AN EFFICIENT HIGH ORDER MULTILEVEL FAST MULTIPOLE ALGORITHM FOR ELECTROMAGNETIC SCATTERING ANALYSIS

X.-M. Pan[1, 2, *], L. Cai[1], and X.-Q. Sheng[1]

[1]Center for Electromagnetic Simulation, School of Information and Electronics, Beijing Institute of Technology, 100081, P. R. China

[2]The ElectroScience Laboratory, The Ohio State University, 1320 Kinnear Road, Columbus, Ohio, OH 43212, USA

Abstract—An efficient higher order MLFMA is developed by using an "extended-tree". With this extended-tree, the size of the box at the finest level is reduced, and the cost associated with the aggregation and disaggregation operations is significantly decreased. The sparse approximate inverse (SAI) preconditioner is utilized to accelerate the convergence of iterative solutions. The Cholesky factorization, instead of the often used QR factorization, is employed to construct the SAI preconditioner for cavity scattering analysis, by taking advantage of the symmetry of the matrix arising from electric field integral equation. Numerical experiments show that the higher order MLFMA is more efficient than its low-order counterpart.

1. INTRODUCTION

Among many full-wave numerical methods, the algorithms based on the method of moments (MoM) [1] have been widely used due to its high fidelity and superior capability to handle arbitrary shaped targets. A typical MoM solution procedure begins with generating a proper mesh for the target of interest and selecting basis functions to model the equivalent electric and magnetic currents. After modeling a target with a set of N expansion functions and performing the traditional Galerkin testing for the integral equations, a $N \times N$ dense impedance matrix is generated. The typical basis functions which are selected for discretizing the target are the RWG [2]; in order to achieve accurate solutions with RWG the average size of each element

Received 2 February 2012, Accepted 22 February 2012, Scheduled 8 March 2012

[*] Corresponding author: Xiao-Min Pan (xmpan@bit.edu.cn).

is typically on the order of 1/10 wavelength (λ). Consequently, the size of the associated MoM matrix grows very rapid as the object size becomes larger with respect to wavelength; this challenges the MoM for a variety of applications. In addition, it is very costly to obtain high accuracy by using RWG basis functions, because they exhibit a low-order convergence rate — the solution accuracy increases slowly with the number of unknowns. A remedy is to employ higher order basis functions.

The development of higher order basis functions for modeling electromagnetic fields has received intense attention recently because of their faster convergence, permitting more accurate results with less efforts than the low-order basis functions. Different kinds of high order basis functions were studied comprehensively in [3–11]. Their efficiency and reliability have already been approved. Higher order electromagnetic modeling is definitely becoming an attractive approach of activity in computational electromagnetics. It is well known that the matrix system arising from MoM is a full matrix. Direct solvers, such as LU, often fail in providing a solution to engineering problems because of limited computational resources. This is the reason why iterative solvers rather than direct ones are always employed in the MoM, especially for applications in real-life where the targets are generally large. To accelerate the matrix-vector multiplication (MVM) in the iterative solution, a higher order multilevel fast multipole algorithm (MLFMA) was developed in [4, 5]. However, implementing higher order discretizations to reduce the number of unknowns and to decrease the computational complexity are somewhat contradictory. Namely, since the number of levels in the MLFMA is determined by the size of elements in the model, the use of large elements associated with the use of higher order basis functions, implies the reduction of the number of levels and thus limited performance of the higher order MLFMA in conjunction with Galerkin-type solutions. As a result, the operations on aggregation and disaggregation become expensive. A solution is to implement the MLFMA based on point-to-point interactions, instead of the traditional basis-to-basis interactions [4].

At the same time, an effective preconditioner is always helpful or even unavoidable for the iterative solution of the higher order MLFMA. If the shape of the target is simple, the block diagonal (BD) preconditioner [12, 13] can perform well. But when the target becomes complex in shape, or when the target involves open structure and electric field integral equation (EFIE) must be used, the resultant MoM matrix would be very ill-conditioned. The BD preconditioner loses its effectiveness. Therefore, a more effective and robust preconditioner is unavoidable. Among many developed preconditioning techniques, the sparse approximate inverse (SAI) preconditioner is one of the most effective and robust [14, 15]. Compared with the LU/ILU typed preconditioners [16, 17], the SAI preconditioner represents an inherent parallelism; incorporating it into the well developed parallel MLFMA is easy. However, an effective SAI preconditioner is always expensive in constructing, especially when the MoM matrix system is extremely ill-conditioned and many nonzero entries are required in the SAI preconditioner.

In this paper, we propose an alternative approach to implement the higher order MLFMA. Our approach creates so-called "extended levels" below the finest level of the MLFMA-tree constructed in the traditional basis-to-basis implementation. The resultant tree structure is denoted as "extended-tree", which has more levels than the original one. The aggregation and disaggregation matrices are assembled based upon the finest level in the extended-tree. Therefore, the finest extended-tree level is where the aggregation operation begins from while the disaggregation operation ends at. Meanwhile, no translation is carried out at extended levels. Other than improving the efficiency of the aggregation/disaggregation operations, the extended-tree approach makes preconditioners in the higher order MLFMA more effective and efficient. In particular, the sparse approximate inverse (SAI) preconditioner based on the Cholesky factorization is developed for analysis on large cavities. Numerical experiments are conducted to investigate the performance of the higher order MLFMA. The effectiveness of the SAI preconditioner is validated by computing the radar cross section (RCS) of a large cavity.

2. FORMULATION OF THE HIGHER ORDER MLFMA

2.1. Higher Order Basis Functions

In this paper, the higher order basis functions are divergence-conforming interpolatory vector basis functions on generalized triangles. The p-order functions can be seen as the product of 0-order RWG functions and p-order interpolatory polynomials with specially arranged arrays of interpolation points.

A generalized triangle is commonly represented in terms of simplex coordinates, ξ_1, ξ_2 and ξ_3. These basis functions are based on the 0-order RWG functions. As shown in Figure 1, the RWG associated with edge 1 has the form of [6]

$$\Lambda_1(\xi_1, \xi_2, \xi_3) = \frac{1}{\Im}(\xi_2 l_3 - \xi_3 l_2) \tag{1}$$

where $\Im$ is the Jacobian, and l_1, l_2 and l_3 represent the edge vector

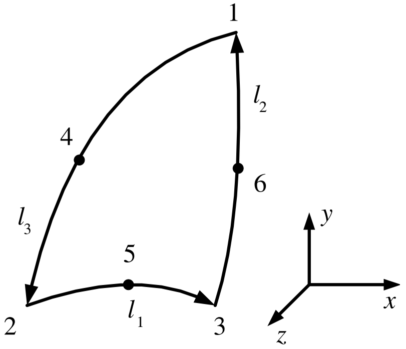


Figure 1. A curvilinear triangle patch.

opposite to the vertex 1, 2 and 3, respectively. Analogous expressions hold for the functions for the other two edges. The higher order basis functions are obtained by forming the cross product of the RWG basis functions with a set of polynomial functions, which are complete to the specified order. The basis function of order p for edge 1 corresponding to the interpolation node (i, j, k) is given by [6]

$$\begin{aligned}&\Lambda_{ijk}^{(1)}(\xi_1,\xi_2,\xi_3) = C_{ijk}^{(1)} R_i^{p+2}(\xi_1)\hat{R}_j^{p+2}(\xi_2)\hat{R}_k^{p+2}(\xi_3)\Lambda_1(\xi_1,\xi_2,\xi_3)\\&0 \le \xi_1,\xi_2,\xi_3 \le 1; \quad i = 0,1,\ldots,p; \quad j,k = 1,2,\ldots,p+1;\\&i+j+k = p+2\end{aligned} \tag{2}$$

where $C_{ijk}^{(1)}$ is the normalization factor chosen to make the normal component of $\Lambda_{ijk}^{(1)}$ unity along edge 1, and $\hat{R}_m^K$ are shifted Silvester-Lagrange interpolation polynomials in the form of

$$\hat{R}_m^K(\xi) = \begin{cases} \frac{1}{(m-1)!}\prod\limits_{i=1}^{m-1}(K\xi - i), & 2 \le m \le K+1\\ 1, & m = 1\end{cases} \tag{3}$$

and $R_m^K(\xi)$ are the Silvester-Lagrange interpolation polynomials in the form of

$$R_m^K(\xi) = \begin{cases} \frac{1}{m!}\prod\limits_{i=0}^{m-1}(K\xi - i), & 1 \le m \le K\\ 1, & m = 0\end{cases} \tag{4}$$

The interpolatory higher order basis functions, in general, have better orthogonality properties than its hierarchical counterparts [6]. In addition, there is a direct physical interpretation of every current- or field-distribution coefficient in the model, since only one basis function is nonzero at every interpolation point.

2.2. MLFMA

For perfectly electric conducting (PEC) objects, discretization and testing of surface integral equations yield a $N \times N$ dense matrix equation in the form of

$$\mathbf{Z} \cdot \mathbf{J} = \mathbf{E}^i, \tag{5}$$

where $\mathbf{Z}$ is the impedance matrix, N the number of unknowns, and $\mathbf{J}$ the coefficients of equivalent current. $\mathbf{E}^i$ corresponds to the discretized incident wave. The matrix Equation (5) can be solved iteratively, and the required MVM can be accelerated by the FMM or MLFMA [13]. The FMM/MLFMA decomposes MVM into two parts: near-field interactions (NFI) and far-field interactions (FFI). The former is computed directly, while the latter is accelerated by FMM/MLFMA. The matrix equation in the context of FMM has a form of

$$\mathbf{Z} \cdot \mathbf{J} = \sum_{p \in B_q} \mathbf{Z}_{qp} \cdot \mathbf{J}_p + \mathbf{V}_q \sum_{p \notin B_q} \alpha_{qp} \cdot \mathbf{V}_p^* \cdot \mathbf{J}_p, \tag{6}$$

where $\mathbf{J}_p$ is the coefficients of the RWG basis functions in the box p, $\mathbf{Z}_{qp}$ the impedance matrix corresponding to the observation box q and source box p, B_q the near neighbors of the box q, α_{qp} the translator, and $\mathbf{V}_q$ and $\mathbf{V}_p^*$ the disaggregation and aggregation matrix, respectively. The disaggregation represents a symmetric operation of the aggregation.

The first term in the right hand side of (6) accounts for the contribution from the self-coupling of box q and its near neighbors. While the second one collects the contribution from the rest boxes. To conduct FFI by the MLFMA, a hierarchical tree structure (MLFMA-tree) is always constructed by a recursive subdivision of the spatial domain. All computations in the FMM/MLFMA are organized by boxes in the MLFMA-tree. FFI in the FMM is realized through three stages: the aggregation, the translation and the disaggregation. In the MLFMA, the interpolation/anterpolation combined with center-shifting operations is required to transfer far-field patterns (FFPs) from a child-box to the parent-box, and vice versa [12, 13, 18]. The operations associated with aggregation can be written as

$$\mathbf{V}_{l-1,p}^*(\hat{\mathbf{k}}_{l-1,n'}) = e^{-j\hat{\mathbf{k}}_{l-1,n} \cdot \mathbf{r}_{pp'}} \sum_{n=1}^{K_l} \mathbf{V}_{l,p}^*(\hat{\mathbf{k}}_{l,n}) W_{n',n}^{l-1,l} \tag{7}$$

where K_l is the number of FFPs for boxes at level l; $\mathbf{r}_{pp'}$ denotes the distance vector between centers of box p and its parent p'; $\mathbf{V}_{l-1,p}^*(\hat{\mathbf{k}}_{l-1,n'})$ and $\mathbf{V}_{l,p}^*(\hat{\mathbf{k}}_{l,n})$ are aggregation matrices for level $(l-1)$ and l; $\hat{\mathbf{k}}_{l-1,n'}$ and $\hat{\mathbf{k}}_{l,n}$ are the directions of FFPs; n' and n are positive integers; $W_{n',n}^{l-1,l}$ is the interpolation matrix.

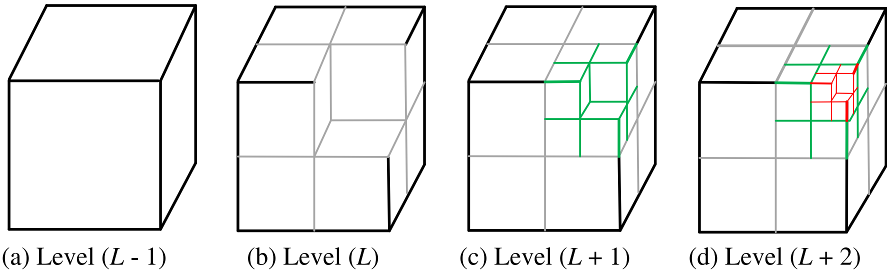


Figure 2. Extended-tree structure (the $(L+1)$-th and $(L+2)$-th level are extended levels).

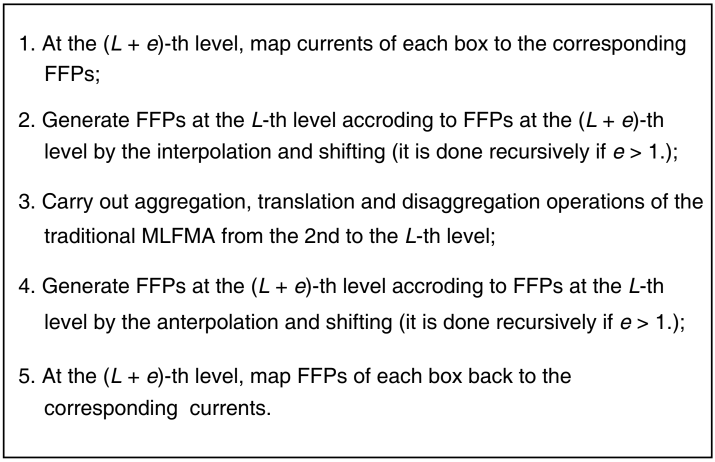
3. EXTENDED-TREE AND SAI PRECONDITIONER

3.1. Extended-tree

In the MLFMA, truncation number at the l-th level is given by [13]

$$L_l = kd_l + 1.8d_\varepsilon^{2/3}(kd_l)^{1/3} \tag{8}$$

where k is wave-number, d_l the diameter of the box size at the l-th level, and d_ε (equal to $\log(1/\varepsilon)$) the number of digits of accuracy ε. The number of FFPs K_l for boxes at l-th level is equal to $(2L_l^2 + 4)$. Obviously, K_l increases rapidly as the box size grows. In the Galerkin-based MoM using the RWG basis function, the box at the finest MLFMA-tree is about a quarter of a wavelength since the edge length is about $\lambda/10$. However, the number of levels in the higher order MLFMA is at least one or two less than that in the low-order MLFMA because of the large patch size. The efficiency of the MLFMA for the higher order MoM tends to decrease as the memory for aggregation and disaggregation matrices increase. To overcome this problem, the MLFMA based on point-to-point interactions was implemented instead of the traditional basis-to-basis interactions in [4]. Since the Gaussian quadrature is used to calculate the matrix elements when the testing and source bases are not close to each other, the corresponding interactions were replaced by interactions among points associated with the Gaussian quadrature. The point-to-point implementation utilized the MLFMA to calculate electromagnetic fields at these Gauss quadrature points. Thus the number of levels used is not limited by the size of patches, making MLFMA more efficient. At the same time, the near interaction part of the MoM matrix is redefined as the difference between the original matrix and the interactions calculated by MLFMA.



1. At the (L + e)-th level, map currents of each box to the corresponding FFPs;
2. Generate FFPs at the L-th level accroding to FFPs at the (L + e)-th level by the interpolation and shifting (it is done recursively if e > 1.);
3. Carry out aggregation, translation and disaggregation operations of the traditional MLFMA from the 2nd to the L-th level;
4. Generate FFPs at the (L + e)-th level accroding to FFPs at the L-th level by the anterpolation and shifting (it is done recursively if e > 1.);
5. At the (L + e)-th level, map FFPs of each box back to the corresponding currents.

Figure 3. Implementation of the MLFMA with extended-tree.

In this paper, another approach is proposed by using the so-called "extended-tree". Figure 2 shows a portion of such an extended-tree, where the $(L + 1)$-th and $(L + 2)$-th level are extended levels. The original MLFMA-tree is based on the traditional basis-to-basis implementation. The extended-tree is obtained by recursively dividing the finest boxes of the original tree until the box size at the finest extended level reaches about 0.25λ.

In general, suppose the L-th level is the finest level at an original tree and the $(L + e)$-th level is the finest extended level with e denoting the number of extended levels. The MLFMA with extended-tree can be described in Figure 3. As described in this figure, the aggregation and disaggregation operations are performed by starting from and ending at the finest extended level. The translation remains unchanged and is only carried out at the original MLFMA-tree levels. Thus the memory can be reduced significantly for the aggregation/disaggregation matrices. If $e > 1$, operations associated with step 2 and 4 can be implemented in a hierarchical manner similar to the aggregation/disaggregation operations in the MLFMA.

The extended-tree is constructed in a way that the one-buffer-box criterion [12, 13] is strictly guaranteed, making the proposed higher order MLFMA error-controllable. As depicted in Figure 3, no translations are done on those auxiliary extended levels. Compared with traditional MLFMA, additional errors in the proposed scheme are those arising from the interpolation/anterpolation operations from

$(L + e)$-th to L-th levels. Mathematically speaking, those errors are controllable, which can be reduced by increasing the number of interpolation points [13]. As consequence, there is no practical constraint on how far the patches can stick out of the boxes at the extended levels, which in turn makes the scheme error-controllable.

As is known, the NFI matrix is filled according to NFI lists [13, 19]. The lists can be set up either based on the original tree or based on the extended-tree in our proposed MLFMA. We suggest the former because it admits a larger spacious domain to be treated by the NFI computations. Compared with the approach to calculate NFI matrix based on the extended-tree, our recommendation requires more NFI matrix entries to be filled. However, the additional cost is not an issue in the higher order modeling since the number of unknowns in the finest boxes at the original tree is generally less than 50. In fact, memory for NFI matrix can still be cut down greatly in our proposed higher order MLFMA compared with that in the low-order MLFMA, as will be shown by the numerical experiments. Conversely, a more effective preconditioner can be constructed if the NFI lists are based on the original MLFMA-tree. This is sometimes of vital importance, especially when the target becomes complex in shape, or when the target involves open structure and EFIE must be used. In our higher order MLFMA, the robust SAI preconditioner is utilized. To take advantage of the symmetry in the matrix arising from EFIE, we have developed an efficient approach to construct the SAI preconditioner.

3.2. SAI Preconditioner

It is known that the preconditioner sparse matrix $\mathbf{M}$ for $\mathbf{A}$ is usually computed by minimizing the Frobenius norm $\|\mathbf{I} - \mathbf{MA}\|_F^2$, where $\mathbf{I}$ is the identity matrix, $\mathbf{M}$ is constrained by the certain sparsity pattern $\mathbf{S}$. The Frobenius norm is usually chosen since it decouples the constrained minimization problem into independent linear LS problems [14, 15]:

$$\min_{\mathbf{M}\in\mathbf{S}} \|\mathbf{I} - \mathbf{MA}\|_F^2 = \sum_{i=1}^{N} \min_{\mathbf{M}_i\in\mathbf{S}_i} \|\mathbf{e}_i - \mathbf{m}_i\mathbf{A}\|_F^2 \tag{9}$$

where $\mathbf{e}_i$ and $\mathbf{m}_i$ are the row vectors of the matrices $\mathbf{I}$ and $\mathbf{M}$. Thus, each $\mathbf{m}_i$ can be solved independently. To solve (9), the QR factorization is generally carried out for a very small matrix $\mathbf{A}_i$ reduced from $\mathbf{A}$ based on the sparsity pattern $\mathbf{S}_i$ corresponding to the i-th row of $\mathbf{S}$, then $\mathbf{m}_i$ is computed from the obtained QR factorization.

When $\mathbf{A}$ is symmetric, Cholesky factorization is more efficient than QR factorization to solve (9) [20]. Suppose t denote the number of nonzeros in row $\mathbf{m}_i$ of $\mathbf{M}$. Thus, the number of floating-points operations required by the QR factorization for the row $\mathbf{m}_i$ is at least $4t^3/3$, while the number of operations required by the Cholesky factorization is $t^3/3$. The latter is at least 4 times faster than the former in constructing a SAI preconditioner.

4. NUMERICAL RESULTS

To validate our proposed higher order MLFMA, numerical experiments, based on the sequential implementation of the algorithm, are carried out on an IBM sever with one Xeon 3.0 GHz CPU and 16.0 GB memory. Our code utilizes GMRES as the iterative solver, where the iteration process is terminated when the 2-norm of the residual vector is reduced to 10^{-3}. In all computations, the largest box enclosing the targets is indexed by the 0-th level.

4.1. Validations on the Extended-tree

Two examples, namely, a sphere and an airplane model, are presented to demonstrate the accuracy and efficiency of our proposed extended-tree scheme.

Firstly, the scattering from a perfect electrical conducting (PEC) sphere of 40λ diameter is calculated. The sphere requires 1,116,300 and 217,150 unknowns for RWG and the first-order interpolatory basis functions to obtain accurate results. A 7-level tree is used in the low-order (RWG) MLFMA. And a 6-level tree is constructed for the higher order MLFMA based on the basis-to-basis implementation. The tree is extended to a 7-level one in our proposed scheme. The finest boxes at the 5th and 6th level are 0.62λ and 0.31λ in size,

Table 1. Statistics on computational resources for the sphere.

		Case1	Case2	Case3
Levels of Tree		7	6	7
Time Filling Matrix (s)	NFI	2781	2216	2224
	Aggregation	29	49	18
Iteration Time (s)		653	535	567
Memory for Matrix (MB)	NFI	1960	298	298
	Aggregation	1260	669	245
Total Memory (MB)		4290	1364	940

respectively. The statistics on computational resources are listed in Table 1 for the three cases: 1) RWG; 2) higher order with original tree; 3) higher order with extended-tree. It is shown that the memory for the aggregation/disaggregation matrices are reduced from 670 MB to 245 MB as the truncation number is decreased from 10 to 6. Figure 4 presents the RCS results with and without the extended level. Compared with Mie series, the root mean square errors in these three cases are 0.21 dB, 0.18 dB and 0.19 dB. It also worth noting that memory for the NFI matrix is significantly cut down from 1960 MB in the RWG MLFMA computation to 298 MB in the higher order MLFMA (both with and without extended levels) although NFI lists are constructed according to the original tree. Our proposed scheme takes good advantage of the higher order basis functions.

The second example is a large airplane model with a 96λ long fuselage. It requires 1,143,057 and 220,510 unknowns for the RWG and the first-order basis functions to obtain accurate results. A 9-level tree is used in the low-order MLFMA. An 8-level tree is constructed for the traditional higher order MLFMA, while it is extended to a 9-level one in our proposed approach. The sizes of finest boxes at the 7th and 8th level are 0.75λ and 0.375λ, respectively. The statistics on computational resources are listed in Table 2 for the three cases: 1) RWG; 2) higher order with original tree; 3) higher order with extended-tree. It is shown that the memory for the aggregation/disaggregation matrices are reduced from 976 MB to 437 MB as the truncation number is decreased from 12 to 8. At the same time, memory for the NFI matrix is cut down from 3113 MB to 534 MB. Figure 5 presents the RCS results with and without the extended level.

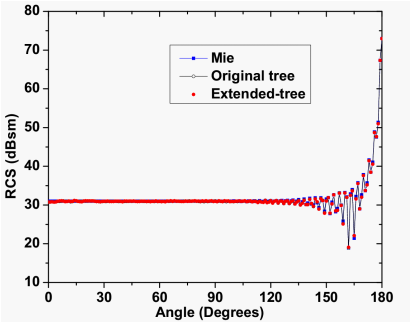


Figure 4. Bistatic RCS from a conducting sphere.

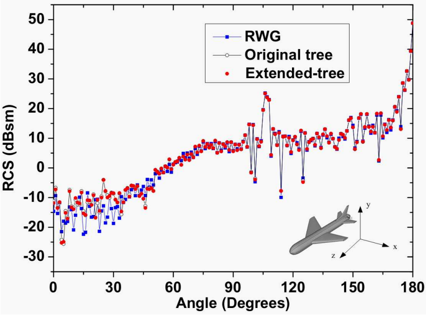


Figure 5. Bistatic RCS from the airplane model.

Table 2. Statistics on computational resources for the airplane.

		Case1	Case2	Case3
Levels of Tree		9	8	9
Time Filling Matrix (s)	NFI	3447	3314	3317
	Aggregation	17	71	32
Iteration Time (s)		1600	1951	2091
Memory for Matrix (MB)	NFI	3113	534	534
	Aggregation	2268	976	437
Total Memory (MB)		6432	1899	1361

Table 3. Performane of the BD preconditioner on the sphere and airplane.

		Iteration Counts		Iteration Time (s)	
		Sphere	Airplane	Sphere	Airplane
Nopre	RWG	44	113	653	1600
	Higher order	57	240	567	2091
BD	RWG	31	33	478	481
	Higher order	24	34	247	312

4.2. Validations on the Preconditioners

In our higher order MLFMA, the NFI lists are based on the original tree instead of the proposed extended-tree. This strategy permits effective preconditioners for the iterative solutions. In the following, effectiveness of the BD and SAI preconditioners for our higher order MLFMA is studied.

4.2.1. The BD Preconditioner

The sphere and the airplane examples in the Section 4.1 are used to show the effectiveness of our strategy. Table 3 presents the iteration counts with and without the BD preconditioner. It can be seen from this table that the cases with BD preconditioner converge quite faster than those without preconditioner. Additionally, a higher acceleration rate is obtained in the higher order MLFMA than in the RWG counterpart. The reason lies in that the NFI lists in our implementation is based on the original MLFMA-tree. More information is reserved to construct the preconditioner in our higher order MLFMA than in the low-order MLFMA. It should be noted that the shape of the above two targets are relatively simple and the resultant MoM matrices are well-conditioned to some extent. If the MoM matrix becomes very ill-conditioned, the BD preconditioner will lose its effectiveness as shown in the following cavity calculations.

4.2.2. The SAI Preconditioner

Scattering from a cylindrical cavity is computed to demonstrate the effectiveness of the SAI preconditioner in the higher order MLFMA. The cavity is 5λ in radius and 12λ in height. It requires 157,684 and 32,890 unknowns for the RWG and first-order basis functions, respectively. A 6-level tree is used in the low-order MLFMA. In the higher order calculation, the MLFMA tree is extended from 5 levels to 6 levels. The sizes of boxes at the 4th and 5th level are 0.75λ and 0.375λ, respectively. For this target, the BD preconditioned systems fail to converge for both RWG and higher order cases and the SAI preconditioner is necessary. To study the convergence, four cases are investigated: 1) RWG; 2) RWG with SAI; 3) higher order; 4) higher order with SAI. The sparsity pattern of the SAI preconditioner is obtained by filtering off 20% entries of $\sum\limits_{p\in B_q} \mathbf{Z}_{qp}$ with small absolute values.

The $\theta\theta$-polarized (the cavity is opened along positive z-axis) monostatic RCS is presented in Figure 6. The convergence histories for different cases are presented in Figure 7 with the (0°, 0°) incident. Figure 8 presents the iteration counts for all the 31 incident angles. The iteration is accelerated by setting the solution of the previous incident angle as the initial guess of the next angle. The computational statistic is listed in Table 4. The iteration time is obtained by adding up the time for all 31 incident angles together.

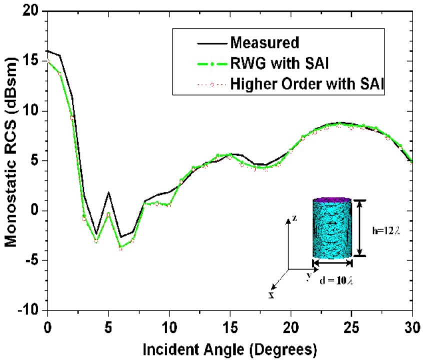


Figure 6. Monostatic RCS of the cylindrical cavity.

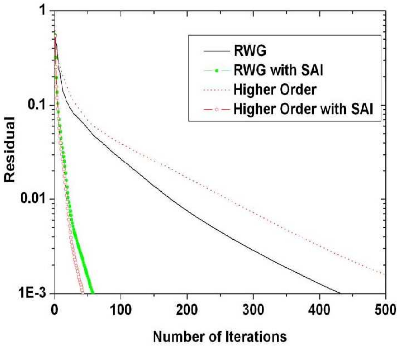


Figure 7. Convergence history.

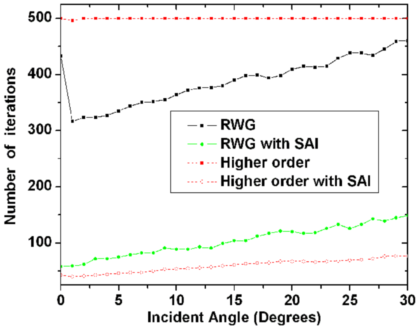


Figure 8. Iteration counts for different incident angles.

Table 4. Statistics on computational resources for the cavity.

		Case1	Case2	Case3	Case4
Levels of Tree		6	6	6	6
Time Filling Matrix (s)	NFI	637	637	358	358
	SAI	–	845	–	113
Iteration Time (s)		15053	4842	8423	1088
Memory for Matrix (MB)	NFI	880	880	156	156
	SAI	–	683	–	124
Total Memory (MB)		1316	1999	257	381

It can be found that the higher order MLFMA computation converges a little more slowly than that of the RWG based MLFMA when no preconditioner is employed. However, after the SAI preconditioner is applied, the higher order computation converges much faster than the RWG one although the same filtering parameter, to say 20%, is used to construct the SAI. The faster convergence results from our strategy to fill the NFI matrix. As is known, coupling between well-separated regions may be still very strong because of the multiple reflections in the cavity computations. The larger the finest box is, the more effective a preconditioner can be obtained. In our computations, the finest box in the proposed higher order MLFMA is twice larger than that in the RWG case. Consequently, a much more effective SAI preconditioner can be constructed.

At the same time, our experiments indicate that constructing a SAI preconditioner becomes much cheaper in the higher order computation than in the low-order case. As shown in Table 4, more than 800 s are used to construct the SAI preconditioner in the RWG case. However, it is decreased to 113 s by a factor of about 8.0. This is because the higher order MLFMA has a much smaller NFI matrix than the low-order one does. This experiment also implies that it is impractical to obtain an effective SAI preconditioner by increasing the finest box size in the low-order MLFMA because of the high construction cost of the SAI preconditioner.

5. CONCLUSIONS

An efficient higher order MLFMA is proposed by constructing the so-called "extended-tree". Memory required by the aggregation/disaggregation matrix is thus reduced a lot. In the proposed higher order MLFMA, the near-field interaction lists are still defined in terms of the original tree. This makes it possible to construct an effective preconditioner. The sparse approximate inverse (SAI) preconditioner is employed to accelerate the iterative solution where the block diagonal preconditioner fails. To take advantage of symmetry of the matrix arising from electric field integral equation, the Cholesky factorization, instead of the QR factorization, is used to accelerate the construction of the SAI preconditioner. Numerical experiments show that the proposed higher order MLFMA is much more efficient than its low-order counterpart. Calculations on the large cavity reveal that the proposed SAI preconditioner can accelerate the convergence significantly.

ACKNOWLEDGMENT

This work was supported by the NSFC under Grants 10832002, 60901005 and by the Excellent Young Scholars Research Fund of Beijing Institute of Technology under Grant 2010YS0502.

REFERENCES

1. Peterson, A. F., S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*, IEEE Press, Piscataway, NJ, 1998.
2. Rao, S. M., D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, Vol. 30, No. 3, 409–418, May 1982.

3. Kolundžija, B. M. and B. D. Popovic, "Entire-domain Galerkin method for analysis of metallic antennas and scatterers," *IEE Proceedings — H*, Vol. 140, No. 1, 1–10, Feb. 1993.
4. Donepudi, K. C., J. M. Song, J. M. Jin, G. Kang, and W. C. Chew, "A novel implementation of multilevel fast multipole algorithm for higher order Galerkin's method," *IEEE Trans. Antennas Propag.*, Vol. 48, 1192–1197, Aug. 2000.
5. Donepudi, K. C., 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, 1069–1078, Jul. 2001.
6. Notaros, B. M., "Higher order frequency-domain computational electromagnetics," *IEEE Trans. Antennas Propag.*, Vol. 56, No. 8, 2251–2276, Aug. 2008.
7. Liu, Z.-L. and J. Yang, "Analysis of electromagnetic scattering with higher-order moment method and NURBS model," *Progress In Electromagnetics Research*, Vol. 96, 83–100, 2009.
8. Eibert, T. F., Ismatullah, E. Kaliyaperumal, and C. H. Schmidt, "Inverse equivalent surface current method with hierarchical higher order basis functions, full probe correction and multilevel fast multipole acceleration," *Progress In Electromagnetics Research*, Vol. 106, 377–394, 2010.
9. Zhang, H.-W., X.-W. Zhao, Y. Zhang, D. Garcia-Donoro, W.-X. Zhao, and C.-H. Liang, "Analysis of a large scale narrow-wall slotted waveguide array by parallel MoM out-of-core solver using the higher order basis functions," *Journal of Electromagnetic Waves and Applications*, Vol. 24, Nos. 14–15, 1953–1965, 2010.
10. Lai, B., H.-B. Yuan, and C.-H. Liang, "Analysis of nurbs surfaces modeled geometries with higher-order MoM based aim," *Journal of Electromagnetic Waves and Applications*, Vol. 25, Nos. 5–6, 683–691, 2011.
11. Klopf, E. M., S. B. Manić, M. M. Ilic, and B. M. Notaros, "Efficient time-domain analysis of waveguide discontinuities using higher order FEM in frequency domain," *Progress In Electromagnetics Research*, Vol. 120, 215–234, 2011.
12. Song, J. M., C. C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, Vol. 45, No. 10, 1488–1493, Oct. 1997.
13. Chew, W. C., J. M. Jin, E. Michielssen, and J. Song, *Fast Efficient Algorithms in Computational Electromagnetics*, Artech House, Boston, MA, 2001.

14. Lee, J., J. Zhang, and C. C. Lu, "Sparse inverse preconditioning of multilevel fast multipole algorithm for hybrid integral equations in electromagnetics," *IEEE Trans. Antennas Propag.*, Vol. 52, No. 9, 2277–2287, Sep. 2004.
15. Carpentieri, B., I. S. Duff, L. Griud, and G. Sylvand, "Combing fast multipole techniques and an approximate inverse preconditioner for large electromagnetism calculations," *SIAM J. Sci. Comput.*, Vol. 27, No. 3, 774–792, 2005.
16. Lee, J., J. Zhang, and C. C. Lu, "Incomplete LU preconditioning for large scale dense complex linear systems from electromagnetic wave scattering problems," *J. Comput. Phys.*, Vol. 185, 158–175, 2003.
17. Malas, T. and L. Guirel, "Incomplete LU preconditioning with the multilevel fast multipole algorithm for electromagnetic scattering," *SIAM J. Sci. Comput.*, Vol. 29, No. 4, 1476–1494, 2007.
18. Pan, X.-M. and X.-Q. Sheng, "A highly efficient parallel approach of multi-level fast multipole algorithm," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 8, 1081–1092, 2006.
19. Pan, X. M. and X. Q. Sheng, "A sophisticated parallel MLFMA for scattering by extremely large targets," *IEEE Antennas Propag. Mag.*, Vol. 50, No. 3, 129–138, Jun. 2008.
20. Chow, E., "Parallel implementation and practical use of sparse approximate inverse preconditioners with a priori sparsity patterns," *Intl. J. High Perf. Comput. Appl.*, Vol. 15, No. 1, 56–74, Feb. 2001.