A MULTILEVEL ALGORITHM FOR SOLVING BOUNDARY INTEGRAL EQUATION †

Cai-Cheng Lu and Weng Cho Chew
Department of Electrical and Computer Engineering
University of Illinois
Urbana, IL 61801

Abstract

In the solution of an integral equation using the Conjugate Gradient (CG) method, the most expansive part is the matrix-vector multiplication, requiring $O(N^2)$ floating point operations. The Fast Multipole Method (FMM) reduced the operation to $N^{1.5}$. In this paper, we apply a multilevel algorithm to this problem and show that the complexity of a matrix-vector multiplication is proportional to $N(\log(N))^2$.

† This work was supported by NASA under contract NASA NAG 2-871, Office of Naval Research under grant N00014-89-J1286, and the Army Research Office under contract DAAL03-91-G-0339, and the National Science Foundation under grant NSF-ECS-92-24466. The computer time was provided by the National Center for Supercomputing Applications (NCSA) at the University of Illinois, Urbana-Champaign.

Published in *Micro. Opt. Tech. Lett.*, Vol. 7, No. 10, pp. 466-470, July, 1994.

File:mlfma1.tex, January 13, 1995

1. Introduction

Multilevel algorithms have been used to generate fast algorithms for Fourier transforms [1] and inversion of matrices in finite element method [2]. They usually involve nesting a smaller problem within a larger problem. Recently, multilevel algorithms have been used to solve integral equations by expediting matrix-vector multiplies [3-5] or by finding the inverse of the integral operator[6]. Interpolation multilevel algorithm has also been proposed [7] although no numerical results have been shown. These algorithms could invert an integral operator in less than $O\left(N^3\right)$ operations [6] and expedite a matrix-vector multiply to $O\left(N \log N\right)$ [3] or $O\left(N\right)$ operations [5].

In this paper,* we will describe a multilevel algorithm for expediting matrix-vector multiply in an iterative solution of boundary integral equation. The algorithm has $O\left(N (\log N)^2\right)$ complexity, and for very large problem, $O\left(N \log N\right)$ complexity. This multilevel algorithm is used together with the fast multipole method [8] to achieve the reduced computational complexity. We call this the multilevel fast multipole algorithm (MLFMA).

2. Description of the Algorithm

A boundary integral equation for E_z incident wave is

$$-\phi_{inc}\left(\mathbf{r}\right) = -\frac{\omega\mu_0}{4}\int_C dl' g\left(\mathbf{r},\mathbf{r}'\right) J\left(\mathbf{r}'\right), \qquad \mathbf{r} \in C. \tag{1}$$

It can be discretized so that [9]

$$-\phi_{inc}\left(\mathbf{r}_j\right) = -\frac{\omega\mu_0}{4}\sum_{i=1}^{N} \triangle l \, g\left(\mathbf{r}_j,\mathbf{r}_i\right) J\left(\mathbf{r}_i\right) = \sum_{i=1}^{N} g_{ji} b_i, \qquad j = 1, 2, \cdots, N \tag{2}$$

where in the two dimensional case,

$$g_{ji} = \begin{cases} H_0^{(1)}\left(k\left|\mathbf{r}_j - \mathbf{r}_i\right|\right), & i \neq j, \\ 1 + \frac{2i}{\pi}\ln(0.163805 k\Delta l) & i = j, \end{cases}$$

and b_i is proportional to the unknown current. The above equation is of the form $\mathbf{b} = \overline{\mathbf{A}} \cdot \mathbf{x}$, and when solved iteratively using the conjugate gradient method, the most costly part is the matrix-vector multiply $\overline{\mathbf{A}} \cdot \mathbf{x}$. which involves N^2 operations

---

* An abridged version of this paper will be presented at the APS/URSI Symposium in Seattle, June 1994

To execute the matrix-vector multiply, we need to evaluate the summation on the right hand of equation (2) for a given j is to compute the field at point $\mathbf{r}_j$ due to sources at $i = 1, 2, \cdots, N, \quad i \neq j$. For E_z polarized wave, these sources can be considered as monopoles since their field can be written as $\psi_0(k_0, \mathbf{r}_{ji})b_i$, where $\psi_\nu(k_0, \mathbf{r}) = H_\nu^{(1)}(k_0 r)e^{i\nu\phi}$ is the cylindrical harmonics, and b_i is the harmonic amplitude.

In order to expedite the matrix-vector multiply, we need to first describe a multilevel algorithm. First, we divide all the subscatterers into multilevel groups. Assuming that there are N subscatterers where $N = 2^p$, and p is an integer. We first divide the N subscatterers into $N/2$ pairs. Then every two successive pairs are grouped to form $N/4$ subgroups. This process is repeated until the highest level (which contains two groups) is reached. As a result, there are altogether $\log_2 N - 1$ levels.

At the lowest level, each scattering center is a monopole. By translation of scattering centers, we can represent the field of several monopoles by that of a single multipole. Since there are two monopole subscatterers in each group, we can denote the monopole amplitudes in group l as b_i, $i = 2l - 1, 2l$, where $l = 1, \ldots, N/2$. Then the multipole amplitude $\mathbf{b}_l$ for group l can be obtained using translation matrices, namely,

$$\mathbf{b}_l = \sum_i \overline{\boldsymbol{\beta}}_{li} b_i$$

where $\mathbf{b}_l$ is a length Q vector, and $\overline{\boldsymbol{\beta}}_{li}$ is a $Q \times 1$ matrix [10]. By so doing, the scattering centers for each group have moved from $\mathbf{r}_i$ for the monopole subscatterers to $\mathbf{r}_l$ for the multipole group.

This grouping process is shown in Figure 1. In this figure, each small circle stands for the scattering center of a group. The actual scattering center of each group is located at the geometrical center of the subgroups contained in the group. They are drawn in different places only for the purpose explanation.

Now we can go through the process of aggregation and disaggregation. The aggregation process is to compute the outgoing wave harmonic coefficients of each group in each level. Mathematically, the aggregation from the n-th to the $(n+1)$-th level is described as

$$\underbrace{\mathbf{b}_{l(n+1)}}_{Q_{n+1}\times 1} = \sum_{i \in G_{l(n+1)}} \underbrace{\overline{\boldsymbol{\beta}}_{li}}_{Q_{n+1}\times Q_n} \cdot \underbrace{\mathbf{b}_{i(n)}}_{Q_n \times 1}, \quad l = 1, \cdots, N/2^n, \quad n = 1, \cdots, \log_2 N - 1. \quad (3)$$

where $\overline{\boldsymbol{\beta}}_{li}$ is a translation matrix [10] and where at the lowest level, $\mathbf{b}_{i(1)} = b_i, i = 1, \cdots, N$. Here, $G_{l(n+1)}$ is a group consisting of two subgroups. In general, $\mathbf{b}_{i(n)}$ is for $i = 1, \cdots, N/2^{n-1}$, and $Q_{n+1} \approx 2Q_n$. Here, Q_n is the number of harmonics used to represent the multipole field of the aggregated subscatterers at the n-th level. The number of harmonics roughly doubles at each aggregation because the size of the subscatterers doubles.

For disaggregation, the inward going wave or harmonic amplitudes due to all the other groups except the nearest-neighbor groups are computed for each group. Denoting the inward going harmonic amplitudes by the vector $\mathbf{S}_{i(n)}$, it can be calculated recursively in a nested manner as

$$\underbrace{\mathbf{S}_{i(n)}}_{Q_n \times 1} = \sum_{i \in g_{l(n)}} \underbrace{\overline{\boldsymbol{\alpha}}_{il}}_{Q_n \times Q_n} \cdot \underbrace{\mathbf{b}_{l(n)}}_{Q_n \times 1} + \underbrace{\overline{\boldsymbol{\beta}}_{il'}}_{Q_n \times Q_{n+1}} \cdot \underbrace{\mathbf{S}_{l'(n+1)}}_{Q_{n+1} \times 1} \tag{4}$$

where $\overline{\boldsymbol{\alpha}}_{il}$ and $\overline{\boldsymbol{\beta}}_{il'}$ are translation matrices [10]. $\mathbf{S}_{l'(n+1)}$ contains the incoming wave from all the other subscatterers except for those belonging to $\mathbf{b}_{l'+1(n+1)}$ and $\mathbf{b}_{l'-1(n+1)}$ which are the nearest neighbors to $\mathbf{S}_{l'(n+1)}$ at the $(n+1)$-th level. $g_{l(n)}$ are groups at the n-th level that are contained in the groups of $\mathbf{b}_{l'\pm 1(n+1)}$, but they are not the nearest neighbors of the group of $\mathbf{S}_{i(n)}$.

After disaggregation, i.e., when $\mathbf{S}_{i(1)}, \quad i = 1, 2, \cdots, N$ are known at the lowest level, we can then compute the summation on the right hand of Equation (2) as

$$\sum_{i=1}^{N} g_{ji} b_i = s_{i(1)} + \sum_{i=j-1}^{j+1} g_{ji} b_i$$

The second term on the right hand side of the above accounts for self and nearest neighbor contributions.

It is seen that at level n, we have $N/2^n$ subgroups, to compute the aggregated multipole amplitude of each subgroup. The matrix-vector multiplies require operation count of $Q_n Q_{n-1} \leq Q_n^2 \approx 2^{2n}$ at each level. The total operations is

$$\sum_{n=1}^{p-1} \frac{N}{2^n} 2^{2n} = N(2^p - 1) \approx N^2$$

where $p = \log_2 N$ is proportional to the number of levels present. Hence, the above method is still an $O(N^2)$ algorithm.

However, the translation operation is a convolution because the (n, m) element of the matrices $\overline{\boldsymbol{\alpha}}$ or $\overline{\boldsymbol{\beta}}$ depends only on $n - m$. Thus $\overline{\boldsymbol{\alpha}} \cdot \mathbf{b}$ or $\overline{\boldsymbol{\beta}} \cdot \mathbf{b}$ can be computed use FFT in $Q_n \log Q_n$ instead of Q_n^2. This results in the total operation count of $N(\log N)^2$. Alternatively, this can be viewed as the diagonalization of the $\overline{\boldsymbol{\alpha}}$ and $\overline{\boldsymbol{\beta}}$ matrices as presented in the next section.

3. Diagonalization of the Matrices

To further reduce the computational complexity, the translation matrices $\overline{\boldsymbol{\alpha}}$ and $\overline{\boldsymbol{\beta}}$ need to be diagonalized in the manner suggested by [8] and also described in [11].

To diagonalize (3), we notice that

$$\left[\overline{\boldsymbol{\beta}}_{li}\right]_{m'n'} = J_{m'-n'}\left(k\rho_{li}\right) e^{-i(m'-n')\phi_{li}}$$

$$= \frac{1}{2\pi}\int_0^{2\pi} d\alpha e^{ik\rho_{li}\cos(\alpha+\phi_{li})+i(m'-n')\left(\alpha-\frac{\pi}{2}\right)}. \tag{5}$$

Therefore,

$$\begin{aligned} I_{m'} &= \left[\overline{\boldsymbol{\beta}}_{li}\cdot\mathbf{b}_{i(n)}\right]_{m'} \\ &= \frac{1}{2\pi}\int_0^{2\pi} d\alpha e^{ik\rho_{li}\cos(\alpha+\phi_{li})} e^{im'\left(\alpha-\frac{\pi}{2}\right)} \sum_{n'=-N'}^{N'} e^{-in'\left(\alpha-\frac{\pi}{2}\right)}\left[\mathbf{b}_{i(n)}\right]_{n'} \\ &= \frac{1}{2\pi}\int_0^{2\pi} d\alpha e^{im'\left(\alpha-\frac{\pi}{2}\right)} e^{ik\rho_{li}\cos(\alpha+\phi_{li})}\widetilde{b}_{i(n)}\left(\alpha\right) \end{aligned} \tag{6}$$

where we have defined

$$\widetilde{b}_{i(n)}\left(\alpha\right) = \sum_{n'=-N'}^{N'} e^{-in'\left(\alpha-\frac{\pi}{2}\right)}\left[\mathbf{b}_{i(n)}\right]_{n'}, \tag{7}$$

and $2N'+1 = Q_n$ in (3). In the above, $\widetilde{b}_{i(n)}\left(\alpha\right)$ is the Fourier series transform of $\left[\mathbf{b}_{i(n)}\right]_{n'}$ and $\left[\mathbf{b}_{i(n)}\right]_{n'}$ is a finite bandwidth signal. Because of this, we can express (7) in its inverse Fourier series transform to obtain

$$\left[\mathbf{b}_{i(n)}\right]_{n'} = \frac{1}{2\pi}\int_0^{2\pi} d\alpha e^{in'\left(\alpha-\frac{\pi}{2}\right)}\widetilde{b}_{i(n)}\left(\alpha\right). \tag{8}$$

Consequently, (3) becomes

$$\widetilde{b}_{l(n+1)}\left(\alpha\right) = \sum_{i\in G_{l(n+1)}} e^{ik\rho_{li}\cos(\alpha+\phi_{li})}\widetilde{b}_{i(n)}\left(\alpha\right) \tag{9}$$

where $e^{ik\rho_{li}\cos(\alpha+\phi_{li})}$ is the diagonalized form of $\overline{\boldsymbol{\beta}}_{li}$ in the transformed space, and the matrix-vector multiply is easily carried out in the transformed space.

To diagonalize (4), we note that it can be written as

$$\left[\mathbf{S}_{i(n)}\right]_{n'} = \sum_{l \in g_{l(n)}} \sum_{m'=-\infty}^{\infty} \left[\overline{\boldsymbol{\alpha}}_{il}\right]_{n'm'} \left[\mathbf{b}_{l(n)}\right]_{m'} W_{n'-m'} + \sum_{m'=-\infty}^{\infty} \left[\overline{\boldsymbol{\beta}}_{il'}\right]_{n'm'} \left[\mathbf{S}_{l'(n+1)}\right]_{m'}. \tag{10}$$

$W_{n'-m'}$ is a window function that truncates the first infinite summation above, which may diverge. In (10), $-N' < n' < N'$ since $\left[\mathbf{S}_{i(n)}\right]_{n'}$ is a finite bandwidth signal. Now, we let

$$\widetilde{S}_{i(n)}(\alpha) = \sum_{n'=-N'}^{N'} e^{-in'\left(\alpha - \frac{\pi}{2}\right)} \left[\mathbf{S}_{i(n)}\right]_{n'} \tag{11}$$

and

$$\left[\mathbf{b}_{l(n)}\right]_{m'} = \frac{1}{2\pi} \int_0^{2\pi} d\alpha' e^{im'\left(\alpha' - \frac{\pi}{2}\right)} \widetilde{b}_{l(n)}(\alpha'). \tag{12}$$

Next, we multiply (10) by $e^{-in'\left(\alpha - \frac{\pi}{2}\right)}$ and sum over n'. Then, the first infinite summation on the right hand side of (10) becomes, after using (12), and exchanging the order of integration and summation,

$$\frac{1}{2\pi} \int_0^{2\pi} d\alpha' \widetilde{b}_{l(n)}(\alpha') \sum_{n'=-N'}^{N'} e^{-in'\left(\alpha - \frac{\pi}{2}\right)} \sum_{m'=-\infty}^{\infty} e^{im'\left(\alpha' - \frac{\pi}{2}\right)} \left[\overline{\boldsymbol{\alpha}}_{il'}\right]_{n'm'} W_{n'-m'}$$

$$= \int_0^{2\pi} d\alpha' \widetilde{b}_{l(n)}(\alpha') \frac{\sin\left[\frac{2N'+1}{2}(\alpha - \alpha')\right]}{2\pi \sin\left(\frac{\alpha - \alpha'}{2}\right)} v_{il}(\alpha') \tag{13}$$

where

$$v_{il}(\alpha') = \sum_{p=-\infty}^{\infty} e^{-ip\left(\alpha' - \frac{\pi}{2}\right)} \left[\overline{\boldsymbol{\alpha}}_{il}\right]_p W_p. \tag{14}$$

In the above, we have made use of the fact that $\overline{\boldsymbol{\alpha}}_{il}$ is a Toeplitz matrix. In Equation (3), the wave amplitude in the transformed space, $\widetilde{b}_{l(n)}(\alpha')$ is propagated from center l to center i by $v_{il}(\alpha')$ and then smoothed by convolving with $\sin\left(\frac{2N'+1}{2}\alpha'\right) / \sin\left(\frac{1}{2}\alpha'\right)$.

The second infinite summation in (10) can be simplified by a similar manipulation. First, we let

$$\left[\mathbf{S}_{l'(n+1)}\right]_{m'} = \frac{1}{2\pi} \int_0^{2\pi} d\alpha' e^{-im\left(\alpha' - \frac{\pi}{2}\right)} \widetilde{S}_{l'(n+1)}(\alpha') \tag{15}$$

so that the infinite summation becomes

$$\frac{1}{2\pi}\int_0^{2\pi} d\alpha' \widetilde{S}_{l'(n+1)}(\alpha') \sum_{n'=-N'}^{N'} e^{-in'\left(\alpha-\frac{\pi}{2}\right)} \sum_{m'=-\infty}^{\infty} e^{-im'\left(\alpha'-\frac{\pi}{2}\right)} \left[\overline{\boldsymbol{\beta}}_{il'}\right]_{n'm'}$$

$$= \int_0^{2\pi} d\alpha' \widetilde{S}_{l'(n+1)}(\alpha') \frac{\sin\left[\frac{2N'+1}{2}(\alpha-\alpha')\right]}{2\pi\sin\left(\frac{\alpha-\alpha'}{2}\right)} e^{ik\rho_{li'}\cos(\alpha'+\phi_{il'})}. \tag{16}$$

Consequently, Equation (4) or (10) in the transformed space, can be written as

$$\widetilde{S}_{i(n)}(\alpha) =$$

$$\frac{\sin\left(\frac{N'+1}{2}\alpha\right)}{2\pi\sin\left(\frac{\alpha}{2}\right)} \otimes \left[\sum_{l\in g_{l(n)}} v_{il}(\alpha)\,\widetilde{b}_{l(n)}(\alpha) + e^{ik\rho_{li'}\cos(\alpha+\phi_{il'})}\widetilde{S}_{l'(n+1)}(\alpha)\right]. \tag{17}$$

It should be noted in (7), that since $\left[\mathbf{b}_{i(\alpha)}\right]_{n'}$ is finite in bandwidth, $\widetilde{b}_{i(n)}(\alpha)$ is uniquely specified by $Q_n = 2N'+1$ points. Since Q_n is small when n is small and Q_n roughly doubles when one goes from a low level to that previous level, the number of points in $\widetilde{b}_{l(n)}(\alpha)$ on the left hand side of (9) is roughly two times more than the number of points in $\widetilde{b}_{i(n)}(\alpha)$ on the right hand side of (9). Therefore, $\widetilde{b}_{i(n)}(\alpha)$ needs to be interpolated to twice its sampling density before it is multiplied by the propagator $e^{ik\rho_{li}\cos(\alpha+\phi_{il})}$, This interpolation can be done with an FFT requiring $O(Q_n \log Q_n)$ operations. Similarly, the convolution in (17) can be performed with $O(Q_n \log Q_n)$ operation. Since there are $\log N$ levels, the computational complexity is $O\left(N(\log N)^2\right)$.

When the problem becomes very large, the interpolation needed in (9) and the smoothing in (17) can be done with the nearest neighbor approximation, with $O(Q_n)$ operations. In this case, the computational complexity is $O(N\log N)$, when $N \to \infty$.

The above algorithm is derived for the electrical field integral equation (EFIE). It can be easily extended to other kinds of integral equations. For example, the combined field integral equation (CFIE) can be written as

$$\sum_{\substack{i=1\\ i\neq j}}^{N} \left(1+\chi\hat{n}_j\cdot\nabla_j\right) H_0^{(1)}\left(k_0|\mathbf{r}_i-\mathbf{r}_j|\right) b_i + g'_{jj}b_j = c_j$$

where

$$c_j = -\left(1+\chi\hat{n}_j\cdot\nabla_j\right)\phi^{inc}(\mathbf{r}_j)$$

$$g'_{jj} = g_{jj} - 2i\chi/\Delta l,$$

and χ is a constant.

This equation can be converted to

$$\sum_{\substack{i=1\\ i\neq j}}^{N} \boldsymbol{\psi}^t(\mathbf{r}_{ji}) \cdot \left[\begin{pmatrix} 0\\ 1\\ 0 \end{pmatrix} - \frac{ik\chi n_j}{2} \begin{pmatrix} 1\\ 0\\ 0 \end{pmatrix} + \frac{ik\chi n_j^*}{2} \begin{pmatrix} 0\\ 0\\ 1 \end{pmatrix} \right] b_i \; + g'_{jj} b_j = c_j.$$

Comparing this equation with Equation (2), we found that in the combined field integral equation, each subscatterer is a multipole of order one. As a result, let $\mathbf{b}^t_{i(1)} = (b_i, 0, 0)$, $(0, b_i, 0)$ and $(0, 0, b_i)$ respectively. The fast algorithm described above can still be used at a cost of increasing the operation count by a factor of 2.5. The numerical results given in the next section are obtained by applying the MLFMA to CFIE.

4. Numerical Results and Conclusions

In Figure 2, we show the current magnitude induced on a circular cylinder of radius 2λ. The incident wave is a TM polarized wave of frequency 300 MH_z and the incident angle is $\phi = \pi$. The agreement with closed form solution is excellent. Figure 3 shows the scattered field of the same cylinder. Figure 4 shows the growth of the CPU time (on Workstation) divided by the number of iterations for solving different size problems. The solid line is for the algorithm in this paper (MLFMA), the line with $+$ is for direct computation of $\overline{\mathbf{A}} \cdot \mathbf{b}$ and the line with $*$ is for the fast multipole method described in [11]. The crossover between this algorithm and FMM is expected around $N = 65536$.

In conclusion, we have developed a multilevel algorithm in two dimensions with $N(\log N)^2$ complexity. We also showed the possibility to reduce the complexity to $N \log N$ for very large problems.

References

[1] J.W. Cooley and J.W. Tukey, "An algorithm for the machine computation of complex Fourier series," *Math. Comp.*, vol. 19, pp. 297-301, 1965

[2] A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[3] V. Roklin, "Rapid solution of integral equations of classical potential theory," *J. Comput. Phys.*, vol. 60, pp. 187-207, 1985.

[4] K. Kalbasi and K.R. Demarest, "A multilevel enhancement of the method of moments," *7th Annual Review of Progress in Applied Computational*

Electromagnetics, Naval Postgraduate School, Monterey, CA, pp. 254-263, March 18-22, 1991.

[5] J. H. Lin and W. C. Chew, "An application of nested equivalence Principle algorithm (NEPAL) in matrix-vector multiplication of iterative algorithms," *IEEE Ant. Prop. Soc. Int. Symp. & URSI Radio Science Meeting,* June 28-July 2, 1993.

[6] W. C. Chew and C. C. Lu, "The use of Huygens' equivalence principle for solving he volume integral equation of scattering," *IEEE Trans. Ant. Propag.* vol. AP-41, no. 7, pp. 897-904, July 1993.

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

[8] V. Rokhlin, "Rapid solution of integral equations of scattering theory in two dimensions." *J. Comput. Phys.,* vol. 86, pp. 414-439, 1990.

[9] R. F. Harrington, *Field Computation by Moment Methods,* reprint ed., Krieger Pub. Co., Malabar, FL, 1983.

[10] W. C. Chew, *Waves and Fields in Inhomogeneous Media,* Van Nostrand Reinhold, New York, 1990.

[11] C. C. Lu and W. C. Chew, "A fast algorithm for solving hybrid integral equation," *IEE Proceedings,* scheduled for December 1993.

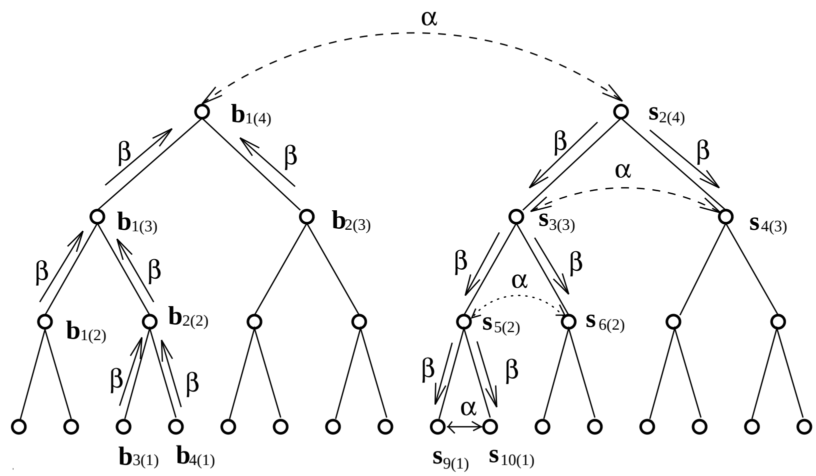

Figure 1 Multilevel groups.

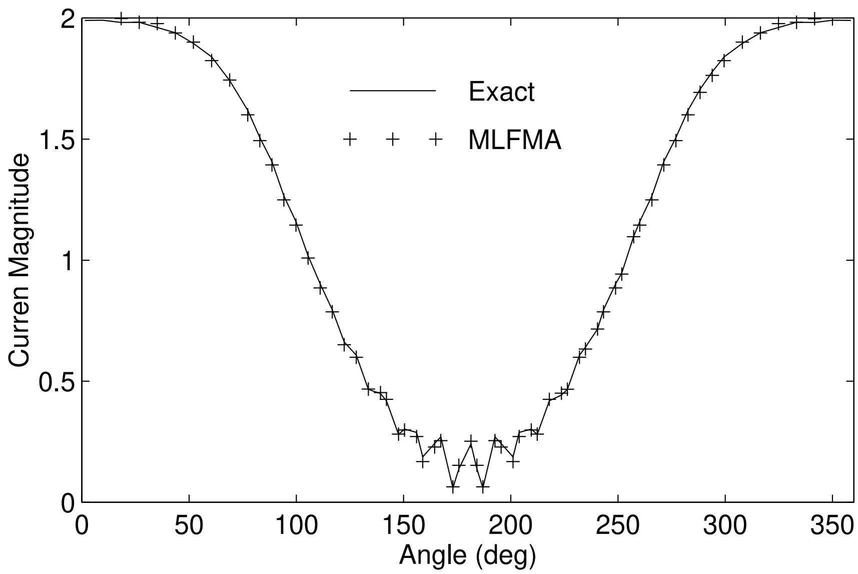

Figure 2 Induced current on a 2λ circular cylinder. Plane wave is polarized in E_z and is incident from $\theta^i = 0^o$. Frequency=300 MHz.

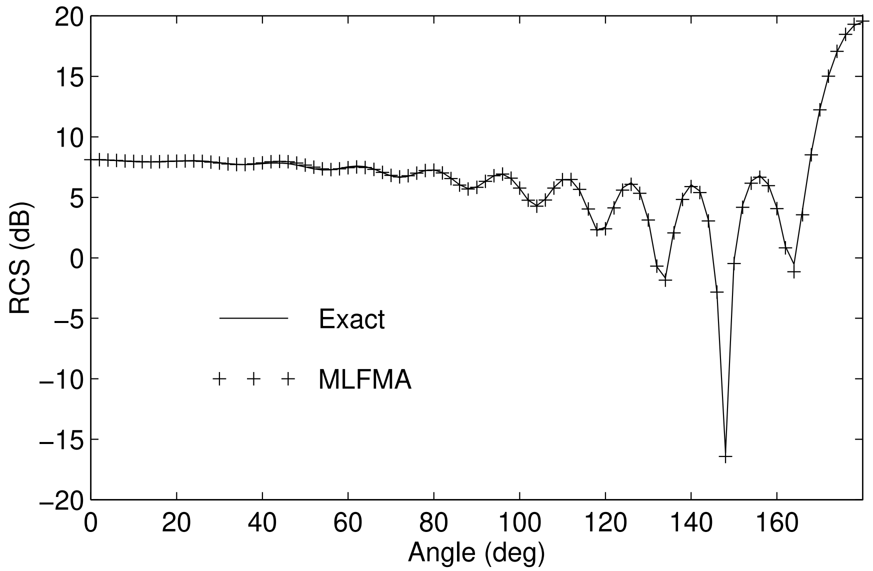

Figure 3 Induced current on a 2λ circular cylinder. Plane wave is polarized in E_z and is incident from $\theta^i = 0^o$. Frequency=300 MHz.

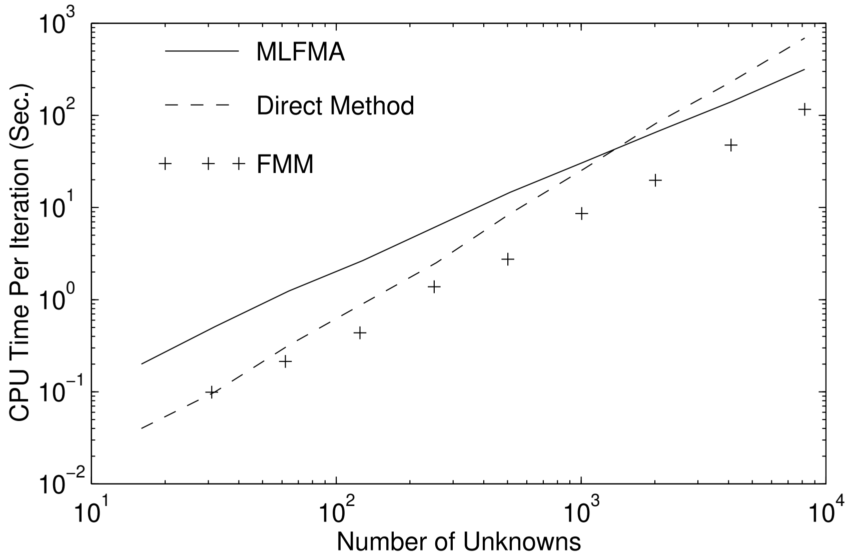

Figure 4 CPU time per iteration vs number of unknowns.