Two-Step Lagrange Interpolation Method for the Multilevel Fast Multipole Algorithm

Özgür Ergül, *Student Member, IEEE*, Idesbald van den Bosch, and Levent Gürel, *Fellow, IEEE*

***Abstract*—We present a two-step Lagrange interpolation method for the efficient solution of large-scale electromagnetics problems with the multilevel fast multipole algorithm (MLFMA). Local interpolations are required during aggregation and disaggregation stages of MLFMA in order to match the different sampling rates for the radiated and incoming fields in consecutive levels. The conventional one-step method is decomposed into two one-dimensional interpolations, applied successively. As it provides a significant acceleration in processing time, the proposed two-step method is especially useful for problems involving large-scale objects discretized with millions of unknowns.**

***Index Terms*—Lagrange interpolation, large-scale problems, multilevel fast multipole algorithm (MLFMA).**

I. Introduction

IT HAS BEEN more than 15 years since the fast multipole method (FMM) was developed for the efficient solution of radiation and scattering problems in electromagnetics [1], [2]. Discretizations of integral equations lead to $N \times N$ dense matrix equations, which can be solved iteratively via a Krylov-subspace algorithm. FMM provides the matrix-vector multiplications (MVMs) required by the iterative algorithms in $\mathcal{O}(N^{3/2})$ time using $\mathcal{O}(N^{3/2})$ memory. By reducing the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N^{3/2})$, FMM enabled the solution of large-scale problems on relatively inexpensive computing platforms. A few years later, the idea behind FMM was extended and applied in a recursive manner, leading to the multilevel fast multipole algorithm (MLFMA) [3], which provides the solution of larger problems by reducing the complexity of MVMs to $\mathcal{O}(N \log^2 N)$ [4] or $\mathcal{O}(N \log N)$[5].

Elements of matrices obtained by discretizating integral-equation formulations can be interpreted as electromagnetic interactions between pairs of discretization elements, i.e., basis and testing functions. In MLFMA, far-field interactions between distant basis and testing functions are calculated efficiently in a group-by-group manner. A multilevel tree structure is constructed by placing the object in a cubic box and recursively dividing the computational domain into subdomains until the size of the boxes is about 0.25λ. In each MVM, three stages of MLFMA—namely, aggregation, translation, and disaggregation—are performed on the tree structure. The aggregation stage involves computating radiated fields for each nonempty box (cluster), from the lowest level to the top of the tree structure. In the lowest level, radiated fields are obtained by combining radiation patterns of the basis functions that are multiplied with the coefficients provided by the iterative algorithm. Following the aggregation stage, radiated fields are converted into incoming fields with the help of translations. Finally, during the disaggregation stage, total incoming fields propagating toward the centers of clusters are calculated from the highest level to the lowest level, where the incoming fields are finally received by the testing functions. In addition to the far-field interactions, there are also $\mathcal{O}(N)$ near-field interactions, which are calculated directly and stored in memory.

In MLFMA, radiated and incoming fields are sampled on the unit sphere as a function of spherical coordinates θ and ϕ. The number of samples required for each cluster is proportional to the size of the cluster as measured by the wavelength. Therefore, to match the different sampling rates of consecutive levels, interpolation and transpose interpolation (anterpolation) [6] are required during aggregation and disaggregation stages, respectively. There are two major ways of implementing interpolations (and anterpolations), namely, through global and local interpolation methods. Global interpolations are usually based on the fast Fourier transform (FFT) along the ϕ direction and the Legendre transform along the θ direction, performed via one-dimensional FMM [4], [7]. Using uniform sampling, FFT can also be used along the θ direction [8]. A resulting MLFMA implementation has $\mathcal{O}(N \log^2 N)$ time complexity, while interpolations are performed without error, provided that the Nyquist criterion is applied for the sampling rate. On the other hand, local interpolation methods introduce errors [9], but they lead to more efficient MLFMA implementations with $\mathcal{O}(N \log N)$ complexity [5].

In general, interpolations and anterpolations constitute the major computational bulk of MLFMA. Therefore, to obtain an efficient solver, it is extremely important to optimize the interpolation/anterpolation routines in MLFMA. In this letter, we consider local Lagrange interpolation, which is preferable due to its favorable computing cost and controllable error [5]. We present a two-step Lagrange interpolation method, which is more efficient than the conventional one-step method. Our method is based on performing the required two-dimensional interpolation as a sequence of two one-dimensional interpolations. By also applying the two-step method for anterpolations, efficiency of

Manuscript received October 16, 2008; revised November 18, 2008. First published December 16, 2008; current version published April 17, 2009. This work was supported by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grants 105E172 and 107E136, and contracts from ASELSAN and SSM.

Ö. Ergül and L. Gürel are with the Department of Electrical and Electronics Engineering and the Computational Electromagnetics Research Center (BiLCEM), Bilkent University, 06800 Bilkent, Ankara, Turkey (e-mail: lgurel@bilkent.edu.tr).

I. van den Bosch is with the Royal Military Academy, Brussels, Belgium.

Digital Object Identifier 10.1109/LAWP.2008.2011063

MLFMA is improved significantly. The decrease in computation time, i.e., the speedup, provided by the proposed two-step method is demonstrated on scattering problems involving millions of unknowns. The two-step method is easy to implement and is especially useful for problems involving large-scale objects.

II. Lagrange Interpolation

Let $f(\theta,\phi)$ be a scalar function representing a radiated or incoming field in MLFMA. Using a two-dimensional Lagrange interpolation, the value of the function at a target point (θ,ϕ) in the fine grid is obtained by using $2p\times 2p$ samples in the coarse grid, i.e.,

$$f(\theta,\phi)\approx\sum_{j=s+1-p}^{s+p} w_j(\phi)\sum_{i=t+1-p}^{t+p} v_i(\theta)f(\theta_i,\phi_j) \tag{1}$$

where $w_j(\phi)$ and $v_i(\theta)$ represent interpolation weights derived as

$$w_j(\phi)=\prod_{\substack{m=s+1-p\\ m\neq j}}^{s+p}\frac{\phi-\phi_m}{\phi_j-\phi_m} \tag{2}$$

for the ϕ direction, and

$$v_i(\theta)=\prod_{\substack{n=t+1-p\\ n\neq i}}^{t+p}\frac{\theta-\theta_n}{\theta_i-\theta_n} \tag{3}$$

for the θ direction, respectively. We note that reference indices s and t in (1)–(3) are determined by the location of the target point (θ,ϕ), with respect to the samples in the coarse grid.

In MLFMA, it is common to choose samples uniformly in the ϕ direction while using Gauss-Legendre points in the θ direction [2]. For level $l=1,\ 2,\ldots,l_{\max}$, the number of samples is

$$T_l=(L_l+1)\quad\text{and}\quad P_l=2(L_l+1) \tag{4}$$

along the θ and ϕ directions, respectively, where L_l is the truncation number determined by the excess bandwidth formula [10], i.e.,

$$L_l\approx 1.73ka_l+2.16(d_0)^{2/3}(ka_l)^{1/3}. \tag{5}$$

In (5), a_l is the box size at level l, and d_0 is the desired digits of accuracy. Interpolation of a function at $T_l\times P_l$ points requires

$$\Theta^{\text{time}}_{\text{one-step}}=4p^2T_lP_l=8p^2(L_l+1)^2 \tag{6}$$

operations. If interpolation weights in the θ and ϕ directions are combined, interpolation in (1) can be expressed as a MVM, i.e.,

$$\tilde{\boldsymbol{F}}\approx\bar{\boldsymbol{I}}_{\tilde{S}S}\cdot\boldsymbol{F} \tag{7}$$

where $\tilde{\boldsymbol{F}}$ and $\boldsymbol{F}$ are one-dimensional arrays of samples in the fine and coarse grids, respectively, and $\bar{\boldsymbol{I}}_{\tilde{S}S}$ represents an $\tilde{S}\times S$ sparse interpolation matrix. For interpolations from level $(l-1)$ to l, $\tilde{S}=2(L_l+1)^2$, $S=2(L_{l-1}+1)^2$, and there are $4p^2$ nonzero elements per row in $\bar{\boldsymbol{I}}_{\tilde{S}S}$. The matrix representation in (7) is preferred due to its simplicity, and it is very useful for an easy implementation of anterpolations in MLFMA. However, the amount of memory required for the interpolation matrix is proportional to

$$\Theta^{\text{memory}}_{\text{matrix}}=4p^2T_lP_l=8p^2(L_l+1)^2 \tag{8}$$

which can be significant for large problems. Considering the original form in (1), it is possible to store the interpolation weights along the θ and ϕ directions separately, in two arrays of sizes $2p(L_l+1)$ and $4p(L_l+1)$, respectively. Then, the total memory used for interpolations from level $(l-1)$ to l becomes

$$\Theta^{\text{memory}}_{\text{array}}=(2p+4p)(L_l+1)=6p(L_l+1) \tag{9}$$

without any change in the number of operations and processing time. The reduction in memory by using the array representation instead of the matrix representation is

$$\frac{\Theta^{\text{memory}}_{\text{array}}}{\Theta^{\text{memory}}_{\text{matrix}}}=\frac{6p(L_l+1)}{8p^2(L_l+1)^2}=\frac{3}{4p(L_l+1)} \tag{10}$$

which is especially significant for higher levels.

III. Two-Stage Lagrange Interpolation

The number of operations required for the conventional (one-step) interpolation method from level $(l-1)$ to l is $8p^2(L_l+1)^2$. This is because there are $(L_l+1)\times 2(L_l+1)$ points in the fine grid (samples for level l) and each of these points has $4p^2$ contributions from the coarse grid. On the other hand, locations of sampling points in the θ and ϕ directions are independent of each other. Therefore, interpolations along the two directions can be performed consecutively, as follows:

- Perform an interpolation along the θ direction as

$$f(\theta,\phi_j)\approx\sum_{i=t+1-p}^{t+p} v_i(\theta)f(\theta_i,\phi_j) \tag{11}$$

  which requires $2pT_lP_{l-1}$ operations.
- Perform an interpolation along the ϕ direction using the result of the first step, i.e.,

$$f(\theta,\phi)\approx\sum_{j=s+1-p}^{s+p} w_j(\phi)f(\theta,\phi_j). \tag{12}$$

  This step requires $2pT_lP_l$ operations.

Consequently, using the two-step method, the processing time required to interpolate the function at $T_l\times P_l$ points is

$$\begin{aligned}\Theta^{\text{time}}_{\text{two-step}}&=2pT_l(P_{l-1}+P_l)\\ &=4p(L_l+1)(L_l+L_{l-1}+2).\end{aligned} \tag{13}$$

Comparing processing times required for the one-step and two-step interpolation methods,

$$\begin{aligned}\frac{\Theta^{\text{time}}_{\text{two-step}}}{\Theta^{\text{time}}_{\text{one-step}}}&=\frac{4p(L_l+1)(L_l+L_{l-1}+2)}{8p^2(L_l+1)^2}\\ &<\frac{8p(L_l+1)(L_l+1)}{8p^2(L_l+1)^2}=\frac{1}{p}\end{aligned} \tag{14}$$

TABLE I
PROCESSING TIME REQUIRED FOR AN AGGREGATION STAGE AND FOR AN MVM WHEN INTERPOLATION/ANTERPOLATION OPERATIONS ARE PERFORMED BY USING ONE-STEP AND TWO-STEP INTERPOLATION METHODS

SPHERE PROBLEMS			AGGREGATION			MATRIX-VECTOR MULTIPLICATION		
Radius (λ)	Mesh (mm)	Unknowns	One-Step (s)	Two-Step (s)	Reduction	One-Step (s)	Two-Step (s)	Reduction
6	5	132,000	0.646	0.360	44%	1.98	1.47	26%
7.5	4	206,499	0.844	0.470	44%	2.52	1.86	26%
10	3	367,821	2.18	1.19	45%	6.21	4.46	28%
15	2	829,881	3.86	2.14	45%	11.1	8.00	28%
20	1.5	1,462,854	9.47	5.28	44%	27.0	19.6	27%
30	1	3,319,524	17.1	9.43	45%	47.0	33.4	29%
40	0.75	5,851,416	40.7	22.3	45%	112	79.2	29%
48	0.625	8,447,808	54.7	30.8	44%	152	109	28%

since $L_{l-1} < L_l$. Therefore, the two-step method is always faster than the one-step method. To store the intermediate array of size $P_{l-1} = 2(L_{l-1}+1)$ between the steps, the two-step method requires a bit more memory than is used in the one-step method, i.e.,

$$\frac{\Theta^{\text{memory}}_{\text{array+two-step}}}{\Theta^{\text{memory}}_{\text{array+one-step}}} = \frac{6p(L_l+1)+2(L_{l-1}+1)}{6p(L_l+1)} < \frac{6p+2}{6p} = 1+\frac{1}{3p}. \quad (15)$$

Nevertheless, the speedup in the two-step method more than compensates for the small increase in memory.

IV. RESULTS

To demonstrate the acceleration provided by the two-step interpolation method, we present the solution of scattering problems involving perfectly conducting spheres of various radii from 6λ to 48λ illuminated by a plane wave. Problems are formulated with the combined-field integral equation and discretized with Rao-Wilton-Glisson (RWG) [11] basis functions. Triangulations with $\lambda/10$ mesh size lead to large matrix equations involving 132,003 to 8,447,808 unknowns. Problems are solved iteratively, with MVMs performed efficiently by MLFMA. Solutions are parallelized into 16 processes on a cluster of AMD Opteron 870 processors. The hierarchical partitioning strategy is used for the efficient parallelization of MLFMA [12]. Far-field interactions are calculated with two digits of accuracy and the interpolation/anterpolation operations are performed using 6×6 stencils ($p = 3$). Table I lists the processing time required for the aggregation stage, in addition to the speedup offered by the two-step interpolation method. Compared to the conventional one-step method, the two-step method reduces the processing time of the aggregation stage by about 45%. To demonstrate the overall improvement, Table I also presents the processing times required for MVMs, which are reduced by 25–30% with the two-step interpolation method. For the largest problem, MVM time is reduced from 152 to 109 seconds.

V. CONCLUSION

We present a two-step Lagrange interpolation method to accelerate the solution of electromagnetics problems with MLFMA. This method is easily implemented by decomposing the conventional one-step method into two successive parts. Acceleration provided by the two-step method is significant, and it is especially useful for large problems.

REFERENCES

[1] V. Rokhlin, "Rapid solution of integral equations of scattering theory in two dimensions," *J. Comput. Phys.*, vol. 86, no. 2, pp. 414–439, Feb. 1990.
[2] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propag. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993.
[3] 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.
[4] M. F. Gyure and M. A. Stalzer, "A prescription for the multilevel Helmholtz FMM," *IEEE Comput. Sci. Eng.*, vol. 5, no. 3, pp. 39–47, Jul.-Sep. 1998.
[5] W. C. Chew, J.-M Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston, MA: Artech House, 2001.
[6] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernels," *Comp. Phys. Commun.*, vol. 65, pp. 24–38, Apr. 1991.
[7] R. Jacob-Chien and B. K. Alpert, "A fast spherical filter with uniform resolution," *J. Comput. Phys.*, vol. 136, no. 2, pp. 580–584, Sep. 1997.
[8] J. Sarvas, "Performing interpolation and anterpolation entirely by fast Fourier transform in the 3-D multilevel fast multipole algorithm," *SIAM J. Numer. Anal.*, vol. 41, no. 6, pp. 2180–2196, Nov. 2003.
[9] O. M. Bucci, C. Gennarelli, and C. Savarese, "Optimal interpolation of radiated fields over a sphere," *IEEE Trans. Antennas Propag.*, vol. 39, no. 11, pp. 1633–1643, Nov. 1991.
[10] S. Koc, J. M. Song, and W. C. Chew, "Error analysis for the numerical evaluation of the diagonal forms of the scalar spherical addition theorem," *SIAM J. Numer. Anal.*, vol. 36, no. 3, pp. 906–921, 1999.
[11] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. AP-30, no. 3, pp. 409–418, May 1982.
[12] 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.