# Fast Nested Cross Approximation Algorithm for Solving Large-Scale Electromagnetic Problems

Yu Zhao, *Student Member, IEEE*, Dan Jiao, *Fellow, IEEE*, and Junfa Mao, *Fellow, IEEE*

***Abstract*— A fast nested cross approximation (NCA) algorithm is developed in this paper for solving large-scale electromagnetic problems. Different from the existing NCA, the proposed method does not rely on the projection of the basis functions onto the dummy interpolation points to select pivots of each cluster. Instead, a purely algebraic and kernel-independent algorithm is developed to find the row and column pivots of all clusters in $\mathcal{O}(N \log N)$ complexity for constant-rank cases with controlled accuracy. This algorithm is then further extended to an $\mathcal{O}(N)$ NCA algorithm, which includes a bottom-up tree traversal for finding the local pivots of each cluster, followed by a top-down procedure to take into account the far field of each cluster. The proposed method has a reduced complexity compared to that reported in the mathematical literature. The resultant nested representation constitutes an $\mathcal{H}^2$-matrix representation of the original dense system of equations, whose solution can be obtained in linear complexity in both iterative and direct solvers. The method is also applicable to variable rank cases, but the complexity therein depends on the rank's relationship with $N$. Various numerical experiments have demonstrated the accuracy and computational performance of the proposed algorithms.**

***Index Terms*— $\mathcal{H}^2$-matrix, kernel independent, linear complexity, nested cross approximation (NCA).**

## I. Introduction

THE surface integral equations (SIEs) are among the most compelling formulations [1] for the analysis of electromagnetic problems due to its reduced problem size. Nevertheless, the classical method of moments (MoM) discretization of the SIEs is computationally prohibitive with a complexity of $\mathcal{O}(N^2)$ in CPU time and memory cost, where $N$ is the number of the basis functions used to discretize the integral equation.

Various fast algorithms have been developed to accelerate the solution of SIEs. The fast multipole algorithm (FMA) and the multilevel FMA (MLFMA) accelerate the matrix-vector product (MVP) in the framework of an iterative solution, which is based on an analytical harmonic expansion of the Green's function [2]. Despite the nearly $\mathcal{O}(N)$ complexity for solving electrically small problems [3], the FMA is kernel dependent and becomes rather complicated once the Green's function cannot be analytically factorized, for example, the multilayered Green's function [4]. The fast Fourier transform (FFT) based algorithms, such as the precorrected-FFT [5], adaptive integral method (AIM) [6], and so on, exploit the convolution form of the Green's function to approximate the system matrix by a block Toeplitz structure. The resultant solution time and memory cost scale as $\mathcal{O}(N^{1.5} \log N)$ and $\mathcal{O}(N^{1.5})$, respectively, for 3-D SIEs.

The low-rank approximation based methods exploit the property that the subblocks representing a far-field interaction are rank deficient. The $\mathcal{H}$-matrix is one of the most representative frameworks [7], [8] for developing low-rank methods. The admissible blocks in a system matrix, which correspond to the far-field interactions, are approximated by low-rank matrices with a controlled error. The low-rank approximation can be obtained by several approaches, such as the Taylor expansion, the polynomial interpolation, the adaptive cross approximation (ACA) [9], [10], and the hybrid cross approximation (HCA) [11], [12]. The ACA is one type of purely algebraic methods, which constructs the approximation from a small number of the original matrix elements. The HCA combines the advantages of both the interpolation method and the ACA, providing a reliable low-rank approximation, as well as enhanced efficiency. Although an $\mathcal{H}$-matrix algorithm can be further accelerated [13], from ACA or HCA, one cannot obtain a nested low-rank representation, which hinders a complexity reduction. The Lagrange interpolation based low-rank approximation exploits the nested property among the Lagrange polynomials and leads to an $\mathcal{H}^2$-matrix [14]. The $\mathcal{H}^2$-matrix is a more efficient structure than the $\mathcal{H}$-matrix and it enables a rigorous linear complexity due to the nested form of the cluster basis. The rank-minimized $\mathcal{H}^2$-matrix was also developed to minimize the computational overhead and optimize the cluster bases [15].

The property of being kernel-independent and, meanwhile, nested is desired in developing fast algorithms. A straightforward approach is to convert an $\mathcal{H}$-matrix to an $\mathcal{H}^2$-matrix in a purely algebraic way [16]. However, this method still suffers from the $\mathcal{O}(N \log N)$ complexity during the assembly of the $\mathcal{H}$-matrix, as well as the algebraic conversion. Several kernel-independent algorithms were developed to construct

Manuscript received December 21, 2017; revised April 28, 2018 and October 31, 2018; accepted April 13, 2019. Date of publication June 19, 2019; date of current version August 16, 2019. This work was supported in part by the National Science Foundation of China under Grant 61831016 and in part by Advance Research Foundation under Grant 61402090602. *(Corresponding authors: Yu Zhao; Dan Jiao.)*

Y. Zhao is with the Key Laboratory of Ministry of Education of Design and Electromagnetic Compatibility of High-Speed Electronic Systems, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: zhaoyu775885@gmail.com).

D. Jiao is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: djiao@purdue.edu).

J. Mao is with the Key Laboratory of Ministry of Education of Design and Electromagnetic Compatibility of High-Speed Electronic Systems, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jfmao@sjtu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMTT.2019.2920894

the $\mathcal{H}^2$-matrix directly. The nested equivalent source approximation (NESA) represents the far-field interaction through properly defined equivalent source distributions and multilevel equivalence [17]. The nested skeletonization scheme (NSS) features a similar idea but removes the inverse source procedure by exploiting the samples on proxy surfaces, which are auxiliary surfaces enclosing the clusters of basis functions [18]. The nested cross approximation (NCA) was first developed as a purely algebraic algorithm for static problems [19]. The conventional NCA constructs the cluster bases via only a small number of matrix entries without the requirement of the auxiliary enclosing surfaces. However, the overall procedure for matrix construction is of $\mathcal{O}(N \log N)$ complexity due to a geometry-projection based pivot selection. This scheme first searches the basis functions closest to the predefined Chebyshev nodes and then performs an ACA to obtain the pivots. The overall CPU time cost is of $\mathcal{O}(N \log N)$ despite the memory cost of $\mathcal{O}(N)$.

The conventional NCA inspires us to develop new purely algebraic methods to construct an $\mathcal{H}^2$-matrix representation. In the proposed method, the geometrical projection from the basis functions to the Chebyshev nodes is avoided. Instead, an algebraic approach with well-controlled accuracy is developed, whose complexity is $\mathcal{O}(N \log N)$. This method is then further accelerated to achieve a reduced complexity of $\mathcal{O}(N)$. The $\mathcal{O}(N)$ method is a two-stage approach, consisting of a bottom-up and a top-down tree traversal, to locate the row and column pivots of all clusters in $\mathcal{O}(N)$ complexity. The accuracy and computational complexity of the proposed methods are validated by a variety of numerical examples.

The rest of this paper is organized as follows. In Section II, we introduce the theoretical and computational background of this work. In Section III, we present the proposed algorithms, which start from another interpretation of the NCA in Section III-A, followed by an $\mathcal{O}(N \log N)$ NCA algorithm described in Section III-B, whose complexity is then further reduced to $\mathcal{O}(N)$ in Section III-C. In Section IV, a number of examples are simulated to verify the accuracy and complexity of the proposed algorithms. Finally, our conclusions are drawn in Section V.

## II. Background

In this section, we review the background of this work, including the electric-field integral equation (EFIE), the $\mathcal{H}^2$-matrix framework, and the ACA-based low-rank approximation.

### A. Method of Moments Based Solution of the EFIE

For perfect electric conductor (PEC) objects in free space, the governing EFIE is

$$\hat{\boldsymbol{n}} \times j\omega\mu_0 \int_s \left( \bar{\boldsymbol{I}} - \frac{\nabla\nabla'}{k_0^2} \right) G_0(\boldsymbol{r}, \boldsymbol{r}') \boldsymbol{J}_s(\boldsymbol{r}') \mathrm{d}s' = \hat{\boldsymbol{n}} \times \boldsymbol{E}_i \tag{1}$$

where $G_0(\boldsymbol{r}, \boldsymbol{r}') = ((e^{-jk_0|\boldsymbol{r}-\boldsymbol{r}'|})/(4\pi|\boldsymbol{r}-\boldsymbol{r}'|))$ is the Green's function, $k_0$ is the wavenumber in free space, and $\boldsymbol{E}_i$ is the incident electric field. Using the MoM, we expand the surface current $\boldsymbol{J}_s$ into $N$ Rao–Wilton–Glisson (RWG) basis functions as the following:

$$\boldsymbol{J}_s(\boldsymbol{r}) = \sum_{i=1}^{N} I_i \boldsymbol{f}_i(\boldsymbol{r}) \tag{2}$$

where $I_i$ is the coefficient of the $i$th RWG basis $\boldsymbol{f}_i$. Based on a Galerkin discretization, (1) leads to a matrix equation of

$$\mathbf{Z}\boldsymbol{I} = \boldsymbol{b} \tag{3}$$

where $\boldsymbol{b}$ is the excitation vector and $\mathbf{Z}$ is the system matrix.

### B. On $\mathcal{H}^2$-Matrix

Based on the method of $\mathcal{H}^2$-matrix [7], [14], the basis functions discretizing $\boldsymbol{J}_s$ in (2) are partitioned into multi-level clusters. The resulting cluster tree is denoted by $\mathcal{T}_{\mathcal{I}}$. The interaction between two cluster trees, $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$, is also characterized by a tree structure, which is called a block cluster tree denoted as $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$. Assume that clusters $t$ and $s$ are at the same level of $\mathcal{T}_{\mathcal{I}}$, $X_t$, and $X_s$ are the geometrical supports of the basis functions contained in $t$ and $s$, respectively. The matrix block $\boldsymbol{Z}_{t,s}$ is approximated by a low-rank matrix if the block cluster $b = t \times s$ satisfies the following admissibility condition [20]:

$$\begin{aligned} \max\{\mathrm{diam}(X_t), \mathrm{diam}(X_s)\} &\leqslant \eta \mathrm{dist}(X_t, X_s) \\ \mathrm{diam}(X) &= \sup\{\mathrm{d}(x,y) | \, x, y \in X\} \\ \mathrm{dist}(X,Y) &= \inf\{\mathrm{d}(x,y) | \, x \in X, y \in Y\} \end{aligned} \tag{4}$$

where $\eta$ is a constant coefficient controlling the admissibility condition, $d(\cdot,\cdot)$ is the Euclidean distance between two sets, $\mathrm{diam}(\cdot)$ is the supremum distance between any two points in a cluster, and $\mathrm{dist}(\cdot,\cdot)$ is the infimum distance between any two points in different clusters [20].

For a block cluster $b = t \times s$ satisfying the admissibility condition, the corresponding matrix block is represented in a factorized form of

$$\boldsymbol{Z}_{t,s} = \boldsymbol{U}_t \boldsymbol{S}_{t,s} \boldsymbol{V}_s{}^{\mathrm{H}} \tag{5}$$

where $\boldsymbol{U}_t \in \mathbb{C}^{\#t \times k_1}$ and $\boldsymbol{V}_s \in \mathbb{C}^{\#s \times k_2}$ are nested cluster bases for row cluster $t$ and column cluster $s$, respectively, and $\boldsymbol{S}_{t,s} \in \mathbb{C}^{k_1 \times k_2}$ is the coupling matrix. Only the leaf cluster bases need to be stored. The nonleaf cluster bases can be obtained by aggregating the children cluster bases using transfer matrices as the following:

$$\boldsymbol{U}_t = \begin{bmatrix} \boldsymbol{U}_{t_1} \boldsymbol{T}_{t_1,t} \\ \boldsymbol{U}_{t_2} \boldsymbol{T}_{t_2,t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_{t_1} & \\ & \boldsymbol{U}_{t_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{T}_{t_1,t} \\ \boldsymbol{T}_{t_2,t} \end{bmatrix} \tag{6}$$

where $t = t_1 \cup t_2$, $\boldsymbol{T}_{t_1,t}$ and $\boldsymbol{T}_{t_2,t}$ are transfer matrices associated with the nonleaf cluster $t$, which transfer the cluster bases from the two children ($t_1$ and $t_2$) to the parent ($t$). If the admissibility condition is not satisfied, the matrix block is represented as a full matrix. The factorized form in (5) together with the nested property of (6) constitutes an $\mathcal{H}^2$-matrix representation of the system matrix $\mathbf{Z}$.

### *C. ACA Based Low-Rank Approximation*

ACA is one type of purely algebraic approaches to generate rank-$k$ matrices [9], [10], [21]. Based on ACA, an admissible block $\boldsymbol{Z}_{t,s}$ is approximated by the expansion of a series of rank-1 matrices [21], each of which is the multiplication of a column vector and a row vector as the following:

$$\boldsymbol{Z}_{t,s} \approx \boldsymbol{Z}^*_{t,s} = \boldsymbol{Z}_{t,\sigma}(\boldsymbol{Z}_{\tau,\sigma})^{-1}\boldsymbol{Z}_{\tau,s} = \sum_{i=1}^{k} \boldsymbol{a}_i \boldsymbol{b}_i^T \tag{7}$$

where $\boldsymbol{Z}_{t,s}, \boldsymbol{Z}^*_{t,s} \in \mathbb{C}^{\#t\times\#s}$, $\tau \subseteq t$, and $\sigma \subseteq s$ are the row and column index sets selected during a heuristic procedure and called *pivots*, $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are the vectors generated from each step of the ACA, and $k$ is the rank of $\boldsymbol{Z^*}_{t,s}$. The stopping criterion is controlled by the ACA tolerance $\epsilon_{ACA}$ with the following condition:

$$\|\boldsymbol{a}_k\|_2\|\boldsymbol{b}_k\|_2 \leqslant \epsilon_{\text{ACA}}\|\boldsymbol{Z}^{*(k)}_{t,s}\|_F \tag{8}$$

where $\boldsymbol{Z}^{*(k)}_{t,s}$ is the $k$th update of the approximation of $\boldsymbol{Z}_{t,s}$, the $\mathcal{L}^2$-norm and Frobenius norm are used for vectors and matrices, respectively.

The number of operations to construct the approximation in (7) is of $\mathcal{O}(k^2(\#t + \#s))$, while the storage required is of $\mathcal{O}(k(\#t + \#s))$. Here, # denotes the cardinality of a set.

## III. Proposed Algorithms

In this section, we first present our view of the NCA method. We then propose an $\mathcal{O}(N \log N)$ algorithm to perform the NCA. After that, we show how to further reduce the complexity of NCA to $\mathcal{O}(N)$.

### *A. Another View of NCA*

When the rank of the admissible blocks is a bounded constant, the matrix size of the factors $\boldsymbol{Z}_{t,\sigma}$ and $\boldsymbol{Z}_{\tau,s}$ in (7) depends linearly on the number of rows and columns. This leads to a complexity of $\mathcal{O}(N \log N)$ in an $\mathcal{H}$-matrix-based fast solver [13]. To achieve a lower complexity, the NCA exploits the nestedness among the low-rank blocks to build an $\mathcal{H}^2$-matrix by [19]

$$\boldsymbol{Z}_{t,s} \approx \boldsymbol{Z}_{t,\sigma_t}(\boldsymbol{Z}_{\tau_t,\sigma_t})^{-1}\boldsymbol{Z}_{\tau_t,\sigma_s}(\boldsymbol{Z}_{\tau_s,\sigma_s})^{-1}\boldsymbol{Z}_{\tau_s,s} \tag{9}$$

where $\tau_t \subseteq t$ and $\sigma_t \subseteq \mathcal{F}(t)$ are the pivots of $t$ and $\mathcal{F}(t)$, respectively, $\sigma_s \subseteq s$ and $\tau_s \subseteq \mathcal{F}(s)$ are the pivots of $s$ and $\mathcal{F}(s)$. Here, $\mathcal{F}(x)$ denotes the far field of cluster $x$, which represents the union of all the clusters that form admissible blocks with $x$ at the same tree level of $x$, and the clusters that form admissible blocks with $x$'s ancestors at higher tree levels. We call the union of the clusters as the local far field if the clusters form admissible blocks with $x$ at $x$'s tree level. Mathematically, $\mathcal{F}(x)$ can be written as

$$\mathcal{F}(x) = \cup\{y \in \mathcal{T}_I | \exists x' \supseteq x, x' \times y \ or \ y \times x' \ \in \mathcal{P}_{\text{adm}}\} \tag{10}$$

where $\mathcal{P}_{\text{adm}}$ represents the union of all admissible blocks.

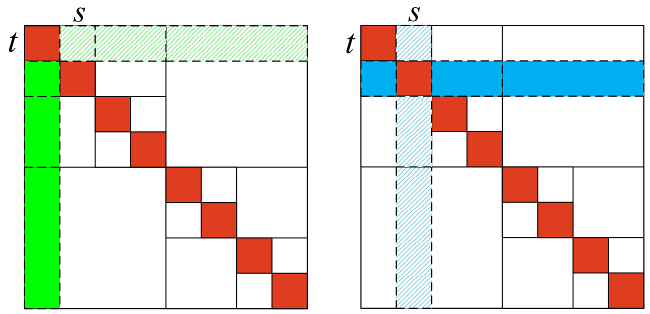
The validity of the nested approximation (9) has been proven in [19] by analyzing numerical errors. Here, we provide an intuitive interpretation of (9). Assume that $t$ and $s$ are two clusters at the same tree level of $\mathcal{T}_{\mathcal{I}}$, and $t \times s$ is admissible. ACA can be applied directly to compress the corresponding matrix $\boldsymbol{Z}_{t,s}$. Since the approximation is required to be nested and reusable for the parent-level clusters, a low-rank approximation should be performed with the superset of $t$ and $s$, instead of the local sets. Two possible approaches are shown in Fig. 1 to represent $\boldsymbol{Z}_{t,s}$ considering the entire $\mathcal{F}(t)$ and $\mathcal{F}(s)$, respectively. If $\boldsymbol{Z}_{t,\mathcal{F}(t)}$ has been approximated by ACA, $\boldsymbol{Z}_{t,s}$, which is just a submatrix of $\boldsymbol{Z}_{t,\mathcal{F}(t)}$, can be written as

$$\boldsymbol{Z}_{t,s} \approx \boldsymbol{Z}_{t,\sigma_t}(\boldsymbol{Z}_{\tau_t,\sigma_t})^{-1}\boldsymbol{Z}_{\tau_t,s} \qquad \forall\ s \subseteq \mathcal{F}(t) \tag{11}$$

where $\tau_t \subseteq t \subseteq \mathcal{F}(s)$, $\sigma_t \subseteq \mathcal{F}(t)$. Based on the same procedure, if $\boldsymbol{Z}_{\mathcal{F}(s),s}$ is approximated by a low-rank matrix via ACA as shown in Fig. 1(b), we have

$$\boldsymbol{Z}_{t,s} \approx \boldsymbol{Z}_{t,\sigma_s}(\boldsymbol{Z}_{\tau_s,\sigma_s})^{-1}\boldsymbol{Z}_{\tau_s,s} \qquad \forall\ t \subseteq \mathcal{F}(s) \tag{12}$$

where $\sigma_s \subseteq s \subseteq \mathcal{F}(t)$, $\tau_s \subseteq \mathcal{F}(s)$. Taking into account the pivots of the row and column clusters simultaneously, the combination of (11) and (12) results in (9).



(a) (b)

Fig. 1. Iillustration of two approaches for representing $\boldsymbol{Z}_{t,s}$. (a) Using $\boldsymbol{Z}_{t,\mathcal{F}(t)}$ (transpose is shown in brighter green). (b) Using $\boldsymbol{Z}_{\mathcal{F}(s),s}$ (transpose is shown in darker blue).

A form like (9) constitutes an $\mathcal{H}^2$-representation, from which we can directly recognize the following row and column cluster bases for $t$ and $s$:

$$\boldsymbol{U}_t = \boldsymbol{Z}_{t,\sigma_t}(\boldsymbol{Z}_{\tau_t,\sigma_t})^{-1} \tag{13}$$

$$\boldsymbol{V}_s = \left((\boldsymbol{Z}_{\tau_s,\sigma_s})^{-1}\boldsymbol{Z}_{\tau_s,s}\right)^H \tag{14}$$

which is only related to $t$ and $s$, respectively, and independent of the clusters being interacted. The coupling matrix for the admissible block formed between $t$ and $s$ can also be readily recognized as

$$\boldsymbol{S}_{t,s} = \boldsymbol{Z}_{\tau_t,\sigma s}. \tag{15}$$

Furthermore, the cluster bases shown in (13) and (14) are nested. To see this point clearly, we consider a nonleaf row cluster $t = t_1 \cup t_2$ in which both $t_1$ and $t_2$ are the two children clusters. Using (13), the cluster basis of $t$ is

$$\boldsymbol{U}_t = \begin{bmatrix} \boldsymbol{U}_t|_{t_1} \\ \boldsymbol{U}_t|_{t_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z}_{t_1,\sigma_t}(\boldsymbol{Z}_{\tau_t,\sigma_t})^{-1} \\ \boldsymbol{Z}_{t_2,\sigma_t}(\boldsymbol{Z}_{\tau_t,\sigma_t})^{-1} \end{bmatrix}. \tag{16}$$

Due to the nested relationship that $\sigma_t \subseteq \mathcal{F}(t) \subseteq \mathcal{F}(t_1)$ and $\sigma_t \subseteq \mathcal{F}(t) \subseteq \mathcal{F}(t_2)$, using (11), we obtain

$$\boldsymbol{U}_t \approx \begin{bmatrix} \boldsymbol{Z}_{t_1,\sigma_{t_1}}(\boldsymbol{Z}_{\tau_{t_1},\sigma_{t_1}})^{-1}\boldsymbol{Z}_{\tau_{t_1},\sigma_t}(\boldsymbol{Z}_{\tau_t,\sigma_t})^{-1} \\ \boldsymbol{Z}_{t_2,\sigma_{t_2}}(\boldsymbol{Z}_{\tau_{t_2},\sigma_{t_2}})^{-1}\boldsymbol{Z}_{\tau_{t_2},\sigma_t}(\boldsymbol{Z}_{\tau_t,\sigma_t})^{-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_{t_1} & \\ & \boldsymbol{U}_{t_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{T}^U_{t_1,t} \\ \boldsymbol{T}^U_{t_2,t} \end{bmatrix} \tag{17}$$

where

$$\boldsymbol{T}^U_{t',t} = \boldsymbol{Z}_{\tau_{t'},\sigma_t}(\boldsymbol{Z}_{\tau_t,\sigma_t})^{-1} \tag{18}$$

is the transfer matrix for row cluster basis. A similar procedure can also be applied to the column clusters to find the transfer matrix $\boldsymbol{T}^V_{s',s}$ for column cluster basis as the following:

$$\boldsymbol{T}^V_{s',s} = (\boldsymbol{Z}_{\tau_{s'},\sigma_s}(\boldsymbol{Z}_{\tau_s,\sigma_s})^{-1})^H. \tag{19}$$

For EFIE, the NCA procedure can be simplified by exploiting the symmetry of the system matrix. As shown in Fig. 1, $t \times \mathcal{F}(t)$ is exactly the transpose of $\mathcal{F}(t) \times t$, and so are $s \times \mathcal{F}(s)$ and $\mathcal{F}(s) \times s$. Take a cluster $x$ as an example. We denote $x = xr$ when $x$ is the row cluster, and $x = xc$ when $x$ is the column cluster. We have $\tau_{xr} = \sigma_{xc}$, i.e., the row pivots of a row cluster $x$ are the same as the column pivots of a column cluster $x$. As we will state in the next section, the symmetry can also be utilized to reduce the cost to determine the pivots for each row and column cluster. In addition, the construction of $\boldsymbol{V}_s$ can be omitted by using

$$\boldsymbol{V}_{xc} = \overline{\boldsymbol{U}}_{xr} \tag{20}$$

and the transfer matrices have a similar symmetric property.

From (13)–(15), and (18), it can be readily seen that the key to develop a low complexity NCA algorithm is to efficiently generate row and column pivots $\tau_t$ and $\sigma_t$ of each cluster $t$. After the pivots being obtained, the number of operations to construct cluster basis $\boldsymbol{U}_t$ ($\boldsymbol{V}_s$) at the leaf level, the transfer matrices $\boldsymbol{T}$ at the nonleaf levels, and the coupling matrices $\boldsymbol{S}_{t,s}$ of each admissible block are all of the linear complexity [19]. Therefore, how to select the pivots representing the blocks formed by each cluster at the same tree level, as well as those at the parent levels, is the key to an efficient NCA algorithm. In the next two sections, we detail the proposed algorithms for locating the pivots.

### *B. Proposed $\mathcal{O}(N \log N)$ NCA Algorithm*

A straightforward implementation of the NCA algorithm results in a complexity of $\mathcal{O}(N^2)$. This is because the row (column) pivots of each cluster should be obtained by considering the entire far field. Take the structure shown in Fig. 2 as an example. To find the cluster $t_1$'s row pivot set $\tau_{t_1}$ and its column pivot set $\sigma_{t_1}$, the ACA must be performed on not only the $(t_1, s_1)$ block but also the $(t_1, s_2)$, and $(t_1, s_3)$ blocks. In other words, the entire far field of $t_1$, which consists of clusters $s_1$, $s_2$, and $s_3$ must be considered. In [19], the basis functions were projected onto the Chebyshev nodes to shrink the matrix size operated by the ACA, which led to a complexity of $\mathcal{O}(N \log N)$. However, the approach required the geometrical information of clusters. In this section, we propose a purely algebraic algorithm with $\mathcal{O}(N \log N)$ complexity to perform the NCA. In addition, it facilitates a new NCA algorithm of a further reduced complexity of $\mathcal{O}(N)$, which is described in Section III-C.

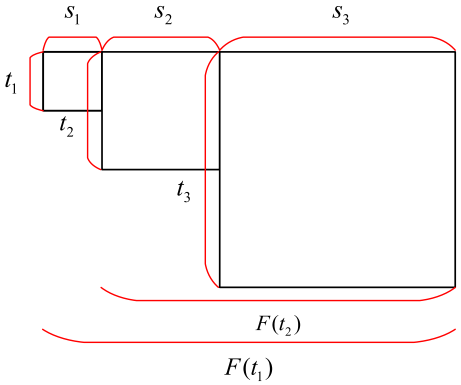


Fig. 2. Illustration of a matrix with the clusters and their far fields highlighted.

To efficiently account for the contribution of the far field of a cluster, we develop a top-down tree traversal algorithm as follows. We start from the root level $l = 0$ and descend along the inverted tree. At the highest level (closest to the root) that contains admissible blocks, the parent-level far field is null. Therefore, the local column pivots $\sigma$ of each cluster at this level are also global ones. As shown in Fig. 2, cluster $t_3$ is such a cluster, of which the parent-level far field is null.

We then descend the cluster tree by one level. We denote $\sigma_{t_p}$ as the column pivots of $t$'s parent cluster $t_p$, and col_set($t$) as the column set of the local far field of $t$, which includes all unknown indices contained in the column clusters that form admissible blocks with $t$ at $t$'s tree level. For each cluster $t$ at this level, the column pivots of the parent cluster $t_p$ are appended to the column set of the local far field of $t$

$$\text{col\_set}(t) \leftarrow \text{col\_set}(t) + \sigma_{t_p}. \tag{21}$$

In this way, we are able to account for the admissible blocks at $t$'s parent level so that the resultant pivots constitute a nested representation of the original matrix. With $\sigma_t$ determined, we perform an ACA with all the indices in $t$ as the row set and the updated col_set($t$) as the column set. Note that col_set($t$) in (21) is a largely reduced far field of $\mathcal{F}(t)$. This can be done because the admissible blocks formed by parent cluster $t_p$ have already been accurately represented by ACA using column pivots $\sigma_{t_p}$. After performing ACA in the reduced far field shown in (21) for a prescribed accuracy, we determine the final row and column pivots of $t$, which is $(\tau_t, \sigma_t)$.

With the aforementioned process, we continue to descend the cluster tree to one level down. For each cluster $t$ at this level, we append the column pivots of its one-level-up parent cluster $t_p$ to the local column set of this cluster, as shown in (21). Note that here $t_p$ is $t$'s immediate parent, as even higher level parents' pivots have already been taken into account in $\sigma_{t_p}$ during the level-by-level procedure. Similarly, new pivots $(\tau_t, \sigma_t)$ for cluster $t$ at this level are obtained. The whole procedure is continued until we reach the leaf level.

To give an example, we use the matrix shown in Fig. 2 to illustrate the whole procedure. In this example, we have $t_1 \subseteq t_2 \subseteq t_3$ and $\mathcal{F}(t_3) = s_3$, $\mathcal{F}(t_2) = s_2 \cup \mathcal{F}(t_3)$, and

$\mathcal{F}(t_1) = s_1 \cup \mathcal{F}(t_2)$. We start from the highest level that has admissible blocks, which is the level of $t_3$. Since the block $t_3 \times s_3$ is an admissible block at the highest level, ACA is directly adopted to obtain $\tau_{t_3}$ and $\sigma_{t_3}$. We then go one level down and use col_set$(t_2) + \sigma_{t_3}$ instead of complete $\mathcal{F}(t_2)$ as the column set to perform an ACA to obtain $t_2$'s pivots. We then continue to descend the tree. At $t_1$'s level, we use col_set$(t_1) + \sigma_{t_2}$ as the column set to perform an ACA to obtain $t_1$'s pivots.

**Algorithm 1** Top-Down Pivot Selection I

```
 1: procedure TOP-DOWN-I($c$)
 2:     $row\_set = c.all\_indices$
 3:     $col\_set = [\,]$
 4:     for each $temp$ that $c \times temp \in \mathcal{P}_{adm}$ do
 5:         $col\_set$.append($temp.all\_indices$)
 6:     end for
 7:     if $c.parent \neq$ NUL & $c.parent.\sigma.size \neq 0$ then
 8:         $col\_set$.append($c.parent.\sigma$)
 9:     end if
10:     $(c.\tau, c.\sigma) =$ ACA($row\_set$, $col\_set$)
11:     for each $child$ of $c$ do
12:         TOP-DOWN-I($child$)
13:     end for
14: end procedure
```

The pseudocode of the top-down tree traversal for nested pivots selection is shown in Algorithm 1. In this algorithm, the argument $c$ denotes a cluster, and the procedure should be invoked from the root cluster. The local far field of $c$ is searched first to generate the column set (col_set), which is described by lines 3–6. The row set (row_set) contains all the indices of $c$, as shown by line 2. If the cluster has a parent (not the root), and the parent's column pivots are not null, the column set col_set is appended by the column pivots of the immediate parent cluster. After the ACA, the new pivots $(c.\tau,\ c.\sigma)$ of $c$ are determined. The procedure recursively proceeds to the children clusters until the leaf level is reached.

The complexity of the top-down pivot selection for constant-rank cases shown in Algorithm 1 can be found as

$$\begin{aligned}\#\text{operations} &\leq \sum_{l=1}^{L} 2^l \mathcal{O}\left(\frac{N}{2^l} + C_{\text{sp}} \cdot \frac{N}{2^l} + k\right) \cdot k^2 \\ &= \mathcal{O}(N \log N). \qquad (22)\end{aligned}$$

Here, $L \propto \log N$ is the height of the cluster tree, $C_{\text{sp}}$ is the sparsity constant representing the maximum number of admissible blocks formed by one cluster, $2^l$ is the number of clusters at level $l$, and $(N/2^l)$ is the number of unknowns contained in a cluster at level $l$. The $C_{\text{sp}} \cdot (N/2^l) + k$ term in (22) denotes the column dimension of the matrix block on which ACA is performed, and here $k$ is the additional column number contributed by the parent cluster.

### *C. Proposed $\mathcal{O}(N)$ NCA Algorithm*

From the Algorithm 1, it can be seen clearly that if we can avoid operating on the complete row and column sets of each admissible block, it is possible to obtain an $\mathcal{O}(N)$ algorithm. In this section, we propose a two-stage algorithm to reach this point. In the first stage, we propose a bottom-up tree traversal algorithm to obtain the local pivots with linear complexity. In the second stage, we exploit the top-down strategy similar to Algorithm 1 to take into account the far field.

*1) Stage I: Bottom-Up Tree Traversal for Local Choices of Cluster Pivots:* Considering an arbitrary nonleaf cluster $t$, we propose to choose the row pivots of $t$ from the union of the row pivots of its two children clusters $t_1$ and $t_2$. Assume that $t = t_1 \cup t_2$, $\tau_{t_1}$ and $\tau_{t_2}$ are the row pivots of $t_1$ and $t_2$ respectively. The row pivots $\tau_t$ can be chosen from

$$\tau_t = \tau_{t_1} \cup \tau_{t_2}. \qquad (23)$$

In this way, we shrink the size of the row set of an admissible block, on which the ACA is performed, from $\#t$ to $\mathcal{O}(k)$. This appears to be a heuristic choice without accuracy control. In fact, (23) is accurate for choice because of the following reason.

Consider an arbitrary matrix $\boldsymbol{Z}_{t,s}$, whose row/column rank is $k$ for a prescribed accuracy. If using ACA to find it, we have

$$\boldsymbol{Z}_{t,s} \approx \boldsymbol{Z}_{t,\sigma}(\boldsymbol{Z}_{\tau,\sigma})^{-1}\boldsymbol{Z}_{\tau,s} \qquad (24)$$

where $\boldsymbol{Z}_{t,\sigma}$ denotes $k$ columns of $\boldsymbol{Z}$, whose column pivots are contained in $\sigma$, and $\boldsymbol{Z}_{\tau,s}$ represents $k$ rows of $\boldsymbol{Z}$, whose row pivots are contained in $\tau$. As can be seen from this representation, any column of $\boldsymbol{Z}$ is a linear superposition of the $k$ columns in $\boldsymbol{Z}_{t,\sigma}$. If one does not choose the $k$ columns in $\sigma$, but choosing another set of linearly independent $k$ columns, whose column pivots are contained in $\sigma^*$, we can find the following relationship between $\boldsymbol{Z}_{t,\sigma^*}$ and $\boldsymbol{Z}_{t,\sigma}$:

$$\boldsymbol{Z}_{t,\sigma^*} \approx \boldsymbol{Z}_{t,\sigma}\boldsymbol{A}_{k\times k} \qquad (25)$$

where $\boldsymbol{A}_{k\times k} = (\boldsymbol{Z}_{\tau,\sigma})^{-1}\boldsymbol{Z}_{\tau,\sigma^*}$, which can be readily obtained using (24). Similarly, any row of $\boldsymbol{Z}$ is a linear superposition of the $k$ rows in $\boldsymbol{Z}_{\tau,s}$, and any $k$ linearly independent rows of $\boldsymbol{Z}$ can be used to build a rank-$k$ model of (24) with the same accuracy. In other words, the choice of row pivots is not unique either. Specifically, if we choose row pivots as $\tau^*$, we have

$$\boldsymbol{Z}_{\tau^*,s} \approx \boldsymbol{B}_{k\times k}\boldsymbol{Z}_{\tau,s} \qquad (26)$$

where $\boldsymbol{B}_{k\times k} = \boldsymbol{Z}_{\tau^*,\sigma}(\boldsymbol{Z}_{\tau,\sigma})^{-1}$. Therefore, if one does not happen to choose $\sigma$ as the column pivot, and $\tau$ as the row pivot; but randomly select a column pivot of $\sigma^*$, and a row pivot of $\tau^*$, a rank-$k$ model of this matrix can be obtained with the same accuracy, because using (25) and (26), the original (24) can be rewritten as

$$\boldsymbol{Z}_{t,s} \approx \boldsymbol{Z}_{t,\sigma^*}(\boldsymbol{A}^{-1}\boldsymbol{Z}_{\tau,\sigma}{}^{-1}\boldsymbol{B}^{-1})\boldsymbol{Z}_{\tau^*,s}. \qquad (27)$$

Hence, choosing another column pivot and row pivot can also yield an accurate representation of the matrix. The only difference between (27) and (24) is that the coupling matrix (the center matrix) now is different.

If the rank at children's level, $l$, is $k$, then the rank of the admissible blocks at the parent's level, which is $l-1$, cannot be greater than $2k$. This is because, for SIE, the electrical

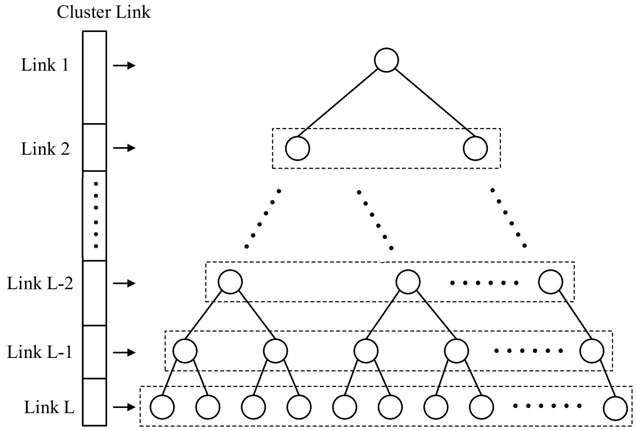


Fig. 3. Cluster link for each level of the cluster tree.

size increases from one level to the other by $\sqrt{2}$, as unknown number doubles. Using a full-rank's growth rate, which is electrical size square, we at most get a rank of $2k$ at parent's level. Therefore, in terms of the number of the row pivots we select for parent cluster $t$, (23) is sufficient. As for which rows to select out of the $\#t$ rows in cluster $t$, as can be seen from the analysis in the paragraph above, any linearly independent $k'$ rows are accurate for choice, where $k'$ is the rank at $t$'s level, and $k' < 2k$. In addition, given any set of rows, the final rows selected using an ACA procedure are linearly independent because of the nature of the ACA algorithm. Therefore, out of the $2k$ rows, the rows selected by ACA are linearly independent. As a result, (23) is a rich set to determine the low-rank model for $t$'s admissible blocks formed at $t$'s level. In the rare case where the number of rows selected from ACA is smaller than the target rank, i.e., the number of the linearly independent rows in $\tau_t$ is smaller than the actual rank of the admissible blocks formed by $t$, more rows can be chosen from the children row pivots by increasing the accuracy via decreasing $\epsilon_{\text{ACA}}$.

Using (23), we proceed from the leaf level all the way up to the highest level that contains admissible blocks. At each level, after choosing the row set from (23), we perform an ACA on the admissible blocks formed by $t$ in such a reduced row set. After the ACA, the final pivots $\tau_t$ of $t$ are obtained. Since the column clusters play the same role as the row clusters in the construction of the $\mathcal{H}^2$-matrix, the column pivots $\sigma_s$ can also be determined in the same way from the two children's column pivot sets. For the symmetric matrices, the column pivots can be chosen the same as the corresponding row pivots, thus avoiding additional computation.

The pseudocode of the bottom-up local pivot selection is depicted in Algorithm 2. Before invoking the bottom-up procedure, the cluster link (c_link) that stores all the pointers to clusters at each level is built by traversing the cluster tree. Each pointer in c_link acts as the head of the link for clusters at the corresponding level, as shown in Fig. 3. We start from the bottom ($l = L$) and ascend level by level along the cluster tree. For a leaf cluster, the row index set contains all the indices in the cluster, and the column index set contains all the indices in the union of the local far field. This is described by lines 6–11. Once ACA is performed (line 21 of Algorithm 2), the row pivots for each leaf cluster are determined and stored

**Algorithm 2** Bottom-Up Local Pivot Selection

```
1: procedure BOTTOM-UP($c\_link$)
2:     $l = L$
3:     while $l > 0$ do
4:         $link = c\_link[l]$
5:         for each $c$ in $link$ do
6:             if $c$ is leaf cluster then
7:                 $row\_set = c.all\_indices$
8:                 $col\_set = [\,]$
9:                 for each $temp$ that $c \times temp \in \mathcal{P}_{adm}$ do
10:                    $col\_set$.append($temp.all\_indices$)
11:                end for
12:            else
13:                $row\_set = [\,]$, $col\_set = [\,]$
14:                for each $child$ of $c$ do
15:                    $row\_set$.append($child.\tau$)
16:                end for
17:                for each $temp$ that $c \times temp \in \mathcal{P}_{adm}$ do
18:                    $col\_set$.append($temp.\tau$)
19:                end for
20:            end if
21:            $(c.\tau, c.\sigma)$ = ACA($row\_set$, $col\_set$)
22:        end for
23:        $l = l - 1$
24:    end while
25: end procedure
```

in $c.\tau$ and they are ready to replace the complete index set of $c$ for the local low-rank approximation. By exploiting the symmetry of system matrix mentioned in (20), the pivot sets $\sigma_{tc}$ for column clusters are directly copied from $\tau_{tr}$. As the procedure moves upward, at each nonleaf level, the pivots of its two children are employed to replace the original indices in the nonleaf clusters before performing ACA. This can be seen from lines 13 to 20.

*2) Stage II: Global Choice of Pivots by a Top-Down Tree Traversal to Take Into Account the Far Field of Each Cluster:* The pivot sets $\tau$ and $\sigma$ resulted from the bottom-up stage contain only the local pivots. In Stage II, we use the top-down approach developed in Algorithm 1 to account for the far field of each cluster by appending the pivots from higher levels (levels closer to the root). However, the pivots of each cluster have been reduced to a small set of $\mathcal{O}(k)$ size due to Stage I. The detailed algorithm of the top-down pivot selection II is shown in Algorithm 3.

The procedure still starts from the root of the cluster tree and performed in a top-down manner. If the cluster $t$ has a parent cluster and the parent cluster has a nonempty pivot set, then the computation is performed. Different from (21), the col_set in Algorithm 3 denotes the union of $t$'s local column pivots $\sigma_t$ from Stage I and $t$'s parent column pivots

$$\text{col\_set}(t) = \sigma_t + \sigma_{t_p}. \tag{28}$$

The generation of the row set is also revised. In Algorithm 1, the row set contains all the indices of $c$. Here in Algorithm 3, we use a reduced row pivot set $c.\tau$. However, the ACA generated pivot sets $c.\tau$ and $c.\sigma$ from Stage I have an equal

**Algorithm 3** Top-Down Pivot Selection II

```
 1: procedure TOP-DOWN-II($c$)
 2:     if $c.parent \neq$ NUL & $c.parent.\sigma.size \neq 0$ then
 3:         $row\_set = [\,]$
 4:         $col\_set = c.\sigma + c.parent.\sigma$
 5:         $bound = col\_set.size$
 6:         if $c$ is leaf cluster or $c.size < bound$ then
 7:             $row\_set = c.all\_indices$
 8:         else
 9:             $des\_clusters$=FIND-OFFSPRING($c$, $bound$)
10:             for each $item$ of $des\_clusters$ do
11:                 $row\_set.append(item.\tau)$
12:             end for
13:         end if
14:         $(c.\tau, c.\sigma) =$ ACA($row\_set$, $col\_set$)
15:     end if
16:     for each $child$ of $c$ do
17:         TOP-DOWN-II($child$)
18:     end for
19: end procedure
```

size. The row_set must be augmented as the col_set has been augmented by the parent cluster's column pivots. We first use the size of the col_set as the bound to determine the size of the row_set. If $c$ is a leaf cluster or the number of all the indices in $c$ is less than the size of the col_set, we directly use all the indices of $c$ to generate the row_set. Otherwise, we call another subroutine FIND-OFFSPRING to search for the highest level of $c$'s descendant clusters (des_clusters), of which the union of all the row pivot sets has a sufficiently large size to match bound. This step is shown in line 9 in Algorithm 3. The row pivot set of each cluster in des_clusters is then appended to the row_set as can be seen from line 10 to 12. After augmenting the row_set, the ACA is performed to determine the final row and column pivot set for $c$, as shown by line 14. The subroutine FIND-OFFSPRING to find the abundant offspring clusters to augment the local row pivot set is depicted in detail in Algorithm 4 shown in the Appendix.

It should be noted that the pivots $\tau_t$ and $\sigma_s$ selected from Stage I, the bottom-up stage, are not complete for representing all admissible blocks since they are from the local far field instead of the entire far field. However, the local pivots are sufficient to feed Stage II, top-down procedure, as the premise. Therefore, the complete pivot selection in the proposed algorithm has the following two steps.

1) Bottom-up local pivot selection.
2) Top-down pivot selection.

In the bottom-up local pivot selection, only the local far field for each cluster is considered. Furthermore, the pivots for a row cluster are chosen from its two children clusters, which bound the matrix size to $\mathcal{O}(k)$, upon which the ACA is performed. At an arbitrary $l$th level, the number of operations of the proposed algorithm is bounded by

$$OP_l \leq 2^l(2k + C_{\text{sp}} \cdot 2k) \cdot k^2. \tag{29}$$

Taking all of the $L$ levels into account, the total number of the operations is bounded by

$$\#\text{operations} \leq \sum_{l=1}^{L} 2^l \mathcal{O}(2k + C_{sp} \cdot 2k) \cdot k^2 \tag{30}$$

which is $\mathcal{O}(N)$, when rank $k$ is a bounded constant. If $k$ grows with tree level, but in a small growth rate, like that in cases where the electrical size is less than 5 wavelengths, the $k$'s growth can be bounded by a rank function given in [14]. In such a case, the complexity of (30) can also be found to be linear.

In the top-down pivot selection II, the indices for the row cluster and the corresponding far field are substituted by the local pivots chosen in the first step, whose size is $\mathcal{O}(k)$. The number of operations, therefore, becomes

$$\#\text{operations} \leq \sum_{l=1}^{L} 2^l \cdot \mathcal{O}(2(k+k) + k + k) \cdot k^2 \tag{31}$$

which is also $\mathcal{O}(N)$ for problems whose electrical size is not large, with the same reasoning given above.

### *D. Iterative and Direct Solutions*

The nested structure is explicitly constructed and represented by an $\mathcal{H}^2$-matrix, thus both the iterative and the direct solutions are feasible. The iterative solution with a fast matrix-vector multiplication operation using the $\mathcal{H}^2$-matrix provides a linear complexity for each iteration [14]. The direct solutions with linear complexity, including both inversion and factorization, have also been developed for general $\mathcal{H}^2$-matrices [15], [23]. We can choose an appropriate solution based on specific applications.

## IV. Numerical Results

In this section, several numerical examples are studied to demonstrate the performance of the proposed methods. In the first example, the radar cross section (RCS) of a PEC sphere is shown to validate the accuracy of the proposed method against the analytical Mie Series solution. Large-scale interconnects, PEC cubes, and the inductor array are then simulated to examine the accuracy, efficiency, and complexity of the proposed method. The Koch Snowflake fractal structure is studied to demonstrate the capability of the proposed method in handling inhomogeneous meshes and multiscaled problems. Finally, the performance of the proposed method in solving electrically larger problems is also examined.

### *A. Accuracy Validation*

We first compute the RCS of a PEC sphere of radius 3 m at 300 MHz. The discretization results in 29 802 unknowns. The incident electric field is a plane wave propagating along $-z$ direction ($\theta = 0°$). The RCS at the $\phi = 0°$ plane from the proposed method is compared with the analytical Mie Series solution using an expansion of 100 modes [24]. In this example, we set the maximum leaf size to be $n_{\max} = 200$, the admissibility condition coefficient $\eta = 1$, and $\epsilon_{\text{ACA}} = 10^{-3}$,

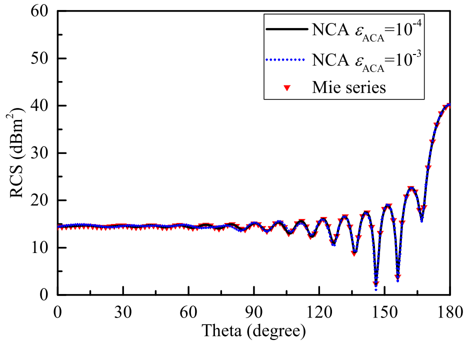
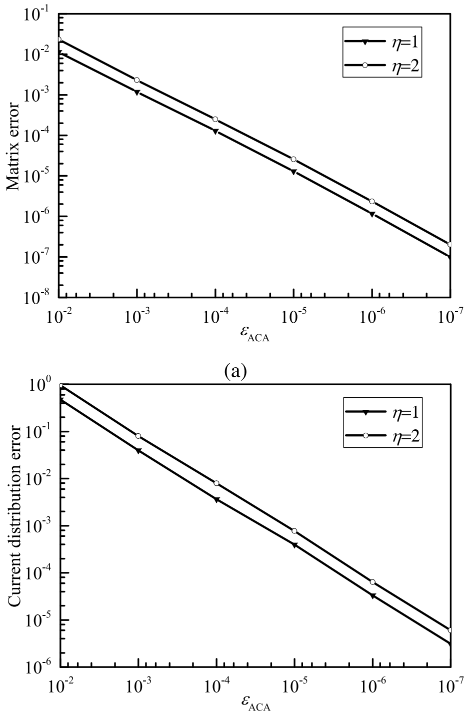


Fig. 4. RCS of a PEC sphere of radius 3 m at 300 MHz.



(b)

Fig. 5. (a) Matrix error with respect to varying ACA tolerance of the sphere example. (b) Current error with respect to $\epsilon_{ACA}$.

$10^{-4}$, respectively, for different accuracy requirements. The conjugate gradient (CG) method is employed as the iterative solver with an error tolerance of $10^{-4}$.

It can be observed from Fig. 4, the accuracy resulting from $\epsilon_{\text{ACA}} = 10^{-4}$ is better than that of $\epsilon_{\text{ACA}} = 10^{-3}$ and both agree well with the Mie series solution. In addition, we examine the entire matrix error and the surface current error with respect to different choices of $\epsilon_{\text{ACA}}$. The matrix error is assessed by $\|\mathbf{Z}^* - \mathbf{Z}\|_F/\|\mathbf{Z}\|_F$, in which $\mathbf{Z}$ is the original system matrix, $\mathbf{Z}^*$ is the $\mathcal{H}^2$-matrix representation from the proposed algorithm, and $F$ denotes the Frobenius norm. To compute this error, only the admissible blocks are involved since inadmissible ones are not approximated. The matrix entries of each admissible block are computed out and their differences with the full-matrix entries are calculated based on the Frobenius norm to evaluate the matrix error. The surface current error is evaluated by $\|I^* - I\|_2/\|I\|_2$, in which $I$ denotes the current from a direct LU solution of MoM, $I^*$ is the current obtained from the proposed method, and norm 2 is used. As shown in Fig. 5, for the same choice of $\eta$, both the entire matrix error and the current distribution error decrease nearly linearly with $\epsilon_{\text{ACA}}$. Furthermore, as expected, for the same choice of $\epsilon_{\text{ACA}}$, the errors from a more stringent choice of admissibility condition ($\eta = 1$) are smaller than those from a less stringent one ($\eta = 2$).

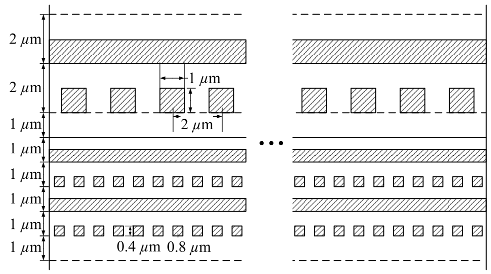
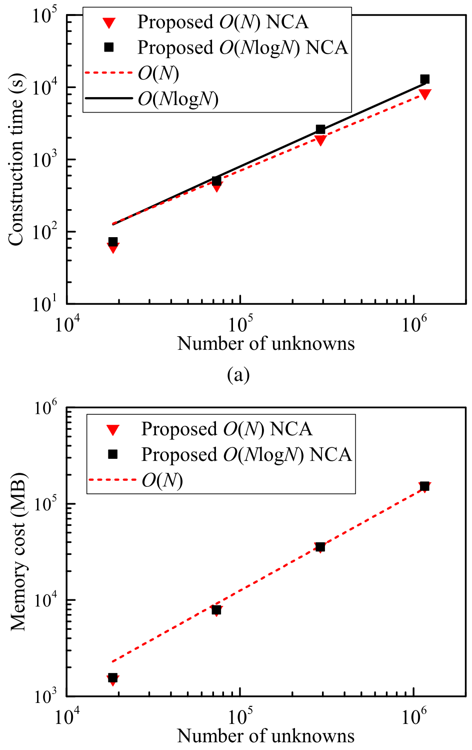


Fig. 6. Large-scale 3-D on-chip interconnects.



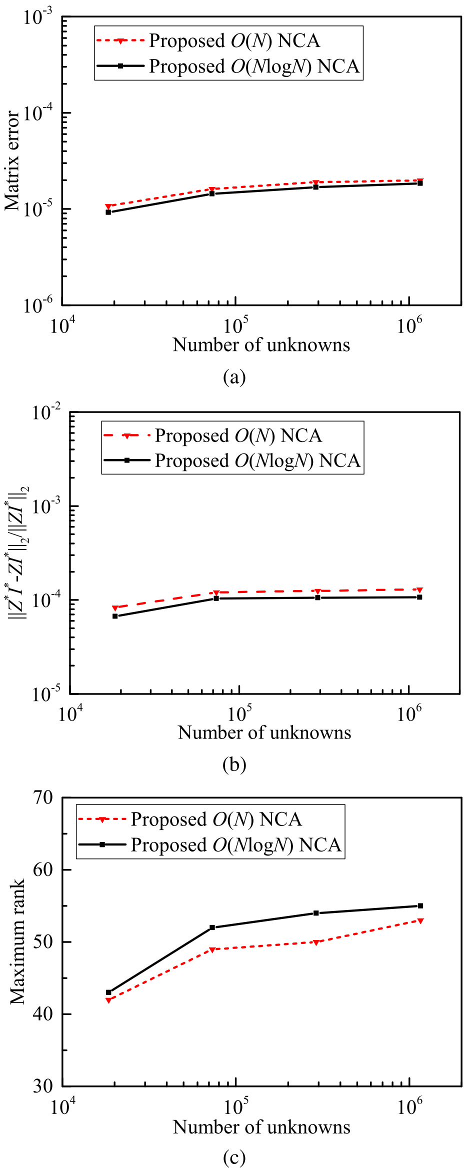
(b)

Fig. 7. Computational performance for simulating the interconnects. (a) CPU time cost for matrix construction. (b) Memory cost.

### *B. Large-Scale 3-D On-Chip Interconnects*

The second example is a large-scale 3-D interconnect structure [25] as shown in Fig. 6. Each conductor has the same length. We simulate a suite of such structures at 10 GHz with the absence of the dielectrics. The number of the conductors increases from 48, 96, 192 to 384, with the number of unknowns ranging from 18 468 to 1 159 656.

We test the performance of both the proposed $\mathcal{O}(N\log N)$ and the $\mathcal{O}(N)$ method. In both settings, the maximum leaf size is $n_{\max} = 200$, the ACA tolerance is $\epsilon_{\text{ACA}} = 10^{-3}$, and the admissibility condition coefficient is $\eta = 1$. The CPU time cost of the construction of the $\mathcal{H}^2$-matrix is shown in Fig. 7(a) for both methods. The complexity is shown to agree with our



(a)

(b)

(c)

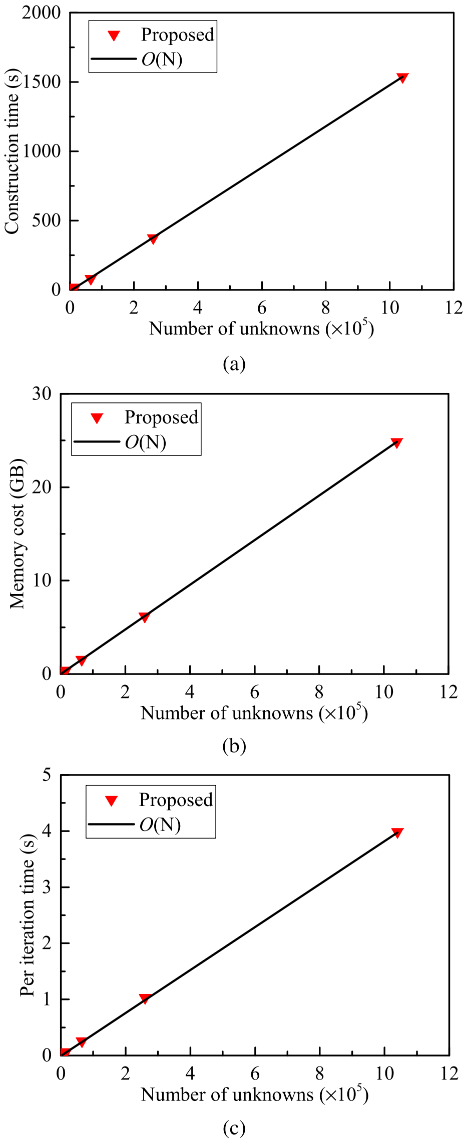
Fig. 8. Error measurements for simulating the interconnects. (a) Matrix error. (b) MVP error. (c) Maximum rank.

theoretical expectations. Since the matrix is represented as an $\mathcal{H}^2$-matrix, the memory cost for both methods are linear, as can be observed from Fig. 7(b).

The matrix error and the MVP error with respect to MoM are shown in Fig. 8. The $\mathcal{O}(N\log N)$ method is shown to have higher accuracy than the $\mathcal{O}(N)$ method using the same choice of accuracy parameters. This can be attributed to the fact that in the proposed $O(N)$ algorithm, the accuracy of ACA is assessed in the reduced set of rows and columns for each admissible block. Although the number of these rows and columns is sufficient for determining the cluster bases, using the same choice of $\epsilon_{\text{ACA}}$, the final matrix error would not be the same. However, good accuracy is achieved in the proposed $O(N)$ algorithm as well.

### C. Large-Scale Array of Spiral Inductors

A suite of large-scale spiral inductor arrays, which contains 1, 4, 16, 64, and 256 elements, is then computed at 10 GHz.



(a)

(b)

(c)

Fig. 9. Computational performance for simulating inductor arrays. (a) CPU time cost for matrix construction. (b) Memory cost. (c) CPU time cost for each iteration.

The inductor array is formed by extending a unit spiral inductor along both $x$- and $y$-axes. The 3-D view and the geometric structure of the inductor array are shown in [26]. The settings of $\eta$, $n_{\max}$, and $\epsilon_{\text{ACA}}$ are the same as those in the second example. The CG iterative solution is adopted with a threshold of $10^{-6}$.

The CPU time and memory cost for the matrix construction of the inductor array are both linear with respect to the number of unknowns, as shown in Fig. 9. The average time cost for MVP in each iteration is shown in Fig. 9(c). The matrix error and the solution error are shown in Fig. 10. It can be observed that the accuracy remains constant as the number of unknowns increases.

### D. Koch Snowflake Fractal Structure

The Koch Snowflake fractal structure is simulated to demonstrate the feasibility of the proposed method for handling multi-scale problems. We also discretize the structure using

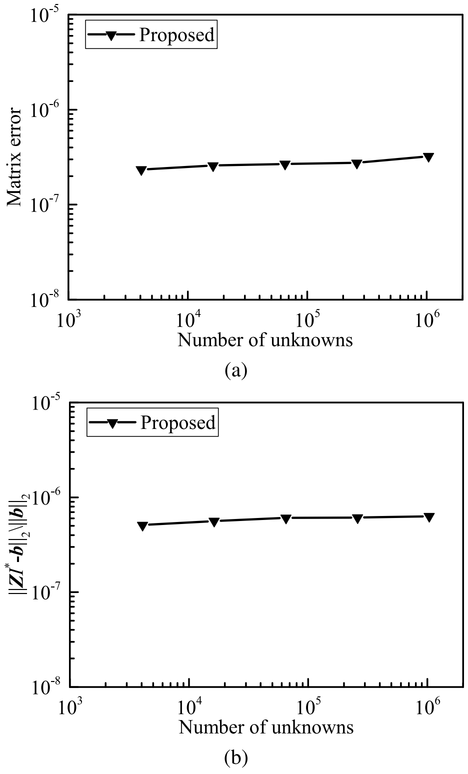


Fig. 10. Error measurements for simulating inductor arrays. (a) Matrix error. (b) Solution error.

inhomogeneous meshes to examine whether the proposed algorithms require a uniform mesh. The snowflake fractals are planar structures. They are constructed by starting from an equilateral triangle and altering each line segment recursively. The detailed process is depicted in [27].

The sequence of the fractal snowflake is shown in Fig. 11. The side length of the starting triangle is 1 m. Inhomogeneous meshes are generated to adjust to different iterations. Since the structure becomes finer as the iteration increases, we fix the largest mesh size ($h_{\max}$) to be $1/12$ m, and the finest mesh size ($h_{\min}$) to be the side length of the fractal equilateral triangles in each iteration. The simulation is performed from iterations 3 to 6, with $h_{\min}/h_{\max}$ ranging from 0.44 to 0.016. The resulting number of unknowns is from 618, 1176, 3474 to 12 696. The solving frequency is fixed at 300 MHz. The maximum leaf size is $n_{\max} = 40$, the admissibility coefficient $\eta = 1$, and the ACA threshold is $\epsilon_{\text{ACA}} = 10^{-3}$. The error estimations of the proposed method are shown in Fig. 12. As can be noticed, both the matrix and current errors can be kept at the same scale for different iterations.

### E. PEC Cubes

In this example, we consider a series of PEC cubes whose side length ranges from $0.25\lambda$ to $2\lambda$, which have been studied in [17]. For each cube, we fix the mesh size to be $h/\lambda = 10^{-2}$. It is worth mentioning that the rank's growth rate with $N$ is independent of the mesh size as can be seen from [22]. The number of unknowns ranges from 12 708, 51 462, 206 820 to 829 332 for discretizing these cubes. The admissibility condition coefficient is chosen as $\eta = 1$, the maximum leaf size is 100, and the ACA tolerance is chosen as $10^{-4}$. The iterative solution settings are the same as those in the first example.

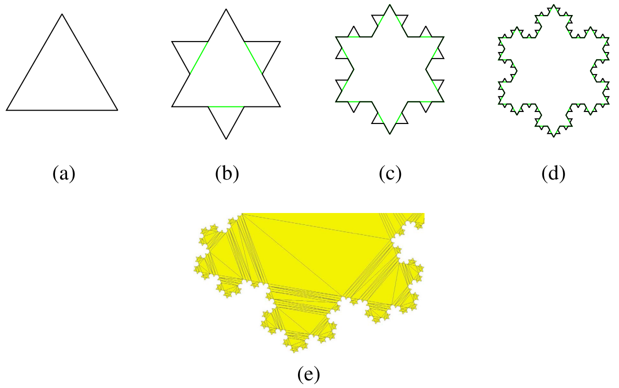


Fig. 11. Koch Snowflake fractal structure. (a) Zeroth iteration. (b) First iteration. (c) Second iteration. (d) Third iteration. (e) Illustration of the irregular mesh.

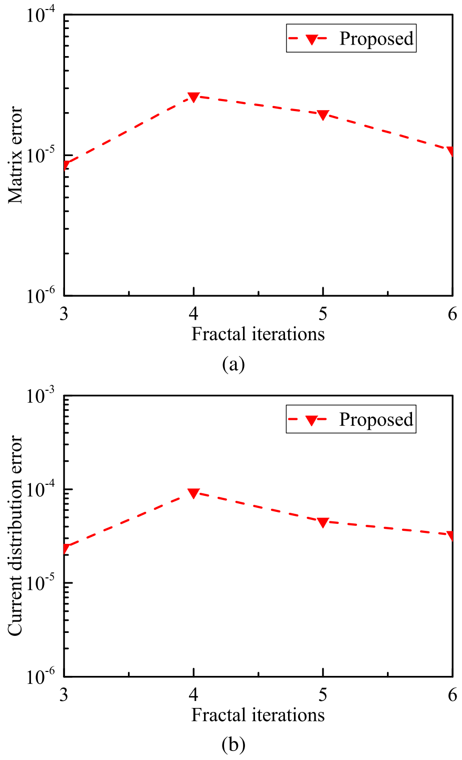


Fig. 12. Error measurement for simulating the Koch Snowflake fractal structure. (a) Matrix error. (b) Current distribution error.

The computational performance of the proposed method under the aforementioned setting is shown in Fig. 13. It can be observed that the CPU time and memory cost increase linearly with the number of unknowns, and the overall performance is no worse than that of [17]. Furthermore, different from [17], the proposed method does not require the predefined auxiliary equivalent basis functions on the sphere surfaces, which provides more flexibility and enables purely algebraic operations. Due to the difficulty in obtaining a full-matrix-based MoM solution, the solution errors are measured by relative residual, which is the relative difference between $ZI^{*}$ and excitation

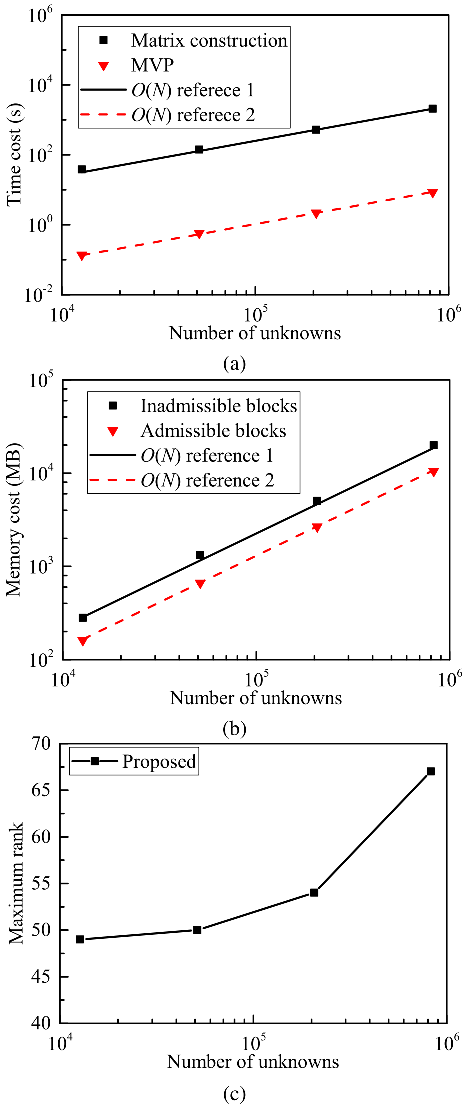


Fig. 13. Computational performance for simulating PEC cubes. (a) CPU time cost for matrix construction and MVP. (b) Memory cost. (c) Maximum rank.

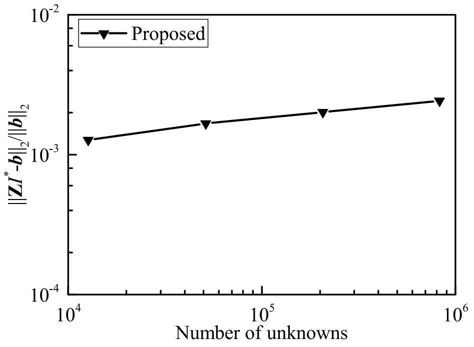


Fig. 14. Solution error for simulating PEC cubes.

vector $\boldsymbol{b}$, as shown in Fig. 14. It is evident that the solution errors are kept as nearly constant, and the maximum rank shown in Fig. 13(c) increases slowly with the increasing number of unknowns.

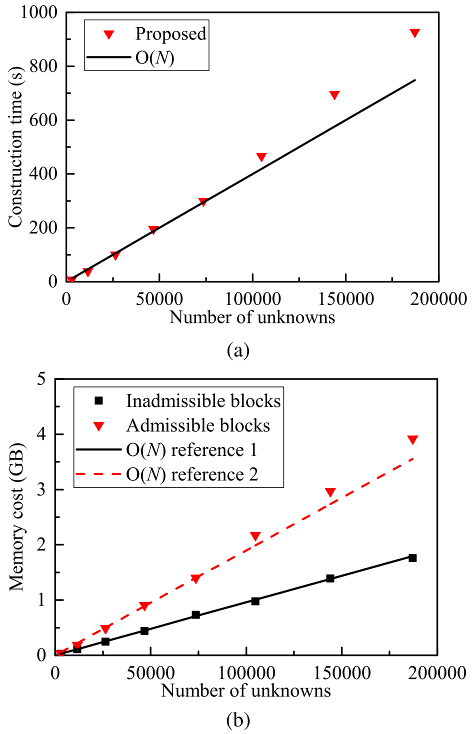


Fig. 15. Computational performance for simulating PEC sphere with increasing radius. (a) CPU time cost for matrix construction. (b) Memory cost.

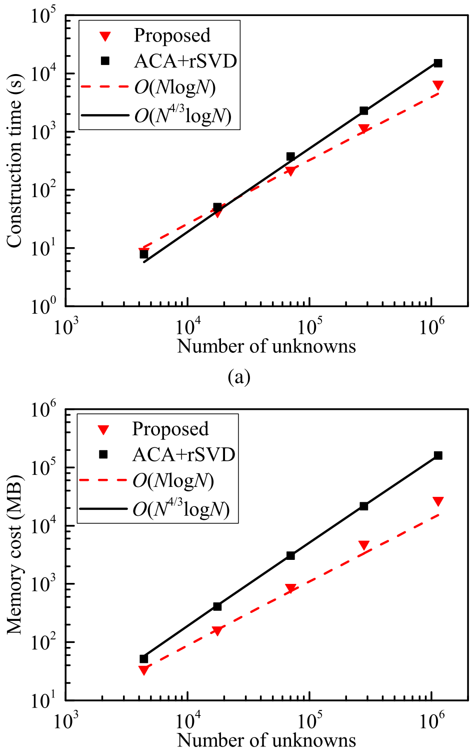
### *F. Electrically Large Analysis*

The computational complexity presented in previous sections requires that the rank of the admissible blocks be bounded by constant, or grows slowly with electrical size. This is not the case for electrically large problems when using an ACA-based method to generate a rank-$k$ model. Here we use a PEC sphere with an increasing radius to illustrate this point. The radius of the sphere increases linearly from 0.5 to 4 m, therefore the corresponding electrical size increases linearly from 1 to 8. The solving frequency is fixed at 300 MHz, and the mesh size is chosen to be 0.04 m. The number of unknowns ranges from 2934 to 187 143. In this setting, $\eta = 1$, $n_{\max} = 50$, and $\epsilon_{\text{ACA}} = 10^{-3}$.

The CPU time and memory cost are shown in Fig. 15. It can be observed that the memory cost for the inadmissible blocks is always linear with the number of unknowns due to the basic property of the $\mathcal{H}^2$-matrix. The CPU time cost and the memory cost of the admissible blocks, which are related to the low-rank compressions, can only keep its linear growth to a certain electrical size, which is about 6 wavelengths in this specific example. However, even in this regime, the method is more efficient than ACA-based methods, as can be seen from the following example.

In this example, we simulate a PEC cube of side length 5 m. The solving frequency ranges from 100 MHz to 1.6 GHz. The mesh size is fixed to be $h/\lambda = 1/8$.

Here, we use the $\mathcal{H}$-matrix as the reference to illustrate the performance of the proposed method in the high-frequency regime. The ACA and the reduced SVD algorithms are adopted to obtain the $\mathcal{H}$-matrix. In Fig. 16, we plot the memory and time cost as a function of $N$, and also reference



(a)

(b)

Fig. 16. Computational performance for simulating electrically larger cube. (a) CPU time cost for matrix construction. (b) Memory complexity.

complexity lines for comparison. As can be seen from Fig. 16, the proposed NCA algorithm is more efficient than ACA due to the nested property.

## V. Conclusion

In this work, a fast NCA algorithm is developed for solving electromagnetics problems. The overall procedure is made purely algebraic without the use of any auxiliary surfaces or geometric projection. Hence, the proposed algorithm can be integrated into the IE solvers for different types of Green's functions with great ease. For constant rank problems, for instance, the circuit simulation or electrically small scattering problems, the proposed method has a complexity of $\mathcal{O}(N)$. Numerical results have demonstrated the accuracy, efficiency, and complexity of the proposed algorithm. It is also worth mentioning that the proposed method is equally applicable to electrically large problems. Although in its current form, the resulting complexity would be higher than $\mathcal{O}(N)$, it is more efficient than ACA-based methods.

## Appendix

### A. Finding Abundant Offspring Clusters

The union of the clusters (clts) is initialized by the cluster $c$. The breadth-first search technique is adopted to generate all the offspring clusters of $c$ at one level in each loop. The function count_pivots(clts) sums up the size of the row pivot set of each cluster in clts. The FIND-OFFSPRING subroutine returns the cls when either all the leaves are included or the bound is reached.

**Algorithm 4** Finding Abundant Offspring Clusters

```
procedure FIND-OFFSPRING($c$, $bound$)
    $clts$ = [ ], $temp$ = [ ]
    $clts$.append($c$)
    while count_pivots($clts$) < $bound$ do
        swap($clts$, $temp$)
        for each $item$ of $temp$ do
            if $item$ is leaf cluster then
                $clts$.append($item$)
            else
                for each $child$ of $item$ do
                    $clts$.append($child$)
                end for
            end if
        end for
        if $clts$.size()==$temp$.size() then
            break
        end if
        $temp$.clear()
    end while
    return $clts$
end procedure
```

## References

[1] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. AP-30, no. 3, pp. 409–418, May 1982.

[2] J. M. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[3] Z. G. Qian, W. C. Chew, and R. Suaya, "Generalized impedance boundary condition for conductor modeling in surface integral equation," *IEEE Trans. Microw. Theory Techn.*, vol. 55, no. 11, pp. 2354–2364, Nov. 2007.

[4] D. Vande Ginste, E. Michielssen, F. Olyslager, and D. De Zutter, "An efficient perfectly matched layer based multilevel fast multipole algorithm for large planar microwave structures," *IEEE Trans. Antennas Propag.*, vol. 54, no. 5, pp. 1538–1548, May 2006.

[5] V. Okhmatovski, M. Yuan, I. Jeffrey, and R. Phelps, "A three-dimensional precorrected FFT algorithm for fast method of moments solutions of the mixed-potential integral equation in layered media," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 12, pp. 3505–3517, Dec. 2009.

[6] K. Yang and A. E. Yilmaz, "A three-dimensional adaptive integral method for scattering from structures embedded in layered media," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1130–1139, Apr. 2012.

[7] W. Hackbusch, "A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.

[8] W. Chai and D. Jiao, "$\mathcal{H}$- and $\mathcal{H}^2$-matrix-based fast integral-equation solvers for large-scale electromagnetic analysis," *IET Microw., Antennas Propag.*, vol. 4, no. 10, pp. 1583–1596, Oct. 2010.

[9] M. Bebendorf, "Approximation of boundary element matrices," *Numerische Math.*, vol. 86, no. 4, pp. 565–589, 2000.

[10] 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. Electromagn. Compat.*, vol. 47, no. 4, pp. 763–773, Nov. 2005.

[11] S. Börm and L. Grasedyck, "Hybrid cross approximation of integral operators," *Numerische Mathematik*, vol. 101, no. 2, pp. 221–249, Aug. 2005.

[12] Y. Zhao, D. Jiao, and J. Mao, "Hybrid cross approximation for electric field integral equation based scattering analysis," in *Proc. IEEE Int. Symp. Antennas Propag.*, Jul. 2017, pp. 721–722.

[13] W. Chai and D. Jiao, "Fast $\mathcal{H}$-matrix-based direct integral equation solver with reduced computational cost for large-scale interconnect extraction," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 3, no. 2, pp. 289–298, Feb. 2013.

[14] W. Chai and D. Jiao, "An $\mathcal{H}^2$-matrix-based integral-equation solver of reduced complexity and controlled accuracy for solving electrodynamic problems," *IEEE Trans. Antennas Propag.*, vol. 57, no. 10, pp. 3147–3159, Oct. 2009.

[15] W. Chai and D. Jiao, "Linear-complexity direct and iterative integral equation solvers accelerated by a new rank-minimized $\mathcal{H}^2$-representation for large-scale 3-D interconnect extraction," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 8, pp. 2792–2805, Aug. 2013.

[16] S. Börm, "Construction of data-sparse H$^2$-matrices by hierarchical compression," *SIAM J. Sci. Comput.*, vol. 31, no. 3, pp. 1820–1839, Feb. 2009.

[17] M. Li, M. A. Francavilla, F. Vipiana, G. Vecchi, and R. Chen, "Nested equivalent source approximation for the modeling of multiscale structures," *IEEE Trans. Antennas Propag.*, vol. 62, no. 7, pp. 3664–3678, Jul. 2014.

[18] M. A. E. Bautista, M. A. Francavilla, P.-G. Martinsson, and F. Vipiana, "$O(N)$ nested skeletonization scheme for the analysis of multiscale structures using the method of moments," *IEEE J. Multiscale Multiphys. Comput. Tech.*, vol. 1, pp. 139–150, 2016.

[19] M. Bebendorf and R. Venn, "Constructing nested bases approximations from the entries of non-local operators," *Numer. Math.*, vol. 121, no. 4, pp. 609–635, Aug. 2012.

[20] S. Börm, L. Grasedyck, and W. Hackbusch, *Hierarchical Matrices* (Lecture Notes in Computational Science and Engineering), vol. 21, 2006.

[21] S. Rjasanow, "Adaptive cross approximation of dense matrices," in *Proc. Int. Assoc. Boundary Element Methods Conf. (IABEM)*, May 2002, pp. 28–30.

[22] W. Chai and D. Jiao, "Theoretical study on the rank of integral operators for broadband electromagnetic modeling from static to electrodynamic frequencies," *IEEE Trans. Compon., Packag. Manuf. Technol.*, vol. 3, no. 12, pp. 2113–2126, Dec. 2013.

[23] W. Chai and D. Jiao, "Direct matrix solution of linear complexity for surface integral-equation-based impedance extraction of complicated 3-D structures," *Proc. IEEE*, vol. 101, no. 2, pp. 372–388, Feb. 2013.

[24] H. Du, "Mie-scattering calculation," *Appl. Opt.*, vol. 43, no. 9, pp. 1951–1956, Mar. 2004.

[25] S. Yan, V. Sarin, and W. Shi, "Sparse transformations and preconditioners for 3-D capacitance extraction," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 24, no. 9, pp. 1420–1426, Sep. 2005.

[26] Y. Zhao and J. Mao, "Equivalent surface impedance-based mixed potential integral equation accelerated by optimized $\mathcal{H}$-matrix for 3-D interconnects," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 1, pp. 22–34, Jan. 2018.

[27] *Koch Snowflake*. [Online]. Available: https://en.wikipedia.org/wiki/Koch_snowflake

**Yu Zhao** (S'14) received the B.S. degree in electronic information engineering from Xidian University, Xi'an, China, in 2013. He is currently pursuing the Ph.D. degree in electronic engineering at the Key Laboratory of Ministry of Education of Design and Electromagnetic Compatibility of High-Speed Electronic Systems, Shanghai Jiao Tong University, Shanghai, China.

He was a Visiting Student with the On-Chip Electromagnetics Group, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. His current research interests include computational and applied electromagnetics, modeling of interconnects and on-chip passive components, fast, and high-capacity numerical methods.

**Dan Jiao** (M'02–SM'06–F'16) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana–Champaign, Urbana, USA, in 2001.

Until 2005, she was a Senior CAD Engineer, a Staff Engineer, and a Senior Staff Engineer with the Technology Computer-Aided Design (CAD) Division, Intel Corporation. In 2005, she joined the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, as an Assistant Professor, where she is currently a Professor. Since 2013, she has been a University Faculty Scholar with Purdue University. She has authored more than 300 papers in refereed journals and international conferences. Her current research interests include computational electromagnetics, high-frequency digital, analog, mixed-signal, and RF integrated circuit (IC) design and analysis, high-performance very large-scale integration (VLSI) CAD, modeling of microscale and nanoscale circuits, applied electromagnetics, fast and high-capacity numerical methods, fast time-domain analysis, scattering and antenna analysis, RF, microwave, millimeter-wave circuits, wireless communication, and bioelectromagnetics.

Dr. Jiao was a recipient of the 2013 S. A. Schelkunoff Prize Paper Award of the IEEE Antennas and Propagation (AP) Society, which recognizes the best paper published in the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION during the previous year, the 2010 Ruth and Joel Spira Outstanding Teaching Award, the 2008 National Science Foundation (NSF) CAREER Award, the 2006 Jack and Cathie Kozik Faculty Startup Award (which recognizes an outstanding new faculty member of the School of Electrical and Computer Engineering, Purdue University), the 2006 Office of Naval Research (ONR) Award under the Young Investigator Program, the 2004 Best Paper Award presented at the Intel Corporation's Annual Corporate-Wide Technology Conference (Design and Test Technology Conference) for her work on generic broadband models of high-speed circuits, the 2003 Intel Corporation's Logic Technology Development (LTD) Divisional Achievement Award, the Intel Corporation's Technology CAD Divisional Achievement Award, the 2002 Intel Corporation's Components Research the Intel Hero Award, the Intel Corporation's LTD Team Quality Award, and the 2000 Raj Mittra Outstanding Research Award presented by the University of Illinois at Urbana–Champaign. She was among the 21 female faculty selected across the country as 2014–2015 Fellow of the Executive Leadership in Academic Technology and Engineering (ELATE) at Drexel, a national leadership program for women in the academic STEM fields. She was among the 85 engineers selected throughout the nation for the National Academy of Engineering's 2011 U.S. Frontiers of Engineering Symposium. She has been the Chair of Women in Engineering of the IEEE AP Society since 2018. She is currently the Chair of the Paper Awards Committee of the IEEE AP Society. She is the General Chair of the 2019 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO). She is an Associate Editor of the IEEE JOURNAL ON MULTISCALE AND MULTIPHYSICS COMPUTATIONAL TECHNIQUES. She has served as a Reviewer for many IEEE journals and conferences.

**Junfa Mao** (M'92–SM'98–F'12) was born in 1965. He received the B.S. degree in radiation physics from the National University of Defense Technology, Changsha, China, in 1985, the M.S. degree in experimental nuclear physics from the Shanghai Institute of Nuclear Research, Chinese Academy of Sciences, Beijing, China, in 1988, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 1992.

Since 1992, he has been a faculty member with Shanghai Jiao Tong University, where he is currently a Chair Professor and the Vice President. From 1994 to 1995, he was a Visiting Scholar with The Chinese University of Hong Kong, Hong Kong. From 1995 to 1996, he was a Post-Doctoral Researcher with the University of California at Berkeley, Berkeley, CA, USA. He has authored or coauthored more than 500 papers (including more than 130 IEEE journal papers). His current research interests include the interconnect and package problem of integrated circuits and systems and the analysis and design of microwave components and circuits.

Dr. Mao was a member of the IEEE Microwave Theory and Techniques Society Fellow Evaluation Committee from 2012 to 2014 and the IEEE Fellow Committee Member in 2015. He is a member of the Chinese Academy of Sciences. He is a Fellow of the China Institute of Electronics (CIE). He was a recipient of the National Natural Science Award of China in 2004, the National Technology Invention Award of China in 2008, the National Science and Technology Advancement Award of China in 2012, six Best Paper Awards of international conferences, and the National Awards of Teaching (twice) in 2005 and 2018. He was the Chair of the IEEE Shanghai Section from 2007 to 2009 and the IEEE MTT-S Shanghai Chapter from 2009 to 2019. He is the Chief Scientist of the National Basic Research Program (973 Program), China, the Project Leader of the Natural Science Foundation of China for Creative Research Groups, and the Cheung Kong Scholar of the Ministry of Education, China. He is the Director of the Microwave Society of CIE. He was the founder of the IEEE Shanghai Section.