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 $\omega\rightarrow 0$, 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.

The integral operator corresponding to the scalar potential part has a null space, which is due to the divergence operator involved. This makes the impedance matrix nearly singular and the matrix equation is difficult to solve [5,7]. Although the vector potential part is much smaller than the scalar potential part, the two parts are equally important at low frequencies. The loss of the contribution from the vector potential makes the solution inaccurate. So an effective numerical method needs to capture the contributions of both parts simultaneously when the frequency goes to zero.

To overcome the problems, most popular methods are based on the loop-tree or loop-star decomposition [5,7,9,11,13,14,17], which can separate the contribution of the vector potential part from that of the scalar potential part. After using the loop-tree or loop-star basis, frequency normalization can be applied to balance the nearly singular impedance matrix at very low frequencies. But it should be noted that the achieved matrix is still ill-conditioned. This is traced to the fact that the divergence of the Rao-Wilton-Glisson (RWG) basis [6] is not suitable for representing charges, which is contained in the calculation of the scalar potential part. The basis rearrangement can be adopted to overcome this problem [20]. By doing so, the spectral property of the impedance matrix can be improved and the iteration count can be reduced dramatically. However, numerical simulations show that iterative solvers fail to converge even with the basis rearrangement when the number of unknowns becomes very large. It is caused by the accumulated numerical error due to subtractions involved in the basis rearrangement. To avoid this numerical error, the patch-pair basis, constructed from the single-patch basis directly, is proposed to express charges [22].

Based on the above ideas, the low-frequency fast multipole algorithm (LF-FMA) has been developed to numerically solve low frequency problems. In this method, the scalar addition theorem has been used to factorize the scalar Green's function. In this paper, 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. For large scale problems, the computation and storage of non-near neighbor interactions are the main parts in the whole simulation process. A low-frequency vector fast multipole algorithm (LF-VFMA) is developed by using the factorization of the dyadic Green's function with the vector addition theorem to save memory for non-near neighbor interactions compared with the LF-FMA. Since the computational complexity of the LF-VFMA is $\mathcal{O}(N)$, it still accelerates the computation compared with traditional iterative solvers. Since vector translators of the LF-VFMA can be expressed by using scalar translators of the LF-FMA, theoretically we can only store scalar translators and the storage of the translator part of the LF-VFMA is the same as that of the LF-FMA. However, to make the LF-VFMA numerically applicable to low-frequency problems, some elements of vector translators should be specially handled and stored. In addition, if we only store scalar translators and these special elements for calculating vector translators in the iteration process, repeated calculations will increase floating point operations dramatically for large scalar problems. So, except for scalar translators and special elements, we can also store some fundamental elements of vector translators to reduce part of repeated calculations, which is at the expense of raising the storage for vector translators. Therefore, the storage for vector translators is larger than that of scalar translators of the LF-FMA. Fortunately, the storage for vector translators is independent of the number of unknowns. Meanwhile, the storage of radiation and receiving patterns in the LF-VFMA can be reduced by 25 percent compared with that of the LF-FMA. As the storage of radiation and receiving patterns depends on the number of unknowns, it becomes the main part of the total storage with the increasing scale of problems. Hence it is effective for large scale problems to reduce the storage of radiation and receiving patterns.

In this paper, we introduce the factorization of the dyadic Green's function in terms of a set of vector multipole fields in Section 2. Section 3 is to briefly review general equations. In Section 4, based on the factorization of the dyadic Green's function given in Section 2, the LF-VFMA is proposed and memory requirement is also discussed. Section 5 shows the computational complexity of the LF-VFMA. Some numerical examples are given in Section 6 to verify the capability of the algorithm. Conclusions are presented in Section 7.

2 Factorization of dyadic Green's function

In this section, we will present the factorization of the dyadic Green's function in terms of two sets of vector multipole fields. It will be further explained that only one set of vector multipole fields can be used to factorize the dyadic Green's function in developing the LF-VFMA for solving low frequency problems.

First, two sets of vector spherical harmonics will be introduced. One set of vector spherical harmonics called Hansen spherical harmonics can be denoted as [16]

$$\mathbf{P}_{J,M}(\theta,\phi)=\mathbf{e}_r Y_{J,M}(\theta,\phi), \tag{2.1a}$$

$$\mathbf{B}_{J,M}(\theta,\phi)=\frac{\nabla_s Y_{J,M}(\theta,\phi)}{\sqrt{J(J+1)}}, \tag{2.1b}$$

$$\mathbf{C}_{J,M}(\theta,\phi)=-\mathbf{e}_r\times\mathbf{B}_{J,M}(\theta,\phi), \tag{2.1c}$$

where the operator ∇_s is defined as

$$\nabla_s=\mathbf{e}_\theta\frac{\partial}{\partial\theta}+\frac{\mathbf{e}_\phi}{\sin\theta}\frac{\partial}{\partial\phi}, \tag{2.2}$$

and $Y_{J,M}$ is the spherical harmonics [2] ($J=0,1,2,\cdots$, and $M=-J,\cdots,-1,0,1,\cdots,J$), which is given by

$$Y_{J,M}(\theta,\phi)=\sqrt{\frac{(J-M)!}{(J+M)!}\frac{2J+1}{4\pi}}P_J^M(\cos\theta)e^{iM\phi},\quad M\geq 0, \tag{2.3}$$

and

$$Y_{J,-M}(\theta,\phi)=(-1)^M Y^*_{J,M}(\theta,\phi),\quad M>0.$$

Here, $P_J^M(x)$ is the associated Legendre functions of the first kind. The above is defined in spherical coordinates, which is denoted by $\mathbf{r}=(r,\theta,\phi)$. The unit vectors are

$$\mathbf{e}_r=(\sin\theta\cos\phi,\ \sin\theta\sin\phi,\ \cos\theta)^T, \tag{2.4a}$$

$$\mathbf{e}_\theta=(\cos\theta\cos\phi,\ \cos\theta\sin\phi,\ -\sin\theta)^T, \tag{2.4b}$$

$$\mathbf{e}_\phi=(-\sin\phi,\ \cos\phi,\ 0)^T. \tag{2.4c}$$

Another set of vector spherical harmonics is denoted as [16,24]

$$\mathbf{Y}^{[J]}_{J,M}(\theta,\phi)=\frac{i}{\sqrt{J(J+1)}}\nabla_s Y_{J,M}(\theta,\phi)\times\mathbf{e}_r, \tag{2.5a}$$

$$\mathbf{Y}^{[J]}_{J-1,M}(\theta,\phi)=\frac{\sqrt{J}}{\sqrt{2J+1}}Y_{J,M}(\theta,\phi)\mathbf{e}_r+\frac{1}{\sqrt{2J+1}\sqrt{J}}\nabla_s Y_{J,M}(\theta,\phi), \tag{2.5b}$$

$$\mathbf{Y}^{[J]}_{J+1,M}(\theta,\phi)=\frac{1}{\sqrt{2J+1}\sqrt{J+1}}\nabla_s Y_{J,M}(\theta,\phi)-\frac{\sqrt{J+1}}{\sqrt{2J+1}}Y_{J,M}(\theta,\phi)\mathbf{e}_r. \tag{2.5c}$$

In electromagnetics and elastodynamics, in order to describe the divergence and curl properties of the vector fields, Hansen multipole fields $\mathbf{M}_{J,M}(\mathbf{r})$, $\mathbf{N}_{J,M}(\mathbf{r})$ and $\mathbf{L}_{J,M}(\mathbf{r})$ are often used, which can be defined as [1,16,24]

$$\mathbf{M}_{JM}(\mathbf{r})=\nabla\times\left[rz_J(kr)\mathbf{P}_{J,M}(\theta,\phi)\right], \tag{2.6a}$$

$$\mathbf{N}_{JM}(\mathbf{r})=\frac{1}{k}\nabla\times\mathbf{M}_{JM}(\mathbf{r}), \tag{2.6b}$$

$$\mathbf{L}_{JM}(\mathbf{r})=\frac{1}{k}\nabla\left[z_J(kr)Y_{J,M}(\theta,\phi)\right], \tag{2.6c}$$

where $z_J(x)$ is the first kind spherical Hankel function of order J, i.e.,

$$z_J(x)=h_J^{(1)}(x),$$

see [4]. We can also express Hansen multipole fields in terms of the other set of vector spherical harmonics [24]

$$\mathbf{M}_{JM}(\mathbf{r})=-i\sqrt{J(J+1)}z_J(kr)\mathbf{Y}^{[J]}_{J,M}(\theta,\phi), \tag{2.7a}$$

$$\mathbf{N}_{JM}(\mathbf{r})=-\frac{J\sqrt{J+1}}{\sqrt{2J+1}}z_{J+1}(kr)\mathbf{Y}^{[J]}_{J+1,M}(\theta,\phi)+\frac{(J+1)\sqrt{J}}{\sqrt{2J+1}}z_{J-1}(kr)\mathbf{Y}^{[J]}_{J-1,M}(\theta,\phi), \tag{2.7b}$$

$$\mathbf{L}_{JM}(\mathbf{r})=\frac{\sqrt{J+1}}{\sqrt{2J+1}}z_{J+1}(kr)\mathbf{Y}^{[J]}_{J+1,M}(\theta,\phi)+\frac{\sqrt{J}}{\sqrt{2J+1}}z_{J-1}(kr)\mathbf{Y}^{[J]}_{J-1,M}(\theta,\phi). \tag{2.7c}$$

For the sake of simplifying the expression of the dyadic Green's function in terms of Hansen multipole fields, we scale $\mathbf{M}_{JM}(\mathbf{r})$ and $\mathbf{N}_{JM}(\mathbf{r})$ as

$$\mathbf{m}_{JM}(\mathbf{r})=\frac{i}{\sqrt{J(J+1)}}\mathbf{M}_{JM}(\mathbf{r})=z_J(kr)\mathbf{Y}^{[J]}_{J,M}(\theta,\phi), \tag{2.8a}$$

$$\begin{aligned}\mathbf{n}_{JM}(\mathbf{r}) &= \frac{i}{\sqrt{J(J+1)}}\mathbf{N}_{JM}(\mathbf{r})\\ &= -\frac{i\sqrt{J}}{\sqrt{2J+1}}z_{J+1}(kr)\mathbf{Y}^{[J]}_{J+1,M}(\theta,\phi)+\frac{i\sqrt{J+1}}{\sqrt{2J+1}}z_{J-1}(kr)\mathbf{Y}^{[J]}_{J-1,M}(\theta,\phi). \end{aligned} \tag{2.8b}$$

By letting

$$\overline{\mathbf{\Psi}}^t_{J,M}(\mathbf{r}) = \left[\mathbf{m}_{J,M}(\mathbf{r}),\mathbf{n}_{J,M}(\mathbf{r}),\mathbf{L}_{J,M}(\mathbf{r})\right] \in \mathbb{C}_{3\times 3}, \tag{2.9}$$

the vector addition theorem based on the Hansen multipole fields can be described as follows [3,8,15]

$$\overline{\mathbf{\Psi}}^t_{J,M}(\mathbf{r})_{3\times 3} = \sum_{J',M'} \mathfrak{R}g\overline{\mathbf{\Psi}}^t_{J',M'}(\mathbf{r}')_{3\times 3}\cdot\overline{\alpha}_{J'M',JM}(\mathbf{r}'')_{3\times 3}, \qquad |\mathbf{r}'|<|\mathbf{r}''|, \tag{2.10a}$$

$$\overline{\mathbf{\Psi}}^t_{J,M}(\mathbf{r})_{3\times 3} = \sum_{J',M'} \overline{\mathbf{\Psi}}^t_{J',M'}(\mathbf{r}')_{3\times 3}\cdot\overline{\beta}_{J'M',JM}(\mathbf{r}'')_{3\times 3}, \qquad |\mathbf{r}'|>|\mathbf{r}''|, \tag{2.10b}$$

$$\mathfrak{R}g\overline{\mathbf{\Psi}}^t_{J,M}(\mathbf{r})_{3\times 3} = \sum_{J',M'} \mathfrak{R}g\overline{\mathbf{\Psi}}^t_{J',M'}(\mathbf{r}')_{3\times 3}\cdot\overline{\beta}_{J'M',JM}(\mathbf{r}'')_{3\times 3}, \quad \forall\, |\mathbf{r}'|,\, |\mathbf{r}''|, \tag{2.10c}$$

where vectors $\mathbf{r}$, $\mathbf{r}'$ and $\mathbf{r}''$ satisfy

$$\mathbf{r}=\mathbf{r}'+\mathbf{r}'', \tag{2.11}$$

and the $\mathfrak{R}g$ operator implies taking the regular part of the function where a spherical Hankel function is replaced by a spherical Bessel function. The subscript $J=1,\cdots,\infty$ and $-J\le M\le J$. If the summation is truncated at $J=J_{\max}$, then the number of terms involved is

$$P=(J_{\max}+1)^2-1.$$

The translator $\overline{\alpha}_{J'M',JM}(\mathbf{r}'')$ has the form

$$\begin{bmatrix} \sqrt{\frac{J'(J'+1)}{J(J+1)}}A_{J'M',JM}(\mathbf{r}'') & \sqrt{\frac{J'(J'+1)}{J(J+1)}}B_{J'M',JM}(\mathbf{r}'') & 0\\ \sqrt{\frac{J'(J'+1)}{J(J+1)}}B_{J'M',JM}(\mathbf{r}'') & \sqrt{\frac{J'(J'+1)}{J(J+1)}}A_{J'M',JM}(\mathbf{r}'') & 0\\ 0 & 0 & \alpha_{J'M',JM}(\mathbf{r}'') \end{bmatrix}, \tag{2.12}$$

where $\alpha_{J'M',JM}$ is the scalar translator. $A_{J'M',JM}$ and $B_{J'M',JM}$ are given in [15,24]. The translator $\overline{\beta}_{J'M',JM}(\mathbf{r}'')$ is the regular part of the translator $\overline{\alpha}_{J'M',JM}(\mathbf{r}'')$, i.e.,

$$\overline{\beta}_{J'M',JM}(\mathbf{r}'')=\mathfrak{R}g\overline{\alpha}_{J'M',JM}(\mathbf{r}'').$$

Moreover, we have [15]

$$\overline{\alpha}_{J'M',JM}(\mathbf{r}_{il}) = \sum_{J_1,M_1}\sum_{J_2,M_2} \overline{\beta}_{J'M',J_1M_1}(\mathbf{r}_{ij})\cdot\overline{\alpha}_{J_1M_1,J_2M_2}(\mathbf{r}_{jk})\cdot\overline{\beta}_{J_2M_2,JM}(\mathbf{r}_{kl}). \tag{2.13}$$

There is another set of vector multipole fields defined by [24]

$$\mathbf{F}_{JM}(\mathbf{r})=z_J(kr)\mathbf{Y}^{[J]}_{J,M}(\theta,\phi), \tag{2.14a}$$

$$\mathbf{H}_{JM}(\mathbf{r})=z_{J+1}(kr)\mathbf{Y}^{[J]}_{J+1,M}(\theta,\phi), \tag{2.14b}$$

$$\mathbf{T}_{JM}(\mathbf{r})=z_{J-1}(kr)\mathbf{Y}^{[J]}_{J-1,M}(\theta,\phi). \tag{2.14c}$$

For this set of vector multipole fields, we let

$$\overline{\widetilde{\boldsymbol{\Psi}}}^{t}_{J,M}(\mathbf{r})=\left[\mathbf{F}_{J,M}(\mathbf{r}),\mathbf{H}_{J,M}(\mathbf{r}),\mathbf{T}_{J,M}(\mathbf{r})\right]\in\mathbb{C}_{3\times 3}, \tag{2.15}$$

and then the similar vector addition theorem based on the set of vector multipole fields can be obtained

$$\overline{\widetilde{\boldsymbol{\Psi}}}^{t}_{J,M}(\mathbf{r})_{3\times 3}=\sum_{J',M'}\Re g\overline{\widetilde{\boldsymbol{\Psi}}}^{t}_{J',M'}(\mathbf{r}')_{3\times 3}\cdot\overline{\widetilde{\alpha}}_{J'M',JM}(\mathbf{r}'')_{3\times 3},\qquad |\mathbf{r}'|<|\mathbf{r}''|, \tag{2.16a}$$

$$\overline{\widetilde{\boldsymbol{\Psi}}}^{t}_{J,M}(\mathbf{r})_{3\times 3}=\sum_{J',M'}\overline{\widetilde{\boldsymbol{\Psi}}}^{t}_{J',M'}(\mathbf{r}')_{3\times 3}\cdot\overline{\widetilde{\beta}}_{J'M',JM}(\mathbf{r}'')_{3\times 3},\qquad |\mathbf{r}'|>|\mathbf{r}''|, \tag{2.16b}$$

$$\Re g\overline{\widetilde{\boldsymbol{\Psi}}}^{t}_{J,M}(\mathbf{r})_{3\times 3}=\sum_{J',M'}\Re g\overline{\widetilde{\boldsymbol{\Psi}}}^{t}_{J',M'}(\mathbf{r}')_{3\times 3}\cdot\overline{\widetilde{\beta}}_{J'M',JM}(\mathbf{r}'')_{3\times 3},\quad \forall\ |\mathbf{r}'|,\ |\mathbf{r}''|. \tag{2.16c}$$

The translator $\overline{\widetilde{\alpha}}_{J'M',JM}(\mathbf{r}'')$ can be written as

$$\overline{\widetilde{\alpha}}_{J'M',JM}(\mathbf{r}'')=\overline{\mathbf{S}}(J')\cdot\overline{\alpha}_{J'M',JM}(\mathbf{r}'')\cdot\overline{\mathbf{S}}^{-1}(J), \tag{2.17}$$

where

$$\overline{\mathbf{S}}(J)=\begin{bmatrix} 1 & 0 & 0 \\ 0 & -\dfrac{i\sqrt{J}}{\sqrt{2J+1}} & \dfrac{\sqrt{J+1}}{\sqrt{2J+1}} \\ 0 & \dfrac{i\sqrt{J+1}}{\sqrt{2J+1}} & \dfrac{\sqrt{J}}{\sqrt{2J+1}} \end{bmatrix}, \tag{2.18a}$$

and

$$\overline{\boldsymbol{\Psi}}^{t}_{J,M}(\mathbf{r})=\overline{\widetilde{\boldsymbol{\Psi}}}^{t}_{J,M}(\mathbf{r})\cdot\overline{\mathbf{S}}(J). \tag{2.18b}$$

The equation similar to (2.13) is as follows

$$\overline{\widetilde{\alpha}}_{J'M',JM}(\mathbf{r}_{il})=\sum_{J_1,M_1}\sum_{J_2,M_2}\overline{\widetilde{\beta}}_{J'M',J_1M_1}(\mathbf{r}_{ij})\cdot\overline{\widetilde{\alpha}}_{J_1M_1,J_2M_2}(\mathbf{r}_{jk})\cdot\overline{\widetilde{\beta}}_{J_2M_2,JM}(\mathbf{r}_{kl}). \tag{2.19}$$

In the above Eq. (2.19), the translator $\overline{\widetilde{\beta}}_{J'M',JM}(\mathbf{r})$ is the regular part of the translator $\overline{\widetilde{\alpha}}_{J'M',JM}(\mathbf{r})$. The translator matrix $\overline{\widetilde{\alpha}}_{J_1M_1,J_2M_2}(\mathbf{r}_{jk})$ translates an outgoing wave from one coordinate system to an incoming wave at another coordinate system and we denote it as an O2I translator. The translator matrix $\overline{\overline{\beta}}_{J_2M_2,JM}(\mathbf{r}_{kl})$ translates an outgoing wave from one coordinate system to an outgoing wave at another coordinate system, which is denoted as an $O2O$ translator, while the translator matrix $\overline{\overline{\beta}}_{J'M',J_1M_1}(\mathbf{r}_{ij})$ translates an incoming wave from one coordinate system to an incoming wave at another coordinate system, which is denoted as an I2I translator.

To further analyze these translators, the more explicit form of the O2I translator $\overline{\overline{\alpha}}_{J_1M_1,J_2M_2}$ is given as follows

$$\begin{bmatrix} a_{11}A_{J_1M_1,J_2M_2} & a_{12}B_{J_1M_1,J_2M_2} & a_{13}B_{J_1M_1,J_2M_2} \\ a_{21}B_{J_1M_1,J_2M_2} & a_{22}^{(1)}A_{J_1M_1,J_2M_2}+a_{22}^{(2)}\alpha_{J_1M_1,J_2M_2} & a_{23}^{(1)}A_{J_1M_1,J_2M_2}+a_{23}^{(2)}\alpha_{J_1M_1,J_2M_2} \\ a_{31}B_{J_1M_1,J_2M_2} & a_{32}^{(1)}A_{J_1M_1,J_2M_2}+a_{32}^{(2)}\alpha_{J_1M_1,J_2M_2} & a_{33}^{(1)}A_{J_1M_1,J_2M_2}+a_{33}^{(2)}\alpha_{J_1M_1,J_2M_2} \end{bmatrix}. \tag{2.20}$$

Here

$$\begin{aligned}
a_{11}&=\frac{\sqrt{J_1(J_1+1)}}{\sqrt{J_2(J_2+1)}}, & a_{12}&=\frac{i\sqrt{J_1(J_1+1)}}{\sqrt{(2J_2+1)(J_2+1)}},\\
a_{13}&=-\frac{i\sqrt{J_1(J_1+1)}}{\sqrt{J_2(2J_2+1)}}, & a_{21}&=-\frac{iJ_1\sqrt{J_1+1}}{\sqrt{(2J_1+1)J_2(J_2+1)}},\\
a_{22}^{(1)}&=\frac{J_1\sqrt{J_1+1}}{\sqrt{(2J_1+1)(2J_2+1)(J_2+1)}}, & a_{22}^{(2)}&=\frac{\sqrt{(J_1+1)(J_2+1)}}{\sqrt{(2J_1+1)(2J_2+1)}},\\
a_{23}^{(1)}&=-\frac{J_1\sqrt{(J_1+1)}}{\sqrt{(2J_1+1)J_2(2J_2+1)}}, & a_{23}^{(2)}&=\frac{\sqrt{J_2(J_1+1)}}{\sqrt{(2J_1+1)(2J_2+1)}},\\
a_{31}&=\frac{i(J_1+1)\sqrt{J_1}}{\sqrt{(2J_1+1)J_2(J_2+1)}}, & a_{32}^{(1)}&=-\frac{(J_1+1)\sqrt{J_1}}{\sqrt{(2J_1+1)(2J_2+1)(J_2+1)}},\\
a_{32}^{(2)}&=\frac{\sqrt{J_1(J_2+1)}}{\sqrt{(2J_1+1)(2J_2+1)}}, & a_{33}^{(1)}&=\frac{(J_1+1)\sqrt{J_1}}{\sqrt{(2J_1+1)(2J_2+1)J_2}},\\
a_{33}^{(2)}&=\frac{\sqrt{J_1J_2}}{\sqrt{(2J_1+1)(2J_2+1)}}.
\end{aligned}$$

Moreover, for low frequencies ($k\to 0$) or small structures ($kr\to 0$) [19,21],

$$A_{J_1M_1,J_2M_2}\approx\mathcal{O}\big(t^{-(J_1+J_2+1)}\big), \tag{2.21a}$$

$$B_{J_1M_1,J_2M_2}\approx\mathcal{O}\big(t^{-(J_1+J_2)}\big), \tag{2.21b}$$

$$\alpha_{J_1M_1,J_2M_2}\approx\mathcal{O}\big(t^{-(J_1+J_2+1)}\big), \tag{2.21c}$$

where t is a parameter that satisfies

$$t/kr\approx\mathcal{O}(1).$$

By using the expression of the O2I translator matrix (2.20), we can calculate the order of its each element. It should be noted that the order of the (3,3) element of the translator

$\overline{\overline{\alpha}}_{J_1M_1,J_2M_2}$ is $\mathcal{O}(t^{-(J_1+J_2-1)})$, which is given as follows by canceling two highest order terms in $a_{33}^{(1)}A_{J_1M_1,J_2M_2}$ and $a_{33}^{(2)}\alpha_{J_1M_1,J_2M_2}$:

$$\begin{aligned}
&a_{33}^{(1)}A_{J_1M_1,J_2M_2}+a_{33}^{(2)}\alpha_{J_1M_1,J_2M_2}\\
=&\frac{(J_1+1)\sqrt{J_1}}{\sqrt{(2J_1+1)(2J_2+1)J_2}}A_{J_1M_1,J_2M_2}+\frac{\sqrt{J_1J_2}}{\sqrt{(2J_1+1)(2J_2+1)}}\alpha_{J_1M_1,J_2M_2}\\
=&\frac{\sqrt{J_1}}{\sqrt{(2J_1+1)(2J_2+1)J_2}}\left[(J_1+1)A_{J_1M_1,J_2M_2}+J_2\alpha_{J_1M_1,J_2M_2}\right]\\
=&\frac{\sqrt{J_1}}{\sqrt{(2J_1+1)(2J_2+1)J_2}}4\pi\sum_{J''}i^{J_1-J_2+J''}z_{J''}(kr'')Y_{J'',M_2-M_1}(\theta'',\phi'')\\
&\times\left[\frac{J_1(J_1+1)+J_2(J_2+1)-J''(J''+1)}{2J_1}+J_2\right]A(M_2,J_2,-M_1,J_1,J''),
\end{aligned}\tag{2.22}$$

where $|J_1-J_2|\le J''\le J_1+J_2-2$, J'' increments by a step of 2, and

$$\begin{aligned}
A(M_2,J_2,-M_1,J_1,J'')=&(-1)^{M_2}\left[(2J_2+1)(2J_1+1)(2J''+1)/4\pi\right]^{\frac{1}{2}}\\
&\cdot\begin{pmatrix} J_2 & J_1 & J''\\ 0 & 0 & 0\end{pmatrix}\cdot\begin{pmatrix} J_2 & J_1 & J''\\ -M_2 & M_1 & M_2-M_1\end{pmatrix}.
\end{aligned}\tag{2.23}$$

Since kr is very small for low-frequency problems and we have approximation formulations in (2.21), two highest order terms in $a_{33}^{(1)}A_{J_1M_1,J_2M_2}$ and $a_{33}^{(2)}\alpha_{J_1M_1,J_2M_2}$ are very large. Theoretically these two highest order terms should cancel each other exactly in the subtraction operation to generate the (3,3) element of the matrix in (2.20). However, in practice, the numerical error due to subtraction of highest order terms will swamp the real value of the (3,3) element. Hence, instead of using those two terms on the left hand side of (2.22), we should adopt the expression on the right hand side of (2.22) to calculate the (3,3) element directly and store it alone. Then we have

$$\overline{\overline{\alpha}}_{J_1M_1,J_2M_2}\approx\begin{bmatrix}\mathcal{O}(t^{-(J_1+J_2+1)}) & \mathcal{O}(t^{-(J_1+J_2)}) & \mathcal{O}(t^{-(J_1+J_2)})\\ \mathcal{O}(t^{-(J_1+J_2)}) & \mathcal{O}(t^{-(J_1+J_2+1)}) & \mathcal{O}(t^{-(J_1+J_2+1)})\\ \mathcal{O}(t^{-(J_1+J_2)}) & \mathcal{O}(t^{-(J_1+J_2+1)}) & \mathcal{O}(t^{-(J_1+J_2-1)})\end{bmatrix}.\tag{2.24}$$

In the like manner, the order of each element of the *O2O* translator $\overline{\overline{\beta}}_{J_2M_2,JM}$ and the I2I translator $\overline{\overline{\beta}}_{J'M',J_1M_1}$ can be obtained.

As to the *O2O* translator $\overline{\overline{\beta}}_{J_2M_2,JM}$,

$$\overline{\overline{\beta}}_{J_2M_2,JM}\approx\begin{bmatrix}\mathcal{O}(t^{|J_2-J|}) & \mathcal{O}(t^{|J_2-J|+1}) & \mathcal{O}(t^{|J_2-J|+1})\\ \mathcal{O}(t^{|J_2-J|+1}) & \mathcal{O}(t^{|J_2-J|}) & \mathcal{O}(t^{|J_2-J|})\\ \mathcal{O}(t^{|J_2-J|+1}) & \mathcal{O}(t^{|J_2-J|}) & \mathcal{O}(t^{|J_2-J|})\end{bmatrix},\tag{2.25}$$

and its (2,3) element contains the cancelation of high order terms for $J_2 \geq J$, which is discussed as follows.

When $J_2 \geq J$, the (2,3) element is

$$\begin{aligned}
&a_{23}^{(1)} \Re g A_{J_2M_2,JM} + a_{23}^{(2)} \beta_{J_2M_2,JM} \\
&= -\frac{J_2\sqrt{(J_2+1)}}{\sqrt{(2J_2+1)J(2J+1)}} \Re g A_{J_2M_2,JM} + \frac{\sqrt{(J_2+1)J}}{\sqrt{(2J_2+1)(2J+1)}} \beta_{J_2M_2,JM} \\
&= \frac{\sqrt{J_2+1}}{\sqrt{(2J_2+1)(2J+1)J}} \left(-J_2 \Re g A_{J_2M_2,JM} + J\beta_{J_2M_2,JM}\right) \\
&= \frac{\sqrt{J_2+1}}{\sqrt{(2J_2+1)(2J+1)J}} 4\pi \sum_{J''} i^{J_2-J+J''} \Re g z_{J''}(kr'') Y_{J'',M-M_2}(\theta'',\phi'') \\
&\quad \times \Big[-\frac{J_2(J_2+1)+J(J+1)-J''(J''+1)}{2(J_2+1)} + J\Big] A(M,J,-M_2,J_2,J''),
\end{aligned} \tag{2.26}$$

where

$$\beta_{J_2M_2,JM} = \Re g \alpha_{J_2M_2,JM}, \qquad |J_2 - J| + 2 \leq J'' \leq J_2 + J,$$

and J'' increments by a step of 2. Therefore, the order of (2,3) element is $\mathcal{O}(t^{(J_2-J+2)})$ for $J_2 \geq J$.

The orders of elements of the I2I translator $\overline{\overline{\beta}}_{J'M',J_1M_1}$ are

$$\begin{bmatrix}
\mathcal{O}(t^{|J'-J_1|}) & \mathcal{O}(t^{|J'-J_1|+1}) & \mathcal{O}(t^{|J'-J_1|+1}) \\
\mathcal{O}(t^{|J'-J_1|+1}) & \mathcal{O}(t^{|J'-J_1|}) & \mathcal{O}(t^{|J'-J_1|}) \\
\mathcal{O}(t^{|J'-J_1|+1}) & \mathcal{O}(t^{|J'-J_1|}) & \mathcal{O}(t^{|J'-J_1|})
\end{bmatrix}, \tag{2.27}$$

and its (3,2) element is the special one for $J_1 \geq J'$.

When $J_1 \geq J'$, the (3,2) element of the I2I translator $\overline{\overline{\beta}}_{J'M',J_1M_1}$ equals

$$\begin{aligned}
&a_{32}^{(1)} \Re g A_{J'M',J_1M_1} + a_{32}^{(2)} \beta_{J'M',J_1M_1} \\
&= -\frac{(J'+1)\sqrt{J'}}{\sqrt{(2J'+1)(2J_1+1)(J_1+1)}} \Re g A_{J'M',J_1M_1} + \frac{\sqrt{(J_1+1)J'}}{\sqrt{(2J_1+1)(2J'+1)}} \beta_{J'M',J_1M_1} \\
&= \frac{\sqrt{J'}}{\sqrt{(2J_1+1)(2J'+1)(J_1+1)}} \left[-(J'+1) \Re g A_{J'M',J_1M_1} + (J_1+1)\beta_{J'M',J_1M_1}\right] \\
&= \frac{\sqrt{J'}}{\sqrt{(2J_1+1)(2J'+1)(J_1+1)}} 4\pi \sum_{J''} i^{J'-J_1+J''} \Re g z_{J''}(kr'') Y_{J'',M_1-M'}(\theta'',\phi'') \\
&\quad \times \Big[-\frac{J'(J'+1)+J_1(J_1+1)-J''(J''+1)}{2J'} + (J_1+1)\Big] A(M_1,J_1,-M',J',J''),
\end{aligned} \tag{2.28}$$

where $|J_1 - J'| + 2 \leq J'' \leq J' + J_1$ and J'' increments by a step of 2. Therefore, the order of (3,2) element of the I2I translator $\overline{\overline{\beta}}_{J'M',J_1M_1}$ is $\mathcal{O}(t^{(J_1-J'+2)})$ for $J_1 \geq J'$.

Just like the (3,3) element of the $O2I$ translator $\overline{\overline{\widetilde{\alpha}}}_{J_1M_1,J_2M_2}$, the (2,3) element of the $O2O$ translator $\overline{\overline{\widetilde{\beta}}}_{J_2M_2,JM}$ and the (3,2) element of the I2I translator $\overline{\overline{\widetilde{\beta}}}_{J'M',J_1M_1}$ should be calculated by using expressions given on the right hand sides of (2.26) and (2.28) respectively and stored alone to guarantee the accuracy of these elements. Then, based on translators mentioned above, we can give the nonuniformly normalized form of the core equation for the 3D undiagonalized dynamic multilevel fast multipole algorithm,

$$\begin{aligned}&\overline{\overline{\widetilde{\alpha}}}^N_{JM,J'M'}(\mathbf{r}_{il},t_{il})\\&=\Big(\frac{t_{il}}{t_{ij}}\Big)^J\sum_{J_1,M_1}\Big(\frac{t_{ij}}{t_{jh}}\Big)^{J_1}\overline{\overline{\widetilde{\beta}}}^N_{JM,J_1M_1}(\mathbf{r}_{ij},t_{ij})\sum_{J_3,M_3}\Big(\frac{t_{jh}}{t_{hp}}\Big)^{J_3}\cdot\overline{\overline{\widetilde{\beta}}}^N_{J_1M_1,J_3M_3}(\mathbf{r}_{jh},t_{jh})\\&\quad\times\sum_{J_4,M_4}\frac{t_{il}}{t_{hp}}\overline{\overline{\widetilde{\alpha}}}^N_{J_3M_3,J_4M_4}(\mathbf{r}_{hp},t_{hp})\cdot\sum_{J_2,M_2}\Big(\frac{t_{pk}}{t_{hp}}\Big)^{J_4}\overline{\overline{\widetilde{\beta}}}^N_{J_4M_4,J_2M_2}(\mathbf{r}_{pk},t_{pk})\Big(\frac{t_{kl}}{t_{pk}}\Big)^{J_2}\\&\quad\times\overline{\overline{\widetilde{\beta}}}^N_{J_2M_2,J'M'}(\mathbf{r}_{kl},t_{kl})\Big(\frac{t_{il}}{t_{kl}}\Big)^{J'},\end{aligned}\tag{2.29}$$

where normalized translators are defined as

$$\overline{\overline{\widetilde{\alpha}}}^N_{J_1M_1,J_2M_2}(\mathbf{r},t)=t^{J_1+J_2+1}\overline{\overline{\widetilde{\alpha}}}_{J_1M_1,J_2M_2}(\mathbf{r}),\tag{2.30a}$$

$$\overline{\overline{\widetilde{\beta}}}^N_{J_1M_1,J_2M_2}(\mathbf{r},t)=t^{J^<-J^>}\overline{\overline{\widetilde{\beta}}}_{J_1M_1,J_2M_2}(\mathbf{r}).\tag{2.30b}$$

Here the superscript $<$ and the superscript $>$ denote the terms at the lower level and higher level, respectively. The parameter t depends on the level, which will be distinguished by different subscripts. Eq. (2.29) has an analogue given by using the scalar addition theorem [19,21,23].

After presenting the vector addition theorem based on different vector multipole fields, we will discuss the factorization of the dyadic Green's function with vector addition theorem. The dyadic Green's function can be expressed as

$$\overline{\mathbf{G}}(\mathbf{r}_1,\mathbf{r}_2)=\overline{\mathbf{I}}g(\mathbf{r}_1,\mathbf{r}_2)-\frac{\nabla_1\nabla_2}{k^2}g(\mathbf{r}_1,\mathbf{r}_2),\tag{2.31}$$

where the scalar Green's function $g(\mathbf{r}_1,\mathbf{r}_2)$ has the expansion in terms of spherical wave functions

$$g(\mathbf{r}_1,\mathbf{r}_2)=\frac{e^{ik|\mathbf{r}_1-\mathbf{r}_2|}}{4\pi|\mathbf{r}_1-\mathbf{r}_2|}=ik\sum_{J,M}\psi_{J,M}(k,\mathbf{r}_1)\Re g\psi^*_{J,M}(k,\mathbf{r}_2),\tag{2.32}$$

and the spherical wave function

$$\psi_{J,M}(k,\mathbf{r})=z_J(kr)Y_{J,M}(\hat{\mathbf{r}}).$$

By using the scalar addition theorem [21], the scalar Green's function can be factorized as

$$g(\mathbf{r}_1,\mathbf{r}_2)=ik\sum_{J,M}\sum_{J',M'}\Re g\psi_{J',M'}(k,\mathbf{r}_{13})\alpha_{J'M',JM}(\mathbf{r}_{34})\Re g\psi^*_{J,M}(k,\mathbf{r}_{24}),\tag{2.33}$$

where $\mathbf{r}_{ij}=\mathbf{r}_i-\mathbf{r}_j$. Using (2.32) in the second term on the right hand side of (2.31), we can also get

$$\frac{\nabla_1\nabla_2}{k^2}g(\mathbf{r}_1,\mathbf{r}_2)=ik\sum_{J,M}\mathbf{L}_{J,M}(\mathbf{r}_1)\Re g\mathbf{L}^{*t}_{J,M}(\mathbf{r}_2). \tag{2.34}$$

Moreover, the tensorial Green's function $\overline{\mathbf{I}}g(\mathbf{r}_1,\mathbf{r}_2)$ can be expressed in terms of the above two sets of vector multipole fields, respectively.

$$\begin{aligned}\overline{\mathbf{I}}g(\mathbf{r}_1,\mathbf{r}_2)=ik\sum_{J,M}\Big(&\mathbf{m}_{J,M}(\mathbf{r}_1)\Re g\mathbf{m}^{*t}_{J,M}(\mathbf{r}_2)+\mathbf{n}_{J,M}(\mathbf{r}_1)\Re g\mathbf{n}^{*t}_{J,M}(\mathbf{r}_2)\\ &+\mathbf{L}_{J,M}(\mathbf{r}_1)\Re g\mathbf{L}^{*t}_{J,M}(\mathbf{r}_2)\Big),\end{aligned} \tag{2.35a}$$

$$\begin{aligned}\overline{\mathbf{I}}g(\mathbf{r}_1,\mathbf{r}_2)=ik\sum_{J,M}\Big(&\mathbf{F}_{J,M}(\mathbf{r}_1)\Re g\mathbf{F}^{*t}_{JM}(\mathbf{r}_2)+\mathbf{H}_{J,M}(\mathbf{r}_1)\Re g\mathbf{H}^{*t}_{JM}(\mathbf{r}_2)\\ &+\mathbf{T}_{J,M}(\mathbf{r}_1)\Re g\mathbf{T}^{*t}_{JM}(\mathbf{r}_2)\Big).\end{aligned} \tag{2.35b}$$

By applying (2.34) and (2.35a) in (2.31), the dyadic Green's function can be written as

$$\begin{aligned}\overline{\mathbf{G}}(\mathbf{r}_1,\mathbf{r}_2)=ik\sum_{J,M}\Big(&\mathbf{m}_{J,M}(\mathbf{r}_{14})\Re g\mathbf{m}^{*t}_{J,M}(\mathbf{r}_{24})+\mathbf{n}_{J,M}(\mathbf{r}_{14})\Re g\mathbf{n}^{*t}_{J,M}(\mathbf{r}_{24})\\ &+\mathbf{L}_{J,M}(\mathbf{r}_{14})\Re g\mathbf{L}^{*t}_{J,M}(\mathbf{r}_{24})\Big)-ik\sum_{J,M}\mathbf{L}_{J,M}(\mathbf{r}_{14})\Re g\mathbf{L}^{*t}_{J,M}(\mathbf{r}_{24}),\end{aligned} \tag{2.36}$$

with (2.10a), Eq. (2.36) becomes

$$\begin{aligned}\overline{\mathbf{G}}(\mathbf{r}_1,\mathbf{r}_2)=ik\sum_{J,M}\sum_{J',M'}&\Re g\overline{\mathbf{\Psi}}^{t}_{J',M'}(\mathbf{r}_{13})\cdot\overline{\alpha}_{J'M',JM}(\mathbf{r}_{34})\cdot\Re g\overline{\mathbf{\Psi}}^{*}_{JM}(\mathbf{r}_{24})\\ &-ik\sum_{J,M}\sum_{J',M'}\Re g\mathbf{L}_{J',M'}(\mathbf{r}_{13})\cdot\alpha_{J'M',JM}(\mathbf{r}_{34})\cdot\Re g\mathbf{L}^{*t}_{J,M}(\mathbf{r}_{24}).\end{aligned} \tag{2.37}$$

On the other hand, substituting (2.34) and (2.35b) into (2.31), we can get another expansion of the dyadic Green's function

$$\begin{aligned}\overline{\mathbf{G}}(\mathbf{r}_1,\mathbf{r}_2)=ik\sum_{J,M}\Big(&\mathbf{F}_{J,M}(\mathbf{r}_{14})\Re g\mathbf{F}^{*t}_{J,M}(\mathbf{r}_{24})+\mathbf{H}_{J,M}(\mathbf{r}_{14})\Re g\mathbf{H}^{*t}_{J,M}(\mathbf{r}_{24})\\ &\cdot\mathbf{T}_{J,M}(\mathbf{r}_{14})\Re g\mathbf{T}^{*t}_{J,M}(\mathbf{r}_{24})\Big)-ik\sum_{J,M}\mathbf{L}_{J,M}(\mathbf{r}_{14})\Re g\mathbf{L}^{*t}_{J,M}(\mathbf{r}_{24}).\end{aligned} \tag{2.38}$$

Then, using (2.16) in (2.38), the dyadic Green's function is re-expressed as

$$\begin{aligned}\overline{\mathbf{G}}(\mathbf{r}_1,\mathbf{r}_2)=ik\sum_{J,M}\sum_{J',M'}&\Re g\overline{\widetilde{\mathbf{\Psi}}}^{t}_{J',M'}(\mathbf{r}_{13})\cdot\overline{\widetilde{\alpha}}_{J'M',JM}(\mathbf{r}_{34})\cdot\Re g\overline{\widetilde{\mathbf{\Psi}}}^{*}_{JM}(\mathbf{r}_{24})\\ &-ik\sum_{J,M}\sum_{J',M'}\Re g\mathbf{L}_{J',M'}(\mathbf{r}_{13})\cdot\alpha_{J'M',JM}(\mathbf{r}_{34})\cdot\Re g\mathbf{L}^{*t}_{J,M}(\mathbf{r}_{24}).\end{aligned} \tag{2.39}$$

From (2.36), (2.38) and the following equation

$$\mathbf{L}_{JM}(\mathbf{r}_1)=\frac{\sqrt{J+1}}{\sqrt{2J+1}}\mathbf{H}_{JM}(\mathbf{r}_1)+\frac{\sqrt{J}}{\sqrt{2J+1}}\mathbf{T}_{JM}(\mathbf{r}_1), \tag{2.40}$$

we know that the dyadic Green's function can be expressed not only with the vector multipole fields $\mathbf{m}_{J,M}(\mathbf{r})$, $\mathbf{n}_{J,M}(\mathbf{r})$ and $\mathbf{L}_{J,M}(\mathbf{r})$, but also with the vector multipole fields $\mathbf{F}_{J,M}(\mathbf{r})$, $\mathbf{H}_{J,M}(\mathbf{r})$ and $\mathbf{T}_{J,M}(\mathbf{r})$. In addition, Eq. (2.40) is important for saving memory in the radiation and receiving patterns of the LF-VFMA, which will be introduced in Section 4.

If we use (2.36) to factorize the dyadic Green's function, for low frequency problems, there is a high order term

$$-\frac{\sqrt{J(J+1)}}{2J+1}z_{J+1}(kr_1)\Re g z_{J-1}(kr_2)\mathbf{Y}_{J+1,M}^{[J]}(\hat{\mathbf{r}}_1)\mathbf{Y}_{J-1,M}^{[J]*t}(\hat{\mathbf{r}}_2),$$

contained in the term $\mathbf{n}_{J,M}(k,\mathbf{r}_1)\Re g\mathbf{n}_{J,M}^{*t}(k,\mathbf{r}_2)$ and a high order term of opposite sign

$$\frac{\sqrt{J(J+1)}}{2J+1}z_{J+1}(kr_1)\Re g z_{J-1}(kr_2)\mathbf{Y}_{J+1,M}^{[J]}(\hat{\mathbf{r}}_1)\mathbf{Y}_{J-1,M}^{[J]*t}(\hat{\mathbf{r}}_2),$$

contained in the term $\mathbf{L}_{J,M}(k,\mathbf{r}_1)\Re g\mathbf{L}_{J,M}^{*t}(k,\mathbf{r}_2)$. The order of these two terms is $\mathcal{O}(k^{-3})$. Since the left hand side of Eq. (2.35a) is $\mathcal{O}(1)$, these two high order terms should cancel each other exactly in theory. However, the numerical error caused by the cancelation of high order terms will swamp the true value of $\overline{\mathbf{I}}g(\mathbf{r}_1,\mathbf{r}_2)$ and eventually lead to wrong numerical solution. Therefore, expressions of dyadic Green's function (2.36) and (2.37) are not suitable for solving low frequency problems. On the other hand, since terms

$$\mathbf{F}_{J,M}(k,\mathbf{r}_1)\Re g\mathbf{F}_{J,M}^{*t}(k,\mathbf{r}_2),\ \ \mathbf{H}_{J,M}(k,\mathbf{r}_1)\Re g\mathbf{H}_{J,M}^{*t}(k,\mathbf{r}_2),\text{ and }\ \mathbf{T}_{J,M}(k,\mathbf{r}_1)\Re g\mathbf{T}_{J,M}^{*t}(k,\mathbf{r}_2),$$

have the same order $\mathcal{O}(k^{-1})$, there is no cancelation error involved in numerically calculating the formulation on the right hand side of (2.35b). Therefore, for low frequency problems, we can adopt Eqs. (2.38) and (2.39) to factorize the dyadic Green's function.

3 General equations

To avoid unnecessary complexity for introducing our method, we assume that the target is a perfect electric conductor. A brief review of the electric-field integral equation based on the loop-tree basis will be given in this section. Definitions of symbols in this section are similar to those in [20].

The electric field integral equation (EFIE) for an arbitrary 3D PEC object can be written as

$$-\hat{\mathbf{t}}(\mathbf{r})\cdot\mathbf{E}^{inc}(\mathbf{r})=i\omega\mu\hat{\mathbf{t}}(\mathbf{r})\cdot\int_S\overline{\mathbf{G}}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')d\mathbf{r}',\quad \mathbf{r}\in S, \tag{3.1}$$

where $\mathbf{E}^{inc}$ is the incident electric field, $\mathbf{J}$ is the surface current, and $\hat{\mathbf{t}}(\mathbf{r})$ is an arbitrary tangential unit vector on the surface S. Let $\mathbf{J}_{L_n}(\mathbf{r})$ be a loop basis function such that $\nabla\cdot\mathbf{J}_{L_n}(\mathbf{r})=0$, and $\mathbf{J}_{C_n}(\mathbf{r})$ be a tree basis function such that $\nabla\cdot\mathbf{J}_{C_n}(\mathbf{r})\neq 0$. Loop and tree basis can be used to model the current in the system and the current $\mathbf{J}$ is expanded as

$$\mathbf{J}(\mathbf{r})=\sum_{n=1}^{N_L} I_{L_n}\mathbf{J}_{L_n}(\mathbf{r})+\sum_{n=1}^{N_C} I_{C_n}\mathbf{J}_{C_n}(\mathbf{r}). \tag{3.2}$$

In the compact matrix form, the above equation is

$$\mathbf{J}(\mathbf{r})=\overline{\mathbf{J}}_L^t(\mathbf{r})\cdot\mathbf{I}_L+\overline{\mathbf{J}}_C^t(\mathbf{r})\cdot\mathbf{I}_C, \tag{3.3}$$

where $\overline{\mathbf{J}}_L(\mathbf{r})$ and $\overline{\mathbf{J}}_C(\mathbf{r})$ are matrices, whose n-th columns are $\mathbf{J}_{L_n}(\mathbf{r})$ and $\mathbf{J}_{C_n}(\mathbf{r})$ respectively, while $\mathbf{I}_L$ and $\mathbf{I}_C$ are column vectors, whose n-th elements are I_{L_n} and I_{C_n} respectively.

With the loop-tree basis, we have the matrix equation of (3.1)

$$\begin{bmatrix}\overline{\mathbf{Z}}_{LL} & \overline{\mathbf{Z}}_{LC}\\ \overline{\mathbf{Z}}_{CL} & \overline{\mathbf{Z}}_{CC}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{I}_L\\ \mathbf{I}_C\end{bmatrix}=\begin{bmatrix}\mathbf{V}_L\\ \mathbf{V}_C\end{bmatrix}, \tag{3.4}$$

where

$$\begin{aligned}
\overline{\mathbf{Z}}_{LL}&=i\omega\mu\langle\overline{\mathbf{J}}_L(\mathbf{r}),\overline{\mathbf{G}}(\mathbf{r},\mathbf{r}'),\overline{\mathbf{J}}_L^t(\mathbf{r}')\rangle=i\omega\mu\langle\overline{\mathbf{J}}_L(\mathbf{r}),\overline{\mathbf{I}}g(\mathbf{r},\mathbf{r}'),\overline{\mathbf{J}}_L^t(\mathbf{r}')\rangle=\overline{\mathbf{Z}}_{LL}^V,\\
\overline{\mathbf{Z}}_{LC}&=i\omega\mu\langle\overline{\mathbf{J}}_L(\mathbf{r}),\overline{\mathbf{G}}(\mathbf{r},\mathbf{r}'),\overline{\mathbf{J}}_C^t(\mathbf{r}')\rangle=i\omega\mu\langle\overline{\mathbf{J}}_L(\mathbf{r}),\overline{\mathbf{I}}g(\mathbf{r},\mathbf{r}'),\overline{\mathbf{J}}_C^t(\mathbf{r}')\rangle=\overline{\mathbf{Z}}_{LC}^V,\\
\overline{\mathbf{Z}}_{CL}&=i\omega\mu\langle\overline{\mathbf{J}}_C(\mathbf{r}),\overline{\mathbf{G}}(\mathbf{r},\mathbf{r}'),\overline{\mathbf{J}}_L^t(\mathbf{r}')\rangle=i\omega\mu\langle\overline{\mathbf{J}}_C(\mathbf{r}),\overline{\mathbf{I}}g(\mathbf{r},\mathbf{r}'),\overline{\mathbf{J}}_L^t(\mathbf{r}')\rangle=\overline{\mathbf{Z}}_{CL}^V,\\
\overline{\mathbf{Z}}_{CC}&=i\omega\mu\langle\overline{\mathbf{J}}_C(\mathbf{r}),\overline{\mathbf{G}}(\mathbf{r},\mathbf{r}'),\overline{\mathbf{J}}_C^t(\mathbf{r}')\rangle=i\omega\mu\langle\overline{\mathbf{J}}_C(\mathbf{r}),\overline{\mathbf{I}}g(\mathbf{r},\mathbf{r}'),\overline{\mathbf{J}}_C^t(\mathbf{r}')\rangle\\
&\quad-\frac{ik^2}{\omega\epsilon}\langle\overline{\mathbf{J}}_C(\mathbf{r}),\frac{\nabla\nabla'}{k^2}g(\mathbf{r},\mathbf{r}'),\overline{\mathbf{J}}_C^t(\mathbf{r}')\rangle=\overline{\mathbf{Z}}_{CC}^V+\overline{\mathbf{Z}}_{CC}^S.
\end{aligned}$$

Moreover, vectors $\mathbf{V}_L$ and $\mathbf{V}_C$ on the right hand side of (3.4) are

$$\mathbf{V}_L=-\big\langle\overline{\mathbf{J}}_L(\mathbf{r}),\ \mathbf{E}(\mathbf{r})\big\rangle, \tag{3.5a}$$

$$\mathbf{V}_C=-\big\langle\overline{\mathbf{J}}_C(\mathbf{r}),\ \mathbf{E}(\mathbf{r})\big\rangle. \tag{3.5b}$$

Since we use the electric field integral equation, when the frequency is very low, the scalar potential part dominates over the vector potential part. This makes (3.4) unbalanced and ill-conditioned. To overcome the low-frequency breakdown, the frequency normalization is proposed. The frequency normalized impedance matrix based on the loop-tree basis is no longer nearly singular and the matrix equation can be solved by direct inversion methods without any problem [20]. The large number of unknowns is needed for complex structures and it is impractical to solve matrix equations of large scale problems with direct inversion methods. Therefore iterative solvers should be considered for such problems. But when the frequency normalized matrix equation is solved with iterative solvers, the iteration count is very large. This is because the divergence of the RWG basis is not suitable for representing charges. Then the basis rearrangement is proposed to improve the spectral property of the impedance matrix [20], with which the iteration count is reduced dramatically. However, the accumulated numerical error, caused by subtractions involved in the basis rearrangement procedure, leads to the divergence of the iteration when the number of unknowns reaches several hundred thousands [22]. To remedy this, the patch-pair basis is constructed from the single-patch basis directly to avoid the error accumulated in subtractions [22], which will be used in the next section.

With the basis rearrangement and frequency normalization [20], Eq. (3.4) becomes

$$\begin{aligned}&\begin{bmatrix}\frac{4}{i\omega\mu}\overline{\mathbf{I}} & \overline{\mathbf{0}}\\ \overline{\mathbf{0}} & \epsilon\overline{\mathbf{K}}^{t-1}\end{bmatrix}\cdot\overline{\mathbf{F}}^{t}\cdot\overline{\mathbf{Z}}^{V}_{\mathrm{RWG}}\cdot\overline{\mathbf{F}}\cdot\begin{bmatrix}\overline{\mathbf{I}} & \overline{\mathbf{0}}\\ \overline{\mathbf{0}} & i\omega\overline{\mathbf{K}}^{-1}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{I}_L\\ \mathbf{Q}\end{bmatrix}\\ &\quad+\begin{bmatrix}\overline{\mathbf{0}} & \overline{\mathbf{0}}\\ \overline{\mathbf{0}} & i\omega\epsilon\overline{\mathbf{K}}^{t-1}\cdot\overline{\mathbf{Z}}^{S}_{CC}\cdot\overline{\mathbf{K}}^{-1}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{I}_L\\ \mathbf{Q}\end{bmatrix}=\begin{bmatrix}\frac{4}{i\omega\mu}\mathbf{V}_L\\ \epsilon\mathbf{K}^{t-1}\cdot\mathbf{V}_C\end{bmatrix},\end{aligned}\tag{3.6}$$

where the element of the matrix $\overline{\mathbf{Z}}^{V}_{\mathrm{RWG}}$ is

$$\left[\overline{\mathbf{Z}}^{V}_{\mathrm{RWG}}\right]_{m,n}=i\omega\mu\langle\mathbf{J}_{R_m}(\mathbf{r}),\overline{\mathbf{I}}g(\mathbf{r},\mathbf{r}'),\mathbf{J}^{t}_{R_n}(\mathbf{r}')\rangle,\quad m,n=1,\cdots,N_R,\tag{3.7}$$

$\mathbf{J}_{R_m}$ is the m-th RWG basis function and N_R is the number of RWG basis functions. The matrix $\overline{\mathbf{F}}^{t}$ is a transformation matrix that changes from the RWG basis to the loop-tree basis. The matrix $\overline{\mathbf{K}}$ corresponds to the process of the basis rearrangement with the relationship

$$\overline{\mathbf{K}}\cdot\mathbf{I}_C=i\omega\mathbf{Q}.\tag{3.8}$$

4 LF-VFMA based on the vector addition theorem

Based on formulations introduced in the last section, we will present the low-frequency vector fast multipole algorithm in this section. The factorization of the dyadic Green's function (2.39) with vector addition theorem is adopted in this method. The storage requirement for radiation and receiving patterns and vector translators is discussed to show the memory efficiency of this method for large scale problems.

We will introduce the low-frequency vector fast multipole algorithm in two steps. First we consider the matrix-vector multiplication in the first term on the left hand side of (3.6), which corresponds to the contribution of the vector potential part. The fast multipole algorithm is used to accelerate the multiplication of the matrix $\overline{\mathbf{Z}}^{V}_{\mathrm{RWG}}$ and a vector. By using (2.35b) and the vector addition theorem of the set of vector multipole fields (2.14) in (3.7), any element in the non-near neighbor part of the matrix $\overline{\mathbf{Z}}^{V}_{\rm RWG}$ can be expressed as

$$\begin{aligned}&\big[\overline{\mathbf{Z}}^{V}_{\rm RWG}\big]_{m,n}=i\omega\mu\langle\mathbf{J}_{R_m}(\mathbf{r}_1),\,\bar{\mathbf{I}}g(\mathbf{r}_1,\mathbf{r}_2),\,\mathbf{J}^{t}_{R_n}(\mathbf{r}_2)\rangle\\ =&-\omega\mu k\sum_{J,M}\sum_{J',M'}\big\langle\mathbf{J}_{R_m}(\mathbf{r}_1),\,\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}^{t}_{J',M'}(\mathbf{r}_{13})\big\rangle\cdot\overline{\overline{\widetilde{\alpha}}}_{J'M',JM}(\mathbf{r}_{34})\cdot\big\langle\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}^{*}_{J,M}(\mathbf{r}_{24}),\,\mathbf{J}^{t}_{R_n}(\mathbf{r}_2)\big\rangle,\end{aligned}\tag{4.1}$$

where $\big\langle\mathbf{J}_{R_m}(\mathbf{r}_1),\,\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}^{t}_{J',M'}(\mathbf{r}_{13})\big\rangle$ and $\big\langle\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}^{*}_{J,M}(\mathbf{r}_{24}),\,\mathbf{J}^{t}_{R_n}(\mathbf{r}_2)\big\rangle$ are receiving and radiation patterns respectively. Note that

$$\begin{aligned}&\big\langle\mathbf{J}_{R_m}(\mathbf{r}_1),\,\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}^{t}_{J',M'}(\mathbf{r}_{13})\big\rangle\\ =&\big[\big\langle\mathbf{J}_{R_m}(\mathbf{r}_1),\mathfrak{R}g\mathbf{F}_{J',M'}(\mathbf{r}_{13})\big\rangle,\big\langle\mathbf{J}_{R_m}(\mathbf{r}_1),\mathfrak{R}g\mathbf{H}_{J',M'}(\mathbf{r}_{13})\big\rangle,\big\langle\mathbf{J}_{R_m}(\mathbf{r}_1),\mathfrak{R}g\mathbf{T}_{J',M'}(\mathbf{r}_{13})\big\rangle\big],\end{aligned}\tag{4.2a}$$

$$\begin{aligned}&\big\langle\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}^{*}_{J,M}(\mathbf{r}_{24}),\,\mathbf{J}^{t}_{R_n}(\mathbf{r}_2)\big\rangle\\ =&\big[\big\langle\mathfrak{R}g\mathbf{F}^{*t}_{J,M}(\mathbf{r}_{24}),\mathbf{J}^{t}_{R_n}(\mathbf{r}_2)\big\rangle,\big\langle\mathfrak{R}g\mathbf{H}^{*t}_{J,M}(\mathbf{r}_{24}),\mathbf{J}^{t}_{R_n}(\mathbf{r}_2)\big\rangle,\big\langle\mathfrak{R}g\mathbf{T}^{*t}_{J,M}(\mathbf{r}_{24}),\mathbf{J}^{t}_{R_n}(\mathbf{r}_2)\big\rangle\big]^{t}.\end{aligned}\tag{4.2b}$$

With the frequency normalization [20] and (2.29), we can get the normalized multilevel expression of (4.1) for the vector fast multipole algorithm,

$$\begin{aligned}\big[\overline{\mathbf{Z}}^{V}_{\rm RWG}\big]_{m,n}=&i\omega\mu\langle\mathbf{J}_{R_m}(\mathbf{r}_1),\,\bar{\mathbf{I}}g(\mathbf{r}_1,\mathbf{r}_2),\,\mathbf{J}^{t}_{R_n}(\mathbf{r}_2)\rangle\\ &=-\omega\mu k\sum_{J',M'}\big\langle\mathbf{J}_{R_m}(\mathbf{r}_1),\,\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}^{Nt}_{J',M'}(\mathbf{r}_{13},t_{13})\big\rangle\cdot\Big(\frac{t_{13}}{t_{34}}\Big)^{J'}\\ &\quad\cdot\sum_{J,M}\frac{1}{t_{34}}\cdot\overline{\overline{\widetilde{\alpha}}}^{N}_{J'M',JM}(\mathbf{r}_{34},t_{34})\cdot\big\langle\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}^{N*}_{J,M}(\mathbf{r}_{24},t_{24}),\,\mathbf{J}^{t}_{R_n}(\mathbf{r}_2)\big\rangle\cdot\Big(\frac{t_{24}}{t_{34}}\Big)^{J},\end{aligned}\tag{4.3}$$

where $\overline{\overline{\widetilde{\alpha}}}^{N}_{J'M',JM}(\mathbf{r}_{34},t_{34})$ is given in (2.29), and

$$\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}^{N}_{J',M'}(\mathbf{r}_{13},t_{13})=t_{13}^{-J'}\,\mathfrak{R}g\overline{\overline{\widetilde{\boldsymbol{\Psi}}}}_{J',M'}(\mathbf{r}_{13},t_{13}).$$

After obtaining the factorization of the matrix $\overline{\mathbf{Z}}^{V}_{\rm RWG}$, the implementation process of the multiplication of the matrix $\overline{\mathbf{Z}}^{V}_{\rm RWG}$ and a vector is similar to the traditional LF-FMA [21]. Then we present the scalar potential part with LF-VFMA. In the scalar potential part, the main calculation is the multiplication of the matrix $\overline{\mathbf{Z}}^{S}_{\rm CC}$ and a vector. By using (2.10a) and (2.34), we can expand the element in the non-near neighbor part of the matrix $\overline{\mathbf{Z}}^{S}_{\rm CC}$ as

$$\begin{aligned}&\big[\overline{\mathbf{Z}}^{S}_{\rm CC}\big]_{m,n}=\frac{-ik^2}{\omega\epsilon}\langle\mathbf{J}_{C_m}(\mathbf{r}_1),\frac{\nabla\nabla'}{k^2}g(\mathbf{r}_1,\mathbf{r}_2),\mathbf{J}^{t}_{C_n}(\mathbf{r}_2)\rangle\\ =&\frac{-ik^2}{\omega\epsilon}\sum_{J,M}\sum_{J',M'}\big\langle\mathbf{J}_{C_m}(\mathbf{r}_1),\,\mathfrak{R}g\mathbf{L}_{J',M'}(\mathbf{r}_{13})\big\rangle\cdot\alpha_{J'M',JM}(\mathbf{r}_{34})\cdot\big\langle\mathfrak{R}g\mathbf{L}^{*t}_{J,M}(\mathbf{r}_{24}),\,\mathbf{J}^{t}_{C_n}(\mathbf{r}_2)\big\rangle.\end{aligned}\tag{4.4}$$

If we adopt (4.4) to calculate the matrix $\overline{\mathbf{Z}}_{CC}^{S}$ in the scalar potential part of low-frequency large-scale problems, the subtraction error accumulated in the basis rearrangement process will lead to the divergence of the iteration. To eliminate subtraction error, we can use the patch-pair basis to express charges [22]. One way to define a complete set of patch-pair basis for a given mesh on a single 3D body or open surface is: one patch is chosen as a common negative patch and each of the other patches, as a positive patch [22].

For defining the single patch basis, we first give the definition of the RWG basis [26]. The RWG basis function $\mathbf{J}_{R_n}$ is defined on the n-th inner edge, which comprises a pair of triangles T_n^+ and T_n^-.

$$\mathbf{J}_{R_n}(\mathbf{r})=\begin{cases} \dfrac{\mathbf{r}-\mathbf{r}_n^+}{2S_n^+}, & \mathbf{r}\in T_n^+, \\ \dfrac{\mathbf{r}_n^- - \mathbf{r}}{2S_n^-}, & \mathbf{r}\in T_n^-, \\ 0, & \text{otherwise}, \end{cases} \tag{4.5}$$

where $S_n^{\pm}$ is the area of triangle $T_n^{\pm}$, respectively. Vectors $\mathbf{r}_n^{\pm}$ represent the free vertices of the triangle pair. Then, for each triangle patch, we define the single patch basis function $J_{P_m}(\mathbf{r})$ as

$$J_{P_m}(\mathbf{r})=\begin{cases} 1/S_m, & \mathbf{r}\in T_m, \\ 0, & \text{otherwise}. \end{cases} \tag{4.6}$$

Then a matrix $\overline{\mathbf{D}}\in\mathbb{R}^{N_P\times N_R}$ [25] is given to reveal the correspondence between the RWG basis function and the triangle pair.

$$\mathbf{D}_{ij}=\begin{cases} 0, & \text{Patch } i \text{ does not belong to RWG } j, \\ 1, & \text{Patch } i \text{ is the positive patch of RWG } j, \\ -1, & \text{Patch } i \text{ is the negative patch of RWG } j. \end{cases} \tag{4.7}$$

For each RWG basis function, we find the two patches that belong to this RWG basis function from the matrix $\overline{\mathbf{D}}$. If $D_{t_1 m}=1$ and $D_{t_2 m}=-1$, it means that the t_1-th patch and the t_2-th patch are the positive and negative patches of the m-th RWG basis function, respectively.

By using patch-pair basis, the matrix in the second term on the left hand side of (3.6) can be written as

$$\begin{bmatrix} \overline{\mathbf{0}} & \overline{\mathbf{0}} \\ \overline{\mathbf{0}} & i\omega\epsilon\overline{\mathbf{K}}^{t-1}\cdot\overline{\mathbf{Z}}_{CC}^{S}\cdot\overline{\mathbf{K}}^{-1} \end{bmatrix}=\begin{bmatrix} \overline{\mathbf{0}} & \overline{\mathbf{0}} \\ \overline{\mathbf{0}} & i\omega\epsilon\overline{\mathbf{W}}^{t}\cdot\overline{\mathbf{Z}}_{PP}^{S}\cdot\overline{\mathbf{W}} \end{bmatrix}, \tag{4.8}$$

where the matrix

$$\overline{\mathbf{W}}=\overline{\mathbf{D}}\cdot\overline{\mathbf{F}}_{CR}^{t}\cdot\overline{\mathbf{K}}^{-1},$$

and $\overline{\mathbf{F}}_{CR}$ is a $N_C\times N_R$ transformation matrix that changes from the RWG basis to the tree basis. Here, N_P is the number of single patches, N_C is the number of tree basis functions and $N_P = N_C + 1$. Moreover, $\overline{\mathbf{Z}}^S_{PP}$ is a $N_P \times N_P$ matrix, whose element in the non-near neighbor part can be written as

$$\begin{aligned}
&\big[\overline{\mathbf{Z}}^S_{PP}\big]_{mn} = -\frac{i}{\omega\epsilon}\big\langle J_{P_m}(\mathbf{r}_1),\, g(\mathbf{r}_1,\mathbf{r}_2),\, J_{P_n}(\mathbf{r}_2)\big\rangle \\
&= \frac{k}{\omega\epsilon}\sum_{J,M}\sum_{J',M'}\big\langle J_{P_m}(\mathbf{r}_1), \mathfrak{R}g\psi_{J',M'}(\mathbf{r}_1-\mathbf{r}_{l'})\big\rangle\cdot\alpha_{J'M',JM}(\mathbf{r}_{l'}-\mathbf{r}_l)\cdot\big\langle \mathfrak{R}g\psi^*_{JM}(\mathbf{r}_2-\mathbf{r}_l), J_{P_n}(\mathbf{r}_2)\big\rangle, \quad (4.9)
\end{aligned}$$

where $\mathbf{r}_{l'}$ denotes the center of the box to which the single patch basis function J_{P_m} belongs.

From (4.8) and (4.9), we see that the main calculation in the scalar potential part of the LF-VFMA has been converted from the multiplication of the matrix $\overline{\mathbf{Z}}^S_{CC}$ and a vector to the multiplication of the matrix $\overline{\mathbf{Z}}^S_{PP}$ and a vector. By doing so, the accumulation of subtraction error can be avoided.

To show the memory saving in the part of radiation and receiving patterns of the LF-VFMA, we first consider radiation and receiving patterns in (4.4), even though they need not be used in practice. Since radiation and receiving patterns are similar, we only consider receiving patterns in the following. By using the integration by parts, receiving patterns in (4.4) can be calculated by using the patch pair basis function

$$\begin{aligned}
&\big\langle \mathbf{J}_{C_m}(\mathbf{r}_1),\, \mathfrak{R}g\mathbf{L}_{J',M'}(\mathbf{r}_1-\mathbf{r}_l)\big\rangle \\
&= -\frac{1}{k}\big\langle \nabla\cdot\mathbf{J}_{C_m}(\mathbf{r}_1), \mathfrak{R}g\psi_{J',M'}(\mathbf{r}_1-\mathbf{r}_l)\big\rangle \\
&= -\frac{1}{k}\Big(\big\langle J_{P_{t_1}}(\mathbf{r}_1), \mathfrak{R}g\psi_{J',M'}(\mathbf{r}_1-\mathbf{r}_l)\big\rangle - \big\langle J_{P_{t_2}}(\mathbf{r}_1), \mathfrak{R}g\psi_{J',M'}(\mathbf{r}_1-\mathbf{r}_l)\big\rangle\Big), \quad (4.10)
\end{aligned}$$

where $\mathbf{r}_l$ is the center of the box to which the RWG basis function $\mathbf{J}_{C_m}$ belongs. Let the center of the box to which the patch basis function $J_{P_{t_i}}$ belongs be $\mathbf{r}_{t_i}$, $i = 1$, 2. It should be noted that $\mathbf{r}_{t_1}$ or $\mathbf{r}_{t_2}$ is not always equal to $\mathbf{r}_l$ [21]. If $\mathbf{r}_{t_1} \neq \mathbf{r}_l$, a translator is needed to transform $\langle J_{P_{t_1}}(\mathbf{r}_1), \mathfrak{R}g\psi_{J',M'}(\mathbf{r}_1-\mathbf{r}_{t_1})\rangle$ to $\langle J_{P_{t_1}}(\mathbf{r}_1), \mathfrak{R}g\psi_{J',M'}(\mathbf{r}_1-\mathbf{r}_l)\rangle$. By using the scalar addition theorem [21], we know

$$\mathfrak{R}g\psi_{J,M}(\mathbf{r}) = \sum_{J',M'}\mathfrak{R}g\psi_{J',M'}(\mathbf{r}')\cdot\beta_{J'M',JM}(\mathbf{r}''), \quad \forall J,\ \forall|\mathbf{r}'|,\ |\mathbf{r}''|, \quad (4.11)$$

where $\mathbf{r} = \mathbf{r}' + \mathbf{r}''$. So it can be obtained that

$$\big\langle J_{P_{t_1}}(\mathbf{r}_1), \mathfrak{R}g\psi_{J'M'}(\mathbf{r}_1-\mathbf{r}_l)\big\rangle = \sum_{J,M}\big\langle J_{P_{t_1}}(\mathbf{r}_1), \mathfrak{R}g\psi_{JM}(\mathbf{r}_1-\mathbf{r}_{t_1})\big\rangle\cdot\beta_{JM,J'M'}(\mathbf{r}_{t_1}-\mathbf{r}_l). \quad (4.12)$$

Since the box to which a patch triangle of an RWG basis function belongs is adjacent to or the same as the box to which the RWG basis function belongs, the vector $\mathbf{r}_{t_1} - \mathbf{r}_l$ only has 27 possibilities at most, which is independent of the number of unknowns. So the storage of translators $\beta_{J'M',JM}$ is small and remains constant.

In addition, by applying (2.40), we can express the third component of receiving patterns (4.2a) in the vector potential part of the LF-VFMA as

$$\begin{aligned}&\left\langle \mathbf{J}_{R_i}(\mathbf{r}_1), \Re g\mathbf{T}_{JM}(\mathbf{r}_1-\mathbf{r}_{l_i})\right\rangle\\ =&\frac{\sqrt{2J+1}}{\sqrt{J}}\left\langle \mathbf{J}_{R_i}(\mathbf{r}_1), \Re g\mathbf{L}_{JM}(\mathbf{r}_1-\mathbf{r}_{l_i})\right\rangle - \frac{\sqrt{J+1}}{\sqrt{J}}\left\langle \mathbf{J}_{R_i}(\mathbf{r}_1), \Re g\mathbf{H}_{JM}(\mathbf{r}_1-\mathbf{r}_{l_i})\right\rangle. \end{aligned} \tag{4.13}$$

Since the subtraction on the right hand side of (4.13) is between two terms with different highest orders, the real value of $\left\langle \mathbf{J}_{R_i}(\mathbf{r}_1), \Re g\mathbf{T}_{JM}(\mathbf{r}_1-\mathbf{r}_{l_i})\right\rangle$ will not be swamped by the subtraction error.

Therefore, in the scalar potential part of the LF-VFMA, we only store receiving patterns $\left\langle J_{P_m}(\mathbf{r}_1), \Re g\psi_{J',M'}(\mathbf{r}_1-\mathbf{r}_{l'})\right\rangle$ with the single patch basis. For receiving patterns (4.2a) in the vector potential part of the LF-VFMA, only the first two components need to be calculated and stored. By using (4.10), (4.12) and (4.13), the third component can be obtained with the second component of receiving patterns in the vector potential part, receiving patterns of the scalar potential part and translators introduced in (4.11). Hence, the total storage of receiving patterns contains three items: one is receiving patterns with the single patch basis in the scalar potential part and the other two are the first two components of receiving patterns in the vector potential part.

We know the storage of receiving patterns in the vector potential part of the LF-FMA contains three items, while the storage of receiving patterns in the scalar potential part of the LF-FMA is the same as that of the LF-VFMA, which contains one item. Therefore, compared with the LF-FMA, the storage of receiving patterns in the vector potential part of the LF-VFMA is reduced from $3N_RP$ to $2N_RP$. Moreover, the storage of receiving patterns in the scalar potential part is still N_PP. Therefore, the total storage of radiation and receiving patterns of the LF-VFMA can be reduced by 25 percent compared with that of the LF-FMA.

As for the storage for vector translators, since elements of vector translators can be calculated with scalar translators of the LF-FMA, theoretically we can only store scalar translators for generating vector translators. However, as we have discussed in Section 2, some elements of vector translators should be calculated and stored alone to avoid numerical error due to the subtraction of two high order terms. The storage of special elements in vector translators approximately equals that of scalar translators. Therefore, the storage for vector translators, containing the storage of scalar translators and special elements, is larger than that for scalar translators, which depends on the number of buffer boxes and the number of levels of FMA. In the numerical simulation, if we only store scalar translators and special elements for calculating vector translators in the iteration process, repeated calculations will increase floating point operations dramatically for large scale problems. Therefore, besides the scalar translators and special elements, we consider to store fundamental elements of vector translators, $A_{J_1M_1,J_2M_2}$ and $B_{J_1M_1,J_2M_2}$ used in (2.20), to reduce part of repeated calculations, which is at the expense of raising storage for vector translators. In addition, we should store translators used in (4.12), whose storage is small and remains constant.

Fortunately, the storage for all translators used in this method is independent of the number of unknowns. With the increase in the number of unknowns, the storage of radiation and receiving patterns, which depends on the number of unknowns, will dominate over that of translators. It is reduced by 25 percent in our method compared with that in the LF-FMA. The analysis of memory requirement implies that the method is superior in saving memory for solving large scale problems.

5 Computational complexity of the LF-VFMA

In the implementation of the LF-VFMA, only nonempty boxes are involved in the computation. If each box at the finest level has on the average M subscatterers, then the number of boxes at the finest level is about N_R/M. Since it is a 3D algorithm, the number of boxes at the next coarser level is roughly $1/I$ of that at the current level, where I is a number smaller than 8. Moreover, the number of levels is roughly

$$L=\log_I \frac{N_R}{M}-1.$$

According to the scale invariance of the low-frequency case, the number of the multipole spherical waves can be chosen the same at different levels. Let the number of the multipole spherical harmonics at the finest level be K.

The floating-point operations related directly to subscatterers are the aggregation from subscatterers to the finest level, the disaggregation from the finest level to subscatterers, and the interactions of the neighbor terms. Different from the LF-FMA, the operations of the LF-VFMA related directly to subscatterers contain the translations described in (4.12), whose floating-point operations is $C_0^t K^2 N_R$ in the worst-case. However, the workload of this kind of translations in practice is much less than that in the worst-case. So the total floating-point operations related to the subscatterers is

$$T_0=C_0' M N_R+C_0'' K N_R+C_0^t K^2 N_R=(C_0' M+C_0'' K+C_0^t K^2)N_R. \tag{5.1}$$

The levels are indexed from the finest level to the coarsest level successively with $l=1,2,\cdots,L$. Compared with LF-FMA, in the processes of aggregations from level l to $l+1$, disaggregations from level $l+1$ to level l and translations at level l, tiny extra calculations are needed for generating the vector translators by using the scalar translators and some special elements of vector translators, which have been calculated and stored in the setup stage. Therefore, the workload for the aggregation from level l to $l+1$ and the disaggregation from level $l+1$ to level l are

$$T_l'=\begin{cases} C'\left(\frac{1}{I}\right)^l \frac{N_R}{M} K^2+C_t'\left(\frac{1}{I}\right)^l \frac{N_R}{M} K, & l=1,2,\cdots,L-1, \\ 0, & l=L. \end{cases} \tag{5.2}$$

The workload for the translations at level l is

$$T_l'' = C''\Big(\frac{1}{I}\Big)^l \frac{N_R}{M} K^2 + C_t''\Big(\frac{1}{I}\Big)^l \frac{N_R}{M} K, \quad l=1,2,\cdots,L, \tag{5.3}$$

so the total workload per iteration is

$$\begin{aligned} T &= T_0 + \sum_{l=1}^{L}(T_l' + T_l'') \\ &= \Big\{ C_0' M + C_0'' K + C_0^t K^2 + \Big[C'\Big(1-\frac{1}{I^{L-1}}\Big) + C''\Big(1-\frac{1}{I^L}\Big)\Big]\frac{K^2}{M} \\ &\quad + \Big[C_t'\Big(1-\frac{1}{I^{L-1}}\Big) + C_t''\Big(1-\frac{1}{I^L}\Big)\Big]\frac{K}{M}\Big\} N_R. \end{aligned} \tag{5.4}$$

In addition, we know that the floating-point operations for applying the basis rearrangement also scale as $\mathcal{O}(N_R)$ [19, 21]. Therefore, the computational complexity of the LF-VFMA is clearly $\mathcal{O}(N_R)$, which is the same as that of the LF-FMA.

6 Numerical examples

This section presents some numerical examples to demonstrate the validity of the LF-VFMA. These examples are all PEC structures in free space.

1. Sphere scattering To verify the efficiency of this method for low frequency problems, the scattering problem of a sphere is considered. An incident x-polarized plane wave is from the z direction onto a PEC sphere with radius 1 m. The triangulation of the sphere contains 28,065 inner edges. By using the 5 level LF-VFMA, we calculate the bistatic radar cross section (RCS) of the PEC sphere with fixed $\phi=0$ at 10^{-7} GHz, where the size of the leafy level box is $2.21465\times10^{-8}\lambda$ (λ is the wavelength). The number of buffer box is 1. Then we compare the result with the Mie series solution. Fig. 1 shows the agreement of the radar cross sections between the numerical result of the LF-VFMA and the Mie series solution.

2. Cube scattering A PEC cube is excited by an x-polarized plane wave incident from the z direction. The edge length of the cube is 1 m. In the numerical simulation, we use four different discretizations from coarsest to densest. The LF-VFMA is used to calculate the bistatic RCS of the PEC cube with fixed $\phi=0$ at 10^{-6} GHz for four meshes. The number of buffer boxes is 1 and the number of multipoles is chosen as five to get a better accuracy. The GMRES-30 is used to solve matrix equations. The computational costs are shown in Table 1 and Table 2. Relative errors of RCS given in the last column of the second table are between the LF-VFMA and LF-FMA for different meshes.

3. Almond scattering We use an x-polarized plane wave incident from the z direction to excite a PEC NASA almond given in Fig. 2, whose length $d=2.52374$ m. The mesh has

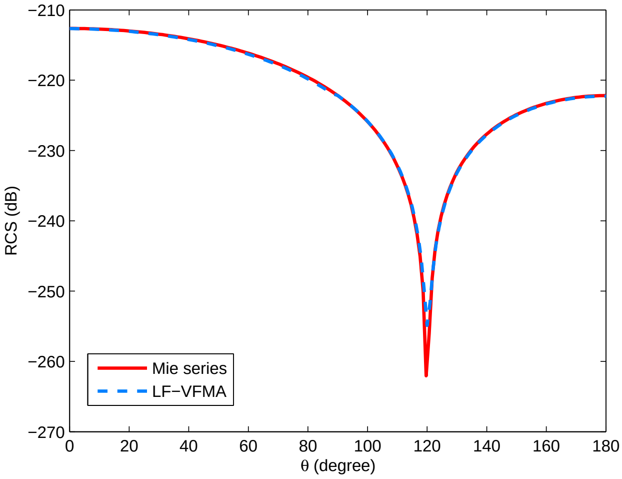

Figure 1: The bistatic RCS of a PEC sphere at 10^{-7} GHz using LF-VFMA. The reference solution is calculated by the Mie series.

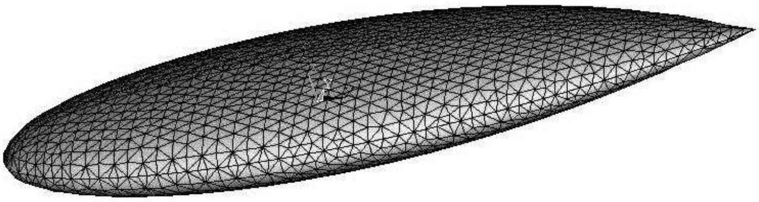

Figure 2: NASA almond.

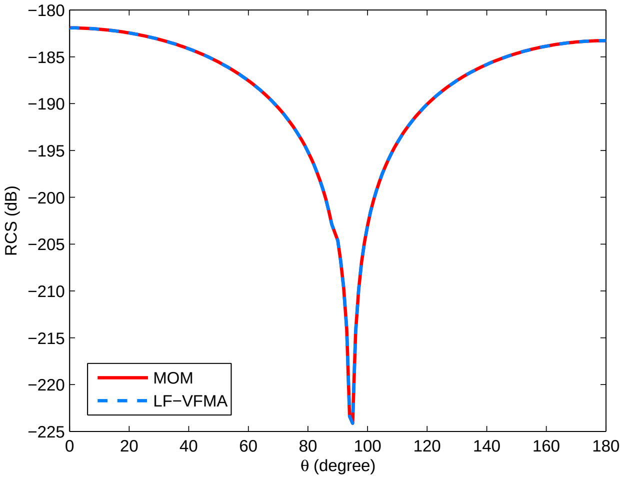

Figure 3: Bistatic RCS of a PEC NASA almond.

Table 1: Memory usages of translators and radiation and receiving patterns of the LF-FMA and LF-VFMA for different meshes.

Method	Number of Unknowns	Number of Levels	Memory of Translators (MB)	Memory of Rad and Rec patterns (MB)
LF-FMA	7,200	4	9.69	14.50
LF-VFMA			47.56	10.55
LF-FMA	45,000	5	12.97	90.64
LF-VFMA			63.49	65.92
LF-FMA	180,000	6	15.70	362.55
LF-VFMA			76.73	263.67
LF-FMA	281,250	6	15.70	566.48
LF-VFMA			76.73	411.99

Table 2: The number of iterations and the average time per iteration of the LF-VFMA for different meshes.

Method	Number of Unknowns	Number of Levels	Number of Iterations	Average Time per Iteration (s)	Relative Error of RCS
LF-VFMA	7,200	4	31	29.18	2.95×10^{-4}
LF-VFMA	45,000	5	71	94.30	1.33×10^{-4}
LF-VFMA	180,000	6	142	330.38	4.74×10^{-4}
LF-VFMA	281,250	6	181	322.24	4.87×10^{-4}

5,574 inner edges. We use the 6 level LF-VFMA to calculate the bistatic RCS of the PEC NASA almond with fixed $\phi=0$ at 10^{-6} GHz. The GMRES-50 is used to solve the matrix equation. The result of the LF-VFMA is compared with that of the method of moments (MOM) in Fig. 3.

7 Conclusions

In this work, the vector addition theorem is adopted for the factorization of the dyadic Green's function. Then the factorization is used to develop the LF-VFMA for realizing memory savings. In this method, the storage of translators, which depends on the number of buffer boxes and the number of levels of FMA, is larger than that of the LF-FMA. Although there is an increase in the storage of translators, it is independent of the number of unknowns. Meanwhile, the storage of radiation and receiving patterns in the non-near neighbor part can be reduced by 25 percent compared with that of the LF-FMA. Since the storage of radiation and receiving patterns depends on the number of unknowns, it will become the main part of the total storage with the increase in the scale of problems. So the reduction of the storage of radiation and receiving patterns is meaningful for large scale problems.

Compared with the LF-FMA, the calculation of near field part remains unchanged, and the total storage for the non-near neighbor part in this method is reduced for large scale problems. Therefore the LF-VFMA is advantageous for solving large scale problems in saving memory.

References

[1] W. W. Hansen, A new type of expansion in radiation problems, Phys. Rev., 47 (1935), 139–143.

[2] J. A. Stratton, Electromagnetic Theory, New York, McGrawHill, 1941.

[3] S. Stein, Addition theorems for spherical wave functions, Quarter. Appl. Math., 19 (1961), 15–24.

[4] M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions, New York, Dover, 1965.

[5] D. R. Wilton and A. W. Glisson, On improving the electric field integral equation at low frequencies, In Proc. URSI Radio Sci. Meet. Dig., Los Angeles, CA, June 1981, pp. 24.

[6] S. M. Rao, D. R. Wilton and A. W. Glisson, Electromagnetic scattering by surface of arbitary shape, IEEE Trans. Antennas. Propagat., 30 (3) (1982), 409–418.

[7] J. R. Mautz and R. F. Harrington, An E-field solution for a conducting surface small or comparable to the wavelength, IEEE Trans. Antennas Propagat., 32 (1984), 330–339.

[8] L. Tsang, J. A. Kong and R. T. Shin, Theory of Microwave Remote Sensing, New York, John Wiley & Sons, 1985.

[9] E. Arvas, R. F. Harrington and J. R. Mautz, Radiation and scattering from electrically small conducting bodies of arbitrary shape, IEEE Trans. Antennas. Propagat., 34 (1986), 66–77.

[10] V. Rokhlin, Rapid solution of integral equations of scattering theory in two dimensions, J. Comput. Phys., 86 (1990), 414–439.

[11] J. S. Lim, S. M. Rao and D. R. Wilton, A novel technique to calculate the electromagnetic scattering by surfaces of arbitrary shape, In Proc. URSI Radio Sci. Meet. Dig., Ann Arbor, MI, June 1993, pp. 322.

[12] C. C. Lu and W. C. Chew, A multilevel algorithm for solving a boundary integral equation of wave scattering, Microwave Opt. Tech. Lett., 7 (10) (1994), 466–470.

[13] W. Wu, A. W. Glisson and D. Kajfez, A comparison of two low-frequency formulations for the electric field integral equation, In Proc. 10th Ann. Rev. Progress in Applied Computat. Electromagn., Monterey, CA, 2, 484–491, Mar. 1994.

[14] M. Burton and S. Kashyap, A study of a recent, moment-method algorithm that is accurate to very low frequencies, Appl. Comput. Electromagn. Soc. J., 10 (3) (1995), 58–68.

[15] W. C. Chew, Waves and Fields in Inhomogeneous Media, New York, IEEE Press, 1995.

[16] G. Dassios and Z. Rigou, Elastic Herglotz functions, SIAM J. Appl. Math., 55 (5) (1995), 1345–1361.

[17] W. Wu, A. W. Glisson and D. Kajfez, A study of two numerical solution procedures for the electric field integral equation at low frequency, Appl. Computat. Electromagn. Soc. J., 10 (3) (1995), 69–80.

[18] J. M. Song and W. C. Chew, Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering, Microwave Opt. Tech. Lett., 10 (1) (1995), 14–19.

[19] J. S. Zhao and W. C. Chew, Three dimensional multilevel fast multipole algorithm from staitc to electrodynamic, Microwave Opt. Tech. Lett., 26 (1) (2000), 43–48.

[20] J. S. Zhao and W. C. Chew, Integral equation solution of Maxwell's equations from zero frequency to microwave frequencies, IEEE transactions on antennas and propagation, 48

(10) (2000), 1635–1645.

[21] W. C. Chew, J. M. Jin, E. Michielssen and J. M. Song, Fast and Efficient Algorithms in Computational, Electromagnetics. Boston, MA, Artech House, 2001.

[22] Y. H. Chu and W. C. Chew, Large-scale computation for electrically small structures using surface-integral equation method, Micro. Opt. Tech. Lett., 47 (6) (2005), 525–530.

[23] L. J. Jiang and W. C. Chew, A mixed-form fast multipole algorithm, IEEE Trans. Antennas. Propagat., 53 (12) (2005), 4145–4156.

[24] B. He and W. C. Chew, Diagonalizations of vector and tensor addition theorems, Commun. Comput. Phys., 4 (4) (2008), 797–819.

[25] Z. G. Qian and W. C. Chew, An augmented electric field integral equation for low frequency electromagnetics analysis, IEEE International Symposium on Antennas and Propagation and USNC/URSI National Radio Science Meeting, San Diego, 2008.

[26] Z. G. Qian and W. C. Chew, Fast full-wave surface integral equation solver for multiscale structure modeling, IEEE Trans. Antennas. Propagat., 57 (11) (2009), 3594–3601.

[27] W. C. Chew, M. S. Tong and B. Hu, Integral Equation Methods for Electromagnetic and Elastic Waves, Morgan and Claypool Publishers, 2009.