Commun. Comput. Phys.
doi: 10.4208/cicp.071209.240310a

Vol. **8**, No. 5, pp. 1183-1207
November 2010

A Low Frequency Vector Fast Multipole Algorithm with Vector Addition Theorem

Yang G. Liu[1] and Weng Cho Chew[2,*]

[1] *Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong.*
[2] *Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong. On Leave of Absence from the University of Illinois, Urbana-Champaign.*

Received 7 December 2009; Accepted (in revised version) 24 March 2010

Communicated by Wei Cai

Available online 23 June 2010

Abstract. In the low-frequency fast multipole algorithm (LF-FMA) [19,20], scalar addition theorem has been used to factorize the scalar Green's function. Instead of this traditional factorization of the scalar Green's function by using scalar addition theorem, we adopt the vector addition theorem for the factorization of the dyadic Green's function to realize memory savings. We are to validate this factorization and use it to develop a low-frequency vector fast multipole algorithm (LF-VFMA) for low-frequency problems. In the calculation of non-near neighbor interactions, the storage of translators in the method is larger than that in the LF-FMA with scalar addition theorem. Fortunately it is independent of the number of unknowns. Meanwhile, the storage of radiation and receiving patterns is linearly dependent on the number of unknowns. Therefore it is worthwhile for large scale problems to reduce the storage of this part. In this method, the storage of radiation and receiving patterns can be reduced by 25 percent compared with the LF-FMA.

AMS subject classifications: 65R20, 78A45, 78M16

PACS: 02.70.Pt, 03.50.De, 89.20.Ff

Key words: LF-VFMA, electromagnetics, low frequency, loop-tree basis, memory saving, surface integral equation, vector addition theorem.

1 Introduction

To meet real needs of society, more complex computational algorithms are needed to numerically simulate and analyze more complex problems. As a popular way to perform

*Corresponding author. *Email addresses:* liuyang@eee.hku.hk (Y. G. Liu), wcchew@hku.hk (W. C. Chew)

complex computational algorithms, modern computational electromagnetics has made great strides forward. However, since the finite power of computers limits the size of problems which can be solved with computational algorithms, computational electromagnetics is confronted with the issue of CPU time usage and memory requirements.

Many numerical modelings of physical phenomena or systems often result in matrix equations, in which matrices are dense. As direct inversion methods require $\mathcal{O}(N^2)$ memory and $\mathcal{O}(N^3)$ central processing unit time, where N is the number of unknowns for solving the problem, it is not suitable to solve large scale problems with direct inversion methods. Hence, iterative solvers for matrix equations have been developed. The bottleneck of iterative solvers is the matrix-vector product. Since, for traditional iterative solvers, the memory and computational complexities of the matrix-vector product scale as $\mathcal{O}(N^2)$, it is still not efficient for solving large scale problems by using traditional iterative solvers. In recent decades, fast-multipole-like algorithms [10,12,18,20,21] have been developed to accelerate the matrix-vector product. Such matrix-vector product can be performed in $\mathcal{O}(N)$ operations or $\mathcal{O}(N\log N)$ operations per iteration depending on the problem. Moreover, in these methods, the memory complexity is the same as the computational complexity. For example, the memory requirement and the number of floating point operations per iteration of the low-frequency multilevel fast multipole algorithm (LF-MLFMA) are both of $\mathcal{O}(N)$. Up to now, since many fast algorithms are quite mature and CPU time and memory usage in fast multipole algorithms scale as

$$\text{Time} \approx C_t N\log N, \qquad \text{Memory} \approx C_m N\log N, \tag{1.1}$$

it is meaningful for large problems to gain efficiency by reducing the constant C_t or C_m in front of the scaling formulas.

Electromagnetic simulations in the low frequency regime are important issues, where the objects or parts can be a tiny fraction of wavelength. Such simulations are often encountered in analyzing electromagnetic phenomena in circuits and antennas. With increasing complexity of circuits or antennas, it is necessary to improve the ability of fast solvers for handling large-scale problems at low frequencies. For achieving this aim, one way is to enhance the memory efficiency of fast solvers. To develop efficient fast solvers for low-frequency large-scale problems, we start by studying the electric field integral equation operator [27]

$$\mathcal{L}_E(\mathbf{J}) = i\omega\mu \int_S g(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')\mathrm{d}\mathbf{r}' - \frac{1}{i\omega\epsilon}\nabla \int_S g(\mathbf{r},\mathbf{r}')\nabla'\cdot\mathbf{J}(\mathbf{r}')\mathrm{d}\mathbf{r}', \tag{1.2}$$

where $\mathbf{J}(\mathbf{r})$ is the surface current on the surface S. The first term is due to the vector potential and it corresponds to the electric field generated by a time varying magnetic field. Its order is $\mathcal{O}(\omega)$. The second term is due to the scalar potential, corresponding to the electric field produced by the charge in the system. Its order is $\mathcal{O}(\omega^{-1})$. When the frequency $\omega \to 0$, the contribution from the vector potential will be lost in the numerical simulation due to finite machine precision. Then Eq. (1.2) will only have the scalar potential part.