A Complexity-Reduced H-Matrix Based Direct Integral Equation Solver with Prescribed Accuracy for Large-Scale Electrodynamic Analysis

Wenwen Chai  
Electrical and Computer Engineering, Purdue University, wchai@purdue.edu

Dan Jiao  
Purdue University - Main Campus Electrical and Computer Engineering, Purdue University, djiao@purdue.edu

Follow this and additional works at: http://docs.lib.purdue.edu/ecetr
A Complexity-Reduced $\mathcal{H}$-Matrix Based Direct Integral Equation Solver with Prescribed Accuracy for Large-Scale Electrodynamic Analysis

Wenwen Chai
Dan Jiao

TR-ECE-11-04
February 15, 2011

School of Electrical and Computer Engineering
1285 Electrical Engineering Building
Purdue University
West Lafayette, IN 47907-1285
A Complexity-Reduced $\mathcal{H}$-Matrix Based Direct Integral Equation Solver with Prescribed Accuracy for Large-Scale Electrodynamic Analysis

Wenwen Chai and Dan Jiao

School of Electrical and Computer Engineering
465 Northwestern Ave.
Purdue University
West Lafayette, IN 47907-2035

-- This work was supported by NSF under award No. 0747578 and No. 0702567.
Abstract

In this paper, we develop an $\mathcal{H}$-matrix-based fast direct integral equation solver that has a significantly reduced computational complexity, with prescribed accuracy satisfied, to solve large-scale electrodynamic problems. In light of the fact that the cost of an $\mathcal{H}$-matrix-based computation of high-frequency problems is not only determined by the block rank that increases with electric size, but also determined by the $\mathcal{H}$-matrix partition, we propose a new parameter, average partition rank $k_{\text{ave}}$, to derive the storage units and operation counts of the $\mathcal{H}$-matrix based computation of electrodynamic problems. Different from block rank, the partition rank $k_{\text{ave}}$ contains the information of the $\mathcal{H}$-matrix partition. We show that the computational cost of an $\mathcal{H}$-matrix-based computation of electrodynamic problems can be significantly reduced without sacrificing accuracy, by minimizing the rank of each admissible block based on accuracy requirements; and by optimizing the $\mathcal{H}$-partition to reduce the number of admissible blocks at each tree level for a prescribed accuracy and for each frequency point. To minimize the rank for a given accuracy, we develop an efficient matrix algebra based method to determine the minimal rank for each admissible block. The algebraic method has a linear complexity, and hence the computational overhead is negligible. To minimize the number of admissible blocks at each tree level for a given accuracy, we develop a new $\mathcal{H}$-partition method that is frequency dependent, and also controlled by accuracy requirements. With the proposed cost reduction methods, we develop a fast LU factorization for directly solving the dense system matrix resulting from an IE-based analysis of large-scale electrodynamic problems. The proposed solver successfully factorizes dense matrices that involve more than 1 million unknowns associated with electrodynamic problems of 96 wavelengths in fast CPU time, modest memory consumption, and with the prescribed accuracy satisfied. As an algebraic method, the underlying fast technique is kernel independent.
A Complexity-Reduced $\mathcal{H}$-Matrix Based Direct Integral Equation Solver with Prescribed Accuracy for Large-Scale Electrodynamic Analysis

Wenwen Chai, and Dan Jiao, Senior Member, IEEE

Abstract—In this paper, we develop an $\mathcal{H}$-matrix-based fast direct integral equation solver that has a significantly reduced computational complexity, with prescribed accuracy satisfied, to solve large-scale electrodynamic problems. In light of the fact that the cost of an $\mathcal{H}$-matrix-based computation of high-frequency problems is not only determined by the block rank that increases with electric size, but also determined by the $\mathcal{H}$-matrix partition, we propose a new parameter, average partition rank $k_{\text{ave}}$, to derive the storage units and operation counts of the $\mathcal{H}$-matrix-based computation of electrodynamic problems. Different from block rank, the partition rank $k_{\text{ave}}$ contains the information of the $\mathcal{H}$-matrix partition. We show that the computational cost of an $\mathcal{H}$-matrix-based computation of electrodynamic problems can be significantly reduced without sacrificing accuracy, by minimizing the rank of each admissible block based on accuracy requirements; and by optimizing the $\mathcal{H}$-partition to reduce the number of admissible blocks at each tree level for a prescribed accuracy and for each frequency point. To minimize the rank for a given accuracy, we develop an efficient matrix algebra based method to determine the minimal rank for each admissible block. The algebraic method has a linear complexity, and hence the computational overhead is negligible. To minimize the number of admissible blocks at each tree level for a given accuracy, we develop a new $\mathcal{H}$-partition method that is frequency dependent, and also controlled by accuracy requirements. With the proposed cost reduction methods, we develop a fast LU factorization for directly solving the dense system matrix resulting from an IE-based analysis of large-scale electrodynamic problems. The proposed solver successfully factorizes dense matrices that involve more than 1 million unknowns associated with electrodynamic problems of 96 wavelengths in fast CPU time, modest memory consumption, and with the prescribed accuracy satisfied. As an algebraic method, the underlying fast technique is kernel independent.

Index Terms—Integral-equation-based methods, electromagnetic analysis, direct solution, $\mathcal{H}$ matrix, large-scale analysis

I. INTRODUCTION

The Integral equation (IE) based computational electromagnetic methods generally lead to dense systems of linear equations. When a direct method is used, the operation count is proportional to $O(N^3)$ and the memory requirement is proportional to $O(N^2)$, with $N$ being the matrix size. When an iterative solver is used, the memory requirement remains the same, and the computing time is proportional to $O(N_{\text{it}}N^2)$, where $N_{\text{it}}$ denotes the total number of iterations required to reach convergence. In recent years, fast solvers such as fast multipole based methods [1]–[3], fast low-rank compression methods [4]–[7], and FFT-based methods [8], [9] have been developed that dramatically reduce the memory requirement of the iterative IE solvers to $O(N)$, and the CPU time to $O(N_{\text{it}}N \log N)$ for electrodynamic problems. This represents an impressive improvement as compared with conventional $O(N^3)$ or $O(N_{\text{it}}N^2)$ techniques.

Fast direct solvers have also been developed for electrodynamic problems. Most recent work can be seen in [10], [11]. LU factorization of $O(N^2)$ time complexity and $O(N^{1.5})$ memory complexity has been reported. Compared to iterative solvers, direct solvers have advantages when the number of iterations is large or the number of right hand sides is large. For example, if there exist $N$ right hand sides, each of which costs $O(N_{\text{it}}N \log N)$ operations, the total cost of the iterative solver will be $O(N_{\text{it}}N^2 \log N)$, which is expensive.

Recently, the $\mathcal{H}$- and $\mathcal{H}^2$-matrix based mathematical framework has been introduced and further developed to reduce the computational complexity of iterative IE-based solvers for electrodynamic problems [12], [13]. It is shown that given a wide range of electric sizes which lead to a wide range of $N$, the dense system of $O(N^2)$ parameters can be compactly stored in $O(N)$ units, and the dense matrix-vector multiplication can be performed in $O(N)$ operations. Moreover, the same order of accuracy can be kept across this range.

In this work, under the mathematical framework of the $\mathcal{H}$ matrix, we develop a fast direct IE solver that has a significantly reduced computational complexity, with the prescribed accuracy satisfied, to solve large-scale electrodynamic problems. The $\mathcal{H}$ (hierarchical)-matrix is a general mathematical framework [14]–[17], which enables a highly compact representation and efficient numerical computation of dense matrices. To be specific, if matrix $C$ is an $m \times n$ off-diagonal block in an $\mathcal{H}$ matrix that describes a far-field interaction specified by a so-called admissibility condition [17], it can be written as $C = AB^T$, where $A$ is of dimension $m \times k$, $B$ is of dimension $n \times k$, and $k$ denotes the rank of $C$ with $k < m$ and $k < n$. Storage requirements and matrix-vector multiplications using $\mathcal{H}$-matrices have been shown to be of complexity $O(N \log N)$. From a mathematical point of view, existing low-rank compression based IE methods developed for electromagnetic analysis can be viewed in the framework of $\mathcal{H}$ matrices, although their technical details could be very different and many of them predated the literature of $\mathcal{H}$ matr-
ces. For example, the direct integral equation solver reported in [10] can be viewed as an \( \mathcal{H} \)-based direct integral equation solver, although \( \mathcal{H} \)-based fast arithmetics was not employed. It successfully solves electrically large integral equations for problem sizes to 1 M unknowns. It is also worth mentioning that the matrices underlying generic Fast Multipole algorithms are \( \mathcal{H}^2 \)-matrices, as noted in [19].

In the literature of \( \mathcal{H} \) matrices, it is shown that matrix-matrix multiplications and matrix inversions using \( \mathcal{H} \)-matrices are of complexity \( O(N \log^2 N) \) [14]–[17], which is very efficient. However, the actual complexity of an \( \mathcal{H} \)-matrix based method is associated with the rank \( k \) of the off-diagonal blocks. For static problems, a constant \( k \) can be used for all problem sizes without sacrificing accuracy, and hence ignored in complexity analysis. For electromagnetic problems, however, the \( k \) required to achieve a given accuracy typically increases with the electric size [12], [21]. Since the rank \( k \) becomes a variable in electromagnetic analysis, the existing complexity analysis developed for \( \mathcal{H} \)-matrix based arithmetics [17], which is based on a constant rank, becomes not proper for analyzing electromagnetic problems. To be specific, if one uses the maximum rank of all the admissible blocks to bound the complexity, the complexity is overestimated. More importantly, the complexity of an \( \mathcal{H} \)-matrix based computation of electromagnetic problems is highly dependent on the \( \mathcal{H} \)-matrix partition, i.e., the number of admissible blocks and the row/column dimension of each block etc. This is an important factor that received little attention in previous research. Existing \( \mathcal{H} \)-matrix partition is based on a geometry based admissibility condition. The resultant partition is by no means optimal especially for electromagnetic problems.

In this work, we develop new bounds of the computational cost for the \( \mathcal{H} \)-matrix-based computation of electromagnetic problems. In light of the fact that the complexity of an \( \mathcal{H} \)-matrix based computation of electromagnetic problems is determined by both block rank (the rank of each admissible block) and \( \mathcal{H} \)-matrix partition, we propose a new parameter, average partition rank \( k_{ave} \), to bound the storage units and operation counts of an \( \mathcal{H} \)-matrix based computation. Different from block rank, the partition rank \( k_{ave} \) contains the information of the \( \mathcal{H} \)-matrix partition.

With the new bounds of the computational cost derived for electromagnetic problems, it becomes clear how to reduce the complexity of an \( \mathcal{H} \)-matrix based computation for solving high-frequency problems. We propose two methods. One is to minimize the rank of each admissible block based on accuracy requirements. Although the block rank is observed to be proportional to the electric size of the block diameter, the actual number should be determined and minimized based on accuracy requirements. We determine the minimal rank for each admissible block by an efficient matrix algebra based method. The algebraic method has a linear complexity, and hence the computational overhead is negligible. The other method for reducing the computational cost is to optimize the \( \mathcal{H} \)-partition to reduce the number of admissible blocks at each tree level based on accuracy requirements for each frequency point. We show that the admissibility condition used in the conventional \( \mathcal{H} \)-partition is empirical instead of theoretical, and hence not optimal for a given accuracy and a given frequency. We hence develop a new \( \mathcal{H} \)-partition method that is frequency dependent, and also controlled by accuracy requirements. With the proposed new partition, the number of admissible blocks at each tree level is significantly reduced compared to that generated by the admissibility condition based, i.e., geometry based partition. The methods we developed for minimizing the rank and optimizing the \( \mathcal{H} \)-partition not only can be used in the proposed solver, but also can be used in other fast integral equation solvers to speed up their computation.

Moreover, we developed an efficient LU-factorization for directly solving the dense system matrix resulting from an \( \mathcal{H} \)-based analysis of large-scale electromagnetic problems. The operation counts of the proposed direct solver are \( O((k_{ave}C_{sp})^2N \log^2 N) \) in LU factorization, \( O((k_{ave}C_{sp})N \log N) \) in LU solution, and the storage requirement is \( O(k_{ave}C_{sp}N \log N) \), all of which were derived theoretically and also verified numerically. The \( C_{sp} \) is the maximal number of blocks that can be formed by a cluster in a block cluster tree. Although both \( k_{ave} \) and \( C_{sp} \) depend on \( N \) in electromagnetic analysis, that is why we include these two parameters in the computational cost, they are reduced to small numbers compared to \( N \) by the proposed methods. We also give a detailed implementation of the \( \mathcal{H} \)-based LU factorization, which is not reported elsewhere.

The remainder of this paper is organized as follows. In Section II, we give the background of the \( \mathcal{H} \)-matrix based analysis of electromagnetic problems. In Section III, we propose the new bounds of the computational cost for the \( \mathcal{H} \)-matrix based computation of electromagnetic problems. In Section IV, we propose methods to reduce the complexity of the \( \mathcal{H} \)-matrix based computation. In Section V, we give a numerical proof on the existence of an \( \mathcal{H} \)-matrix based representation of the inverse and LU factors of an IE based system matrix for an electromagnetic problem. In Section VI, we give a number of pseudo-codes to show a detailed implementation of the LU factorization using \( \mathcal{H} \) matrices, which is not reported in the mathematical literature. In Section VII, we analyze computational cost. In Section VIII, we present numerical results to demonstrate the accuracy and efficiency of the proposed direct IE solver. In Section IX, we summarize our findings.

II. BACKGROUND

The \( \mathcal{H} \)-matrix based methods are kernel independent. In the following, we use an electric-field integral equation as an example to illustrate the basic concept of \( \mathcal{H} \)-matrix based methods.

A. Electric Field Integral Equation

We consider the electric-field integral equation (EFIE) [1], [22]

\[
E_i|_{tan} = \int_{S} \left[ j \omega \mu J_S(r) g(r,r') \right]_{tan} dS' - \int_{S} \left[ \frac{j}{\omega \epsilon} (\nabla' \cdot J_S(r')) \nabla' g(r,r') \right]_{tan} dS',
\]

(1)
in which Green’s function \( g(r, r') = \frac{e^{-j|r-r'|}}{|r-r'|} \), \( J_S \) is the induced surface current density, \( \omega \) is angular frequency, \( \kappa \) is wave number, and subscript \( tan \) denotes tangential component.

By expanding the unknown \( J_S \) using the RWG basis functions [22], a Method of Moment based solution of (1) results in the following linear system of equations

\[
\mathbf{GI} = \mathbf{V},
\]

where

\[
G_{mn} = \iint_{S_m} dS \int_{S_n} dS' [j\omega \mu J_m(r) \cdot J_n(r') - \frac{j}{\omega \varepsilon} (\nabla \cdot J_m(r)) (\nabla' \cdot J_n(r'))] g(r, r'),
\]

and

\[
V_m = \int_{S_m} J_m(r) \cdot \mathbf{E}_i dS.
\]

The conventional way to solve (2) could be very expensive, since the entries of \( G \) are all nonzero. In the following section, we introduce the \( \mathcal{H} \) matrix as a data-sparse representation of \( G \), from which a significant reduction in computational complexity can be achieved.

\section*{B. Definition of the \( \mathcal{H} \) Matrix}

An \( \mathcal{H} \) matrix is generally associated with an admissibility condition [17]. To define an admissibility condition, we denote the whole index set containing the indices of the basis functions in the computational domain by \( I = \{1, 2, \cdots, N\} \), where \( N \) is the total number of unknowns. Considering two subsets \( t \) and \( s \) of the \( I \), the admissibility condition is defined as

\[
(t, s) \text{ are admissible} = \begin{cases} 
\text{True} & \text{if} \min\{\text{diam}(Q_t), \text{diam}(Q_s)\} \\
\leq \eta \text{dist}(Q_t, Q_s); & \text{otherwise},
\end{cases}
\]

where, as shown in Fig. 1, \( Q_{t,s} \) is the minimal subset of the space containing the supports of all basis functions belonging to \( t \) or \( s \), \text{diam}(\cdot) \) is the Euclidean diameter of a set, \text{dist}(\cdot) \) is the Euclidean distance of two sets, and \( \eta \) is a positive parameter that can be used to control the admissibility condition. If subsets \( t \) and \( s \) do not satisfy the admissibility condition, they are called inadmissible. The admissibility condition shown in (5) is rather empirical than theoretical. It is controlled by an empirical parameter \( \eta \) instead of a prescribed accuracy. In this work, we will develop a new method to partition a matrix into admissible and inadmissible blocks, which is controlled by accuracy requirements.

In an \( \mathcal{H} \)-matrix representation, an inadmissible block keeps its original full-matrix representation; while an admissible block has a factorized low-rank form. To be specific, an admissible block \( G^{t,s} \) that is formed by subsets \( t \) and \( s \) can be written as a factorized form

\[
G^{t,s} = \mathbf{AB}^T,
\]

where \( G^{t,s} \in \mathbb{C}^{m \times n}, \mathbf{A} \in \mathbb{C}^{m \times k}, \mathbf{B} \in \mathbb{C}^{n \times k}, \text{and} \ k \in \mathbb{N} \) is the rank of \( G^{t,s} \). Here, as long as \( k \) is less than the minimum of \( m \) and \( n \), \( G^{t,s} \) is low rank. The \( k \) is not required to be \( O(1) \).

If all the blocks \( G^{t,s} \) formed by the admissible \((t, s)\) in \( G \) can be represented by a factorized low-rank form shown in (6), \( G \) has an \( \mathcal{H} \)-matrix representation ( [17], p. 18). Clearly, to store admissible \( G^{t,s} \), we only need to store \( \mathbf{A} \) and \( \mathbf{B} \), the cost of which is \( O(k(m + n)) \) in contrast with the original storage that is \( O(mn) \). Similar cost reduction in matrix-vector and matrix-matrix multiplication can be achieved. Therefore, if we are able to represent the dense matrix resulting from an IE based analysis by an \( \mathcal{H} \) matrix, we can reduce the complexity of IE-based solutions significantly. The detail of the \( \mathcal{H} \)-matrix representation of an IE-based system matrix is given in next subsection.

\section*{C. \( \mathcal{H} \)-Matrix Representation of the IE-Based System Matrix}

The essential idea of an \( \mathcal{H} \)-matrix representation is to represent the off-diagonal matrix block that satisfies the admissibility condition by a rank-\( k \) matrix shown in (6).

There are three representative schemes that can be used to generate a rank-\( k \) matrix of an IE-based dense matrix block: interpolation, Taylor expansion, and adaptive cross approximation (ACA). In [12], [13], an interpolation scheme is used to obtain an \( \mathcal{H} \)-representation without any compression cost for electrodynamic kernels. The error bound of the interpolation based \( \mathcal{H} \)-representation was derived. It was shown that exponential convergence with respect to the number of interpolation points can be achieved irrespective of the electric size. Taylor expansion can also be used to locally replace the kernel function by degenerate approximations, as shown in ( [17], p. 10). Its accuracy is controlled by the expansion order. Neither interpolation nor Taylor expansion involves rank compression cost. Another approach is to directly compute a low rank approximation of the original matrix up to a prescribed accuracy. The representative method is ACA [17], [20], which is purely algebraic. In [10], [21], ACA was used to solve electromagnetic problems. For each matrix block, the cost of ACA is linear. A simple flow to obtain the \( \mathcal{H} \)-matrix representation of a given matrix block is shown in Fig. 2.
The complexity of H-matrix based arithmetics has been conducted for kernel functions that do not change with frequencies [17]. Storage requirements and matrix-vector multiplications using H-matrices have been shown to be of complexity \(O(N \log N)\). Moreover, the inverse of an H matrix can be obtained in \(O(N \log^2 N)\) complexity. In this section, we show that for electrodynamic problems, the complexity bounds of H-matrix based computations need to be re-derived. In light of the fact that the cost of an H-matrix-based computation of high-frequency problems is not only determined by the block rank that increases with electric size, but also determined by the H-partition, we propose a new parameter, average partition rank \(k_{ave}\), to bound the computational cost of the H-matrix-based computation of electrodynamic problems. With the new bounds, it becomes clear how to reduce the complexity of an H-matrix based computation for solving high-frequency problems.

A. Proposed definition of average partition rank, \(k_{ave}\), for the H-based computation of electrodynamic problems

For frequency independent kernels, a constant rank \(k\) across all the admissible blocks is sufficient to generate a constant order of accuracy for the H-based computation of the dense system matrix. This can be theoretically verified from the error bound for an H-matrix based representation of a static kernel [17]. For frequency dependent kernels, however, a constant \(k\), in general, cannot guarantee a constant order of accuracy across a wide range of electric sizes. This can be seen from the error bound of an H-matrix-based representation of electrodynamic problems [12], [13]. This can also be seen from ACA-based rank revealing presented in [21]. As a result, the complexity analysis of H-based computations given in the mathematical literature, which is based on a constant rank, is not proper for analyzing electrodynamic problems.

Take the storage complexity of an H-matrix as an example, which is also the complexity of an H-matrix based matrix-vector multiplication. In an H matrix, since each admissible block \(G_{m_i \times n_i}^{m_i \times n_i}\) has a factorized form \(A_{m_i \times k_i}B_{k_i \times n_i}^{k_i \times n_i}\), with rank \(k_i\), the storage is reduced from \(m_i \times n_i\) units to \(k_i(m_i + n_i)\) units. By summing up the storage of all the admissible blocks, we obtain

\[
\text{Storage} = \sum_{i=1}^{nk} k_i(m_i + n_i),
\]

where \(nk\) is the total number of admissible blocks. The above can be evaluated as

\[
\text{Storage} = \sum_{i=0}^{l} \sum_{i=1}^{nk_l} k_i(m_i + n_i),
\]

where \(l\) is tree level, \(p\) is tree depth, and \(nk_l\) is the number of admissible blocks at level \(l\). Clearly, as can be seen from (7), if we use the maximal \(k\) among all the admissible blocks, \(k_{max}\), to bound the computational complexity, we will overestimate the complexity because many admissible blocks have a rank much smaller than \(k_{max}\). To give an example, in Fig. 4, we plot the storage of the blocks that have the largest rank \(k_{max}\) in comparison with that of the rest of the blocks that have a rank smaller than \(k_{max}\), for a cone sphere across a wide range of electric sizes. The electric size ranges from \(\kappa a = 2\) to \(\kappa a = 600\), where \(\kappa\) is wave number and \(a\) is the largest physical dimension of the cone sphere. It is clear that the total storage is not dominated by the cost of storing rank-\(k_{max}\) blocks. In fact, the storage of the rank-\(k_{max}\) blocks is much less than that of the rest of the blocks. This is because although \(k_{max}\) is large, the number of admissible blocks that have rank \(k_{max}\) is also the smallest.

Furthermore, as can be seen from (8), in addition to the block rank \(k_i\), the number of admissible blocks at each tree level \(nk_l\) also plays an important role in determining the
computational complexity. If \( nk_1 \) is proportional to \( 2^l \), then the advantage of the \( \mathcal{H} \)-based computation of high-frequency problems may not be significant since the block rank \( k_i \) is observed to be proportional to the electric size of the block diameter. However, in reality, \( nk_1 \) is not proportional to \( 2^l \). This will become clear in the following subsections. In addition, there exists a big space to optimize the \( \mathcal{H} \)-partition to reduce \( nk_1 \) without sacrificing accuracy.

From the aforementioned analysis, the complexity of an \( \mathcal{H} \)-based computation of high-frequency problems is determined by the block rank \( k_i \) as well as the number of admissible blocks at each tree level and the row/column dimension of each admissible block, i.e. \( \mathcal{H} \)-partition. In light of this fact, we propose a new parameter, average partition rank \( k_{ave} \), to derive the computational cost of an \( \mathcal{H} \)-based computation of high-frequency problems. We define average partition rank \( k_{ave} \) as the following:

\[
k_{ave} = \frac{\sum_{i=1}^{nk} k_i (m_i + n_i)}{\sum_{i=1}^{nk} (m_i + n_i)}\]

where \( nk \) is the total number of admissible blocks, \( k_i \) is the rank of the \( i \)-th admissible block, \( m_i \) and \( n_i \) are the number of rows and columns of the admissible block. From the above definition, it is clear that \( k_{ave} \) is not only related to the block rank \( k_i \), but also related to the \( \mathcal{H} \)-partition. Different partitions can result in different \( k_{ave} \). That is why we name \( k_{ave} \) as average partition rank to distinguish it from the block rank that does not contain any information about partition.

**B. Proposed bounds of storage units and operation counts for the \( \mathcal{H} \)-based computation of electrodynamic problems**

By using (9), the storage complexity (7) can be written as

\[
\text{Storage} = k_{ave} \sum_{i=1}^{nk} (m_i + n_i),
\]

The summation of the \( m_i \) and \( n_i \) over all the admissible blocks can be evaluated as the following:

\[
\sum_{i=1}^{nk} (m_i + n_i) \leq \sum_{i=0}^{p} \sum_{l=0}^{nk} (\frac{N}{2^l} \times 2) = \sum_{i=0}^{p} nk_1 (\frac{N}{2^l} \times 2),
\]

in which we use the fact that the row/column dimension of a block at level \( l \) is \( \frac{N}{2^l} \), where \( l = 0 \) represents the root level. The above should be evaluated based on actual \( nk_1 \), which is problem dependent and partition dependent. Here, we use the fact that \( nk_1 \) is less than \( 2^l C_{sp} \) to bound (11), substituting (11) into (10), we thus obtain

\[
\text{Storage} = O(k_{ave} C_{sp} N \log N).
\]

Therefore, we can use the average partition rank defined in (9) to bound the storage units.

Similarly, we can use an average square partition rank defined below to bound the operation counts of the \( \mathcal{H} \)-based computation.

\[
(k^2)_{ave} = \frac{\sum_{i=1}^{nk} k_i^2 (m_i + n_i)}{\sum_{i=1}^{nk} (m_i + n_i)}.
\]

Take the matrix-matrix multiplication as an example, the cost for each admissible block involved in the multiplication is \( C_{sp} k_i^2 (m_i + n_i) \) ([17], pp. 127-130). By summing up the cost of all the admissible blocks across all the tree levels, we
From (13) and (9), it can be seen that the $p$ where $k=m+n$ for a 3D conducting plate from 2 to 30 wavelengths. It can thus obtain $k$ value of $k$ on $N$ $k$ subsections, we present methods for reducing $k$ problems, we have to reduce $k$ computational cost of an $H$-matrix representation, so that we can still use $k$ for the given $H$ partition is also minimized.

Given an accuracy requirement, singular value decomposition (SVD) is the most accurate method to obtain the minimum rank that can meet the accuracy requirement for an admissible block. However, if we directly apply SVD to the original full matrix to obtain its $H$-matrix representation, although the resultant rank is minimal, the computational cost is high.

On the other hand, we can use interpolation, Taylor expansion, and ACA based approaches to efficiently convert a full matrix block to an $H$-matrix representation. However, the resultant rank is, in general, not the minimal one that is necessary to satisfy the accuracy requirement. In other words, the resultant rank can be much larger than what is necessary to satisfy a prescribed accuracy.

In this paper, we first use ACA+ ([17], pp. 71-74), which is a variant of ACA, to efficiently compute an $H$-matrix representation. We then apply SVD to the $H$-matrix representation to determine the actual rank that is needed to satisfy the accuracy requirement. By doing so, we keep the advantages of both SVD and ACA-based methods. The resultant rank is minimal, and meanwhile it is obtained in linear complexity for each block. In the following, we give more details of the proposed approach.

First, we use ACA+ to numerically obtain a factorized form of an admissible block. Conventional ACA fails in some examples [20]. ACA+ does not have such a problem. In addition, ACA+ involves less storage and computational cost than ACA. The detailed procedure of ACA+ is very similar to that of the conventional ACA. The difference between them is as follows. At the beginning of an ACA+ algorithm, a reference row and a reference column has to be chosen. After the ACA+ is completed, we obtain a factorized form $A_{m,k}B_{n,k}^T$. For such a factorized form, SVD can be efficiently performed by a reduced SVD ([17], p. 108). The resultant computational cost is $O(k^2(m+n))$, which is linear.

To test the effectiveness of the proposed approach to determining the minimal rank of an admissible block, we prove that $A_{m,k}$ is a minimal rank, the resultant average partition rank $k_{ave}$ is based on a prescribed accuracy for a given $H$-partition.

Given an $H$-partition, to minimize $k_{ave}$, for each admissible block, we determine a minimal rank based on a prescribed accuracy. The reason is obvious. If each admissible block has a minimal rank, the resultant average partition rank $k_{ave}$ for the given $H$ partition is also minimized.

IV. PROPOSED METHODS FOR REDUCING THE COMPUTATIONAL COST OF $H$-MATRIX BASED DIRECT SOLUTION OF ELECTRODYNAMIC PROBLEMS

From (12) and (15), it can be seen that to reduce the computational cost of an $H$-based computation of electrodynamic problems, we have to reduce $k_{ave}C_{sp}$. In the following two subsections, we present methods for reducing $k_{ave}C_{sp}$.
simulated a 3D conducting plate of 10 wavelengths, and a 3D conducting sphere of 8 wavelengths respectively. The ϵ in ACA+ and SVD was set to be 10^{-4}. In Fig. 6, we plot the rank distributions with the proposed scheme (ACA+ and SVD) and with ACA+ only in the lowest \( \mathcal{H} \)-partition level that has an admissible block. This level is the closest to the root of a block cluster tree, and hence the admissible blocks therein have the largest electric size. The horizontal axis of Fig. 6 is the admissible block index. It can be seen from Fig. 6 that by using the proposed method, the rank in most admissible blocks is reduced by half. Moreover, the same accuracy is achieved. The accuracy is measured by \( \| G - \tilde{G} \| / \| G \| \), where a Frobenius norm is used. The accuracy is 1.021689 \times 10^{-5} without the proposed rank minimization, and 1.079537 \times 10^{-5} with the proposed minimization for the plate example. Clearly, the rank is reduced without sacrificing accuracy. The same is true for the sphere example. The accuracy is 1.931210 \times 10^{-5} without the proposed rank minimization, and 1.996355 \times 10^{-5} with the proposed minimization. In addition, with the proposed method, the new rank distribution becomes more uniform across all the admissible blocks as can be seen from Fig. 6.

**B. Proposed method for optimizing \( \mathcal{H} \)-matrix partition to reduce \( k_{ave} C_{sp} \)**

Existing \( \mathcal{H} \)-matrix partition is based on the admissibility condition as shown in (5). It is not an optimal one for frequency dependent kernels. This can also be understood from the fact that the admissibility condition (5) is frequency independent. Physically speaking, whether the interaction between two regions can be represented by a low-rank block or not is frequency dependent. For example, for a given accuracy, it is possible that an off-diagonal block that is inadmissible at certain frequency becomes admissible at a lower frequency. It is also possible that multiple small admissible blocks can be merged into a single admissible block with prescribed accuracy. In other words, they start to become admissible in a lower level of an inverted block cluster tree when frequency changes. However, the admissibility condition given in (5) is empirical instead of theoretical. It is controlled by an empirical parameter \( \eta \), instead of a prescribed accuracy.

In the following, we show our proposed algorithm that can optimize an \( \mathcal{H} \)-matrix partition based on a prescribed accuracy. The resultant \( \mathcal{H} \)-matrix partition is frequency dependent. It significantly reduces the number of admissible blocks at each tree level, with the prescribed accuracy satisfied.

The proposed \( \mathcal{H} \)-matrix partition algorithm for a prescribed accuracy \( \epsilon_{opt} \) is shown in (17). A new error tolerance \( \epsilon_{opt} \) is introduced instead of reusing \( \epsilon \) in (16) to facilitate separated accuracy control of the \( \mathcal{H} \)-partition optimization and the rank minimization for each admissible block. In (17), the original partition given by (5) is used as an initial guess from which we construct an optimized partition.

### \( \mathcal{H} \)-Partition Optimization

**Procedure** \( \mathcal{H} - \mathcal{P}_{opt}(\mathcal{P}, \epsilon_{opt}) \)

(Input \( \mathcal{P} \) is the original \( \mathcal{H} \) partition,
output \( \mathcal{P} \) is overwritten by an optimized \( \mathcal{H} \) partition)

If \( \mathcal{P} \) is a non-leaf off-diagonal matrix block

\[
\text{for } (i = 0; i < 4; i + +) \\
\text{if } \mathcal{P}(i) \text{ is an inadmissible block} \\
\text{Rk\_Factor}(\mathcal{P}(i), \epsilon_{opt}) \\
\text{end if} \\
\text{if } \mathcal{P}(i) \text{ is a non-leaf block} \\
\mathcal{H} - \mathcal{P}_{opt}(\mathcal{P}(i), \epsilon_{opt}) \\
\text{end if} \\
\text{end for} \\
\text{If all blocks in } \mathcal{P} \text{ are admissible blocks} \\
\text{Merge\_Rkblocks}(\mathcal{P}, \epsilon_{opt}) \\
\text{end if} \\
\text{end if} \]

(17)

In (17), the function \text{Rk\_Factor} is to factorize a full-matrix block to a rank-\( k \) matrix shown in (6) based on a prescribed accuracy. The procedure is the combined ACA+
with SVD, which is detailed in section IV-A. The function \texttt{Merge\_Rkblocks} is to merge multiple small admissible blocks to a single one based on the prescribed accuracy. To give an example, four admissible sub-blocks can be merged into one admissible block as follows.

\[
\begin{bmatrix}
A_1(f_1)B_1(f_1)^T & A_2(f_2)B_2(f_2)^T \\
A_3(f_3)B_3(f_3)^T & A_4(f_4)B_4(f_4)^T
\end{bmatrix} =
\begin{bmatrix}
A_1(f_1) \\
0
\end{bmatrix}
\begin{bmatrix}
B_1(f_1) \\
0
\end{bmatrix}^T +
\begin{bmatrix}
A_2(f_2) \\
0
\end{bmatrix}
\begin{bmatrix}
B_2(f_2) \\
0
\end{bmatrix}^T +
\begin{bmatrix}
A_3(f_3) \\
0
\end{bmatrix}
\begin{bmatrix}
B_3(f_3) \\
0
\end{bmatrix}^T +
\begin{bmatrix}
A_4(f_4) \\
0
\end{bmatrix}
\begin{bmatrix}
B_4(f_4) \\
0
\end{bmatrix}^T
\]

\[
\tilde{A}_1(f_1)\tilde{B}_1(f_1)^T + \tilde{A}_2(f_2)\tilde{B}_2(f_2)^T + \tilde{A}_3(f_3)\tilde{B}_3(f_3)^T + \tilde{A}_4(f_4)\tilde{B}_4(f_4)^T = AB^T,
\]

where the \(f_i (i = 1, 2, 3, 4)\) denotes the electric size the blocks are associated with, and the addition in the final step is carried out by the truncated addition operation in [17], p. 110, with the new rank \(k\) determined based on the accuracy requirement \(\epsilon_{opt}\). Different from static problems, in electrodynamic problems, when the size of the admissible block increases, its rank also increases in general. Consequently, although each merging operation reduces four admissible blocks to one block, it also increases the rank of the resultant block. Therefore, we should check which one is computationally more efficient: merging or not merging. This can be assessed by comparing the storage requirement of the merged block with that of the four children admissible blocks. If the former is less than the latter, we perform merging; otherwise, we do not perform merging, instead we keep the original four blocks. To be more specific, we check whether \(k_{merge} (m_{merge} + n_{merge}) \leq \sum_{j=1}^{4} k_j (m_j + n_j)\) is satisfied or not, where \(k_{merge}\) is the rank of the big block resulting from the merging operation, \(m_{merge}\) and \(n_{merge}\) is the row(column) dimension of the block, and \(k_j\) is the rank of the children admissible blocks. If it is satisfied, we merge blocks based on accuracy requirements; if not, we keep the original blocks.

In (17), we only perform two basic operations: making an inadmissible block in the off-diagonal part admissible based on a prescribed accuracy via function \texttt{Rk\_Factor}, or merging small admissible blocks to be a big admissible block based on the accuracy requirement through function \texttt{Merge\_Rkblocks}. Both operations do not increase the number of inadmissible blocks. In fact, the number of inadmissible blocks is even reduced due to the first operation. Therefore, given an \(\mathcal{H}\)-partition, the proposed optimization algorithm does not increase the number of inadmissible blocks. If the number of admissible blocks is reduced, the total number of blocks is also reduced.

To validate the effectiveness of the proposed \(\mathcal{H}\)-partition optimization algorithm, we simulated a conducting plate from 2\(\lambda\) to 60\(\lambda\), the number of unknowns of which is from 1,160 to 1,078,800. The \(\epsilon_{opt}\) was chosen as \(10^{-3}\). In Fig. 7, we plot the maximum number of admissible blocks that can be formed by one cluster in a block cluster tree \(C_{ad}\) in the original \(\mathcal{H}\)-partition, and that in the optimized \(\mathcal{H}\)-partition. Clearly, the \(C_{ad}\) is reduced significantly. Since the proposed \(\mathcal{H}\)-partition optimization does not increase the number of inadmissible blocks as analyzed above, the \(C_{sp}\) is also reduced significantly.

When the \(\mathcal{H}\)-partition changes, the average partition rank \(k_{ave}\) also changes. Thus, in addition to \(C_{sp}\), we need to examine the product of \(k_{ave}\) and \(C_{sp}\) to assess the success of the proposed \(\mathcal{H}\)-partition optimization algorithm, since it is \(k_{ave}C_{sp}\) that determines the storage and time complexity as can be seen from (12) and (15). In Fig. 8(a), we plot \(k_{ave}C_{ad}\) for the plate example simulated above with respect to \(\kappa a\), where \(\kappa\) is the wave number, and \(a\) is the side length of the plate. We compare the \(k_{ave}C_{ad}\) generated by the proposed method, and that generated by the conventional method that is based on an ACA-based rank scheme and an admissibility condition based \(\mathcal{H}\)-partition. It is clear that the proposed method greatly reduces \(k_{ave}C_{ad}\). In Fig. 8(b), we plot \(k_{ave}C_{ad}\) for a sphere of diameter \(a\) with respect to \(\kappa a\) from 2\(\lambda\) to 45\(\lambda\). Again, \(k_{ave}C_{ad}\) is greatly reduced by the proposed method. Since the proposed \(\mathcal{H}\)-partition optimization does not increase the number of inadmissible blocks as analyzed above, the \(k_{ave}C_{sp}\) is also reduced significantly.

Although \(C_{sp}\) is a parameter that can be used to qualitatively measure the number of blocks formed by an \(\mathcal{H}\)-partition, it does not give a quantitative measurement of the number of blocks obtained by the \(\mathcal{H}\)-partition. To provide a quantitative analysis, in Fig. 9(a), we plot the exact number of blocks with respect to tree level generated by the original \(\mathcal{H}\)-partition that is based on the admissibility condition in comparison with
the number of blocks generated by the proposed partition, for a $15\lambda$ sphere. In Fig. 9(b), we plot the same for a $40\lambda$ plate. Clearly, compared to the original partition, the proposed partition significantly reduces the number of blocks at each tree level, and hence significantly reducing the computational cost. In addition, across the entire tree depth, it is observed that the number of blocks is not necessarily reduced by half when one ascends the inverted tree by one level from leaf level $l = p$ to root level $l = 0$, which can be seen from both conventional partition and the proposed partition.

C. Study on the dependence of $k_{ave}$ and $C_{sp}$ of the proposed methods with respect to scatterer shape

We varied the scattering shape continuously from plate, Sierpiski gasket, cylinder, open cone, cone sphere, to sphere. We plotted the maximal rank $k_{max}$ versus electric size for these scatterers. For comparison, we also plotted the average partition rank $k_{ave}$ obtained by the proposed methods for the same accuracy. As can be seen from the figures in the left column of Fig. 10 and Fig. 11, for all these scatterers, $k_{ave}$ is much smaller compared to $k_{max}$. In addition, the rate of the change of $k_{ave}$ with respect to electric size is much slower than that of $k_{max}$. Denoting the electric size by $a/\lambda$, where $a$ is the maximal size of the structure, $k_{max}$ is observed to be $O(a/\lambda)$. However, $k_{ave}$ is not, it is much smaller than $O(a/\lambda)$. Theoretically speaking, this is because $k_{ave}$ is not only determined by the rank of each admissible block, but also determined by the number of admissible blocks at each tree level, and hence the $H$-partition. In addition, compared to $k_{max}$, $k_{ave}$ is less shape dependent.

In the right column of Fig. 10 and Fig. 11, we plot the number of admissible blocks obtained by the proposed method with respect to electric size for a variety of scatter shapes. The dependence of the $C_{sp}$ on the scatter shape is shown to be little. In addition, with proposed methods, both $k_{ave}$ and $C_{sp}$ are minimized to be small compared to $N$.

D. Study on the dependence of $k_{ave}$ and $C_{sp}$ on accuracy requirements

We also tested the dependence of $k_{ave}$ and $C_{sp}$ with respect to accuracy. The results in Fig. 10 and Fig. 11 were generated based on $\epsilon = 10^{-4}$ and $\epsilon_{opt} = 10^{-3}$, where $\epsilon$ is a parameter shown in (16) for controlling the accuracy of rank reduction, and $\epsilon_{opt}$ is a parameter shown in (17) for controlling the accuracy of the $H$-partition optimization. To test the dependence with respect to accuracy, we set $\epsilon = 10^{-2}$ and $\epsilon_{opt} = 10^{-6}$. We used a plate and a sphere as examples, and tested the dependence of $k_{ave}$ and $C_{sp}$ generated by the proposed methods with respect to accuracy across a wide range of electric sizes. As can be seen from Fig. 12, with the accuracy requirement increased, $k_{ave}$ and $\sqrt{(k^2)_{ave}}$ increase. However, the increase is small compared to the three orders of magnitude increase in accuracy. In
addition, the dependence of $k_{ave}$ and $\sqrt{\langle k^2 \rangle_{ave}}$ with respect to electric size is similar to what is observed for a lower order of accuracy. In addition, with the accuracy requirement increased, the number of admissible blocks also increases. Again, the increase is minor compared to the three orders of magnitude increase in accuracy. Moreover, the frequency dependence of $C_{sp}$ is also similar to that for a lower order of accuracy.
Fig. 11. The $k_{ave}$ and $C_{sp}$ generated by the proposed method with respect to electric size for a variety of scatterer shapes (Continued from Fig. 10). (a-b) Cone sphere. (c-d) Sphere.

Fig. 12. The dependence of $k_{ave}$ and $C_{sp}$ generated by the proposed method with respect to accuracy requirements over a wide range of electric sizes (Accuracy setting 1: $\epsilon = 10^{-4}$ and $\epsilon_{opt} = 10^{-3}$; Accuracy setting 2: $\epsilon = 10^{-7}$ and $\epsilon_{opt} = 10^{-6}$). (a-b) Plate. (c-d) Sphere.
V. ON THE EXISTENCE OF THE \( \mathcal{H} \)-MATRIX REPRESENTATION OF \( G^{-1} \) AND \( G \)'S LU FACTORS

In this section, we numerically prove the existence of an \( \mathcal{H} \)-matrix representation of the inverse of \( G \) and \( G \)'s LU factors by examining their rank distributions. The detailed procedure is as follows: we directly compute \( G^{-1} \) and \( G \)'s LU factors without introducing any approximation; we then use SVD to obtain the rank distribution in \( G^{-1} \) and \( G \)'s LU factors. Since the computation requires a complete full-matrix form of \( G^{-1} \) and \( G \)'s LU factors, it is not practical to use a problem having a large electric size as an example. We thus considered a conducting plate of 6\( \lambda \) and a conducting sphere of 4\( \lambda \). We computed the rank of each off-diagonal block that satisfies the admissibility condition (5) at the lowest level of a block cluster tree. The accuracy requirement in SVD is set to be \( \frac{k}{\min(m, n)} \) with respect to matrix block index. It is clear that the off-diagonal blocks that satisfy the admissibility condition in \( G^{-1} \) and \( G \)'s LU factors are low rank. Therefore, not only the original matrix \( G \) can be represented by an \( \mathcal{H} \)-matrix, but also \( G^{-1} \) and \( G \)'s LU factors can be represented by an \( \mathcal{H} \)-matrix. Furthermore, the rank distribution of \( G^{-1} \) and \( G \)'s LU factors is very similar to that of \( G \). In addition, both \( G^{-1} \) and \( G \)'s LU factors have a smaller rank than the original matrix, with the rank of \( G \)'s LU factors being the smallest. This suggests that the \( \mathcal{H} \)-matrix partition constructed for the original matrix \( G \) is equally applicable to \( G^{-1} \) and \( G \)'s LU factors. In Fig. 13, \( U \)'s rank distribution is plotted to represent \( L \)'s and \( U \)'s rank distribution since these two have a very similar rank distribution, with \( U \)'s rank being slightly larger.

VI. PROPOSED FAST IMPLEMENTATION OF LU FACTORIZATION

In this section, we show how to perform a fast LU factorization using the \( \mathcal{H} \)-matrix based representation of \( G \) and \( G \)'s LU factors. Based on the findings in the section above, we use the \( \mathcal{H} \)-partition constructed for \( G \) for the \( \mathcal{H} \)-partition of \( L \) and \( U \).

The \( \mathcal{H} \)-based LU factorization has been discussed in ([17], p. 119). However, no detailed implementation is given. In the following, we give a number of pseudo-codes to show a fast implementation of \( \mathcal{H} \)-based LU factorization, which is not reported anywhere else. It is worth mentioning that [10] did an ACA-based LU factorization, the procedure of which is very different from the proposed one. The fast \( \mathcal{H} \)-based LU factorization proposed in this paper has a factorization cost of \( O((k_{ave} C_{sp})^2 N \log^2 N) \), a solution cost of \( O((k_{ave} C_{sp}) N \log N) \), and memory consumption of \( O((k_{ave} C_{sp}) N \log N) \), with \( k_{ave} \) and \( C_{sp} \) minimized for a prescribed accuracy by the methods described in section IV.

A. LU factorization basics

Given an IE-based system matrix \( G \), we cast it into a form

\[
G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}.
\]  

(19)

The LU decomposition can be recursively computed by the following equation:

\[
G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \cdot \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = LU.
\]  

(20)

B. Proposed fast implementation of the LU factorization

We developed a pseudo-code shown in (21) to recursively perform LU factorization.

\[ \text{LU-Decomposition G=LU} \]  

Procedure \( \mathcal{H} \)-LU(G)

\( (G \) is the input matrix overwritten by \( L \) and \( U \))

If \( G \) is a non-leaf block

\[ \mathcal{H} - LU(G_{11}) \rightarrow L_{11}, U_{11}, \]  

Solve_{LU}(L_{11}, G_{12}) \rightarrow U_{12}, \]  

Solve_{LU}(G_{21}, U_{11}) \rightarrow L_{21}, \]  

\[-L_{21} \times U_{12} + G_{22} \rightarrow G_{22}, \]  

\[ \mathcal{H} - LU(G_{22}) \rightarrow L_{22}, U_{22}. \]  

else

\[ \text{Full-LU}(G) \]  

(21)
The underlying algorithm is as follows. When $G$ is a non-leaf matrix block, we recursively call (21) until $G_{11}$ is a full matrix block. We then directly compute the LU factors of the $G_{11}$ using a full-matrix-based LU factorization, which generates $L_{11}$ and $U_{11}$. Next, we call function Solve_LX shown in (22) and Solve_XU to compute $U_{12}$, and $L_{21}$ respectively.

Algorithm for Solving a Lower Triangular System

$LX = G$, with $G$ being an $H$ matrix

Procedure Solve_LX($L,G$)

(L and $G$ are input matrices, $G$ is overwritten by $X$)

If $L$ is a non-leaf block

If $G$ is a non-leaf block

Solve_LX($L_{11},G_{11}$), Solve_LX($L_{11},G_{12}$)

$-L_{21} \times G_{11} + G_{21} \rightarrow G_{21}$, Solve_LX($L_{22},G_{21}$)

$-L_{21} \times G_{12} + G_{22} \rightarrow G_{22}$, Solve_LX($L_{22},G_{22}$)

else if $G$ is an admissible block

Solve_LF($L,A$)

else

Solve_LF($L,G$)

end if

else

Full_LX($L,G$)

(Solve a full-matrix triangular system)

end

(22)

The $Solve_LX(L,G)$ is to solve a lower triangular system $LX = G$, where $L$ and $G$ are input matrices having $H$-representations, and $X$ is the solution. The $Solve_XU(G, U)$ is to solve an upper triangular system, which can be derived in a similar fashion as (22). In (22), a function $Solve_LF$ is called. Similar to $Solve_LX$, $Solve_LF$ also solves a triangular system. The difference is that the right-hand-side matrix for $Solve_LX$ is an $H$ matrix, whereas that for $Solve_LF$ is a full matrix. The pseudo-code of $Solve_LF$ is given in (23).

Algorithm for Solving a Lower Triangular System

$LX = F$, with $F$ being a Full Matrix

Procedure Solve_LF($L,F$)

(L and $F$ are input matrices, $F$ is overwritten by $X$)

If $L$ is a non-leaf block

Solve_LF($L_{11},F_{11}$)

$-L_{21} \times F_{1} + F_{2} \rightarrow F_{2}$

Solve_LF($L_{22},F_{2}$)

else

Full_LX($L,F$)

(Solve a full-matrix triangular system)

end

(23)

In the final step of (21), we use $U_{12}$ and $L_{21}$ to update $G_{22}$, and then call (21) recursively until $L_{22}$ and $U_{22}$ are computed. As can be seen from (20) to (23), efficient LU factorization relies on efficient block multiplication and block addition. In next subsection, we show how to efficiently perform these two operations for a prescribed accuracy.

C. Fast implementation of the block multiplication $G^{b} = G^{b1} \times G^{b2}$

We give a pseudo-code of computing $G^{b} = G^{b1} \times G^{b2}$ in (24).

Recursive Multiplication Algorithm

Procedure $H$-mult($G^{b1},G^{b2},G^{b},\epsilon_{LU}$)

If ($G^{b1},G^{b2},G^{b}$ are all non-leaf blocks)

for ($i =0; i<2; i++$)

for ($j =0; j<2; j++$)

for ($k =0; k<2; k++$)

$\hat{H}$-mult($G^{b1}(i,k),G^{b2}(k,j),G^{b}(i,j)$)

else if ($G^{b}$ is a non-leaf block, $G^{b1}$ or $G^{b2}$ is a leaf block)

Multiply_RK($G^{b1},G^{b2},G^{b}$)

$G^{b} \hat{=} \ G^{b} + G^{b}$

else if $G^{b}$ is an admissible block

Multiply_RK($G^{b1},G^{b2},G^{b}$)

else if $G^{b}$ is an inadmissible block

Multiply_Full($G^{b1},G^{b2},G^{b}$)

end if

(24)

where, $b, b_1$, and $b_2$ represent three blocks in the same level of an $H$-partition, $\epsilon_{LU}$ represents a prescribed accuracy. If $G^{b},G^{b1},$ and $G^{b2}$ are all non-leaf blocks, we recursively call (24). If one of $G^{b1}$ and $G^{b2}$ is a leaf block, or $G^{b}$ is an admissible block, we call function Multiply_RK shown in (25) to compute an admissible product. In (24), the addition is performed based on the prescribed accuracy $\epsilon_{LU}$, which is denoted by $\hat{=}_{LU}$. The detailed procedure of the addition is given in the following subsection.

Procedure Multiply_RK($G^{b1},G^{b2},G^{b},\epsilon_{LU}$)

if $G^{b1}$ and $G^{b2}$ are both non-leaf blocks

for ($i =0; i<2; i++$)

for ($j =0; j<2; j++$)

for ($k =0; k<2; k++$)

Multiply_RK($G^{b1}(i,k),G^{b2}(k,j),G^{b}(i,j)$)

$G^{b} \hat{=} \ G^{b} + G^{b}$

($G^{b}$ is a non-leaf block)
In (25), there are two multiplication cases. One is to multiply an admissible block $G^{b1}$ by an admissible block of an $AB^T$ form, for which we can compute $G^{b1}A$ as a new $A$. The other multiplication case is to multiply $G^{b1}$ by a full matrix block $F$, for which we can first apply SVD to $F$ to generate a form $AB^T$ based on the prescribed accuracy $\epsilon_{LU}$. If $G^b$ is a full matrix block, a normal full matrix multiplication is computed. The additions in (25) again are performed based on $\epsilon_{LU}$.

D. Fast implementation of the block addition $G^b = G^{b1} + G^{b2}$

Two cases are involved in the addition operations. One is to multiply an admissible block $G^{b1}$ by an admissible block of an $AB^T$ form, for which we can compute $G^{b1}A$ as a new $A$. The other multiplication case is to multiply $G^{b1}$ by a full matrix block $F$, for which we can first apply SVD to $F$ to generate a form $AB^T$ based on the prescribed accuracy $\epsilon_{LU}$. If $G^b$ is a full matrix block, a normal full matrix multiplication is computed. The additions in (25) again are performed based on $\epsilon_{LU}$.

VII. COMPLEXITY ANALYSIS

Three numerical procedures are involved in the proposed direct IE solver: rank minimization, $H$-partition optimization, and LU-based direct matrix solution. We analyze the complexity of each in the following.

The complexity of the rank minimization scheme for each admissible block, described in section IV-A, is linear. This is because both ACA+ and reduced SVD have a linear complexity for each admissible block [17], [20].

For the $H$-partition optimization shown in (17), two basic operations are involved. One is the factorization of non-leaf blocks. This operation is carried out by the function $\text{Rk} \_ \text{Factor}$ based on ACA+ and SVD. The other is the conversion of non-leaf blocks into an admissible block shown in (18). This operation is carried out by the function $\text{Merge} \_ \text{RkBlocks}$ by using an SVD based truncated addition. Since both ACA+ and SVD have a linear complexity for each matrix block, the two basic operations involved in (17) also have a linear cost.

From procedure (21), it can be seen that at the leaf level, the computation of the recursive LU factorization essentially includes a full-matrix LU factorization, a full-matrix solution of a lower triangular system, and a full-matrix solution of an upper triangular system, all of which have the same complexity as a full-matrix block multiplication. At all the other levels, a number of block-matrix multiplications are computed, which have the same recursive pattern as that in an $H$-based matrix-matrix multiplication. Therefore, the $H$-based LU factorization has the same complexity as $H$-based multiplication, which is bounded by $O((k_{ave}C_{sp})^2N \log^2 N)$ for electrodynamic problems as derived in (15). The $H$-based LU solution has the same complexity as the $H$-based matrix-vector multiplication, and hence its cost is $O((k_{ave}C_{sp})N \log N)$ for electrodynamic problems as derived in (12). Here, $k_{ave}C_{sp}$ is minimized for a given accuracy by the methods proposed in section IV.

VIII. NUMERICAL RESULTS

To test the performance of the proposed direct IE solver, we simulated a PEC (perfect electrically conducting) plate, a PEC sphere, and a PEC cylinder from a small number of unknowns to over 1 million unknowns, from small electric sizes to over 95 wavelengths. In all these examples, $\eta = 1$ and $\text{leafsize} = 32$ were used. Note that $\eta = 1$ was used to generate an initial $H$-partition, which was later replaced by the optimized $H$-partition by the method proposed in Section IV-B. The error tolerance $\epsilon_{opt}$ used in the $H$-partition optimization was set as $10^{-3}$. The error tolerance $\epsilon_{LU}$ used in the $LU$ factorization was $10^{-2}$. The computer used was a Dell’s PowerEdge 6950s server with 8222SE AMD Opteron processors. Double precision was employed in all the simulations.

First, we tested the accuracy of the $H$-matrix representation obtained by the proposed rank minimization and partition optimization methods. The error of the $H$-matrix representation is measured by $\|G - \tilde{G}\| / \|G\|$, where $G$ is the $H$-matrix representation of the original $G$, and the Frobenius norm is used. We simulated a PEC plate from 2 $\lambda$ to 14 $\lambda$, and a PEC sphere from 2 $\lambda$ to 8 $\lambda$. Fig. 14 shows the accuracy of the $H$-matrix representation generated from the proposed method.
Excellent accuracy can be observed in the entire range of electric sizes. Since the assessment of the matrix error requires the knowledge of the full matrix \( G \), we did not simulate larger problem sizes in this testing case.

Next, we tested the accuracy and efficiency of the proposed direct LU based IE solver. The first example is a conducting sphere illuminated by a normally incident plane wave. The electric size of the sphere is from \( 2 \lambda \) to \( 14 \lambda \). The discretization results in unknowns from 3,688 to 1,152,368. The average partition rank \( k_{\text{ave}} \) resulting from the proposed rank minimization and \( \mathcal{H} \)-partition optimization methods is shown in Fig. 15(a) in the entire range of electric sizes. It can be seen that \( k_{\text{ave}} \) is minimized to be a small number compared to \( N \). In Fig. 15(b), we plot the memory cost of the proposed solver. The theoretical expectation is also plotted for comparison, which is shown by the red solid line. Excellent agreement with the theoretical analysis is observed. To test the accuracy, in Fig. 16(a) and (b), we plot the E-plane bi-static RCS simulated for \( 12 \lambda \) and \( 26 \lambda \) respectively. The RCS is shown to agree well with the analytical Mie-Series solution. In Fig. 16(c), we plot the solution error. Less than 6% error is observed in the entire frequency band.

The second example is a 3D conducting plate, the electric size of which is from \( 2 \lambda \) to \( 60 \lambda \). The discretization results in 1,160 unknowns to 1,078,800 unknowns. The average partition rank \( k_{\text{ave}} \) resulting from the proposed rank minimization and partition optimization methods is shown in Fig. 17(a). It is clear that the average rank is minimized to be a small number, in addition, for the plate example, it is controlled to be almost a constant in the entire range of electric sizes. The \( C_{sp} \) for this example can be seen from Fig. 7(b). In Fig. 17(b), we plot the memory cost of the proposed direct IE solver, which agrees well with the theoretical prediction depicted by the solid line. In Fig. 17(c), we plot the solution error of the proposed direct IE solver measured by \( \| \mathbf{G}^{-1} - V \| / \| V \| \). Good accuracy can be observed. In addition, the accuracy is kept to be almost a constant in the entire range. From Fig. 17(d) to (f), we plot the computational cost of the proposed direct LU solver has a very good agreement with the theoretical prediction depicted by the solid lines. The computation for the 60 \( \lambda \) case having over 1 million unknowns was finished within 10-hour LU decomposition time, 55-second LU solution time, and costing 31.5 GB storage only.

The last example is a conducting cylinder, the length of which is from \( 2 \lambda \) to 96 \( \lambda \). The ratio of length to radius is 20. The number of unknowns is from 1,391 to 1,075,200. In Fig. 18(a), we plot the average partition rank \( k_{\text{ave}} \) resulting from the proposed rank minimization and partition optimization methods in the entire electric-size range. In Fig. 18(b), we plot \( C_{ad} \) versus the electric size. Without the proposed \( \mathcal{H} \)-partition optimization, the \( C_{ad} \) is between 47 and 51. Clearly, \( C_{ad} \), and hence \( C_{sp} \) is reduced greatly. In Fig. 19(a), we plot the solution error with respect to the number of unknowns. Good accuracy is observed. Note that the error is controllable by \( \epsilon \), \( \epsilon_{\text{opt}} \), and \( \epsilon_{LU} \). If better accuracy is required, it can be obtained by reducing the error tolerances. From Fig. 19(b) to (d), we show the computational cost of the proposed direct LU solver in LU factorization, LU Solution, and storage. Again, an excellent agreement with theoretical analysis is observed. The LU factorization for the 1,075,200 unknown case costs less than 20 hours and 37.6 GB memory. The LU solution time is 85 seconds only. Compared to results obtained for similar examples reported in open literature, the proposed direct IE solver is much more efficient in both CPU time and memory consumption, even though double precision was used for computation.

The above direct matrix solutions were all generated based on \( \epsilon = 10^{-4}, \epsilon_{opt} = 10^{-3}, \) and \( \epsilon_{LU} = 10^{-2} \). To test the performance of the proposed direct solver for a higher order of accuracy, we set \( \epsilon = 10^{-5}, \epsilon_{opt} = 10^{-4}, \) and \( \epsilon_{LU} = 10^{-3} \). We simulated the same plate example simulated in Fig. 17. In Fig. 20(a), we plot the solution error with respect to the number of unknowns. An excellent accuracy can be observed in the entire range of electric sizes, where the worst error is shown to be less than 0.33%, in comparison with the 3% error achieved by the previous accuracy setting. From Fig. 20(b) to (d), we show the computational cost of the proposed direct LU solver in memory, LU factorization, and LU solution. Again, an excellent agreement with theoretical analysis is observed.
IX. CONCLUSION

In this work, we further develop the $\mathcal{H}$-matrix based mathematical framework to accelerate the direct solution of the integral-equation-based analysis of electrodynamic problems. We show that the existing complexity analysis of the $\mathcal{H}$-matrix based computation, which is based on a constant rank, is not proper for analyzing electrodynamic problems. This is because the rank of each block required by an electrodynamic kernel for a given accuracy increases with the electric size of the block diameter. It is not a constant any more. However, using the maximal rank of all the admissible blocks to bound the complexity also overestimates the complexity, as demonstrated clearly in Fig. 4. In addition, the complexity of an $\mathcal{H}$-matrix based computation of electrodynamic problems is highly dependent on the $\mathcal{H}$-matrix partition, i.e. the number of admissible blocks at each tree level and etc. This is an important factor that received little attention in previous research. Traditionally, the number of admissible blocks at tree level $l$ is treated as $O(2^l)$, and the rank of each admissible block at tree level $l$ is regarded as $O(\frac{N}{2^l})$. As a result, the gain in computational efficiency of an $\mathcal{H}$-matrix based method for solving high-frequency problems is pessimistic.

In this work, we show that there exists a big space to optimize the $\mathcal{H}$-matrix partition for frequency dependent problems. Existing $\mathcal{H}$-matrix partition is based on a geometry based admissibility condition. This condition is controlled by an empirical parameter instead of a prescribed accuracy. The resultant partition is by no means optimal especially for electrodynamic problems. We hence develop a new $\mathcal{H}$-partition method that is frequency dependent, and also directly controlled by accuracy requirements. With the proposed new partition, the number of admissible blocks at each tree level is significantly reduced compared to that generated by the conventional geometry based $\mathcal{H}$-partition. We also show that the number of admissible blocks at tree level $l$ is not $O(2^l)$.

In addition, the block rank is observed to be proportional to the electric size of the block diameter, as can be seen from Fig. 10 and Fig. 11. Since the number of unknowns in a surface...
IE based method is proportional to the electric size's square, the rank of each admissible block at tree level \( l \) is bounded by \( O\left(\frac{N}{2^l}\right)^{0.5} \) instead of \( O\left(\frac{N}{2^l}\right) \), thus the block is low rank. In addition, the actual number of the block rank should be determined and minimized based on accuracy requirements. This paper provides an efficient matrix algebra based method to perform this task with negligible computational overhead.

In light of the fact that the complexity of an \( H \)-matrix based computation of electrodynamic problems is determined not only by block rank (the rank of each admissible block) but also by \( H \)-matrix partition, we propose a new parameter, average partition rank \( k_{\text{ave}} \), to bound the storage units and operation counts of an \( H \)-matrix based computation. Different from block rank, the partition rank \( k_{\text{ave}} \) contains the information of the \( H \)-matrix partition. Based on \( k_{\text{ave}} \), we develop new bounds of the computational cost for the \( H \)-matrix-based computation of electrodynamic problems. The new bounds suggest that the smaller the product \( k_{\text{ave}}C_{sp} \) is, the smaller the computational cost. For electrodynamic problems, the \( k_{\text{ave}} \) and \( C_{sp} \) are frequency dependent and hence electric size dependent. The objective of this work is to minimize the \( k_{\text{ave}}C_{sp} \) for each frequency point to reduce computational cost. The \( k_{\text{ave}} \) for a given partition is minimized by finding a minimal rank of each admissible block for a given accuracy. The \( C_{sp} \) and \( k_{\text{ave}}C_{sp} \) are reduced by developing a new \( H \)-partition algorithm. By minimizing \( k_{\text{ave}}C_{sp} \) for each frequency point, we significantly reduce the cost of the \( H \)-matrix based direct solution of electrodynamic problems.

Moreover, we developed an efficient LU-factorization for directly solving the dense system matrix resulting from an IE-based analysis of large-scale electrodynamic problems. The operation counts of the proposed direct solver are \( O((k_{\text{ave}}C_{sp})^2N\log^2 N) \) in LU factorization, \( O((k_{\text{ave}}C_{sp})N\log N) \) in LU solution, and the memory consumption is \( O(k_{\text{ave}}C_{sp}N\log N) \), all of which were verified theoretically and also numerically. Numerical results have demonstrated the accuracy and efficiency of the proposed direct IE solver for simulating large-scale electrodynamic problems, with the \( N \)-dependent \( k_{\text{ave}} \) and \( C_{sp} \) minimized to be small compared to \( N \). The dense matrix involving over 1 million unknowns formulated for a 96-wavelength problem is factorized in fast CPU run time, modest memory usage, and with prescribed accuracy satisfied. The proposed methods for minimizing the rank and optimizing the \( H \)-partition not only can be used in the proposed solver, but also can be employed in other fast integral equation solvers such as fast multipole
Fig. 17. Simulation of a PEC plate from 2λ to 60λ. (a) Average partition rank. (b) Memory cost. (c) Solution error. (d) $\mathcal{H}$-matrix construction time. (e) LU factorization time. (f) LU solution time.

based methods.

REFERENCES

Fig. 18. Simulation of a PEC cylinder from $2 \lambda$ to $96 \lambda$. (a) $k_{\text{ave}}$ versus $N$. (b) $C_{ad}$ versus $N$.

Fig. 19. Simulation of a PEC cylinder from $2 \lambda$ to $96 \lambda$. (a) Solution error. (b) LU factorization time. (c) LU solution time. (d) Memory Cost.


Fig. 20. Simulation of a PEC plate from $2\lambda$ to $60\lambda$ based on a higher-order accuracy setting ($\epsilon = 10^{-5}$, $\epsilon_{opt} = 10^{-4}$, and $\epsilon_{LU} = 10^{-3}$). (a) Solution error. (b) Memory Cost. (c) LU factorization time. (d) LU solution time.