersection region could not be considered due to this violation. For this reason, all boxes touching a source box including the source box itself must be calculated in the traditional way, using MoM. These neighboring boxes are sometimes called buffer boxes. In Figure 2.1, the buffer square of side length a is not shown but separates the same-size left and right boxes.

Near interactions done in the traditional MoM way can be reduced by decreasing the box sizes down to a minimum box size determined by accuracy constraints. The points, $i_1, \ldots, j_1, \ldots$, discussed above, actually represent the common edge centers of the basis functions. RWG (Rao, Wilton, and Glisson) rooftop triangular

basis functions are used to represent the geometry of the surface scatterers being solved. The center of the common edge between two adjoining triangles forming an RWG basis function determines into which box it belongs. The smallest box generally has a side length larger than $\lambda/4$ where λ is the wavelength, but depending on target discretization, it may be better to limit the size to 1.5 times the average edge length of the triangle basis functions. Average edge lengths are generally around 0.1λ, but edge lengths in general can vary between typical values of 0.02λ and 0.2λ. If the geometry discretization is too coarse, the surface currents cannot be accurately represented. If they are too fine, we run into low frequency breakdown problems along with excessive unknowns to solve [5].

In summary, a target is discretized with an appropriate basis function. These basis functions are sorted into groups. The basis functions in and around a group are calculated using MoM. All the other interactions are calculated using aggregation, translation, and disaggregation. Using this three-step process, the computational complexity is reduced to $O(N^{1.5})$ [5].

2.3 Aggregation

The grouping of sources in the first step of FMM is called aggregation. Aggregation is simply the summation of all sources into a radiation pattern emanating from the center of each group. This far field radiation pattern represents the sum effect of the sources contained in each box. This far field pattern is valid outside of the circle (2-D) or sphere (3-D) containing the source square (2-D) or cube (3-D). Of course, the number of samples required to effectively represent these far field patterns depends on the diameter of the circle or sphere enclosing the sources. More samples are required for larger boxes to capture a richer pattern. As will be discussed next, each unique radiation pattern is used to translate information to the appropriate field boxes.

2.4 Translation

In FMM for 3-D, there are at most $3^3 - 1$ cubes touching any given box. For a given source box, including itself, there are a maximum of 27 boxes (one buffer box case) requiring near interaction calculations. The rest of the boxes beyond the buffer boxes use the process of translation to account for the interactions.

Translation is the bridge over which outgoing waves are converted into incoming waves. Since this takes a far-field radiation pattern and changes it into incoming waves, it is often given the name 'outgoing to incoming' (O2I). It is actually a far-field to near-field transform function. This stage is based on the addition theorem and can be regarded as a dense or full matrix operation. However, through careful manipulation, this dense matrix can be diagonalized. The diagonalization of the translation operation is what gives the method the desired computational acceleration [5].

Fundamental to FMM is the addition theorem. Since real world large scale problems in EM generally require a 3-D approach, the addition theorem for the Green's function [5] in 3-D is given by

$$\frac{e^{ik|\mathbf{D}+\mathbf{d}|}}{|\mathbf{D}+\mathbf{d}|} = ik\sum_{l=0}^{\infty}(-1)^l(2l+1)j_l(kd)h_l^{(1)}(kD)P_l(\hat{\mathbf{d}}\cdot\hat{\mathbf{D}}). \tag{2.1}$$

In this equation $j_l(kd)$ and $h_l^{(1)}(kD)$ are spherical functions of the first kind (Bessel and Hankel, respectively) and $P_l(\hat{\mathbf{d}}\cdot\hat{\mathbf{D}})$ is the l-th order Legendre polynomial. Their arguments will be defined shortly but are based on the positions of the source and field points. Using the identity,

$$j_l(kd)P_l(\hat{\mathbf{d}}\cdot\hat{\mathbf{D}}) = \frac{1}{4\pi i^l}\int_0^{2\pi}\int_0^{\pi} e^{i\mathbf{k}\cdot\mathbf{d}}P_l(\cos(\alpha))\sin(\alpha)d\alpha d\beta \tag{2.2}$$

where $\cos(\alpha) = \hat{\mathbf{k}}\cdot\hat{\mathbf{D}}$ and $\hat{\mathbf{k}} = \cos(\beta)\sin(\alpha)\hat{\mathbf{x}} + \sin(\beta)\sin(\alpha)\hat{\mathbf{y}} + \cos(\alpha)\hat{\mathbf{z}}$, we can substitute Equation (2.2) into Equation (2.1) to get

$$\frac{e^{ik|\mathbf{D}+\mathbf{d}|}}{|\mathbf{D}+\mathbf{d}|} \approx \frac{ik}{4\pi}\int_0^{2\pi}\int_0^{\pi} e^{i\mathbf{k}\cdot\mathbf{d}}\sin(\alpha)\sum_{l=0}^{L}(-1)^l(2l+1)h_l^{(1)}(kD)P_l(\cos(\alpha))d\alpha d\beta \tag{2.3}$$

where the integration has been swapped with the summation, which has been truncated to L. With these equations in place, we can apply them to a source group and a field group. In applying Equation (2.3), it is appropriate to illustrate how a single source point is translated to a field point. In Figure 2.2, the vectors relating the source and field points to the previous equations are given as $\mathbf{r}_{ji} = \mathbf{r}_{m'i} + \mathbf{r}_{mm'} + \mathbf{r}_{jm}$, $\mathbf{D} = \mathbf{r}_{mm'}$, and $\mathbf{d} = \mathbf{r}_{m'i} + \mathbf{r}_{jm}$. Therefore, $\mathbf{D}+\mathbf{d} = \mathbf{r}_{ji}$ from Equation (2.3). This figure also shows the aggregation of a point i to the center m' and the translation from m' to m and then disaggregation from m to the field point j.

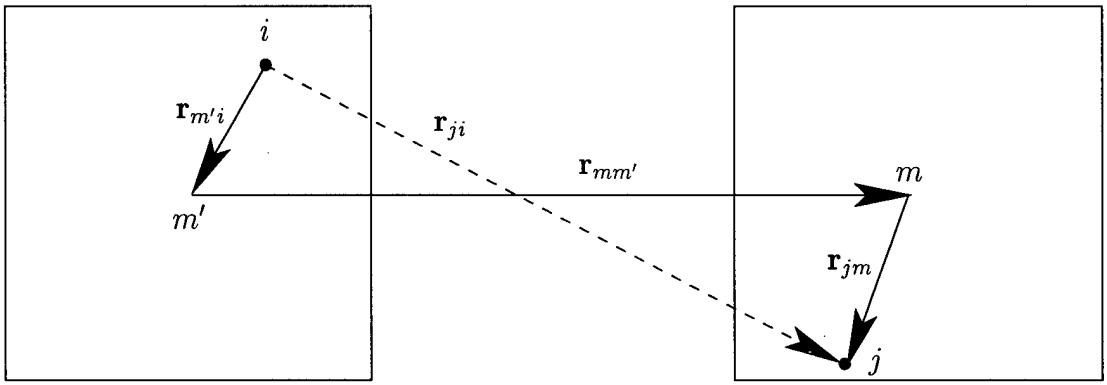


Figure 2.2 Source point and field point vectors in two boxes. The box on the left contains a source point i while the box on the right contains a field point j. Two paths from the source point to the field point are shown. The trivector path from i to j is used in FMM.

We can now give the scalar Green's function in terms of the translator, $T_{mm'}$, as

$$\frac{e^{ikr_{ji}}}{r_{ji}} = \int_0^{2\pi} \int_0^{\pi} \sin(\alpha) e^{i\mathbf{k}\cdot(\mathbf{r}_{jm}+\mathbf{r}_{m'i})} T_{mm'} d\alpha d\beta \tag{2.4}$$

where

$$T_{mm'} = \frac{ik}{4\pi} \sum_{l=0}^{L} i^l (2l+1) h_l^{(1)}(kr_{mm'}) P_l(\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}_{mm'}). \tag{2.5}$$

The integration is performed over the unit sphere and α corresponds to θ while β is the integration around ϕ. The integration over the unit sphere is performed using at least $2L^2$ Gaussian quadrature points. In implementing the integration, two processes are used to simplify the integral. First, the translation vector $\mathbf{D} = \mathbf{r}_{mm'}$ is aligned with the $\hat{\mathbf{z}}$-axis such that $\hat{\mathbf{k}} \cdot \hat{\mathbf{r}}_{mm'} = \cos(\theta)$. Secondly, the $\sin(\alpha)$ is removed by transforming the integration over $d\alpha = d\theta$ by letting $u = \cos(\alpha)$ and $du = -\sin(\alpha)d\alpha$. Equation (2.4) then becomes

$$\frac{e^{ikr_{ji}}}{r_{ji}} = \int_0^{2\pi} \int_{-1}^{1} e^{i\mathbf{k}\cdot(\mathbf{r}_{jm}+\mathbf{r}_{m'i})} T_{mm'} du d\beta \tag{2.6}$$

and Equation (2.5) becomes

$$T_{mm'} = \frac{ik}{4\pi} \sum_{l=0}^{L} i^l (2l+1) h_l^{(1)}(kr_{mm'}) P_l(u) \tag{2.7}$$

The integration over the new variable u involves the multiplication of an L-th order Legendre polynomial with an exponential function. The order of this part of the integrand can be considered to be approximately $2L$ because it is bandlimited, and Gaussian quadrature allows for exact integration of a signal of order $2L$ using $L+1$ Gaussian points. The integration over $\beta = \phi$ is over a complete period and requires $2L$ points. Thus, in total, $2L^2 + 2L$ points are required for the integration. It is important to note that the translator $T_{mm'}$ is independent of the source and field point vectors. This independence allows for a diagonalized translation operator. The translated radiation patterns are then used to interact with the individual field points.

2.5 Disaggregation

The disaggregation process is usually done at the same time as the translation operation. In this way, the effect of the radiation patterns of grouped sources is applied to individual field points. Every box, in turn, becomes a field box receiving the information from distant source boxes through translation. For a more comprehensive treatment of FMM technology see [5]. In the end, each point corresponding to a basis function can be found and the resulting coefficients used to calculate important target characteristics such as RCS. In FMM, the computational complexity is $O(N^{1.5})$ [5]. To improve the computational complexity, it is necessary to take the algorithm to a new level.

CHAPTER 3

MULTILEVEL FAST MULTIPOLE ALGORITHM

3.1 Introduction

This chapter extends FMM described in the previous chapter to a multilevel algorithm named the multilevel fast multipole algorithm (MLFMA). An integral part of the extension to a multilevel scheme requires both interpolation and anterpolation of the radiation patterns between differently sized boxes. The size of the box determines the levels. Tree structure, levels, upward pass, downward pass, and filtering (including interpolation and anterpolation) will be discussed. The advantage of extending FMM for the Helmholtz equation to multiple levels is the reduction in computational complexity from $O(N^{1.5})$ to $O(N \log N)$ [5], where N represents the number of unknowns describing the target.

3.2 Tree Structure

To reduce the computational cost, the number of translations used in FMM must be reduced. This requires a tree structure connecting boxes of different sizes. In the case of FMM, the translations are between equally sized boxes. Therefore, the finer the boxes, the more translations must be made. With MLFMA the number of translations can be reduced by aggregating radiation patterns from smaller boxes to form larger box radiation patterns that are translated to larger-size boxes with appropriate separation. This is easiest to see using a 2-D grid as shown in Figure 3.1 where two example source squares, shown as shaded, are translated using MLFMA. In Figure 3.1, the grids have been produced by successive subdivisions. Level 1 includes the four squares produced by the first subdivision. Translations cannot be done on level 1 because all boxes touch each other. Level 2 contains 16 squares and it is the highest level where translations can be performed. In MLFMA, we want to perform translations at the highest level possible. Translation is only possible to squares that are not touching. Level 3 has 64 squares and in this example it will be the lowest level.

In Figure 3.1, we assume that every square contains information that must be related to every other square in the most efficient manner. On the left column, a source square in the lower left part of the grid is shown with 7 translations between level 3 squares. Rather than performing 48 more translations on this level, the pattern from this source box is combined with the neighboring three patterns as shown in the middle diagrams to form a larger square box pattern which only needs to be translated 12 times to the centers of larger level 2. On the right column of Figure 3.1, a source square is translated to 27 level 3 squares. None of these 27 squares qualifies for higher level 2 translations. Using a single buffer square, 27 is the maximum number of translations for a given level from a single square and is $6^2 - 3^2$. However, there are $7^2 - 3^2$ possible translation vectors on a single level from arbitrary source squares. The remaining 28 translations can be performed on level 2 using just 7 translations. Furthermore, these 7 translations include the information aggregated from three other level 3 source squares. Computational savings, through a more efficient translation process, are an important feature of the multilevel algorithm. The reason that more levels are desirable is because lower level translations are more efficient than near interactions computed using MoM.

The center of a box is important and the algorithm revolves around these centers. In a way, the centers can be considered command centers. This analogy fits particularly well with the multilevel aspect of the algorithm. A command center has a commander who is responsible for subordinates. Commanders are also required to communicate directly with colleagues on an equal level and superiors on a higher level. After describing how the command structure is created, the details in a command context will be addressed.

When MLFMA is implemented, the program must first create a tree structure that resembles an organization chart. In 2-D, a quad-tree is created by successive binary cuts in both directions. In 3-D, an oct-tree is created in which a cube around the target is subdivided into eight equal cubes. The number of cubes at any level is given by $2^{\mathbf{N_{dim}}*\mathbf{L_{lev}}}$ where $\mathbf{N_{dim}}$ is the dimension (2 or 3) of the problem and $\mathbf{L_{lev}}$ is the level number. In Figure 2.1 there are $2^{2*3} = 64$ boxes on level 3 of the 2-D space. Since the target must be contained inside the square at level 0, the maximum dimension A of the geometry in Cartesian space is used as the level 0 length. Thus, the length of the lowest level is given by $A/(2^{L_{low}})$, where L_{low} is the number of the lowest level. As described in Section 2.2, this has a minimum

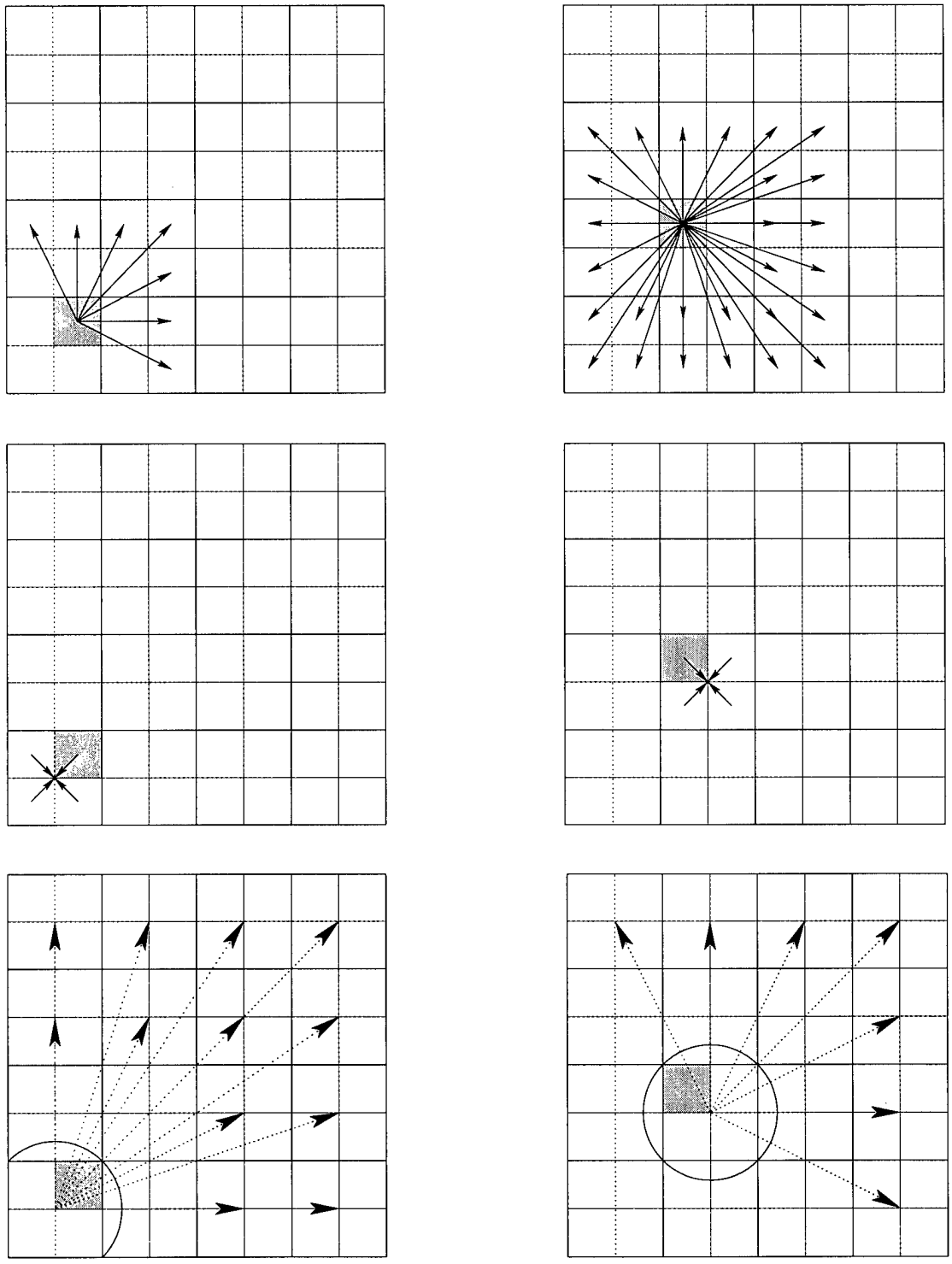


Figure 3.1 Two examples of multilevel translations. The top shows translations on level 3. The middle figures illustrate aggregation of radiation patterns to larger level 2 boxes. The bottom shows level 2 translations.

which should carefully be considered with regards to the average size of the basis functions. Therefore, a maximum level exists. Pushing the tree structure to the maximum level maximizes the computational efficiency of the algorithm.

Since the target bounding box is known, the number of levels can be chosen and the size of the smallest boxes can be found. The basis functions, indexed by spatial points and loosely referred to as particles, are then sorted into these smallest boxes. Naturally, many of the boxes are empty. The tree is pruned of these empty boxes. Each box will have an average number of particles based on the total number of particles divided by the number of filled boxes. The number of particles in any given box will vary. With the assumption that a target's surface is flat as it cuts through a box and that the angle of the cut is a uniform random variable, it is possible to calculate that 1.9 out of 4 squares in the 2-D case will be crossed on the average. In 3-D when a cube is subdivided into 8 cubes at a lower level, empirically we observe that about 4 out of the 8 cubes are filled with particles. Obviously, this can be very geometry dependent and if the number of particles in a cube gets close to 4 or less, pushing additional levels will not continue this trend. In the limit, if every box contains only a single particle, subdividing the box will produce only one single filled box.

With multiple levels and a pruned tree of filled boxes, the command structure and organization is in place. In the context of a command structure, it is possible for a commander to have $2^{N_{dim}}$ filled boxes, but generally only half of these boxes are filled. The next sections will discuss the steps taken on each box.

3.3 Traversing the Tree

Each filled box must be considered in order to capture the total interaction of each particle with every other particle. Recall that the purpose of MLFMA is to accelerate the matrix vector product without requiring the full matrix storage. Rather than causing individual particles to interact with each other, we now have a tree structure in place to handle the interactions. Traversing the tree begins at the lowest level.

Each lowest level command section or box has some particles. The first step is to compute the direct interaction of these particles with themselves and each other. Then each of the neighboring boxes and their particles must be considered.

After accounting for all of these near interactions, the particles in each box are aggregated to the center of the box, which we will refer to as the commander (CC). The commander is then responsible to create an outgoing radiation pattern from the aggregated sources. The CC then must determine which commanders should receive this information through translation. These are the boxes outside of the buffer boxes, which cannot receive higher-level translations.

The upward pass begins by shifting the radiation pattern up one level in the chain of command. This communication requires special upsampling so that the commander's supervisor can integrate the radiation patterns from the subordinate boxes and then create an appropriate radiation pattern for their level. Since the higher level command box has twice the edge length of the subordinate commander boxes, higher-level CCs need more samples to describe a richer radiation pattern. On levels that are above the lowest level, or smallest boxes, near interactions are not issues. The key is to collect the information from the subordinate command boxes and then to translate to appropriate same-level commanders separated in the usual sense. For efficiency, each CC will shift their patterns to their supervisor who will continue the same process. This process of translating and shifting goes until level 2 is reached and the effect of the original sources at the lowest level of command has been provided to the entire space.

In this process, it is easy to see that upper level commanders have more details to work with as they collect the information of subordinates and pass it across the tree or up the tree in the most efficient way. Although the upward pass ends at level 2, the process is not complete. We must now consider the downward pass and its objective to provide the needed information to every lowest level commander for distribution to each of the field particles. In the downward pass, translated information is received and the outgoing radiation patterns are converted to incoming wave patterns and combined with other incoming wave patterns. These incoming wave patterns are then filtered or anterpolated down to the subordinate commanders who combine information from the top levels with translated information from same level or colleague commanders. The resulting patterns are formed and the process of passing the information down the tree continues until the lowest level commanders have the required data to interact with each of their field points. This completes the process of traversing the tree.

In summary, each lowest level CC has facilitated the interaction of each of their

source particles with the field particles of every other lowest level commander. This was accomplished through direct interactions for self and near neighbors, aggregation, translation, upshifting, downshifting, and disaggregation. The algorithm is not difficult to understand, although the mathematics and finer details tend to cloud the big picture. As in command, however, the finer details always remain important.

3.4 Signal Processing

Probably the most important finer detail of MLFMA is signal processing. This is also essential in the principle of command. In the context of MLFMA, signal processing includes the upsampling (interpolation) and downsampling (anterpolation) of radiation patterns. Filtering is necessary because the radiation patterns of finer lower-level boxes do not require as much detail or samples as coarser higher-level boxes. Therefore, when a CC provides a radiation pattern up the chain of command, it must first be interpolated and then shifted to the next level CC. The interpolation allows the higher level CC to receive the pattern with the correct sampling points.

On the downward pass the process of anterpolation allows the lower level CC to receive a downshifted pattern with fewer but an adequate number of sample points. These are easily combined with translated outgoing to incoming patterns from colleague commanders with the same number of samples. This continues down to the lowest level commanders that distribute or disaggregate the incoming wave pattern to each field point.

3.5 Final Details

This chapter has described MLFMA. The detailed mathematics including the incorporation of MoM have been intentionally left out. A good treatment of the subject can be found in the first three chapters of [5] where the derivations for both the electric and magnetic field integral equations are given in detail. Also, the mathematical details of shifting, interpolation, and anterpolation are provided. With this background, it is now possible to understand the error sources of MLFMA. As expected, such a fast method will come with some tradeoffs between speed and accuracy.

CHAPTER 4

ERROR CONTROL

4.1 Introduction

When using MLFMA to produce RCS data, there are several key error sources. Depending on the type of targets and the accuracy required, these errors can be controlled to some degree. Understanding and controlling these errors is the purpose of this chapter. Considering a perfect electric conductor (PEC) target, these errors fall into the following classes: geometry modeling, integral equation discretization, matrix equation solving, and translation operation. The translation operator and its factorization are at the core of the multilevel algorithm and together form an important part to understand. This chapter will be primarily focused on the effect of this translation error and a new approach to error control. The recently developed new approach for 2-D problems [2] has been extended in this research to 3-D problems [7] and will be the main focus of this chapter.

4.2 Geometry Modeling

There are differences between a computer model and the physical model that it represents. Targets with flat surfaces are the easiest to model and usually have the most faithful representation. For example, a cube can be modeled perfectly with just eight points. However, all physical cubes will have rounded edges and corners due to the tolerances of the physical manufacturing process. Depending on the frequency of interest, these differences may be quite insignificant. For curved surface models, a linear representation is an approximation and will therefore be an error source. Since most large-scale problems involve complex structures, there will be a difference between the geometrical representation and the actual target.

These complex targets may be formed by the combination of flat surfaces joined with curved and doubly curved surfaces. When basis functions are chosen based on flat triangular patches, the surface can be meshed carefully to capture the key scattering structures. Of course, narrow tips result in triangles that are too

small or narrow for good computational stability. This means that the geometry modeler must have a clear understanding of the effect of the small details and how to represent them.

Many companies interested in EM modeling hire Ph.D.'s to work on CAD modeling. They apply EM theory in the construction of models that are most suitable to a particular EM code simulation. Constructing the geometry in an optimal way that captures the physics of the problem requires a solid understanding of computational electromagnetics. For curved surface models, higher order basis functions that describe the curvature can be used to more faithfully represent the shape, but the tradeoff is increased complexity. The point is that geometry fidelity and choice of basis functions lead to an important error source. This error source is one of the hardest to quantify.

4.3 Integral Equation Discretization

Another source of error arises from MoM. Since MLFMA is tied to MoM, this particular error source should be mentioned. Integral equation discretization is the process of converting an integral equation into a matrix equation. The way this is generally done is by using a basis function as defined over the geometry surface. Then a testing function is used and integration over the basis functions and testing functions is performed to find the matrix elements in MoM. The error associated with integral equation discretization or converting a continuous integral to a discrete matrix is not studied in more detail in this document.

4.4 Matrix Equation Solving

A discretized integral equation can be represented as a linear system of equations

$$\bar{\mathbf{A}} \cdot \mathbf{x} = \mathbf{b} \tag{4.1}$$

in which the matrix $\bar{\mathbf{A}}$ is the impedance matrix, $\mathbf{x}$ represents the unknown coefficients of the chosen basis functions, and $\mathbf{b}$ is a known vector based on the incident wave. For small MoM problems, the matrix can be inverted and then multiplied by $\mathbf{b}$ on the right hand side of the equation to find $\mathbf{x}$. However, for large problems, the cost of matrix inversion is too high. These systems are solved with iterative

methods such as the conjugate gradient method. In such an iterative method, a solution vector is guessed and the matrix-vector product is compared to the known vector. Using the comparison, a new solution vector is chosen and the process is iterated until the comparison, as measured by what is called the residual error, becomes small enough. Often this residual error is chosen to be 10^{-3}.

This residual error means that the unknown coefficients are only approximate values. The unknown coefficients of the basis functions represent the surface currents. These surface currents are used to calculate the scattered field. The relationship between the error in the scattered field as a function of residual error is very dependent on geometry and aspect angles of both the transmitter and receiver. Fortunately, approximate currents often lead to reasonably accurate RCS values as the integration of the currents for the far field is a smoothing process. Another advantage of MLFMA is that it is matrix free.

4.5 Translation

The hardest error to control finds its root in the heart of FMM. The key to FMM is the diagonalized translation operation. As described in Chapter 2, the translation operator is repeated below in Equation (4.2). The derivation of the translation operator used in 3-D MLFMA is given in many sources [5, 8, 9, 10, 11, 12, 13] and is based on the addition theorem. The series representation of the translation operator must be carefully truncated to avoid excessive error. With insufficient terms, the series is a poor approximation, but with too many terms the divergent nature of the series emerges. Since computational methods require truncation for evaluation and due to the finite precision of floating point numbers, this truncation number must be carefully chosen. The series representation of the translation operator is given as [5]

$$T_{mm'} = \frac{ik}{4\pi} \sum_{l=0}^{L} i^l (2l+1) h_l^{(1)}(kr_{mm'}) P_l(\hat{\mathbf{r}}_{mm'} \cdot \hat{\mathbf{k}}) \tag{4.2}$$

where $h_l^{(1)}(kr_{mm'})$ is a spherical Hankel function of the first kind, $P_l(\hat{\mathbf{r}}_{mm'} \cdot \hat{\mathbf{k}})$ is a Legendre polynomial of order l, $\mathbf{r}_{mm'}$ is the translation vector, and $\hat{\mathbf{k}}$ is a unit vector used for integration over the unit sphere. This is the key series used to approximate the Green's function by integrating over the unit sphere using approximately $2L^2 + 2L$ Gaussian quadrature points and the relationship

$$\frac{e^{ikr_{ji}}}{r_{ji}} = \int d^2\hat{k} e^{i\mathbf{k}\cdot(\mathbf{r}_{jm}+\mathbf{r}_{m'i})} T_{mm'}. \tag{4.3}$$

Figure 4.1 is useful for visualizing the translation of a source point to a field point. This is an orthogonal view of two cubes and an outline of the sphere (diameter=$\sqrt{3}a$) enclosing them. The upper cube with labeled points has been rotated to capture the 3-D perspective. Note that this third cube could fit between the lower two cubes and could be a buffer cube. The **r**-vectors represent the vectors between group (or box) centers, field, and source points. The trailing subscript is the starting point of the vector. Thus, swapping subscripts is the same as negating the vector. Rather than calculating the direct path from the source to the field point $\mathbf{r}_{ji}$, shown as a dashed line, the source point is aggregated to the box center on the left, then translated to the box center on the right, and finally disaggregated to the field point. Mathematically, this path is $\mathbf{r}_{jm}+\mathbf{r}_{mm'}+\mathbf{r}_{m'i}$ and is represented with the solid vectors. The translation does not depend on the positions of the source and field points; however, the truncation of this series in Equation (4.2) is highly dependent on these positions. In this illustration, the source and field points are at opposite corners of their box groups. This produces the highest error in the evaluation of the scalar Green's function given in Equation (4.3). We refer to this worst case as position 81 where the digits represent the corner of the cube where the field and source points are located. Similar worst-case positions include {18, 27, 36, 45, 54, 63, 72}.

It is important to note that a sphere enclosing a source point cannot intersect a sphere enclosing a field point. This would violate the requirements of the addition theorem [14]. Thus, the adjacent boxes, including diagonal boxes, containing field points require a direct evaluation of the interaction. Therefore, the nearest translation is between boxes separated orthogonally by one buffer box and is twice the distance of the box edge length a, as seen in Figure 4.1. The error associated with the Green's function and its approximate representation is due to the position vectors of the source and field points relative to their box centers. Obviously, when their relative positions are the same, the approximation is the best, but when the vectors are maximum and in opposite directions, this leads to the largest error. As illustrated in Figure 4.1, $|\mathbf{r}_{m'i}+\mathbf{r}_{jm}|$ is at a maximum. In 3-D, this maximum length is $\sqrt{3}a$ and there are eight positions of source and field points where this occurs. This maximum distance can be used to predict the truncation number L,

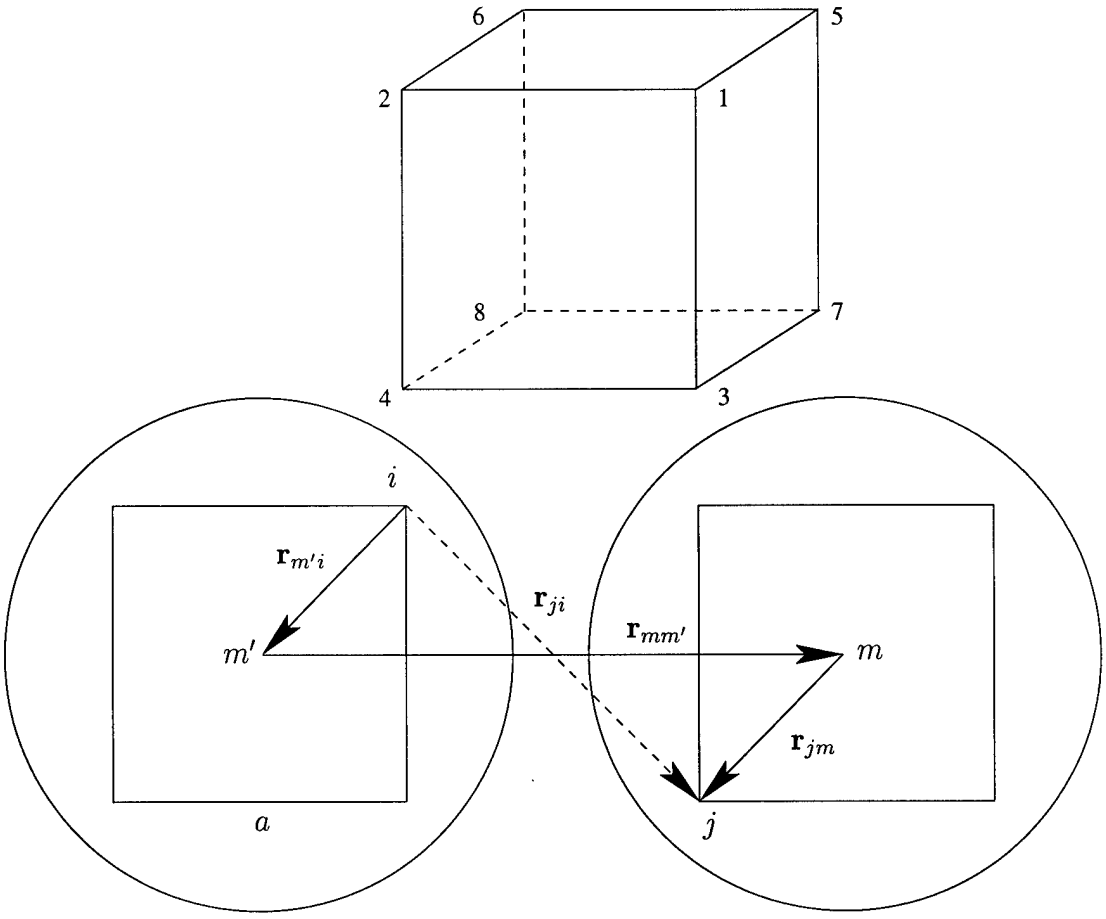


Figure 4.1 An orthogonal view of two cubes and the outlines of the spheres containing them. The top cube with corners labeled from 1-8 gives a perspective view. Source point i is at point 1 and field point j is at point 8. This is referred to as position 81.

in the series of the translator formula in Equation (4.2).

4.5.1 Series truncation

Previous work [5, 13, 15] applied the refined excess bandwidth formula,

$$L \approx kd + 1.8(d_0)^{\frac{2}{3}}(kd)^{\frac{1}{3}}, \tag{4.4}$$

in order to truncate the series for good results. In this formula, $d = |\mathbf{r}_{m'i} + \mathbf{r}_{jm}|$, the wavenumber is k, and d_0 is the desired digits of accuracy. For the worst case, $d = \sqrt{3}a$. It says that, for a given box size, a reasonable truncation L, can be calculated and used to terminate the series.

Figure 4.2 shows the actual error compared to that predicted by the excess bandwidth formula for increasing L. This example, for a fixed $ka = 20$, shows increasing disagreement for $d_0 > 2$ (relative error less than 10^{-2}). We also note that the minimum error occurs when the machine precision accuracy of $d_0 \approx 15$ is used in Equation (4.4). This is a key observation in developing the new approach. The excess bandwidth formula applies when $L < kr_{mm'}$ or, in other words, when the group centers are sufficiently separated relative to the truncation. By increasing the buffer boxes separating two groups, larger truncation values can be chosen. In this plot, one can easily see how initially after a certain number of terms ($L > \sqrt{3}ka > 34$ in this case), the error begins to decrease and after reaching an actual minimum (at $L = 67$), the divergent nature is seen. This also shows that the excess bandwidth formula can predict and achieve certain error levels; however, the desired number of digits of accuracy used in Equation (4.4) is not always achievable and is different from reality when the box separation and box size are small and high accuracy is desired.

Using the excess bandwidth formula in Equation (4.4) and the worst case position vectors in 3-D pointing to field and source points in opposite corners of the box (see Figure 4.1), we can solve for the digits of accuracy as

$$d_0 = \left[\frac{L - \sqrt{3}ka}{2.2(ka)^{\frac{1}{3}}} \right]^{1.5}. \tag{4.5}$$

This equation relates to the convergence of the Bessel function when it is $O(10^{-d_0})$. In the same way that d_0 represents the digits of accuracy with respect to the convergence rate of the Bessel function, using the asymptotic approximation of the spherical Hankel function [16] when $kr_{mm'} \sim O(l)$, it can be shown [5] that the divergence of the Hankel function series when it is $O(10^{+d_1})$ leads to

$$d_1 = \left[\frac{L - kr_{mm'}}{1.8(kr_{mm'})^{\frac{1}{3}}} \right]^{1.5}, \tag{4.6}$$

where d_1 represents the number of digits lost by numerical evaluation. This can be written in terms of the number of buffer boxes n and box size a as

$$d_1 = \left[\frac{L - (n+1)ka}{1.8((n+1)ka)^{\frac{1}{3}}} \right]^{1.5}. \tag{4.7}$$

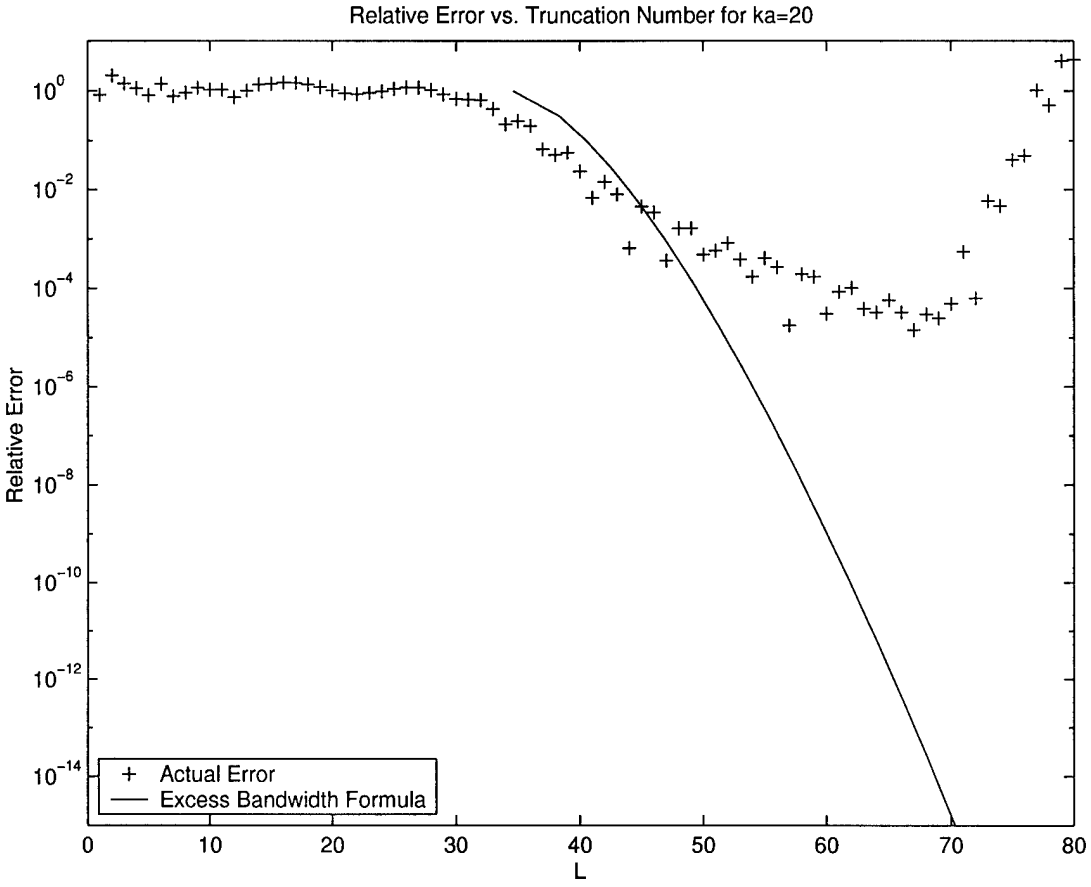


Figure 4.2 Error comparison for fixed $ka = 20$ and changing L. The optimal truncation is $L = 67$ for this one buffer box worst case. When $L > 67$, the series is clearly diverging. We note that theoretically $L > 40$, in this example, is outside of the valid range for applying the excess bandwidth formula although numerically it achieves a better error up to $L = 45$. Assuming that L is chosen using Equation (4.4) with 15 digits accuracy, the actual error does not even achieve 5 digits of accuracy. This is expected since we are also outside of the applicable range of this formula.

By subtracting $d_0 - d_1$, the actual expected digits of accuracy can be found. Since machine precision limits the digits of accuracy to about 15, for small box sizes, the digits lost will be larger when L is increased in Equation (4.7). Also, if the number of buffer boxes n is increased, then d_1 will shrink. Equation (4.4) produces L_{max} based on $d_0 = 15$ as

$$L_{max} = \sqrt{3}ka + 13.2(ka)^{\frac{1}{3}}. \tag{4.8}$$

L_{max} is the truncation number that should produce the minimum possible error. Since $d_0 = 15$ implies a relative error of 10^{-15}, the example in Figure 4.2 shows a large gap between the true minimum error and that predicted by the excess bandwidth formula. If $L > L_{max}$, the error grows due to the finite machine precision and the divergent series embedded in Equation (4.3). Using $L = L_{max}$ in Equation (4.7), we can find $d_{min}^{(n)} = d_0 - d_1$, which is the true digits of accuracy associated with the minimum error for n buffer boxes,

$$d_{min}^{(n)} = 15 - \left[\frac{L_{max} - (n+1)ka}{1.8((n+1)ka)^{\frac{1}{3}}}\right]^{1.5}. \tag{4.9}$$

To capture the effect of lost digits using one buffer box, $n = 1$, leads to a $d_{min}^{(1)}$ that will represent the theoretical error bound,

$$d_{min}^{(1)} = 15 - \left[\frac{L_{max} - 2ka}{2.3(ka)^{\frac{1}{3}}}\right]^{1.5}. \tag{4.10}$$

Obviously, it is best to choose $L \leq L_{max}$ since this represents the error floor. When $(n+1)ka < L < L_{max}$, the new approach is required for precise error control. It is desirable to calculate a better truncation number in this part of the controllable region, and this is what the new approach does. When $L < (n+1)ka$, the new approach uses Equation (4.4) to choose the truncation.

In order to choose the truncation properly in the extended controllable region where the excess bandwidth formula does not achieve the desired error level, the new approach is necessary. In the new approach, we find the truncation number based on a desired error level e_r and the intersection of this error level with the minimum error level and the border where the excess bandwidth formula applies (see [2, 17, 18] and Figure 4.3).

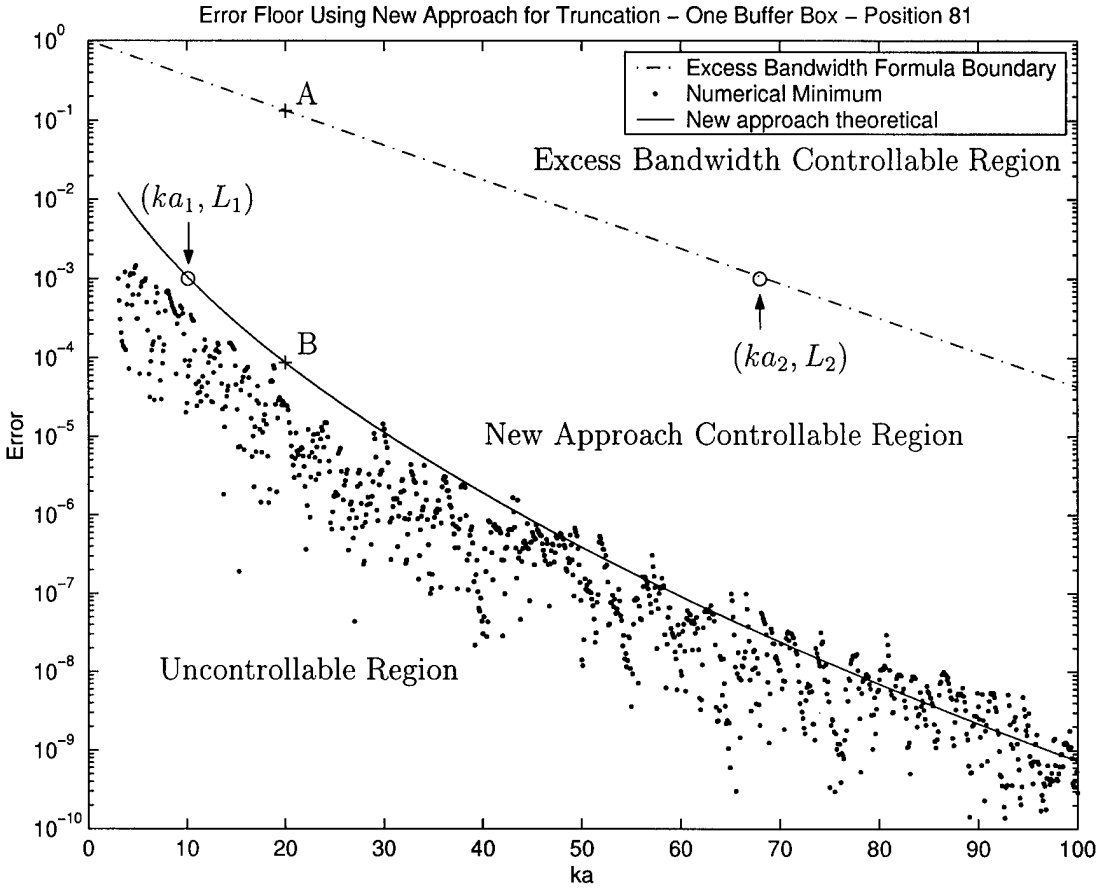


Figure 4.3 Error floor using new approach for truncation and one buffer box. The boundary between the uncontrollable and controllable regions is shown as the new approach theoretical minimum line. The upper dash-dot line is the boundary between the upper region where the error can be controlled using the excess bandwidth formula and the middle region where the new approach must be used for precise error control. For a fixed ka, for example $ka = 20$, the error level can be controlled to levels between point A and point B using the new approach. For an error level above point A, the excess bandwidth formula can control the error. Finally, for this $ka = 20$, we cannot theoretically control the error to levels below point B.

From these two intersection points, (ka, L) pairs are found. For both points, the same desired error level is used to find (ka_1, L_1) and (ka_2, L_2) as illustrated in Figure 4.3 for $e_r = 10^{-3}$. The first ordered pair, on the left, borders the uncontrollable region and is given by

$$ka_1 = \left\lceil \frac{13.2 - 1.8(n+1)^{\frac{1}{3}}\left(15 - \log\frac{1}{e_r}\right)^{\frac{2}{3}}}{n+1-\sqrt{3}} \right\rceil^{1.5} \tag{4.11}$$

and

$$L_1 = \sqrt{3}ka_1 + 13.2(ka_1)^{\frac{1}{3}}. \tag{4.12}$$

These equations are derived by solving Equation (4.9) for ka_1 where the subscript refers to the first ordered pair. Of course, L_{max} from Equation (4.8) must be substituted into Equation (4.9). Above, L_1 is simply L_{max} from Equation (4.8) with ka_1. The second ordered pair is given by

$$ka_2 = \frac{3.2\log\frac{1}{e_r}}{\left(n+1-\sqrt{3}\right)^{1.5}} \tag{4.13}$$

and

$$L_2 = (n+1)ka_2 \tag{4.14}$$

and meets the boundary where the excess bandwidth formula can be used. Equation (4.13) comes from setting $(n+1)ka_2$ equal to the right hand side of Equation (4.4) with $d = \sqrt{3}a_2$. Then the equation is solved for ka_2. Here, L_2 forms the upper boundary of the new approach controllable region. Interpolating between these two points provides an adjusted L necessary to maintain the required relative error level for a given ka between ka_1 and ka_2. Using this new approach, the error can be controlled for 3-D translation operations in FMM and MLFMA.

4.5.2 Results

The new approach allows us to establish the minimum possible error for a given box size. If the desired error is higher than this minimum, the error can be controlled using the new approach. If the desired error is found in the region where $L < (n+1)ka$, the refined excess bandwidth formula given in Equation (4.4)

is used to determine the proper truncation number. When $L > (n+1)ka$, the new approach must be used to control or minimize the error. All of the numerical results discussed here involve the worst case errors in which the source and field points are at opposite corners of their boxes and with only one buffer-box group separation.

In Figure 4.3, we show that the minimum error for position 81 (worst case) can be found by setting $d_0 = 15$ in Equation (4.4). This gives the L_{max} of Equation (4.8) that can be substituted into Equation (4.10) to find the theoretical minimum error. This is labeled in this figure as the 'new approach theoretical.' Numerical results are given where the numerical minimum error is found compared to the theoretical predicted minimum error. Figure 4.3 shows that the numerical minimum results tend to be slightly lower than the theoretical minimum for smaller ka; this is due to the approximate form of the refined excess bandwidth formula. The variation of the true minimum is due to rounding. Since the true minimum was desired, the lowest error was found through numerical experimentation. It matches the new approach theoretical line fairly well. Previously, it has been shown [2] that the excess bandwidth formula applies when $L < (n+1)ka$ where n is the number of buffer-box separation between the groups. This upper-bound line is the boundary between the two controllable regions.

As a function of the distances between box centers, the minimum error for the worst case can be predicted for boxes just outside the buffer box layer. Note that the maximum distance ($\sqrt{12}a$) for the first layer beyond the one buffer box is greater than the two buffer box minimum distance ($3a$). The translations on a level will have lower errors when the ratio of the translation distance to the box size increases. This is shown in Figure 4.4 where the legend represents the number of boxes in each orthogonal direction ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$) to get to the field box from the source box. There are $5^3 - 3^3 = 98$ different translation directions one buffer box away. However, Figure 4.4 shows the relative error associated with these six unique distances.

Using interpolated values of L we can establish the predicted error level inside the extended controllable region where the excess bandwidth formula breaks down. Figure 4.5 shows constant error levels where the excess bandwidth formula is used, and in the region where interpolated values of L are used, the theoretical error level dips downward. This is because the interpolation picks L to be slightly more

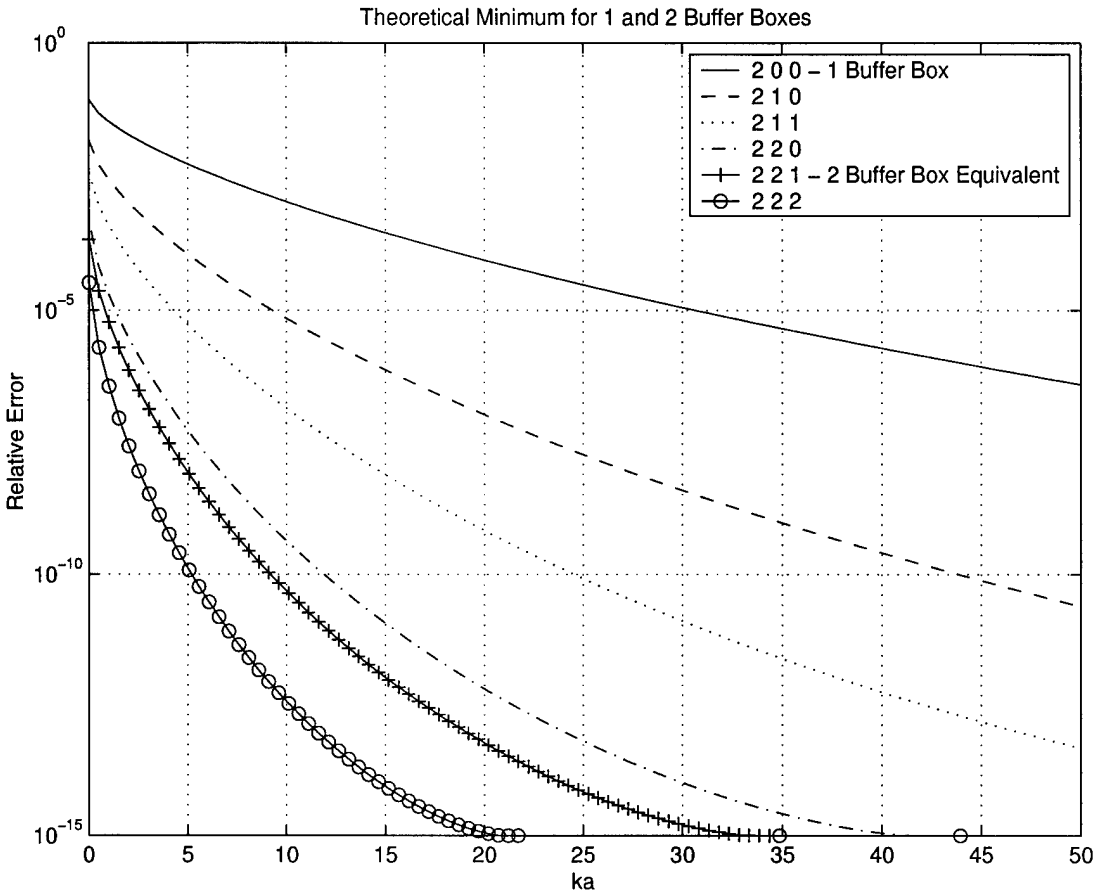


Figure 4.4 Predicted minimum error for different translation distances. There are 98 different translation directions one buffer-box layer away. However, as this plot shows, there are only six unique distances. For example, (2 2 1) corresponds to (**x**, **y**, **z**) and is a distance of three times the side length of the box.

than necessary to achieve the fixed error level.

To illustrate the new approach, Figure 4.6 compares the new approach to the excess bandwidth formula with an arbitrary three digits of desired accuracy. It is easy to see the two boundaries between the three regions of chosen L values for varying ka. When $ka > 67$, we use the excess bandwidth formula. When $ka < 10$, we cannot make the error smaller than the desired error level, so we use L_{max} to achieve the best possible error. In the middle region we use interpolated values of L as described in the formulation. This plot highlights the utility of the new approach in achieving a better error control than the excess bandwidth formula.

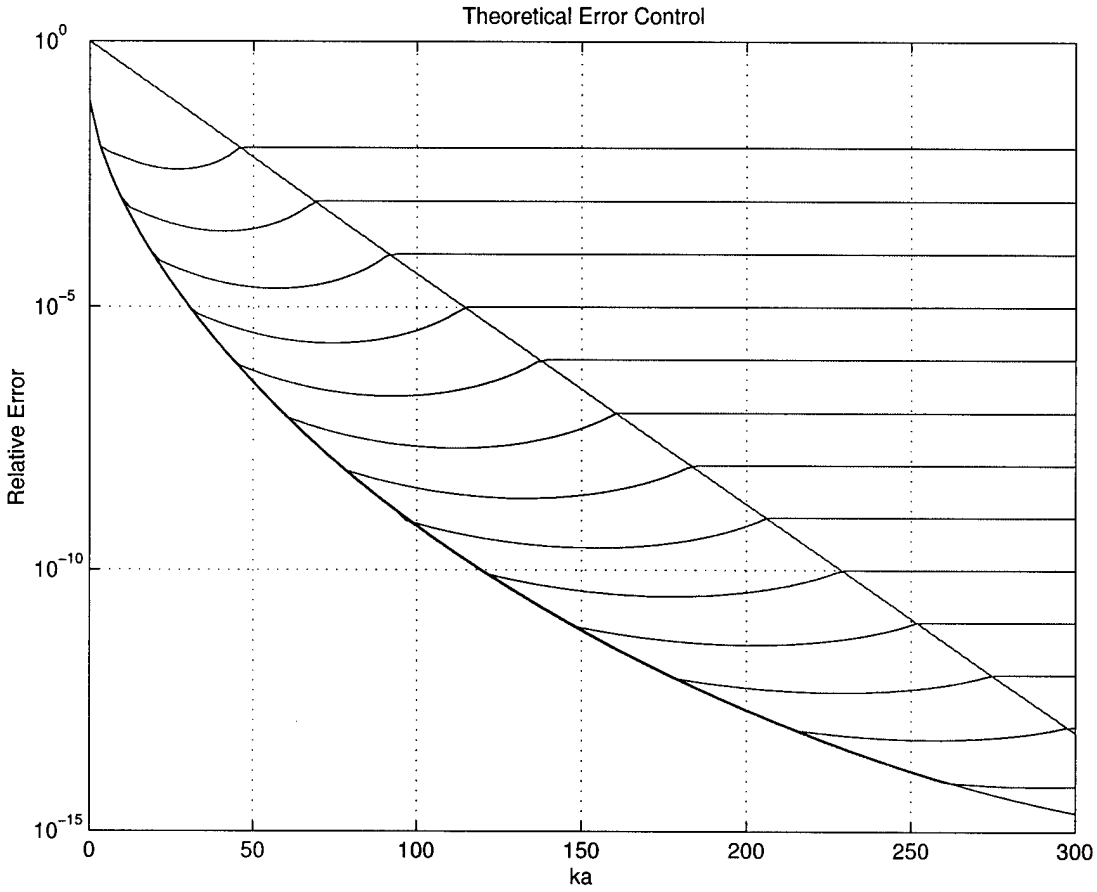


Figure 4.5 Theoretical error levels using carefully chosen truncation values. The downward curvature indicates too many terms used in series. Digits of accuracy range from integer values of 2 to 14.

The truncation numbers corresponding to the theoretical minimum error in Figure 4.3 are shown in Figure 4.7 as the upper solid $L = L_{max}$ line. The boundary line between the upper two controllable regions represents the $L = 2ka$ line. This corresponds to the lower solid line in Figure 4.7. From Figure 4.7, it is clear that the lower error achieved by the new approach requires more terms in the series. This figure also has both the values of L chosen by the new approach and the excess bandwidth formula for $d_0 = 3$ (the example illustrated in Figure 4.6). Naturally, the values of L chosen when $L < 2ka$ are the same for the new approach and the excess bandwidth formula. This is seen at $ka > 67$ where the lines intersect and become one. Interpolation is used when $2ka < L < L_{max}$, and L_{max} is used in the

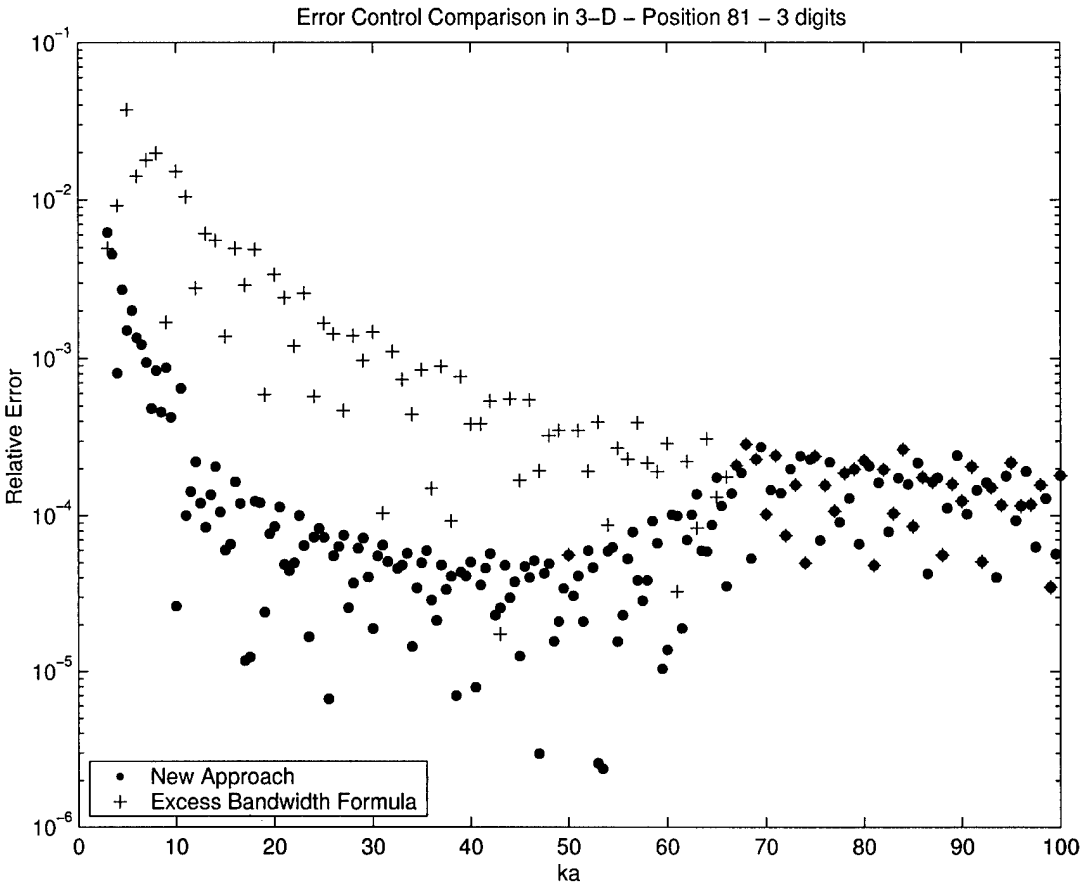


Figure 4.6 Comparison between the new approach and the excess bandwidth formula for a desired three digits of accuracy. The new approach uses the excess bandwidth formula for $ka > 67$ and the minimum error for $ka < 10$. The difference highlights the need for the new approach in the region $(n+1)ka < L < L_{max}$.

uncontrollable region to minimize the error.

Figure 4.8 compares the truncation number when $3 < ka < 10$ for the actual minimum error which is around 10^{-3} and the L selected using numerical results. The deviation is $+1$ to -2 between the predicted L that would achieve the minimum error and the actual L values that are used for the minimum error. In other words, the minimum was often found using slightly fewer terms than predicted. This figure also shows the number L found by the excess bandwidth formula with $d_0 = 3$ to show how many more terms are required to achieve the minimum error as compared to what the excess bandwidth formula predicts.

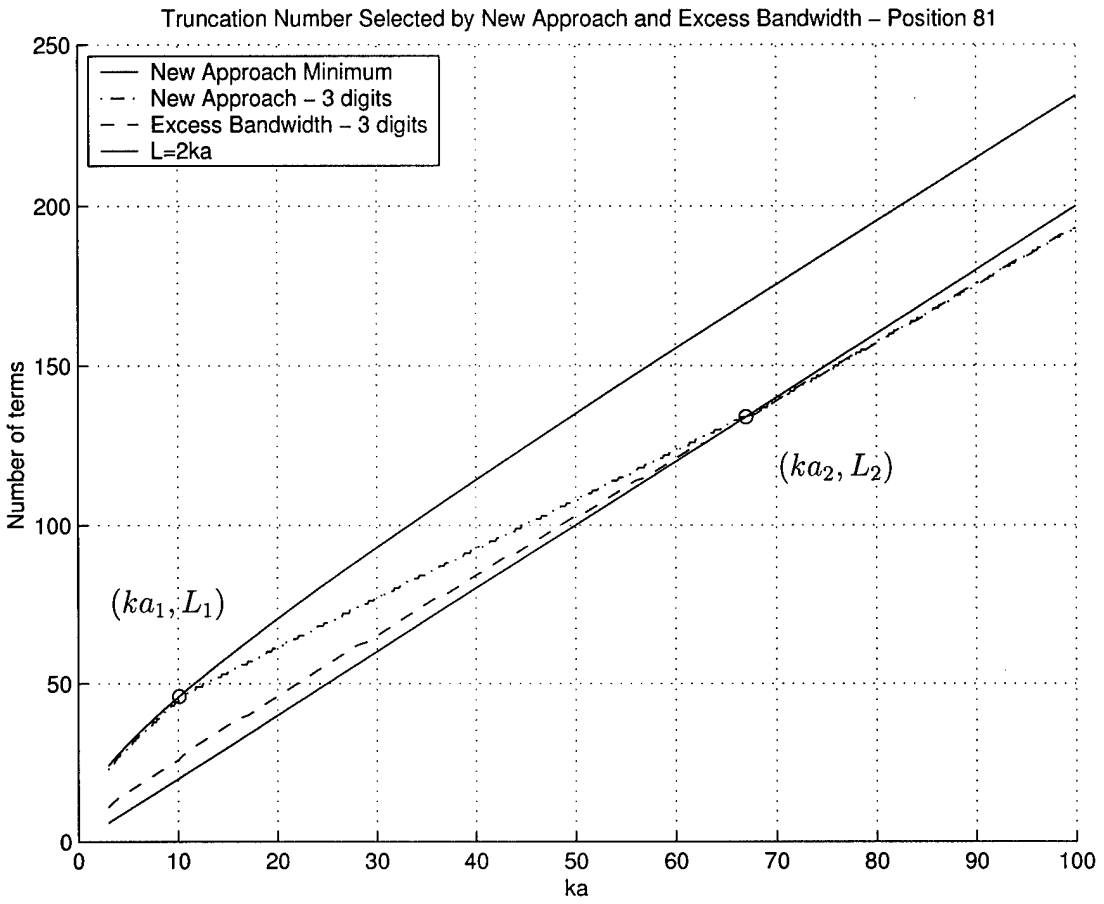


Figure 4.7 Truncation number for varying ka. The theoretical number of terms needed by the new approach to achieve the minimum error for varying ka is compared to the number of terms selected using the excess bandwidth formula with $d_0 = 3$. The line for one buffer box where $L = 2ka$ is shown, and it intersects the line where the new approach is used to control the error ($d_0 = 3$).

Finally, the new approach and excess bandwidth formula are used together to achieve a fixed error for increasing ka. In the extended error controllable region, the interpolated values of L are used to simulate the relative error using the new approach. Figure 4.9 shows the numerical data for fixed error levels as ka is changed. In each case, the required accuracy is achieved. Using higher order interpolation in the region where the new approach is used would lead to a better prediction of L and, hence, to a flatter response. To use higher order interpolation would require root finding of the midpoint truncation L_{mid} and curve fitting between (ka_1, L_1),

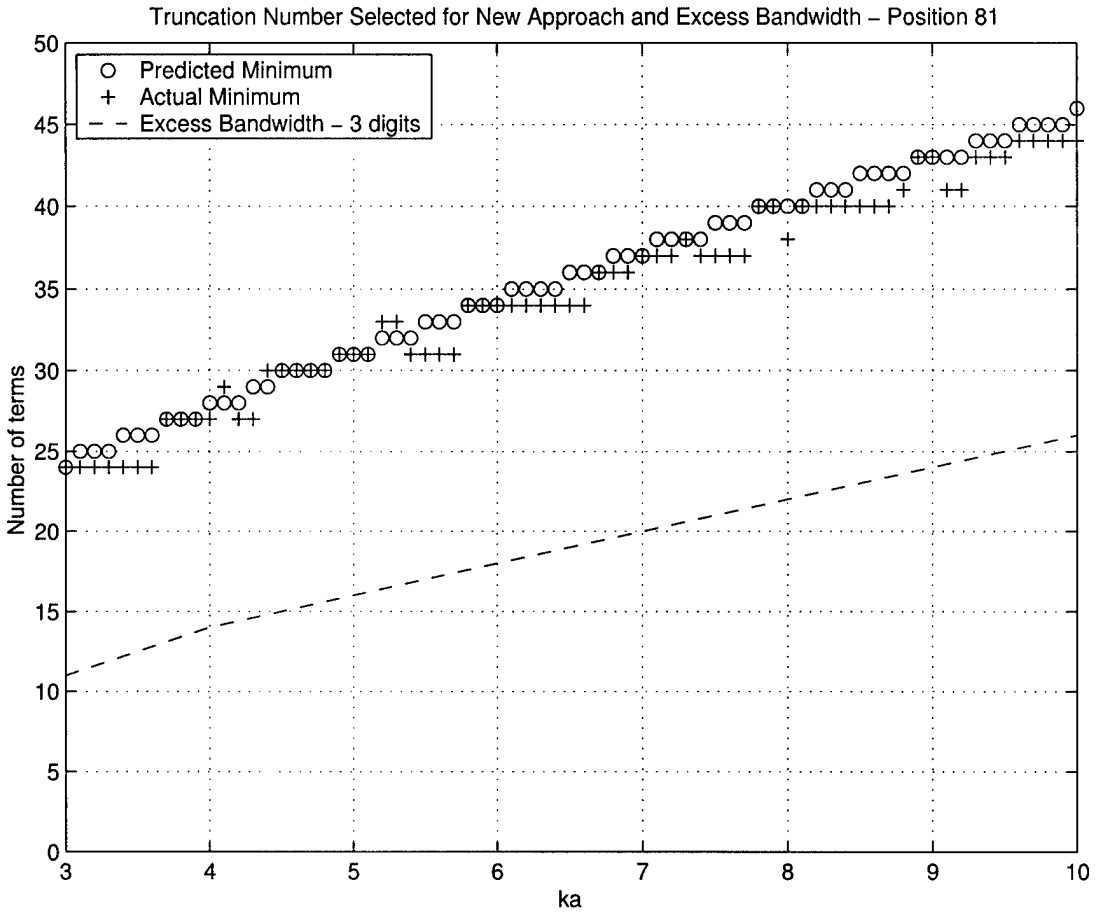


Figure 4.8 Actual versus the predicted number of terms to achieve the minimum error. Only small differences are seen. The number of terms selected using the excess bandwidth formula with $d_0 = 3$ is also shown for comparison.

(ka_{mid}, L_{mid}), and (ka_2, L_2).

The point of these results is to show that the error can be better controlled to achieve the desired accuracy using the new approach as a companion to the excess bandwidth formula. Of course, accuracy in the region where the error is uncontrollable can only be improved by increasing the buffer boxes which is a tradeoff to efficient calculations.

4.5.3 Better error control

When the truncation number is chosen using horizontal interpolation, where the interpolation is done using (ka_1, L_1) and (ka_2, L_2), we see that the theoretical

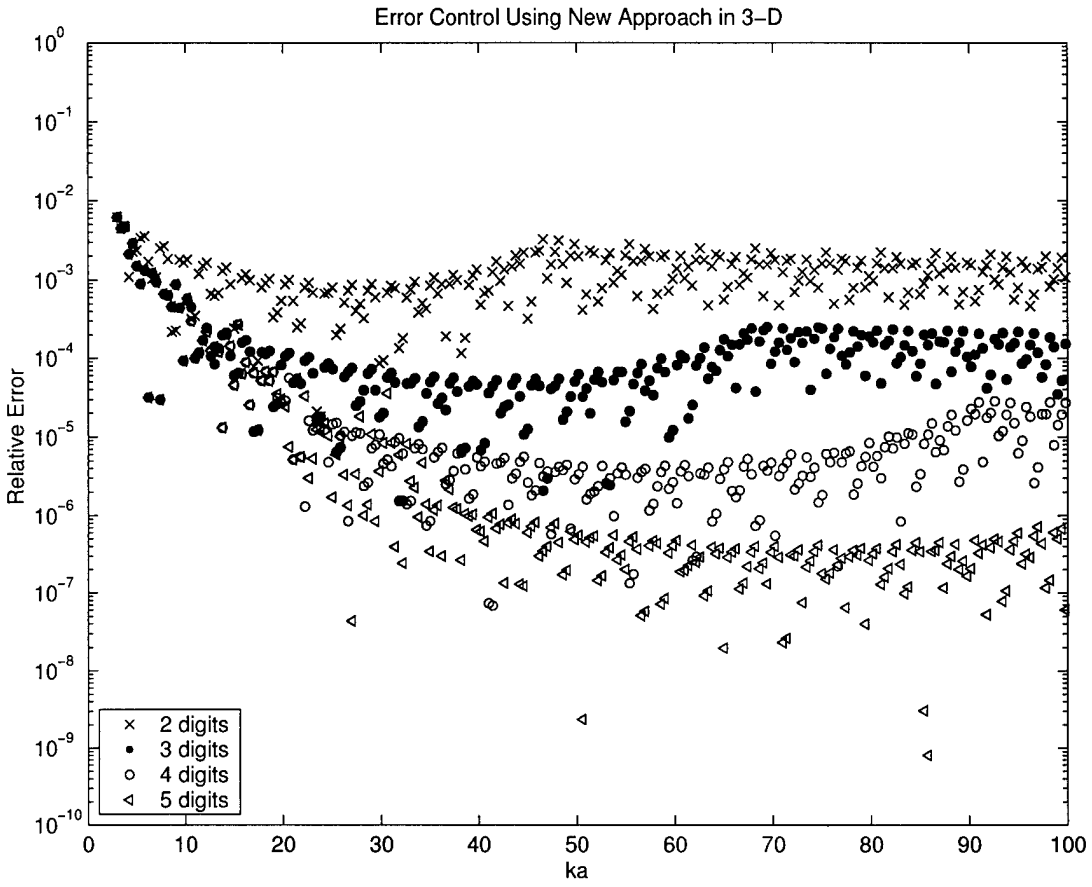


Figure 4.9 Numerical results using the new approach to control error level. Fixed error levels are achieved by employing the new approach. The legend gives the number of digits accuracy desired. The slight dip in the extended error controllable region where the new approach is used can be eliminated through higher order interpolation [2].

error is lower than predicted using linear interpolation. Of course, the error will even be lower when the source and/or field points are closer to their box centers or near each other's relative position. The bending down of the error is due to larger truncation, L. Using a root solver, a precise truncation can be found to place the error at a fixed point. However, since the horizontal span of the minimum error line and the error line for $L = 2ka$ at a fixed error level is larger than the vertical span, vertical interpolation should be better. Figure 4.10 shows the theoretical error for fixed error levels using vertical and the previously discussed horizontal

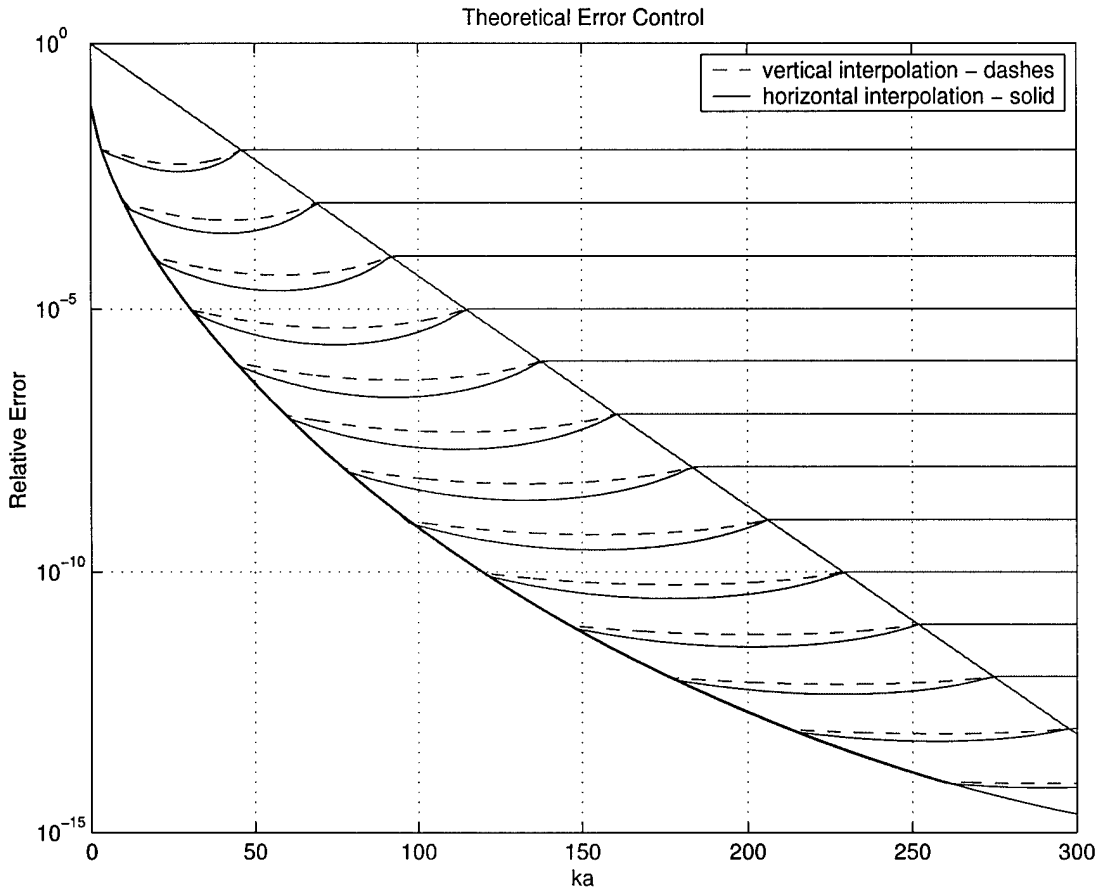


Figure 4.10 Vertical interpolation versus horizontal interpolation. The vertical interpolation produces a flatter response but still has the typical dip.

interpolation. Referring to Figure 4.3 the vertical interpolation simply uses points A and B at a fixed ka but different error levels to interpolate the desired truncation number.

Looking at one last case as purely an academic exercise, we examine the collinear case where the source points and field points are actually outside the box but inside the sphere enclosing the box. We see that the collinear case is actually the worst case. Figure 4.11 shows the collinear case where the vector to the source point and to the field point are along the translation direction. Fortunately, we find that for practical problems, the error will always be lower than this collinear case.

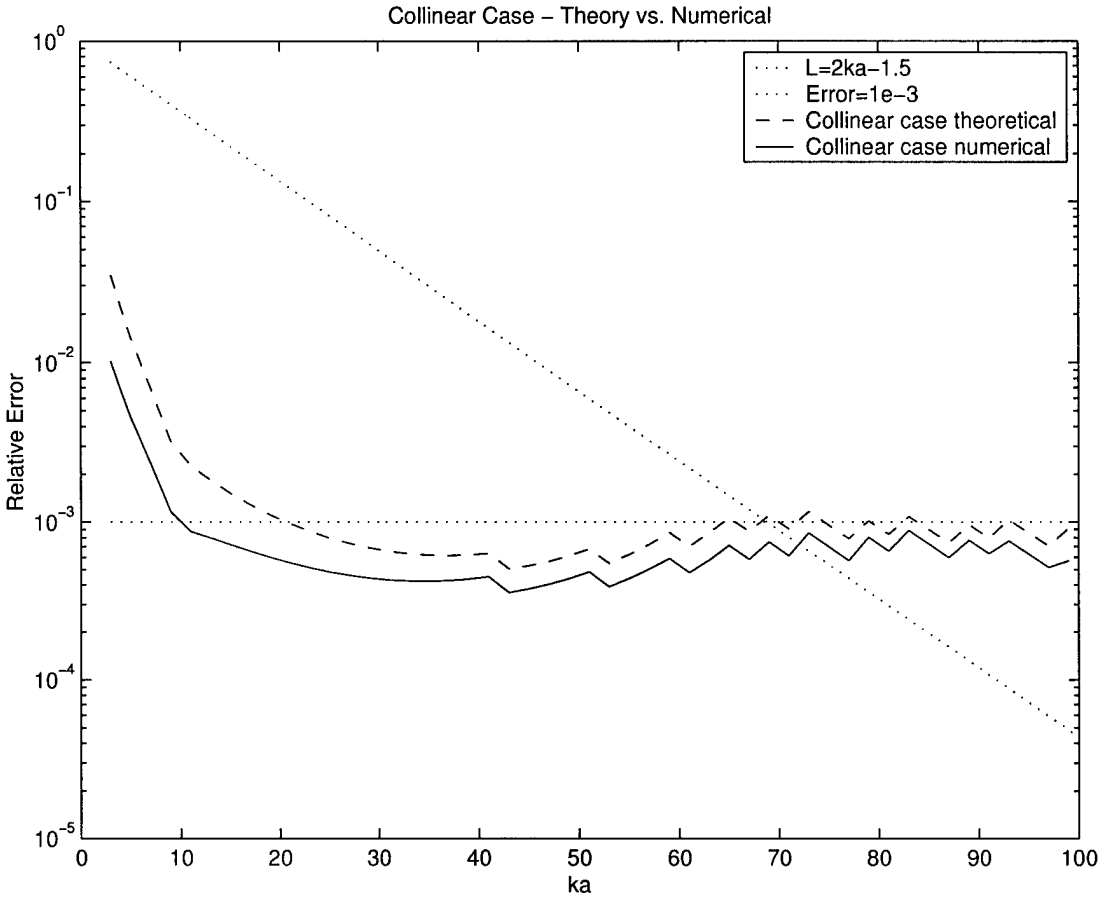


Figure 4.11 Vertical interpolation for the collinear case. The collinear case does not occur because the source and field points would fall outside of the boxes even though they are inside the group spheres.

This chapter has described some of the error sources of MLFMA and deeply explored the errors associated with the translation operator. The new approach can be used to predict and control the error.

Multilevel Fast Multipole Algorithm for Electromagnetic Scattering by Large Complex Objects

Jiming Song, *Member, IEEE*, Cai-Cheng Lu, *Member, IEEE*, and Weng Cho Chew, *Fellow, IEEE*

***Abstract*— The fast multipole method (FMM) and multilevel fast multipole algorithm (MLFMA) are reviewed. The number of modes required, block-diagonal preconditioner, near singularity extraction, and the choice of initial guesses are discussed to apply the MLFMA to calculating electromagnetic scattering by large complex objects. Using these techniques, we can solve the problem of electromagnetic scattering by large complex three-dimensional (3-D) objects such as an aircraft (VFY218) on a small computer.**

***Index Terms*—Electromagnetic scattering, numerical analysis.**

I. Introduction

RECENTLY, many researchers in the electromagnetics community have investigated iterative solvers for integral equations of electromagnetic scattering problems. The integral equation is discretized into a matrix equation by the method of moments (MoM). The resultant matrix equation is then solved by, for example, the conjugate gradient (CG) method, requiring $O(N^2)$ operations for the matrix-vector multiplies in each iteration, where N is the number of unknowns. A number of techniques have been proposed to speed up the evaluation of the matrix-vector multiply. The impedance matrix localization (IML) technique [1] allows the MoM matrix to be replaced by a matrix with localized clumps of large elements. The use of wavelet basis functions [2] reduces the solution time by a constant factor but not the computational complexity. The complex multipole beam approach (CMBA) [3] represents the scattered field in a series of beams produced by multipole sources located in the complex space, but it is efficient only for smooth surfaces. The multilevel matrix decomposition algorithm (MLMDA) [4] permits a fast matrix-vector multiply by decomposing the MoM matrix into a large number of blocks, each describing the interaction between distant scatterers. The multiplication of each block with a vector is executed using a multilevel scheme that resembles a fast Fourier transform (FFT).

The fast multipole method (FMM) [5]–[9] was originally proposed by Rokhlin to evaluate particle simulations and to solve static integral equation rapidly. Barnes and Hut [10] and Hernquist [11] performed n-body simulation using hierarchical method which is simpler than the FMM. But its computational complexity of $O(N \log N)$ is more than that of the FMM, which is $O(N)$ where N is the number of particles. The FMM was extended by Rokhlin to solve acoustic wave scattering problems [12] and then to solve electromagnetic scattering problems by many researchers in both two dimensions [13]–[17] and three dimensions [18]–[20]. A two-level FMM reduces both the complexity of a matrix-vector multiply and memory requirement from $O(N^2)$ to $O(N^{1.5})$ where N is the number of unknowns. A three-level FMM reduces it to $O(N^{4/3})$ [12], [21]. With a nonnested method, using the ray-propagation fast multipole algorithm (RPFMA) [16], [17], a two-level FMM reduces the complexity to $O(N^{4/3})$ also. The multilevel fast multipole algorithm (MLFMA) [22]–[25] further reduces the complexity and memory requirement. Dembart and Yip [23], [24] have implemented the MLFMA using signature function, interpolation, and filtering, with a complexity of $O(N \log^2 N)$. Song and Chew [25], implemented the MLFMA with $O(N \log N)$ complexity and memory requirement using translation, interpolation, anterpolation (adjoint interpolation), and a grid-tree data structure.

The numerical results for the radar cross section (RCS) of some simple objects like the sphere, cube, and the NASA almond are reported in [19], [20], and [25]. Since they are closed smooth objects that are not very thin, the combined field integral equation (CFIE) with uniform grids has a small condition number and converges very fast. In this paper, we will apply the MLFMA to large complex three-dimensional (3-D) objects such as an aircraft (VFY218). The number of modes required, preconditioner, near singularity extraction, and the choice of the initial guess will be discussed.

II. Multilevel Fast Multipole Algorithm (MLFMA)

To implement a multilevel fast multipole algorithm (MLFMA), we enclose the entire object in a large cube first, which is then partitioned into eight smaller cubes. Each subcube is then recursively subdivided into smaller cubes until the edge length of the finest cube is about half a wavelength. Cubes at all levels are indexed. At the finest level we find the cube in which each basis function resides by comparing the coordinates of the center of the basis function with the center of cube. We further find nonempty cubes by sorting. Only nonempty cubes are recorded using tree-structured data

Manuscript received March 19, 1996; revised June 4, 1997. This work was supported in part by the Office of Naval Research under Grant N00014-95-1-0872, by the National Science Foundation under Grant NSF ECS 93-02145, and by AFOSR under an MURI Grant.

The authors are with the Center for Computational Electromagnetics, Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801 USA.

Publisher Item Identifier S 0018-926X(97)07215-3.

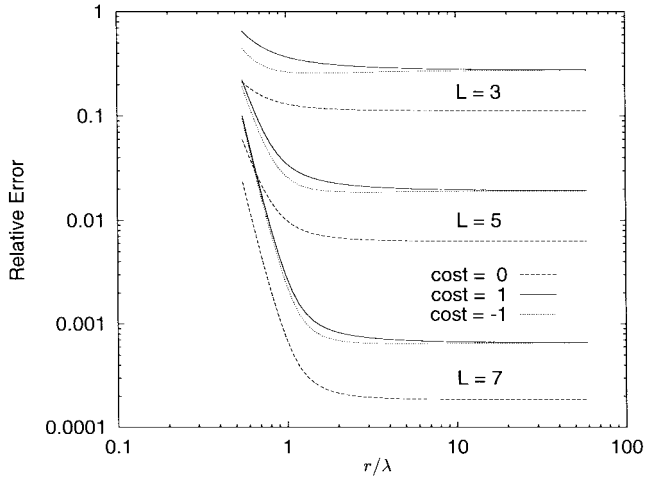


Fig. 1. Relative error in the dynamic scalar potential truncated for the first $L+1$ terms (3) as functions of the distance r/λ for $d/\lambda = 0.4$ where cost $= \hat{d} \cdot \hat{r}$.

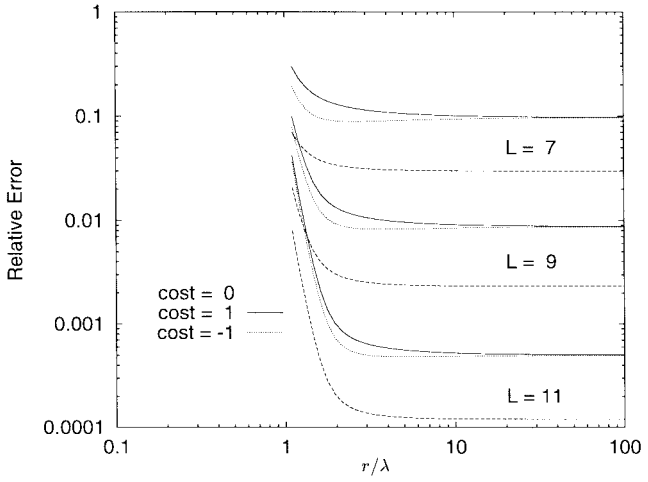


Fig. 2. Relative error in the dynamic scalar potential truncated for the first $L+1$ terms (3) as functions of the distance r/λ for $d/\lambda = 0.8$ where cost $= \hat{d} \cdot \hat{r}$.

at all levels [10], [11]. Thus, the computational cost depends only on the nonempty cubes.

A. Number of Modes

The addition theorem for 3-D dynamic scalar Green's function has the form [18], [26]

$$\frac{e^{ik|\mathbf{r}+\mathbf{d}|}}{|\mathbf{r}+\mathbf{d}|} = ik \sum_{l=0}^{\infty} (-1)^l (2l+1) j_l(kd) h_l^{(1)}(kr) P_l(\hat{d} \cdot \hat{r}) \quad (1)$$

where k is the wavenumber, j_l is a spherical Bessel function of the first kind, $h_l^{(1)}$ is a spherical Hankel function of the first kind, P_l is a Legendre polynomial, $\mathbf{r}$ and $\mathbf{d}$ are two vectors, r and d are their amplitudes with $d < r$, and $\hat{r}$ and $\hat{d}$ are their unit vectors, respectively. In this paper, $e^{-i\omega t}$ time convention is used. Using small argument approximations of j_l and $h_l^{(1)}$, we obtain the addition theorem for the 3-D static Green's function

$$\frac{1}{|\mathbf{r}+\mathbf{d}|} = \frac{1}{r} \sum_{l=0}^{\infty} (-1)^l \left(\frac{d}{r}\right)^l P_l(\hat{d} \cdot \hat{r}). \quad (2)$$

In numerical simulations, the infinite series in (1) and (2) are truncated as

$$\frac{e^{ik|\mathbf{r}+\mathbf{d}|}}{|\mathbf{r}+\mathbf{d}|} \approx ik \sum_{l=0}^{L} (-1)^l (2l+1) j_l(kd) h_l^{(1)}(kr) P_l(\hat{d} \cdot \hat{r}) \quad (3)$$

$$\frac{1}{|\mathbf{r}+\mathbf{d}|} \approx \frac{1}{r} \sum_{l=0}^{L} (-1)^l \left(\frac{d}{r}\right)^l P_l(\hat{d} \cdot \hat{r}). \quad (4)$$

For the static case, the number of modes (L) needed in (4) depends on the ratio of d to r for a given desired accuracy. This means that we can use the same number of modes for different cube sizes. Due to oscillatory nature of dynamic fields, the dynamic case is more complicated than the static case. In Figs. 1 and 2 we plot the relative error in (3) as functions of r/λ for different L and $\hat{d} \cdot \hat{r}$. Fig. 1 is for $d = 0.4\lambda$ and Fig. 2 is for $d = 0.8\lambda$. From these two figures, the accuracy does not increase even when r increases beyond $2d$. When d increases, the number of modes L required to maintain the same accuracy increases.

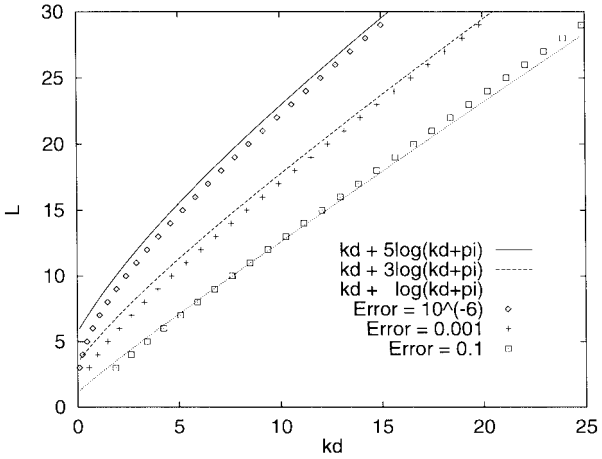


Fig. 3. Number of modes needed in (3) as functions of kd for different accuracies ($\hat{d} \cdot \hat{r} = 1$, $r/\lambda = \infty$). Some semi-empirical formulas are plotted for comparison.

In Fig. 3, we plot the number of modes L needed in (3) as functions of kd for different accuracies. Some semi-empirical formulas are plotted on the same figure for comparison. To obtain less than 0.1 relative error

$$L = kd + \ln(\pi + kd) \quad (5)$$

should be used in (3), and

$$L = kd + 5 \ln(\pi + kd) \quad (6)$$

should be used for less than 10^{-6} relative error. Equation (6) is the same as the one given in [18] for single precision. The FMM is applied to off-diagonal matrix elements only, which are two to three orders less than diagonal matrix elements for electromagnetic scattering problems. Hence, from our numerical experience, L calculated from (5) suffices for decent current solutions and RCS.

The MLFMA is used to speed up the matrix-vector multiply in the iterative methods. It decomposes the matrix-vector

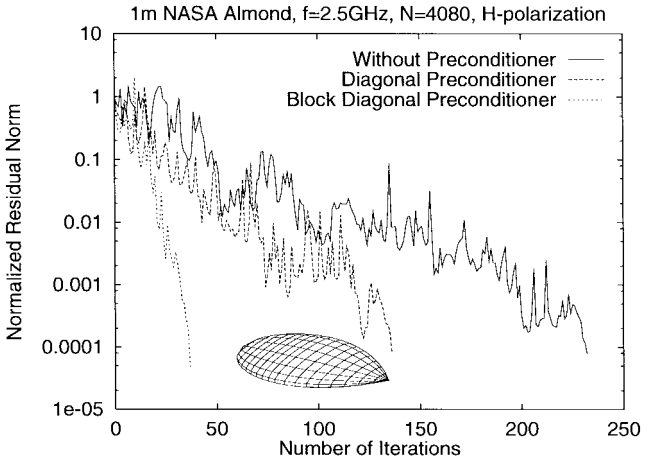


Fig. 4. Comparison of the convergence of solutions of CFIE using the biconjugate gradient (BiCG) method for a 1-m NASA almond at 2.5 GHz with or without different preconditionings.

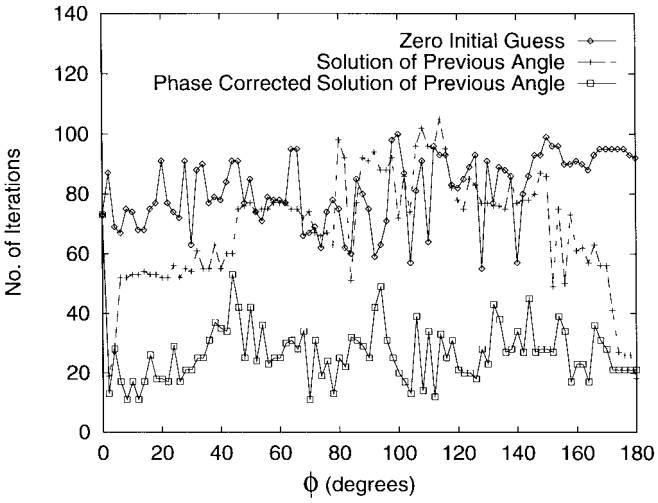


Fig. 5. Number of iterations as functions of incident angles for different initial guesses: using zero initial guess for all angles and using the solution of the previous angle as the initial guess for the next angle with/without phase corrections.

multiply into two sweeps [27]: the first sweep consists of constructing multipole expansions for each nonempty cube at all levels. Since the multipole expansions are used for calculating the fields outside the cube, they are called outer multipole expansions. As one progresses from the finest level to the coarsest level, the cube becomes larger and the number of modes required in the multipole expansions increases. To construct outer multipole expansions for each nonempty cube at all levels, the outer multipole expansions are computed at the finest level and then the expansions for larger cubes are obtained using interpolation and shifting. The second sweep consists of constructing local multipole expansions contributed from well-separated cubes at all levels. At the coarsest level, the local multipole expansions contributed from well-separated cubes are calculated using translation. At the other levels, the local expansions for smaller cubes include the contributions from parent cubes using shifting and anterpolation (adjoint interpolation) [28] and from well-separated cubes at this level but not well-separated ones at the parent level. The anterpolation matrix is the transpose of the interpolation matrix.

B. Block-Diagonal Preconditioner

The CPU time for iterative methods is proportional to the number of iterations needed to get the desired accuracy. The convergence rate depends on spectral properties of the MoM matrix. Hence, one may want to transform the original matrix equation $\overline{\mathbf{A}} \cdot \mathbf{x} = \mathbf{b}$ into $\overline{\mathbf{M}}^{-1} \cdot \overline{\mathbf{A}} \cdot \mathbf{x} = \overline{\mathbf{M}}^{-1} \cdot \mathbf{b}$ that has the same solution, but with a more favorable spectral property where $\overline{\mathbf{M}}^{-1}$ is called a preconditioner.

If basis functions in one of the finest cubes are considered as one group, the matrix $\overline{\mathbf{A}}$ has block structure and can be further divided as

$$\overline{\mathbf{A}} \cdot \mathbf{x} = (\overline{\mathbf{A}}_0 + \overline{\mathbf{A}}_1) \cdot \mathbf{x} + \overline{\mathbf{A}}_2 \cdot \mathbf{x} \tag{7}$$

where matrices $\overline{\mathbf{A}}_0$ and $\overline{\mathbf{A}}_1$ account for nearby interactions and can be derived directly from the MoM matrix and $\overline{\mathbf{A}}_0$ is the block-diagonal part. The matrix $\overline{\mathbf{A}}_2$ accounts for far interactions and $\overline{\mathbf{A}}_2 \cdot \mathbf{x}$ is performed by the MLFMA. Choosing $\overline{\mathbf{A}}_0^{-1}$ as a preconditioner, we have

$$\overline{\mathbf{A}}_0^{-1} \cdot \overline{\mathbf{A}} \cdot \mathbf{x} = \mathbf{x} + \overline{\mathbf{A}}_0^{-1} \cdot (\overline{\mathbf{A}}_1 \cdot \mathbf{x} + \overline{\mathbf{A}}_2 \cdot \mathbf{x}). \tag{8}$$

Since $\overline{\mathbf{A}}_0$ can be replaced by its LU decomposition (LUD) form for $\overline{\mathbf{A}}_0^{-1}$, the block-diagonal preconditioner needs no extra memory and no extra CPU time in each matrix-vector multiply. $\overline{\mathbf{A}}_0$ is a block-diagonal matrix with a block size of M, which is the number of unknowns in one cube. When M is a constant, the LUD of $\overline{\mathbf{A}}_0$ takes $O(M^3N/M) = O(N)$ operations. In Fig. 4, we plot the normalized residual norm as functions of iteration numbers for cases without preconditioning, diagonal preconditioning, and block-diagonal preconditioning. We find the current solution for a 1-m NASA almond at 2.5 GHz for the wave incidence on the tip. The incident electric field is parallel to its broad side. It is observed that block-diagonal preconditioning converges much faster than the other two.

C. Near-Singularity Extraction and Choice of Initial Guess

For very thin objects (like a wing), CFIE (combined field integral equation) [29] has a smaller condition number than those of an electric field integral equation (EFIE) and a magnetic field integral equation (MFIE). The null-space solutions of the EFIE will not radiate and null-space solutions of the MFIE will radiate. Therefore, both the EFIE and the MFIE cannot give correct current solutions, while the EFIE gives a correct RCS but the MFIE does not. However, the CFIE always gives a correct current solution as well as a correct RCS.

For finite-thickness objects only the self terms have a singularity and only self-singularity extraction [30] is needed. For very thin objects, both self- and near-singularity extractions [31] are required to obtain correct matrix elements.

For iterative solutions of monostatic RCS, different incident angles require different iterative solutions. Since a small change in the incident angle corresponds to a small change in the current, we use the current solution from the previous angle

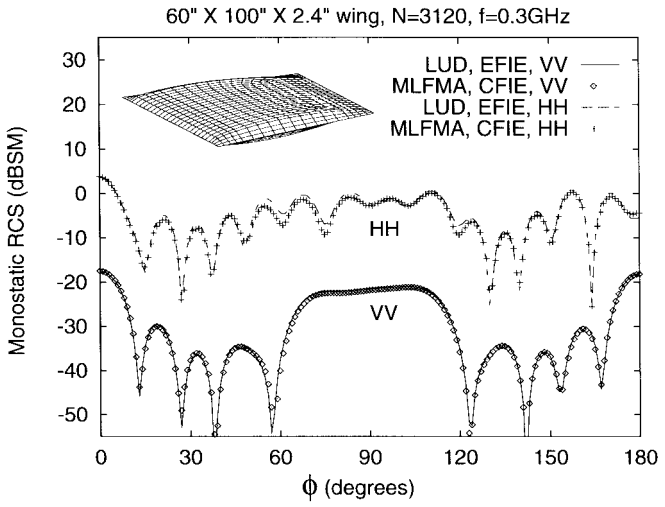


Fig. 6. Monostatic RCS of the wing (2080 flat triangular patches divided from Northrop curvilinear quad patch model) at 300 MHz as functions of ϕ in the horizontal plane.

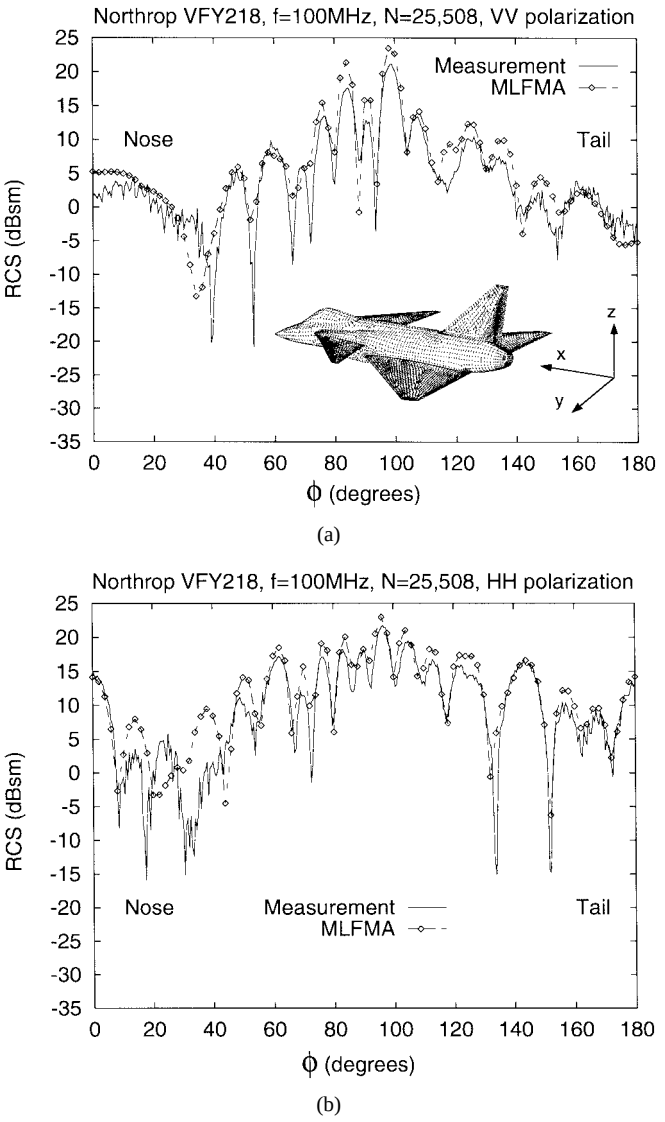


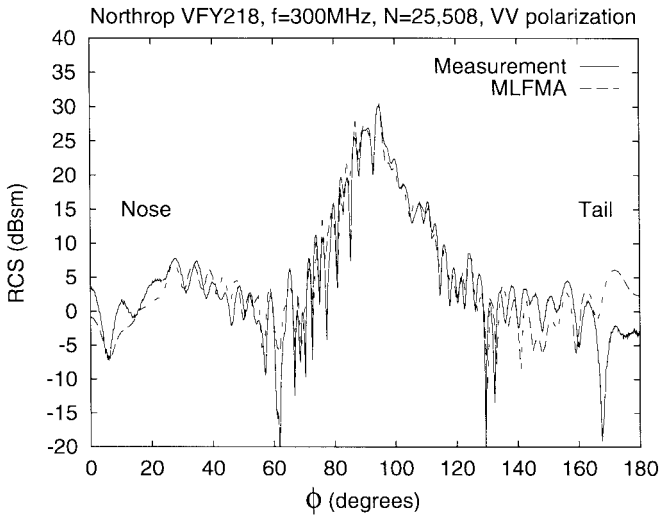
Fig. 7. Monostatic RCS of the aircraft VFY218 (Northrop curvilinear quad patch model) at 100 MHz as functions of ϕ in the horizontal plane. The measurement data are from the Naval Air Warfare Center. (a) VV polarization. (b) HH polarization.

with phase correction as the initial guess for the next angle. This technique reduces the number of iterations significantly. As an illustration, we calculate the monostatic RCS from the VFY218 at 100 MHz for vertical (VV) polarization. The VFY218 is shown in the inset of Fig. 7(a). The wings of the VFY218 are on the x-y plane (horizontal plane). Zero degree ($\phi = 0$) corresponds to the incidence angle on the nose. The VFY218 is 609 in (15.5 m) from nose to tail, 350.4 in (8.9 m) from one wing to another, and 161.4 in (4.1 m) from top to bottom. In Fig. 5, we plot the number of iterations for different incident angles using three kinds of initial guesses. The first case, which uses zero as the initial guess for all angles, needs about 85 iterations on the average for each angle. The second case, which uses the solution of the previous angle (2° step size) as the initial guess for the next angle, needs about 65 iterations per angle. The third case, which uses the phase-corrected solution of the previous angle as the initial guess for the next angle, needs only about 30 iterations per angle.

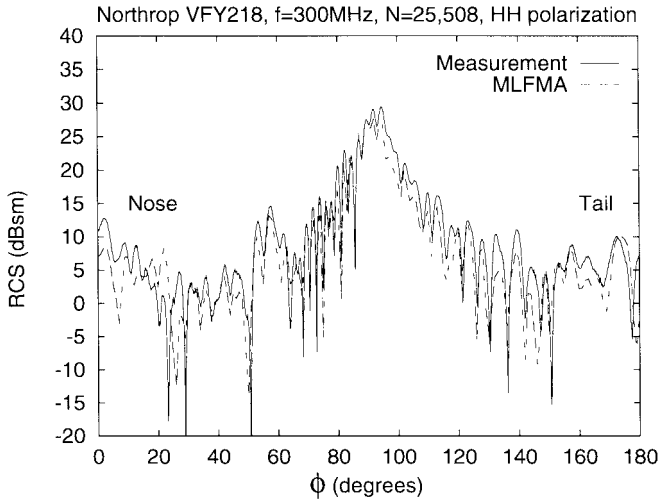
III. Numerical Results

The MLFMA has been implemented based on flat triangular patches and curvilinear quad patches using both Galerkin's method and line matching where the testing functions are constant along the line joining the centers of two adjacent patches. For curvilinear quad patches, generalized rooftop functions are used as basis functions [30]–[32]. The Rao, Wilton, and Glisson (RWG) [33] basis functions are used for flat triangular patches. The number of modes L calculated from (5) is used for numerical simulations. The code is verified by comparing the results with those in the published literature for conducting objects with different shapes like sphere, plate, cube, NASA almond, etc. Our numerical results agree very well with the analytical solutions, the measurements, and the LUD solutions. Both the memory requirements and the CPU time per iteration are of $O(N \log N)$ and a 110 592 unknown problem can be solved within 24 h on a SUN Sparc10 [25] (6 h for setup, 17 h for 29 iterations to real 0.001 normalized residual error, and 1 h for calculating 901 points of bistatic RCS).

Fig. 6 shows the monostatic RCS of a wing at 300 MHz using the LUD for the EFIE and the MLFMA for the CFIE. The wing size is 60 in $\times$ 100 in $\times$ 2.4 in and is originally modeled by Northrop using curvilinear quad patches. Dividing each quad patch as two flat triangular patches leads to a 3120 unknown problem. The wing is on the x-y plane, and zero degree ($\phi = 0$) corresponds to normal incidence to the 60 in short edge. The thickness in the z direction is only about 2% to 4% of the lengths in the x and y directions. If the near-singularity extraction is not used, we cannot obtain a correct RCS from the CFIE. Using the near-singularity extraction, we obtain a good RCS agreement between the EFIE and the CFIE. This 3120 unknown problem can also be solved using the LUD on a workstation. It is found that the RCS calculated using the MLFMA agrees very well with that using the LUD for both the EFIE and the CFIE. In Fig. 6, we plot the RCS calculated



(a)



(b)

Fig. 8. Monostatic RCS of the aircraft VFY218 (Northrop curvilinear quad patch model) at 300 MHz as functions of ϕ in the horizontal plane. The measurement data are from the Naval Air Warfare Center. (a) VV polarization. (b) HH polarization.

using the LUD for the EFIE and the MLFMA for the CFIE only. Good agreement is observed between the results.

Fig. 7(a) and (b) shows the monostatic RCS for the aircraft (VFY218) at 100 MHz as functions of ϕ in the horizontal plane using the Northrop curvilinear quad patch model for horizontal (HH) and VV polarizations, respectively. Zero degree ($\phi = 0$) corresponds to an incidence angle on the nose. A five-level MLFMA is used. The measurement data are from the Naval Air Warfare Center, China Lake, CA. Good agreement between the numerical results and the measurements for both HH and VV polarizations is observed. For this 25 508 unknown problem, the MLFMA needs 167 MB of memory for this single-precision code and requires 45 h of the CPU time on one processor of an SGI Challenge machine (32 bits for each word, 25 Mflops based on the LINPACK benchmark) for 182 incident angles. In contrast, the LUD solution is estimated to need 5.2 GB of memory and 400 h of the CPU time for LUD and $O(N^2)$ calculations for each incident angle. We estimate that only for 1600 incident angles, the MLFMA would need the same CPU time as the LUD solution. But it needs memory (167 MB) much less than the LUD solution (5.2 GB). The comparison is more in favor of MLFMA when N becomes larger.

The longest edge in the Northrop VFY218 curvilinear quad patch model is 0.106λ at 100 MHz. We use the same model to predict the RCS at 300 MHz using the MLFMA. The monostatic RCS for HH and VV polarizations is shown in Fig. 8(a) and (b), respectively. The numerical results are in good agreement with the measurements.

IV. Conclusions

The MLFMA has been implemented for both flat triangular patch and curvilinear quad patch geometry descriptions to speed up the matrix-vector multiplies. Both the memory requirements and the CPU time per iteration are of $O(N \log N)$. Using a block-diagonal preconditioner, near-singularity extraction, and phase corrected previous solution for the initial guess, we can solve for the electromagnetic scattering by large complex 3-D objects such as an aircraft (VFY218) on a small computer.

Acknowledgment

The authors would like to thank M. I. Sancer and G. Antilla of Northrop for providing the geometry models of the wing and VFY218. The computer time was provided by the National Center for Supercomputing Applications (NCSA) at the University of Illinois, Urbana-Champaign. Since the submission of the manuscript for publication on March 14, 1996, the authors have solved the VFY218 at 2 GHz and problems involving 1.9 million unknowns using resources at CCEM and NCSA.

References

[1] F. X. Canning, "Solution of IML form of moment method problems in 5 iterations," *Radio Sci.*, vol. 30, no. 5, pp. 1371–1384, Sept./Oct. 1995.
[2] R. L. Wagner and W. C. Chew, "A study of wavelets for the solution of electromagnetic integral equations," *IEEE Trans. Antennas Propagat.*, vol. 43, pp. 802–810, Aug. 1995.
[3] A. Boag and R. Mittra, "Complex multipole beam approach to electromagnetic scattering problems," *IEEE Trans. Antennas Propagat.*, vol. 42, pp. 366–372, Mar. 1994.
[4] E. Michielssen and A. Boag, "Multilevel evaluation of electromagnetic fields for the rapid solution of scattering problems," *Microwave Opt. Technol. Lett.*, vol. 7, no. 17, pp. 790–795, Dec. 5, 1994.
[5] V. Rokhlin, "Rapid solution of integral equations of classical potential theory," *J. Comput. Phys.*, vol. 60, no. 2, pp. 187–207, Sept. 15, 1985.
[6] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comput. Phys.*, vol. 73, pp. 325–348, Dec. 1987.
[7] ——, "Fast methods in the three dimensions," in *Vortex Methods*, C. Anderson and C. Greengard, Eds. Berlin, Germany: Springer-Verlag, 1988, lecture notes in Mathem., 1360.
[8] J. Carrier, L. Greengard, and V. Rokhlin, "A fast adaptive multipole algorithm for particle simulations," *SIAM J. Sci. Stat. Comput.*, vol. 9, pp. 669–686, July 1988.
[9] J. Ambrosiano, L. Greengard, and V. Rokhlin, "The fast multipole method for gridless particle simulation," *Comput. Phys. Commun.*, vol. 48, pp. 117–125, 1988.
[10] J. Barnes and P. Hut, "A hierarchical $O(N \log N)$ force calculation algorithm," *Nature*, vol. 324, pp. 446–449, Dec. 1986.
[11] L. Hernquist, "Hierarchical N-body methods," *Comput. Phys. Commun.*, vol. 48, pp. 107–115, 1988.

[12] V. Rokhlin, "Rapid solution of integral equations of scattering theory in two dimensions," *J. Comput. Phys.*, vol. 36, no. 2, pp. 414–439, Feb. 1990.
[13] N. Engheta, W. D. Murphy, V. Rokhlin, and M. S. Vassiliou, "The fast multipole method (FMM) for electromagnetic scattering problems," *IEEE Trans. Antennas Propagat.*, vol. 40, pp. 634–641, June 1992.
[14] L. R. Hamilton, M. A. Stalzer, R. S. Turley, J. L. Visher, and S. M. Wandzura, "Scattering computation using the fast multipole method," in *IEEE APS Int. Symp. Dig.*, Ann Arbor, MI, June 1993, pp. 852–855.
[15] C. C. Lu and W. C. Chew, "Fast algorithm for solving hybrid integral equations," *Proc. Inst. Elect. Eng.*, vol. 140, pt. H, no. 6, pp. 455–460, Dec. 1993.
[16] R. Coifman, V. Rokhlin, and S. Wandzura, "Faster single-stage multipole method for the wave equation," in *10th Annu. Rev. Progress Appl. Computat. Electromagn.*, Monterey, CA, Mar. 1994, pp. 19–24.
[17] R. L. Wagner and W. C. Chew, "A ray-propagation fast multipole algorithm," *Microwave Opt. Technol. Lett.*, vol. 7, no. 10, pp. 435–438, July 1994.
[18] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propagat. Mag.*, vol. 35, pp. 7–12, June 1993.
[19] J. M. Song and W. C. Chew, "Fast multipole method solution using parametric geometry," *Microwave Opt. Technol. Lett.*, vol. 7, no. 16, pp. 760–765, Nov. 1994.
[20] ———, "Fast multipole method solution of the combined field integral equation," in *11th Annu. Rev. Progress Appl. Computat. Electromagn.*, Monterey, CA, Mar. 1995, pp. 629–636.
[21] V. Rokhlin, "Diagonal forms of translation operators for the Helmholtz equation," *Appl. Computat. Harmon. Analysis*, vol. 1, pp. 82–93, 1993.
[22] C. C. Lu and W. C. Chew, "A multilevel algorithm for solving boundary integral equations of wave scattering," *Microwave Opt. Technol. Lett.*, vol. 7, no. 10, pp. 466–470, July 1994.
[23] B. Dembart and E. Yip, "A 3-D fast multipole method for electromagnetics with multiple levels," in *11th Annu. Rev. Progress Appl. Computat. Electromagn.*, Monterey, CA, Mar. 1995, pp. 621–628.
[24] M. A. Epton and B. Dembart, "Multipole translation theory for the three-dimensional Laplace and Helmholtz equations," *SIAM J. Sci. Comput.*, vol. 16, no. 4, pp. 865–897, July 1995.
[25] J. M. Song and W. C. Chew, "Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering," *Microwave Opt. Technol. Lett.*, vol. 10, no. 1, pp. 14–19, Sept. 1995.
[26] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover, 1965.
[27] C. R. Anderson, "An implementation of the fast multipole method without multipoles," *SIAM J. Sci. Stat. Comput.*, vol. 13, no. 4, pp. 923–947, Apr. 1992.
[28] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernels," *Comput. Phys. Commun.*, vol. 65, pp. 24–38, 1991.
[29] J. R. Mautz and R. F. Harrington, "H-field, E-field, and combined-field solutions for conducting bodies of revolution," *Archiv für Elektronik Übertragung* vol. 32, pp. 157–163, 1978.
[30] J. M. Song and W. C. Chew, "Moment method solutions using parametric geometry," *J. Electromagn. Waves and Appl.*, vol. 9, nos. 1/2, pp. 71–83, 1995.
[31] G. Antilla, Y. C. Ma, P. V. Alstine, and M. I. Sancer, "Hybrid finite element-method of moments for electromagnetic prediction of complex 3-D geometries," in *Course Note, IEEE AP-S Int. Symp. URSI Radio Sci. Meet.*, Newport Beach, CA, June 1995.
[32] G. E. Antilla and N. G. Alexopoulos, "Scattering from complex three-dimensional geometries using a curvilinear hybrid finite-element-integral equation approach," *J. Opt. Soc. Amer. A*, vol. 11, no. 4, pp. 1445–1457, Apr. 1994.
[33] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propagat.*, vol. AP-30, pp. 409–418, May 1982.

Jiming Song (S'92–M'95), for photograph and biography, see p. 245 of the February 1997 issue of this TRANSACTIONS.

Cai-Cheng Lu (S'93–M'95), for photograph and biography, see p. 543 of the March 1997 issue of this TRANSACTIONS.

Weng Cho Chew (S'79–M'80–SM'86–F'93), for photograph and biography, see p. 245 of the February 1997 issue of this TRANSACTIONS.

Optimal Interpolation of Translation Operator in Multilevel Fast Multipole Algorithm

Özgür Ergül, *Student Member, IEEE,* and Levent Gürel, *Senior Member, IEEE*

***Abstract*—Lagrange interpolation of the translation operator in the three-dimensional multilevel fast multipole algorithm (MLFMA) is revisited. Parameters of the interpolation, namely, the number of interpolation points and the oversampling factor, are optimized for controllable error. Via optimization, it becomes possible to obtain the desired level of accuracy with the minimum processing time.**

***Index Terms*—Lagrange interpolation, multilevel fast multipole algorithm, translation operator.**

I. Introduction

THE multilevel fast multipole algorithm (MLFMA) [1], [2] requires translations to convert the radiated fields of the basis clusters into incoming waves for the testing clusters. In a matrix–vector multiplication, translations are performed between the clusters that are at the same level but far from each other. Through the factorization of the Green's function, translation operators are independent from the radiation and receiving patterns of the basis and testing clusters, respectively [3]. To be employed repeatedly, these operators are calculated and stored in the memory before the iterations.

Since direct calculation of the translation operators requires $O(N^{3/2})$ operations, where N is the number of unknowns, processing time for their setup increases rapidly and becomes substantial as problem size grows. As a remedy, a two-step computation is suggested based on the interpolation of the translation operator [4]: First, the translation operator is expressed as a band-limited function of a variable φ and it is sampled at $O(N)$ points with respect to this variable. Second, the operator is evaluated at the required points by interpolation from the previous samples. With an efficient interpolation algorithm, processing time for the calculation of the translation operators is reduced to $O(N)$.

In [4], Lagrange interpolation was proposed to efficiently fill the translation matrices for large problems. However, the parameters of the interpolation, namely, the number of interpolation points and the oversampling factor, were fixed. With the parameters fixed, the interpolation error is not controllable and the processing time is not minimized. In this paper, we revisit the Lagrange interpolation of the translation operators and optimize the parameters of the interpolation to obtain the desired level of accuracy with minimum processing time. The optimal parameters are compared to the fixed parameters to demonstrate the improvement obtained with the optimization.

Manuscript received May 30, 2006; revised August 17, 2006. This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grant 105E172, by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), and by contracts from ASELSAN and SSM.

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, TR-06800, Bilkent, Ankara, Turkey (e-mail: ergul@ee.bilkent.edu.tr; lgurel@bilkent.edu.tr).

Digital Object Identifier 10.1109/TAP.2006.886562

II. Lagrange Interpolation of the Translation Operators

A three-dimensional (3-D) translation operator between a pair of basis and testing clusters at the same level can be written as

$$T(k,\hat{\boldsymbol{k}},\boldsymbol{D})=\frac{ik}{4\pi}\sum_{l=0}^{L}(i)^l(2l+1)h_l^{(1)}(kD)P_l(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{D}}) \quad (1)$$

where $h_l^{(1)}$ is the spherical Hankel function of the first kind, P_l is the Legendre polynomial, k is the wavenumber, and $\hat{\boldsymbol{k}}$ is a unit vector representing the angular directions. The centers of the basis and testing clusters are separated by the vector $\boldsymbol{D}$, where

$$\boldsymbol{D}=\hat{\boldsymbol{D}}D. \quad (2)$$

The summation in (1) is truncated at L, where L is the number of multipoles required to accurately represent the spectral contents of both the translation operator and the related radiation and receiving patterns. Considering cubic clusters with edges a and using the excess bandwidth formula [5] for the worst case scenario [6]

$$L\approx 1.73ka+2.16(d_0)^{2/3}(ka)^{1/3} \quad (3)$$

where d_0 is the desired number of digits of accuracy.

In Fig. 1(a), the truncation number L is plotted with respect to d_0 and for different values of the cluster size a increasing by a factor of two from 0.25λ to 64λ, where λ is the wavelength. For any problem, 0.25λ corresponds to the size of the clusters at the lowest level of the multilevel tree structure. On the other hand, the size of the largest clusters depends on the size of the problem. Fig. 1(a) demonstrates that L grows rapidly as the cluster size increases. For a fixed a, however, L increases gradually with respect to d_0 and the variation is small for large a.

Processing time required to calculate the translation operator in (1) is measured on a 1.8-GHz 64-bit Opteron-244 processor. In Fig. 1(b), the processing time is plotted with respect to the same parameters as in Fig. 1(a). The values are given for a single interaction between a pair of basis and testing clusters while a typical problem requires the calculation of numerous cluster–cluster interactions. Since $L=O(ka)$, the processing

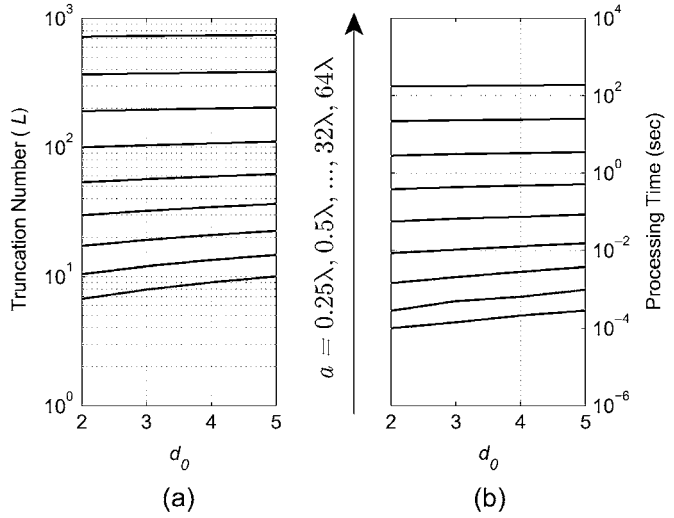


Fig. 1. (a) Truncation number as a function of d_0 and the cluster size a. (b) Processing time to compute the translation function for a single cluster–cluster interaction. In both figures, there are nine curves for different values of the cluster size increasing by a factor of two from 0.25λ to 64λ. The lowest and highest curves correspond to 0.25λ and 64λ, respectively.

time to evaluate (1) for a fixed $\hat{\boldsymbol{k}}$ is $O(ka)$. In addition, the number of angular directions $\hat{\boldsymbol{k}}$ is $O(L^2)$ and the processing time to evaluate (1) becomes $O(k^3a^3)$ for a cluster–cluster interaction. For low levels of MLFMA, $O(k^3a^3) = O(1)$, which is acceptable although the number of clusters in these levels is $O(N)$. However, for the largest clusters of a problem, $O(ka) = O(N^{1/2})$ and $O(k^3a^3) = O(N^{3/2})$. Therefore, as N becomes large, the processing time required to calculate the translation operators for a problem is dominated by the evaluations for the high-level clusters, although the number of these clusters is $O(1)$. In addition, the setup time for the translation matrix becomes dominant compared to the time required for other parts of MLFMA, even the matrix–vector multiplications that can be performed in $O(N \log N)$ time.

Defining the variable $\varphi = \cos^{-1}(\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{D}})$, the translation operator can be expressed as a band-limited function of φ [4] as

$$T(k, D, \varphi) = \frac{ik}{4\pi} \sum_{l=0}^{L} (i)^l (2l+1) h_l^{(1)}(kD) P_l(\cos\varphi). \quad (4)$$

Choosing an oversampling factor s and sampling the operator along φ from 0 to 2π at $\lfloor sL \rfloor = O(N)$ equally spaced points ($\lfloor \cdot \rfloor$ represents the floor operation), i.e., at $\varphi_i = 2\pi(i-1)/(\lfloor sL \rfloor - 1)$ and $i = 1, \ldots, \lfloor sL \rfloor$, the translation operator can be obtained by Lagrange interpolation at any point as

$$\tilde{T}(k, D, \varphi) = \sum_{i=f+1-p}^{f+p} T(k, D, \varphi_i) w_i(\varphi) \quad (5)$$

where $\tilde{T}$ represents the translation function perturbed by the interpolation error

$$f = \left\lfloor \frac{\varphi(\lfloor sL \rfloor - 1)}{2\pi} + 1 \right\rfloor \quad (6)$$

and

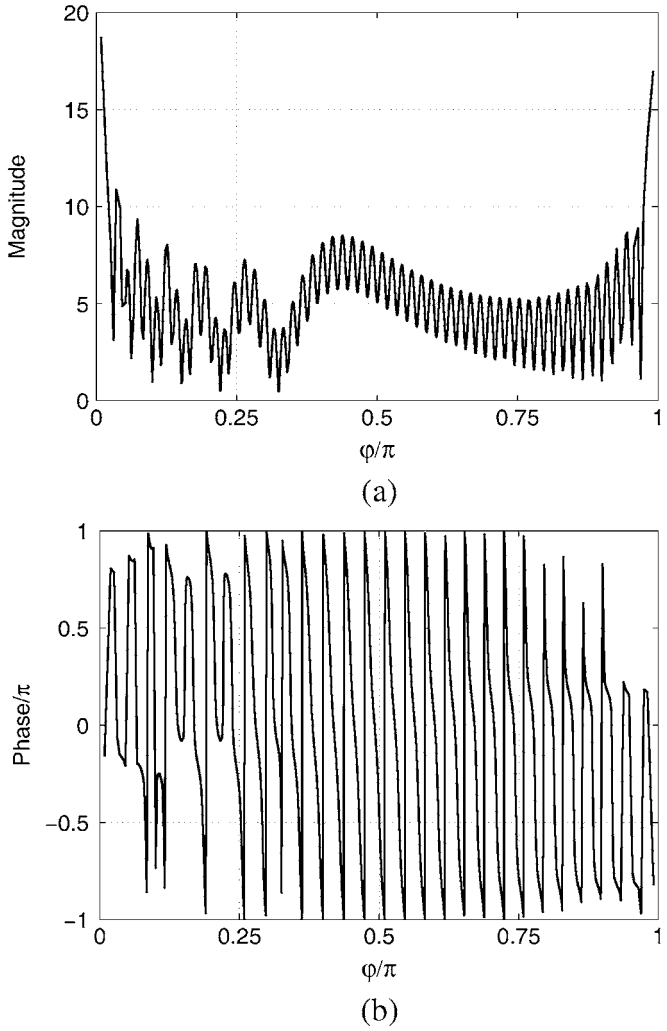


Fig. 2. (a) Magnitude and (b) phase of the translation function with respect to φ for the case of $a = 4\lambda$, $d_0 = 3$, and $\boldsymbol{D} = \hat{\boldsymbol{x}} 2a$.

$$w_i(\varphi) = \prod_{\substack{j=f+1-p \\ j \neq i}}^{f+p} \frac{\varphi - \varphi_j}{\varphi_i - \varphi_j}. \quad (7)$$

In (5) and (7), p is the number of interpolation points employed at each side of the target location φ.

III. Optimal Interpolation

Fig. 2(a) and (b) depicts the magnitude and phase of the translation operator, respectively, for two clusters separated by $\boldsymbol{D} = \hat{\boldsymbol{x}} 2a$, where $a = 4\lambda$. The number of accurate digits d_0 is 3 and $L = 57$. We perform the direct calculation of the translation operator, where the function is evaluated at the required points by using (4). In the ϕ direction, there are $2(L+1) = 116$ samples that are equally spaced from 0 to 2π. In the θ direction, there are $(L+1) = 58$ samples (zeros of the Legendre polynomial) and they are not equally spaced. Then, there are a total of $2(L+1)^2 = 6728$ distinct $\hat{\boldsymbol{k}}$ directions to evaluate the translation operator. It should be noted that the transform from (1) to (4) not only depends on L, but also on the relative positions of the clusters, i.e., it also depends on $\boldsymbol{D}$.

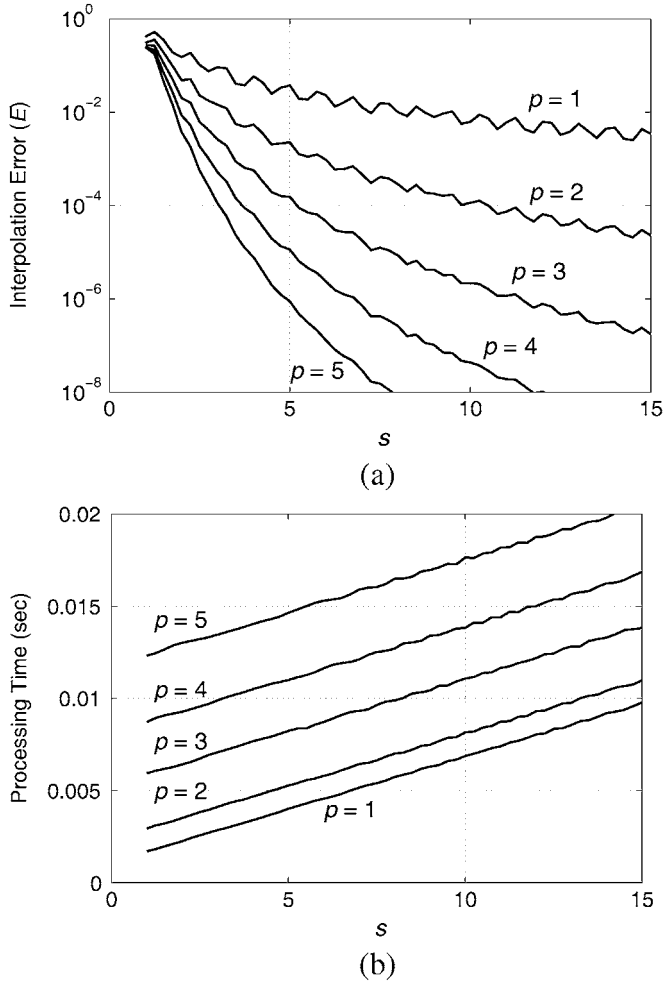


Fig. 3. (a) Interpolation error and (b) processing time with respect to interpolation parameters p and s for the translation function in Fig. 2.

Before the translation matrix is filled via Lagrange interpolation, the parameters s and p must be determined. For fixed values of d_0 and a, we perform a scan over the s and p parameters to find their optimal values. Fig. 3(a) demonstrates the interpolation error with respect to s and p for the case in Fig. 2. The interpolation error is defined as

$$E = \max\{E_n\}, \qquad E_n = \frac{|\tilde{T}(\varphi_n) - T(\varphi_n)|}{\max_{\varphi}\{|T(\varphi)|\}} \tag{8}$$

where $n = 1, \ldots, 2(L+1)^2$ and φ_n represents the sampling points. The interpolation error decreases when p or s is increased. In this case $d_0 = 3$, which means that MLFMA computes the interactions with three digits of accuracy. Thus, (p, s) pairs leading to larger than 10^{-3} error are not allowable. In other words, the error introduced by the interpolation of the translation operator should be adjusted according to the desired level of accuracy.

This strategy yields a set of (p, s) pairs satisfying the error criterion. Optimization is completed by choosing the (p, s) pair with the minimum processing time. As shown in Fig. 3(b), processing time (measured on a 1.8-GHz 64-bit Opteron-244 processor) to evaluate the translation operator increases as p or s is increased. Then, there exists an optimal (p, s) pair satisfying the desired level of accuracy with the minimum processing time. We scan the parameters p and s for various values of a and d_0. All possible values of $\boldsymbol{D}$ according to the one-box-buffer scheme [6] are also checked. In the end, we obtain the optimal values listed in Table I with the corresponding speedup compared to the direct calculation. We note that the values presented in Table I do not depend on the computer platform. The optimal (p, s) pairs are valid for $a \geq 4\lambda$ and they are found to be independent of $\boldsymbol{D}$. For smaller clusters, such as $a = \lambda$ or 2λ, the interpolation does not lead to a significant speedup, and therefore, we prefer to calculate these translations directly. In the case of much smaller clusters, such as $a = 0.25\lambda$ or 0.5λ, direct calculation is faster than the interpolation for any (p, s) pair satisfying the desired accuracy.

TABLE I
SPEEDUP OBTAINED BY USING THE OPTIMAL (p, s) PAIR FOR $a \geq 4\lambda$

d_0	(p, s)	$a = 4\lambda$	$a = 8\lambda$	$a = 16\lambda$	$a = 32\lambda$	$a = 64\lambda$
2	(2,3.5)	14.0	27.5	54.3	108.3	216.0
3	(2,6.5)	10.8	20.2	40.0	77.0	151.9
4	(3,6.0)	7.9	15.0	28.9	56.9	113.7
5	(3,8.5)	7.1	13.0	24.7	48.4	96.6

Fig. 4(a) and (b) compares the optimal (p, s) pairs to the fixed $p = 3, s = 5.0$ values suggested in [4]. In Fig. 4(a), the interpolation error is plotted with respect to the box size a from 4λ to 64λ and for different levels of accuracy, i.e., for $d_0 = 2, 3, 4$, and 5 corresponding to $10^{-2}, 10^{-3}, 10^{-4}$, and 10^{-5} relative errors, respectively. In the optimized case, the error is always below the desired level of accuracy. However, with fixed parameters, the error is not controllable and is localized around 10^{-4}. The corresponding speedup is plotted in Fig. 4(b), where it increases with increasing box size and decreases with increasing number of accurate digits in the optimized case. This relationship is also evident in Table I. Comparing Fig. 4(a) and Fig. 4(b), the following observations can be made.

1) For $d_0 = 2$ and 3, fixed $p = 3, s = 5.0$ satisfies the desired level of accuracy but the optimal (p, s) pairs provide higher speedup.
2) For $d_0 = 4$ and 5, the fixed $p = 3, s = 5.0$ seems to give higher speedup compared to the optimal (p, s) pairs, however, the accuracy is not satisfied with the fixed parameters.

Based on these observations, we conclude that optimization is essential to improve the interpolation of the translation operator.

IV. RESULTS

To demonstrate the overall improvement obtained with interpolation, we present the results of a scattering problem involving a conducting sphere of radius 20λ. This is a 1,462,854-unknown problem solved by a parallel MLFMA implementation with seven levels. The problem is solved on a cluster of 32 2.6-GHz Pentium-4 Celeron processors. The box size is 0.25λ for the lowest level and 16λ for the highest level. As an example, if the number of accurate digits d_0 is set to 3, then L takes values from 8 to 195. We use the one-box-buffer scheme and reduce the number of translations by exploiting the symmetry [7]. During the setup phase of the program, each processor checks all of its

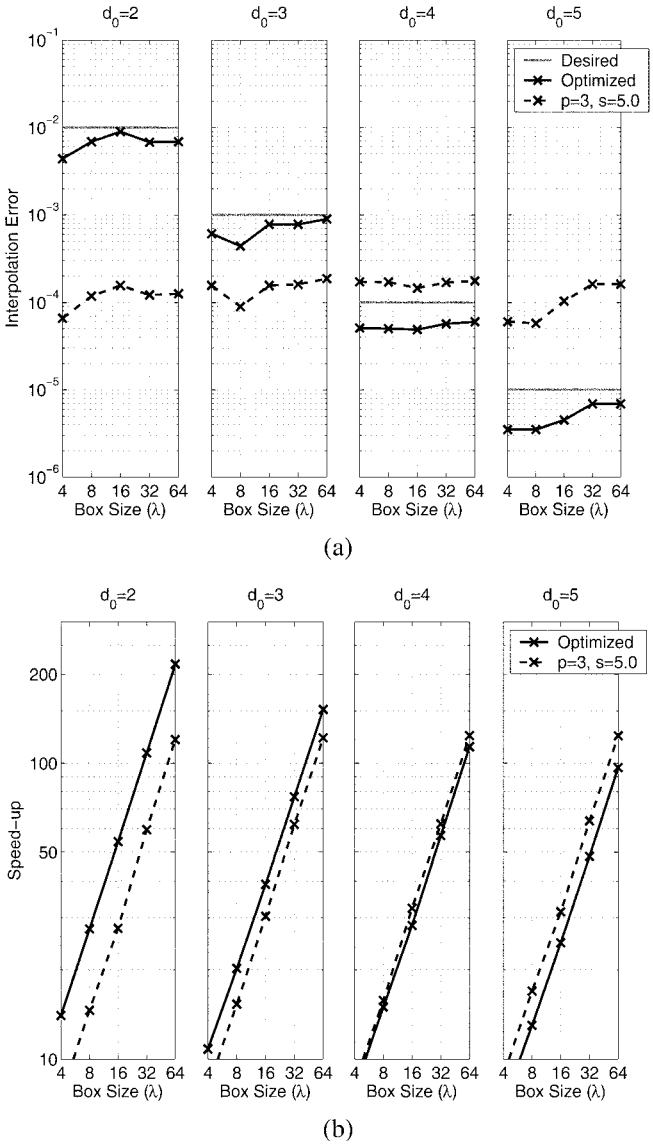


Fig. 4. (a) Interpolation error and (b) corresponding speedup for different box sizes from 4λ to 64λ and for $d_0 = 2, 3, 4, 5$. ($\boldsymbol{D} = \hat{\boldsymbol{x}}2a$).

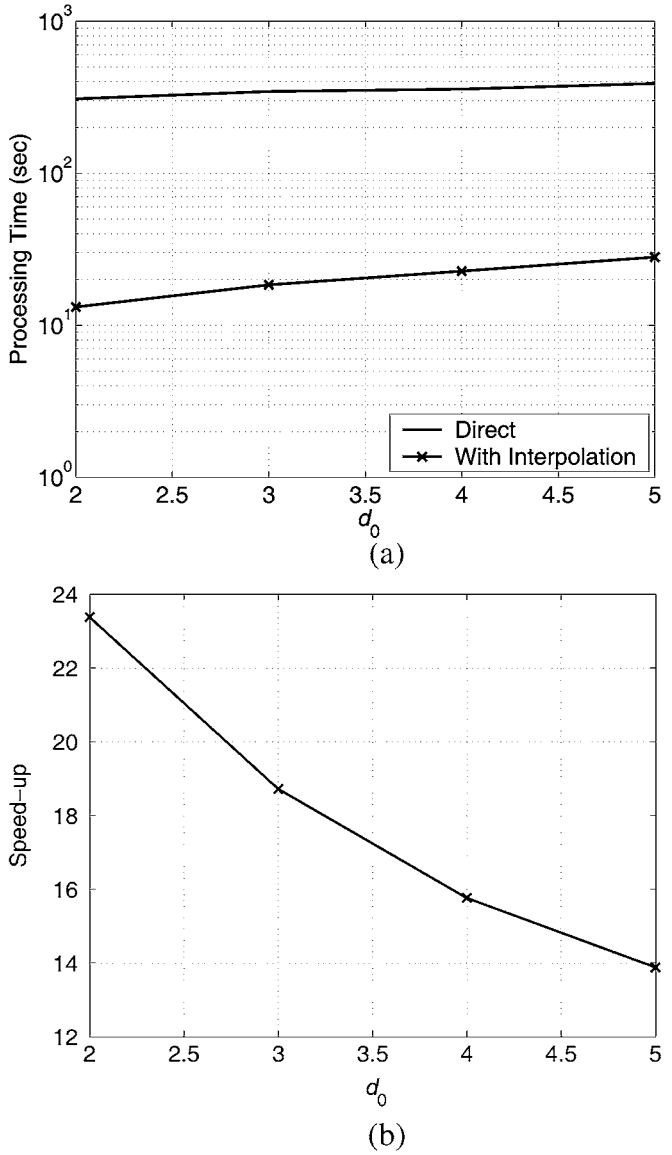


Fig. 5. (a) Processing time to compute the translation operators for a 1,462,854-unknown sphere problem. (b) Speedup obtained with optimal interpolation compared to direct calculation of the translation operators.

cluster–cluster interactions to eliminate the unneeded translations.

In Fig. 5(a), processing time for the calculation of the translation operators is plotted with respect to d_0. For both types of calculations (direct and interpolated), the maximum is chosen among the processing times spent by 32 processors. In Fig. 5(b), the speedup obtained by the interpolation method over direct calculation is plotted as a function of d_0. The speedup is over 14 up to $d_0 = 5$.

V. Conclusion

In this paper, we revisited the Lagrange interpolation of the translation operator in 3-D MLFMA. We optimized the number of interpolation points p and the oversampling factor s. In this way, the error becomes controllable and the processing time required to satisfy the desired level of accuracy is minimized.

References

[1] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston, MA: Artech House, 2001.

[2] C.-C. Lu and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[3] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propag. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993.

[4] J. Song and W. C. Chew, "Interpolation of translation matrix in MLFMA," *Microwave Opt. Technol. Lett.*, vol. 30, no. 2, pp. 109–114, Jul. 2001.

[5] S. Koc, J. M. Song, and W. C. Chew, "Error analysis for the numerical evaluation of the diagonal forms of the scalar spherical addition theorem," *SIAM J. Numer. Anal.*, vol. 36, no. 3, pp. 906–921, 1999.

[6] M. L. Hastriter, S. Ohnuki, and W. C. Chew, "Error control of the translation operator in 3D MLFMA," *Microwave Opt. Technol. Lett.*, vol. 37, no. 3, pp. 184–188, May 2003.

[7] S. Velamparambil, W. C. Chew, and J. Song, "10 million unknowns: Is that big?," *IEEE Antennas Propag. Mag.*, vol. 45, no. 2, pp. 43–58, Apr. 2003.

Özgür Ergül (S'98) was born in Yozgat, Turkey, in 1978. He received the B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2001 and 2003, respectively. He is currently pursuing the Ph.D. degree at Bilkent University.

Since 2001, he has served as a Teaching and Research Assistant in the Department of Electrical and Electronics Engineering at Bilkent University. From 2000 to 2005, he was affiliated with the Computational Electromagnetics Research Center (BilCEM). His research interests include fast and accurate algorithms for the solution of large and complicated structures, parallel programming, and iterative techniques.

Mr. Ergül's academic endeavors are supported by the Scientific and Technical Research Council of Turkey (TUBITAK) in the framework of a national Ph.D. scholarship.

Levent Gürel (S'87–M'92–SM'97) received the B.Sc. degree from the Middle East Technical University (METU), Ankara, Turkey, in 1986 and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign (UIUC), Urbana, in 1988 and 1991, respectively, all in electrical engineering.

He joined the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, in 1991, where he worked as a Research Staff Member on the electromagnetic compatibility (EMC) problems related to electronic packaging, on the use of microwave processes in the manufacturing and testing of electronic circuits, and on the development of fast solvers for interconnect modeling. Since 1994, he has been a faculty member in the Department of Electrical and Electronics Engineering of the Bilkent University, Ankara, where he is currently a Professor. He was a Visiting Associate Professor at the Center for Computational Electromagnetics (CCEM) of the UIUC for one semester in 1997. He returned to the UIUC as a Visiting Professor during 2003–2005, and as an Adjunct Professor during 2005–2006. He founded the Computational Electromagnetics Research Center (BiLCEM) at Bilkent University in 2005, where he is serving as the Director. His research interests include the development of fast algorithms for computational electromagnetics (CEM) and the application thereof to scattering and radiation problems involving large and complicated scatterers, antennas, and radars; frequency-selective surfaces; high-speed electronic circuits; optical and imaging systems; nanostructures; and metamaterials. He is also interested in the theoretical and computational aspects of electromagnetic compatibility and interference analyses. Ground penetrating radars and other subsurface scattering applications are also among his research interests.

Among the recognitions of Prof. Gürel's accomplishments, the two prestigious awards from the Turkish Academy of Sciences (TUBA) in 2002 and the Scientific and Technical Research Council of Turkey (TUBITAK) in 2003 are the most notable. He served as the Chairman of the AP/MTT/ED/EMC Chapter of the IEEE Turkey Section from 2000 to 2003. He founded the EMC Chapter in Turkey in 2000. He served as the Co-Chairman of the 2003 IEEE International Symposium on Electromagnetic Compatibility. He is a member of the General Assembly of the European Microwave Association, a member of the USNC of the International Union of Radio Science (URSI), and the Chairman of Commission E (Electromagnetic Noise and Interference) of URSI Turkey National Committee. He is currently serving as an Associate Editor of *Radio Science*.

Fast and accurate solutions of extremely large integral-equation problems discretised with tens of millions of unknowns

L. Gürel and Ö. Ergül

The solution of extremely large scattering problems that are formulated by integral equations and discretised with tens of millions of unknowns is reported. Accurate and efficient solutions are performed by employing a parallel implementation of the multilevel fast multipole algorithm. The effectiveness of the implementation is demonstrated on a sphere problem containing more than 33 million unknowns, which is the largest integral-equation problem ever solved to our knowledge.

Introduction: For numerical solutions of scattering problems in electromagnetics, integral-equation formulations provide accurate results when they are discretised appropriately by using small elements with respect to wavelength [1]. Simultaneous discretisations of the scatterer and the integral equations lead to dense matrix equations, which can be solved iteratively using efficient acceleration methods, such as the multilevel fast multipole algorithm (MLFMA) [2]. However, accurate solutions of many real-life problems require discretisations with millions of elements, which result in matrix equations with millions of unknowns. For the solutions of these large-scale problems, MLFMA must be parallelised, but this is not trivial owing to the complicated structure of the algorithm [3–5]. In this Letter, we report an implementation of the parallel MLFMA that is able to solve problems discretised with tens of millions of unknowns. Specifically, we present the results of a scattering problem involving a sphere of radius 96λ, where an accurate solution requires a discretisation with more than 33 million unknowns. To the best of our knowledge, this is the solution of the largest integral-equation problem reported up to now.

Parallel implementation of MLFMA: MLFMA performs the matrix-vector multiplications related to an $N \times N$ dense matrix equation in $\mathcal{O}(NL)$ time using $\mathcal{O}(NL)$ memory, where $L = O(\log N)$ is the number of levels of the tree structure, which is constructed by recursively dividing the computational domain into sub-domains (clusters). MLFMA calculates the far-field interactions between the radiating (basis) and receiving (testing) elements in a group-by-group manner consisting of three stages: aggregation, translation and disaggregation. For each matrix-vector multiplication required by the iterative solver, these stages are performed on the tree structure in a multilevel manner. There are also $O(N)$ near-field interactions that are calculated directly and stored in the memory to be used multiple times.

In the aggregation step, radiation patterns of the clusters are computed from the bottom of the tree structure to the top. Before the iterations, radiation patterns of the basis functions are calculated and stored in the memory. Owing to the nature of the Helmholtz equation, sampling rates of the radiation patterns depend on the sizes of the clusters. Using the excess-bandwidth formula and considering the worst-case scenario [6], we determine the number of samples for each level according to the desired accuracy. The samples are chosen uniformly in the ϕ direction while they are selected as the Gauss-Legendre points in the θ direction. During the aggregation process, sampling rates of the consecutive levels are matched by employing a local Lagrange interpolation algorithm with enhanced accuracy [7].

For the parallelisation of the aggregation process, we choose a level of distribution (LoD) to divide the clusters among the processors. Using a load-balancing algorithm, the levels below the LoD are distributed among the processors by assigning each cluster and its parent cluster to the same processor. In this way, aggregation operations can be performed independently in each processor from the bottom of the tree structure up to the LoD without any communication [8]. In the higher levels above the LoD, however, radiation patterns are distributed among the processors, instead of the clusters [5]. This is required in order to improve the load balancing since the higher levels include fewer clusters with densely-sampled fields. Then, an all-to-all communication is required at the LoD to switch between the two strategies applied in the lower and higher levels of the tree structure. We also note that one-to-one communications are required in the higher levels, where the fields are distributed among the processors and the interpolations in a processor require samples that are stored in other processors [5].

In the translation stage of MLFMA, radiated fields of the clusters are converted into incoming fields for other clusters. Translations are performed between pairs of clusters when the clusters are far from each other while their parent clusters are electromagnetically close to each other. Using a one-box-buffer scheme, there are $O(1)$ translation operations for each cluster in any level. Translation functions to perform these operations are calculated and stored in memory before the iterations. Using regularly-spaced cubic clusters, we significantly reduce the number of different translation functions required for each level [9]. In addition, calculation of the translation operators is accelerated by using local interpolation techniques that are optimised according to the desired level of accuracy [10]. Each translation operator is an infinite summation that must be truncated [2], where the truncation number is also determined by the excess-bandwidth formula [6].

In the lower levels below the LoD, some of the translations can be performed in each processor without any communication, while the rest are related to the clusters that are assigned to different processors so that communications to complete these translations are inevitable. We carefully organise the required data transfers by matching the processors appropriately using a communication map. For the upper levels above the LoD, all translations can be performed without any communication; this is another advantage of distributing fields instead of clusters [5]. After the translations, the disaggregation stage is performed as the inverse of the aggregation process. Incoming fields are calculated for each cluster from the top of the tree structure to the lowest level. The incoming field to a cluster is a combination of the incoming field to its parent cluster and the incoming fields due to the translations. We use transpose interpolation to accurately match the different sampling rates of the successive levels [7]. At the end of the disaggregation, a numerical integration is performed for each testing function in the lowest level to complete the matrix-vector multiplications related to the far-field interactions. Finally, matrix-vector multiplications related to near-field interactions are performed directly. For high efficiency, it is crucial to distribute the near-field interactions among the processors according to a load-balancing strategy, which usually leads to different partitioning schemes for the near-field and far-field interactions [8].

Table 1: MLFMA solution of a sphere problem with 33 791 232 unknowns

Geometry size (diameter)	192λ
Number of processors	16
Number of levels	9
Smallest cluster size	0.19λ
Total number of clusters	5 904 951
Number of clusters in lowest level	4 344 205
Number of near-field interactions	3 732 101 432
Truncation numbers (2 digits of accuracy)	6 to 546
Number of iterations (BiCGStab and 10^{-3} residual error)	21
Setup time (minutes)	177
Solution time (minutes)	265
Time for matrix-vector multiplication (s)	370
Memory for translation functions (GB)	2
Memory for radiation/receiving patterns (GB)	56
Memory for nearfield interactions (GB)	28
Memory for aggregation/disaggregation arrays (GB)	79

Results: To demonstrate the efficiency and accuracy of our implementation, we present the results of a sphere problem with radius 96λ. The discretisation of the problem with a mesh size of $\lambda/10$ leads to 33 791 232 unknowns when Rao-Wilton-Glisson [11] functions are employed as the basis and testing functions on triangular domains. The scattering problem is formulated with the combined-field integral equation [1] and iteratively solved by a biconjugate-gradient-stabilised (BiCGStab) algorithm. The solution is performed on a cluster of quad-core Intel Xeon 5355 processors connected via an Infiniband network and the results are summarised in Table 1, where we list the clustering information, processing times and memory usage. Using a block-diagonal preconditioner, only 21 iterations are

required to reduce the residual error below 10^{-3}. Parallelising the solution into 16 processes, the iterative solution is completed in 265 min. Finally, the bistatic radar cross-section (RCS) values are shown in Fig. 1, where the computed values sampled at 0.1° are in agreement with the analytical curve obtained by a Mie-series solution. In the Figure, 180° corresponds to the forward-scattering direction and the root-mean-square error [3] of the RCS is only 0.915 dB in the 170–180° range.

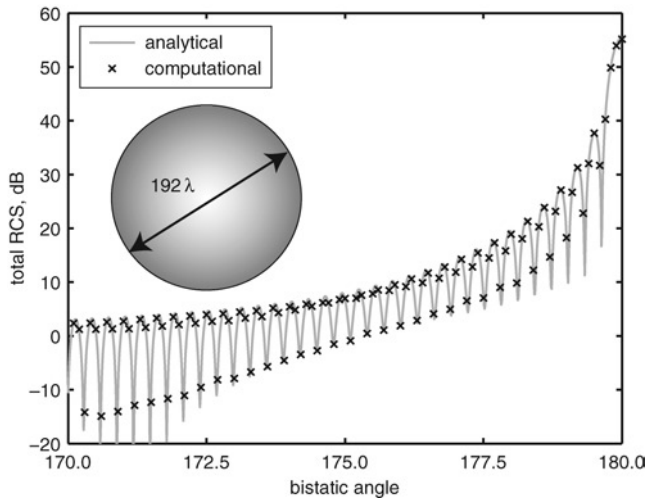


Fig. 1 *Bistatic RCS of sphere of radius 96λ*

Computational values obtained by solution of 33 791 232-unknown problem are in agreement with analytical curve obtained by Mie-series solution

Conclusions: We have presented the integral-equation solution of a scattering problem involving a sphere of radius 96λ discretised with 33 791 232 unknowns, which corresponds to the solution of a dense matrix equation with more than 10^{15} nonzero elements. This is the largest integral-equation problem reported to date. By employing an efficient implementation of the parallel MLFMA, it becomes possible to solve such large-scale problems on relatively inexpensive computational platforms.

Acknowledgments: This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grant 105E172, by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), and by contracts from ASELSAN and SSM. Computer time was provided in part by a generous allocation from Intel Corporation.

5 March 2007
Electronics Letters online no: 20070639
doi: 10.1049/el:20070639

L. Gürel (*Computational Electromagnetics Research Center (BiLCEM) and Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara TR-06800, Turkey*)

E-mail: lgurel@bilkent.edu.tr

Ö. Ergül (*Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara TR-06800, Turkey*)

References

1 Poggio, A.J., and Miller, E.K.: 'Computer techniques for electromagnetics' (Permagon Press, Oxford, UK, 1973, chap. 4)
2 Song, J., Lu, C.-C., and Chew, W.C.: 'Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects', *IEEE Trans. Antennas Propag.*, 1997, **45**, (10), pp. 1488–1493
3 Hastriter, M.L.: 'A study of MLFMA for large-scale scattering problems', PhD thesis, University of Illinois at Urbana-Champaign, 2003
4 Sylvand, G.: 'Performance of a parallel implementation of the FMM for electromagnetics applications', *Int. J. Numer. Meth. Fluids*, 2003, **43**, pp. 865–879
5 Velamparambil, S., and Chew, W.C.: 'Analysis and performance of a distributed memory multilevel fast multipole algorithm', *IEEE Trans. Antennas Propag.*, 2005, **53**, (8), pp. 2719–2727
6 Chew, W.C., Jin, J.-M., Michielssen, E., and Song, J.: 'Fast and efficient algorithms in computational electromagnetics' (Artech House, Boston MA, USA, 2001)
7 Ergül, Ö., and Gürel, L.: 'Enhancing the accuracy of the interpolations and anterpolations in MLFMA', *IEEE Antennas Wirel. Propag. Lett.*, 2006, **5**, pp. 467–470
8 Ergül, Ö., and Gürel, L.: 'Efficient parallelization of multilevel fast multipole algorithm'. Proc. European Conference on Antennas and Propagation (EuCAP), 2006, 350094
9 Velamparambil, S., Chew, W.C., and Song, J.: '10 million unknowns: Is it that big?', *IEEE Antennas Propag. Mag.*, 2003, **45**, (2), pp. 43–58
10 Ergül, Ö., and Gürel, L.: 'Optimal interpolation of translation operator in multilevel fast multipole algorithm', *IEEE Trans. Antennas Propag.*, 2006, **54**, (12), pp. 3822–3826
11 Rao, S.M., Wilton, D.R., and Glisson, A.W.: 'Electromagnetic scattering by surfaces of arbitrary shape', *IEEE Trans. Antennas Propag.*, 1982, **AP-30**, (3), pp. 409–418

Hierarchical parallelisation strategy for multilevel fast multipole algorithm in computational electromagnetics

Ö. Ergül and L. Gürel

A hierarchical parallelisation of the multilevel fast multipole algorithm (MLFMA) for the efficient solution of large-scale problems in computational electromagnetics is presented. The tree structure of MLFMA is distributed among the processors by partitioning both the clusters and the samples of the fields appropriately for each level. The parallelisation efficiency is significantly improved compared to previous approaches, where only the clusters or only the fields are partitioned in a level.

Introduction: Surface integral equations are commonly used to formulate electromagnetic scattering and radiation problems involving complicated three-dimensional objects with arbitrary shapes [1]. By discretising the integral-equation formulations, we obtain dense matrix equations. They can be solved iteratively by accelerating the matrix-vector multiplications using the multilevel fast multipole algorithm (MLFMA) [2]. Using MLFMA, matrix-vector multiplications related to an $N \times N$ dense matrix equation can be performed in $O(N\log N)$ time using $O(N\log N)$ memory. However, accurate solutions of many real-life problems require discretisations with millions of unknowns, which cannot be solved easily by the sequential implementations of MLFMA running on a single processor. To solve such large problems, it is helpful to increase computational resources by assembling parallel computing platforms and at the same time by parallelising MLFMA. In this way, it has become possible to solve problems with 20–30 million unknowns on relatively inexpensive computing platforms [3–8]. On the other hand, parallelisation of MLFMA is not trivial owing to the complicated structure of this algorithm. Simple parallelisation strategies usually fail to provide efficient solutions because of the communication among the processors and the unavoidable duplication of some of the computations over multiple processors [9]. In this Letter, we present a hierarchical strategy for the efficient parallelisation of MLFMA. We compare our strategy with previous parallelisation schemes to demonstrate the improved efficiency, especially when the number of processors is large.

Tree structure of MLFMA: Elements of an $N \times N$ matrix obtained by the discretisation of a surface integral equation correspond to the interactions of the basis and testing functions defined on the surface of the object. MLFMA performs the matrix-vector multiplications efficiently by calculating these interactions in a group-by-group manner involving three main stages, i.e. aggregation, translation and disaggregation [2]. These stages are performed in a multilevel scheme using a tree structure constructed by including the scatterer in a cubic box and recursively dividing the computational domain into subboxes. During the aggregation stage, radiated fields at the centres of the clusters (nonempty boxes) are calculated proceeding from the bottom of the tree structure to the highest level. Then, the translation stage is performed by translating the radiated fields at the centres of the clusters to the incoming fields at the centres of other clusters in the same level. Finally, the total incoming fields at the centres of the clusters are calculated from the top of the tree structure to the lowest level during the disaggregation stage.

In the lowest level of the multilevel tree, there are $O(N)$ clusters. The number of clusters decreases from each level to the next upper level and it becomes $O(1)$ in the highest level involving translations. The number of samples for the radiated and incoming fields depends on cluster size as measured by the wavelength. Therefore, fields of the clusters in the lower levels are sampled coarsely, while the fields of the clusters in the higher levels require finer sampling. Considering the number of clusters and the samples of the fields, all levels of MLFMA have $O(N)$ complexity in terms of processing time and memory. As a consequence, an efficient parallelisation of MLFMA should attempt to obtain the best partitioning for each level by minimising the communications and duplications among the processors.

Partitioning of multilevel tree: For the parallelisation of MLFMA, the main task is to distribute the tree structure among the processors. A simple partitioning of a three-level tree is shown in Fig. 1*a*, where the levels are represented by two-dimensional rectangular boxes including various numbers of clusters (horizontal dimension) and samples of the fields (vertical dimension). Each level is partitioned among eight processors. In the simple partitioning scheme, clusters in all levels are distributed among the processors and each cluster at any level is assigned to a single processor. This strategy works efficiently for lower levels involving many clusters. For higher levels, however, it is difficult to distribute small numbers of clusters among the processors without duplication [9]. In addition, dense communications among the processors during the translations become significant for higher levels since large amounts of data are transferred, which reduces the efficiency of the parallelisation significantly [6, 9].

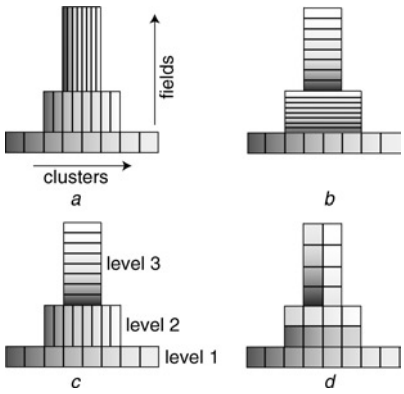


Fig. 1 *Various strategies for partitioning of tree structure of MLFMA*
a Simple partitioning, where clusters are distributed in all levels
b, *c* Hybrid partitioning with shared and distributed levels
d Hierarchical partitioning

To improve the parallelisation efficiency, a hybrid partitioning approach is introduced in [6], where different strategies are applied for lower and higher levels of the tree structure. As shown in Figs. 1*b* and 1*c*, the simple partitioning scheme is preserved in lower (distributed) levels so that the clusters in these levels are still distributed among the processors. In higher (shared) levels, however, processor assignments are made on the basis of the fields of the clusters, not on the basis of the clusters themselves. In other words, each cluster is shared by all processors and each processor is assigned to the same portion of the fields of all clusters. In this way, higher levels are distributed efficiently among the processors, since the fields in those levels have high sampling rates. In addition, the translations in the shared levels can be performed efficiently without any communication among the processors.

The hybrid partitioning strategy increases the parallelisation efficiency significantly compared to the simple partitioning approach. Nevertheless, there are some levels at the middle of the tree structure (such as level 2 in Fig. 1) where distributing neither the fields nor the clusters among the processors is efficient. For such levels, even though distributing the fields eliminates the communication during the translations, dense communication is required elsewhere, i.e. for the interpolation and anterpolation operations during the aggregation and disaggregation stages, respectively [6]. Although such one-to-one data transfers are not problematic for higher levels (such as level 3 in Fig. 1), they become important for lower levels, where the number of processors is comparable to the number of samples. Therefore, even if the numbers of the shared and distributed levels are optimised, sufficient parallelisation efficiency may not be achieved.

In this Letter, we introduce a hierarchical partitioning scheme to further improve the parallelisation efficiency compared to the hybrid approach. This strategy is illustrated in Fig. 1*d*, where the partitioning is performed in both directions (clusters and samples of the fields) for all levels; we adjust the partitioning appropriately by considering the numbers of clusters and the samples of the fields at each level. In the lowest level, the clusters are distributed among the processors without any partitioning for the fields. Then, in the next level (level 2), the samples of the fields are divided between pairs of processors, while we reduce the number of partitions for the clusters by a factor of two. As we proceed to higher levels, the numbers of partitions for the clusters and the fields are systematically decreased and increased, respectively. In this way, the computations for all levels are distributed among the processors with improved load-balancing compared to partitioning with respect to only clusters or only samples of the fields.

With the strategy of partitioning in both dimensions, three different types of communications are required for each level (except for the lowest level) in the hierarchical parallelisation scheme. Consider level 2 in Fig. 1*d*; some of the processors need to communicate during the translations because of the partitioning of the clusters. Similarly, one-to-one communications are required during the aggregation and disaggregation stages owing to the partitioning of the fields. In addition to these, we also need data exchanges among the processors to modify the number of partitions between any two consecutive levels. Although the hierarchical partitioning increases the types of communication compared to the simple and the hybrid approaches, the amount of data transferred is not increased and the number of communication events is reduced. Hence, larger data packages are transferred at fewer times. This improves both communications and the load-balancing significantly.

Results: To demonstrate the improved efficiency of the hierarchical parallelisation, we present the solution of a scattering problem involving a conducting sphere of radius 20 λ discretised with 1 462 854 unknowns. The sphere is illuminated by a plane wave and seven-level MLFMA is used to solve the problem on a cluster of quad-core Intel Xeon 5355 processors connected via an Infiniband network. Fig. 2 shows the efficiency when the solution is parallelised into 2, 4, 8, 16, 32, 64 and 128 processors.

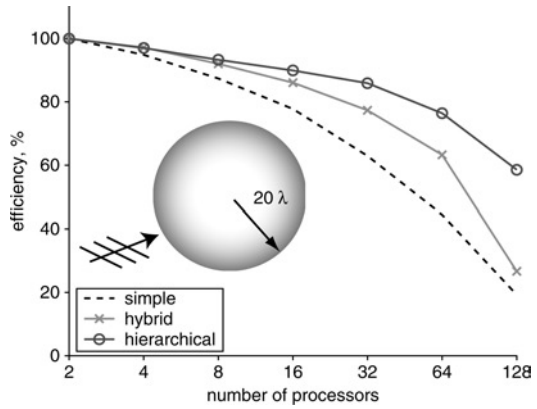


Fig. 2 *Parallelisation efficiency for solution of scattering problem involving sphere of radius 20 λ discretised with 1 462 854 unknowns*

The parallelisation efficiency is defined as

$$\varepsilon_p = \frac{2T_2}{pT_p} \tag{1}$$

where T_p is the processing time of the solution with p processors. Fig. 2 shows that the hierarchical parallelisation improves the efficiency significantly compared to both simple and hybrid parallelisation approaches. All parallelisation schemes are optimised via load-balancing algorithms. Although the hybrid parallelisation, which includes three shared levels, performs better than the simple parallelisation scheme, its efficiency drops below 30% for 128 processors. In this case, the hierarchical parallelisation provides 60% efficiency, which corresponds to 38-fold speed-up compared to the two-processor solution. Using 128 processors and the hierarchical parallelisation scheme, the total processing time, including the setup and the iterative solution with 27 BiCGStab iterations, is only 300 s for this 1.5-million-unknown problem.

Conclusions: Using a hierarchical strategy, the parallelisation efficiency of MLFMA can be improved significantly. Compared to previous approaches based on partitioning in one direction (only clusters or only samples of the fields), hierarchical parallelisation provides higher efficiency, especially when the number of processors is large.

Acknowledgments: This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grants 105E172 and 107E136, by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), and by contracts from ASELSAN and SSM. Computer time was provided in part by a generous allocation from Intel Corporation.

18 August 2007
Electronics Letters online no: 20082282
doi: 10.1049/el:20082282

Ö. Ergül and L. Gürel (*Computational Electromagnetics Research Center (BiLCEM) and Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara TR-06800, Turkey*)

E-mail: ergul@ee.bilkent.edu.tr

References

1 Poggio, A.J., and Miller, E.K.: 'Computer techniques for electromagnetics' (Pergamon Press, Oxford, UK., 1973, chap. 4)
2 Song, J., Lu, C.-C., and Chew, W.C.: 'Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects', *IEEE Trans. Antennas Propag.*, 1997, **45**, (10), pp. 1488–1493
3 Velamparambil, S., Chew, W.C., and Song, J.: '10 million unknowns: is it that big?', *IEEE Antennas Propag. Mag.*, 2003, **45**, (2), pp. 43–58
4 Hastriter, M.L.: 'A study of MLFMA for large-scale scattering problems', PhD thesis, University of Illinois at Urbana-Champaign, 2003
5 Sylvand, G.: 'Performance of a parallel implementation of the FMM for electromagnetics applications', *Int. J. Numer. Methods Fluids*, 2003, **43**, pp. 865–879
6 Velamparambil, S., and Chew, W.C.: 'Analysis and performance of a distributed memory multilevel fast multipole algorithm', *IEEE Trans. Antennas Propag.*, 2005, **53**, (8), pp. 2719–2727
7 Gürel, L., and Ergül, Ö.: 'Fast and accurate solutions of extremely large integral-equation problems discretised with tens of millions of unknowns', *Electron. Lett.*, 2007, **43**, (9), pp. 499–500
8 Ergül, Ö., and Gürel, L.: 'Fast and accurate solutions of large-scale scattering problems with parallel multilevel fast multipole algorithm'. Proc. IEEE Antennas and Propagation Soc. Int. Symp, 2007, pp. 3436–3439
9 Ergül, Ö., and Gürel, L.: 'Efficient parallelization of multilevel fast multipole algorithm'. Proc. European Conf. on Antennas and Propagation (EuCAP), 2006, 350094

Efficient Parallelization of the Multilevel Fast Multipole Algorithm for the Solution of Large-Scale Scattering Problems

Özgür Ergül, *Student Member, IEEE,* and Levent Gürel, *Senior Member, IEEE*

***Abstract*—We present fast and accurate solutions of large-scale scattering problems involving three-dimensional closed conductors with arbitrary shapes using the multilevel fast multipole algorithm (MLFMA). With an efficient parallelization of MLFMA, scattering problems that are discretized with tens of millions of unknowns are easily solved on a cluster of computers. We extensively investigate the parallelization of MLFMA, identify the bottlenecks, and provide remedial procedures to improve the efficiency of the implementations. The accuracy of the solutions is demonstrated on a scattering problem involving a sphere of radius 110λ discretized with 41 883 638 unknowns, the largest integral-equation problem solved to date. In addition to canonical problems, we also present the solution of real-life problems involving complicated targets with large dimensions.**

***Index Terms*—Electromagnetic scattering, fast solvers, integral equations, multilevel fast multipole algorithm (MLFMA), parallel algorithms.**

I. Introduction

SURFACE integral equations are commonly used to formulate scattering problems involving three-dimensional conducting bodies with arbitrary shapes [1]. These formulations provide accurate results when they are discretized appropriately by using small elements with respect to wavelength. Simultaneous discretizations of the scatterer and the integral equations lead to dense matrix equations, which can be solved iteratively using efficient acceleration methods, such as the multilevel fast multipole algorithm (MLFMA) [2]. However, accurate solutions of many real-life problems require discretizations with millions of elements leading to matrix equations with millions of unknowns. To solve these large problems, it is helpful to increase computational resources by assembling parallel computing platforms and at the same time by parallelizing the solvers.

Of the various parallelization schemes for MLFMA, the most popular use distributed-memory architectures by constructing clusters of computers with local memories connected via fast networks [3]–[11]. Parallelization tools are available, such as the message passing interface (MPI). Such tools provide many communication protocols to organize parallel solutions. However, parallelization of MLFMA is not trivial because of the complicated structure of this algorithm [11]. Simple parallelization strategies usually fail to provide efficient solutions because of the communications between the processors and the unavoidable duplication of some of the computations over multiple processors. Consequently, there have been many efforts to improve the parallelization of MLFMA by minimizing duplications and communications [7]–[12]. Thanks to these efforts, it has become possible to solve 20–30 million unknowns on relatively inexpensive computing platforms [8], [9], [13].

In this paper, we present the details of a parallel MLFMA implementation for the efficient solution of scattering problems involving tens of millions of unknowns. We extensively investigate the parallelization procedure by focusing on different parts of the algorithm and identifying the obstacles to parallelization efficiency. Our approach involves load-balancing and partitioning techniques to distribute the tasks equally among the processors and to minimize the interprocessor communications. We demonstrate the accuracy and efficiency of our implementations on canonical problems involving sphere geometries of various sizes. Specifically, we are able to solve problems with more than 40 million unknowns on relatively inexpensive platforms. In addition to canonical problems, we also solve real-life problems involving complicated geometries discretized with large numbers of unknowns.

The scattering problems considered in this paper involve closed surfaces, which can be formulated with the combined-field integral equation (CFIE) [1]. CFIE provides better-conditioned matrix equations than the electric-field integral equation (EFIE) and the magnetic-field integral equation (MFIE) [14]–[16]. Using CFIE, iterative convergence is achieved rapidly and it can be further accelerated by employing simple and efficient preconditioners.

The rest of the paper is organized as follows. In Section II, we examine the MLFMA solutions, focusing on the computational requirements. Section III explores efficient parallelization of MLFMA by investigating each part of the algorithm in detail. Section IV presents the results, followed by our concluding remarks in Section V.

II. Solution of Integral Equations by MLFMA

For the solution of scattering problems involving three-dimensional conducting bodies with arbitrary shapes, discretization of the surface integral equations leads to $N \times N$ dense

Manuscript received June 24, 2007; revised November 8, 2007. Published August 6, 2008 (projected). This work was supported in part by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grants 105E172 and 107E136, in part by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), and in part by contracts from ASELSAN and SSM. Computer time was provided in part by a generous allocation from Intel Corporation.

The authors are with the Department of Electrical and Electronics Engineering and the Computational Electromagnetics Research Center (BiLCEM), Bilkent University, TR-06800 Bilkent, Ankara, Turkey (e-mail: ergul@ee.bilkent.edu.tr; lgurel@bilkent.edu.tr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAP.2008.926757

matrix equations

$$\sum_{n=1}^{N} Z_{mn}^{E,M,C} a_n = v_m^{E,M,C}, \quad m = 1, 2, \ldots, N \tag{1}$$

where a_n represents the unknown coefficients of the basis functions $\boldsymbol{b}_n(\boldsymbol{r})$ for $n = 1, 2, \ldots, N$ to model the surface current density, i.e.,

$$\boldsymbol{J}(\boldsymbol{r}) \approx \sum_{n=1}^{N} a_n \boldsymbol{b}_n(\boldsymbol{r}). \tag{2}$$

Expressions for the matrix elements (Z_{mn}^{E}, Z_{mn}^{M}, and Z_{mn}^{C}) and the elements of the right-hand side vector (v_m^{E}, v_m^{M}, and v_m^{C}) for EFIE, MFIE, and CFIE, respectively, are presented in [17]. For the solution of problems involving closed surfaces, CFIE is preferable since it is free of the internal-resonance problem [18] and provides better-conditioned matrix equations than EFIE and MFIE [14]–[16]. This favorable quality of CFIE is crucial for the rapid convergence of iterative solutions. In this paper, CFIE is discretized by employing Rao-Wilton-Glisson (RWG) [19] functions defined on planar triangles for numerical solutions.

A. Solutions With MLFMA

MLFMA splits the matrix-vector multiplications (MVMs) required by the iterative solvers as

$$\bar{\boldsymbol{Z}} \cdot \boldsymbol{x} = \bar{\boldsymbol{Z}}_{NF} \cdot \boldsymbol{x} + \bar{\boldsymbol{Z}}_{FF} \cdot \boldsymbol{x}. \tag{3}$$

In (3), the near-field interactions denoted by $\bar{\boldsymbol{Z}}_{NF}$ are calculated directly and stored in memory, while the far-field interactions ($\bar{\boldsymbol{Z}}_{FF}$) are computed approximately in a group-by-group manner. For a single-level fast multipole algorithm, we calculate the far-field interactions as presented in [20]. In MLFMA, those interactions are calculated in a multilevel scheme using a tree structure constructed by including the scatterer in a cubic box and recursively dividing the computational domain into subboxes. The tree structure of MLFMA includes $L = O(\log N)$ levels. At level l from 1 to L, the number of nonempty boxes (clusters)[1] is N_l, where $N_1 = O(N)$ and $N_L = O(1)$. Each MVM involves four main stages.

- Near-field interactions: In MLFMA, near-field interactions are used directly to perform the multiplication

$$\boldsymbol{y} = \bar{\boldsymbol{Z}}_{NF} \cdot \boldsymbol{x}. \tag{4}$$

  The number of near-field interactions is proportional to N^2/N_1 and the near-field matrix $\bar{\boldsymbol{Z}}_{NF}$ has a sparsity of $1/N_1$.
- Aggregation: Radiated fields at the centers of the clusters are calculated from the bottom of the tree structure to the highest level.

[1]In this paper, the term "cluster" is used in two different contexts. Its meanings in "clusters of computers" and to indicate the nonempty boxes in the MLFMA tree should be distinguishable from the context.

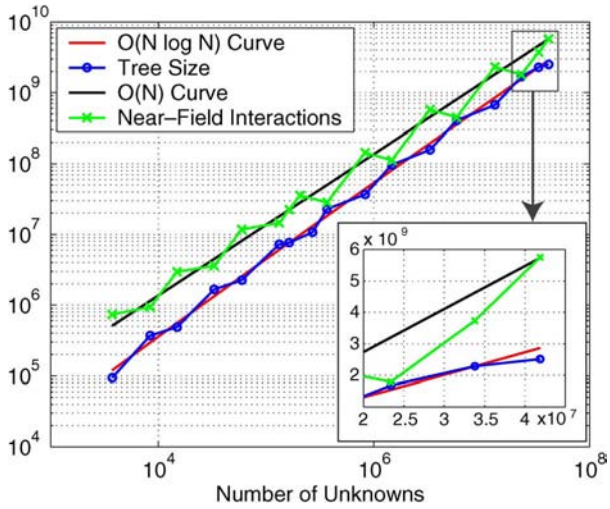


Fig. 1. Tree size and the number of near-field interactions for the solutions of the sphere problems using top-down strategy to construct the multilevel tree.

- Translation: Radiated fields are translated into incoming fields. For a basis cluster at any level, there are $O(1)$ testing clusters to translate the radiated field.
- Disaggregation: The incoming fields at the centers of the clusters are calculated from the top of the tree structure to the lowest level. At the lowest level, the incoming fields are multiplied by the receiving patterns of the testing functions and angular integrations are performed to complete the MVM.

In our MLFMA implementations, radiated and incoming fields are sampled uniformly in the ϕ direction, while we use the Gauss-Legendre quadrature in the θ direction [21]. There are a total of $(T_l + 1) \times (2T_l + 2)$ samples required for a cluster in level l, where T_l is the truncation number, i.e., the number of harmonics used to calculate the translation operators. To determine the value of T_l for each level, we use the excess bandwidth formula considering the worst-case scenario according to a one-box-buffer scheme [22], i.e.,

$$T_l \approx 1.73 k a_l + 2.16 (d_0)^{2/3} (k a_l)^{1/3} \tag{5}$$

where a_l is the box size at level l and d_0 is the desired digits of accuracy. Oscillatory nature of the Helmholtz solutions requires that the truncation number T_l and the sampling rate for the radiated and incoming fields depend on cluster size as measured by the wavelength ($\lambda = 2\pi/k$). During the aggregation and disaggregation stages, we employ local Lagrange interpolation and anterpolation methods to match the different sampling rates of the consecutive levels [23], [24].

B. Computational Requirements of MLFMA

When MLFMA is used, memory requirement for a MVM (M_{MVM}) is proportional to the tree size S_T, i.e.,

$$M_{MVM} \propto S_T = \sum_{l=1}^{L} 2 N_l (T_l + 1)^2. \tag{6}$$

TABLE I
MAJOR PARTS OF MLFMA AND THEIR COMPUTATIONAL REQUIREMENTS

	MEMORY		
PART	PROPORTIONAL TO	COMPLEXITY	SIGNIFICANCE
MVM	$\sum_{l=1}^{L} N_l(T_l+1)^2$	$O(N \log N)$	Significant
Radiation and Receiving Patterns	$N(T_1+1)^2$	$O(N)$	Significant
Translation Operators	$\sum_{l=1}^{L} (T_l+1)^2$	$O(N)$	Insignificant
Near-Field Interactions	N^2/N_1	$O(N)$	Significant
	PROCESSING TIME		
PART	PROPORTIONAL TO	COMPLEXITY	SIGNIFICANCE
MVM	$\sum_{l=1}^{L} c_l N_l(T_l+1)^2$	$O(N \log N)$	Significant
Radiation and Receiving Patterns	$N(T_1+1)^2$	$O(N)$	Insignificant
Translation Operators	$\sum_{l=1}^{L} (T_l+1)^2$	$O(N)$	Insignificant
Near-Field Interactions	N^2/N_1	$O(N)$	Significant

The processing time (T_{MVM}) is also related to the tree size as

$$T_{MVM} \propto \sum_{l=1}^{L} c_l N_l (T_l+1)^2 \tag{7}$$

where c_l represents relative weights for levels $l = 1, 2, \ldots, L$. Asymptotically, as N increases, $N_l(T_l+1)^2$ becomes $O(N)$ and the complexity of the MVM is $O(N \log N)$. Although this is true in general, measurements may present deviations from the ideal case depending on the construction technique for the tree structure, even when N is very large. For example, we usually employ a top-down strategy to build the multilevel tree for large problems. In this strategy, the smallest possible cubic box is used to enclose the target completely. Then, the computational domain is recursively divided into subdomains until the size of the clusters in the lowest level is in the 0.15λ–0.30λ range. In Fig. 1, tree size (S_T) is plotted as a function of the number of unknowns for the solution of scattering problems involving sphere geometries of various sizes, when the top-down strategy is used to construct the multilevel tree and the number of accurate digits d_0 is 2. The radius of the sphere changes from λ to 110λ corresponding to 3723 and 41 883 648 unknowns (edges), respectively, using $\lambda/10$ triangulation. We observe that the tree size oscillates around the $O(N \log N)$ curve. Due to such local variations, processing time and memory requirement for the MVMs with respect to N cannot be strictly proportional to $N \log N$. As an example, the tree size grows only by 50% when the number of unknowns increases from 23 405 664 to 41 883 648. Then, the memory requirement for the MVMs increases by about 50%, which is below the asymptotical estimation of 85%.

The radiation and receiving patterns of the basis and testing functions are sampled according to the sampling rate of the lowest level clusters. Using the RWG functions, these patterns are calculated analytically and stored in memory before the iterative solutions. Applying a Galerkin scheme and using the same sets of basis and testing functions, CFIE implementations require only two sets of patterns for each RWG function [25]. We also reduce the number of samples to $(T_1/2+1) \times (2T_1+2)$ using the symmetry of the patterns. Although the processing time to calculate the radiation and receiving patterns is negligible, significant amount of memory is required to store them.

Similar to the radiation and receiving patterns, translation operators are also calculated and stored in memory before the iterations. Using cubic (identical) clusters, there is a maximum of $7^3 - 3^3 = 316$ different translations in each level, independent of the number of clusters [7]. Although using cubic clusters reduces the number of translation operators significantly, we also need interpolation methods to calculate these operators in $O(N)$ time [26], [27]. With the optimization of the interpolations, both calculation time and memory for the translation operators are insignificant compared to the other parts of the implementation, especially when the problem size is large.

Processing time for the initial setup of MLFMA (prior to the iterative solution) is dominated by calculating near-field interactions and it is proportional to N^2/N_1. The amount of memory to store the near-field interactions is also significant and comparable to the memory used for the radiation and receiving patterns. Asymptotically, $N_1 = O(N)$ and the near-field interactions has a complexity of $O(N)$. However, similar to the MVMs, local variations in the processing time and memory requirement for the near-field interactions may exhibit behavior different than the asymptotical estimation. This is because, as depicted in Fig. 1, the number of near-field interactions oscillates around the $O(N)$ curve when a top-down strategy is used to construct the tree structure. Consequently, variation in processing time and memory with respect to N can be higher or lower than the asymptotically linear estimate.

As a summary, Table I lists the major parts of MLFMA and their computation requirements for the solution of large problems.

III. EFFICIENT PARALLELIZATION OF MLFMA

Because of its complicated structure, parallelization of MLFMA is not trivial. Simple parallelization schemes usually lead to inefficient solutions due to dense communications between the processors, duplication of computations, and unbalanced distribution of the workload among processors. Several issues must be carefully considered to obtain an efficient parallelization of MLFMA [7]–[12].

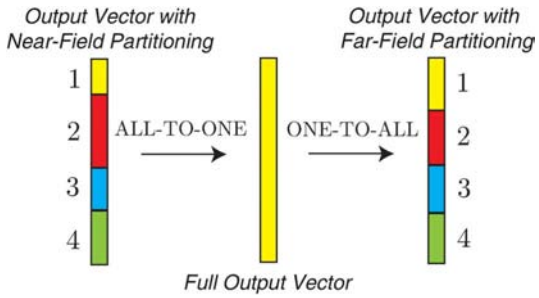


Fig. 2. Communications performed in each MVM to match the near-field and far-field partitioning schemes.

- Partitioning: For high efficiency, it is essential to distribute the tree structure among the processors with minimal duplication. This is achieved by using different partitioning strategies for the lower and higher levels of the tree structure [11]. In the lower levels (distributed levels), there are many clusters with small numbers of samples for the radiated and incoming fields. Therefore, it is appropriate to distribute the clusters in these levels by assigning each of them to a single processor. In higher levels (shared levels), however, it is easier to distribute the fields among the processors by assigning each cluster to all processors, since there are a few clusters in these levels with large numbers of samples. Calculation of the far-field interactions are organized according to the partitioning of the tree structure (far-field partitioning).
- Load-balancing: Parallelization cannot be achieved efficiently without distributing the tasks equally among the processors. We apply load-balancing for both the distributed and shared levels to improve the parallelization of the far-field interactions. For high efficiency, it is also essential to distribute the near-field interactions using a load-balancing algorithm [12].
- Communications: In parallel MLFMA, processors need to communicate with each other to transfer data. Using appropriate partitioning schemes and load-balancing algorithms significantly reduces the data traffic. However, the remaining communications must be organized carefully. For high efficiency, it is also essential to use high-speed networks to connect the processors.

In the following subsections, we provide the details of the efficient parallelization of MLFMA.

A. Setup Part

The setup part consists of preparing the near-field interactions, radiation and receiving patterns, translation operators, and preconditioners for the iterative solutions.

1) Near-Field Interactions: Near-field interactions should be distributed among the processors using a load-balancing algorithm. Considering the sparse near-field matrix, the rows $m = 1, 2, \ldots, N$ are assigned to the processors in such a way that all processors have approximately equal numbers of near-field interactions (near-field partitioning). Distributing the rows equally among the processors usually fails to provide good load-balancing, even for the solution of problems involving symmetrical geometries, such as a sphere. After distribution, the near-field interactions are calculated in each processor without any communication.

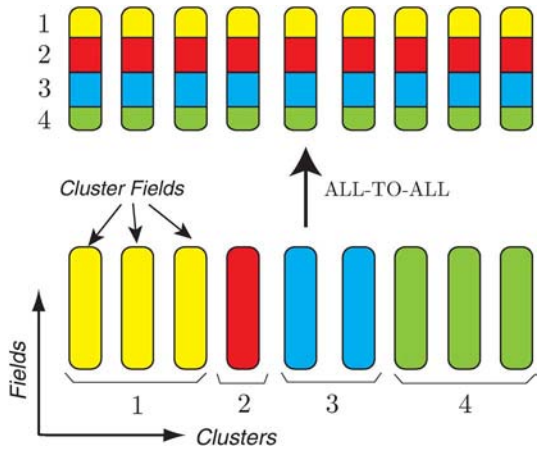


Fig. 3. All-to-all communications performed at LoD to change the far-field partitioning scheme from the distributed levels to the shared levels.

2) Radiation and Receiving Patterns: According to the far-field partitioning of the tree structure, the lowest-level clusters are distributed among the processors. Then, each processor calculates and stores the radiation and receiving patterns of the basis and testing functions included in its local tree.

3) Translation Operators: In the setup of MLFMA, each processor is tasked with calculating a set of translation operators that will be required during the MVMs. For a translation at a distributed level, where each cluster is assigned to a single processor, the operator is calculated by the processor working on the testing cluster. Due to symmetry, a translation operator can be used for many interactions in a level. Therefore, in the distributed levels, some of the translation operators are duplicated and included in more than one processor; this is allowable because of the negligible cost of the operators at the low levels. There is no duplication in the shared levels, where the fields are distributed and the translation operators are also partitioned among the processors.

4) Preconditioner: With CFIE, iterative solvers can be easily accelerated by employing simple and efficient preconditioners [15]. We use the block-diagonal preconditioner (BDP) [2] based on the self interactions of the lowest level clusters. The construction of BDP requires negligible time and memory, and its efficient parallelization is relatively easy to achieve.

B. Solution Part

For the iterative solutions, we employ Krylov subspace algorithms that are parallelized efficiently [28]. These algorithms require MVMs and the solutions of a sparse equation involving the preconditioner matrix $\bar{\boldsymbol{M}}$, i.e.,

$$\bar{\boldsymbol{Z}} \cdot \boldsymbol{x} = \boldsymbol{y} \tag{8}$$

$$\bar{\boldsymbol{M}} \cdot \boldsymbol{y} = \boldsymbol{x} \tag{9}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the input and output vectors, respectively; both are distributed according to the far-field partitioning. Before an MVM or a preconditioner solution, the partitions of the

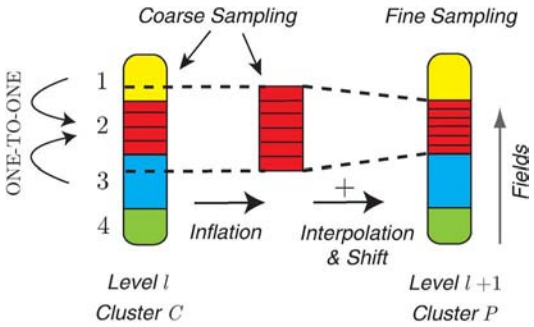


Fig. 4. Interpolations in the shared levels involving one-to-one communications.

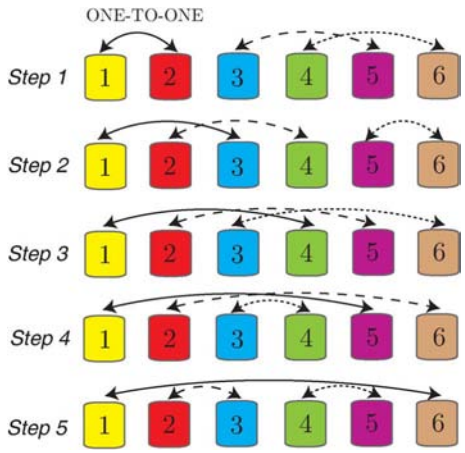


Fig. 5. Processor pairing for the translations in the distributed levels.

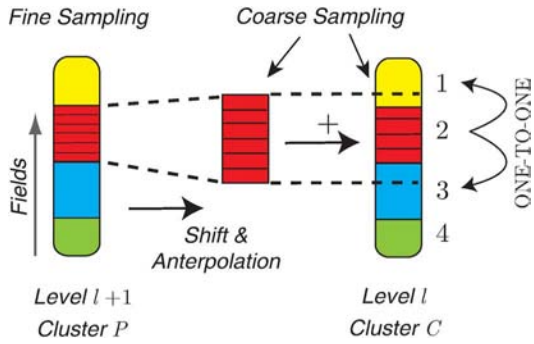


Fig. 6. Anterpolations (transpose interpolations) in the shared levels involving one-to-one communications.

TABLE II
COMMUNICATIONS REQUIRED IN THE MATRIX-VECTOR MULTIPLICATIONS BY PARALLEL MLFMA

PART	COMMUNICATION
Near-Field Stage	All-to-One and One-to-All
Distributed Aggregation/Disaggregation	None
Aggregation/Disaggregation in LoD	All-to-All
Shared Aggregation/Disaggregation	One-to-One
Distributed Translation	One-to-One
Shared Translation	None

input vector $\boldsymbol{x}$ are combined together using the "gather" operation of MPI. Each MVM involves the use of near-field interactions, as well as the calculation of the far-field interactions via the aggregation, translation, and disaggregation stages.

1) Near-Field Stage: To match the near-field and far-field partitioning schemes during the MVMs, all-to-one and one-to-all communications are required, as depicted in Fig. 2. After the near-field computations are performed in negligible time, the partitioning of the output vector is modified for the iterative solver. The processing time for these communications is also negligible.

2) Aggregation Stage: In the highest distributed level, which we call the level of distribution (LoD), the clusters are distributed among the processors using a load-balancing algorithm that considers the combined load of all descendants (children, grandchildren, etc.) of each cluster at LoD. The combined load for a cluster is the size of the subtree attached to the cluster; we account for all descendants, each weighted by the number of field samples. The load-balancing algorithm assigns the whole branch of the tree starting at an LoD cluster to the same processor. Then, in the distributed levels, each cluster and all its subclusters are assigned to the same processor. In this way, the aggregation stage up to LoD can be performed without any communication. At LoD, the partitioning scheme is changed by employing an all-to-all communication, as shown in Fig. 3. For each cluster, the samples of the radiated field stored in a processor is distributed among all processors. In the shared levels above LoD, $(T_l+1)\times 2(T_l+1)$ samples on the θ - ϕ space are partitioned along the θ direction.

From LoD to the highest level L, the aggregation stage involves one-to-one communications that are required for the interpolation of the fields. This is illustrated in Fig. 4, where an interpolation is performed on the samples of cluster C. As an example, only the interpolation in processor 2 is depicted although similar operations are also performed in the other processors. To compute the data at each sample in the fine grid, a set of samples are used in the coarse grid. Even though a local interpolation method is used, some of those coarse samples may be located in other processors. Therefore, one-to-one communications are performed to provide the required data (inflation). After the data is prepared, interpolation and shifting operations are performed to include the contribution of the cluster C in the radiated field of its parent cluster P.

We note that the communications in the shared levels are mainly required between the processors located "close to each other." In other words, the processor with index n_p requires data from its "neighbors," i.e., n_p-1 and n_p+1. On the other hand, depending on the partitioning and the number of interpolation points, more data might be required from other processors next to the neighbors. We apply a load-balancing algorithm to distribute the fields appropriately so that the amount of the data transferred among all processors is minimized. However, as the number of processes increases and the fields are distributed over many processors, dense one-to-one communications cannot be avoided; this may reduce the efficiency of the parallelization.

Finally, for each problem, we carefully choose the number of distributed and shared levels by an optimization. For this purpose, we assign LoD to a series of possible levels and monitor

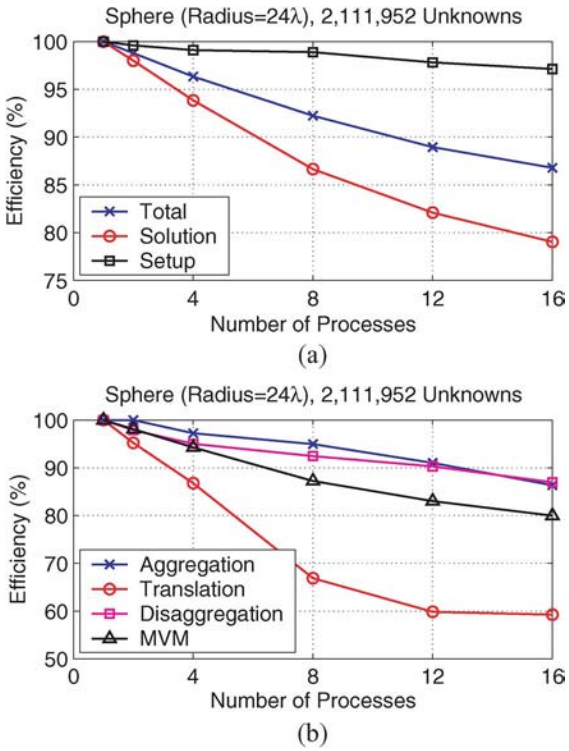


Fig. 7. Parallelization efficiency for the solution of a scattering problem involving a sphere of radius 24λ discretized with 2 111 952 unknowns.

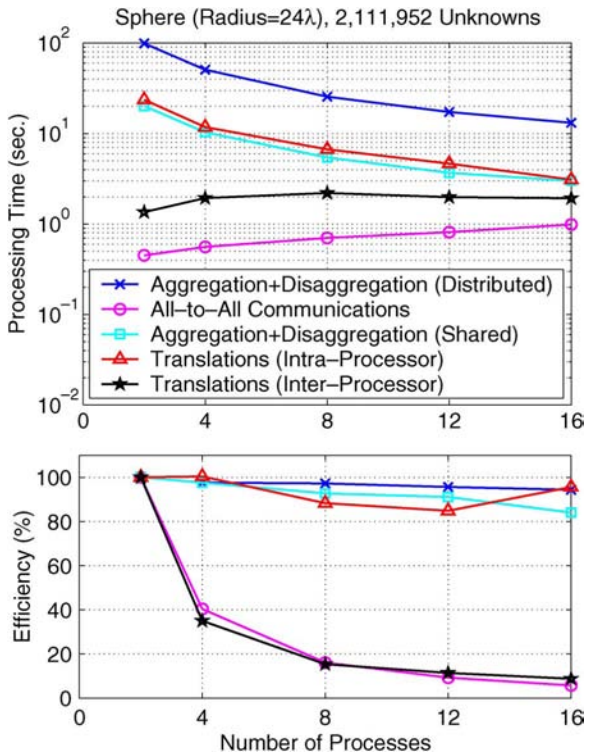


Fig. 8. Processing time and parallelization efficiency for various categorized parts of the MVMs for the solution of a scattering problem involving a sphere of radius 24λ discretized with 2 111 952 unknowns.

the distribution of the clusters and the fields. For some of the levels (higher levels), distribution of the fields is better than the distribution of the clusters, i.e., samples of the fields can be partitioned evenly among the processors, but not the clusters. For the others (lower levels), however, clusters can be partitioned easily, while it is difficult to partition the fields among the processors. Then, we choose LoD such that distributing the fields (clusters) is more preferable for all levels above (below) LoD. The choice of LoD depends on the tree structure (hence the geometry of the target) as well as the number of processors. However, our measurements show that, for a given problem, LoD is insensitive to the latter parameter if only a small number (e.g., 2 to 16) of processors are employed.

3) Translation Stage: The translation stage is one of the most critical parts for the efficiency of the parallelization. This is because dense one-to-one communications are required between the processors for the translations in the distributed levels. In general, each processor sends some data (radiated fields) to all other processors. We organize these communications using a communication map, which consists of interaction layers to match the processors. For p processors, it can be shown that the communications can be achieved in $p-1$ steps, as depicted in Fig. 5 for a 6-process case. After the processors are paired, the following operations are performed on the receiver and sender sides.

- The sender and receiver determine the cluster-cluster interactions involving the basis clusters on the sender side and testing clusters on the receiver side.
- The radiated fields of the basis clusters are sent one by one.
- When the radiated field of a basis cluster is received by the receiver, all of the translations involving this basis cluster and the testing clusters owned by the receiver are performed. This ensures that the same data is not transferred more than once.

To improve the efficiency of the translations, we use nonblocking send and receive operations of MPI to transfer the data. In the shared levels, all the translations are performed without any communication since the fields are distributed among the processors and a processor is assigned to the same portion of the radiated or incoming fields for all clusters.

4) Disaggregation Stage: The disaggregation stage is generally the inverse of the aggregation stage. The incoming fields are calculated at the center of each cluster from the top of the tree structure to the lowest level using the anterpolation and shift operations. For a cluster in level $l < L$, the incoming field is the combination of the translated field from the far-field clusters and the incoming field to the center of its parent. In the shared levels, anterpolation produces samples in the coarse grid, some of which should be sent to the "neighboring" processors. This is illustrated in Fig. 6, where processor 2 performs the anterpolation operation on the samples of cluster P for its subcluster C. Some of the resulting data in the coarse grid is used locally, while the rest are sent to other processors, i.e., exactly the reverse of the interpolation. As the disaggregation operation proceeds down to LoD, the partitioning is changed via an all-to-all communication. Then, the disaggregation is performed from LoD to the lowest level without any communication. In the lowest level, each processor performs the angular integrations and produces a partition of the output vector $\boldsymbol{y}$.

To sum up, Table II lists the communications required at each stage of the MVMs.

TABLE III
SOLUTIONS OF LARGE SPHERE PROBLEMS WITH MLFMA PARALLELIZED INTO 16 PROCESSES

Diameter	160λ	192λ	220λ
Unknowns	23,405,664	33,791,232	41,883,648
CLUSTERING			
Number of Levels	9		
Smallest Cluster Size	0.16λ	0.19λ	0.21λ
Number of Clusters	5,769,254	5,904,951	5,975,507
Lowest-Level Clusters	4,225,343	4,344,205	4,405,952
Near-Field Sparsity	3.27×10^{-6}	3.27×10^{-6}	3.28×10^{-6}
Truncation Number	5 to 457	6 to 546	6 to 623
Tree Size	1.68×10^{9}	2.29×10^{9}	2.51×10^{9}
PROCESSING TIME (Intel Xeon 5355 processors connected via an Infiniband network)			
Setup Times (minutes)	94	183	274
BiCGStab Iterations	17	21	19
MVM Time (seconds)	270	372	441
Solution Time (minutes)	155	264	290
MEMORY USAGE			
Translation Operators	1.5 GB	2.1 GB	2.7 GB
Radiation and Receiving Patterns	25.8 GB	57.4 GB	71.2 GB
Near-Field Interactions	13.4 GB	27.8 GB	42.9 GB
BD Preconditioner	1.3 GB	2.6 GB	4.0 GB
MVM	40.3 GB	55.9 GB	65.3 GB

IV. RESULTS

First, we demonstrate the efficiency of MLFMA parallelization for the solution of a scattering problem involving a sphere of radius 24λ. The problem is discretized with 2 111 952 unknowns and solved on a cluster of Intel Xeon processors connected via an Infiniband network. Fig. 7 depicts the efficiency (with respect to the solution with a single processor) when the solution is parallelized into 2, 4, 8, 12, and 16 processes. The parallelization efficiency is defined as

$$\varepsilon_p = \frac{T_1}{pT_p} \tag{10}$$

where T_p is the processing time of the solution with p processes. Fig. 7(a) shows that the overall efficiency (setup and iterative solution) is above 85% when the number of processes is 16. In this case, efficiency ratios for the setup and the solution parts are about 97% and 80%, respectively. We observe in Fig. 7(a) that the setup part is parallelized very efficiently, since this part is communication-free and the computations (especially the near-field interactions) are perfectly distributed to the processors using a load-balancing algorithm.

In Fig. 7(b), we present the parallelization efficiency for the aggregation, translation, and disaggregation stages, in addition to the overall efficiency for the MVMs. The near-field stage is not considered because of its negligible time. We observe that aggregation and disaggregation stages are parallelized with about 87% efficiency, while efficiency for the translation stage is 59% for the 16-process case. To further investigate the parallelization, Fig. 8 presents processing time and efficiency (with respect to the solution with 2 processors) for various categorized parts of the MVMs. Our observations are as follows.

- Aggregation and disaggregation stages in the distributed levels ($l = 1,2,3,4$ for this problem) constitute the significant part of the processing time of MVM. These stages are perfectly parallelized, thanks to the load-balancing algorithm for distributed levels.
- The parallelization efficiency of the aggregation and disaggregation stages in the shared levels (from $l = 5$ to $l = L = 7$ in this problem) is also quite high. However, the efficiency drops to about 80% for the 16-process case. This is due to the increasing amount of one-to-one communications for interpolations and anterpolations.
- Parallelization efficiency of the communication-free (intraprocessor) translations is in the 80%–100% range. All of the translations in the shared levels and some of those in the distributed levels are communication-free.
- Translations that are performed with communications (interprocessor translations) and all-to-all communications performed at LoD exhibit reduced efficiency as the number of processes increases. Since they take longer processing time, the interprocessor translations affect the overall efficiency more than the all-to-all communications.

In general, interprocessor translations are the bottleneck of the parallelization. Since these translations are performed in the distributed levels, their negative contributions can be minimized

by increasing the number of shared levels. However, aggregation and disaggregation in low levels cannot be performed efficiently by partitioning the coarsely sampled fields. As discussed in Section III, we carefully determine the number of distributed and shared levels to optimize the parallelization efficiency for the solution of each problem.

In Table III, we present the solutions of very large scattering problems involving spheres of radii 80λ, 96λ, and 110λ, which are discretized with 23 405 664, 33 791 232, and 41 883 648 unknowns, respectively. For all three problems, 9-level MLFMA is employed and parallelized into 16 processes. The numbers of distributed and shared levels are 6 and 3, respectively. Using a top-down strategy, the cluster size in the lowest level is 0.16λ–0.21λ. Each of the tree structures contains about six million clusters and most of them are used in the lowest level. The number of near-field interactions increases with the problem size and the sparsity of the near-field matrix is almost constant. The near-field interactions are calculated with 1% error. The smallest and largest truncation numbers are also listed in Table III when the far-field interactions are calculated with two digits of accuracy.

Table III shows that the setup time increases proportionally to N^2 since the sparsity of the near-field matrix is constant and the number of near-field interactions is proportional to N^2. On the other hand, the processing time for the MVMs, which is related to the tree size, increases more slowly than $O(N \log N)$. As discussed in Section II, these local deviations from the asymptotical estimates are expected depending on the clustering technique used for the tree structure. As depicted in Fig. 1, the tree size and the number of near-field interactions oscillate around the $O(N \log N)$ and $O(N)$ curves, respectively. Local variations of these quantities corresponding to the three large problems in Table III are magnified in the inset of Fig. 1. We observe that the tree size (hence the computational requirements for the MVMs) increases slower than the asymptotical estimate of $O(N \log N)$. On the other hand, due to a top-down clustering scheme, the number of near-field interactions (hence the computational requirements for the near-field part) grows faster than $O(N)$. We emphasize that this behavior is local and depends on the strategy to construct the tree structure, the overall complexity of MLFMA is still $O(N \log N)$.

We also observe in Table III that the maximum number of biconjugate-gradient-stabilized (BiCGStab) iterations to reduce the residual error below 10^{-3} is 21. Using the BDP, iterative solution of the 42-million-unknown problem requires only 290 min, while each MVM is performed in 441 s. Table III also lists the total memory usage for different parts of the algorithm using the single-precision representation for the complex numbers.

In Fig. 9, we further present the details of the solution of the 23-million-unknown problem involving a sphere of radius 80λ. In Fig. 9(a), the total processing time is depicted for all processes from 1 to 16. After the input and the clustering part$^{(1)}$, computations of the translation operators$^{(2)}$ and the radiation/receiving patterns$^{(4)}$ require negligible time. Calculation of the near-field interactions$^{(3)}$ dominates the setup time, which is about 94 min. Then the solution part$^{(5)}$, involving a total of 34 MVMs, is performed in about 155 min. The processing time for a MVM is depicted in Fig. 9(b),

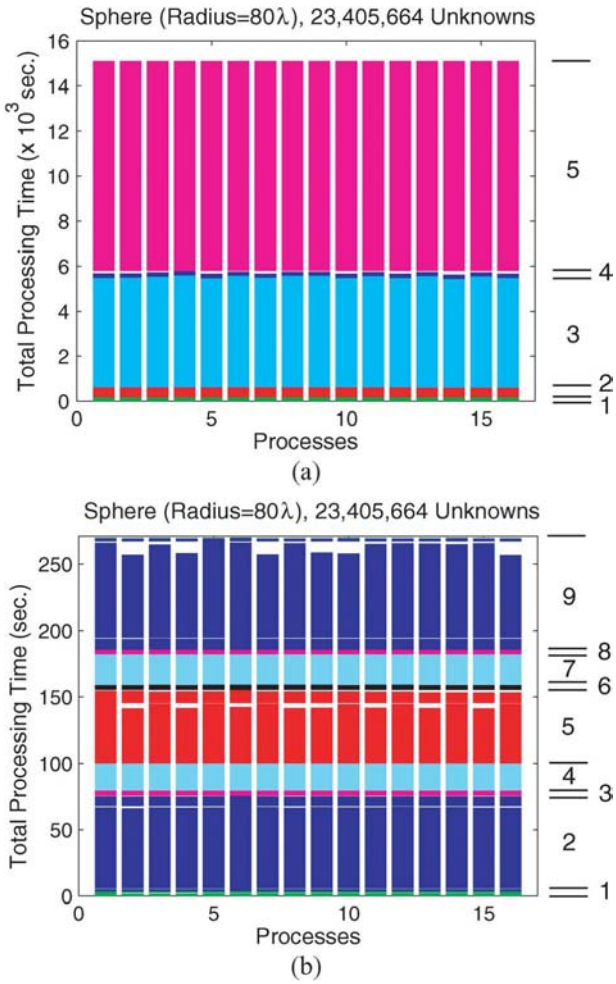


Fig. 9. Time diagrams for the solution of a scattering problem involving a sphere of radius 80λ discretized with 23 405 664 unknowns. (a) Overall time includes the input and clustering parts$^{(1)}$, calculation of the translation matrices$^{(2)}$, calculation of the near-field interactions$^{(3)}$, calculation of the radiation and receiving patterns$^{(4)}$, and the iterative solution$^{(5)}$. (b) Matrix-vector multiplications include the near-field stage$^{(1)}$, aggregation in the distributed levels$^{(2)}$, all-to-all communications in LoD$^{(3,8)}$, aggregation in the shared levels$^{(4)}$, translations without communications$^{(5)}$, translations with communications$^{(6)}$, disaggregation in the shared levels$^{(7)}$, and disaggregation in the distributed levels followed by the receiving operation$^{(9)}$. In the diagrams, white areas correspond to waits before the operations that require synchronization.

including the near-field stage$^{(1)}$, aggregation/disaggregation in the distributed levels$^{(2,9)}$, all-to-all communications$^{(3,8)}$, aggregation/disaggregation in the shared levels$^{(4,7)}$, communication-free (intraprocessor) translations$^{(5)}$, and interprocessor translations$^{(6)}$. The most problematic parts in terms of parallelization efficiency, i.e., all-to-all communications and interprocessor translations, require negligible time compared to other parts of the MVM. This is commonly observed with large-sized problems and supports the conclusion that the parallelization efficiency for a fixed number of processes usually increases as the problem size grows.

To present the accuracy of the solutions, Fig. 10 depicts the normalized bistatic radar cross section (RCS/λ^2) values in decibels (dB) for a sphere of radius 110λ discretized with 41 883 648 unknowns. We believe this is the solution of the largest integral-equation problem ever reported. Solutions of

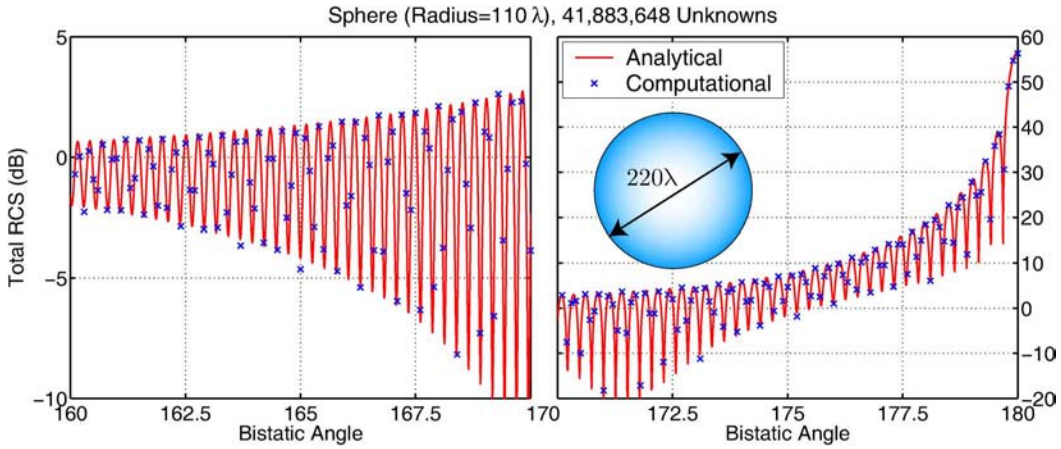


Fig. 10. Bistatic RCS (in dB) of a sphere of radius 110λ discretized with 41 883 648 unknowns from 160° to 180°, where 180° corresponds to the forward-scattering direction.

integral-equation problems with 20 million and 33 million unknowns were reported in [8] and [13], respectively. Analytical values obtained by a Mie-series solution is plotted as a reference from 160° to 180°, where 180° corresponds to the forward-scattering direction. Fig. 10 shows that the computational values sampled at 0.1° are in agreement with the analytical curve. For more quantitative information, we define a relative error as

$$e_R = \frac{||\boldsymbol{A}-\boldsymbol{C}||_2}{||\boldsymbol{A}||_2} \tag{11}$$

where $\boldsymbol{A}$ and $\boldsymbol{C}$ are the analytical and computational RCS values, respectively, $||.||_2$ is the l^2-norm defined as

$$||\boldsymbol{x}||_2 = \sqrt{\sum_{s=1}^{S} |\boldsymbol{x}[s]|^2} \tag{12}$$

and S is the number of samples. The relative error is 3.87%, 4.67%, and 4.67% in the 160°–170°, 170°–180°, and 0°–180° ranges, respectively. We note that the relative error in the RCS values is about 5%, although we calculate the near-field and far-field interactions with 1% error. The extra error is due to the low-order discretization of CFIE. For the same discretization of a scattering problem with the RWG functions, MFIE (thus, CFIE) is consistently inaccurate to calculate the scattered fields compared to EFIE, even MFIE (and CFIE) is better conditioned than EFIE [29]. A remedy to this accuracy problem is to use higher-order basis functions, such as the linear-linear basis functions discussed in [17].

Finally, we present the solution of a real-life problem involving the Flamme, which is a stealth airborne target, as detailed in [30]. The scattering problem is solved at 16 GHz and the maximum dimension of the Flamme is 6 m, corresponding to 320λ. Using $\lambda/10$ triangulation, the problem is discretized with 24 782 400 unknowns. Fig. 11 presents the bistatic RCS values in dBm^2 when the target is illuminated by a plane wave propagating in the x-y plane at a 30° angle from the x axis (from $\phi = 30°$). Both θ and ϕ polarizations are considered.

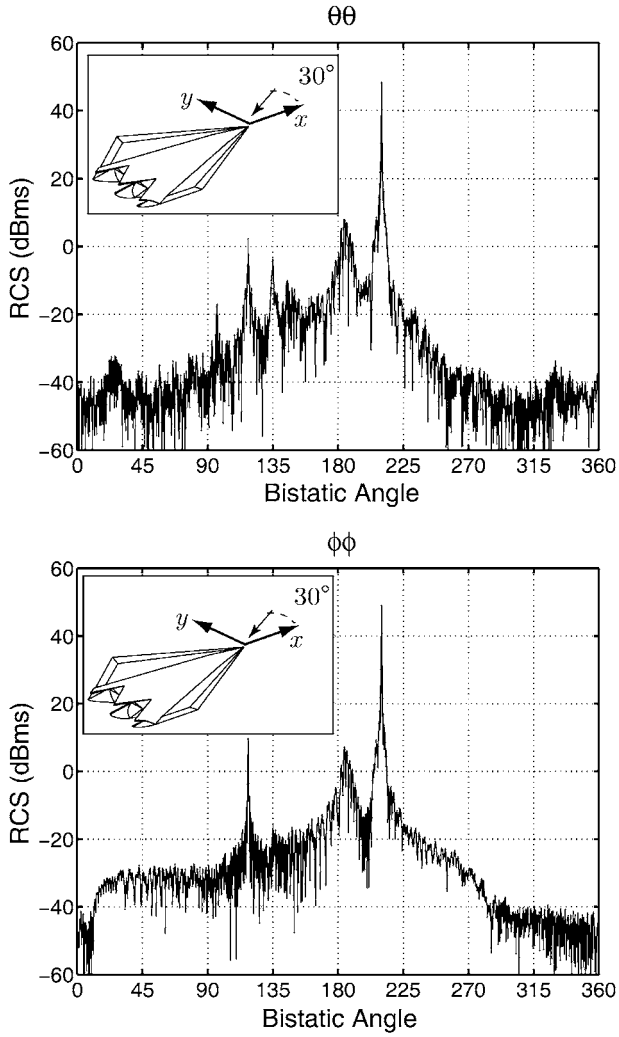


Fig. 11. Bistatic RCS (in dBm^2) of the stealth airborne target Flamme at 16 GHz. Maximum dimension of the Flamme is 6 m corresponding to 320λ. The target is illuminated by a plane wave propagating in the x-y plane at a 30° angle from the x axis, as also depicted in the inset.

The copolar RCS values are plotted on the x-y plane as a function of the bistatic angle ϕ. In the plots, 30° and 210° correspond to the back-scattering and forward-scattering directions,

respectively. Solution of this problem is performed by a 10-level MLFMA (6 distributed and 4 shared levels) parallelized into 16 processes. After the setup, which takes about 104 min, the problem is solved twice (for two polarizations) in about 490 min. Using BiCGStab and BDP, the numbers of iterations to reduce the residual error below 10^{-3} are 42 and 35, respectively, for the θ and the ϕ polarizations of the plane-wave excitation. Both near-field and far-field interactions are calculated with 1% error and the total memory usage is 139 GB using the single-precision representation.

V. Concluding Remarks

In this paper, we consider fast and accurate solutions of large-scale scattering problems discretized with tens of millions of unknowns using a parallel MLFMA implementation. We investigate the parallelization of MLFMA and improve the efficiency of the implementations. Some of the major steps for the efficient parallelization of MLFMA are as follows.

- Distribute the near-field interactions equally among the processors using a load-balancing algorithm.
- Determine the shared and distributed levels appropriately by choosing an optimal LoD.
- Distribute the clusters in LoD among the processors by considering the combined load of all descendants of each cluster.
- Assign each cluster and its subclusters to the same processor for the levels below LoD.
- Distribute the samples of the fields among the processors using a load-balancing algorithm (to reduce one-to-one communications) for the levels above LoD.
- Use a communication map to pair the processors for the translations in the distributed levels. Transfer the required data using nonblocking send and receive operations.

We demonstrate the accuracy of our implementations by considering a canonical scattering problem involving a sphere of radius 110λ discretized with 41 883 638 unknowns. To the best of our knowledge, this is the largest integral-equation problem ever solved.[2] In addition to the solution of various extremely large canonical problems, we also demonstrate the effectiveness of our implementation on a real-life problem involving the Flamme geometry with a size larger than 300λ.

References

[1] A. J. Poggio and E. K. Miller, "Integral equation solutions of three-dimensional scattering problems," in *Computer Techniques for Electromagnetics*, R. Mittra, Ed. Oxford, U.K.: Pergamon, 1973, ch. 4.

[2] J. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[3] S. Velamparambil, J. E. Schutt-Aine, J. G. Nickel, J. M. Song, and W. C. Chew, "Solving large scale electromagnetic problems using a linux cluster and parallel MLFMA," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, 1999, vol. 1, pp. 636–639.

[4] S. Velamparambil, J. M. Song, and W. C. Chew, "A portable parallel multilevel fast multipole solver for scattering from perfectly conducting bodies," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, 1999, vol. 1, pp. 648–651.

[2]Prior to the publication of this paper, the authors reported in January 2008 the solution of a larger scattering problem involving a sphere of radius 150λ discretized with 85 148 160 unknowns.

[5] K. C. Donupedi, J.-M. Jin, S. Velamparambil, J. Song, and W. C. Chew, "A higher order parallelized multilevel fast multipole algorithm for 3-D scattering," *IEEE Trans. Antennas Propag.*, vol. 49, no. 7, pp. 1069–1078, Jul. 2001.

[6] S. Velamparambil, W. C. Chew, and M. L. Hastriter, "Scalable electromagnetic scattering computations," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, 2002, vol. 3, pp. 176–179.

[7] S. Velamparambil, W. C. Chew, and J. Song, "10 million unknowns: Is it that big?," *IEEE Antennas Propag. Mag.*, vol. 45, no. 2, pp. 43–58, Apr. 2003.

[8] M. L. Hastriter, "A study of MLFMA for large-scale scattering problems," Ph.D. dissertation, Univ. Illinois at Urbana-Champaign, , 2003.

[9] G. Sylvand, "Performance of a parallel implementation of the FMM for electromagnetics applications," *Int. J. Numer. Meth. Fluids*, vol. 43, pp. 865–879, 2003.

[10] J. Dong, S. L. Chai, and J. J. Mao, "An automatic load-balanced parallel multilevel fast multipole method for large scale electromagnetic scattering problem," in *Proc. Asia-Pacific Conf.*, 2005.

[11] S. Velamparambil and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2719–2727, Aug. 2005.

[12] Ö. Ergül and L. Gürel, "Efficient parallelization of multilevel fast multipole algorithm," in *Proc. Europ. Conf. Antennas Propag. (EuCAP)*, 2006, no. 350094.

[13] L. Gürel and Ö. Ergül, "Fast and accurate solutions of integral-equation formulations discretised with tens of millions of unknowns," *Electron. Lett.*, vol. 43, no. 9, pp. 499–500, Apr. 2007.

[14] D. R. Wilton and J. E. Wheeler, III, "Comparison of convergence rates of the conjugate gradient method applied to various integral equation formulations," in *Progress in Electromagn. Res. PIER 05*, 1991, pp. 131–158.

[15] L. Gürel and Ö. Ergül, "Comparisons of FMM implementations employing different formulations and iterative solvers," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, 2003, vol. 1, pp. 19–22.

[16] L. Gürel and Ö. Ergül, "Extending the applicability of the combined-field integral equation to geometries containing open surfaces," *IEEE Antennas Wireless Propag. Lett.*, vol. 5, pp. 515–516, 2006.

[17] Ö. Ergül and L. Gürel, "Linear-linear basis functions for MLFMA solutions of magnetic-field and combined-field integral equations," *IEEE Trans. Antennas Propag.*, vol. 55, no. 4, pp. 1103–1110, Apr. 2007.

[18] J. R. Mautz and R. F. Harrington, "H-field, E-field, and combined field solutions for conducting bodies of revolution," *AEÜ*, vol. 32, no. 4, pp. 157–164, Apr. 1978.

[19] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. AP-30, no. 3, pp. 409–418, May 1982.

[20] Ö. Ergül and L. Gürel, "The use of curl-conforming basis functions for the magnetic-field integral equation," *IEEE Trans. Antennas Propag.*, vol. 54, no. 7, pp. 1917–1926, Jul. 2006.

[21] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propag. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993.

[22] S. Koc, J. M. Song, and W. C. Chew, "Error analysis for the numerical evaluation of the diagonal forms of the scalar spherical addition theorem," *SIAM J. Numer. Anal.*, vol. 36, no. 3, pp. 906–921, 1999.

[23] Ö. Ergül and L. Gürel, "Enhancing the accuracy of the interpolations and anterpolations in MLFMA," *IEEE Antennas Wireless Propag. Lett.*, vol. 5, pp. 467–470, 2006.

[24] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernels," *Comp. Phys. Comm.*, vol. 65, pp. 24–38, Apr. 1991.

[25] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston, MA: Artech House, 2001.

[26] J. Song and W. C. Chew, "Interpolation of translation matrix in MLFMA," *Microw. Opt. Technol. Lett.*, vol. 30, no. 2, pp. 109–114, Jul. 2001.

[27] Ö. Ergül and L. Gürel, "Optimal interpolation of translation operator in multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 54, no. 12, pp. 3822–3826, Dec. 2006.

[28] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc Users Manual*. Argonne, IL: Argonne National Lab., 2004.

[29] Ö. Ergül and L. Gürel, "Investigation of the inaccuracy of the MFIE discretized with the RWG basis functions," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, 2004, vol. 3, pp. 3393–3396.

[30] L. Gürel, H. Bağcı, J. C. Castelli, A. Cheraly, and F. Tardivel, "Validation through comparison: Measurement and calculation of the bistatic radar cross section (BRCS) of a stealth target," *Radio Sci.*, vol. 38, no. 3, Jun. 2003.

Özgür Ergül (S'98) was born in Yozgat, Turkey, in 1978. He received the B.Sc. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2001 and 2003, respectively, where he is currently working toward the Ph.D. degree.

Since 2001, he has served as a Teaching and Research Assistant in the Department of Electrical and Electronics Engineering, Bilkent University. He has been with the Computational Electromagnetics Group at Bilkent University from 2000 to 2005 and with the Computational Electromagnetics Research Center (BiLCEM) since 2005. His research interests include fast and accurate algorithms for the solutions of large and complicated structures, integral equations, parallel programming, and iterative techniques.

Mr. Ergül was a recipient of the 2007 IEEE Antennas and Propagation Society Graduate Fellowship and the 2007 Leopold B. Felsen Award for Excellence in Electrodynamics. He is the Secretary of Commission E (Electromagnetic Noise and Interference) of URSI Turkey National Committee. His academic endeavors are supported by the Scientific and Technical Research Council of Turkey (TUBITAK) through a Ph.D. scholarship.

Levent Gürel (S'87–M'92–SM'97) received the B.Sc. degree from the Middle East Technical University (METU), Ankara, Turkey, in 1986, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign (UIUC) in 1988 and 1991, respectively, all in electrical engineering.

He joined the Thomas J. Watson Research Center of the International Business Machines Corporation, Yorktown Heights, NY, in 1991, where he worked as a Research Staff Member on the electromagnetic compatibility (EMC) problems related to electronic packaging, on the use of microwave processes in the manufacturing and testing of electronic circuits, and on the development of fast solvers for interconnect modeling. Since 1994, he has been a faculty member with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, where he is currently a Professor. He was a Visiting Associate Professor with the Center for Computational Electromagnetics (CCEM) of the UIUC for one semester in 1997. He returned to the UIUC as a Visiting Professor during 2003–2005, and as an Adjunct Professor after 2005. He founded the Computational Electromagnetics Research Center (BiLCEM) at Bilkent University in 2005, where he is serving as the Director. His research interests include the development of fast algorithms for computational electromagnetics (CEM) and the application thereof to scattering and radiation problems involving large and complicated scatterers, antennas and radars, frequency-selective surfaces, high-speed electronic circuits, optical and imaging systems, nanostructures, and metamaterials. He is also interested in the theoretical and computational aspects of electromagnetic compatibility and interference analyses. Ground-penetrating radars and other subsurface scattering applications are also among his research interests. Since 2006, his research group has been breaking several world records by solving extremely large integral-equation problems, the largest involving as many as 85 million unknowns.

Prof. Gürel's accomplishments include two prestigious awards from the Turkish Academy of Sciences (TUBA) in 2002 and the Scientific and Technical Research Council of Turkey (TUBITAK) in 2003. He served as the Chairman of the AP/MTT/ED/EMC Chapter of the IEEE Turkey Section in 2000–2003. He founded the IEEE EMC Chapter in Turkey in 2000. He served as the Co-chairman of the 2003 IEEE International Symposium on Electromagnetic Compatibility. He is a member of the General Assembly of the European Microwave Association, a member of the USNC of the International Union of Radio Science (URSI), and the Chairman of Commission E (Electromagnetic Noise and Interference) of URSI Turkey National Committee. He organized and served as the General Chair of the CEM'07 Computational Electromagnetics International Workshop in 2007. He is currently serving as Associate Editor of the IEEE ANTENNAS AND WIRELESS PROPAGATION LETTERS and *Radio Science* .

A Hierarchical Partitioning Strategy for an Efficient Parallelization of the Multilevel Fast Multipole Algorithm

Özgür Ergül, *Student Member, IEEE,* and Levent Gürel, *Fellow, IEEE*

***Abstract*—We present a novel hierarchical partitioning strategy for the efficient parallelization of the multilevel fast multipole algorithm (MLFMA) on distributed-memory architectures to solve large-scale problems in electromagnetics. Unlike previous parallelization techniques, the tree structure of MLFMA is distributed among processors by partitioning both clusters and samples of fields at each level. Due to the improved load-balancing, the hierarchical strategy offers a higher parallelization efficiency than previous approaches, especially when the number of processors is large. We demonstrate the improved efficiency on scattering problems discretized with millions of unknowns. In addition, we present the effectiveness of our algorithm by solving very large scattering problems involving a conducting sphere of radius 210 wavelengths and a complicated real-life target with a maximum dimension of 880 wavelengths. Both of the objects are discretized with more than 200 million unknowns.**

***Index Terms*—Large-scale problems, multilevel fast multipole algorithm, parallelization, scattering problems, surface integral equations.**

I. Introduction

SURFACE integral equations are commonly used to formulate scattering and radiation problems involving three-dimensional conducting bodies with arbitrary shapes [1]. The application of boundary conditions for the electric field and the magnetic field on the surface of an object leads to the electric-field integral equation (EFIE) and the magnetic-field integral equation (MFIE), respectively. For closed surfaces, EFIE and MFIE can be combined to obtain the combined-field integral equation (CFIE), which is free of the internal-resonance problem [2]. Numerical solutions of integral equations require the discretization (e.g., triangulation) of surfaces. Then, unknown surface currents are expanded in a series of basis functions, and integral equations are tested by employing a set of testing functions. Finally, solutions of resulting $N \times N$ dense matrix equations provide the expansion coefficients, which can be used to compute the scattered or radiated electric and magnetic fields everywhere.

Surface integral equations provide accurate results when they are discretized appropriately by using small elements with respect to wavelength. Therefore, when a problem involves a large object with dimensions of several wavelengths, its accurate discretization leads to a large matrix equation with hundreds of thousands of unknowns. Such a large problem can be solved iteratively, where the required matrix-vector multiplications (MVMs) are performed efficiently by the multilevel fast multipole algorithm (MLFMA) [3]. For an $N \times N$ dense matrix equation, MLFMA reduces the complexity of MVMs from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$, allowing for the solution of large problems with limited computational resources. On the other hand, accurate solutions of many real-life problems require discretizations with millions of elements, leading to matrix equations with millions of unknowns, which cannot easily be solved with sequential implementations of MLFMA running on a single processor. To solve such large problems, it is helpful to increase computational resources by assembling parallel computing platforms and, at the same time, by parallelizing MLFMA.

The parallelization of MLFMA is not trivial because of the complicated structure of this algorithm. Simple parallelization techniques usually fail to provide efficient solutions, due to communications among processors, poor load-balancing of the workload, and unavoidable duplications of computations over multiple processors. Advanced parallelization techniques have been developed to improve the parallelization of MLFMA by using novel partitioning strategies, load-balancing algorithms, and optimizations for communications [4]–[11]. This way, it has become possible to solve problems with tens of millions of unknowns on relatively inexpensive computing platforms with distributed-memory architectures [4]–[6], [9], [10].

Recently, we developed a hierarchical partitioning strategy that is well suited for the multilevel structure of MLFMA [12]. With the enhanced load-balancing offered by the hierarchical strategy, parallelization of MLFMA can be improved significantly. In this paper, we provide the details of our parallelization algorithm. We employ canonical problems involving sphere geometries of various sizes for the comparison of the hierarchical strategy with previous approaches. We show that the efficiency of the parallelization is improved drastically, especially when the number of processors is large. Improved efficiency provided

Manuscript received June 23, 2008; revised October 16, 2008. Current version published June 03, 2009. This work was supported in part by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grants 105E172 and 107E136, in part by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), and in part by contracts from ASELSAN and SSM.

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, TR-06800 Bilkent, Ankara, Turkey and also with the Computational Electromagnetics Research Center (BiLCEM), Bilkent University, TR-06800 Bilkent, Ankara, Turkey (e-mail: ergul@ee.bilkent.edu.tr; lgurel@bilkent.edu.tr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAP.2009.2019913

by the hierarchical strategy is also demonstrated on scattering problems discretized with more than 100 million unknowns. Finally, we present the solutions of very large scattering problems involving a sphere of radius 210λ and a stealth airborne target with a maximum dimension of 880λ, which are discretized with 204,823,296 and 204,664,320 unknowns, respectively, and λ denotes the wavelength.

The rest of the paper is organized as follows. In Section II, we summarize an efficient implementation of MLFMA, focusing on the main stages of the algorithm. Section III presents the parallelization of MLFMA using the hierarchical partitioning strategy. We investigate the communications among processors in Section IV and compare our parallelization technique with the previous approaches in Section V. Finally, numerical results are presented in Section VI, followed by our concluding remarks in Section VII.

II. Multilevel Fast Multipole Algorithm

For perfectly-conducting objects, discretizations of surface integral equations lead to $N \times N$ dense matrix equations in the form of

$$\sum_{n=1}^{N} Z_{mn} a_n = v_m, \quad m = 1, 2, \ldots, N \tag{1}$$

where the matrix elements Z_{mn} for $m, n = 1, 2, \ldots, N$ can be interpreted as electromagnetic interactions of discretization elements, i.e., basis and testing functions. The matrix equation (1) can be solved iteratively via a Krylov subspace algorithm, where the required MVMs are performed efficiently by MLFMA [3]. In general, MLFMA splits MVMs as

$$\bar{\boldsymbol{Z}} \cdot \boldsymbol{x} = \bar{\boldsymbol{Z}}_{NF} \cdot \boldsymbol{x} + \bar{\boldsymbol{Z}}_{FF} \cdot \boldsymbol{x} \tag{2}$$

where near-field interactions denoted by $\bar{\boldsymbol{Z}}_{NF}$ are calculated directly and stored in memory to perform the partial multiplications $\bar{\boldsymbol{Z}}_{NF} \cdot \boldsymbol{x}$, while multiplications involving far-field interactions, i.e., $\bar{\boldsymbol{Z}}_{FF} \cdot \boldsymbol{x}$, are performed approximately and efficiently. In this section, we briefly describe an efficient implementation of MLFMA by summarizing the main stages of the algorithm.

A. Discretization of the Object

Without losing generality, we consider a smooth object with an electrical dimension of kD, where $k = 2\pi/\lambda$ is the wavenumber. Discretization (triangulation) of the object with $\lambda/10$ mesh size leads to N unknowns, where $N = \mathcal{O}(k^2 D^2)$. As basis and testing functions, we use Rao-Wilton-Glisson (RWG) [13] functions defined on planar triangles.

B. Clustering

To calculate electromagnetic interactions in a multilevel scheme, a tree structure is constructed by placing the object in a cubic box and recursively dividing the computational domain into subdomains, until the box size is about 0.25λ. A multilevel tree structure with $(L+2) = \mathcal{O}(\log(kD)) = \mathcal{O}(\log N)$ levels is obtained by considering nonempty boxes (clusters)[1]. At level l from 1 to L, the number of clusters can be approximated as

$$N_l \approx 4^{(1-l)} N_1 \tag{3}$$

where $N_1 = \mathcal{O}(N)$. In other words, the number of clusters decreases approximately by a factor of four from a level to the next upper level.

The tree structure in MLFMA can be constructed by using a top-down or a bottom-up strategy [10]. In the top-down strategy, the size of the largest cube enclosing the object is minimized, while the size of the smallest boxes at the lowest level depends on the size of the object and the number of levels. In the bottom-up strategy, however, the size of the smallest boxes is fixed to some value (such as 0.25λ), and the sizes of the boxes at higher levels are recursively doubled until the whole object is enclosed by the largest box. For a given problem, one of the two strategies can be preferable in terms of efficiency and accuracy.

C. Sampling

For each cluster in the tree structure, radiated and incoming fields are defined and sampled on the unit sphere. We choose samples regularly spaced in the ϕ direction and use the Gauss-Legendre quadrature in the θ direction [14]. For level $l = 1, 2, \ldots, L$, the number of samples is $S_l^\theta = (T_l + 1)$ and $S_l^\phi = 2(T_l + 1)$ along θ and ϕ directions, respectively, where T_l is the truncation number determined by the excess bandwidth formula [15], i.e.,

$$T_l \approx 1.73 k a_l + 2.16 (d_0)^{2/3} (k a_l)^{1/3}. \tag{4}$$

In (4), a_l is the box size at level l, and d_0 is the desired digits of accuracy. The sampling rate depends on the cluster size as measured by the wavelength ($k a_l = 2\pi a_l/\lambda$), and the total number of samples can be approximated as

$$S_l = S_l^\theta S_l^\phi \approx 2^{(l-1)} S_1^\theta \times 2^{(l-1)} S_1^\phi = 4^{(l-1)} S_1 \tag{5}$$

where $S_1 = \mathcal{O}(1)$.

D. Far-Field Interactions

In MLFMA, far-field interactions are calculated in a cluster-by-cluster manner using the diagonalization and factorization of the homogenous-space Green's function [14]. In each MVM, three main stages, i.e., aggregation, translation, and disaggregation, are performed as described below.

1) Aggregation: In this stage, radiated fields of clusters are calculated from the bottom of the tree structure to the highest level ($l = L$). At the lowest level, radiation patterns of basis functions, which are calculated during the setup of MLFMA, are multiplied with the coefficients provided by the iterative solver

[1]In this paper, L represents the number of effective levels, where MLFMA stages, i.e., aggregation, translation, and disaggregation, are performed. The actual number of levels is $(L+2)$, but the highest two levels are not used directly in MLFMA.

and combined to obtain the radiated fields of the smallest clusters. Then, the radiated fields of clusters at higher levels are obtained by shifting and combining the radiated fields of clusters at lower levels. During the aggregation stage, we use a local Lagrange interpolation between successive levels to match different sampling rates for fields.

2) Translation: In this stage, radiated fields computed during the aggregation stage are translated into incoming fields. For each cluster at any level, there are $\mathcal{O}(1)$ clusters to translate the radiated field to. In addition, using the symmetry of cubic (identical) clusters, the number of different translation operators is $\mathcal{O}(1)$, independent of the level [4]. Translation operators are calculated during the setup of MLFMA in $\mathcal{O}(N)$ processing time using local interpolation methods [16].

3) Disaggregation: This stage involves the calculation of total incoming fields at cluster centers from the top of the tree structure to the lowest level. At the highest level, the total incoming field for a cluster is obtained by the combination of incoming fields due to translations. At lower levels, however, the incoming field to the center of a cluster involves a contribution from the incoming field to the center of its parent cluster. We use transpose interpolation (anterpolation) between consecutive levels during the disaggregation stage to match different sampling rates of the levels [17]. Following the disaggregation operations at the lowest level, incoming fields are received by the testing functions. Similar to the radiation patterns of basis functions, receiving patterns of testing functions are also calculated during the setup of MLFMA.

Considering the three stages of MLFMA, the processing time and memory required for all operations at level l is proportional to the product of the number of clusters and the number of samples, i.e.,

$$N_l S_l \approx 4^{(1-l)} N_1 4^{(l-1)} S_1 = N_1 S_1 = \mathcal{O}(N). \qquad (6)$$

We note that all levels of MLFMA have equal importance with $\mathcal{O}(N)$ complexity in terms of processing time and memory.

E. Near-Field Interactions

In MLFMA, there are also $\mathcal{O}(N^2/N_1) = \mathcal{O}(N)$ near-field interactions, which are calculated directly in the setup stage of the program and stored in memory to be used multiple times during the iterations. These interactions are between the basis and testing functions that are located close to each other. We use singularity extraction techniques [18]–[21] and Gaussian quadratures [22] in order to calculate the near-field interactions accurately and efficiently.

III. Hierarchical Parallelization of MLFMA

The main task in the parallelization of MLFMA on distributed-memory architectures is partitioning the multilevel tree structure among processors. Simple parallelization techniques, based on distributing clusters among processors, usually fail to provide efficient solutions. This is mainly due to dense communications between processors, duplication of computations, and unbalanced distribution of the workload among processors [7], [8]. Since such problems arise mostly at the higher levels of MLFMA, a hybrid parallelization technique, which applies different partitioning strategies for the lower and the higher levels, is developed to improve the efficiency [7]–[10]. In this technique, processor assignments are made on the basis of fields of clusters at the higher levels. In other words, each cluster at higher levels is shared by all processors, while each processor is assigned to the same portion of fields for all clusters. Even though the hybrid parallelization technique increases the parallelization efficiency significantly, compared to simple parallelization approaches, the improvement can be insufficient, especially when the number of processors is large.

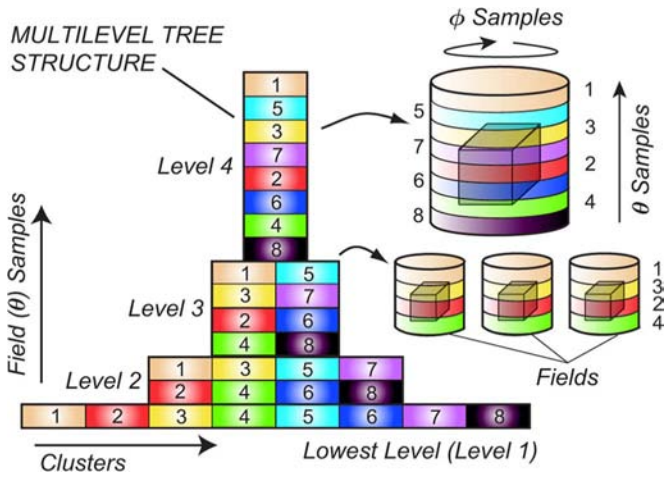


Fig. 1. Distribution of a four-level tree structure among eight processors using the hierarchical partitioning strategy.

In this section, we provide the details of the hierarchical parallelization of MLFMA for the efficient solution of large-scale problems. The hierarchical parallelization is based on the simultaneous partitioning of clusters and their fields at all levels. We adjust the partitioning in both directions (clusters and samples of fields) appropriately by considering the number of clusters and the number of samples at each level. As an example, Fig. 1 depicts a four-level tree structure ($L = 4$), where levels are represented by two-dimensional rectangles. Horizontal and vertical dimensions of rectangles correspond to clusters and θ samples of fields, respectively. The tree structure is partitioned among eight processors labeled 1 to 8. At the lowest level, clusters are distributed among eight processors, and each cluster is assigned to a single processor, without any partitioning of field samples. Then, at the next level ($l = 2$), field samples are partitioned among two groups of processors, i.e., (1,3,5,7) and (2,4,6,8), while the number of cluster partitions is reduced to four. At this level, samples of each cluster are shared by two processors. As we proceed to the higher levels, the number of partitions for clusters and samples of fields are systematically decreased and increased, respectively.

In the following subsections, we present the hierarchical parallelization of MLFMA in detail by considering the main stages of the algorithm.

A. Partitioning of the Tree Structure

We consider the parallelization of MLFMA on a cluster of p processors, where $p = 2^i$ for some integer i. Using the hierar-

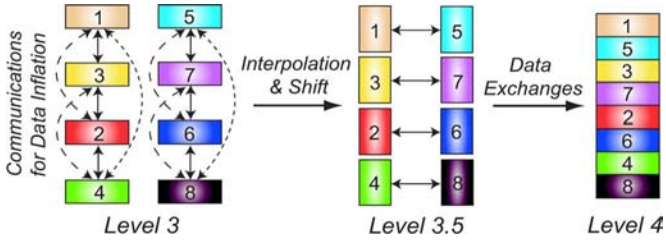


Fig. 2. Aggregation operations from level 3 to level 4 for the partitioned tree structure in Fig. 1.

chical partitioning strategy, the number of partitions for clusters at level l is chosen as

$$p_{l,c} = \max\left\{\frac{p}{2^{(l-1)}}, 1\right\} = \max\left\{p\,2^{(1-l)}, 1\right\}. \quad (7)$$

We note that clusters are not partitioned for levels $l > \log_2(p)$, if such a level exists. The number of clusters assigned to each processor can be approximated as

$$N_l^p \approx \frac{N_l}{p_{l,c}} \approx \left\{\begin{array}{ll} \frac{2^{(1-l)}N_1}{p}, & l \le \log_2(p) \\ 4^{(1-l)}N_1, & l > \log_2(p) \end{array}\right\}. \quad (8)$$

In addition, samples of the fields are divided into

$$p_{l,s} = \frac{p}{p_{l,c}} = \min\{2^{(l-1)}, p\} \quad (9)$$

partitions along the θ direction for level l. Field samples are partitioned only along the θ direction for an easy implementation of interpolation/anterpolation operations [7]. The number of θ samples assigned to each processor is

$$S_l^{\theta,p} \approx \frac{S_l^\theta}{p_{l,s}} \approx \left\{\begin{array}{ll} S_1^\theta, & l \le \log_2(p) \\ \frac{2^{(l-1)}S_1^\theta}{p}, & l > \log_2(p) \end{array}\right\}. \quad (10)$$

Also considering the sampling in the ϕ direction, the total number of samples per processor can be written as

$$S_l^p = S_l^{\theta,p} S_l^\phi \approx \left\{\begin{array}{ll} 2^{(l-1)}S_1, & l \le \log_2(p) \\ \frac{4^{(l-1)}S_1}{p}, & l > \log_2(p) \end{array}\right\}. \quad (11)$$

Finally, the size of the local data at each processor is

$$N_l^p S_l^p \approx \frac{N_1 S_1}{p} \quad (12)$$

for $l = 1, 2, \ldots, L$.

B. Aggregation Stage

For the partitioned tree structure in Fig. 1, aggregation operations from level 3 to level 4 are depicted in Fig. 2 and can be listed as follows.

1) One-to-One Communications for Data Inflation: During aggregations from a level to the next higher level, interpolations are required to increase the sampling rate for radiated fields. Using local Lagrange interpolation, each target point in the fine grid has contributions from a set of neighboring points in the coarse grid. Therefore, when samples of fields are partitioned among processors, interpolations in each processor need samples located in other processors. Consequently, before interpolations, one-to-one communications are required between pairs of processors to inflate the local data, in accordance with the interpolation requirements.

For the partitioned tree structure in Fig. 1, aggregation from level 3 to level 4 requires one-to-one communications within two separate groups of processors that are located in the same columns, i.e., (1,2,3,4) and (5,6,7,8), as depicted in Fig. 2. As an important advantage of the hierarchical partitioning strategy, distribution of the θ samples into large numbers of partitions is avoided. Therefore, communications are required mostly between pairs of processors located "next to each other." For example, processor 3 communicates mainly with processors 1 and 2, but not with processor 4. Processors 3 and 4 would need to communicate with each other if the number of the θ samples required for the interpolation is larger than the number of the θ samples per processor. However, using the hierarchical strategy, the number of partitions along the θ direction, hence the number of the θ samples per processor, can be adjusted such that those secondary communications between "distant" processors are avoided.

2) Interpolation and Shifting: When the required data is prepared by one-to-one communications for a cluster, its radiated field is interpolated and shifted to the center of its parent cluster. Temporary levels involving parent clusters and field samples after the interpolation and shifting operations are denoted as intermediate levels. As an example, for the partitioned tree structure in Fig. 1, level 3.5 is depicted in Fig. 2. Following the interpolation, the number of samples along the θ direction assigned to each processor is doubled, i.e.,

$$S_{l+1/2}^{\theta,p} \approx 2S_l^{\theta,p} \approx \left\{\begin{array}{ll} 2S_1^\theta, & l \le \log_2(p) \\ \frac{2^l S_1^\theta}{p}, & l > \log_2(p) \end{array}\right\}. \quad (13)$$

At the same time, the number of clusters in each processor can be written as

$$N_{l+1/2}^p \approx \frac{N_l^p}{4} \approx \left\{\begin{array}{ll} \frac{2^{-(1+l)}N_1}{p}, & l \le \log_2(p) \\ 4^{(-l)}N_1, & l > \log_2(p) \end{array}\right\}. \quad (14)$$

Intermediate levels are defined temporarily and used to arrange the data in each processor, before communications are performed to modify the partitioning according to the hierarchical strategy.

3) Data Exchanges: From an intermediate level $(l + 1/2)$ to the next level $(l + 1)$, data is exchanged among processors, if $l \le \log_2(p)$. As depicted in Fig. 2, processors are paired according to their positions in the partitioning scheme. Each processor performs the following communications.

- Send half of the field samples of each cluster at the intermediate level;
- Receive the complementary data, which involves field samples of some clusters, from the associated processor.

With data exchanges, the number of clusters in each processor is doubled with respect to the number of clusters at the intermediate level, while the number of samples along the θ direction is halved. Then, we have

$$N_{l+1}^p \approx \left\{\begin{array}{ll} \frac{2^{(-l)}N_1}{p}, & (l+1) \le \log_2(p) \\ 4^{(-l)}N_1, & (l+1) > \log_2(p) \end{array}\right\} \quad (15)$$

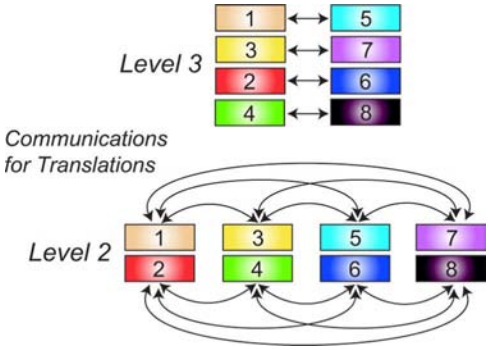


Fig. 3. One-to-one communications during the translation stage at levels 2 and 3 of the partitioned tree structure in Fig. 1.

and

$$S_{l+1}^{\theta,p} \approx \left\{ \begin{array}{ll} S_1^{\theta}, & (l+1) \leq \log_2(p) \\ \frac{2^l S_1^{\theta}}{p}, & (l+1) > \log_2(p) \end{array} \right\} \tag{16}$$

which agree with the expressions in (8) and (10), respectively.

C. Translation Stage

Using the hierarchical partitioning strategy, one-to-one communications are also required during the translation stage, since clusters are partitioned, and some translations are needed among clusters located in different processors. These communications are achieved by pairing the processors and transferring the radiated fields of clusters between the pairs. As depicted in Fig. 3, communications are required only among the processors that are located in the same row of the partitioning. For example, communications at level 2 are performed within two separate groups of processors, i.e., (1,3,5,7) and (2,4,6,8). In general, for "inter-processor" translations at level l, each processor is paired one by one with the other $(p_{l,c}-1)$ processors. Once a pairing is established, radiated fields of clusters are transferred, and translations are performed by the receiver processor.

In addition to inter-processor translations, there are also "intra-processor" translations that are related to clusters located in the same processor. These translations can be performed independently in each processor, without any communication.

D. Disaggregation Stage

The parallelization of the disaggregation stage is very similar to the parallelization of the aggregation stage. In general, operations in the aggregation stage are performed in a reverse manner.

1) Data Exchanges: When incoming fields are calculated at cluster centers at level l, partitioning is modified via data exchanges among the processors. This way, the partitioning at level $(l-1/2)$ is generated as required for anterpolation and shifting operations.

2) Anterpolation and Shifting: Incoming fields at the centers of clusters are anterpolated and shifted to the centers of their subclusters at level $(l-1)$.

3) One-to-One Communications for Data Deflation: Since the anterpolation is the transpose of the interpolation, some of the samples obtained from an anterpolation operation should be sent to other processors. This is because interpolations during the aggregation stage are performed using the inflated data prepared by one-to-one communications among processors. As the reverse of this process, anterpolations produce inflated data, which must be deflated via one-to-one communications. Following an anterpolation operation, some of the resulting data are used locally, while the rest are sent to other processors. Similar to the communications during interpolations, data are transferred mostly among neighboring processors in the same column of the partitioning scheme.

IV. Communications in the Hierarchical Parallelization of MLFMA

Using the hierarchical partitioning strategy, computations on the tree structure are distributed among processors with improved load-balancing, compared to previous strategies based on partitioning with respect to only clusters or only samples of fields. However, there are still unavoidable communications among processors, which may reduce the efficiency of the parallelization. In this section, we investigate these communications in detail.

A. Communications in the Aggregation/Disaggregation Stages

During an interpolation operation in a processor at level $l = 2,3,\ldots,(L-1)$, the amount of data required from other processors for each cluster is proportional to the number of samples in the ϕ direction. Considering also the number of clusters per processor, the communication time for interpolations at level l can be written as

$$\begin{aligned} t_{\mathrm{int},l} &\propto 2^{(l-1)} S_1^{\phi} \left\{ \begin{array}{ll} \frac{2^{(1-l)} N_1}{p}, & l \leq \log_2(p) \\ 4^{(1-l)} N_1, & l > \log_2(p) \end{array} \right\} \\ &= \left\{ \begin{array}{ll} \frac{N_1 S_1^{\phi}}{p}, & l \leq \log_2(p) \\ \frac{N_1 S_1^{\phi}}{2^{(l-1)}}, & l > \log_2(p) \end{array} \right\} \\ &\leq \frac{N_1 S_1^{\phi}}{p}, \quad (l = 2,3,\ldots,L-1). \end{aligned} \tag{17}$$

We note that the communication time $t_{\mathrm{int},l}$ tends to decrease with the increasing number of processors p.

To switch the partitioning scheme from level to level, each processor exchanges half of its data produced during the aggregation stage. The processing time for these communications can be expressed as

$$\begin{aligned} t_{\mathrm{exc},l} &\propto \left\{ \begin{array}{ll} \frac{N_1 S_1}{p}, & l \leq \log_2(p) \\ 0, & l > \log_2(p) \end{array} \right\} \\ &\leq \frac{N_1 S_1}{p}, \quad (l = 1,2,\ldots,L-1) \end{aligned} \tag{18}$$

where the upper bound is again inversely proportional to the number of processors p. The processing time required for communications during the disaggregation stage is the same as the time required for communications during the aggregation stage.

B. Communications in the Translation Stage

During the translation stage, each processor is paired one by one with $(p_{l,c}-1) = \max\{p2^{(1-l)}-1, 0\}$ processors to perform inter-processor translations. For each pair, the number of cluster-cluster interactions required to be performed is proportional to the number of clusters per processor. In addition, the

size of the data transferred in each interaction is proportional to the number of local samples per cluster, i.e., S_l^p. Therefore, the communication time for translations can be written as

$$t_{\text{trans},l} \propto \max\left\{p2^{(1-l)}-1,0\right\} N_l^p S_l^p \leq \frac{N_1 S_1}{2^{(l-1)}}, \quad (l=1,2,\ldots,L). \tag{19}$$

The communication time for translations can be significant, especially at the lower levels of MLFMA.

V. Comparisons With Previous Parallelization Techniques

In this paper, we compare the hierarchical parallelization technique with two previous approaches, namely, the simple and the hybrid parallelization techniques. As mentioned in Section III, the simple parallelization of MLFMA is based on the distribution of clusters among processors at all levels. A major disadvantage of this technique is the difficulty in distributing a small numbers of clusters at the higher levels of the tree structure [8]. When the number of processors is large, those clusters must be duplicated over multiple processors. Otherwise, a large amount of data is required to be communicated during the aggregation and disaggregation stages. In addition, when using the simple parallelization technique, translations involve dense communications among processors [7], [8].

The hybrid parallelization technique was successfully developed to improve the parallelization of MLFMA [7]. In this technique, the lower (distributed) levels of MLFMA are partitioned as in the simple technique, i.e., clusters are distributed among processors. In the higher (shared) levels, however, samples of fields are distributed, instead of clusters. Unlike the hierarchical parallelization, samples in a shared level are distributed among all processors, without any partitioning of clusters. Distributing samples provides improved load-balancing and communication-free translations for the higher levels of the tree structure. On the other hand, problems arise for some levels at the middle of the tree, where it is not efficient to distribute either fields or clusters among processors [10].

The hierarchical parallelization technique provides two important advantages, compared to simple and hybrid techniques.

- Improved load-balancing: Partitioning both clusters and samples of fields leads to an improved load-balancing of the workload among processors at each level. Consequently, duplication of computations, which may occur in the simple parallelization, and waits for the synchronization of processors are minimized.
- Reduced communications: Although the hierarchical partitioning increases the types of communications, compared to simple and hybrid approaches, the amount of data transferred is decreased. In addition, due to the improved load-balancing, the average package size is enlarged, and the number of communication events is reduced. As a result, the communication time is significantly shortened.

Finally, another important advantage of the hierarchical parallelization algorithm appears when MLFMA is employed on a cluster with multiprocessor nodes. Most of the mainboards built recently have multiple processors connected via high-speed buses. Then, parallel computers are constructed by clustering a number of multiprocessor computing units (nodes), instead of processors. Resulting parallel computers are highly nonuniform, since communications among processors in the same node are significantly faster than those among processors located in different nodes. Using multicore processors further complicates the situation, since communications within nodes also have diverse rates, depending on whether the inter-core communications are taking place in the same processor or between two processors in the same node. The hierarchical parallelization technique is suitable for such parallel platforms. As an example, let the tree structure in Fig. 1 be partitioned among two nodes, each having four processors, i.e., processors 1–4 and processors 5–8 are located in two different nodes. Then, all communications during the aggregation and disaggregation stages from level 1 to level 3 are performed "inside" nodes. Inter-node communications are required only for translations and data exchanges during the aggregation/disaggregation stages between level 3 and level 4. In general, the hierarchical partitioning strategy facilitates the processor arrangements in nonuniform platforms to minimize inter-node communications. However, in this paper, we do not use this advantage directly; hence, the improved efficiency obtained with the hierarchical parallelization is general and valid for all types of distributed-memory architectures.

VI. Results

The results of this paper can be categorized into three parts. First, we demonstrate the improved efficiency provided by the hierarchical parallelization strategy, compared to the previous parallelization approaches, on scattering problems involving spheres of various sizes discretized with millions of unknowns. Second, parallelization efficiency is demonstrated on large-scale scattering problems involving a sphere (a canonical object) and an airborne target Flamme [23] (a complicated object). Finally, we present the solution of very large scattering problems involving a sphere of radius 210λ and the Flamme with a maximum dimension of 880λ, which are discretized with 204,823,296 and 204,664,320 unknowns, respectively. These are the solutions of the largest problems of their classes ever reported in the literature, to the best of our knowledge.

A. Formulation and Solution Parameters

In this paper, scattering problems involve closed conductors, which can be formulated with CFIE. Matrix equations provided by CFIE are usually better conditioned than those obtained with EFIE and MFIE [24], [25]. Using CFIE, iterative convergence is achieved rapidly, and it can be further accelerated by employing simple preconditioners, such as a block-diagonal preconditioner (BDP). In all solutions, problems are discretized with about $\lambda/10$ mesh size, and near-field interactions are calculated with maximum 1% error. For small problems involving 1.5–13.5 million unknowns, tree structures are constructed by using a bottom-up strategy, and far-field interactions are calculated with three digits of accuracy. For large problems (involving more than 100 million unknowns), we use a top-down

strategy to construct the tree structures, while the far-field interactions are calculated with two digits of accuracy. During the aggregation stage, interpolations from a level to the next higher level are performed using 6×6 samples in the coarse grid for each sample in the fine grid. Finally, iterative solutions are performed using the biconjugate-gradient-stabilized (BiCGStab) algorithm [26] accelerated with BDP, and the residual error for the iterative convergence is set to 10^{-6} and 10^{-3} for small and large problems, respectively.

B. Parallel Computing Platforms

Scattering problems are solved on three different parallel clusters, each involving 16 computing nodes.

- Tigerton Cluster: Each node has 32 gigabytes (GB) of memory and two quad-core Intel Xeon Tigerton processors with 2.93 GHz clock rate;
- Harpertown Cluster: Each node has 32 GB of memory and two quad-core Intel Xeon Harpertown processors with 3.00 GHz clock rate;
- Dunnington Cluster: Each node has 48 GB of memory and four six-core Intel Xeon Dunnington processors with 2.40 GHz clock rate.

In all three clusters, memory in a node is available to all cores in the node. The nodes are connected via Infiniband networks, while the processors in a node are connected through high-speed mainboard buses. In the context of parallelization, we use the terms "processor" and "core" synonymously. For a solution on p processors, we use the maximum number of nodes available, i.e., the number of processes per node is minimized. In other words, if a code is parallelized into 2^i processes, and if $2^i \leq 16$, then we use 2^i nodes, each running only one process. When $2^i > 16$, however, the solution is parallelized over 16 nodes, and $2^i/16$ processors are employed per node.

C. Parallel Efficiency Results and Comparisons

The solutions presented in this subsection are performed on the Tigerton cluster. Fig. 4 presents the parallelization efficiency obtained for the solution of a scattering problem involving a sphere of radius 20λ discretized with 1,462,854 unknowns. Fig. 4(a) depicts the efficiency for the total time (including the setup and iterations), when the solution is parallelized onto 2, 4, 8, 16, 32, 64, and 128 processors. The parallelization efficiency is defined as

$$\varepsilon_p = \frac{T_1}{pT_p} \tag{20}$$

where T_p is the processing time of the solution with p processors. Fig. 4(a) shows that the hierarchical scheme improves the parallelization efficiency significantly, compared to simple and hybrid approaches, especially when the number of processors is large. The hybrid parallelization performs better than the simple parallelization; however, its efficiency drops rapidly for $p > 32$, and it becomes inefficient, compared to the hierarchical parallelization. Using 128 processors, the hierarchical parallelization technique provides 58% efficiency, corresponding to a 74-fold speedup with respect to the single-processor solution.

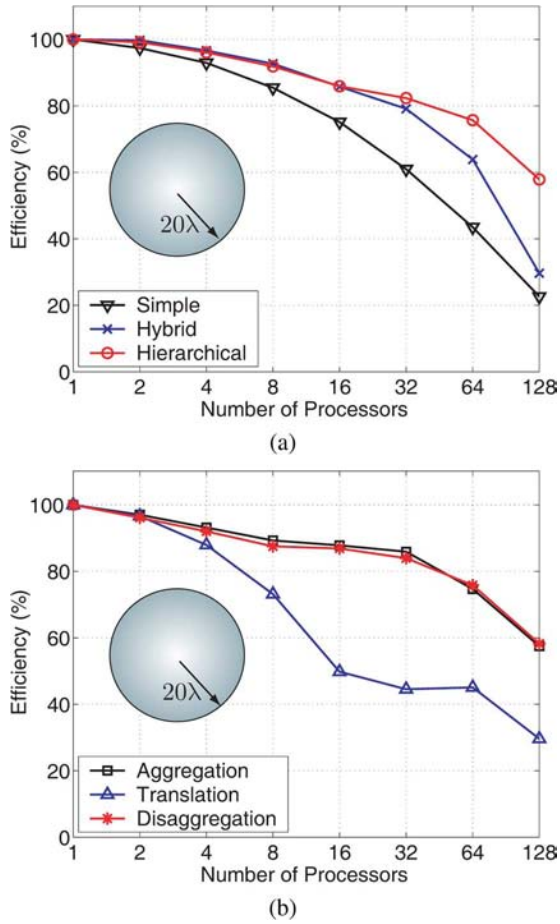


Fig. 4. Parallelization efficiency with respect to the number of processors for the solution of a scattering problem involving a sphere of radius 20λ discretized with 1,462,854 unknowns. (a) Overall efficiency including setup and iterations, when the solution is parallelized by using simple, hybrid, and hierarchical techniques. (b) Efficiencies for MLFMA stages, i.e., aggregation, translation, and disaggregation, using the hierarchical technique.

Fig. 4(b) presents the parallelization efficiency for the three stages of MVMs, i.e., aggregation, translation, and disaggregation, using the hierarchical strategy. We observe that the translation stage is a major bottleneck in the hierarchical parallelization of MLFMA. For a solution on 128 processors, the parallelization efficiency of translations drops below 30%. This is because the communication time for translations, given in (19), does not scale with the number of processors p, unlike the communication time for the aggregation and disaggregation stages. In addition, many communications required for inter-processor translations occur among processors located in different nodes. Then, the rate of communications during the translation stage is mostly limited by the speed of the Infiniband network. Nevertheless, even the parallelization of translations is improved with the hierarchical parallelization technique, and the overall efficiency provided by the hierarchical algorithm is consistently higher than those obtained with simple and hybrid approaches.

Fig. 5 presents the parallelization efficiency for solutions of scattering problems involving spheres of radii 40λ and 60λ discretized with 5,851,416 and 13,278,096 unknowns, respectively, where the efficiency is defined with respect to solutions

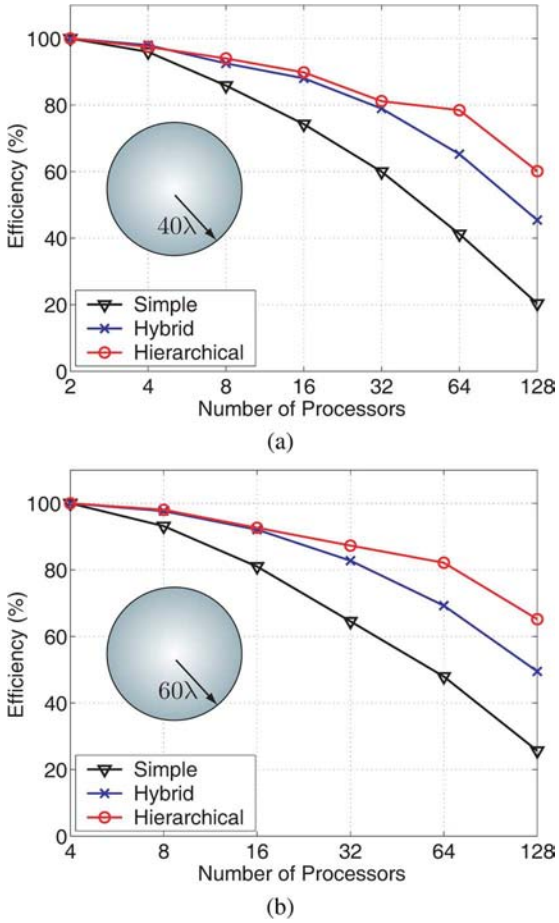


Fig. 5. Parallelization efficiency for the solution of scattering problems involving (a) a sphere of radius 40λ discretized with 5,851,416 unknowns and (b) a sphere of radius 60λ discretized with 13,278,096 unknowns. Parallel efficiency is defined with respect to two and four processors, respectively.

with two and four processors. Similar to the previous results, the parallelization efficiency is increased significantly by using the hierarchical parallelization technique.

Even though Figs. 4 and 5 compare the relative performances of different parallelization techniques, we emphasize that they do not provide the complete information for the efficiency of solutions. In general, one should also consider the following factors.

- Clock Rate of the Processors: Although using faster processors leads to faster solutions, the parallelization efficiency can be degraded as the computation time is reduced. This is because the communication time becomes more significant when the processing time for computations is small.
- Efficient Implementation of the Algorithm: We carefully implement MLFMA by minimizing the processing time, which may also have an adverse effect on the parallelization efficiency. For example, as opposed to common implementations of MLFMA, we calculate and store radiation and receiving patterns of basis and testing functions during the setup of the program, and we use them efficiently during iterations. Calculating the patterns on the fly in each MVM without storing them is also a common practice for low-memory implementations. That would increase the processing time, but the parallelization efficiency would also increase, since those calculations can be parallelized very efficiently.
- Accuracy Parameters: The accuracy of solutions also affects the parallelization efficiency. For example, most of the communications during the aggregation and disaggregation stages could be avoided by reducing the number of interpolation points. This would increase the parallelization efficiency, but the accuracy of the solutions would deteriorate.

TABLE I
SOLUTIONS OF SPHERE PROBLEMS

Sphere Radius	20λ	40λ	60λ
Unknowns	1,462,854	5,851,416	13,278,096
Number of Iterations	27	30	43
SETUP TIME (seconds)			
Simple	184	1065	2630
Hybrid	89	356	812
Hierarchical	88	348	797
SOLUTION TIME (seconds)			
Simple	512	2467	5400
Hybrid	442	1162	3122
Hierarchical	184	804	2257

TABLE II
TOTAL PROCESSING TIME AND PARALLELIZATION EFFICIENCY FOR THE SOLUTION OF SCATTERING PROBLEMS DISCRETIZED WITH MORE THAN 100 MILLION UNKNOWNS

SPHERE (Radius: 360λ, Number of Unknowns: 135,164,928)			
	16 Processors	32 Processors	64 Processors
Total Time (minutes)	975	511	292
Efficiency	100%	96%	84%
FLAMME (Length: 720λ, Number of Unknowns: 134,741,760)			
	16 Processors	32 Processors	64 Processors
Total Time (minutes)	1186	646	345
Efficiency	100%	92%	86%

We note that parallel-efficiency results presented in Figs. 4 and 5 are obtained under strict circumstances, using an *efficient* and *accurate* implementation of MLFMA on a cluster of processors with a relatively *high clock rate*. To quantify the efficiency of the solutions, Table I lists processing times, when the three problems are solved on 128 processors. Using the hierarchical parallelization technique, the largest problem with 13,278,096 unknowns is solved in less than one hour.

D. Parallel Efficiency for Large-Scale Problems

Table II presents the solution of scattering problems discretized with more than 100 million unknowns. A sphere of radius 180λ is discretized with 135,164,928 unknowns and solved by a 10-level MLFMA. We also consider a stealth airborne target, namely, the Flamme [23], having a maximum dimension of 6 meters (720λ at 36 GHz) and discretized with 134,741,760 unknowns. This problem is solved by an 11-level

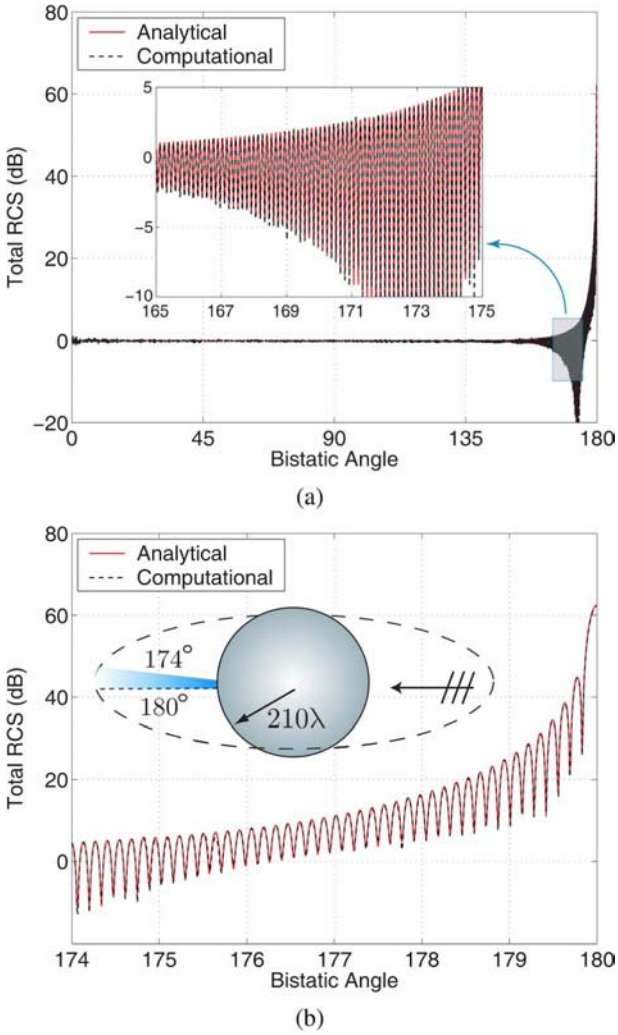


Fig. 6. Bistatic RCS (in dB) of a sphere of radius 210λ discretized with 204,823,296 unknowns (a) from 0° to 180° and (b) from 174° to 180°, where 0° and 180° correspond to back-scattering and forward-scattering directions, respectively.

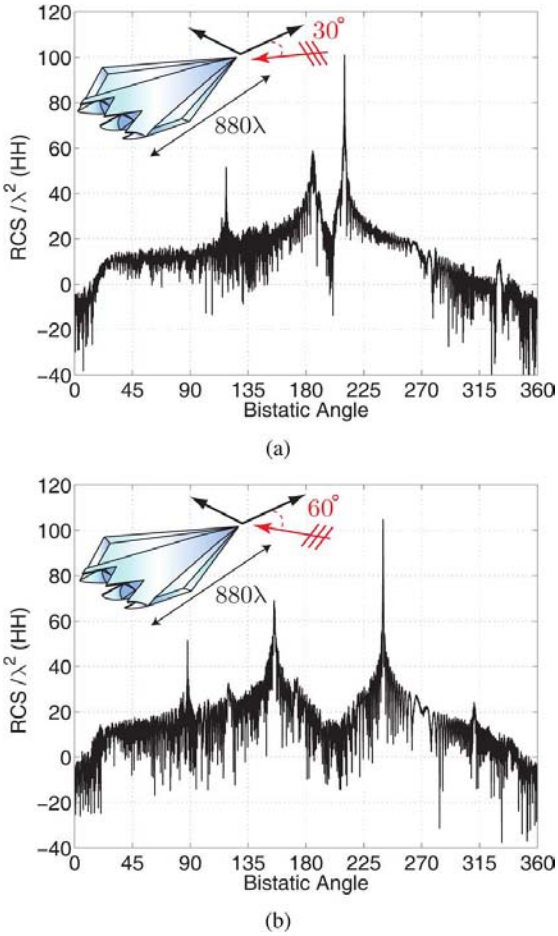


Fig. 7. Normalized co-polar bistatic RCS (RCS/λ^2 in dB) of the stealth airborne target Flamme at 44 GHz. Maximum dimension of the Flamme is 6 meters, corresponding to 880λ. The target is illuminated by plane waves propagating in the x-y plane at (a) 30° and (b) 60° angles from the x axis, with the electric field polarized in ϕ direction (horizontal polarization).

MLFMA. Both of the objects are illuminated by a plane wave, and solutions are performed on 16, 32, and 64 processors of the Harpertown cluster. The number of iterations is 23 and 44 for the sphere and the Flamme, respectively. Table II lists the total processing times including the setup and iterative solution parts, and the parallelization efficiency obtained for 32 and 64 processors with respect to 16 processors. Using 64 processors, parallelization efficiency is more than 80% for both problems. Due to this relatively high efficiency provided by the hierarchical partitioning strategy, we are able to perform each solution in five to six hours.

E. Solutions of Very Large Problems

Finally, we present the solutions of very large scattering problems discretized with more than 200 million unknowns. A sphere of radius 210λ is discretized with 204,823,296 unknowns and solved on 64 processors of the Dunnington cluster. Fig. 6(a) presents the normalized bistatic radar cross section values ($\mathrm{RCS}/\pi a^2$, where a is the radius of the sphere in meters) in decibels (dB) from 0° to 180° such that 0° corresponds to the back-scattering direction. Fig. 6(b) presents the same results from 174° to 180°. We observe that computational values are in agreement with the analytical values obtained by a Mie-series solution. The solution of the problem using the hierarchical parallelization strategy requires 645 minutes. Following the setup, which takes about 280 minutes, the iterative solution involving 25 iterations is performed in 360 minutes.

Fig. 7 presents the solution of a scattering problem involving the complicated target Flamme at 44 GHz. The maximum dimension of the Flamme is 880λ at this frequency. Discretization of the problem with $\lambda/10$ mesh size leads to 204,664,320 unknowns. As depicted in the insets of Fig. 7, the nose of the Flamme is directed towards the x axis, and it is illuminated by two plane waves (individually) propagating in the x-y plane at 30° and 60° angles from the x axis. The electric field is polarized in the ϕ direction (horizontal polarization). After the setup, which takes 265 minutes, the problem is solved twice for the two excitations in about 1300 minutes. The number of iterations for 30° and 60° illuminations are 38 and 42, respectively. Fig. 7 presents the normalized co-polar bistatic RCS (RCS/λ^2 in dB) on the x-y plane as a function of the bistatic angle ϕ. For the 30°

illumination in Fig. 7(a), 30° and 210° correspond to back-scattering and forward-scattering directions, respectively. We observe that the back-scattered RCS of the Flamme is extremely low; it is 90 dB less than the forward-scattered RCS. This is also observed for the 60° illumination in Fig. 7(b), where the back-scattered RCS at 60° is significantly lower than the forward-scattered RCS at 240°, due to the stealth property of the Flamme.

VII. Conclusion

We present the details of a hierarchical partitioning strategy for the efficient parallelization of MLFMA on relatively inexpensive computing platforms with distributed-memory architectures. Our algorithm is based on partitioning both clusters and field samples among processors at all levels of the multilevel tree structure. This way, load-balancing is improved significantly, compared to previous parallelization approaches based on partitioning with respect to only clusters or only samples of fields. We demonstrate the improved efficiency provided by the hierarchical technique on large scattering problems discretized with millions of unknowns. We also present the solution of very large scattering problems discretized with more than 200 million unknowns. For accurate investigations of complicated targets, such as the Flamme, solutions obtained with parallel MLFMA are extremely important. This is because approximate methods, such as physical optics (PO), may not provide accurate results for those problems, even when objects are large.

Acknowledgment

The authors would like to thank J. Wilcox of Intel Corporation for the expert technical support and the invaluable assistance he provided to facilitate the experiments on parallel computers. Computer time was provided in part by a generous allocation from Intel Corporation.

References

[1] A. J. Poggio and E. K. Miller, "Integral equation solutions of three-dimensional scattering problems," in *Computer Techniques for Electromagnetics*, R. Mittra, Ed. Oxford, U.K.: Pergamon Press, 1973, ch. 4.

[2] J. R. Mautz and R. F. Harrington, "H-field, E-field, and combined field solutions for conducting bodies of revolution," *AEÜ*, vol. 32, no. 4, pp. 157–164, Apr. 1978.

[3] J. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[4] S. Velamparambil, W. C. Chew, and J. Song, "10 million unknowns: Is it that big?," *IEEE Antennas Propag. Mag.*, vol. 45, no. 2, pp. 43–58, Apr. 2003.

[5] M. L. Hastriter, "A study of MLFMA for large-scale scattering problems," Ph.D. dissertation, Univ. Illinois at Urbana-Champaign, 2003.

[6] G. Sylvand, "Performance of a parallel implementation of the FMM for electromagnetics applications," *Int. J. Numer. Meth. Fluids*, vol. 43, pp. 865–879, 2003.

[7] S. Velamparambil and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2719–2727, Aug. 2005.

[8] Ö. Ergül and L. Gürel, "Efficient parallelization of multilevel fast multipole algorithm," presented at the Proc. Eur. Conf. on Antennas and Propag. (EuCAP), 2006.

[9] L. Gürel and Ö. Ergül, "Fast and accurate solutions of integral-equation formulations discretised with tens of millions of unknowns," *Electron. Lett.*, vol. 43, no. 9, pp. 499–500, Apr. 2007.

[10] Ö. Ergül and L. Gürel, "Efficient parallelization of the multilevel fast multipole algorithm for the solution of large-scale scattering problems," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2335–2345, Aug. 2008.

[11] J. Fostier and F. Olyslager, "An asynchronous parallel MLFMA for scattering at multiple dielectric objects," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2346–2355, Aug. 2008.

[12] Ö. Ergül and L. Gürel, "Hierarchical parallelisation strategy for multilevel fast multipole algorithm in computational electromagnetics," *Electron. Lett.*, vol. 44, no. 1, pp. 3–5, Jan. 2008.

[13] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, no. 3, pp. 409–418, May 1982.

[14] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: a pedestrian prescription," *IEEE Antennas Propag. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993.

[15] S. Koc, J. M. Song, and W. C. Chew, "Error analysis for the numerical evaluation of the diagonal forms of the scalar spherical addition theorem," *SIAM J. Numer. Anal.*, vol. 36, no. 3, pp. 906–921, 1999.

[16] Ö. Ergül and L. Gürel, "Optimal interpolation of translation operator in multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 54, no. 12, pp. 3822–3826, Dec. 2006.

[17] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernels," *Comp. Phys. Comm.*, vol. 65, pp. 24–38, Apr. 1991.

[18] R. D. Graglia, "On the numerical integration of the linear shape functions times the 3-D Green's function or its gradient on a plane triangle," *IEEE Trans. Antennas Propag.*, vol. 41, no. 10, pp. 1448–1455, Oct. 1993.

[19] R. E. Hodges and Y. Rahmat-Samii, "The evaluation of MFIE integrals with the use of vector triangle basis functions," *Microw. Opt. Technol. Lett.*, vol. 14, no. 1, pp. 9–14, Jan. 1997.

[20] P. Y.-Oijala and M. Taskinen, "Calculation of CFIE impedance matrix elements with RWG and $\hat{\boldsymbol{n}}\times$ RWG functions," *IEEE Trans. Antennas Propag.*, vol. 51, no. 8, pp. 1837–1846, Aug. 2003.

[21] L. Gürel and Ö. Ergül, "Singularity of the magnetic-field integral equation and its extraction," *IEEE Antennas Wireless Propag. Lett.*, vol. 4, pp. 229–232, 2005.

[22] D. A. Dunavant, "High degree efficient symmetrical Gaussian quadrature rules for the triangle," *Int. J. Numer. Meth. Eng.*, vol. 21, pp. 1129–1148, 1985.

[23] L. Gürel, H. Bağcı, J. C. Castelli, A. Cheraly, and F. Tardivel, "Validation through comparison: measurement and calculation of the bistatic radar cross section (BRCS) of a stealth target," *Radio Sci.*, vol. 38, no. 3, Jun. 2003.

[24] D. R. Wilton and J. E. Wheeler, III, "Comparison of convergence rates of the conjugate gradient method applied to various integral equation formulations," *Progr. in Electromagn. Res.*, pp. 131–158, 1991.

[25] L. Gürel and Ö. Ergül, "Extending the applicability of the combined-field integral equation to geometries containing open surfaces," *IEEE Antennas Wireless Propag. Lett.*, vol. 5, pp. 515–516, 2006.

[26] H. A. v. d. Vorst, "Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 13, no. 2, pp. 631–644, Mar. 1992.

Özgür Ergül (S'98) received the B.Sc. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2001 and 2003, respectively, where he is currently working toward the Ph.D. degree.

Since 2001, he has served as a Teaching and Research Assistant in the Department of Electrical and Electronics Engineering, Bilkent University. He has been affiliated with the Computational Electromagnetics Group at Bilkent University from 2000 to 2005 and with the Computational Electromagnetics Research Center (BiLCEM) since 2005. His research interests include fast and accurate algorithms for the solution of electromagnetics problems involving large and complicated structures, integral equations, parallel programming, and iterative techniques.

Mr. Ergül is a recipient of the 2007 IEEE Antennas and Propagation Society Graduate Fellowship and the 2007 Leopold B. Felsen Award for Excellence in Electrodynamics. He is the Secretary of Commission E (Electromagnetic Noise and Interference) of URSI Turkey National Committee. His academic endeavors are supported by the Scientific and Technical Research Council of Turkey (TUBITAK) through a Ph.D. scholarship.

Levent Gürel (S'87–M'92–SM'97–F'09) received the B.Sc. degree from the Middle East Technical University (METU), Ankara, Turkey, in 1986 and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign (UIUC) in 1988 and 1991, respectively, all in electrical engineering.

He joined the Thomas J. Watson Research Center of the International Business Machines Corporation, Yorktown Heights, NY, in 1991, where he worked as a Research Staff Member on the electromagnetic compatibility (EMC) problems related to electronic packaging, on the use of microwave processes in the manufacturing and testing of electronic circuits, and on the development of fast solvers for interconnect modeling. Since 1994, he has been a faculty member in the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, where he is currently a Professor. He was a Visiting Associate Professor at the Center for Computational Electromagnetics (CCEM), UIUC, for one semester in 1997. He returned to the UIUC as a Visiting Professor in 2003–2005, and as an Adjunct Professor after 2005. He founded the Computational Electromagnetics Research Center (BiLCEM) at Bilkent University in 2005, where he is serving as the Director. His research interests include the development of fast algorithms for computational electromagnetics (CEM) and the application thereof to scattering and radiation problems involving large and complicated scatterers, antennas and radars, frequency-selective surfaces, high-speed electronic circuits, optical and imaging systems, nanostructures, and metamaterials. He is also interested in the theoretical and computational aspects of electromagnetic compatibility and interference analyses. Ground penetrating radars and other subsurface scattering applications are also among his research interests. Since 2006, his research group has been breaking several world records by solving extremely large integral-equation problems, most recently the largest involving as many as 205 million unknowns.

Prof. Gürel's many accomplishments include two prestigious awards from the Turkish Academy of Sciences (TUBA) in 2002 and the Scientific and Technical Research Council of Turkey (TUBITAK) in 2003, which are the most notable. He served as the Chairman of the AP/MTT/ED/EMC Chapter of the IEEE Turkey Section in 2000–2003. He founded the IEEE EMC Chapter in Turkey in 2000. He served as the Cochairman of the 2003 IEEE International Symposium on Electromagnetic Compatibility. He is the organizer and General Chair of the CEM'07 ve CEM'09 Computational Electromagnetics International Workshops held in 2007 and 2009. He is a member of the USNC of the International Union of Radio Science (URSI) and the Chairman of Commission E (Electromagnetic Noise and Interference) of URSI Turkey National Committee. He served as a member of the General Assembly of the European Microwave Association (EuMA) during 2006–2008. He is currently serving as an Associate Editor for *Radio Science, the IEEE Antennas and Wireless Propagation Letters, Journal of Electromagnetic Waves and Applications (JEMWA)*, and *Progress in Electromagnetics Research (PIER)*.