A Weak Scalability Study of the Parallel Computation of the Translation Operator in the MLFMA

B. Michiels*   I. Bogaert*   J. Fostier*   D. De Zutter*

Abstract —

This paper studies the weak scaling behavior of the parallel computation of the translation operator in the three-dimensional (3D) Multilevel Fast Multipole Algorithm (MLFMA). First, two algorithms and their serial complexities are investigated. Then, the parallelization of these two algorithms and the arising issues regarding the complexity are discussed.

1   INTRODUCTION

To solve Maxwell's equations the Method of Moments (MoM), combined with the Multilevel Fast Multipole Algorithm (MLFMA), can be used. The MLFMA relies on Gegenbauer's addition theorem, which factorizes the Green's function into $L+1$ multipoles, allowing the interactions between the discretized segments of the MoM to be described by means of radiation patterns and translations [1].

The subject of interest in this paper is the translation operator of the MLFMA in three dimensions (3D). In the 3D-MLFMA the translation operator is

$$T(\vec{k}, \vec{R}_T) = \sum_{l=0}^{L} (-j)^l (2l+1) h_l^{(2)}(kR_T) P_l(\cos(\theta_T)) \quad (1)$$

with

$$\theta_T = \arccos\left(\vec{1}_k \cdot \vec{1}_{R_T}\right) \quad (2)$$

$P_l(.)$ and $h_l^{(2)}(.)$ stand for the Legendre polynomial and spherical Hankel function of the second kind of order l respectively. $\vec{R}_T$ is the translation direction, while $\vec{k}$ denotes a vector in the k-space of the radiation pattern.

Fig. 1 shows a radiation pattern that is uniformly sampled in the θ- and ϕ-directions. The sampling rate of each dimension is of the order of L, hence the time and memory complexities for the calculation of a radiation pattern are $\mathcal{O}(L^2)$.

The translation has a rotational symmetry axis, which is determined by $\vec{R}_T$, and therefore it depends only on θ_T and not on ϕ_T (see Fig. 2). The

*Department of Information Technology (INTEC), Ghent University, Ghent, B-9000, Belgium http://emweb.intec.ugent.be

sampling rate of the θ_T-direction is also $\mathcal{O}(L)$ and for each θ_T-sampling point $L+1$ multipoles have to be computed. Therefore the computation of a translation operator requires $\mathcal{O}(L^2)$ number of operations.

For each level one goes up in the MLFMA-tree, the value of L increases approximately by a factor of two. In that case, the size of a radiation pattern and the number of operations to calculate a translation grows roughly by a factor of four.

In order to be able to perform larger MoM-MLFMA simulations, multiple CPU cores can be employed. A parallel algorithm can be assessed by means of its "weak" scaling behavior when both the problem size N and the number of processes P are proportionally increased. An algorithm with a serial complexity of $\mathcal{O}(C(N))$ is weakly scalably parallelized when the complexity of each individual process is not higher than $\mathcal{O}(C(N)/P)$ or, equivalently, $\mathcal{O}(C(N)/N)$, as $P = \mathcal{O}(N)$.

A scalable parallel matrix-vector multiplication in the 3D-MLFMA can be achieved using the so-called Blockwise Hierarchical Partitioning Scheme (B-HiP). When going one level up in the MLFMA-tree that uses the B-HiP, the number of partitions P is increased by a factor of four [2] and each partition is assigned to an individual process. Hence, P is of the order of L^2, just like the radiation pattern and the translation operator. The blockwise partitioning of a radiation pattern is shown Fig. 1. Both the θ- and ϕ-direction are divided in $\sqrt{P}$ parts, hence the length of the θ- and ϕ-range per partition are $\mathcal{O}(1)$, i.e. independent of L. As a result, the size of each radiation pattern partition does not exceed $\mathcal{O}(1)$ [3].

2   CALCULATION OF THE TRANSLATION OPERATOR

The values of $T(\vec{k}, \vec{R}_T)$ have to be calculated in each sampling point of the radiation pattern. This paper discusses two methods: the direct calculation of the translation and the calculation of the translation using interpolations.
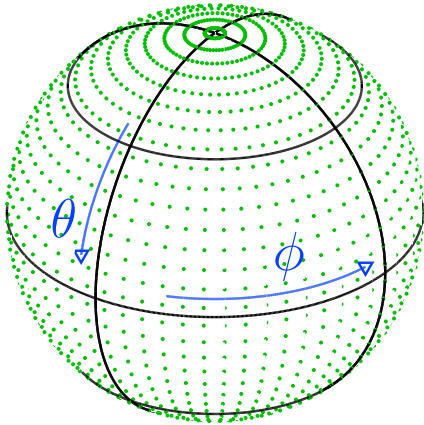
Figure 1: Blockwise partitioning of a uniformly sampled radiation pattern. The green dots denote the sampling points, the black lines correspond to the different partitions.
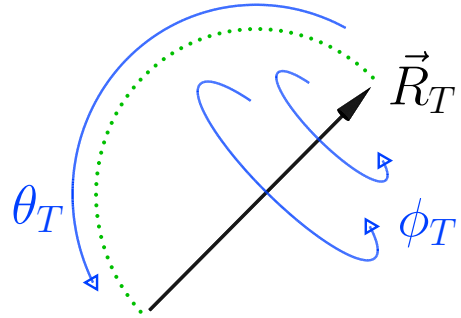

Figure 2: Representation of the translation operator. The translation is axis-symmetrical (i.e. independent of ϕ_T) and depends only on θ_T.

2.1 Direct method

The most direct computation of the translation operator is to evaluate expression (1) in every radiation pattern sampling point. However, this requires $\mathcal{O}(L^3)$ number of operations, as the size of a radiation pattern is $\mathcal{O}(L^2)$ and (1) contains a summation of the contributions of the $L+1$ multipoles.

2.2 Interpolation method

An alternative method is to make use of local interpolations [4] that use for instance Lagrange polynomials [5]. First, (1) is computed in every θ_T-sampling point. To perform this step $\mathcal{O}(L^2)$ number of operations are needed. Then, $T(\vec{k}, \vec{R}_T)$ is calculated in each sampling point of the radiation pattern, employing a local interpolator. As the number of required neighboring sampling points does not exceed $\mathcal{O}(1)$, this step also has a complexity of $\mathcal{O}(L^2)$. As a result, the calculation of a translation can be performed with a time complexity of $\mathcal{O}(L^2)$. Therefore, this method is superior to the direct method from a complexity point of view.

3 PARALLELIZATION

In this section the parallelization of the two methods, discussed in the previous section, is investigated. Both the problem size N, i.e. the size of the radiation pattern and the complexity of the translation operator, and the number of partitions P are $\mathcal{O}(L^2)$. The objective of the parallel algorithm is

to calculate the value of $T(\vec{k}, \vec{R}_T)$ in each sampling point of the partition of the radiation pattern.

3.1 Direct method

As mentioned above, the size of each radiation pattern partition is $\mathcal{O}(1)$ and for each evaluation of (1) a series of $L+1$ contributions needs to be computed. As a result, the complexity of this parallel algorithm is $\mathcal{O}(L)$.

The described algorithm is indeed a scalable parallelization of the *direct method*, but of course it is not a scalable way to calculate the translation operator, as the direct method does not yield the lowest possible complexity to compute a translation.

3.2 Interpolation method

The parallelization of the first step of the interpolation method is straightforward. Both the multipoles and the θ_T-sampling points can be partitioned in a blockwise way, similar to the blockwise partitioning of the radiation pattern, depicted in Fig. 1. This means that the $L+1$ multipoles and $\mathcal{O}(L)$ θ_T-sampling points are divided in $\sqrt{P}$ parts. Each partition of the translation contains $\mathcal{O}(1)$ multipoles and $\mathcal{O}(1)$ θ_T-sampling points, resulting in a $\mathcal{O}(1)$ complexity per translation partition.

In order to obtain a $\mathcal{O}(1)$ complexity per translation partition, it is essential to be able to compute Hankel functions and Legendre polynomials of an arbitrary order l in $\mathcal{O}(1)$ operations. Recently, such a method for the Legendre polynomials has been developed [6]. The Hankel functions can be

evaluated with an $\mathcal{O}(1)$ complexity using the Amos libraries [7].

In the second step, the local interpolations have to be performed in the sampling points of each partition. The number of neighboring θ_T-points is $\mathcal{O}(1)$, as is the size of a radiation pattern. Therefore this step also has a complexity of $\mathcal{O}(1)$.

However, in order to execute the local interpolation, the required θ_T-sampling points have to be communicated. From Fig. 1 one can see that the sampling points of a radiation pattern accumulate at the poles. As a result, some θ_T-sampling points need to be communicated to more radiation pattern partitions than others, in order to do the interpolation. Consequently, the translation partitions that contain these θ_T-points need to send their data to more radiation pattern partitions. This can endanger the weak scalability of the parallelization if the communication complexity is higher than $\mathcal{O}(1)$.

Figure 3: Normalized distribution of the radiation pattern sampling points as a function of θ_T for $\alpha_T = \pi/4$ and $L = 300$.

4 DISTRIBUTION

In the previous section we saw that the communication step in the parallelization of the interpolation method is strongly dependent on the orientation of the translation direction $\vec{R}_T$ with respect to the z-axis of the radiation pattern. If the azimuth angle of $\vec{R}_T$ is α_T, then the poles of the radiation pattern correspond to $\theta_T = \alpha_T$ and $\theta_T = \pi - \alpha_T$ in the coordinate system of the translation.

The distribution of the radiation pattern sampling points as a function of θ_T shows how many times a particular θ_T-sampling point and its neighboring sampling points are needed for the local interpolation. As an example, $\alpha_T = \pi/4$ is considered and L is set equal to 300. The sampling rates of the θ-, ϕ- and θ_T-direction are chosen equal to $N_\theta = L+1$, $N_\phi = 2L+4$, $N_{\theta_T} = 4L+8$. The distribution of the radiation pattern sampling points as a function of θ_T is shown in Fig. 3. The distribution is normalized such that $\int_0^\pi f(\theta_T)d\theta_T = 2\pi^2$, as θ ranges from 0 to π and ϕ from 0 to 2π. As one can see, the two peaks of the distribution, which correspond to the radiation pattern sampling points at the poles, are indeed located at $\theta_T = \alpha_T$ and $\theta_T = \pi - \alpha_T$.

The shape of the distribution shows that the communication of the θ_T-sampling points is very unbalanced for the different translation partitions, which can endanger the weak scalability of the parallelization of the interpolation method. Further research aims at solving this problem.
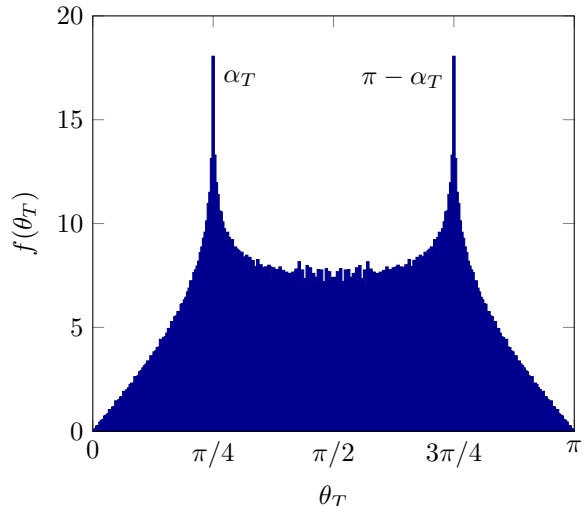
5 CONCLUSION

In this paper the parallelization of the calculation of the translation operator in the 3D-MLFMA and its weakly scaling behavior is studied. First, two serial algorithms to compute the translation operator, the direct method and the interpolation method, are discussed. The direct method has a complexity of the order of L^3, while the interpolation method requires only $\mathcal{O}(L^2)$ operations. Then, the parallelization of these two methods is considered. In case of the direct method, a straightforward partitioning yields a $\mathcal{O}(L)$ complexity per process. For the interpolation method however, the communication complexity is strongly dependent on the translation direction, which determines the distribution of the radiation pattern sampling points as a function of θ_T. From the shape of the distribution one can deduce that the communication is unbalanced, which makes the development of a weakly scalable parallel computation of the translation a challenging problem, needing further research.

Acknowledgment

References

[1] W.C. Chew, J. Jin, E. Michielssen and J. Song, "Fast and Efficient Algorithms in Computational Electromagnetics", Artech House, 2001.

[2] Ö. Ergül and L. Gürel, "A Hierarchical Partitioning Strategy for an Efficient Parallelization of the Multilevel Fast Multipole Algorithm", IEEE Transactions on Antennas and Propagation, vol. 57, no. 6, pp. 1740–1750, June 2009

[3] B. Michiels, J. Fostier, I. Bogaert and D. De Zutter, "Weak Scalability Analysis of the Distributed-Memory Parallel MLFMA", IEEE Transactions on Antennas and Propagation, submitted for publication, 2013.

[4] J. Song and W.C. Chew, "Interpolation Of Translation Matrix In MLFMA", Microwave And Optical Technology Letters, vol. 30, no. 2, pp. 109–114, July 2001.

[5] Ö. Ergül and L. Gürel, "Optimal Interpolation of Translation Operator in Multilevel Fast Multipole Algorithm", IEEE Transactions on Antennas and Propagation, vol. 54, no. 12, pp. 3822–3826, December 2006.

[6] I. Bogaert, B. Michiels and J. Fostier, "$\mathcal{O}(1)$ Computation Of Legendre Polynomials And Gauss-Legendre Nodes And Weights For Parallel Computing", SIAM Journal on Scientific Computing, vol. 34, no. 3, pp. C83–C101, 2012.

[7] D.E. Amos, "A portable package for Bessel functions of a complex argument and nonnegative order", ACM Transactions on Mathematical Software (TOMS), vol. 12, no. 3, pp. 265–273, September 1986.