

# An Efficient Parallel Multilevel Fast Multipole Algorithm for Large-scale Scattering Problems

Hu Fangjing, Nie Zaiping, Hu Jun

School of Electronic Engineering,  
University of Electronic Science and Technology of China  
Chengdu 610054, PRC  
hfj1010@gmail.com

**Abstract**—In this paper, we present an efficient parallel multilevel fast multipole algorithm (MLFMA) for three dimensional scattering problems of large-scale objects. Several parallel implantation tricks are discussed and analyzed. Firstly, we propose a method that reduces truncation number without loss of accuracy. Furthermore, a matrix-sliced technique, allowing data in the memory transforming into the hard disk, is applied here, in order to solve the problem of extremely large targets. Finally, a transition level scheme is adopted to improve the parallel efficiency. We demonstrate the capability of our code by considering a sphere of  $220\lambda$  discretized with 48,879,411 unknowns and a square patch of  $200\lambda$  discretized with 10,150,143 unknowns. The bi-static RCS is calculated within 41.5 GB memory for the first object and 14.7 GB for the second one.

**Index terms**—parallel algorithm, RCS calculation, multilevel fast multipole algorithm, electromagnetic scattering.

## I. INTRODUCTION

Integral equation methods are widely used for solving electromagnetic scattering problems, and the multilevel fast multipole algorithm (MLFMA) has established itself as one of the most powerful among the different acceleration methods [1]. However, for many real-life problems, the discretization of these large-scale targets lead to millions of unknowns. The maximum size that can be solved is limited even with modern computers. Thus, it is necessary to develop an efficient parallel algorithm in order to solve these very large-scale problems.

Of the various parallelization schemes for MLFMA, the most popular is the distributed memory architectures by constructing clusters of

computers with local memories connected via fast networks [1]-[5]. However, the parallel implementation of MLFMA is not trivial, owing to the complicated structure of this algorithm. Without careful parallel schemes, the algorithm may fail to produce accurate results. Thus, a series of implementation tricks have been developed for the efficient parallelization of MLFMA in [2]-[5]. But even with these implementations, the algorithm has to face memory-hungry problem for many extremely large problems.

In this paper, we present an efficient parallel MLFMA algorithm that integrating a series of implementation tricks proposed in [2]-[4]. In particular, a novel trick for reducing the truncation number is presented and the technique, that slices the matrix data and save it to the hard disk, is applied in our code, in order to optimize the memory usage and solve the memory-hungry problem. To demonstrate the capability of our parallel MLFMA code, the bi-static RCS of a sphere with a diameter of  $220\lambda$ , containing more than 50 million unknowns, and a patch of  $200\lambda$ , containing about 10 million unknowns, are successfully solved.

## II. PARALLEL IMPLEMENTATION OF MLFMA

A series of implementation tricks for parallel MLFMA are developed in [2]-[4], most of which focus on memory optimization. We can say that reducing the RAM requirement can never be over-emphasized. In this section, several tricks that integrating in our code will be introduced and analyzed.

### A. Integral Equation Formulation

In this section, we consider the scattering of electromagnetic waves from perfectly conducting objects.

For a perfectly conducting object, the well-known electric-field integral equation (EFIE) can be written as [2]

$$\begin{aligned} \frac{ik\eta}{4\pi} \hat{t} \cdot \int_S \bar{G}(\bar{r}, \bar{r}') \cdot \bar{J}(\bar{r}') dS' \\ = -\hat{t} \cdot \bar{E}^i(\bar{r}) \quad r \in S \end{aligned} \quad (1)$$

with

$$\begin{aligned} \bar{G}(\bar{r}, \bar{r}') &= \left[ \bar{I} - \frac{1}{k^2} \nabla \nabla' \right] g(\bar{r}, \bar{r}') \\ &= \left[ \bar{I} - \frac{1}{k^2} \nabla \nabla' \right] \frac{e^{ik|\bar{r}-\bar{r}'|}}{|\bar{r}-\bar{r}'|} \end{aligned}$$

In Equation (1),  $\eta$  is the impedance of free space,  $S$  is the surface boundary of the scatterer, and  $\hat{t}$  is the unit tangent vector at any given point on  $S$ . Furthermore,  $\bar{J}$  is the unknown surface electric current,  $\bar{E}^i$  is the incident electric-field vector, and  $\bar{I}$  is the unit dyad.

If the surface of the object is closed, it can also be described using the magnetic-field integral equation (MFIE)

$$\begin{aligned} -\hat{t} \cdot \frac{\bar{J}(\bar{r})}{2} + \frac{1}{4\pi} \hat{t} \cdot \bar{n} \times \nabla \times \int_S g(\bar{r}, \bar{r}') \cdot \bar{J}(\bar{r}') dS' \\ = -\hat{t} \cdot \bar{n} \times \bar{H}^i(\bar{r}) \end{aligned} \quad (2)$$

where  $\bar{n}$  is the unit normal vector, and  $\bar{H}^i$  is the incident magnetic-field vector.

However, both EFIE and MFIE suffer from nonunique solutions at resonant frequencies. To alleviate this problem, for a surface-closed target, we used the combined-field integral equation (CFIE) which is defined by the relation

$$\alpha \text{ EFIE} + \eta(1-\alpha) \text{ MFIE} \quad (3)$$

where  $\alpha \in [0,1]$  is called the combination coefficient.

In order to solve these equations numerically, we should model the surface with flat triangular patches and expand the current in term of RWG basis functions [7]. Applying Galerkin's method, the integral equation is then reduced to a system of linear equations. The matrix element of EFIE

and MFIE is given by

$$Z_{mn}^E = \frac{ik\eta}{4\pi} \int_S dS f_m(\bar{r}) \cdot \int_S \bar{G}(\bar{r}, \bar{r}') \cdot f_n(\bar{r}') dS' \quad (4)$$

$$\begin{aligned} Z_{mn}^M &= - \int_S dS f_m(\bar{r}) \cdot \frac{f_n(\bar{r}')}{2} \\ &+ \frac{1}{4\pi} \int_S dS f_m(\bar{r}') \cdot \hat{n} \times \nabla \times \int_S g(\bar{r}, \bar{r}') \cdot f_n(\bar{r}') dS' \end{aligned} \quad (5)$$

where  $f_n(\bar{r})$  is the  $n$ th RWG basis function.

The matrix element corresponding to CFIE can then be derived as  $Z_{mn} = \alpha Z_{mn}^E + \eta(1-\alpha) Z_{mn}^M$ . Then the integral equations reduce to the matrix equation

$$[Z]_{N \times N} \bullet [J]_{N \times 1} = [V]_{N \times 1} \quad (6)$$

where  $N$  is the number of unknowns.

Equation (6) can be solved using an iterative method such as the Generalized Minimal Residual Algorithm (GMRES). The detailed discussions for the parallelized version of GMERS can be found in the literature [2].

### B. A Novel Method for Reducing the Truncation Number

In MLFMA, the memory requirement and the CPU time depend heavily on the truncation number,  $L$ , which is normally determined by the size of box,  $D$  [6]. We should determine the minimum value of  $D$  in order to reduce the truncation number, and thus save the memory requirement and CPU time. The relation between the truncation number  $L$  and  $D$  can be expressed as

$$L = kD + \ln(\pi + kD) \quad (7)$$

Previously,  $D$  is determined by the real size of box in each level. The truncation number  $L$  then can be calculated using equation (7). However, the value of  $D$  obtained in this way is not a minimum, for there are spaces for many boxes [4]. In [4], one method is proposed to determine the minimal  $D$  at each level by finding the maximum distance of the edge-distance located in each box.

In this section, we present another method, by finding the equivalent maximum distance in each box, to determine the value of  $D$ . The equivalent maximum distance on level  $L$  is defined as

$$d(l) = \sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2 + (z_{\max} - z_{\min})^2}$$

where  $x_{\min}, y_{\min}, z_{\min}$  and  $x_{\max}, y_{\max}, z_{\max}$  are the minimum and maximum coordinates, respectively, among the triangular patch pair center points for each box on one level. Figure 1 shows the relation of the equivalent maximum distance,  $d$ , and  $D$  concerning with the real size of box for a two-dimensional problem. The solid and dashed line represents  $d$  and  $D$  respectively, and the white nodes are the patch pair center points in the box.

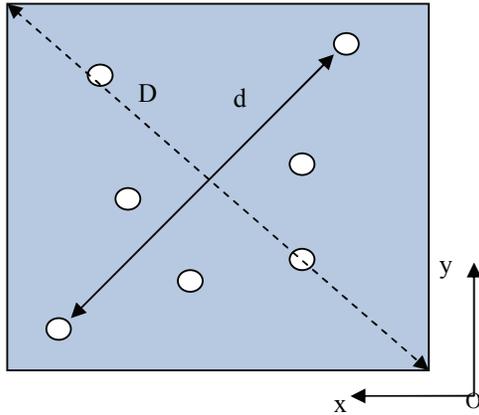


Fig. 1. The relation of  $d$  and  $D$  for a two-dimensional problem.

We could calculate every  $d$  in every box at each level, find the maximal one, and designate it as the equivalent value of  $D$ . Thus, the truncation number at each level can be determined by

$$L = kd(l) + \ln(\pi + kd(l)) \quad (8)$$

For the equivalent maximum distance  $d$  is less than the value of  $D$ , thus the truncation number,  $L$ , can be reduced without loss of the accuracy. For a target discretized with tens of millions unknowns, the time consuming on finding the equivalent maximum distance can be neglected. Also, this method can be efficiently parallelized. Figure 2 shows the Bi-static RCS of a sphere with a diameter of  $4\lambda$ , the result shows our

method agrees well with the MIE series.

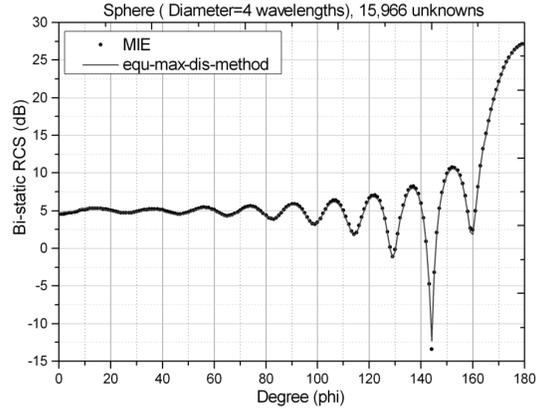


Fig. 2. The Bi-static RCS of our method and the MIE series.

### C. The Matrix-Sliced-to-Disk Technique

For many extremely large targets even modern servers and computers will encounter memory-hungry problems. The memory in MLFMA is mainly consumed in setting up the matrix equation. The idea that transforms the matrix data into the hard disk is straight-forward. There are three main reasons for adopting this technique to our parallel MLFMA code:

- 1) This approach allows us to solve extremely big problems without having to worry about the memory consumption. Memory is almost used for other parts of MLFMA such as the oct-tree rather than the matrix equation.
- 2) From an economic and convenient point of view, this approach helps our code to be more scalable and meet the demand of some low-performance computers. With this approach, we can solve a problem with about one million unknowns on a single computer of only 2 GB memory.
- 3) With the swift improvements in the hard drive storage technology, the difference of the I/O speed of memory and the hard disk will be reduced, making this approach more and more attractive, as shown in Fig. 3.

Actually, this technique will cost slightly more time than without it, for the I/O operation of hard disk is relatively slower than that of memory. Thus it is necessary to compare the CPU time and the elapsed time (which express the total time from the start of a program to the

end of it), in order to evaluate the efficiency of this technique. Sphere with different electrical sizes are considered here. The diameters of sphere range from 20 wavelengths to 220 wavelengths. The CPU and elapsed time for different sizes are depicted in Figure. 3. All the calculations are carried out on one computer with 4 CPUs and a high performance SAS hard disk. The result shows that up to nearly 50 million unknowns, the performance with this technique is only 4.5% slower than that without it. For the situation that the number of unknowns less than 10 million, the differences of with and without this technique can be well neglected.

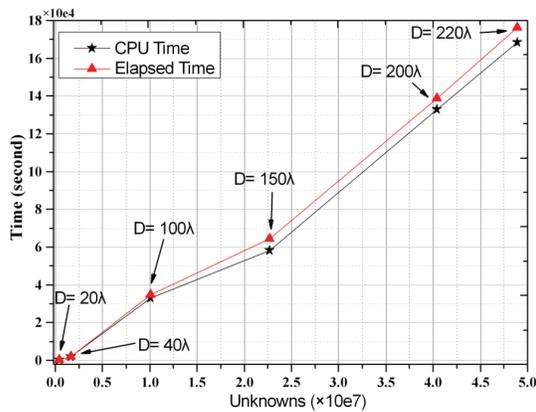


Fig. 3. The comparisons of CPU time and elapsed time for different electrical sizes of sphere.

The solid-state store technique is the trend. With the technical improvement of high performance hard disk, the influence of this relatively slower I/O operation will be less significant. Thus, we can say that this technique will be much more practical in the future.

**D. The Transition Level Scheme**

An important part of parallel MLFMA is the parallel efficiency. Previously, the boxes were distributed equally among the processors. It is natural that this parallel approach can achieve good load-balancing in fine levels. However, it is difficult to achieve good load-balancing in the coarse levels with this approach, since the number of boxes is small in those levels. This usually degrades the parallel efficiency and performance of parallel MLFMA code.

A transition level scheme is proposed in [3] in order to improve the parallel efficiency. In the levels that are finer than the transition level, the boxes are distributed equally among the processors; in the levels that are coarser than the transition level, the far-field pattern and translation matrix are distributed equally among the processors. However, this scheme causes additional communication between processors. In order to reduce this problem of communication and to restrict each processor to communicate with only two nearby processors at most, it is proved in [4] that the transition level should be the level where the truncation number,  $L$ , is not less than twice the number of processors,  $p$ . To obtain good parallel efficiency, we usually choose  $L=2p$ .

**E. The Efficiency of Parallel MLFMA**

The efficiency of parallel algorithm is defined as

$$\eta = \frac{T_1}{pT_p} \times 100\%$$

where  $p$  is the number of processors,  $T_p$  is the CPU time consumed for  $p$  processors.

To demonstrate the efficiency of our Parallel MLFMA, the bi-static RCS of a sphere of diameter  $40\lambda$  is calculated. The total parallel efficiency and matrix-vector multiplication parallel efficiency is shown in Fig. 4. We can see that the efficiency is above 80% even for 16 processors.

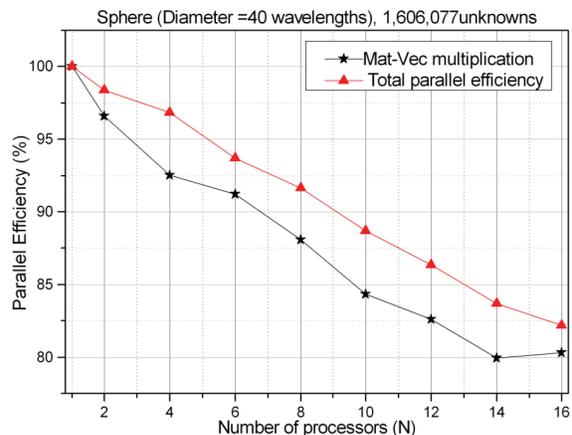


Fig. 4. The parallel efficiency for a sphere of diameter  $40\lambda$  from 1 to 16 processors.

### III. CAPABILITY OF THE PARALLEL MLFMA

A parallel MLFMA code has been developed by implementing several tricks presented in the above sections. To demonstrate the capability of our code, we first calculate the bi-static RCS of a sphere of diameter  $220\lambda$  discretized with 48,879,411 unknowns. The incident angle is  $(90^\circ, 0^\circ)$ , and the scanning plane is  $xy$  plane with 1801 sampling points from  $(0^\circ, 180^\circ)$ . The parallelized GMRES is adopted to solve the matrix equation, and the residual error is 0.005. The simulation is carried out on one single computer with 8 Xeon 3.0 GHz CPUs and 64 GB memory. The detail resources used in this calculation is shown in Table 1.

Table 1. The computational resources for a sphere of  $220\lambda$  by the parallel MLFMA.

General Information		CPU Time (min)	
Geometry size (wavelength)	220	Geometry information	5.6
Number of processors	8	Set up of near-field matrix	199.1
Number of iterations	19	Set up of far-field matrix	67.3
Total memory (GB)	41.5	Iteration and solution	1121.5
Total time (hr)	26.9	RCS calculation	0.63

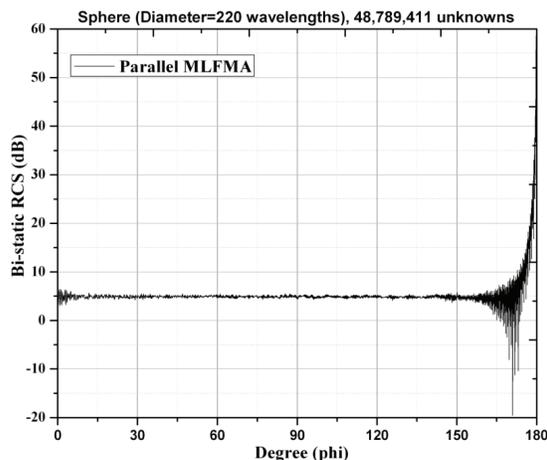


Fig. 5. The bi-static RCS for a sphere of diameter  $220\lambda$ .

To further demonstrate the capability of our parallel MLFMA code, we calculate the bi-static RCS of a square patch of size of  $200 \times 200$  wavelengths with 10,150,143 unknowns. Since this is an open structure, EFIE is used to solve this problem. The patch is located in the  $yz$  plane with its center at the origin. The incident angle is  $(90^\circ, 0^\circ)$  and the scanning plane is  $xy$  with 1801 sampling points from  $(0^\circ, 180^\circ)$ . The parallelized GMRES is adopted to solve the matrix equation, and the residual error is 0.001. This simulation is carried out on one computer with 8 Xeon 3.0 GHz CPUs and 64GB memory. In this problem, only 4 CPUs are used. The detailed resources for this calculation are shown in Table 2.

Table 2. The computational resources for a square patch of  $200\lambda$  by the parallel MLFMA.

General Information		CPU Time (min)	
Geometry size (wavelength)	200	Geometry information	2.6
Number of processors	4	Set up of near-field matrix	40.1
Number of iterations	142	Set up of far-field matrix	13.4
Total memory (GB)	14.7	Iteration and solution	624.9
Total time (hr)	11.5	RCS calculation	0.37

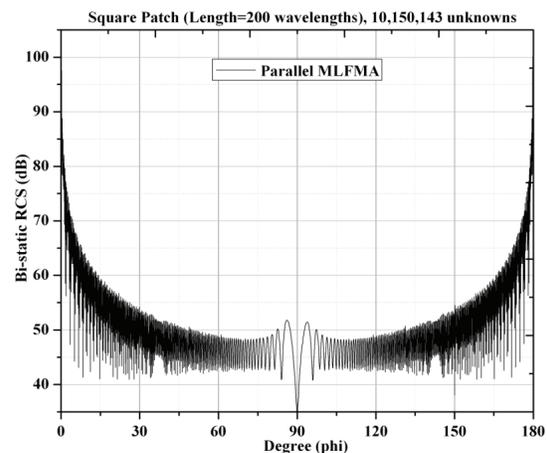


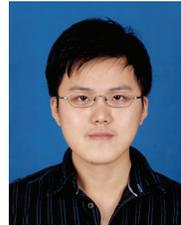
Fig. 6. The bi-static RCS for a square patch of length  $200\lambda$ .

#### IV. CONCLUSIONS

In this paper, several implementation tricks of parallel MLFMA have been introduced and analyzed. Firstly, we proposed a modified truncation number method in order to reduce the memory and CPU time usage; secondly, a technique that sliced matrix to hard disk is applied to fulfill the memory demand for extremely large problems; finally, a transition level scheme is introduced in order to improve the parallel efficiency. With these tricks, memory usage can be reduced. We demonstrate the capability of our code by considering a sphere of diameter  $220\lambda$ , containing nearly 50 million unknowns and a square patch with a length of  $200\lambda$ , involving approximately 10 million unknowns.

#### REFERENCES

- [1] W. C. Chew, J. M. Jin, E. Michielssen, and J. M. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Norwood, MA, Artech House, 2001.
- [2] S. Velamparambil, W. C. Chew, and J. M. Song. "10 Million Unknowns: Is It That Big?" *IEEE Transaction on Antennas Propagation Magazine*, vol. 45, no. 2, pp. 43-58, April 2003.
- [3] S. Velamparambil and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 8, pp. 2719–2727, August 2005.
- [4] X. M. Pan and X. Q. Sheng. "A Sophisticated Parallel MLFMA for Scattering by Extremely Large Targets". *IEEE Antennas and Propagation Magazine*, vol. 50, no. 3, pp. 129-138, June 2008.
- [5] O. Ergul and L. Gurel, "Efficient Parallelization of the Multilevel Fast Multipole Algorithm for the Solution of Large-Scale Scattering Problems," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2335-2345, August 2008.
- [6] R. Coifman, V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription," *IEEE Antennas and Propagation Magazine*, vol. 35, no.3, pp. 7-12, June 1993.
- [7] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 3, pp. 409–418, May 1982.
- [8] T. Malas, O. Ergul and L. Gurel, "Sequential and Parallel Preconditioners for Large-Scale Integral- Equation Problems," *Computational Electromagnetics Workshop*, pp.35–43, August 2007.



**Hu Fangjing** received the B.S. degree in electromagnetic field and microwave technique from the University of Electronic Science and Technology of China (UESTC), Chengdu, in 2007, where he is currently working toward the M.S. degree. His research has mainly focused on fast integral equation methods and in particular, multilevel fast multipole method and its parallelization.



**Nie Zaiping** was born in Xi'an, China, in 1946. He received the B.S. degree in radio engineering and the M.S. degree in electromagnetic field and microwave technology from the Chengdu Institute of Radio Engineering (now UESTC: University of Electronic Science and Technology of China), Chengdu, China, in 1968 and 1981, respectively. From 1987 to 1989, he was a Visiting Scholar with the Electromagnetics Laboratory, University of Illinois, Urbana. Currently, he is a Professor with the Department of Electromagnetic Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has published more than 380 papers. His research interests include antenna theory and techniques, field and waves in inhomogeneous media, computational electromagnetics, electromagnetic scattering and inverse scattering, new techniques for antenna in mobile communications, transient electromagnetic theory and applications.



**Hu Jun** received the B.S., M.S., and Ph.D. degrees in electromagnetic field and microwave technique from the University of Electronic Science and Technology of China (UESTC), Chengdu, in 1995, 1998, and 2000, respectively. During 2001 he was with the Center of Wireless Communication in the City University of Hong Kong, Kowloon, as a Research Assistant. He is currently an Associate Professor with the School of Electronic Engineering of UESTC. He is the author or coauthor of over 90 technical papers. His current research interests include computational electromagnetics, electromagnetic scattering, and radiation.