# Advanced Partitioning and Communication Strategies for the Efficient Parallelization of the Multilevel Fast Multipole Algorithm[†]

Özgür Ergül[1*] and Levent Gürel[2,3]
[1]Department of Mathematics and Statistics
University of Strathclyde, G11XH, Glasgow, UK
[2]Department of Electrical and Electronics Engineering
[3]Computational Electromagnetics Research Center (BiLCEM)
Bilkent University, TR-06800, Bilkent, Ankara, Turkey
ozgur.ergul@strath.ac.uk, lgurel@bilkent.edu.tr

## Introduction

Large-scale electromagnetics problems can be solved efficiently with the multilevel fast multipole algorithm (MLFMA) [1], which reduces the complexity of matrix-vector multiplications required by iterative solvers from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$. Parallelization of MLFMA on distributed-memory architectures enables fast and accurate solutions of radiation and scattering problems discretized with millions of unknowns using limited computational resources. Recently, we developed a hierarchical partitioning strategy [2], which provides an efficient parallelization of MLFMA, allowing for the solution of very large problems involving hundreds of millions of unknowns. In this strategy, both clusters (subdomains) of the multilevel tree structure and their samples are partitioned among processors, which leads to improved load-balancing. We also show that communications between processors are reduced and the communication time is shortened, compared to previous parallelization strategies in the literature. On the other hand, improved partitioning of the tree structure complicates the arrangement of communications between processors.

In this paper, we discuss communications in detail when MLFMA is parallelized using the hierarchical partitioning strategy. We present well-organized arrangements of communications in order to maximize the efficiency offered by the improved partitioning. We demonstrate the effectiveness of the resulting parallel implementation on a very large scattering problem involving a conducting sphere discretized with 375 million unknowns.

## Communications in the Aggregation Stage

In order to describe the organization of communications during the matrix-vector multiplications with MLFMA, particularly for the aggregation stage, Fig. 1 depicts a simple example involving a hierarchical partitioning of two consecutive levels among eight processors labeled from 0 to 7. For the sake of brevity, we will mainly focus on communications involving processor 2. At level $l$, clusters and field samples are divided into four and two partitions, respectively. The partitioning is changed at level $l+1$, where clusters and field samples are divided into two and four partitions. To facilitate this rearrangement of partitioning, we define an intermediate level $l+1/2$, as also illustrated in Fig. 1. A set of clusters and a set of field samples assigned to a processor $p$ at any level $l$ are represented by $C_p^l$ and

[†]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).

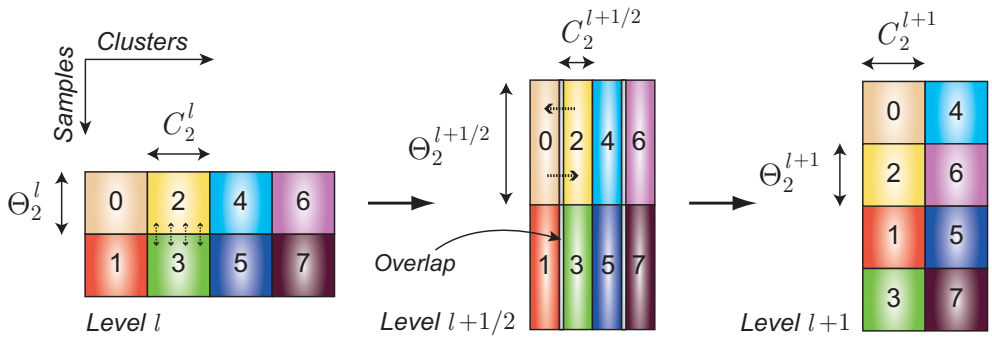


Fig. 1. Partitioning of two consecutive levels among eight processors using the hierarchical partitioning strategy.

Table 1: Pseudocode of major operations performed by a processor $p$ for an aggregation step from level $l$ to $l+1$.

do for each cluster $c \in C_p^{l+1/2}$
  do for each sub-cluster $c_s \in Sub\{c\}$ & $c_s \in C_p^l$
    get $\mathcal{F}_p\{c_s\}$ from the local aggregation array
    $\mathcal{F}_p\{c_s\} \rightarrow$ column-wise (neighboring) communications $\rightarrow \mathcal{F}_{p+}\{c_s\}$
    $\mathcal{F}_{p+}\{c_s\} \rightarrow$ local interpolation $\rightarrow \tilde{\mathcal{F}}_p\{c_s\}$
    divide data into two parts: $\tilde{\mathcal{F}}_p\{c_s\} = \tilde{\mathcal{F}}_p^A\{c_s\} \bigcup \tilde{\mathcal{F}}_p^B\{c_s\}$
    $\tilde{\mathcal{F}}_p^A\{c_s\} \rightarrow$ shift radiation center and add to the local aggregation array
    $\tilde{\mathcal{F}}_p^B\{c_s\} \rightarrow$ shift radiation center and add to a communication array
send and receive data (communication array) to modify the partitioning

$\Theta_p^l$, respectively. We assume that partitioning (numbers of partitions along clusters and samples) is optimized to minimize the processing time and memory. At actual levels ($l$ and $l+1$), there is no overlapping data between processors. At the intermediate level, however, samples for a single cluster can be duplicated at some of the neighboring processors.

Table 1 lists the major operations performed for an aggregation step from level $l$ to level $l+1$ using an intermediate level $l+1/2$. The first loop is constructed over local clusters at the intermediate level. Then, for a given cluster $c$, local sub-clusters $Sub\{c\}$ are considered in the second loop. We note that processors that are in the same column of the intermediate-level partitioning handle the same set of clusters and sub-clusters, and this requires a perfect synchronization for efficiency. For a given sub-cluster $c_s$, a set of field samples $\mathcal{F}_p\{c_s\}$ is extracted from the aggregation array, and the local data is inflated via one-to-one communications between neighboring processors. Next, the radiated field represented by an inflated set of samples $\mathcal{F}_{p+}\{c_s\}$ is interpolated according to the sample rate of the next level. At this point, the contribution of the sub-cluster $c_s$ to the radiation of the parent cluster $c$ can be obtained by shifting the radiation center for the field values at the samples. But, since the partitioning is required to be changed, the inner loop is completed by dividing the samples $\tilde{\mathcal{F}}_p\{c_s\}$ into two parts. Some of the samples, i.e., $\tilde{\mathcal{F}}_p^A\{c_s\}$, are added to the local aggregation array, while the others, i.e., $\tilde{\mathcal{F}}_p^B\{c_s\}$, are added to a communication array to be sent to a corresponding processor. After all clusters at the intermediate level are processed, data collected in the communication arrays are exchanged between pairs of processors to modify the partitioning.

Table 2: Pseudocode of major operations performed by a processor $p$ for inter-processor translations.

```
do for each processor p' ≠ p
    do for each level l (also check if any translation is required or not)
        do for each cluster c_b that is required to be transferred from p'
            receive F_{p'}{c_b}
            do for each cluster c_a owned by p and in the far-field list of c_b
                F_{p'}{c_b} → translate → G_p{c_b ⇝ c_a}
                add G_p{c_b ⇝ c_a} to the local disaggregation array
        do for each cluster c_b that is required to be transferred to p'
            send F_p{c_b}
```

In the example depicted in Fig. 1, processor 2 communicates with processor 3 to inflate the local data for each sub-cluster $c_s$. When the loops in Table 1 are completed and the intermediate level is constructed, processor 2 exchanges data with processor 0 to obtain the partitioning at level $l+1$. Considering the data arrangement from level $l+1/2$ to level $l+1$, we note that

$$\Theta_0^{l+1/2} = \Theta_2^{l+1/2} = \Theta_0^{l+1} \bigcup \Theta_2^{l+1} \tag{1}$$

$$C_0^{l+1} = C_2^{l+1} = C_0^{l+1/2} \bigcup C_2^{l+1/2}. \tag{2}$$

## Communications in the Translation Stage

Communications between processors are also required during the translation stage, since clusters are partitioned, and some translations are required among clusters located in different processors. Table 2 lists the major operations performed for those inter-processor translations. In general, each processor $p$ needs to be paired with all other processors ($p' \neq p$) to exchange data. Given a pair of processors, constructing a single connection between them and exchanging all data at once leads to better performance than pairing the processors repetitively. Hence, as shown in Table 2, the first loop for inter-processor translations is pairing processors. When a pairing is established, each level is considered one by one in the second loop. At level $l$, communications are required only if the processors are located in the same row of the partitioning. If this is the case, field samples $\mathcal{F}_{p'}\{c_b\}$ are transferred from processor $p'$ and translations are performed on the receiver side (processor $p$). For a given cluster $c_b$ whose field is transferred from processor $p'$, incoming fields are computed for each cluster $c_a$ that are in the far-field list of $c_b$ and owned by processor $p$. Then, incoming fields $\mathcal{G}_p\{c_b \rightsquigarrow c_a\}$ are added to the local disaggregation array. We note that processor $p$ also sends field samples $\mathcal{F}_p\{c_b\}$ to $p'$ for inter-processor translations to be performed by $p'$.

When two processors are paired, the amount of communications required for inter-processor translations between them depends on the partitions of the tree structure assigned to the processors. Specifically, processors may need to communicate for some levels while they may not for others. Consequently, for efficient inter-processor translations, we carefully determine the order of pairing among processors. For example, consider the partitioning of two levels among eight processors in Fig. 1. When processor 2 is paired with processor 6, they possibly communicate at levels $l$ and $l+1$ since they are located in the same row of the

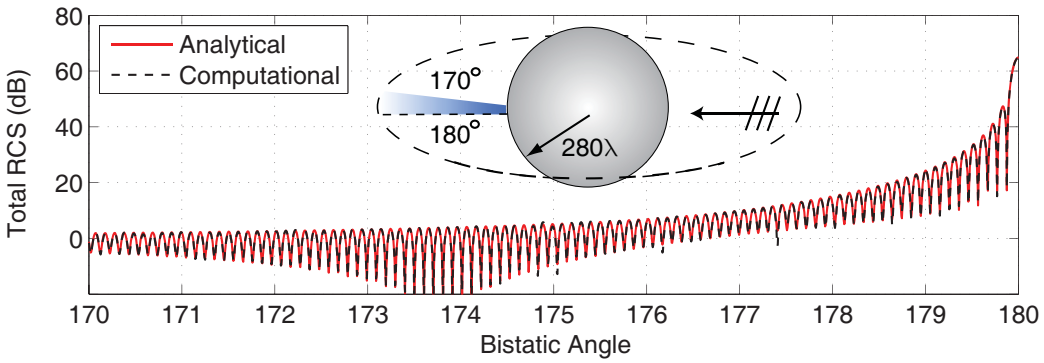


Fig. 2. Solution of a scattering problem involving a metallic sphere of diameter $560\lambda$ discretized with 374,490,624 unknowns. Normalized RCS (dB) is plotted as a function of bistatic angle from $170^\circ$ to $180^\circ$, where $180^\circ$ corresponds to the forward-scattering direction.

partitioning at both levels. Then, at the same time, we match processor 0 with processor 4, processor 1 with processors 5, and processor 3 with processor 7, because all these pairs involve processors that are located in the same row at both levels. This strategy leads to a very good synchronization among processors and avoids possible delays between pairing rounds.

## Efficient Solutions of Very Large Problems

In order to demonstrate the effectiveness of the developed parallel MLFMA implementation using the hierarchical partitioning and communication strategies presented in this paper, Fig. 2 presents the solution of a scattering problem involving a metallic sphere of diameter $560\lambda$. The sphere is illuminated by a plane wave propagating in the $-x$ direction and the discretization of its surface with the Rao-Wilton-Glisson functions on $\lambda/10$ triangles leads to a 374,490,624$\times$374,490,624 dense matrix equation. Both near-field and far-field interactions are calculated with maximum 1% error and convergence to 0.001 residual error is achieved in 31 iterations using the biconjugate-gradient-stabilized (BiCGStab) algorithm. For the solution, which takes about 21 hours, an 11-level MLFMA is parallelized into 64 processes on a cluster of quad-core Intel Nehalem processors with 2.67 GHz clock rate. The total amount of memory used to solve the problem is 1.3 Terabytes. Fig. 2 presents the normalized bistatic radar cross section (RCS) in decibels (dB) on the $x$-$y$ plane as a function of the bistatic angle $\phi$ from $170^\circ$ to $180^\circ$, where $180^\circ$ corresponds to the forward-scattering direction. Computational values provided by the parallel MLFMA implementation are compared with an analytical Mie-series solution, where we observe perfect agreement.

## 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] 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.