MLFMA Memory Reduction Techniques for Solving Large-Scale Problems

Mert Hidayetoğlu and Levent Gürel
Bilkent University Computational Electromagnetics Research Center (BiLCEM), Ankara, TR-06800, Turkey
lgurel@gmail.com

***Abstract*—We present two memory reduction methods for the parallel multilevel fast multipole algorithm. One of these methods uses data structures out of core, and the other parallelizes the data structures related to input geometry. With these methods, large-scale electromagnetic scattering problems can be solved on modest parallel computers.**

I. Introduction

The multilevel fast multipole algorithm (MLFMA) is efficient for iterative solutions of electromagnetics scattering and radiation problems [1]. Its low complexity allows us to solve large-scale problems and the required computation time can be reduced further by parallelization [2].

Scattering and radiation problems of arbitrary geometries can be formulated with surface integral equations. Then, the problem is discretized using the method of moments to obtain the matrix equation

$$\overline{\boldsymbol{Z}} \cdot \boldsymbol{x} = \boldsymbol{b}, \qquad (1)$$

where $\overline{\boldsymbol{Z}}$ is the known impedance matrix, $\boldsymbol{b}$ is the known excitation vector, and $\boldsymbol{x}$ is the unknown coefficient vector of basis functions. When iterative methods are used for solution, matrix-vector multiplications (MVMs) must be performed. For N unknowns, a naive MVM has a computational complexity of $\mathcal{O}(N^2)$, whereas MLFMA reduces the complexity to $\mathcal{O}(N \log N)$. MLFMA uses pre-calculated $\overline{\boldsymbol{Z}}_{\text{NF}}$ directly, while calculating $\overline{\boldsymbol{Z}}_{\text{FF}}$ using a multilevel scheme within a tree structure, where $\overline{\boldsymbol{Z}}_{\text{NF}}$ and $\overline{\boldsymbol{Z}}_{\text{FF}}$ represent near-field and far-field interactions, respectively:

$$\overline{\boldsymbol{Z}} \cdot \boldsymbol{x} = \overline{\boldsymbol{Z}}_{\text{NF}} \cdot \boldsymbol{x} + \overline{\boldsymbol{Z}}_{\text{FF}} \cdot \boldsymbol{x}. \qquad (2)$$

The size of the problem is limited to the memory capacity of the computer on which the algorithm runs. To solve large-scale problems, the memory bottleneck can be overcome by efficient parallelization, memory recycling, and out-of-core methods. In this paper, we introduce an out-of-core method and data structure parallelization related to the input geometry.

II. Out-of-Core Method

The main idea of an out-of-core method is to use disk drives, such as hard disk or solid-state drives, as data storage devices. Disk drives are slower compared to random access memories. Therefore, their intensive usage is not efficient. An efficient way to use disk drives is to store only specifically chosen data structures in them.

As shown in Fig. 1, data structures of near-field interactions and radiation/receiving patterns are calculated and stored on the disk during setup. In the iterative solution, out-of-core data is fetched and is used in MVMs along with in-core data.

Required time for fetching the out-of-core data from the disk scales with $\mathcal{O}(N/p)$ for each MVM, where p is the number of processes. Then, the solution time is increased by $\mathcal{O}(IN/p)$, where I is the number of iterations. One can further decrease the I/O time by using solid-state drives because they are faster than hard disk drives in general and by writing out-of-core data in binary form.

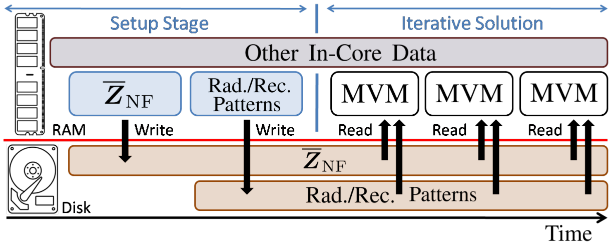

Fig. 1. Out-of-core method for MLFMA.

III. Parallelization of Data Structures

Parallelization of MLFMA reduces both the required time and memory by distributing the computation among processes. Efficient parallelization of MLFMA is achieved with the hierarchical partitioning strategy [3], which parallelizes the iterative-solution part with great success. However, leaving the pre-processing (i.e., processing the input geometry before the setup stage) sequential can cause important bottlenecks for very large problems. As a remedy, in this work, the pre-processing stage is parallelized by distributing both the computation and data among processes. This way, the memory consumptions of both the pre-processing and setup stages are decreased.

Components of the input geometry, such as nodes and triangles, are distributed among processes in a load-balanced manner. To process the geometry data, each process needs all geometry components, therefore intense communication among processes is performed in a cyclic manner.

Figure 2 shows the communication scheme among four processes. Each process passes its data to its neighbour process in every iteration. This allows every process to have a chance to obtain any fraction of the geometry data. After p iterations, each process has their initial portion of data.

978-1-4799-3540-6/14/$31.00 ©2014 IEEE

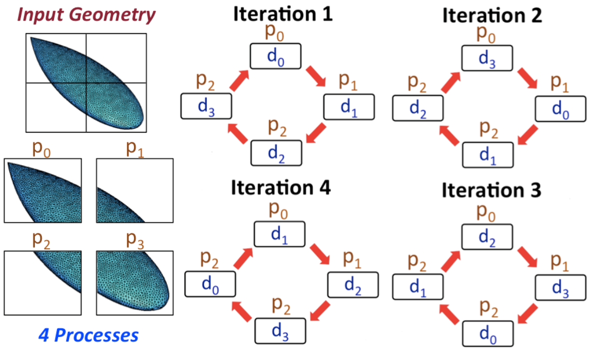

Fig. 2. Cyclic communications among processes.

IV. Numerical Results

To demonstrate the memory reduction of MLFMA, we consider a set of solutions for a conducting sphere, discretized with various numbers of unknowns. The sphere has a radius of 0.3 m and the mesh size is chosen as 0.1λ, where λ is the wavelength in free-space.

A sample problem involving a conducting sphere with radius of 400λ and 670,138,368 unknowns is solved using Bi-CGSTAB solver in 31 iterations to satisfy 1% residual error. CPU time of the solution is 45 hours using 128 processes. Allocated memory spaces, as well as disk usage are recorded at certain checkpoints to observe the memory consumptions of MLFMA and reduced-memory MLFMA (RM-MLFMA).

Figure 3 shows the recorded memory usage of the programs: pre-processing is performed between checkpoints 1 and 15, setup is performed between checkpoints 15 and 47, and MVMs begin after memory checkpoint 47. We consider only the first 60 memory checkpoints beacause memory usage does not increase after the iterative solution has begun. In MLFMA, pre-processing is performed by a single processor, whereas it is distributed among multiple processors in RM-MLFMA.

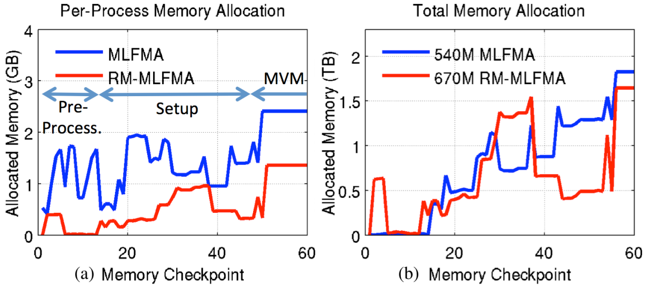

Fig. 3. (a) Per-process memory requirements of a problem involving 53 million unknowns using 128 processes. (b) Total memory requirements of problems involving 540 and 670 million unknowns using 64 and 128 processes, respectively.

Figure 3(a) shows the memory consumption of the solution of a scattering problem of a sphere involving 53 million unknowns. Peak per-process memory is observed as 2.47 GB and 1.37 GB and the average CPU times of an MVM are 43.10 and 43.13 seconds for MLFMA and RM-MLFMA, respectively. In this problem, RM-MLFMA reduces the required memory consumption by 44.53% without a significant CPU time increase.

We also solved spheres with 540 million and 670 million unknowns using MLFMA and RM-MLFMA, respectively. Fig. 3(b) shows the memory consumptions of those solutions. Using the out-of-core method, MVM memory consumption is reduced and the solution of the problem involving 670 million unknowns is made possible with RM-MLFMA using 2 TB of memory, while the largest problem we can solve with the same amount of memory using MLFMA is limited to 540 million unknowns. Fig. 4 presents the radar cross section (RCS) of the solution. Computational results obtained with RM-MLFMA agree well with the analytical Mie-series solutions.

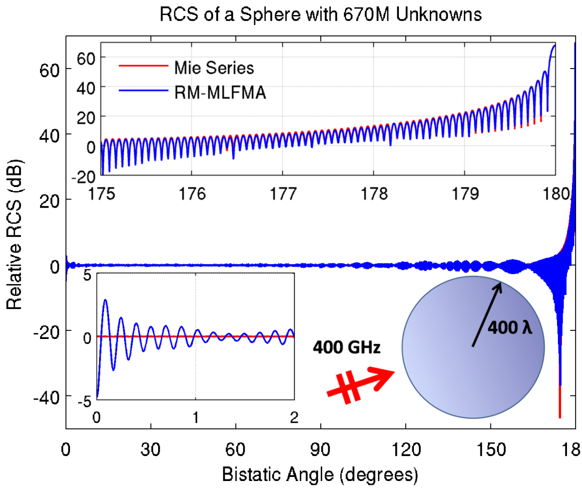

Fig. 4. RCS of a sphere with 670 million unknowns.

V. Conclusions

MLFMA memory is reduced by using an out-of-core storage strategy and by parallelizing the pre-processing data structures. The out-of-core method and pre-processing parallelization are used to reduce the memory consumptions of MVM and setup stages, respectively. With the reduced-memory MLFMA, large-scale electromagnetic scattering problems are solved involving as many as 670 million unknowns with less than 2 TB memory.

Acknowledgement

This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grant 111E203, by Schlumberger-Doll Research (SDR), and by contracts from ASELSAN, Turkish Aerospace Industries (TAI), and the Undersecretariat for Defense Industries (SSM).

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 Propag.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[2] Ö. Ergül and L. Gürel, "Accurate solutions of extremely large integral-equation problems in computational electromagnetics," *Proc. IEEE*, vol. 101, no. 2, pp. 342–349, Feb. 2013.

[3] L. Gürel and Ö. Ergül, "Hierarchical parallelization of the multilevel fast multipole algorithm (MLFMA)," *Proc. IEEE*, vol. 101, no. 2, pp. 332–341, Feb. 2013.