

A Pre-splitting Green's Function based Hybrid Fast Algorithm for Multiscale Problems

Guang-Yu Zhu^{1,2}, Wei-Dong Li², Wei E. I. Sha¹, Hou-Xing Zhou², and Wei Hong²

¹College of Information Science and Electronic Engineering
Zhejiang University, Hangzhou, 310027, China
gyzhu.china@163.com

²State Key Laboratory of Millimeter Waves
Frontiers Science Center for Mobile Information Communication and Security
Southeast University, Nanjing, 210096, China

Abstract – Based on the splitting form of the Green's function, a hybrid fast algorithm is proposed for efficient analysis of multiscale problems. In this algorithm, the Green's function is a priori split into two parts: a spectrally band-limited part and a spatially localized part. Then, the fast Fourier transforms (FFT) utilizing the global Cartesian grid and the matrix compression method aided by an adaptive octree grouping are implemented for these two parts, respectively. Compared with the traditional methods which only employ the FFT for acceleration, the proposed hybrid fast algorithm is capable of maintaining low memory consumption in multiscale problems without compromising time cost. Moreover, the proposed algorithm does not need cumbersome geometric treatment to implement the hybridization, and can be established in a concise and straightforward manner. Several numerical examples discretized with multiscale meshes are provided to demonstrate the computational performance of proposed hybrid fast algorithm.

Index Terms – Fast algorithm, fast Fourier transform, Green's function, matrix compression, multiscale problems.

I. INTRODUCTION

Many real-world electromagnetic (EM) problems require multiscale discretization [1–4]. When the method of moments (MoM) [5], along with classical fast algorithms, is used to model the multi-scale problems, a single fast algorithm [6–12] is often insufficient to achieve satisfactory computational performance. Instead, due to capturing both the circuit physics and wave physics [1], hybrid fast algorithms [19–22] generally provide a more practical path to efficiently solve EM multiscale problems.

As we know, the multilevel fast multipole algorithm (MLFMA) is widely used for efficient simula-

tion of electrically large problems [6–8]. However, the MLFMA encounters the sub-wavelength breakdown [8] when dense mesh occurs. As a remedy, low-frequency fast algorithms [23, 24] were studied. Correspondingly, a series of hybrid algorithms were developed [25–30], such as mixed-form FMA [19], interpolative decomposition (ID)-MLFMA [20], and adaptive cross approximation (ACA)-MLFMA [21].

Different from the MLFMA, the pre-correction-fast Fourier transforms (FFT)-based methods [9–18] (e.g., adaptive integral method (AIM) [9], pre-corrected FFT (P-FFT) [10], and integral equation FFT (IE-FFT) [11]) are free from the sub-wavelength breakdown. Nevertheless, due to the intrinsic uniformity of the global Cartesian grids, the pre-correction-FFT-based methods do not allow for sufficient and varying spatial resolutions to handle multiscale discretizations. That is, by simply tuning the grid spacing, it is hard to achieve high computational efficiency for far interactions and maintain low storage efficiency for the near matrix at the same time [31].

Recently, in order to overcome this shortcoming of the pre-correction-FFT-based methods, a hybrid fast algorithm was developed in [31]. In this algorithm, the pre-correction-FFT-based methods is augmented with the matrix compression method (ACA) [36–38]. However, to implement such a hybridization, the procedures are not as straightforward as they seem [31]. In particular, a complicated spatial grouping strategy is inherently needed, and the relations between the required auxiliary geometric data structures (such as the Cartesian grid, the expansion box, and the tree-cube) are complex and require cautious treatments [31]. Such a situation motivates us to reconsider the overall framework of the pre-correction-FFT-based methods and wonder if there exists a more elegant way to develop a hybrid algorithm that consists of

both the FFT and the matrix compression methods (e.g., ACA).

As an interesting alternative of the well-studied pre-correction-FFT-based methods, some pre-splitting-FFT-based method was developed in [32, 33]. Basically, the pre-splitting-FFT-based method bears similarities with the well-known pre-correction-FFT-based methods, since it also uses the Cartesian grid and the FFT as the essentials for the acceleration of the spatial convolutions. However, different from the widely-used pre-correction-FFT-based methods which typically utilize the Cartesian grid-based near-field corrections (based on some local expansion boxes) to compensate for the near-field contributions, the pre-splitting-FFT-based method [32, 33] instead resorts to some splitting form of the Green's function so as to realize accurate and efficient computations. Despite of its unique feature, this pre-splitting-based framework has not attracted enough attentions in the computational electromagnetics (CEM) community, and a hybrid fast algorithm based on such framework has never been explored yet.

In this work, based on the splitting form of the Green's function, a hybrid fast algorithm is proposed for efficiently analyzing multiscale problems. Specifically, the pre-splitting-FFT-based method originally devised for the problems with quasi-uniform discretizations is here further enhanced with the matrix compression method (ACA). Thus, a hybrid fast algorithm using both the FFT and the matrix compression method is established. In particular, the proposed hybrid fast algorithm is capable of maintaining low memory consumption in multiscale problems without compromising time cost, thus manifesting itself as a favorable multiscale extension of the pure FFT-based method. Furthermore, the required auxiliary geometric data structures herein (including the Cartesian grid and the spatial grouping octree) can be constructed independent of each other, and are not intertwined as in [31]. Consequently, the proposed hybrid algorithm can be implemented in a straightforward manner without any cumbersome geometric treatment.

In the following, after introducing the basic formulations and implementation key points in Section II, several numerical examples discretized with multiscale meshes are then provided in Section III to demonstrate the computational performances of the proposed hybrid fast algorithm for multiscale problems.

II. FORMULATION

A perfectly electric conductor (PEC) object illuminated by an incident plane wave is considered. The combined field integral equation (CFIE) is employed to model this typical electromagnetic scattering problem. Throughout the paper, we use λ to denote the wave-

length in free space. Moreover, we use k and η to denote the wave number and the wave impedance in free space, respectively.

A. Combined field integral equation

The CFIE is the linear combination of the electric field integral equation (EFIE) and the magnetic field integral equation (MFIE), which is conventionally expressed as

$$\text{CFIE} = \alpha \text{EFIE} + (1 - \alpha) \eta \text{MFIE}, \quad (1)$$

where α denotes the combination factor and $0 \leq \alpha \leq 1$, the EFIE is formulated as

$$\mathbf{E}^{\text{inc}}(\mathbf{r}) \Big|_{\text{tan}} = \left[jk\eta \int_S \left(\mathbf{J}(\mathbf{r}') + \frac{\nabla \nabla'}{k^2} \mathbf{J}(\mathbf{r}') \right) G(\mathbf{r}, \mathbf{r}') ds' \right] \Big|_{\text{tan}}, \quad (2)$$

and the MFIE is formulated as

$$\mathbf{H}^{\text{inc}}(\mathbf{r}) \Big|_{\text{tan}} = \left[\frac{\hat{\mathbf{n}} \times \mathbf{J}(\mathbf{r}')}{2} + P.V. \int_S \mathbf{J}(\mathbf{r}') \times \nabla G(\mathbf{r}, \mathbf{r}') ds' \right] \Big|_{\text{tan}}. \quad (3)$$

In the above, \mathbf{E}^{inc} and \mathbf{H}^{inc} are the incident electric field and the incident magnetic field, respectively. Moreover, $G(\mathbf{r}, \mathbf{r}')$ denotes the Green's function with the field point \mathbf{r} and the source point \mathbf{r}' . Furthermore, $\mathbf{J}(\mathbf{r}')$ denotes the surface current.

By using the method of moments (MoM) [5] with the RWG function [39], the CFIE can be discretized into a matrix equation

$$\mathbf{Z}^{[N \times N]} \cdot \mathbf{I}^{[N \times 1]} = \mathbf{V}^{[N \times 1]}, \quad (4)$$

where \mathbf{Z} is the impedance matrix, \mathbf{I} is the unknown surface current coefficients vector, \mathbf{V} is the righthand side vector of the incident field, and N is the number of unknowns. More specifically, the impedance matrix \mathbf{Z} corresponding to the CFIE is described by

$$\mathbf{Z}^{\text{CFIE}} = \alpha \mathbf{Z}^{\text{EFIE}} + (1 - \alpha) \eta \mathbf{Z}^{\text{MFIE}}, \quad (5)$$

where

$$\begin{aligned} \mathbf{Z}_{mn}^{\text{EFIE}} &= jk\eta \int_{S_m} ds \mathbf{f}_m(\mathbf{r}) \cdot \int_{S_n} G(\mathbf{r}, \mathbf{r}') \mathbf{f}_n(\mathbf{r}') ds' \\ &- j \frac{\eta}{k} \int_{S_m} ds [\nabla \cdot \mathbf{f}_m(\mathbf{r})] \cdot \int_{S_n} G(\mathbf{r}, \mathbf{r}') [\nabla \cdot \mathbf{f}_n(\mathbf{r}')] ds', \end{aligned} \quad (6)$$

and

$$\begin{aligned} \mathbf{Z}_{mn}^{\text{MFIE}} &= \frac{1}{2} \int_{S_m} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}) ds \\ &+ \int_{S_m} ds [\hat{\mathbf{n}} \times \mathbf{f}_m(\mathbf{r})] \cdot \int_{S_n} \nabla G(\mathbf{r}, \mathbf{r}') \times \mathbf{f}_n(\mathbf{r}') ds'. \end{aligned} \quad (7)$$

In the above, $m = 1, \dots, N$ and $n = 1, \dots, N$. Moreover, \mathbf{f}_m and \mathbf{f}_n denote the m th and n th RWG functions, respectively. Furthermore, S_m and S_n are the supports of the corresponding RWG functions, respectively.

B. Pre-splitting the Green's function

In the following, the splitting forms of the Green's function and its gradient are introduced. First, consider the Green's function in the EFIE, namely

$$G(\mathbf{r}, \mathbf{r}') = \frac{e^{-jkR}}{4\pi R}, \quad (8)$$

where

$$R = |\mathbf{r} - \mathbf{r}'| = \sqrt{(x-x')^2 + (y-y')^2 + (z-z')^2} \quad (9)$$

denotes the distance between the field point \mathbf{r} and the source point \mathbf{r}' . Clearly, the above can also be equivalently represented as

$$G(\mathbf{r}, \mathbf{r}') = \text{Re}[G(\mathbf{r}, \mathbf{r}')] + j\text{Im}[G(\mathbf{r}, \mathbf{r}')], \quad (10)$$

where

$$\text{Re}[G(\mathbf{r}, \mathbf{r}')] = \frac{\cos kR}{4\pi R}, \quad (11)$$

$$\text{Im}[G(\mathbf{r}, \mathbf{r}')] = -\frac{\sin kR}{4\pi R}. \quad (12)$$

A basic property of $G(\mathbf{r}, \mathbf{r}')$ is that, this function is singular for $R = 0$, and its magnitude is globally nonzero for $R > 0$. Also, note that the singular behavior of $G(\mathbf{r}, \mathbf{r}')$ comes from $\text{Re}[G(\mathbf{r}, \mathbf{r}')]$.

Using the pre-splitting strategy [32, 33], the Green's function (8) can be written as two parts, namely

$$G(\mathbf{r}, \mathbf{r}') = G_E(\mathbf{r}, \mathbf{r}') + G_P(\mathbf{r}, \mathbf{r}'), \quad (13)$$

where

$$G_E(\mathbf{r}, \mathbf{r}') = \begin{cases} \text{Re}[G(\mathbf{r}, \mathbf{r}')] - \Phi(R), & R < \delta \\ 0, & R \geq \delta \end{cases}, \quad (14)$$

$$G_P(\mathbf{r}, \mathbf{r}') = \begin{cases} \Phi(R) + j\text{Im}[G(\mathbf{r}, \mathbf{r}')], & R < \delta \\ G(\mathbf{r}, \mathbf{r}'), & R \geq \delta \end{cases}, \quad (15)$$

with δ being a splitting threshold, and $\Phi(R)$ being some auxiliary function. Particularly, the above two parts manifest the following properties.

- Function $G_E(\mathbf{r}, \mathbf{r}')$ is a singular and spatially localized function. It has nonzero function value only when $R < \delta$, and $G_E(\mathbf{r}, \mathbf{r}') = 0$ for $R \geq \delta$.
- Function $G_P(\mathbf{r}, \mathbf{r}')$ is a globally oscillatory and spectrally band-limited function. It is smooth and bounded for $\forall \mathbf{r}$ and $\forall \mathbf{r}'$.

To have more intuitive understanding on the properties of function $G_E(\mathbf{r}, \mathbf{r}')$ and $G_P(\mathbf{r}, \mathbf{r}')$, the corresponding function images are shown in Fig. 1. Moreover, from a physical and spectral perspective, $G_E(\mathbf{r}, \mathbf{r}')$ mainly captures the *evanescent waves*, and $G_P(\mathbf{r}, \mathbf{r}')$ mainly captures the *propagating waves*.

To achieve the above splitting and corresponding properties, the auxiliary function $\Phi(R)$ is specially designed and one simple way is to choose $\Phi(R)$ to be a polynomial of the following form [32, 33]

$$\Phi(R) = aR^3 + bR^2 + c, \quad (16)$$

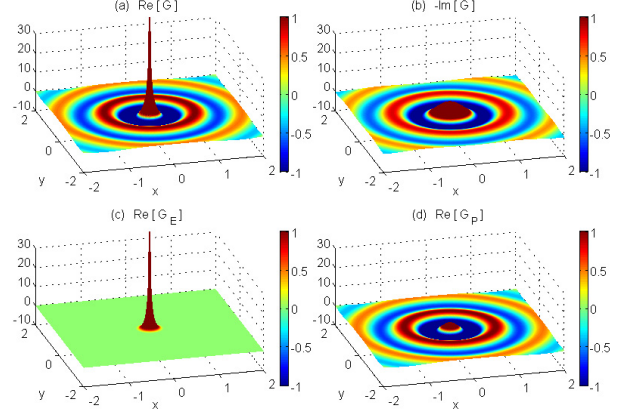


Fig. 1. The Green's function is a priori split into a singular but spatially localized part G_E and a globally oscillatory but spectrally band-limited part G_P . The relevant magnitudes of the original Green's function and of these two parts are visualized in the figure.

where a , b , and c are unknown coefficients. Clearly, the above is a three order polynomial without the linear term, and it implies

$$d\Phi(0)/dR = 0, \quad (17)$$

meaning that the first order derivative function of $\Phi(R)$ is zero at $R = 0$ (where $\mathbf{r} = \mathbf{r}'$).

Given the splitting threshold δ , the coefficients in (16) can be determined by some continuity conditions at $R = \delta$ as follows

$$\begin{cases} \Phi(\delta) = \text{Re}[G(\delta)] \\ d\Phi(\delta)/dR = d\text{Re}[G(\delta)]/dR \\ d^2\Phi(\delta)/dR^2 = d^2\text{Re}[G(\delta)]/dR^2 \end{cases}, \quad (18)$$

or, by solving the following matrix equation

$$\begin{bmatrix} \delta^3 & \delta^2 & 1 \\ 3\delta^2 & 2\delta & 0 \\ 6\delta & 2 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \text{Re}[G(\delta)] \\ d\text{Re}[G(\delta)]/dR \\ d^2\text{Re}[G(\delta)]/dR^2 \end{bmatrix}. \quad (19)$$

The derivative functions of $\text{Re}[G(R)]$ used in the above are given in the appendix. Thus, after $\Phi(R)$ is obtained, the splitting representation (13) is completely determined.

The polynomial choice (16), the implicit continuity condition (17), and the explicit continuity condition (18) are based on the consideration that, after the splitting, the resultant $G_P(\mathbf{r}, \mathbf{r}')$ (or, more precisely, $\text{Re}[G_P(\mathbf{r}, \mathbf{r}')]$) should be smooth everywhere for $\forall \mathbf{r}$ and $\forall \mathbf{r}'$, so that it can be interpolated accurately without much difficulty (the interpolation procedures will be discussed in the next subsection).

To make it more clear, without loss of generality, consider the following representative situation. The source point \mathbf{r}' is fixed to the coordinate origin, and the field point \mathbf{r} is moving along the x axis. Then, the corresponding function values of $G(\mathbf{r}, \mathbf{r}')$ and $G_P(\mathbf{r}, \mathbf{r}')$

are evaluated, as shown in Fig. 3. We can clearly see that $\text{Re}[G(\mathbf{r}, \mathbf{r}')] is singular when \mathbf{r} approaches \mathbf{r}' , while $\text{Re}[G_P(\mathbf{r}, \mathbf{r}')] is continuous and smooth everywhere. Here, $\text{Re}[G_P(\mathbf{r}, \mathbf{r}')] is smooth at $R = 0$ (where $\mathbf{r} = \mathbf{r}'$) as a result of (17), and smooth at the splitting point $R = \delta$ as a result of (18).$$$

Next, consider the gradient of the Green's function in the MFIE, namely

$$\nabla G(\mathbf{r}, \mathbf{r}') = \ddot{G}(R) \nabla R, \quad (20)$$

with

$$\ddot{G}(R) = \text{Re}[\ddot{G}(R)] + j \text{Im}[\ddot{G}(R)], \quad (21)$$

where

$$\text{Re}[\ddot{G}(R)] = -\frac{\cos kR + kR \sin kR}{4\pi R^2}, \quad (22)$$

$$\text{Im}[\ddot{G}(R)] = -\frac{kR \cos kR - \sin kR}{4\pi R^2}, \quad (23)$$

and

$$\nabla R = \frac{x-x'}{R} \hat{\mathbf{x}} + \frac{y-y'}{R} \hat{\mathbf{y}} + \frac{z-z'}{R} \hat{\mathbf{z}}. \quad (24)$$

Similar to (8), function $\nabla G(\mathbf{r}, \mathbf{r}')$ is singular for $R = 0$, and its magnitude is globally nonzero for $R > 0$. Also, note that the singular behavior of $\nabla G(\mathbf{r}, \mathbf{r}')$ comes from $\text{Re}[\ddot{G}(\mathbf{r}, \mathbf{r}')].$

Again, using the pre-splitting strategy [32, 33], the gradient Green's function (20) can be written as two parts, namely

$$\nabla G(\mathbf{r}, \mathbf{r}') = \nabla G_E(\mathbf{r}, \mathbf{r}') + \nabla G_P(\mathbf{r}, \mathbf{r}'), \quad (25)$$

where

$$\nabla G_E(\mathbf{r}, \mathbf{r}') = \begin{cases} \{\text{Re}[\ddot{G}(R)] - \Phi_g(R)\} \nabla R, & R < \delta \\ 0, & R \geq \delta \end{cases}, \quad (26)$$

$$\nabla G_P(\mathbf{r}, \mathbf{r}') = \begin{cases} \{\Phi_g(R) + j \text{Im}[\ddot{G}(R)]\} \nabla R, & R < \delta \\ \nabla G(\mathbf{r}, \mathbf{r}'), & R \geq \delta \end{cases}, \quad (27)$$

with δ being the splitting threshold, and $\Phi_g(R)$ being some auxiliary function.

Here, the following auxiliary function is used to achieve the desired splitting, namely

$$\Phi_g(R) = aR^3 + bR^2 + cR. \quad (28)$$

Note that, different from (16), the above is a three order polynomial without the constant term, and it implies

$$\Phi_g(0) = 0, \quad (29)$$

which clearly means that the function value of $\Phi_g(R)$ is zero at $R = 0$ (where $\mathbf{r} = \mathbf{r}'$).

Given the splitting threshold δ , the coefficients in (28) can be determined by some continuity conditions at $R = \delta$ as follows

$$\begin{cases} \Phi_g(\delta) & = \text{Re}[\ddot{G}(\delta)] \\ d\Phi_g(\delta)/dR & = d\text{Re}[\ddot{G}(\delta)]/dR \\ d^2\Phi_g(\delta)/dR^2 & = d^2\text{Re}[\ddot{G}(\delta)]/dR^2 \end{cases}, \quad (30)$$

or, by solving the following matrix equation

$$\begin{bmatrix} \delta^3 & \delta^2 & \delta \\ 3\delta^2 & 2\delta & 1 \\ 6\delta & 2 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \text{Re}[\ddot{G}(\delta)] \\ d\text{Re}[\ddot{G}(\delta)]/dR \\ d^2\text{Re}[\ddot{G}(\delta)]/dR^2 \end{bmatrix}. \quad (31)$$

The derivative functions of $\text{Re}[\ddot{G}(R)]$ used in the above are given in the appendix. Thus, after $\Phi_g(R)$ is obtained, the splitting representation (25) is completely determined.

Similar as before, (28), (29), and (30) are based on the consideration that, after the splitting, the resultant $\nabla G_P(\mathbf{r}, \mathbf{r}')$ (or, more precisely, each Cartesian component $\text{Re}[\nabla G_P(\mathbf{r}, \mathbf{r}')]_{\sigma}$, $\sigma = \{x, y, z\}$) should be smooth everywhere for $\forall \mathbf{r}$ and $\forall \mathbf{r}'$, so that it can be interpolated accurately without much difficulty.

To make it more clear, consider the representative situation of \mathbf{r}' and \mathbf{r} as before. Then, the corresponding function values of $[\nabla G(\mathbf{r}, \mathbf{r}')]_x$ and $[\nabla G_P(\mathbf{r}, \mathbf{r}')]_x$ are evaluated, as shown in Fig. 4. Here, $\text{Re}[\nabla G_P(\mathbf{r}, \mathbf{r}')]_x$ is smooth at $R = 0$ (where $\mathbf{r} = \mathbf{r}'$) as a result of (29), and smooth at the splitting point $R = \delta$ as a result of (30).

Note that the auxiliary function (28) chosen for the splitting of $\text{Re}[\nabla G(\mathbf{r}, \mathbf{r}')]_{\sigma}$ ($\sigma = \{x, y, z\}$) is different from the auxiliary function (16) chosen for the splitting of $\text{Re}[G(\mathbf{r}, \mathbf{r}')]_{\sigma}$. Such difference is due to the fact that, unlike $\text{Re}[G(\mathbf{r}, \mathbf{r}')]_{\sigma}$, $\text{Re}[\nabla G(\mathbf{r}, \mathbf{r}')]_{\sigma}$ ($\sigma = \{x, y, z\}$) manifest different (i.e., positive or negative) singular behaviors when \mathbf{r} approaches \mathbf{r}' along different directions (e.g., along $+x$ or $-x$ direction), which can be deduced from the definition of (24), or, more intuitively, observed from Fig. 3 and Fig. 4.

With the splitting form of the Green's function (13) and its gradient (25), the impedance matrix \mathbf{Z}^{CFIE} can be correspondingly split into two parts, yielding

$$\mathbf{Z}^{\text{CFIE}} = \mathbf{Z}_E + \mathbf{Z}_P, \quad (32)$$

where \mathbf{Z}_E is related to $G_E(\mathbf{r}, \mathbf{r}')$ and $\nabla G_E(\mathbf{r}, \mathbf{r}')$, while \mathbf{Z}_P is related to $G_P(\mathbf{r}, \mathbf{r}')$ and $\nabla G_P(\mathbf{r}, \mathbf{r}')$. For convenience, \mathbf{Z}_E and \mathbf{Z}_P are termed here as the *evanescent matrix* and the *propagating matrix*, respectively. When the commonly-used Krylov subspace iterative method [35] is employed to solve the matrix equation (4), the matrix-vector product $\mathbf{Z}^{\text{CFIE}} \cdot \mathbf{I}$ is computed. Using the splitting representation (32), this product can be equivalently implemented as

$$\mathbf{Z}^{\text{CFIE}} \cdot \mathbf{I} = \mathbf{Z}_E \cdot \mathbf{I} + \mathbf{Z}_P \cdot \mathbf{I}. \quad (33)$$

As will be shown in the following subsections, by fully exploiting the properties of the splitting form of the Green's function and its gradient, both $\mathbf{Z}_E \cdot \mathbf{I}$ and $\mathbf{Z}_P \cdot \mathbf{I}$ can be evaluated in an efficient manner. For clarity, before going into further discussion, basic concepts and key elements of the presented hybrid fast algorithm are illustrated in Fig. 2.

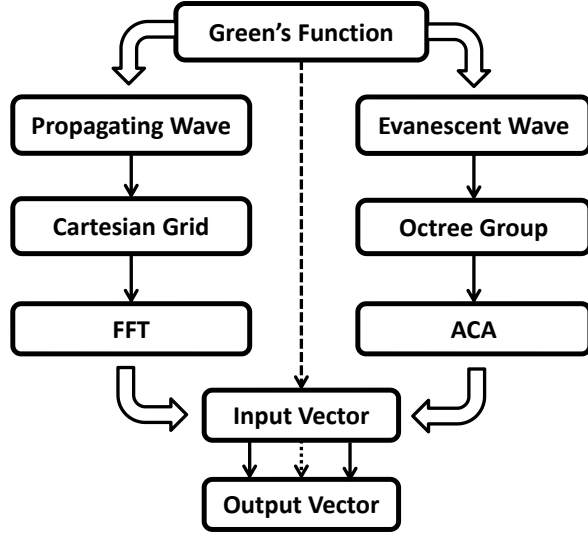


Fig. 2. Systematic diagram of the flow chart and relevant key elements of the presented hybrid fast algorithm. Particularly, the two auxiliary geometric data structures, including the Cartesian grid and the octree grouping, can be constructed independently.

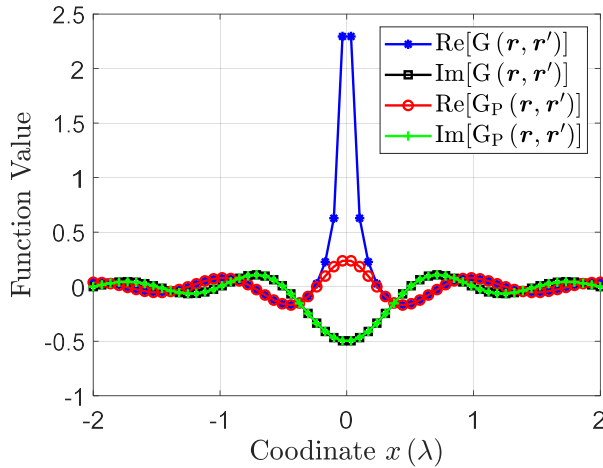


Fig. 3. The function values of $G(\mathbf{r}, \mathbf{r}')$ and $G_P(\mathbf{r}, \mathbf{r}')$. The source point \mathbf{r}' is fixed to the coordinate origin, and the field point \mathbf{r} is moving along the x axis.

C. Propagating matrix calculation

Due to the unique properties (smoothness and translation invariance) of $G_P(\mathbf{r}, \mathbf{r}')$ and $\nabla G_P(\mathbf{r}, \mathbf{r}')$, the matrix-vector product $\mathbf{Z}_P \cdot \mathbf{I}$ can be calculated accurately and efficiently, with the aid of proper interpolation and the FFT.

To this end, some auxiliary geometric data structures need to be defined first. Specifically, the object considered is enclosed by a three dimensional (3D) rectan-

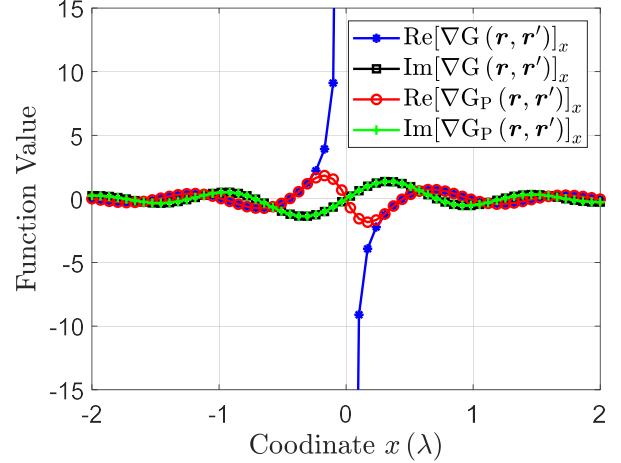


Fig. 4. The function values of $[\nabla G(\mathbf{r}, \mathbf{r}')]_x$ and $[\nabla G_P(\mathbf{r}, \mathbf{r}')]_x$. The source point \mathbf{r}' is fixed to the coordinate origin, and the field point \mathbf{r} is moving along the x axis.

gular bounding box. Then, this bounding box is subdivided regularly along each Cartesian dimension, yielding a cluster of global Cartesian grids [9–14]. The grid spacings along these Cartesian dimensions are here denoted by h_x, h_y, h_z , respectively. The number of grids along these Cartesian dimensions are denoted by N_x, N_y, N_z , respectively. Thus, a total of $N_{\text{Grid}} = N_x \times N_y \times N_z$ Cartesian grids are defined.

Recall that $G_P(\mathbf{r}, \mathbf{r}')$ and $\nabla G_P(\mathbf{r}, \mathbf{r}')$ obtained from the splitting process are smooth everywhere for $\forall \mathbf{r}$ and $\forall \mathbf{r}'$. Thus, they can be well approximated using the following Cartesian grid-based sampling expansions

$$G_P(\mathbf{r}, \mathbf{r}') = \sum_{\mathbf{u} \in B_m} \sum_{\mathbf{v} \in B_n} \beta_{\mathbf{u}}(\mathbf{r}) G_P(\mathbf{u}, \mathbf{v}) \beta_{\mathbf{v}}(\mathbf{r}'), \quad (34)$$

$$\nabla G_P(\mathbf{r}, \mathbf{r}') = \sum_{\mathbf{u} \in B_m} \sum_{\mathbf{v} \in B_n} \beta_{\mathbf{u}}(\mathbf{r}) \nabla G_P(\mathbf{u}, \mathbf{v}) \beta_{\mathbf{v}}(\mathbf{r}'), \quad (35)$$

for $\forall \mathbf{r}$ and $\forall \mathbf{r}'$, where B_m and B_n denote the corresponding local expansion boxes that enclose \mathbf{r} and \mathbf{r}' , respectively. Note that, depending on the relative position between \mathbf{r} and \mathbf{r}' , B_m and B_n can be either well-separated or overlapping, as illustrated in Fig. 5. Moreover, \mathbf{u} and \mathbf{v} denote the coordinates of the Cartesian grids related to B_m and B_n , respectively. Furthermore, $\beta_{\mathbf{u}}(\mathbf{r})$ (or $\beta_{\mathbf{v}}(\mathbf{r}')$) denotes the 3D Lagrange interpolation basis function based on the Cartesian grids of B_m (or B_n), which is defined concretely by

$$\beta_{\mathbf{u}}(\mathbf{r}) = \beta_{p_x}(x) \beta_{p_y}(y) \beta_{p_z}(z), \quad (36)$$

where

$$\beta_{p_\sigma}(\sigma) = \prod_{p_\sigma=0, p_\sigma \neq p'_\sigma}^{M_\sigma} \frac{(\sigma - \sigma_{p'_\sigma})}{(\sigma_{p_\sigma} - \sigma_{p'_\sigma})}, \quad \sigma = \{x, y, z\}, \quad (37)$$

with M_σ being the expansion order and σ_{p_σ} being the coordinates of the Cartesian grids involved. Clearly,

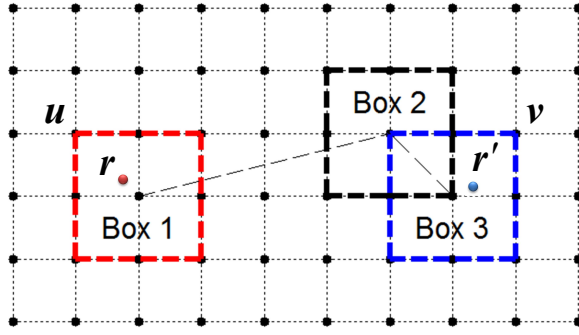


Fig. 5. Computation corresponding to the propagating matrix \mathbf{Z}_P can be accelerated by the FFT using the Global Cartesian grid and the local expansion boxes. Typical examples of the expansion boxes are illustrated using different colors. Here, Box 1 and Box 2 are well-separated, while Box 2 and Box 3 are overlapping.

when $M_x = M_y = M_z = M$, a total of $(M+1)^3$ Cartesian grids are involved in (36).

With the expansions (34) and (35), $\mathbf{Z}_P^{\text{CFIE}} \cdot \mathbf{I}$ can be correspondingly rewritten as

$$\begin{aligned} \mathbf{Z}_P^{\text{CFIE}} \cdot \mathbf{I} &= j\Phi k \eta \bar{\Pi} \cdot [\mathbf{G}_P(\mathbf{u}, \mathbf{v})] \cdot \bar{\Pi}^T \cdot \mathbf{I} \\ &- j\alpha \frac{\eta}{k} \bar{\Pi}_d \cdot [\mathbf{G}_P(\mathbf{u}, \mathbf{v})] \cdot \bar{\Pi}_d^T \cdot \mathbf{I} \\ &+ (1 - \alpha) \eta \bar{\Pi}_g \cdot [\nabla \mathbf{G}_P(\mathbf{u}, \mathbf{v})] \times \bar{\Pi}^T \cdot \mathbf{I}, \end{aligned} \quad (38)$$

where $[\mathbf{G}_P]$ and $[\nabla \mathbf{G}_P]_\sigma$ are triple block Toeplitz matrices of dimension $N_{\text{Grid}} \times N_{\text{Grid}}$. Moreover, $\bar{\Pi}$, $\bar{\Pi}_d$, and $\bar{\Pi}_g$ are sparse matrices of dimension $N \times N_{\text{Grid}}$, defined concretely by

$$\bar{\Pi} = \int_S \begin{bmatrix} \mathbf{f}_1(\mathbf{r}) \\ \mathbf{f}_2(\mathbf{r}) \\ \vdots \\ \mathbf{f}_N(\mathbf{r}) \end{bmatrix} [\beta_1(\mathbf{r}), \dots, \beta_{N_{\text{Grid}}}(\mathbf{r})] ds, \quad (39)$$

$$\bar{\Pi}_d = \int_S \begin{bmatrix} \nabla \cdot \mathbf{f}_1(\mathbf{r}) \\ \nabla \cdot \mathbf{f}_2(\mathbf{r}) \\ \vdots \\ \nabla \cdot \mathbf{f}_N(\mathbf{r}) \end{bmatrix} [\beta_1(\mathbf{r}), \dots, \beta_{N_{\text{Grid}}}(\mathbf{r})] ds, \quad (40)$$

$$\bar{\Pi}_g = \int_S \begin{bmatrix} \hat{\mathbf{n}} \times \mathbf{f}_1(\mathbf{r}) \\ \hat{\mathbf{n}} \times \mathbf{f}_2(\mathbf{r}) \\ \vdots \\ \hat{\mathbf{n}} \times \mathbf{f}_N(\mathbf{r}) \end{bmatrix} [\beta_1(\mathbf{r}), \dots, \beta_{N_{\text{Grid}}}(\mathbf{r})] ds. \quad (41)$$

Exploiting the block Toeplitz structure of $[\mathbf{G}_P]$ and $[\nabla \mathbf{G}_P]_\sigma$, the matrix-vector product $\mathbf{Z}_P \cdot \mathbf{I}$ can then be eval-

uated efficiently by the FFT as follows

$$\begin{aligned} \mathbf{Z}_P^{\text{CFIE}} \cdot \mathbf{I} &= j\alpha k \eta \bar{\Pi} \cdot \mathcal{F}^{-1} \{ \mathcal{F} \{ [\mathbf{G}_P] \} \cdot \mathcal{F} \{ \bar{\Pi}^T \cdot \mathbf{I} \} \} \\ &- j\alpha \frac{\eta}{k} \bar{\Pi}_d \cdot \mathcal{F}^{-1} \{ \mathcal{F} \{ [\mathbf{G}_P] \} \cdot \mathcal{F} \{ \bar{\Pi}_d^T \cdot \mathbf{I} \} \} \\ &+ (1 - \alpha) \eta \bar{\Pi}_g \cdot \mathcal{F}^{-1} \{ \mathcal{F} \{ [\nabla \mathbf{G}_P] \} \times \mathcal{F} \{ \bar{\Pi}^T \cdot \mathbf{I} \} \}, \end{aligned} \quad (42)$$

where \mathcal{F} and \mathcal{F}^{-1} denote the forward and inverse FFT.

Notably, in the conventional pre-correction-based framework [9–13], the original Green's function $G(\mathbf{r}, \mathbf{r}')$ is approximated with the grid-based sampling expansion. Differently, for the pre-splitting-based framework [32, 33] adopted herein, the function $G_P(\mathbf{r}, \mathbf{r}')$ is instead approximated with the grid-based sampling expansion, as in (34). Since $G_P(\mathbf{r}, \mathbf{r}')$ is smooth everywhere without singularity, it is possible to approximate $G_P(\mathbf{r}, \mathbf{r}')$ with sufficiently good accuracy for arbitrary \mathbf{r} and \mathbf{r}' by proper interpolation. Consequently, unlike [9–13, 31], the Cartesian grid-based near-field corrections are not necessary here.

D. Evanescent matrix calculation

Referring to (14) and (26), we know that \mathbf{Z}_E , dominated by $G_E(\mathbf{r}, \mathbf{r}')$ and $\nabla G_E(\mathbf{r}, \mathbf{r}')$, has nonzero elements only when $R < \delta$. Therefore, with quasi-uniform meshes, \mathbf{Z}_E is a typical sparse matrix with few nonzero elements, and thus it can be stored without much effort. However, in multiscale problems, dense mesh always occurs. Thus, near-field interactions within $R < \delta$ increase substantially. In this case, the nonzero elements of \mathbf{Z}_E will increase dramatically.

Fortunately, in the pre-splitting-based framework adopted, this difficulty can be trivially addressed. In a nut shell, we first construct a spatial grouping only based on the geometric position of the basis functions, and then apply the matrix compression methods based on such a grouping to achieve memory reductions of \mathbf{Z}_E .

Here, a standard adaptive octree is utilized for accomplishing the spatial grouping, and the basic procedures for its construction are outlined as follows.

1. A root cube enclosing the whole object is constructed.
2. The nonempty cubes are subdivided into eight sub-cubes recursively until the number of basis functions within each cube is smaller than a prescribed constant.
3. An octree is built when the process finishes.

Accordingly, several concepts related to the octree are then defined as follows.

- Those cubes that do not need to be further subdivided are called *leaf cubes*. Here, the leaf cubes are denoted by C_s , with s being the index of some leaf cube. It should be noted that not all of the leaf cubes

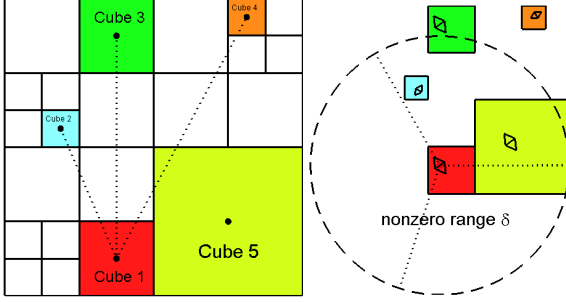


Fig. 6. The storage of \mathbf{Z}_E is effectively compressed by using the spatial grouping octree and the matrix compression method. Illustration of the octree grouping is shown in the left subfigure, where some leaf cubes are highlighted using colors. Here, Cube 5 is neighbor cube of Cube 1, while Cube 2, Cube 3, Cube 4 are all well-separated cubes of Cube 1. Besides, the nonzero range of $G_E(\mathbf{r}, \mathbf{r}')$, characterized by $R < \delta$, is illustrated in the right subfigure. Note that the establishment of the grouping and the application of the compression can be realized independent of the Cartesian grids and the FFT.

are located at the same octree level. Typical examples of the leaf cubes are shown in Fig. 6 and are highlighted with colors.

- Two leaf cubes (e.g., C_s and C_t) are called *neighbor cubes* if they share at least one vertex, such as Cube 1 and Cube 5; otherwise, they are called *well-separated cubes*, such as Cube 1 and Cube 3. For convenience, we use $C_s \cap C_t \neq \emptyset$ to denote that two cubes are neighbor cubes, and use $C_s \cap C_t = \emptyset$ to denote that two cubes are well-separated cubes.

Note that, in [31], the conventional octree grouping cannot be directly applied, and a more involved grouping scheme is developed, in which the cubes for grouping the basis functions are induced from the Cartesian grids. In contrast, here the basis function grouping does not have to be dependent on the Cartesian grids, and can be implemented independently and conventionally. In other words, the commonly-used adaptive octree can be directly employed to achieve the required groupings, and the treatments of the required auxiliary geometric data structures herein are thus simple and straightforward.

On the basis of the spatial groupings above, \mathbf{Z}_E can then be rewritten as

$$\mathbf{Z}_E = \{\mathbf{Z}_E^{st}\}_{C_s \cap C_t \neq \emptyset} + \{\mathbf{Z}_E^{st}\}_{C_s \cap C_t = \emptyset}, \quad (43)$$

where

- $\{\mathbf{Z}_E^{st}\}_{C_s \cap C_t \neq \emptyset}$ denote those matrix subblocks corresponding to the interactions between neighbor cubes. They are directly computed and stored.

- $\{\mathbf{Z}_E^{st}\}_{C_s \cap C_t = \emptyset}$ denote those matrix subblocks corresponding to the interactions between well-separated cubes. These matrix subblocks can be low-rank. They can be compressed and stored in a compact form.

Here, the ACA [36–38] is employed for accomplishing the matrix compression. Detailed procedures of the ACA can be found in, for example, [37, 38]. Correspondingly, $\{\mathbf{Z}_E^{st}\}_{C_s \cap C_t = \emptyset}$ can be then factorized as

$$\{\mathbf{Z}_E^{st}\}_{C_s \cap C_t = \emptyset} = \{\mathbf{U}_E^s \cdot \mathbf{V}_E^t\}_{C_s \cap C_t = \emptyset}. \quad (44)$$

Thus, the matrix-vector product $\mathbf{Z}_E \cdot \mathbf{I}$ can be calculated as

$$\begin{aligned} \mathbf{Z}_E \cdot \mathbf{I} &= \{\mathbf{Z}_E^{st}\}_{C_s \cap C_t \neq \emptyset} \cdot \mathbf{I} + \{\mathbf{Z}_E^{st}\}_{C_s \cap C_t = \emptyset} \cdot \mathbf{I} \\ &= \{\mathbf{Z}_E^s \cdot \mathbf{I}^t\}_{C_s \cap C_t \neq \emptyset} + \{\mathbf{U}_E^s \cdot (\mathbf{V}_E^t \cdot \mathbf{I}^t)\}_{C_s \cap C_t = \emptyset}. \end{aligned} \quad (45)$$

Some implementation details for the compression of $\{\mathbf{Z}_E^{st}\}_{C_s \cap C_t = \emptyset}$ are further clarified as follows. In particular, as shown in Fig. 6, due to the localized nature of \mathbf{Z}_E decided by $R < \delta$, for a basis function ϕ in Cube 1, all the basis functions in Cube 2 have full interactions with ϕ , while all the basis functions in Cube 4 have zero interactions with ϕ . Meanwhile, only parts of the basis functions in Cube 3 and Cube 5 have nonzero interactions with ϕ . Hence, for two well separated groups with index sets s and t , only the submatrices representing nonzero interactions is to be compressed. Thus, the matrix \mathbf{Z}_E^{st} which represents the interactions between the two groups is in fact described more precisely by

$$\mathbf{Z}_E^{st} = \mathbf{P}^{ss} \begin{bmatrix} \mathbf{Z}_E^{s't'} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \mathbf{Q}^{tt}, \quad (46)$$

with subsets $s' \subset s$ and $t' \subset t$. Here, \mathbf{P}^{ss} and \mathbf{Q}^{tt} are permutation matrices, $\mathbf{Z}_E^{s't'}$ denotes the matrix subblock corresponding to nonzero interactions. Note that, depending on the relative position between the two cubes considered, there can be many zero interactions in \mathbf{Z}_E^{st} , due to the sparsity pattern of \mathbf{Z}_E . Therefore, the matrix compression is in fact applied to $\mathbf{Z}_E^{s't'}$ rather than \mathbf{Z}_E^{st} , and we have

$$\mathbf{Z}_E^{s't'} = \mathbf{U}_E \mathbf{V}_E, \quad (47)$$

where $|s'|$ and $|t'|$ denote the sizes of index sets s' and t' , respectively. Meanwhile, the dimension of the matrix \mathbf{U}_E is $|s'| \times \Gamma$, and the dimension of the matrix \mathbf{V}_E is $\Gamma \times |t'|$. Here, Γ denotes the compression rank of $\mathbf{Z}_E^{s't'}$ for a given tolerance. Ultimately, the compressed matrices \mathbf{U}_E and \mathbf{V}_E in (47) are stored, and are applied to the surface current vector \mathbf{I} in an efficient manner. For many practical problems that involve multiscale discretization, the matrix block $\mathbf{Z}_E^{s't'}$ can be effectively compressed, and the memory consumption for \mathbf{U}_E and \mathbf{V}_E are greatly reduced compared to that for \mathbf{Z}_E^{st} .

III. NUMERICAL RESULTS

In this section, the computational performance of the proposed hybrid algorithm is demonstrated through sev-

eral numerical examples. The conjugate gradient method (CG) [35] is used as the iterative solver, and the tolerance is set to 1.0E-3. The diagonal preconditioner is used to improve the convergence. Unless otherwise stated, the grid spacings (i.e., h_x, h_y, h_z) for the FFT acceleration are by default set to the common value $h = 0.1 \lambda$, and the expansion orders (i.e., M_x, M_y, M_z) are by default set to the common value $M = 3$. Furthermore, the well-known FFTW [34] is used to perform the FFT. For clarity, in the following, the traditional pre-splitting-FFT-based algorithm which employs the uncompressed \mathbf{Z}_E will be termed as the PSG-FFT algorithm, and the proposed hybrid fast algorithm (enhanced with the ACA) will be referred to as the PSG-FFT-ACA algorithm.

A. Sphere

A PEC sphere of radius 1.0λ is first considered. The surface of the sphere is discretized with multiscale triangular meshes, resulting in 34,764 RWG functions. In particular, as shown in Fig. 7, about a quarter of the surface is discretized with dense meshes of edge lengths about 0.02λ , and the rest is discretized with regular meshes of edge lengths about 0.1λ .

For this example, the standard MoM (without using any acceleration technique), the traditional PSG-FFT algorithm, and the proposed PSG-FFT-ACA algorithm are adopted as the solution methods. After solving the corresponding matrix equations as formulated in (4), the surface current coefficients (i.e., the solutions of equation (4)) \mathbf{I}^{MoM} , $\mathbf{I}^{\text{PSG-FFT}}$, and $\mathbf{I}^{\text{PSG-FFT-ACA}}$ are then obtained, respectively. Here, it should be noticed that both $\mathbf{I}^{\text{PSG-FFT}}$ and $\mathbf{I}^{\text{PSG-FFT-ACA}}$ depend on the choice of the pre-splitting threshold δ . However, when the standard MoM is used, δ is not involved, and \mathbf{I}^{MoM} is there-

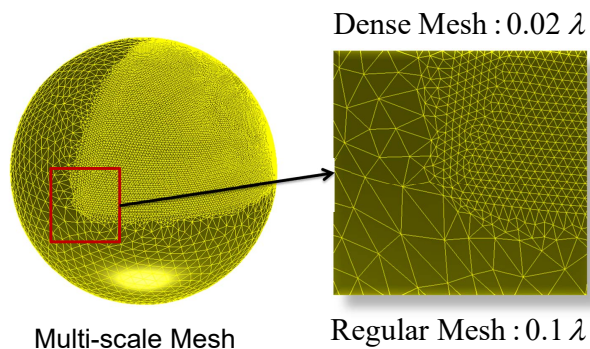


Fig. 7. Mesh configurations of the PEC sphere. The radius of the sphere is 1.0λ . Here, about a quarter of the spherical surface is discretized with a relatively dense mesh of 0.02λ , and the rest of the surface is discretized with a regular mesh of 0.1λ . For clarity, the enlarged view of the multiscale mesh is also visualized in the above.

fore independent of δ . Thus, the solution \mathbf{I}^{MoM} can be used as a proper reference, to evaluate the accuracy of $\mathbf{I}^{\text{PSG-FFT}}$ and $\mathbf{I}^{\text{PSG-FFT-ACA}}$ for varying δ . The relative errors of the surface current coefficients are defined as follows

$$\mathcal{E} = \frac{\|\mathbf{I}^{\text{Test}} - \mathbf{I}^{\text{Reference}}\|_2}{\|\mathbf{I}^{\text{Reference}}\|_2}, \quad (48)$$

where \mathbf{I}^{Test} denotes the surface current coefficients to be tested, $\mathbf{I}^{\text{Reference}}$ denotes some specified surface current coefficients to be used as the reference, and $\|\cdot\|_2$ denotes the 2-norm of vector. Based on (48), the relative errors of both the traditional PSG-FFT algorithm and the proposed PSG-FFT-ACA algorithm for several different δ are calculated and shown in Fig. 8. We can see that both algorithms can achieve accurate results and show similar accuracy level. In particular, within the range of the δ between 0.15λ and 0.65λ , the errors of both algorithms decrease with increasing δ . Furthermore, the accuracies of both algorithms are not strongly sensitive with respect to the δ considered.

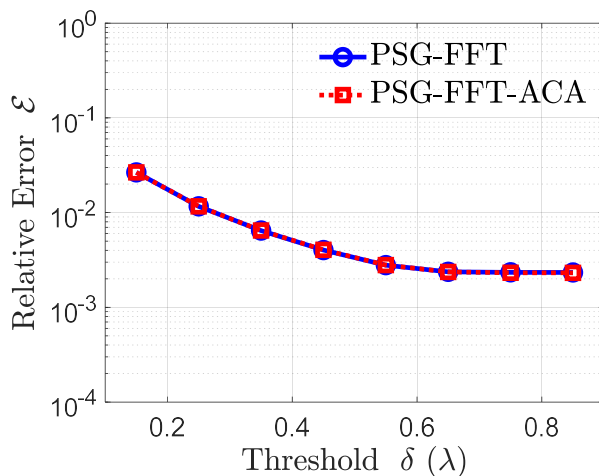


Fig. 8. Relative errors \mathcal{E} of the surface currents coefficients \mathbf{I} with respect to different splitting threshold δ . The relative errors of both the traditional PSG-FFT algorithm and the proposed PSG-FFT-ACA algorithm are illustrated. The surface currents coefficients obtained using the standard MoM are used as the reference.

In the following, the choice of δ is discussed. In principle, δ can be of any value. However, in practice, a moderate value of δ is suggested, due to the following considerations.

- If δ is chosen be relatively large, the number of nonzero elements of the evanescent matrix \mathbf{Z}_E will then become relatively large. Clearly, in this case, the corresponding computational cost will increase.

In Table 1, the computational costs with respect to δ are recorded. It can be observed that the time and memory consumptions of both the traditional PSG-FFT algorithm and the proposed PSG-FFT-ACA algorithm increase as δ increase. Hence, due to efficiency considerations, δ is not suggested to be too large.

- If δ is chosen to be relatively small, the number of nonzero elements of the evanescent matrix \mathbf{Z}_E will become relatively small, and the computational cost is correspondingly reduced. However, from Fig. 8, it is observed that the accuracy of the solution is slightly reduced when δ becomes smaller. Hence, due to accuracy considerations, δ is not suggested to be too small as well.

Based on the above considerations, the δ between 0.15λ and 0.65λ can be proper choice. In practice, by tuning the splitting threshold δ within this range, we can trade off between accuracy and computational cost.

Besides, the well-studied IE-FFT algorithm [17], being a typical instance of the pre-correction-based methods, is also used to calculate this example. Similar to the configurations of the pre-splitting-based methods employed, the grid spacing h for the IE-FFT is set to 0.1λ and the expansion order M is set to 3. Then, using \mathbf{I}^{MoM} as the reference, the relative error of the obtained surface current coefficients $\mathbf{I}^{\text{IE-FFT}}$ is calculated, which is $2.96\text{E-}3$. Referring to Fig. 8, we can see that such error is close to the error of the PSG-FFT(-ACA) algorithm with δ about 0.45λ . Moreover, for the IE-FFT algorithm, the memory requirement is 3.09718 G and the CPU time required is 1.08727 m. Referring to Table 1, we can see that such cost is slightly lower than that of the PSG-FFT algorithm with δ about 0.45λ . From the above numerical results, it can be known that the computational performance of the IE-FFT algorithm during the

solution stage is slightly better than that of the PSG-FFT algorithm. However, it should be emphasized that such slight advantage of the IE-FFT algorithm is at the price of the additional grid-based numerical correction operations during the setup (precomputation) stage.

B. Monopole antenna on top of large platform

In the following, a more realistic example is considered, i.e., a monopole antenna on top of a large platform, as shown in Fig. 9. Clearly, this is a typical multiscale structure and often encountered in many practical scenarios. The width and height of the platform are 4.0λ and 0.3λ , respectively. Moreover, the width and height of the monopole antenna are 0.06λ and 0.6λ , respectively. The object surface is discretized with the multiscale triangular meshes. Specifically, the monopole antenna is discretized with meshes of edge length about 0.01λ , while the large platform is discretized with meshes of edge length about 0.1λ . With such mesh, a total of 24,807 RWG functions are generated.

The standard MoM and the proposed PSG-FFT-ACA algorithm are employed to calculate this example. Here, the pre-splitting threshold δ is chosen to be 0.35λ . Using \mathbf{I}^{MoM} as the reference, the relative errors of $\mathbf{I}^{\text{PSG-FFT-ACA}}$ are calculated for different grid spacings h and different expansion orders M . For clarity, the relative errors are illustrated in Fig. 10. The corresponding time and memory consumptions are recorded in Table 2. It can be clearly seen that, for decreasing grid spacing h and increasing expansion order M , the relative error decreases and the corresponding computational cost increases. In other words, by tuning h and M , we can trade off between accuracy and computational cost.

Table 1: Time and memory for the sphere model with different splitting threshold δ

$\delta(\lambda)$	PSG-FFT		PSG-FFT-ACA	
	Memory (GB)	CPU Time (m)	Memory (GB)	CPU Time (m)
$0.15(\lambda)$	0.9924	0.6601	0.9917	0.7087
$0.25(\lambda)$	1.9707	0.8078	1.6581	0.8167
$0.35(\lambda)$	3.1454	1.1445	2.2623	0.8679
$0.45(\lambda)$	4.6662	1.2814	2.9167	0.9177
$0.55(\lambda)$	6.3642	2.0607	3.5851	0.9812
$0.65(\lambda)$	8.3831	1.8004	4.2083	1.0390
$0.75(\lambda)$	10.514	2.1802	4.3177	1.0939
$0.85(\lambda)$	11.528	2.4548	5.1668	1.2197

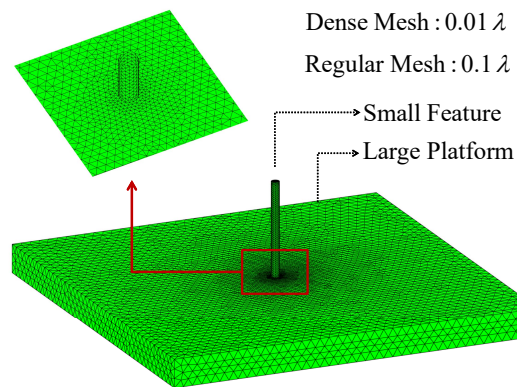


Fig. 9. Mesh configurations of a small monopole antenna on top of a large platform. A multiscale mesh is used to discretize this object. The monopole antenna is discretized using dense mesh of about 0.01λ , while the large platform is discretized with regular mesh of about 0.1λ .

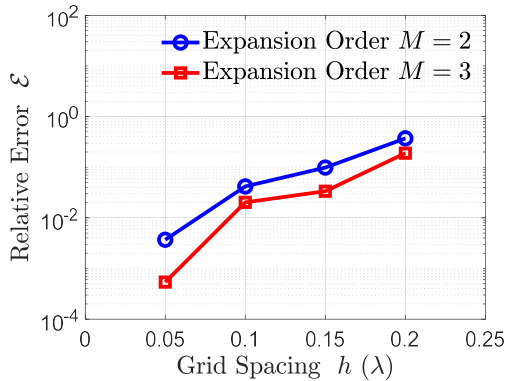


Fig. 10. Relative errors ε of the surface currents coefficients with respect to different grid spacings h and expansion orders M . The relative errors of the proposed PSG-FFT-ACA algorithm are illustrated. The surface currents coefficients obtained using the standard MoM are used as the reference.

Table 2: Time and memory for the monopole-platform model with different grid spacing h and expansion order M

$h(\lambda)$	$M = 2$		$M = 3$	
	Memory (GB)	CPU Time (m)	Memory (GB)	CPU Time (m)
0.05(λ)	1.0074	0.7863	1.1094	0.8256
0.10(λ)	0.8643	0.2280	0.9662	0.2792
0.15(λ)	0.8311	0.1554	0.9304	0.2106
0.20(λ)	0.8243	0.1155	0.9261	0.1801

C. Cone-shape object

To further study the effectiveness of the matrix compression enhancement, a PEC cone model is then considered, as shown in Fig. 11. The height of the cone is 3.0λ . The radii of the top and the bottom faces are 0.03λ and 0.6λ , respectively. The surface is discretized with a gradually-varied meshes. In particular, three mesh configurations are considered. For the three cases, the edge length for the regular part of the mesh is fixed to 0.1λ , while the edge lengths for the dense part of the mesh are set to 0.01λ , 0.005λ , and 0.0025λ , respectively. For clarity, the mesh settings considered are illustrated in Fig. 11.

For each of these mesh configurations, we solve the corresponding equations first using the PSG-FFT algorithm which employs the uncompressed \mathbf{Z}_E . Then, we recalculate these problems using the PSG-FFT-ACA algorithm which compresses \mathbf{Z}_E with the ACA. Here, the

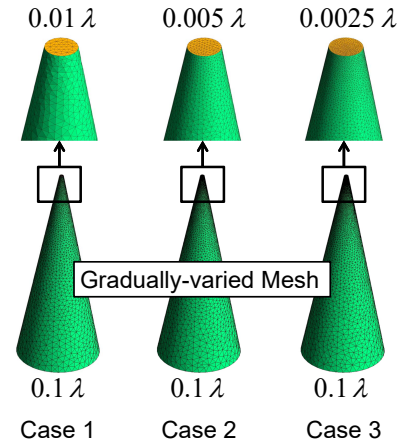


Fig. 11. Mesh configurations of the PEC cone. A gradually-varied mesh is used to discretize the cone-shape object. The edge length for the regular part of the mesh is 0.1λ . The edge lengths for the dense part of the mesh are 0.01λ , 0.005λ , and 0.0025λ , respectively. For clarity, the enlarged views are also shown in the above.

splitting threshold δ is chosen to be 0.35λ . Notice that, with such configuration of δ , all the interactions corresponding to \mathbf{Z}_E occur within the subwavelength regime. Thus, the ACA compression can be applied effectively [21, 37]. In Table 3, 4, 5, we tabulate the time and memory consumptions of both the PSG-FFT algorithm and the PSG-FFT-ACA algorithm for the three mesh cases considered. It can be seen that, for all the cases, the PSG-FFT-ACA algorithm shows improved computational performances relative to the PSG-FFT algorithm.

Table 3: Time and memory for the cone model of case 1

Algorithm	Memory (GB)		CPU Time	
	\mathbf{Z}_E	Total	$\mathbf{Z}_E \cdot \mathbf{I}$ (s)	Total (m)
PSG-FFT	0.3313	0.6286	12.128	0.6955
PSG-FFT-ACA	0.2418	0.4738	8.3050	0.6194

Table 4: Time and memory for the cone model of case 2

Algorithm	Memory (GB)		CPU Time	
	\mathbf{Z}_E	Total	$\mathbf{Z}_E \cdot \mathbf{I}$ (s)	Total (m)
PSG-FFT	0.9611	1.6176	69.751	2.3978
PSG-FFT-ACA	0.4935	0.8778	35.127	1.8475

Table 5: Time and memory for the cone model of case 3

Algorithm	Memory (GB)		CPU Time	
	Z_E	Total	$Z_E \cdot I$ (m)	Total (m)
PSG-FFT	2.6465	4.2615	181.47	4.4874
PSG-FFT-ACA	1.1579	2.0570	78.678	2.8009

Further inspections show that the degree of improvements becomes more prominent when the dense mesh is finer. To make this more clear, we calculate the compression ratios of the evanescent matrix Z_E and use this as an indicator to measure the degree of improvements of the matrix compression enhancement. In Fig. 12, the compression ratios with respect to different mesh configurations are illustrated. We can clearly see that the compression ratio becomes higher when the dense mesh becomes finer. That is, the effectiveness of the proposed hybrid algorithm (with matrix compression enhancement) becomes more prominent when the ratio between the regular mesh size and the dense mesh size is larger.

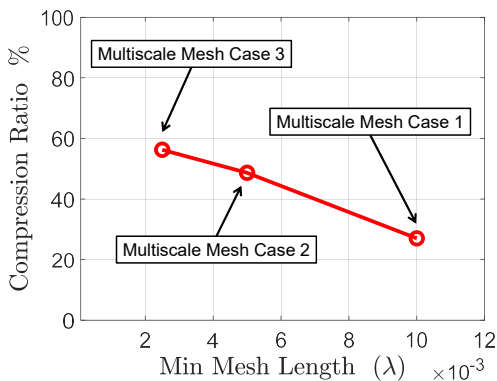


Fig. 12. Compression ratios of the cone with different mesh configurations.

D. Aircraft model

A relatively large PEC aircraft model with multi-scale meshes is finally considered, as shown in Fig. 13. In this model, both the length and the wingspan of the aircraft are about 16λ . The surface is discretized with triangular meshes, resulting in 120,072 RWG functions. Here, a part of the aircraft surface on the wings is discretized with dense meshes of edge lengths about 0.02λ , and the rest of the aircraft surface is discretized with regular meshes of edge lengths about 0.1λ .

The bistatic RCS curves obtained by the PSG-FFT algorithm and the PSG-FFT-ACA algorithm are illus-

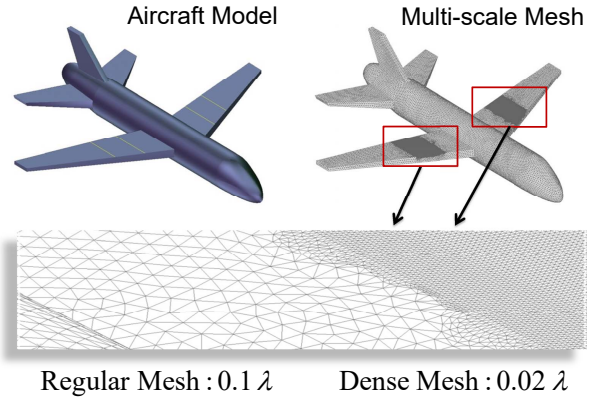


Fig. 13. Geometry model and corresponding multiscale mesh of the PEC aircraft. Here, a part of the aircraft surface on the wings are discretized with a relatively dense mesh of 0.02λ , while the rest of the aircraft surface is discretized with a regular mesh of 0.1λ . The enlarged view of the multiscale mesh is also highlighted in the above.

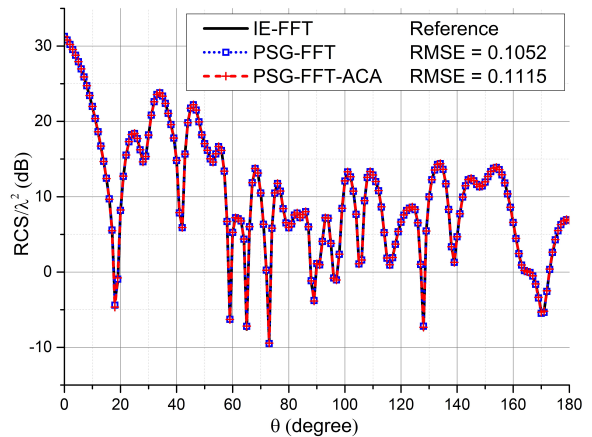


Fig. 14. Bistatic RCS curves of the PEC aircraft model. The result obtained by the IE-FFT is used as the reference. The RMSE of the PSG-FFT and the PSG-FFT-ACA are calculated.

trated in Fig. 14. The pre-splitting threshold δ is here chosen to be 0.35λ . The bistatic RCS curve obtained by the well-tested IE-FFT algorithm [17] is also shown in the figure and used as the reference to calculate the root mean square error (RMSE). We can see that these curves agree well with each other.

The time and memory consumptions of both the PSG-FFT algorithm and the PSG-FFT-ACA algorithm are tabulated in Table 6. From the table, we can see that the memory requirement of the evanescent matrix Z_E in the PSG-FFT-ACA algorithm is reduced by about 34.1218%, compared with that of the PSG-FFT

Table 6: Time and memory for the aircraft model

Algorithm	Memory (GB)		CPU Time	
	Z_E	Total	$Z_E \cdot I$ (s)	Total (m)
PSG-FFT	3.5558	7.7408	76.910	23.580
PSG-FFT- ACA	2.3425	5.4033	48.053	21.431

algorithm. Meanwhile, the time cost for performing $Z_E \cdot I$ in the whole solution process is reduced by about 37.52%. Thus, the time and memory cost related to the evanescent matrix Z_E can be greatly reduced, as a result of the matrix compression enhancement.

Nevertheless, the total cost of the overall algorithm are determined not only by the cost related to the evanescent matrix Z_E , but also determined by the cost related to the propagating matrix Z_P . For this relatively large object, the dense mesh region only occupies a relatively small portion of the overall object surface, and the ratio between the edge length (0.1λ) of the regular mesh and the edge length (0.02λ) of the dense mesh is here not extremely large. Thus, the time cost for performing $Z_P \cdot I$ still dominates the time cost for performing $Z \cdot I$, and the reduction of the time cost for performing $Z_E \cdot I$ therefore only has a minor influence on the total time cost for performing $Z \cdot I$. As a result, although the total time can be reduced due to the reduced cost related to Z_E , the reduction of the total time does not look very striking for this example.

IV. CONCLUSIONS

In this paper, a hybrid fast algorithm has been established for efficiently analyzing multiscale problems. In this algorithm, the Green's function is a priori split into a singular but spatially localized part, and a smooth, oscillatory but bandlimited part. Then, the fundamental blocks for acceleration, i.e., the FFT and the matrix compression method (ACA), are applied to these two parts, respectively. Compared with the traditional methods which only employ the FFT for acceleration, the proposed hybrid algorithm can maintain low memory consumption in multiscale cases without compromising time cost, thus manifesting itself as a favorable multiscale extension of the traditional pure FFT-based methods. Furthermore, the required auxiliary geometric data structures herein can be constructed independent of each other, thus permitting the two kinds of acceleration methods to be connected seamlessly in a concise and elegant manner.

On the basis of the present work, several future directions may be explored. First, except for the Lagrange interpolation used in this work, other more

accurate interpolation schemes (such as the Gaussian interpolation [15] or the fitting techniques [12]) can be employed to further improve the computational performance of the overall algorithm. Moreover, although only the PEC problems are considered here, extensions to the dielectric problems (especially inhomogeneous cases [16]) are clearly feasible. Furthermore, effective parallelization (especially on heterogeneous architectures [13]) of the proposed hybrid algorithm can be interesting subject of future research.

APPENDIX

The derivative functions of $\text{Re}[G(R)]$, which are used in (18) and (19), are given below

$$\frac{d}{dR} \text{Re}[G(R)] = -\frac{\cos(kR) + kR \sin(kR)}{4\pi R^2}, \quad (49)$$

$$\frac{d^2}{dR^2} \text{Re}[G(R)] = -\frac{(k^2 R^2 - 2) \cos(kR) - 2kR \sin(kR)}{4\pi R^3}. \quad (50)$$

The derivative functions of $\text{Re}[\ddot{G}(R)]$, which are used in (30) and (31), are given below

$$\frac{d}{dR} \text{Re}[\ddot{G}(R)] = -\frac{(k^2 R^2 - 2) \cos(kR) - 2kR \sin(kR)}{4\pi R^3}, \quad (51)$$

$$\frac{d^2}{dR^2} \text{Re}[\ddot{G}(R)] = -\frac{3(k^2 R^2 - 2) \cos(kR) - kR(k^2 R^2 - 6) \sin(kR)}{4\pi R^4}. \quad (52)$$

ACKNOWLEDGMENT

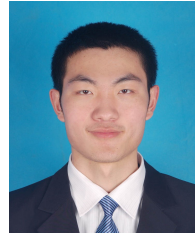
This work was supported in part by the Fundamental Research Funds for the Central Universities (No. 2242022k30003), in part by the National Key Research and Development Project (No. 2019YFB1803202), and in part by the National Science Foundation of China (Nos. 62293492, 62131008, 61975177, U20A20164).

REFERENCES

- [1] W. C. Chew and L. J. Jiang, "Overview of large-scale computing: the past, the present, and the future," *Proceedings of the IEEE*, vol. 101, no. 2, pp. 227-241, Feb. 2013.
- [2] M.-K. Li and W. C. Chew, "Multiscale simulation of complex structures using equivalence principle algorithm with high-order field point sampling scheme," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2389-2397, 2007.
- [3] Z.-G. Qian and W. C. Chew, "Fast full-wave surface integral equation solver for multiscale structure modeling," *IEEE Trans. Antennas Propag.*, vol. 57, no. 11, pp. 3594-3601, Nov. 2009.

- [4] J. Zhu, S. Omar, and D. Jiao, "Solution of the electric field integral equation when it breaks down," *IEEE Trans. Antennas Propag.*, vol. 62, no. 8, pp. 4122-4134, Aug. 2014.
- [5] R. F. Harrington, *Field Computation by Moment Methods*. New York: The MacMillian Co., 1968.
- [6] J. M. Song and W. C. Chew, "Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering," *Microw. Opt. Technol. Lett.*, vol. 10, no. 1, pp. 14-19, Sep. 1995.
- [7] J.-M. Song, C.-C. Lu, and W. C. Chew, "Multi-level fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1488-1493, Oct. 1997.
- [8] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston, MA: Artech House, 2001.
- [9] M. Bleszynski, E. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Sci.*, vol. 31, no. 5, pp. 1225-1251, 1996.
- [10] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures," *IEEE Trans. Computer-Aided Design*, vol. 16, no. 10, pp. 1059-1072, Oct. 1997.
- [11] S. M. Seo and J.-F. Lee, "A fast IE-FFT algorithm for solving PEC scattering problems," *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1476-1479, May 2005.
- [12] J.-Y. Xie, H.-X. Zhou, W. Hong, W.-D. Li, and G. Hua, "A highly accurate FGG-FG-FFT for the combined field integral equation," *IEEE Trans. Antennas Propag.*, vol. 61, no. 9, pp. 4641-4652, Sep. 2013.
- [13] S. Peng and C.-F. Wang, "Precorrected-FFT method on graphics processing units," *IEEE Trans. Antennas Propag.*, vol. 61, no. 4, pp. 2099-2107, Apr. 2013.
- [14] L.-W. Li, Y.-J. Wang, and E.-P. Li, "MPI-based parallelized precorrected FFT algorithm for analyzing scattering by arbitrarily shaped three-dimensional objects," *Electromagn. Waves Appl.*, vol. 17, pp. 1489-1491, Jan. 2003.
- [15] B. Lai, X. An, H.-B. Yuan, Z.-S. Chen, C.-M. Huang, and C.-H. Liang, "A novel Gaussian interpolation formula-based IE-FFT algorithm for solving EMscattering problems," *Microw. Opt. Tech. Lett.*, vol. 51, no. 9, pp. 2233-2236, Sep. 2009.
- [16] X.-C. Nie, L.-W. Li, N. Yuan, T. S. Yeo, and Y.-B. Gan, "Precorrected-FFT solution of the volume integral equation for 3-D inhomogeneous dielectric objects," *IEEE Trans. Antennas Propag.*, vol. 53, no. 1, pp. 313-320, Jan. 2005.
- [17] J.-Y. Xie, H.-X. Zhou, W.-D. Li, and W. Hong, "IE-FFT for the combined field integral equation applied to electrically large objects," *Microw. Opt. Tech. Lett.*, vol. 54, no. 4, pp. 891-896, Apr. 2012.
- [18] S. Peng, C.-F. Wang, and J.-F. Lee, "Analyzing PEC scattering structure using an IE-FFT algorithm," *ACES Journal*, vol. 24, no. 2, Apr. 2009.
- [19] L. J. Jiang and W. C. Chew, "A mixed-form fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 53, no. 12, pp. 4145-4156, Dec. 2005.
- [20] X.-M. Pan, J.-G. Wei, Z. Peng, and X.-Q. Sheng, "A fast algorithm for multiscale electromagnetic problems using interpolative decomposition and multi-level fast multipole algorithm," *Radio Sci.*, vol. 47, no. 1, Feb. 2012.
- [21] L.-F. Ma, Z. Nie, J. Hu, and S.-Q. He, "Combined MLFMA-ACA algorithm application to scattering problems with complex and fine structure," *Asia Pacific Microwave Conference*, pp. 802-805, 2009.
- [22] W.-B. Kong, H.-X. Zhou, K.-L. Zheng, and W. Hong, "Analysis of multiscale problems using the MLFMA with the assistance of the FFT-based method," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 9, pp. 4184-4188, 2015.
- [23] J. S. Zhao and W. C. Chew, "Three dimensional multilevel fast multipole algorithm from static to electrodynamic," *Microw. Opt. Technol. Lett.*, vol. 26, no. 1, pp. 43-48, 2000.
- [24] L. J. Jiang and W. C. Chew, "Low-frequency fast inhomogeneous plane-wave algorithm," *Microw. Opt. Tech. Lett.*, vol. 40, no. 2, pp. 117-122, Jan. 2004.
- [25] T. Xia, L. L. Meng, Q. S. Liu, H. H. Gan, and W. C. Chew, "A low-frequency stable broadband multilevel fast multipole algorithm using plane wave multipole hybridization," *IEEE Trans. Antennas Propag.*, vol. 66, no. 11, pp. 6137-6145, Nov. 2018.
- [26] H. Wallen and J. Sarvas, "Translation procedures for broadband MLFMA," *Prog. Electromagn. Res.*, vol. 55, pp. 47-78, 2005.
- [27] D. Wulf and R. Bunger, "An efficient implementation of the combined wideband MLFMA/LF-FIPWA," *IEEE Trans. Antennas Propag.*, vol. 57, no. 2, pp. 467-474, Feb. 2009.

- [28] D. T. Schobert and T. F. Eibert, "Fast integral equation solution by multilevel Green's function interpolation combined with multilevel fast multipole method," *IEEE Trans. Antennas Propag.*, vol. 60, no. 9, pp. 4458-4463, Sep. 2012.
- [29] I. Bogaert, J. Peeters, and F. Olyslager, "A nondirective plane wave MLFMA stable at low frequencies," *IEEE Trans. Antennas Propag.*, vol. 56, no. 12, pp. 3752-3767, Dec. 2008.
- [30] O. Ergul and B. Karaosmanoglu, "Broadband multilevel fast multipole algorithm based on an approximate diagonalization of the Green's function," *IEEE Trans. Antennas Propag.*, vol. 63, no. 7, pp. 3035-3041, July 2015.
- [31] W.-B. Kong, H.-X. Zhou, K.-L. Zheng, X. Mu, and W. Hong, "FFT-based method with near-matrix compression," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 11, pp. 5975-5983, 2017.
- [32] A. Bepalov, "Cost-effective solution of the boundary integral equations for 3D Maxwell problems," *Russ. J. Numer. Anal. Math. Modelling*, vol. 14, no. 5, pp. 403-428, 1999.
- [33] A. Bepalov, "On the use of a regular grid for implementation of boundary integral methods for wave problems," *Russ. J. Numer. Anal. Math. Modelling*, vol. 15, no. 6, pp. 469-488, 2000.
- [34] M. Frigo and S. Johnson, FFTW Manual [Online]. Available: <http://www.fftw.org/>
- [35] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Philadelphia, PA, USA: SIAM, 1994.
- [36] M. Bebendorf and S. Rjasanow, "Adaptive low-rank approximation of collocation matrices," *Computing*, vol. 70, no. 1, pp. 1-24, 2003.
- [37] K. Zhao, M. N. Vouvakis, and J.-F. Lee, "The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems," *IEEE Trans. Electromag. Compat.*, vol. 47, no. 4, pp. 763-773, Nov. 2005.
- [38] H.-X. Zhou, G.-Y. Zhu, W.-B. Kong, and W. Hong, "An upgraded ACA algorithm in complex field and its statistical analysis," *IEEE Trans. Antennas Propag.*, vol. 65, no. 5, pp. 2734-2739, May 2017.
- [39] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, no. 3, pp. 409-418, May 1982.



Guang-Yu Zhu Guang-Yu Zhu received the B.D. in electrical engineering from Nanjing Agricultural University, Nanjing, China, in 2012, and the Ph.D. degree in radio engineering from Southeast University, Nanjing, China, in 2020, respectively. He is currently a postdoctoral research fellow with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. His research interests are in computational electromagnetics with focus on fast algorithms and domain decomposition methods.



Wei-Dong Li Wei-Dong Li received the M.S. degree in mathematics and the Ph.D. degree in radio engineering from Southeast University, Nanjing, China, in 2003 and 2007, respectively. From January 2008 to January 2009, he was a visiting scholar with the Technische University Darmstadt, Germany. He is currently an associate professor with the State Key Laboratory of Millimeter Waves and the School of Information Science and Engineering, Southeast University. He has authored or coauthored over 40 technical articles. He serves as a reviewer for the *IEEE Transactions on Antennas and Propagation* and *IET Microwave, Antennas and Propagation*. His research interests are in computational EM with focus on integral-equation-based overlapped domain decomposition method, multilevel fast multipole algorithm, fast and accurate inter/extrapolation techniques, and basic functions.



Wei E. I. Sha Wei E.I. Sha received the B.S. and Ph.D. degrees in electronic engineering from Anhui University, Hefei, China, in 2003 and 2008, respectively. From July 2008 to July 2017, he was a postdoctoral research fellow and then a research assistant professor with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong. From March 2018 to March 2019, he worked as a Marie Sklodowska-Curie Individual Fellow with University College London, London, U.K. From October 2017, he joined the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, where

he is currently a tenure-tracked assistant professor. He has authored or coauthored 180 refereed journal articles, 150 conference publications (including five keynote talks and one short course), nine book chapters, and two books. His Google Scholar citation is 8193 with H-index of 45. His research interests include theoretical and computational research in electromagnetics and optics, focusing on the multiphysics and interdisciplinary research. His research involves fundamental and applied aspects in computational and applied electromagnetics, nonlinear and quantum electromagnetics, micro- and nano-optics, optoelectronic device simulation, and multiphysics modeling. Dr. Sha is a member of Optical Society of America (OSA). He served as a reviewer for 60 technical journals and technical program committee member for ten IEEE conferences. He was a recipient of the Applied Computational Electromagnetics Society (ACES) Technical Achievement Award in 2022 and PIERS Young Scientist Award in 2021. In 2015, he was awarded Second Prize of Science and Technology from Anhui Province Government, China. In 2007, he was awarded the Thousand Talents Program for Distinguished Young Scholars of China. He also received six Best Student Paper Prizes and one Young Scientist Award with his students. He also served as an associate editor for *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, *IEEE Open Journal of Antennas and Propagation*, and *IEEE Access*.



Hou-Xing Zhou Hou-Xing Zhou received the M.S. degree in mathematics from Southwest Normal University, Chongqing, China, in 1995, and the Ph.D. degree in radio engineering from Southeast University, Nanjing, China, in 2002. Since 2002, he has been with the State Key Laboratory of Millimeter Waves, Southeast University. He is currently a professor with the School of Information Science and Engineering, Southeast University. He has authored or coauthored 40 journal articles. His main research interests include numerical algorithms in computational EM, including the fast algorithm for spatial domain dyadic Green's functions of stratified media, the multilevel fast multiple algorithm, the FFT-based fast algorithm, the IE-based domain decomposition method, the FEM-BI-based domain decomposition method, the higher order method of moments, and the parallel computation based on GPU/multicore-CPU platform. In addition, the full-wave simulation technology for the electromagnetic field distribution around the base-station is one of his current interests.



Wei Hong Wei Hong received the B.S. degree from the University of Information Engineering, Zhengzhou, China, in 1982, and the M.S. and Ph.D. degrees from Southeast University, Nanjing, China, in 1985 and 1988, respectively, all in radio engineering. Since 1988, he has been with the State Key Laboratory of Millimeter Waves and serves for the director of the lab, since 2003. In 1993, 1995, 1996, 1997, and 1998, he was a short-term visiting scholar with the University of California at Berkeley and at Santa Cruz, respectively. He is currently a professor with the School of Information Science and Engineering, Southeast University. He has been involved in numerical methods for electromagnetic problems, millimeter wave theory and technology, antennas, RF technology for wireless communications, and so on. He has authored and coauthored over 300 technical publications and two books. Dr. Hong was an elected IEEE MTT-S AdCom Member, from 2014 to 2016. He is a fellow of CIE, the vice president of the CIE Microwave Society and Antenna Society, and the chair of the IEEE MTT-S/AP-S/EMC-S Joint Nanjing Chapter. He was twice awarded the National Natural Prizes, thrice awarded the first-class Science and Technology Progress Prizes issued by the Ministry of Education of China and Jiangsu Province Government. He also received the Foundations for China Distinguished Young Investigators and for Innovation Group issued by NSF of China. He served as the associate editor for the *IEEE Transactions on Microwave Theory and Techniques* from 2007 to 2010, and one of the guest editors for the 5G special issue of the *IEEE Transactions on Antennas and Propagation*, in 2017.