

# An Efficient Parallel Implementation of the Multilevel Fast Multipole Algorithm for Rigorous Solutions of Large-Scale Scattering Problems

Özgür Ergül<sup>#1</sup> and Levent Gürel<sup>\*2</sup>

<sup>#</sup>*Department of Mathematics and Statistics*

*University of Strathclyde, G11XH, Glasgow, UK*

<sup>\*</sup>*Department of Electrical and Electronics Engineering*

*Computational Electromagnetics Research Center (BiLCEM)*

*Bilkent University, TR-06800, Ankara, Turkey*

<sup>1</sup>ozgur.ergul@strath.ac.uk, <sup>2</sup>lgurel@bilkent.edu.tr

**Abstract**—We present the solution of large-scale scattering problems discretized with hundreds of millions of unknowns. The multilevel fast multipole algorithm (MLFMA) is parallelized using the hierarchical partitioning strategy on distributed-memory architectures. Optimizations and load-balancing algorithms are extensively used to improve parallel MLFMA solutions. The resulting implementation is successfully employed on modest parallel computers to solve scattering problems involving metallic objects larger than  $1000\lambda$  and discretized with more than 300 million unknowns.

## I. INTRODUCTION

Electromagnetics problems can be solved accurately and efficiently with the multilevel fast multipole algorithm (MLFMA) [1]. For an  $N \times N$  dense matrix equation, MLFMA reduces the complexity of matrix-vector multiplications from  $\mathcal{O}(N^2)$  to  $\mathcal{O}(N \log N)$ , allowing for the iterative solution of large-scale problems discretized with large numbers of unknowns. Nevertheless, many real-life problems require discretizations with millions of unknowns, which cannot easily be solved with sequential implementations of MLFMA. In order to solve such very large problems, MLFMA can be parallelized on distributed-memory architectures [2]–[5]. However, due to the complicated structure of this algorithm, this is not a trivial process. Recently, we developed a hierarchical partitioning strategy [6],[7], which significantly improves the parallelization of MLFMA compared to previous parallelization techniques. Using the hierarchical strategy, we were able to solve scattering problems discretized with more than 200 million unknowns on relatively inexpensive computing platforms [7].

Although the hierarchical strategy provides improved partitioning of the tree structures constructed in MLFMA, solutions of large-scale problems require many other robust techniques to handle large data structures, to organize communications between processors, and to economically use the available memory. Optimizations and load-balancing algorithms are required at each stage of the program to improve parallel solutions. In this paper, we present our recent efforts to solve large-

scale scattering problems using a parallel implementation of MLFMA. We demonstrate the effectiveness and robustness of the developed implementation by solving scattering problems involving metallic objects larger than  $1000\lambda$  and discretized with more than 300 million unknowns.

## II. PARALLEL MLFMA IMPLEMENTATION

### A. Robust Construction of the Tree Structure

For an object with an electrical dimension of  $kD$ , where  $k = 2\pi/\lambda$  is the wavenumber, a multilevel tree structure with  $L = \mathcal{O}(\log(kD))$  levels is constructed by placing the object in a cube and recursively dividing the object into subdomains. For efficient solutions, subdomains at the lowest level ( $l = 1$ ) should be small, but they should be large enough to avoid excessive errors caused by the low-frequency breakdown of MLFMA. In our typical solutions with maximum 1% error, we choose the size of the subdomains at the lowest level in the  $0.15\lambda - 0.3\lambda$  range. For an object larger than  $615\lambda$ , the tree structure involves at least 13 levels. Although we consider only nonempty subdomains and most objects lead to sparse octrees, constructing a tree structure with large numbers of levels can be difficult and it can easily become a bottleneck of the MLFMA implementation.

Table I summarizes a robust technique that can be used to construct a multilevel tree structure with large numbers of levels. The first loop is constructed over basis/testing functions and we locate each basis/testing function in a subdomain at the lowest level. Starting from level  $L-1$ , one of eight subdomains containing that particular basis/testing function is determined at each level. Given a subdomain  $C$  at level  $l > 1$ , indices of eight subdomains  $C' \in C$  at level  $l-1$  can be found easily using the properties of octrees. Indices of subdomains at the lowest level containing the basis/testing functions are stored in an array called *subdomains*. When all basis/testing functions are processed, the *subdomains* array storing the indices according to the full octree is sorted using a quick-sort algorithm. This allows us to trace the array rapidly to determine the number of distinct subdomains at the lowest

TABLE I  
PSEUDOCODE FOR ROBUST CONSTRUCTION OF THE MULTILEVEL TREE  
STRUCTURE

```

do for each basis/testing function  $n = 1, 2, \dots, N$ 
  do for each level  $l = (L - 1), (L - 2), \dots, 1$ 
    place the function in one of eight subdomains
     $subdomains[n] \leftarrow$  full-octree index of the subdomain
    at the lowest level
sort  $subdomains$  array
count number of distinct subdomains at the lowest level
renumber subdomains at all levels

```

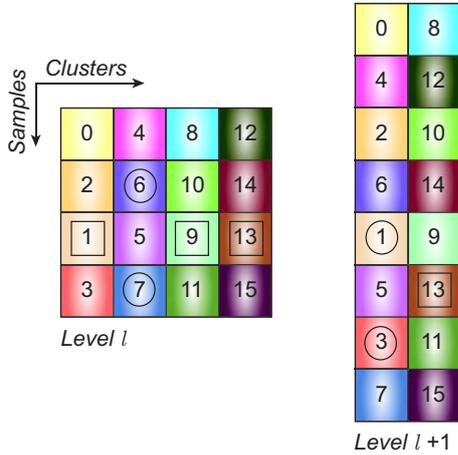


Fig. 1. Partitioning maps of two consecutive levels into 16 processes using the hierarchical strategy. Each processor (or process) handling a group of clusters and a portion of the field spectrum is denoted by a number. Processors that need to communicate with Processor 5 are marked with circles and squares.

level, as well as the number of basis/testing functions in each subdomain. Finally, subdomains are renumbered at all levels, considering only nonempty ones. The complexity of this technique is  $\mathcal{O}(N \log N)$ , which is appropriate for an MLFMA implementation.

### B. Hierarchical Partitioning Strategy

MLFMA can be parallelized efficiently using the hierarchical partitioning strategy, which is based on partitioning both subdomains and field samples among processors [7]. A typical partitioning of two consecutive levels into 16 processes using the hierarchical strategy is depicted in Fig. 1. At level  $l$ , the number of partitions, both along subdomains (horizontal direction) and samples (vertical direction), is four. At level  $l+1$ , however, the partitioning is changed, subdomains are divided into two partitions, and samples are divided into eight partitions. In general, the partitioning at each level is optimized using load-balancing algorithms such that the processing time and the memory required by the MLFMA implementation are minimized. As detailed in [7], the hierarchical strategy provides important advantages, compared to previous parallelization techniques for MLFMA. Specifically, partitioning both subdomains and samples of fields leads to improved load-balancing among processors at all levels. In addition, communications between processors are reduced and communication time is significantly shortened.

### C. Communications

Using the hierarchical partitioning strategy, there are three different types of communications required among processors during matrix-vector multiplications [7],[8]. Here we describe these communications by considering Processor 5 in Fig. 1; other processors also perform similar communications. During aggregation and disaggregation stages, Processor 5 needs to communicate with two neighboring processors in the same column, i.e., Processors 6 and 7 at level  $l$  and Processors 1 and 3 at level  $l+1$ . These (first type) communications need perfect synchronization between processors and their efficiency can be improved with load-balancing algorithms. Then, during the translation stage, Processor 5 communicates with processors in the same row of the partitioning map, i.e., Processors 1, 9, and 13 at level  $l$  and Processor 13 at level  $l+1$ . For these (second type) communications, the order of pairing is important and directly affects the efficiency [8]. For example, at level  $l$ , Processor 5 can be paired with Processors 1, 9, and 13 in different orders, such as  $\{1, 9, 13\}$ ,  $\{1, 13, 9\}$ ,  $\{9, 1, 13\}$ ,  $\{9, 13, 1\}$ ,  $\{13, 1, 9\}$ , and  $\{13, 9, 1\}$ , but only one of them is optimal in terms of the processing time. In practice, we consider the overall tree structure to determine the order of communications among processors. Load-balancing algorithms are also helpful to improve the synchronization and to avoid waiting periods between pairing rounds. Finally, from level  $l$  to level  $l+1$ , Processor 5 exchanges data with Processor 1 to modify the partitioning. Considering levels  $l$  and  $l+1$ , this (third type) communication is performed once, but it involves large data transfers between processors. This is an extra communication type introduced by the hierarchical strategy [7], but we emphasize that it results in an improvement in terms of parallelization. Specifically, this type of communication replaces many (first and second type) communications that would be required during aggregation, disaggregation, and translation stages if the hierarchical strategy was not applied. Instead of transferring many small packages, the hierarchical strategy enables us to collect them and communicate the same amount of data with larger packages, which effectively reduces the communication time.

### D. Memory Recycling

Solutions of large problems require efficient use of the available memory. In our MLFMA implementation, we utilize memory recycling as much as possible to solve larger problems with limited computational resources. We accomplish this with a three-point strategy:

- 1) Allocate memory for a data structure just before its storage is required, not earlier.
- 2) Deallocate memory used for a data structure whenever it becomes useless and thus it will not be used again as the program continues.
- 3) Rearrange the program by relocating code segments such that items (1) and (2) can be further applied to reduce the memory requirement.

Relocation of code segments, particularly in the input and setup stages of the MLFMA implementation, can effectively

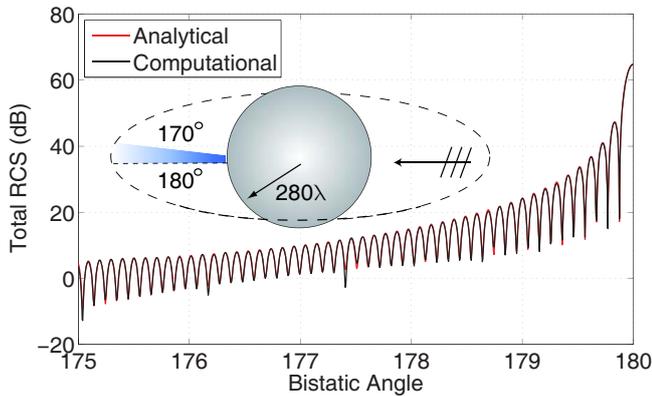


Fig. 2. Solution of a scattering problem involving a metallic sphere of diameter  $560\lambda$  discretized with 374,490,624 unknowns. Normalized RCS (dB) is plotted as a function of bistatic angle from  $175^\circ$  to  $180^\circ$ , where  $180^\circ$  corresponds to the forward-scattering direction.

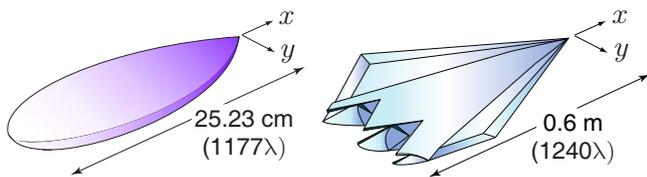


Fig. 3. Very large metallic objects (NASA Almond and Flamme) whose scattering problems are solved with the parallel MLFMA implementation. Both objects are discretized with more than 300 million unknowns.

reduce the instantaneous memory usage and prevent memory overflows prior to the iterative solution (matrix-vector multiplications) stage of the program.

### E. Optimization of the Peak Memory

Applying memory recycling, all unnecessary data structures are deallocated; only essential data structures remain allocated before the iterative solution. During the iterative solution, a majority of the memory is used for near-field interactions, radiation/receiving patterns of basis/testing functions, translation operators, and aggregation/disaggregation arrays [4]. For solving large problems on distributed-memory architectures, it is essential to distribute those data structures equally among processors. Otherwise, even though the total amount of memory is sufficient to solve a problem, the memory required by a specific processor can exceed the maximum memory available for that processor, and this may prevent the solution of the problem. Hence, the peak memory of the parallel MLFMA implementation should be carefully optimized such that all processors require approximately the same amount of memory during iterative solutions. We note that this optimization is different from the load-balancing scheme for the multilevel tree structure, which is essential to minimize the processing time. For the optimization of peak memory, we consider all significant contributions in terms of memory, such as near-field interactions, in addition to the tree structure.

In order to demonstrate the accuracy and efficiency of the developed parallel MLFMA implementation, we present the solution of a scattering problem involving a metallic sphere of diameter  $560\lambda$  illuminated by a plane wave propagating in the  $-x$  direction. Discretization of the sphere with the Rao-Wilton-Glisson functions on  $\lambda/10$  triangles leads to a  $374,490,624 \times 374,490,624$  matrix equation. Both near-field and far-field interactions are calculated with maximum 1% error using a 13-level MLFMA ( $L = 13$ ). The solution is parallelized into 64 processes on a cluster of quad-core Intel Nehalem processors with a 2.67 GHz clock rate (Nehalem cluster). Convergence to 0.001 residual error is achieved in 31 iterations using the biconjugate-gradient-stabilized (BiCGStab) algorithm. The total processing time is 21 hours and the total memory required for the solution is 1.3 TB (1330 GB). Fig. 2 presents the normalized bistatic radar cross section (RCS) of the sphere in decibels (dB) on the  $x$ - $y$  plane as a function of the bistatic angle  $\phi$  from  $175^\circ$  to  $180^\circ$ , where  $180^\circ$  corresponds to the forward-scattering direction. We observe that computational values provided by the parallel MLFMA implementation agree perfectly with an analytical Mie-series solution.

Next, we present the solution of scattering problems involving two important metallic targets from the literature, namely, the NASA Almond and the Flamme [9], as depicted in Fig. 3. The NASA Almond is investigated at 1.4 THz, where its size corresponds to  $1177\lambda$ , and it is discretized with 306,696,192 unknowns. The Flamme is investigated at 620 GHz, where its size corresponds to  $1240\lambda$ , and it is discretized with 308,289,024 unknowns. Both targets are located on the  $x$ - $y$  plane such that their noses are directed towards the  $x$  axis, and they are illuminated by a plane wave propagating in the  $-x$  direction with the electric field polarized in the  $\phi$  direction. The NASA Almond and the Flamme problems are solved in 11 and 17 hours, respectively, by employing a 14-level MLFMA parallelized into 64 processes on the Nehalem cluster using a total of 1.3 TB memory.

Fig. 4 illustrates the amount of memory (in GB) used by each process as a function of time for the solution of the Flamme problem. Only one matrix-vector multiplication is considered since the memory requirement is exactly the same for all matrix-vector multiplications. We observe that the used memory is not monotonically increasing and it fluctuates due to allocations and deallocations (memory recycling). In addition, Process 0 uses more memory than the other processes during input and setup stages since we allocate some sequential arrays only for this process. It is also remarkable that all processes use nearly the same amount of memory during the matrix-vector multiplication, thanks to the optimizations.

Finally, Fig. 5 presents the bistatic RCS in dB meter squares (dBms) of the NASA Almond and the Flamme on the  $x$ - $y$  plane as a function of the bistatic angle  $\phi$ . We observe that the cross-polar RCS of the NASA Almond is quite low compared to its co-polar RCS, and the RCS of the NASA

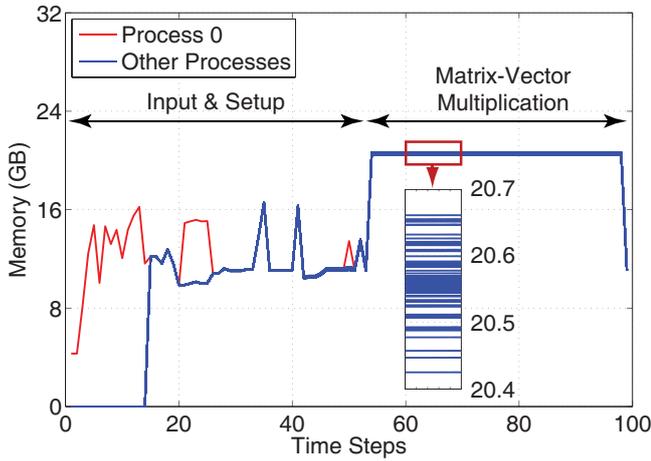


Fig. 4. Memory (GB) used by each process as a function of time for the solution of the Flamme problem in Fig. 3.

Almond exhibits a visible peak only in the forward-scattering direction. On the other hand, the cross-polar RCS of the Flamme is significant and comparable to its co-polar RCS, and the Flamme RCS exhibits two significant peaks at around  $150^\circ$  and  $210^\circ$ , due to specular reflections from the two straight edges of the nearly flat surfaces of this target.

#### IV. CONCLUSION

An efficient parallel implementation of MLFMA using the hierarchical partitioning strategy is presented for rigorous solutions of very large scattering problems. The developed implementation is successfully used to solve difficult problems involving metallic objects larger than  $1000\lambda$  and discretized with more than 300 million unknowns.

#### ACKNOWLEDGMENT

This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under the Research Grant 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. Özgür Ergül was also supported by a Research Starter Grant provided by the Faculty of Science at the University of Strathclyde. Computer time was provided in part by a generous allocation from the Turkish Academic Network and Information Center (ULAKBIM).

#### REFERENCES

- [1] J. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propagat.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.
- [2] 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.
- [3] 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.
- [4] 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.

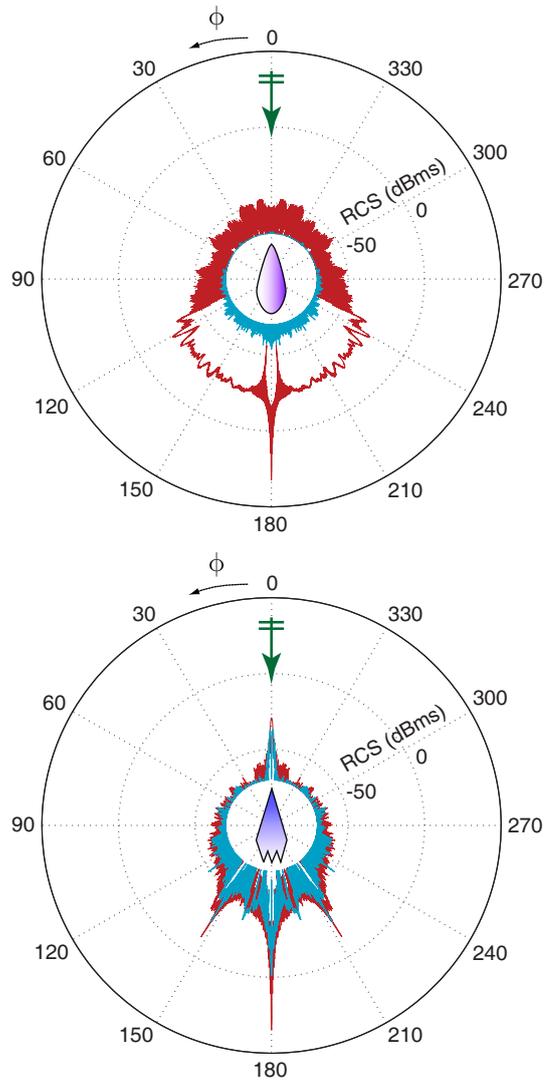


Fig. 5. Co-polar (red) and cross-polar (blue) bistatic RCS (dBms) on the  $x$ - $y$  plane of the NASA Almond and the Flamme depicted in Fig. 2. RCS values lower than  $-70$  dBms are omitted. Both targets are illuminated by a plane wave propagating in the  $-x$  direction with the electric field polarized in the  $\phi$  direction.

- [5] 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.
- [6] 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.
- [7] Ö. Ergül and L. Gürel, "A hierarchical partitioning strategy for an efficient parallelization of the multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 57, no. 6, pp. 1740–1750, June 2009.
- [8] Ö. Ergül and L. Gürel, "Advanced partitioning and communication strategies for the efficient parallelization of the multilevel fast multipole algorithm," *2010 IEEE International Symposium on Antennas and Propagation and CNC/USNC/URSI Radio Science Meeting*, Toronto, Ontario, Canada, July 2010.
- [9] 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, June 2003.