Existence of $\mathcal{H}$-Matrix Representations of the Inverse Finite-Element Matrix of Electrodynamic Problems and $\mathcal{H}$-Based Fast Direct Finite-Element Solvers

Haixin Liu and Dan Jiao, *Senior Member, IEEE*

***Abstract*—In this work, we prove that the sparse matrix resulting from a finite-element-based analysis of electrodynamic problems can be represented by an $\mathcal{H}$ matrix without any approximation, and the inverse of this sparse matrix has a data-sparse $\mathcal{H}$-matrix approximation with error well controlled. Two proofs are developed. One is based on the general eigenvalue-based solution to the ordinary differential equations, and the other is based on the relationship between a partial differential operator and an integral operator. Both proofs have reached the same conclusion. Based on the proof, we develop an $\mathcal{H}$-matrix-based direct finite-element solver of $O(kN \log N)$ memory complexity and $O(k^2 N \log^2 N)$ time complexity for solving electromagnetic problems, where k is a small parameter that is adaptively determined based on accuracy requirements, and N is the number of unknowns. Both inverse-based and LU-based direct solutions are developed. The LU-based solution is further accelerated by nested dissection. A comparison with the state-of-the-art direct finite element solver that employs the most advanced sparse matrix solution has shown clear advantages of the proposed direct solver. In addition, the proposed solver is applicable to arbitrarily-shaped three-dimensional structures and arbitrary inhomogeneity.**

***Index Terms*—$\mathcal{H}$ matrix, direct solution, electromagnetic analysis, fast solvers, finite element methods, nested dissection.**

I. Introduction

COMPARED to other computational electromagnetic methods such as finite-difference-based methods and integral-equation-based methods, finite-element methods (FEM) have demonstrated a strong capability in handling both irregular geometries and arbitrary inhomogeneity. A finite-element-based analysis of a complex electromagnetic problem generally results in a large-scale system matrix. Although the matrix is sparse, solving it can be a computational challenge when the problem size is large. A traditional direct solution is computationally intensive. As yet, no linear complexity has been reported for FEM-based direct solutions to general electromagnetic problems. In [1], the optimal operation count of the direct solution of an FEM matrix was shown to be $O(N^{1.5})$, where N is the matrix dimension. Recent exploration of fast direct solutions for FEM-based electromagnetic analysis can be seen in [2], [3], where two-dimensional problems were studied. State-of-the-art finite-element-based solvers rely on iterative approaches to solve large-scale matrices. The resultant computational complexity is $O(N_{it}N_{rhs}N)$, where N_{it} is the number of iterations, and N_{rhs} is the number of right hand sides. When N_{it} and N_{rhs} are large, iterative solutions become inefficient. In addition, the complexity is problem dependent since the iteration number N_{it} is, in general, problem dependent.

In this work, we consider the fast direct solution of FEM based matrices for solving electromagnetic problems. Our solution is built upon the observation that although the inverse of an FEM-based matrix generally leads to a dense matrix, this matrix can be thought of as "data-sparse," i.e., it can be specified by few parameters. There exists a general mathematical framework called the "hierarchical ($\mathcal{H}$) matrix" framework [4]–[7], which enables a highly compact representation and efficient numerical computation of the dense matrices. To be specific, if matrix $\mathbf{C}$ is an $m \times n$ off-diagonal block in an $\mathcal{H}$ matrix which describes interactions on upper levels in the hierarchy, it can be written as $\mathbf{C} = \mathbf{A}\mathbf{B}^T$ where $\mathbf{A}$ is of dimension $m \times r$, $\mathbf{B}$ is of dimension $n \times r$, and r denotes the rank of $\mathbf{C}$ with $r < m$ and $r < n$. Storage requirements and matrix-vector multiplications using $\mathcal{H}$-matrices have been shown to be of complexity $O(N \log N)$. Moreover, the inverse of an $\mathcal{H}$ matrix can be obtained in $O(N \log^2 N)$ complexity. In [14], [17], such an $\mathcal{H}$-matrix based form of the system matrix was used in the integral equation based methods to solve large-scale electrodynamic problems involving over one million unknowns. In [14], [15], the error bound of the $\mathcal{H}$-and $\mathcal{H}^2$-matrix-based representation of an electrodynamic problem was derived for integral equation based analysis. It was shown that exponential convergence of the error with respect to the number of interpolation points can be achieved irrespective of the electric size. In addition, different from static cases in which a constant rank can maintain the same order of accuracy regardless of problem size, the rank required by an electrodynamic system for a given accuracy is a variable with respect to tree level, electric size, admissible block, and admissibility condition. It is also worth mentioning that the matrices underlying generic Fast Multiple Algorithms [18]–[21] are $\mathcal{H}^2$-matrices, as noted in [22], which are a special class of the $\mathcal{H}$ matrix.

In the mathematical literature, the existence of an $\mathcal{H}$-matrix approximation was only proved for elliptic partial differential equations (PDE) [8]. Although the $\mathcal{H}$-matrix-based technique

Manuscript received March 17, 2010; accepted August 24, 2010. Date of publication November 09, 2010; date of current version December 10, 2010. This work was supported by NSF under Award No. 0747578 and Award No. 0702567.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906 USA (e-mail: liu127@purdue.edu; djiao@purdue.edu)

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

Digital Object Identifier 10.1109/TMTT.2010.2086532

has been successful in the integral-equation based solution of Maxwell's equations, in an FEM-based framework, it has been mainly used for solving elliptic PDEs such as a Poisson equation. No work has been reported for the finite-element-based solution of wave equations. The research challenges here are: First, one has to prove that there exists an $\mathcal{H}$-matrix-based representation of the inverse of the FEM-based matrix for electrodynamic problems so that the accuracy of the $\mathcal{H}$-based approach can be controlled; Second, one has to develop a direct solver that is faster than state-of-the-art direct sparse solvers so that it is worthwhile to explore an $\mathcal{H}$-based fast solution. Third, one has to demonstrate the accuracy and complexity of this fast solver by theoretical analysis in addition to numerical experiments since the conclusions drawn from numerical experiments are often problem dependent.

In [9]–[11], we have published preliminary results on a fast $\mathcal{H}$-inverse-based direct solver for the FEM-based analysis of electromagnetic problems. The main contributions of this paper are as follows. *First*, we theoretically proved the existence of an $\mathcal{H}$-matrix-based representation of the FEM matrix and its inverse for electrodynamic problems. We realize the fact that it is difficult to develop such a proof solely from a mathematical point of view. However, by synthesizing electromagnetic physics and mathematics, such a proof becomes obvious. *Second*, we developed an $\mathcal{H}$-matrix-based direct FEM solver of $O(kN\log N)$ memory complexity and $O(k^2N\log^2 N)$ time complexity for solving vector wave equations, where k is a small parameter that is adaptively determined based on accuracy requirements. In this direct solver, we developed both inverse-based direct solution and LU-decomposition-based direct solution with accuracy well controlled. In addition, we incorporated nested dissection [1] to further expedite the $\mathcal{H}$-LU-based solution of vector wave equations. *Third*, we performed a theoretical analysis of the computational complexity of the proposed fast direct solver. In addition, we analyzed the accuracy of the proposed direct solver and show that it is error controllable. *Last but not least*, we compared the proposed direct solver with the state-of-the-art direct FEM solver that employs the most advanced sparse matrix solution such as UMFPACK 5.0 [12]. UMFPACK has been adopted by Matlab for fast sparse matrix solution. It has incorporated almost all the advanced sparse matrix techniques such as the multifrontal method and the AMD ordering for solving large-scale sparse matrices. The proposed solver is shown to outperform the UMFPACK 5.0 in both matrix decomposition and matrix solution time without sacrificing accuracy.

The remainder of this paper is organized as follows. In Section II, the vector FEM-based analysis of general electromagnetic problems is outlined. In Section III, the existence of the $\mathcal{H}$-matrix representation of the FEM matrix and its inverse is proved for electrodynamic problems. In Section IV, the detailed numerical procedure of the proposed direct solver is given. In Section V, the complexity and accuracy of the proposed solver are analyzed. In Section VI, the choice of simulation parameters is discussed. In Section VII, numerical results are shown to demonstrate the accuracy and almost linear complexity of the proposed direct FEM solver. Section VIII relates to our conclusions.

II. Vector FEM-Based Analysis of General Electromagnetic Problems

Consider the second-order vector wave equation

$$\nabla \times \left(\frac{1}{\mu_r}\nabla \times \mathbf{E}\right) - k_0^2\varepsilon_r\mathbf{E} = -jk_0Z_0\mathbf{J} \tag{1}$$

subject to boundary conditions:

$$\hat{n} \times \mathbf{E} = \mathbf{P} \quad \text{on } S_1 \tag{2}$$

$$\frac{1}{\mu_r}\hat{n} \times (\nabla \times \mathbf{E}) + \gamma_e\hat{n} \times (\hat{n} \times \mathbf{E}) = \mathbf{U} \quad \text{on } S_2. \tag{3}$$

The boundary condition in (3) can be used to truncate the computational domain for an FEM-based analysis, where γ_e and $\mathbf{U}$ can be frequency and position dependent.

An FEM-based solution to the above boundary value problem results in a linear system of equations [13]

$$\mathbf{Y}\{E\} = \{I\} \tag{4}$$

where $\mathbf{Y}$ can be written as

$$\mathbf{Y} = -k_0^2\mathbf{T} + \mathbf{S} + \mathbf{B} \tag{5}$$

in which

$$\begin{aligned}\mathbf{T} &= \int\int\int_V [\varepsilon_r\mathbf{N}_i \cdot \mathbf{N}_j]dV\\ \mathbf{S} &= \int\int\int_V \left[\frac{1}{\mu_r}(\nabla \times \mathbf{N}_i) \cdot (\nabla \times \mathbf{N}_j)\right]dV\\ \mathbf{B} &= \int\int_{s_2} [\gamma_e(\hat{n} \times \mathbf{N}_i) \cdot (\hat{n} \times \mathbf{N}_j)]dS\end{aligned} \tag{6}$$

where V denotes the computational domain, and $\mathbf{N}$ is the vector basis used to expand unknown $\mathbf{E}$. In (6), $\mathbf{T}$ is known to be a mass matrix, and $\mathbf{S}$ is known to be a stiffness matrix. $\mathbf{T}$ is positive definite, $\mathbf{S}$ is semi-positive definite, and the combined system $\mathbf{Y}$ is, in general, indefinite.

When the problem size is large, solving $\mathbf{Y}$ is a computational challenge despite its sparsity. In Section III, we show that $\mathbf{Y}$ and its inverse $\mathbf{Y}^{-1}$ both can be represented by an $\mathcal{H}$ matrix, from which a significant reduction in computational complexity can be achieved.

III. Existence of $\mathcal{H}$-Matrix Representation of the FEM Matrix and its Inverse for Electrodynamic Problems

An $\mathcal{H}$ matrix is generally associated with an admissibility condition [6]. To define an admissibility condition, we denote the whole index set containing the indexes of the basis functions in the computational domain by $\mathcal{I} = \{1, 2, \ldots, N\}$, where N is the total number of unknowns. Considering two subsets t and s of the $\mathcal{I}$, the admissibility condition is defined as

$$\min\{diam(\Omega_t), diam(\Omega_s)\} \leq \eta\, dist(\Omega_t, \Omega_s) \tag{7}$$

where Ω_t is the minimal subset of the space containing the supports of all basis functions belonging to t, $diam(\cdot)$ is the Euclidean diameter of a set, $dist(\cdot,\cdot)$ is the Euclidean distance between two sets, and η is a positive parameter. If subsets t and s satisfy (7), they are admissible; otherwise, they are inadmissible.

Denoting the matrix block formed by t and s by $Y_{t\times s}$, if all the blocks $Y_{t\times s}$ formed by the admissible (t, s) in $\mathbf{Y}$ can be represented by a low-rank matrix, $\mathbf{Y}$ is an $\mathcal{H}$ matrix. In other words, if $\mathbf{Y}$ possesses the following property

$$\mathbf{Y} \in \mathbb{C}^{\mathcal{I}\times\mathcal{I}} : \mathbf{Y}_{t\times s} \text{ is low rank for all admissible } (t, s) \quad (8)$$

it is an $\mathcal{H}$ matrix.

From the above definition of an $\mathcal{H}$ matrix, it is clear that the FEM system matrix $\mathbf{Y}$ formulated for an electrodynamic problem as shown in (5) is exactly an $\mathcal{H}$ matrix. This is because when the admissibility condition (7) is satisfied, the subsets t and s are geometrically disconnected, and hence the basis functions in these two sets cannot belong to the same element, and hence the matrix entries in $\mathbf{Y}_{t\times s}$, are all zero. Therefore, the FEM matrix resulting from the analysis of a general electromagnetic problem always has an exact $\mathcal{H}$-matrix representation without involving any approximation.

Next, we prove the inverse of $\mathbf{Y}$ also allows for an $\mathcal{H}$-matrix representation. We develop two proofs. One is based on the general solution to the ordinary differential equations; the other is based on the relationship between a partial differential operator and an integral operator in the context of electromagnetics. Both clearly prove the existence of an $\mathcal{H}$-matrix representation of the inverse of the FEM matrix for electrodynamic analysis.

A. *Proof Based on the General Solution to the Ordinary Differential Equations*

The FEM-based system of (4) can be rewritten in time domain as

$$\mu_0\varepsilon_0\mathbf{T}\frac{d^2\{E\}}{dt^2} + \mathbf{S}\{E\} = \{I(t)\}. \quad (9)$$

Here, for simplicity, we omit the absorbing boundary condition. We will address it at the end of this subsection.

For an ordinary differential equation with constant coefficients like (9), from [23]–[25], its solution in frequency domain can be written directly as

$$\{E\} = \mathbf{V}\left(\Lambda - k_0^2\mathbf{I}\right)^{-1}\mathbf{V}^T\{I\} \quad (10)$$

where $\mathbf{I}$ is the identity matrix, Λ is a diagonal matrix with its i-th entry being the i-th eigenvalue of the following system:

$$\mathbf{S}v = \lambda\mathbf{T}v \quad (11)$$

and $\mathbf{V}$ is the matrix containing all the eigenvectors v. From (10), it is clear that the field solution $\{E\}$ is nothing but a linear combination of all the eigenvectors of (11). The weight of each eigenvector is determined by $\left(\lambda_i - k_0^2\right)^{-1} V_i^T\{I\}$, where V_i is the i-th column of $\mathbf{V}$, i.e., the i-th eigenvector.

Clearly, given a frequency point ω, and hence k_0, not all the eigenvectors contribute equally to the final solution. Only those eigenvectors that have eigenvalues close to k_0^2 have a large weight, and hence need to be included in the final solution. Furthermore, among these eigenvectors, only those that have a nontrivial projection to the excitation vector, i.e., a nonzero $V_i^T\{I\}$, need to be considered. As a result, given an accuracy requirement ϵ, any eigenvalue that satisfies the following condition:

$$\left|\frac{1}{\lambda_i - k_0^2}\right| < \varepsilon \quad (12)$$

can be ignored with error well controlled. As a result, only a subset of eigenvectors need to be taken into account when constructing the solution of $\{E\}$. Thus, (10) can be rewritten as

$$\{E\} = \tilde{\mathbf{V}}_{N\times k}\left(\tilde{\Lambda}_{k\times k} - k_0^2\mathbf{I}_{k\times k}\right)^{-1}(\tilde{\mathbf{V}}_{N\times k})^T\{I\} \quad (13)$$

where $\tilde{\mathbf{V}}_{N\times k}$ is a subset of $\mathbf{V}$, which includes the k eigenvectors that have the largest weights, and $\tilde{\Lambda}_{k\times k}$ contains corresponding eigenvalues.

From (4) and (13), it is obvious that the inverse of the FEM matrix can be written as

$$\mathbf{Y}^{-1} = \mathbf{A}\mathbf{B}^T \quad (14)$$

in which $\mathbf{A} = \tilde{\mathbf{V}}_{N\times k}$, $\mathbf{B} = \tilde{\mathbf{V}}_{N\times k}\left(\tilde{\Lambda}_{k\times k} - k_0^2\mathbf{I}_{k\times k}\right)^{-1}$. Hence, the inverse of the FEM matrix $\mathbf{Y}$ is a low rank matrix with error well controlled. As a result, we prove that the inverse of the FEM matrix for electrodynamic analysis has an $\mathcal{H}$-matrix representation.

From the proof developed above, it can also be seen clearly that for a given frequency, the k eigenvalues that should be incorporated in the field solution may not be the largest k eigenvalues of the FEM system. Instead, they are the k eigenvalues that are the closest to the frequency being investigated. Furthermore, when the frequency changes, the number of eigenvalues that should be considered in the final solution may change also in order to keep the same accuracy. Hence, the rank for the $\mathcal{H}$-based representation of an electrodynamic problem is, in general, a function of frequency. In addition, even if the frequency being considered is infinity, still there is only a limited number of eigenvalues that need to be considered. For example, there is no need to consider the contribution of a DC mode to the field response at an infinitely large frequency. We only need to consider those modes having eigenvalues that are close to infinity.

The above proof is developed without considering the first-order term in (5) such as matrix $\mathbf{B}$ if $\gamma_e = jk$. Such a first-order term can originate from either absorbing boundary conditions or material loss. For an FEM system that has a first-order term, the general solution to the ordinary differential equations of any order in [23] is equally applicable, for which a quadratic eigenvalue analysis [25] can be conducted. Such an analysis again reveals a limited number of modes that can be present in the field solution for any frequency, given an accuracy requirement.

B. *Proof Based on the Relationship Between a Partial Differential Operator and an Integral Operator*

In addition to the proof developed above, we also developed a proof by using the relationship between a partial differential operator and an integral operator in the context of electromagnetics. In the following, we will first use free space as an example, and then generalize the proof to inhomogeneous cases.

Consider the electric field $\mathbf{E}$ due to an arbitrary current distribution $\mathbf{J}$ in free space. The current distribution $\mathbf{J}$ can always be decomposed into a group of electric dipoles $\tilde{I}_i l_i$, where $\tilde{I}_i$ is the current of the i-th element and is the length of the i-th current element. Using the FEM-based method, we solve a system (4) to obtain $\mathbf{E}$, where the right-hand-side vector $\{I\}$ has the following entries for a normalized $\mathbf{N}$

$$I_i = -j\omega\mu_0 \tilde{I}_i l_i. \tag{15}$$

On the other hand, $\mathbf{E}$ due to any current distribution $\mathbf{J}$ can be evaluated from the following integral:

$$\mathbf{E} = -j\omega\mu_0 \int\int\int_V \left(\mathbf{J}G_0 + \frac{1}{k_0^2}\nabla' \cdot \mathbf{J}\nabla G_0\right) dV' \tag{16}$$

where G_0 is free-space Green's function.

For a group of electric dipoles $\tilde{I}_n l_n (n = 1, 2, \ldots N)$, the $\mathbf{E}$ at any space point $\mathbf{r}$ can be obtained from (16) as

$$\mathbf{E}(\mathbf{r}) = -j\omega\mu_0 \sum_{n=1}^{N} [\, I_n l_n \hat{l}_n(\mathbf{r}') G_0(\mathbf{r}, \mathbf{r}') + \frac{1}{k_0^2}\nabla\left[I_n l_n \nabla \cdot (\hat{l}_n(\mathbf{r}') G_0(\mathbf{r}, \mathbf{r}'))\right] \tag{17}$$

where $\hat{l}_n$ is the unit vector tangential to the n-th current element. The above simply means that $\mathbf{E}$ is the summation of each dipole's contribution.

By sampling (17) at the center point of each edge in a 3-D finite-element based discretization, and testing (17) by the unit vector tangential to the edge, we obtain

$$\{E\} = \mathbf{Z}\{I\} \tag{18}$$

where $\{I\}$ is the same as that in (4), the entries of which are given in (15), and $\mathbf{Z}$ is a dense matrix having the following matrix elements:

$$\mathbf{Z}_{mn} = \frac{1}{-j\omega\mu_0}\left\{ -j\omega\mu_0 \hat{t}_m(\mathbf{r}_m) \cdot \hat{l}_n(\mathbf{r}'_n)\, G_0(\mathbf{r}_m, \mathbf{r}'_n) - \frac{j}{\omega\varepsilon}\hat{t}_m(\mathbf{r}_m) \cdot \nabla\left[\nabla \cdot \left(\hat{l}_n(\mathbf{r}') G_0(\mathbf{r}, \mathbf{r}')\right)\right]\right\} \tag{19}$$

where $\hat{t}_m$ is the unit vector tangential to the m-th edge, $\mathbf{r}_m$ denotes the center point of the m-th edge, $\mathbf{r}'_n$ denotes the point where the n-th current element is located. In (18), $\{E\}$ vector has the following entries:

$$E_m = \hat{t}_m(\mathbf{r}_m) \cdot \mathbf{E}(\mathbf{r}_m) \tag{20}$$

which is the same as the $\{E\}$ vector in (4).

Comparing (18) to (4), it is clear that the inverse of the FEM matrix $\mathbf{Y}$ is $\mathbf{Z}$, the elements of which are given in (19). If we can prove $\mathbf{Z}$ has an $\mathcal{H}$-matrix representation with error well controlled, $\mathbf{Y}^{-1}$ also has an $\mathcal{H}$-matrix approximation. Such a proof in fact has already been given in [14], [15], in which we show that the dense system matrix resulting from the analysis of an electrodynamic problem can be represented by an $\mathcal{H}$-matrix or an $\mathcal{H}^2$-matrix with error bounded irrespective of the electric size. Different from static cases in which a constant rank can maintain the same order of accuracy regardless of problem size, the rank required by an electrodynamic system for a given accuracy is a variable with respect to electric size, tree level, admissible block, and admissibility condition.

In an inhomogeneous problem, the $\mathbf{E}$ field due to a group of electric dipoles $\{\tilde{I}_i l_i\}$ can be written as

$$\mathbf{E} = \mathbf{E}^{\mathrm{inc}} + \mathbf{E}^{sca} \tag{21}$$

where $\{E^{\mathrm{inc}}\} = \mathbf{Z}\{I\}$. Thus, (21) can be written as

$$\mathbf{Z}_1\{\mathbf{E}\} = -\mathbf{Z}\{I\} \tag{22}$$

where $\mathbf{Z}_1$ is a matrix. Comparing (22) to (4), it can be seen that

$$\mathbf{Y}^{-1} = -\mathbf{Z}_1^{-1}\mathbf{Z}. \tag{23}$$

Since $\mathbf{Z}$ is an $\mathcal{H}$-matrix, even if $\mathbf{Z}_1^{-1}$ is a full matrix, $\mathbf{Y}^{-1}$ is still an $\mathcal{H}$-matrix. This can be readily proved as follows. Since $\mathbf{Z}$ is an $\mathcal{H}$-matrix, its admissible blocks can be represented by $\mathbf{Z} = \mathbf{AB}^T$ where $\mathbf{A}$ is of dimension $m \times k$, $\mathbf{B}$ is of dimension $m \times k$, where $k < m$. Multiplying a full matrix $\mathbf{C}$ by $\mathbf{AB}^T$ still yields an $\mathcal{H}$-matrix $\mathbf{DB}^T$ with $\mathbf{D} = \mathbf{CA}$ that is of dimension $m \times k$. As a result, the existence of the $\mathcal{H}$-matrix representation for the inhomogeneous cases is also proved.

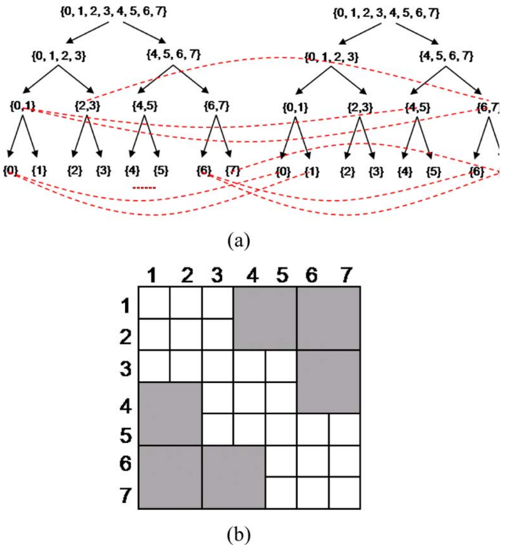
IV. Fast Direct Solution of the FEM System Matrix

Once the existence of the $\mathcal{H}$-matrix representation is proved for $\mathbf{Y}$ and $\mathbf{Y}^{-1}$, the $\mathcal{H}$-matrix arithmetics can be used to significantly accelerate the solution of $\mathbf{Y}$. In our proposed fast direct solver, we first build a block cluster tree to efficiently store the $\mathcal{H}$-matrix-based representation of $\mathbf{Y}$, its inverse, as well as $\mathbf{Y}$'s $\mathbf{LU}$ factors. This tree structure is also used to efficiently capture the hierarchical dependence in the $\mathcal{H}$-matrix. We then perform fast inverse and LU factorization based on $\mathcal{H}$-based representation of $\mathbf{Y}$. To further expedite the $\mathcal{H}$-based LU factorization, we incorporate nested dissection [1] to reduce the number of nonzero blocks to be computed. In addition, we develop an adaptive truncation scheme to systematically control the accuracy of $\mathcal{H}$-based operations for an accurate analysis of electrodynamic problems.

A. *Cluster Tree and Block Cluster Tree Construction*

We use a block cluster tree to efficiently store the $\mathcal{H}$-matrix-based representation of the FEM system matrix $\mathbf{Y}$, its inverse, as well as $\mathbf{Y}$'s $\mathbf{LU}$ factors. To construct a block cluster tree, a cluster tree needs to be built first. For the index set of the basis functions $\mathcal{I} = \{1, 2, \ldots, N\}$, we construct a cluster tree $T_{\mathcal{I}}$, which is a tree with vertex set V and edge set E as shown by the left (right) part of Fig. 1(a). Each vertex in the tree is called as a cluster. The set of children for a cluster $t \in T_{\mathcal{I}}$ is denoted by children (t). The root of the tree is the index set $\mathcal{I} = \{1, 2, \ldots, N\}$.

To construct a cluster tree, we start from the full index set of basis functions $\mathcal{I}$. We split the computational domain into two subdomains. We continue to split until the number of unknowns in each subdomain is less than or equal to the $leafsize$ $(n_{\min})$



(a)

(b)

Fig. 1. (a) A block cluster tree. (b) An $\mathcal{H}$-matrix structure.

which is a parameter to control the tree depth. Clusters with indexes no more than $leafsize$ are leaves. The set of leaves of $\mathcal{I}$ is denoted by $\mathcal{L}_{\mathcal{I}}$. In Fig. 1(a), the left (right) part is a cluster tree $T_{\mathcal{I}}$ with $N = 8$ and tree depth $p = 3$. The total number of clusters in this tree is 15.

A block cluster tree $T_{\mathcal{I}} \times_{\mathcal{J}}$ is built from two cluster trees $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$, and a given admissibility condition. Each block cluster $b \in T_{\mathcal{I}} \times_{\mathcal{J}}$ has the form $b = (t, s)$ with clusters $t \in T_{\mathcal{I}}$ and $s \in T_{\mathcal{J}}$, and b, t, s being in the same level. In a Galerkin-based FEM procedure, the testing function is chosen the same as the basis function. Therefore, the block cluster tree is constructed between the cluster tree $T_{\mathcal{I}}$ and itself. To build a block cluster tree $T_{\mathcal{I}} \times_{\mathcal{I}}$, we test blocks level by level starting with the root clusters of $T_{\mathcal{I}}$ and $T_{\mathcal{I}}$, and descending in the tree. Given two clusters $t \in T_{\mathcal{I}}$ and $s \in T_{\mathcal{I}}$, we check whether the admissibility condition is satisfied or not. If the two clusters are admissible, we stop at this level, draw a link between the two clusters as shown in Fig. 1(a), and do not check their children. If they are not admissible, we repeat the procedure for all combinations of the children of t and the children of s. The construction process stops when either at least one of t and s is a leaf or clusters t and s satisfy the admissibility condition. This procedure results in an $\mathcal{H}$-matrix structure as shown in Fig. 1(b). Each matrix block corresponds to a link drawn between $T_{\mathcal{I}}$ and $T_{\mathcal{I}}$ as shown in Fig. 1(a). Links drawn at the upper level of the tree correspond to admissible blocks denoted by $\mathcal{L}^{+}_{\mathcal{I}\times\mathcal{I}}$, while those drawn at the bottommost level represent inadmissible ones denoted by $\mathcal{L}^{-}_{\mathcal{I}\times\mathcal{I}}$. In Fig. 1(b), admissible blocks are represented by shaded blocks.

B. Representation of the FEM System Matrix, its Inverse, and LU Factors by an $\mathcal{H}$ Matrix

In an $\mathcal{H}$ matrix, inadmissible blocks are stored in a full matrix form, namely all the matrix entries are stored without any approximation. Admissible blocks $\mathbf{M}_{t\times s}$ are stored in a factorized form: $\mathbf{M}_{t\times s} = \mathbf{AB}^T$, where $\mathbf{A}$ is a $t \times k$ matrix and $\mathbf{B}$ is an $s \times k$ matrix, with k being the rank of the admissible block.

When constructing an $\mathcal{H}$-matrix-based representation of the FEM matrix $\mathbf{Y}$, all the non-zero matrix entries in $\mathbf{Y}$ are stored in inadmissible blocks and admissible blocks do not need to be filled because they are all zero. But we still have to form a block cluster tree to identify all the admissible blocks at each tree level because these blocks will be filled by the factorized $\mathbf{A}$ and $\mathbf{B}$ during the process of inverse or LU factorization. The rank in each admissible block is adaptively determined based on a required level of accuracy, the detail of which is given in Section E.

C. Fast Direct Inverse

The procedure of $\mathcal{H}$-based inverse is given in [6]. Here, we outline the algorithm to facilitate complexity and accuracy analysis to be developed in Section V for the proposed direct solver.

Rewriting the FEM matrix $\mathbf{Y}$ in the following form:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} \end{pmatrix}. \tag{24}$$

The inverse of $\mathbf{Y}$ can be done recursively by using (25), shown at the bottom of the page, where $\mathbf{S} = \mathbf{Y}_{22} - \mathbf{Y}_{21}\mathbf{Y}_{11}^{-1}\mathbf{Y}_{12}$. All the additions $\oplus$ and multiplications $\otimes$ in (25) are performed by $\mathcal{H}$-based arithmetics defined in [6] and [7], which is much faster than conventional matrix additions and multiplications. For example, for dense matrices, a formatted addition using $\mathcal{H}$-based arithmetics has $O(N \log N)$ complexity, and a formatted multiplication has $O(N \log^2 N)$ complexity. A pseudo-code for the $\mathcal{H}$-inverse is given in (26), shown at the bottom of the next page.

D. Fast LU Decomposition With Nested Dissection

Since what is to be solved in (4) is $\mathbf{Y}^{-1}\{I\}$ instead of $\mathbf{Y}^{-1}$, and the number of right hand sides is smaller than N in many applications, an LU-factorization-based direct solution is generally more efficient than an inverse-based direct solution. In addition, in an LU factorization process, the input matrix can be overwritten by $\mathbf{L}$ and $\mathbf{U}$ factors, thus the memory usage can be cut by half. In contrast, when computing inverse, a temporary $\mathcal{H}$-matrix $\mathbf{X}$ is needed as shown in (26), which increases memory usage.

The proposed LU-based direct solution has three components: (1) $\mathcal{H}$-based recursive LU factorization; (2) matrix solution by $\mathcal{H}$-based backward and forward substitution; and

$$\mathbf{Y}^{-1} = \begin{pmatrix} \mathbf{Y}_{11}^{-1} \oplus \mathbf{Y}_{11}^{-1} \otimes \mathbf{Y}_{12} \otimes \mathbf{S}^{-1} \otimes \mathbf{Y}_{21} \otimes \mathbf{Y}_{11}^{-1} & -\mathbf{Y}_{11}^{-1} \otimes \mathbf{Y}_{12} \otimes \mathbf{S}^{-1} \\ -\mathbf{S}^{-1} \otimes \mathbf{Y}_{21} \otimes \mathbf{Y}_{11}^{-1} & \mathbf{S}^{-1} \end{pmatrix} \tag{25}$$

(3) acceleration by nested dissection. The first two components have been developed in $\mathcal{H}$-matrix arithmetics [6], [7]. We will brief the first two, and focus on the third component.

1) Recursive LU Factorization: We use an $\mathcal{H}$-matrix block $\mathbf{Y}_{tt}$ to demonstrate the $\mathcal{H}$-LU factorization process, where t is a non-leaf cluster in the cluster tree $T_{\mathcal{I}}$. Since t is a non-leaf, block $t \times t$ is not a leaf block. Hence, $\mathbf{Y}_{tt}$ can be subdivided into four sub blocks:

$$\mathbf{Y}_{tt} = \begin{pmatrix} \mathbf{Y}_{t_1t_1} & \mathbf{Y}_{t_1t_2} \\ \mathbf{Y}_{t_2t_1} & \mathbf{Y}_{t_2t_2} \end{pmatrix} \tag{27}$$

where t_1 and t_2 are the children of t in the cluster tree $T_{\mathcal{I}}$.

Assuming $\mathbf{Y}$ can be factorized into $\mathbf{L}$ and $\mathbf{U}$ matrices, $\mathbf{Y}$ can also be written as

$$\begin{aligned}\mathbf{Y}_{tt} &= \mathbf{L}_{tt}\mathbf{U}_{tt} = \begin{pmatrix} \mathbf{L}_{t_1t_1} & 0 \\ \mathbf{L}_{t_2t_1} & \mathbf{L}_{t_2t_2} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{t_1t_1} & \mathbf{U}_{t_1t_2} \\ 0 & \mathbf{U}_{t_2t_2} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{L}_{t_1t_1}\mathbf{U}_{t_1t_1} & \mathbf{L}_{t_1t_1}\mathbf{U}_{t_1t_2} \\ \mathbf{L}_{t_2t_1}\mathbf{U}_{t_1t_1} & \mathbf{L}_{t_2t_1}\mathbf{U}_{t_1t_2} + \mathbf{L}_{t_2t_2}\mathbf{U}_{t_2t_2} \end{pmatrix}. \end{aligned} \tag{28}$$

By comparing (27) and (28), it can be seen that the LU factorization can be computed recursively as follows:

1) Compute $\mathbf{L}_{t1t1}$ and $\mathbf{U}_{t1t1}$ by $\mathcal{H}$-LU factorization $\mathbf{Y}_{t1t1} = \mathbf{L}_{t1t1}\mathbf{U}_{t1t1}$;
2) Compute $\mathbf{U}_{t1t2}$ by solving $\mathbf{L}_{t1t1}\mathbf{U}_{t1t2} = \mathbf{Y}_{t1t2}$;
3) Compute $\mathbf{L}_{t2t1}$ by solving $\mathbf{L}_{t2t1}\mathbf{U}_{t1t1} = \mathbf{Y}_{t2t1}$;
4) Compute $\mathbf{L}_{t2t2}$ and $\mathbf{U}_{t2t2}$ by $\mathcal{H}$-LU factorization $\mathbf{L}_{t2t2}\mathbf{U}_{t2t2} = \mathbf{Y}_{t2t2} - \mathbf{L}_{t2t1}\mathbf{U}_{t1t2}$; (29)

If $t \times t$ is a leaf block, $\mathbf{Y}_{tt}$ is not subdivided. It is stored in full matrix format, and factorized by a conventional pivoted LU factorization.

In Step 2), a matrix equation $\mathbf{L}_{tt}\mathbf{X}_{\mathrm{ts}} = \mathbf{Y}_{\mathrm{ts}}$ needs to be solved, where $\mathbf{L}_{tt}$ is a lower triangular matrix. In Step 3), $\mathbf{X}_{\mathrm{ts}}\mathbf{U}_{\mathrm{ss}} = \mathbf{Y}_{\mathrm{ts}}$ needs to be solved, where $\mathbf{U}_{tt}$ is an upper triangular matrix. These two are solved by recursive block forward and backward substitution based on $\mathcal{H}$ arithmetics.

2) Matrix Solution by Backward and Forward Substitution: After $\mathbf{Y}$ is factorized as $\mathbf{Y} = \mathbf{LU}$, FEM system $\mathbf{Y}\{E\} = \{I\}$ can be solved in two steps: 1) Solve the lower triangular system $\mathbf{L}\{x\} = \{I\}$; 2) Solve the upper triangular system $\mathbf{U}\{E\} = \{x\}$. In the first step, lower triangular system $\mathbf{L}_{tt}\{x_t\} = I_t$ is solved recursively by forward substitution as follows.

If $t \times t$ is not a leaf block, $\mathbf{L}_{tt}$ is subdivided and the lower triangular system can be written as

$$\begin{pmatrix} \mathbf{L}_{t_1t_1} & 0 \\ \mathbf{L}_{t_2t_1} & \mathbf{L}_{t_2t_2} \end{pmatrix} \begin{pmatrix} x_{t_1} \\ x_{t_2} \end{pmatrix} = \begin{pmatrix} I_{t_1} \\ I_{t_2} \end{pmatrix} \tag{30}$$

where t_1 and t_2 are the children of t in the cluster tree $T_{\mathcal{I}}$. We can write (30) as

$$\begin{pmatrix} \mathbf{L}_{t_1t_1}x_{t_1} \\ \mathbf{L}_{t_2t_1}x_{t_1} + \mathbf{L}_{t_2t_2}x_{t_2} \end{pmatrix} = \begin{pmatrix} I_{t_1} \\ I_{t_2} \end{pmatrix}. \tag{31}$$

By comparing both sides of (31), we obtain $\{x\}$ by:

1) solving x_{t1} from $\mathbf{L}_{t1t1}x_{t1} = I_{t1}$;
2) solving x_{t2} from $\mathbf{L}_{t2t2}x_{t2} = I_{t2} - \mathbf{L}_{t2t1}x_{t1}$.

If $t \times t$ is a leaf block, $\mathbf{L}_{tt}$ is not subdivided and x_t is solved by a conventional forward substitution. Note that different from the construction of $\mathcal{H}$-based $\mathbf{L}$, solving a lower triangular system $\mathbf{L}_{tt}\{x_t\} = I_t$ is exact without introducing any approximation. Solving the upper triangular system can be done in a similar way.

3) Acceleration by Nested Dissection: Our numerical experiments show that the advantage of the $\mathcal{H}$-based LU over the state-of-the-art sparse factorization such as UMFPACK is not that obvious since the latter incorporates the most advanced ordering technique, which almost minimizes the number of nonzeros to be processed. We hence further accelerate the $\mathcal{H}$-based LU factorization by nested dissection. It is known that the smaller the number of nonzeros to be processed in an LU process, the better the computational efficiency. Nested dissection [1] can be used as an ordering technique to reduce the number of nonzero blocks to be computed in LU factorization. In addition, this scheme naturally fits the $\mathcal{H}$-based framework compared to many other ordering techniques. It serves an efficient approach to construct a block cluster tree.

We divide the computational domain into three parts: two domain clusters D_1 and D_2 which do not interact with each other, and one interface cluster "I" which interacts with both domain clusters.

Since the domain clusters D_1 and D_2 do not have interaction, their crosstalk entries in the FEM matrix $\mathbf{Y}$ are all zero. If we order the unknowns in D_1 and D_2 first and the unknowns in I last, the resultant matrix will have large zero blocks as shown in

```
Recursive inverse algorithm($\mathbf{X}$ is used for temporary storage)
Procedure H-inverse($\mathbf{Y}, \mathbf{X}$)($\mathbf{Y}$ is input matrix, $\mathbf{X}$ is inverse)
 $If$ matrix $\mathbf{Y}$ is a non-leaf matrix block
   H-inverse ($\mathbf{Y}_{11}, \mathbf{X}_{11}$)
   $\mathbf{Y}_{21} \otimes \mathbf{X}_{11} \rightarrow \mathbf{X}_{21}, \mathbf{X}_{11} \otimes \mathbf{Y}_{12} \rightarrow \mathbf{X}_{12}, \mathbf{X}_{22} \oplus (-\mathbf{X}_{21} \otimes \mathbf{Y}_{12}) \rightarrow \mathbf{X}_{22}$
   H-inverse ($\mathbf{X}_{22}, (\mathbf{Y}^{-1})_{22}$)
$-(\mathbf{Y}^{-1})_{22} \otimes \mathbf{X}_{21} \rightarrow (\mathbf{Y}^{-1})_{21}, -\mathbf{X}_{12} \otimes (\mathbf{Y}^{-1})_{22} \rightarrow (\mathbf{Y}^{-1})_{12},\ \mathbf{X}_{11} \oplus (-(\mathbf{Y}^{-1})_{12} \otimes \mathbf{X}_{21}) \rightarrow (\mathbf{Y}^{-1})_{11}$
$else$
      Inverse ($\mathbf{Y}$) (normal full matrix inverse)
```

(26)

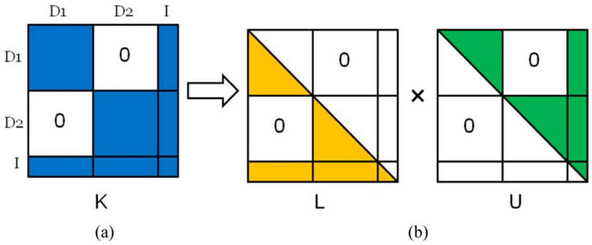


Fig. 2. (a) A nested dissection based partition. (b) Matrix patterns in LU factors.

Fig. 2(a). These zero blocks are preserved during the LU factorization as shown in Fig. 2(b), and hence the computation cost of LU factorization is reduced.

We further partition the domain clusters D_1 and D_2 into three parts. This process continues until the number of unknowns in each cluster is smaller than $leafsize$ ($n_{\min}$), or no interface edges can be found to divide the domain. Since the matrices in the non-zero blocks are stored and processed by $\mathcal{H}$-matrix techniques in the proposed direct solver, the computational complexity is significantly reduced compared to a conventional nested dissection based LU factorization.

E. Adaptive Truncation for Accurate Electrodynamic Analysis

As proved in Section III, the inverse of FEM matrix $\mathbf{Y}$ can be represented by an $\mathcal{H}$ matrix. However, which rank to use in the admissible blocks is unknown beforehand. In addition, the choice of rank for electrodynamic problems is more complicated compared to static problems. If a constant rank is used across the tree level of a block cluster tree, accuracy may not be guaranteed if the constant rank is too small. If the constant rank is chosen to be very large, the computational efficiency will be sacrificed since for many admissible blocks, a large rank may not be necessary. To address this issue, we developed an adaptive truncation scheme in the proposed direct solver, i.e., the rank for each admissible block is determined adaptively based on a required level of accuracy. The detail is given as follows.

In the original FEM matrix $\mathbf{Y}$, all the admissible blocks are zero and hence do not need to be stored. An admissible block becomes non-zero during the inverse/LU process when adding the sum of several matrices to this block or adding the product of two matrices to this block. To give an example, consider $\mathbf{M}_{t\times s} = \mathbf{M}^1_{t\times s} \oplus \mathbf{M}^2_{t\times s}$, where $\mathbf{M}^1_{t\times s} = \mathbf{A}_1\mathbf{B}_1^T$, $\mathbf{M}^2_{t\times s} = \mathbf{A}_2\mathbf{B}_2^T$ and they have the rank $k1$ and $k2$ respectively. The direct addition $\mathbf{M}^{'}_{t\times s} = \mathbf{M}^1_{t\times s} + \mathbf{M}^2_{t\times s} = \mathbf{A}_1\mathbf{B}_1^T + \mathbf{A}_2\mathbf{B}_2^T = [\mathbf{A}_1\mathbf{A}_2][\mathbf{B}_1\mathbf{B}_2]^T$ has rank $k1+k2$. To determine which rank is necessary, the singular value decomposition of $\mathbf{M}^{'}_{t\times s}$ is first performed:

$$\mathbf{M}^{'}_{t\times s} = \mathbf{U}^{'}\Sigma^{'}\mathbf{V}^{'T} \tag{32}$$

where $\mathbf{U}^{'}$ is a $|t| \times (k1+k2)$ matrix, $\mathbf{V}^{'}$ is a $|s| \times (k1+k2)$ matrix, and $\Sigma^{'}$ is a $(k1+k2)\times(k1+k2)$ diagonal matrix with diagonal entries: $\Sigma^{'}_{11} \geq \Sigma^{'}_{22} \geq \cdots \geq \Sigma^{'}_{(k1+k2)(k1+k2)} > 0$. We then truncate $\mathbf{M}^{'}_{t\times s}$ as

$$\mathbf{M}_{t\times s} = \mathbf{U}\Sigma\mathbf{V}^T \tag{33}$$

where $\mathbf{U} = \mathbf{U}^{'}|_{|t|\times k}$, $\mathbf{V} = \mathbf{V}^{'}|_{|s|\times k}$, $\Sigma = \mathrm{diag}(\Sigma_{11},\ldots,\Sigma_{kk})$, and k satisfies

$$\Sigma_{kk} > \Sigma^{'}_{11} * \varepsilon \text{ and } \Sigma_{k+1,k+1} \leq \Sigma^{'}_{11} * \varepsilon \tag{34}$$

where ε is the relative truncation error chosen based on the required level of accuracy. The adaptive truncation for adding the product of two matrices to an admissible block can be conducted in a similar fashion.

Unlike the fixed truncation scheme, the rank k here is not a constant. It is determined by the truncation accuracy of each admissible block adaptively. In case that the new rank k is larger than the original rank, the storage of $\mathbf{A}$ and $\mathbf{B}$ matrices need to be expanded to accommodate the larger rank. In addition, the singular value decomposition is performed by using $\mathcal{H}$-based arithmetics, which has a linear complexity of $O(k^2\max(|t|,|s|))$ [6].

V. Complexity and Accuracy Analysis

A. Complexity Analysis

In the following, we give a detailed complexity analysis for the inverse and LU factorization, which is different from what is reported in the literature for $\mathcal{H}$-based inverse and LU factorization [6]. The latter is based on analogy without accounting for the actual number of operations. In addition, the proposed complexity analysis takes electrodynamic problems into consideration.

Before proceeding to the detail, we introduce an important parameter, sparsity constant c_{sp}, which is used extensively in the complexity analysis. Defining the number of blocks $t \times s \in T_{\mathcal{I}\times\mathcal{J}}$ associated with a given cluster $t \in T_{\mathcal{I}}$ by

$$c^r_{sp}(T_{\mathcal{I}\times\mathcal{J}}, t) := |\{s \subset \mathcal{J} : t \times s \in T_{\mathcal{I}\times\mathcal{J}}\}| \tag{35}$$

and that associated with $s \in T_{\mathcal{J}}$ by

$$c^c_{sp}(T_{\mathcal{I}\times\mathcal{J}}, s) := |\{t \subset \mathcal{I} : t \times s \in T_{\mathcal{I}\times\mathcal{J}}\}| \tag{36}$$

the sparsity constant c_{sp} of $T_{\mathcal{I}\times\mathcal{J}}$ is defined as

$$c_{sp}(T_{\mathcal{I}\times\mathcal{J}}) := \max\left\{\max_{t\in T_{\mathcal{I}}} c^r_{sp}(T_{\mathcal{I}\times\mathcal{J}}, t),\ \max_{s\in T_{\mathcal{J}}} c^c_{sp}(T_{\mathcal{I}\times\mathcal{J}}, s)\right\}. \tag{37}$$

Despite a complicated mathematical definition, graphically, c_{sp} is the maximum number of links that can exist in each tree level in Fig. 1(a).

1) Inverse Complexity: The procedure $\mathcal{H}$-inverse shown in (26) can be divided into two sub procedures to analyze its complexity: 1) H-inverse_M, which only performs $\mathcal{H}$-based multiplications; 2) H-inverse_A, which only performs $\mathcal{H}$-based additions.

In sub-procedure H-inverse_M, each leaf block cluster $r \times t$ is computed twice. Each computation is performed by

$$\mathbf{Y}_{r\times t} = \sum_{l=0}^{p} \sum_{s\in S(r\times t,l)} \mathbf{Y}_{r\times s}\cdot\mathbf{Y}_{s\times t} \tag{38}$$

where $S(r \times t, l) = \{s \in T_{\mathcal{I}} | \mathcal{F}^l(r) \times s \in T_{\mathcal{I}\times\mathcal{I}},\ s \times \mathcal{F}^l(t) \in T_{\mathcal{I}\times\mathcal{I}}$ and at least one of the two is a leaf$\}$. $\mathcal{F}^l(r)$ represents the parent block clusters of cluster r in level l. If $r \times s$ is a leaf block, it is either an admissible leaf or an inadmissible leaf. If $r \times s$ is admissible, its corresponding matrix block is stored as $\mathbf{M}_{r\times s} = \mathbf{AB}^T$, where $\mathbf{A}$ is a $|r| \times k$ matrix and $\mathbf{B}$ is a $|s| \times k$ matrix. The product of $\mathbf{M}_{r\times s}$ and block cluster $s \times t$ is an admissible matrix block $\mathbf{M}_{r\times t} = \mathbf{AC}^T$, where $\mathbf{C}$ is computed by multiplying $\mathbf{B}^T$ by block cluster $s \times t$, which involves k $\mathcal{H}$-matrix-vector multiplications and hence has the complexity of $kC_{sp}k_1O(\max\{|s|, |t|\}\log(\max\{|s|, |t|\}))$, where

$$k_1 = \max\{k, n_{\min}\}. \tag{39}$$

If $r \times s$ is inadmissible, its matrix size is at most $n_{\min} \times n_{\min}$. So the multiplication with block cluster $s \times t$ involves at most $n_{\min}$ $\mathcal{H}$-matrix-vector multiplications and has $n_{\min}C_{sp}k_1O(\max\{|s|, |t|\}\log(\max\{|s|, |t|\}))$ complexity. If the block cluster tree is balanced, in level l, $\max\{|s|, |t|\}$ can be approximated by $N/2^l$. Therefore, overall, the complexity of multiplying $r \times s$ by $s \times t$ is

$$Complexity((r \times s) \otimes (s \times t)) \leq k_1^2 C_{sp} O\left(\frac{N}{2^l}\log\left(\frac{N}{2^l}\right)\right). \tag{40}$$

The complexity of Hinverse_M can then be obtained by summing the cost for multiplying $r \times s$ and $s \times t$ in each level:

$$\begin{aligned} &Complexity(\text{H-inverse\_M}) \\ &\quad\leq 2\sum_{l=0}^{p}\sum_{r\times s\in\mathcal{L}(T,l)}\sum_{s\times t\in T^{(l)}} N_{(r\times s)\otimes(s\times t)} \\ &\quad+ 2\sum_{l=0}^{p}\sum_{s\times t\in\mathcal{L}(T,l)}\sum_{r\times s\in T^{(l)}} N_{(r\times s)\otimes(s\times t)} \\ &\quad\leq 4\sum_{l=0}^{p}\sum_{r\times s\in\mathcal{L}(T,l)}\sum_{s\times t\in T^{(l)}} k_1^2 C_{sp} O\left(\frac{N}{2^l}\log\left(\frac{N}{2^l}\right)\right) \end{aligned} \tag{41}$$

where $\mathcal{L}(T, l)$ denotes the set of leaves in block cluster tree T in level l. Since the number of blocks satisfying $r \times s \in \mathcal{L}(T, l)$ for certain cluster s is smaller than C_{sp}, and there are at most $2^l C_{sp}$ block clusters in level l, we have

$$\begin{aligned} &Complexity(\text{H-inverse\_M}) \\ &\quad\leq 4\sum_{l=0}^{p}\sum_{s\times t\in T^{(l)}} C_{sp}k_1^2 C_{sp} O\left(\frac{N}{2^l}\log\left(\frac{N}{2^l}\right)\right) \\ &\quad\leq 4\sum_{l=0}^{p} 2^l C_{sp}k_1^2 C_{sp}^2 O\left(\frac{N}{2^l}\log\left(\frac{N}{2^l}\right)\right) \\ &\quad\leq 4k_1^2 C_{sp}^3 O\left(N\sum_{l=0}^{\log N}(\log N - l)\right) \\ &\quad= 2k_1^2 C_{sp}^3 O(N\log N(\log N + 1)) \\ &\quad\sim O\left(k_1^2 N \log^2(N)\right). \end{aligned} \tag{42}$$

As for the complexity of H-inverse_A, since the complexity of formatted addition is $C_{sp}k_1^2O(N\log N)$ [6], the complexity of Hinverse_A can be obtained by adding the cost of formatted addition level by level as the following:

$$\begin{aligned} &Complexity(\text{H-inverse\_A}) \\ &\quad\leq 2\sum_{l=0}^{p}\sum_{r\times t\in T^{(l)}} C_{sp}k_1^2 \\ &\qquad\times O(\max\{|r|, |t|\}\log(\max\{|s|, |t|\})) \\ &\quad\leq 2C_{sp}^2 k_1^2 \sum_{l=0}^{p} 2^l O\left(\frac{N}{2^l}\log\frac{N}{2^l}\right) \\ &\quad= 2C_{sp}^2 k_1^2 O\left(N\sum_{l=0}^{p}(\log N - l)\right) \\ &\quad\leq 2C_{sp}^2 k_1^2 O\left(N\sum_{l=0}^{\log N}(\log N - l)\right) \\ &\quad= C_{sp}^2 k_1^2 O(N\log N(\log N + 1)) \\ &\quad\sim O\left(k_1^2 N\log^2 N\right) \end{aligned} \tag{43}$$

Therefore, the total complexity of inverse is

$$\begin{aligned} Complexity(\mathbf{Y}^{-1}) &= Complexity(\text{H-inverse\_M}) + \\ &\quad Complexity(\text{H-inverse\_A}) \sim O(k_1^2 N\log^2 N). \end{aligned} \tag{44}$$

2) LU Factorization and Solution Complexity: As can be seen from (29), the LU factorization of $\mathbf{Y}_{tt}$ is computed in four steps. In these four steps, $\mathbf{Y}_{t1t1}$, $\mathbf{Y}_{t1t2}$, and $\mathbf{Y}_{t2t1}$ are computed once, $\mathbf{Y}_{t2t2}$ is computed twice. Since in inverse, each block is computed twice, the complexity of $\mathcal{H}$-based LU factorization is bounded by $\mathcal{H}$-based-inverse, which is $O(k^2N\log^2 N)$.

After obtaining the $\mathcal{H}$-LU factorization, the FEM system is solved by the algorithm outlined in Section IV-D2. Since the matrix entries are stored in the leaf block clusters, matrix solving is done in the leaf block clusters similar to $\mathcal{H}$-matrix based matrix-vector multiplication. If the diagonal leaf block $t \times t$ is inadmissible, full matrix forward and backward substitutions are performed to solve $\mathbf{L}_{tt}x_t = b_t$, which requires $O(|t|^2)$ operations. If the off-diagonal leaf block $t \times s$ is inadmissible, full matrix-vector multiplication is performed, which requires $O(|t||s|)$ operations. If the off-diagonal leaf block $t \times s$ is admissible, the matrix is stored in a factorized form: $\mathbf{M}_{t\times s} = \mathbf{AB}^T$, which requires $kO(|t| + |s|)$ operations. The total complexity of matrix solving is hence

$$\begin{aligned} &Complexity(\text{LU\_Solve}) \\ &\quad= \sum_{t\times s\in\mathcal{L}^-} O(|t||s|) + \sum_{t\times s\in\mathcal{L}^+} kO(|t| + |s|) \\ &\quad\leq Storage(\text{an } \mathcal{H} \text{ matrix of rank } k) \\ &\quad\sim O(kN\log N) \end{aligned} \tag{45}$$

where $\mathcal{L}^-$ denotes all the inadmissible leaves, and $\mathcal{L}^+$ denotes all the admissible leaves.

B. Accuracy Analysis

From the proof developed in Section III, there exists an $\mathcal{H}$-matrix-based representation of the inverse of the FEM matrix $\mathbf{Y}$. In such a representation, which block is admissible and which block is inadmissible are determined by an admissibility condition. Rigorously speaking, this admissibility condition should be determined based on $\mathbf{Y}^{-1}$. However, since $\mathbf{Y}^{-1}$ is unknown, we determine it based on $\mathbf{Y}$. Apparently, this will induce error. However, as analyzed in Section III, the $\mathbf{Y}$'s inverse can be mapped to the dense matrix formed for an integral operator. For this dense matrix, the admissibility condition used to construct an $\mathcal{H}$-matrix representation has the same form as (7) as shown in [14]–[16]. Thus, the $\mathcal{H}$-matrix structure, i.e., which block can have a potential low-rank approximation and which block is a full matrix, is formed correctly for Y^{-1}. In addition, the accuracy of the admissibility condition (7) can be controlled by η.

In the inverse and LU factorization process, the rank of each admissible block is adaptively determined based on the accuracy requirement as shown in Section IV-E. If the rank is determined to be a full rank based on the adaptive truncation scheme, then a full rank will be used. Thus, the low-rank approximation for each admissible block is also error controllable through parameter ε used in the adaptive truncation scheme.

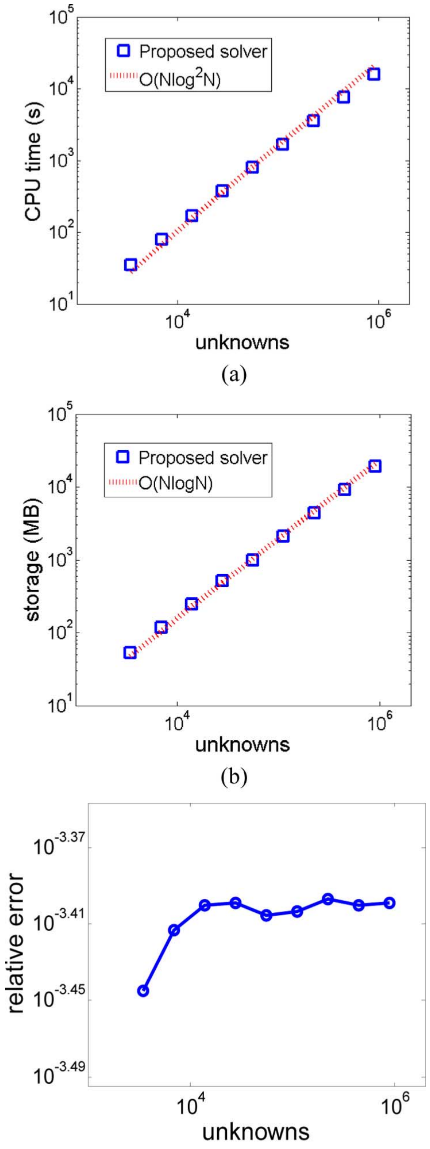
Based on the aforementioned two facts, the error of the proposed direct solver is controllable.

VI. Choice of Simulation Parameters

There are only three parameters to choose in the proposed direct solver: η in (7), $n_{\min}$ (leafsize), and ε in (34) for adaptively determining the rank. The smaller η is, the better the accuracy. However, the computation will become inefficient if η is too small. For all the electrodynamic simulations conducted in this work, we choose $\eta = 1$. The parameter ε can be chosen based on a required level of accuracy. For example, ε can be set to 10^{-4} if 0.01% error is required. As for leafsize $n_{\min}$, if it is chosen to be too large, on one hand, the accuracy becomes better; on the other hand, larger full matrix blocks will be formed, and hence computation becomes slow. Therefore, we determine the leafsize $n_{\min}$ by balancing CPU time and error. In the simulation conducted in this work, $n_{\min}$ is in the range of (10, 50).

VII. Numerical Results

To demonstrate the accuracy and almost linear complexity of the proposed direct FEM solver, we simulated a number of static and electrodynamic examples from small unknowns to over one million unknowns, from small electric sizes to more than sixty wavelengths.



(c)

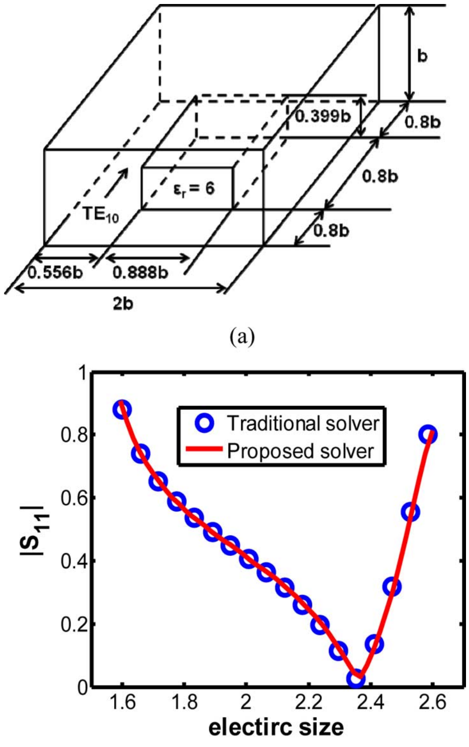
Fig. 3. Performance of the proposed direct inverse in simulating a shielded bus structure. (a) CPU time for computing $\mathbf{Y}^{-1}$. (b) Storage of $\mathbf{Y}^{-1}$. (c) Relative error of the inverse.

A. Shielded Bus Structure

A shielded microstrip line [13, pp. 115-116] was first simulated to demonstrate the feasibility of the proposed solver in static electromagnetic applications. Node-based triangular basis functions were used. The proposed direct inverse was used to simulate this example. The simulation parameters were chosen as $n_{\min} = 10$ and $\eta = 2$. A fixed rank $k = 4$ was used for such a static simulation. To test the large-scale modeling capability of the proposed direct solver, we increased the size of the original problem by adding more lines parallel to the original microstrip line, resulting in 23 K unknowns to 0.8 million unknowns. In Fig. 3(a) and (b), we plot the CPU time and storage of the proposed direct FEM solver with respect to the number of unknowns. The time complexity and storage complexity show an excellent agreement with our theoretical prediction represented by the dashed line, which shows a memory complexity of $O(N \log N)$, and a time complexity of $O(N \log^2 N)$. Meanwhile, good accuracy is achieved in the entire range as can be seen from Fig. 3(c). The relative error in Fig. 3(c) is measured by the inverse error $\left\| I - \mathbf{Y}_{\mathbf{H}}^{-1}\mathbf{Y} \right\|_F / \|I\|_F$, which is less than 0.5% in the entire range.

B. Waveguide Discontinuity

The validity of the proposed solver in solving electrodynamic problems was first demonstrated by a dielectric-loaded waveguide problem shown in Fig. 4(a) ([13, p. 202]). The rectangular



(a)

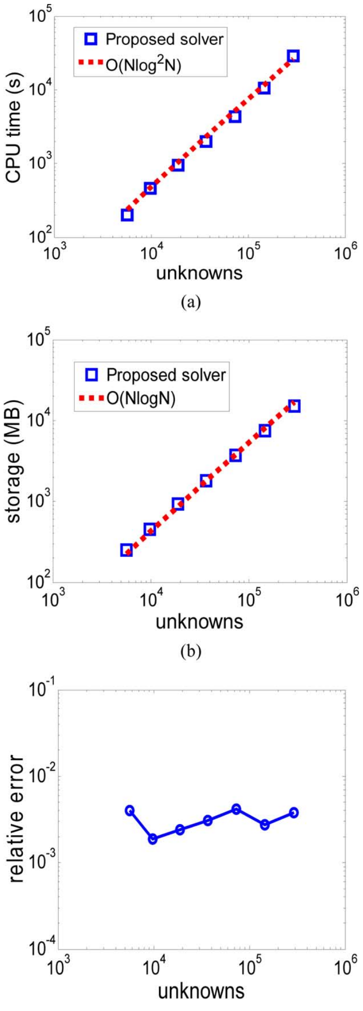
(b)

Fig. 4. (a) Illustration of the dielectric-loaded waveguide. (b) $|S_{11}|$ simulated by traditional and proposed solvers.

waveguide was loaded by a dielectric obstacle with $\varepsilon_r = 6$. The computational domain was discretized by prism elements. The vector prism basis functions [13] were used to expand the unknown $\mathbf{E}$ in each element. The mesh size was chosen to be 1/25 of the wavelength. The proposed direct inverse was used to simulate this example. The simulation parameters were chosen as $n_{\min} = 50$ and $\eta = 1$. The rank k varied from 4 to 6. In Fig. 4(b), we plotted $|S_{11}|$ computed using the proposed direct solver with respect to electric size. An excellent agreement with the reference result [13] computed using a traditional FEM solver is observed.

To test the large-scale modeling capability of the proposed direct inverse, we increased the size of the original problem by increasing the length of the waveguide as well as the loaded dielectric rod. The length was increased from 4.8 b to 256.8 b, resulting in an electric size from ~ 1.2 wavelengths to ~ 64 wavelengths. The number of unknowns increased from 5.63 K to 0.3 M. In Fig. 5, the CPU time and memory cost are plotted as a function of the number of unknowns. Once again, the time complexity and storage complexity of the proposed solver agree very well with the theoretical prediction which is plotted in dashed line. Moreover, a constant order of accuracy is achieved in the entire range. The relative inverse error $\left\|I - \mathbf{Y}_{\mathbf{H}}^{-1}\mathbf{Y}\right\|_F / \|I\|_F$ is less than 1.5% in the entire range. Note that in our simulation, to test the general capability of the proposed solver, we did not take advantage of the fact that the unknowns are increased only along one dimension in this typical example. Otherwise, the complexity can be further reduced to linear [26].

We also used UMFPACK 5.0 [12], a state-of-the-art sparse matrix solver that incorporates most advanced multifrontal and ordering techniques, to simulate the 0.3 M unknown problem. It takes UMFPACK ~ 4.8 s to solve one column of the inverse of the FEM matrix, and the time to compute the entire inverse is approximately $4.8\ \mathrm{s} \times 0.3\ \mathrm{M} \approx 1.4\ \mathrm{Ms}$. If we store all the computed columns of the inverse matrix, UMFPACK soon fails due to the lack of memory. In contrast, the proposed solver only takes 26 Ks to compute the entire inverse with relative error no greater than 1.5%, and memory usage no greater than 15 GB.



(a)

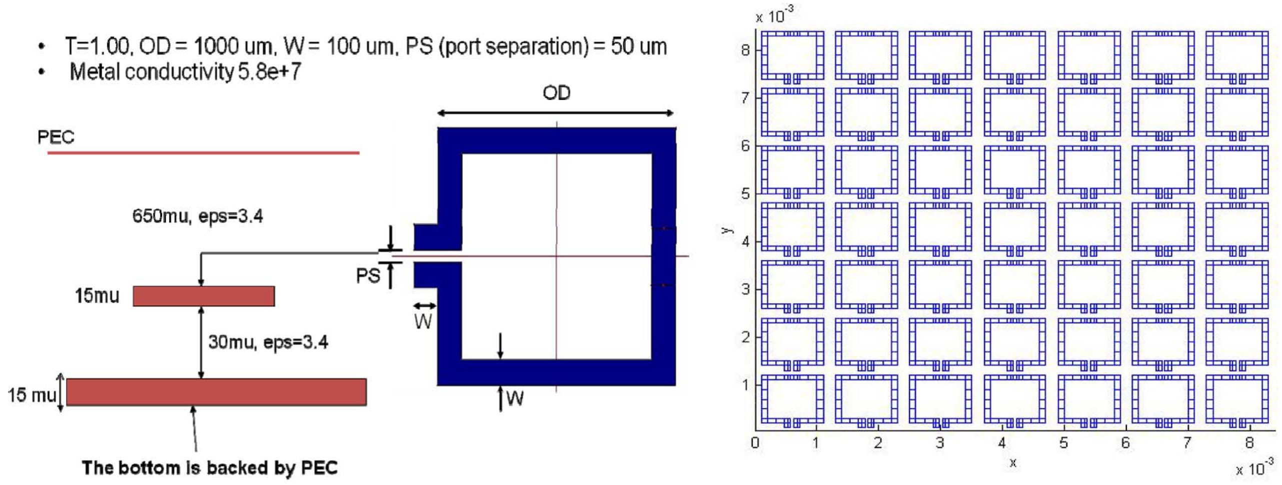
(b)

(c)

Fig. 5. Performance of the proposed direct inverse in simulating a dielectric-loaded waveguide from 1.2 wavelengths to 64 wavelengths. (a) CPU time for computing $\mathbf{Y}^{-1}$. (b) Storage of $\mathbf{Y}^{-1}$. (c) Inverse error.

C. Inductor Array

A large-scale package inductor array was simulated to demonstrate the accuracy and efficiency of the proposed $\mathcal{H}$-LU-based direct solver accelerated by nested dissection. The geometry and material data of each inductor is shown in Fig. 6(a), and a 7 by 7 inductor array is shown in Fig. 6(b). We simulated a series of inductor arrays from a 2 by 2 array to a 7 by 7 array, the number of unknowns of which ranged from 117,287 to 1,415,127. The simulation parameters were chosen as $n_{\min} = 32$ and $\eta = 1$. The adaptive truncation with $\varepsilon = 10^{-4}$ was used to adaptively determine the rank for each admissible block. In Table I, we gave the rank distribution with

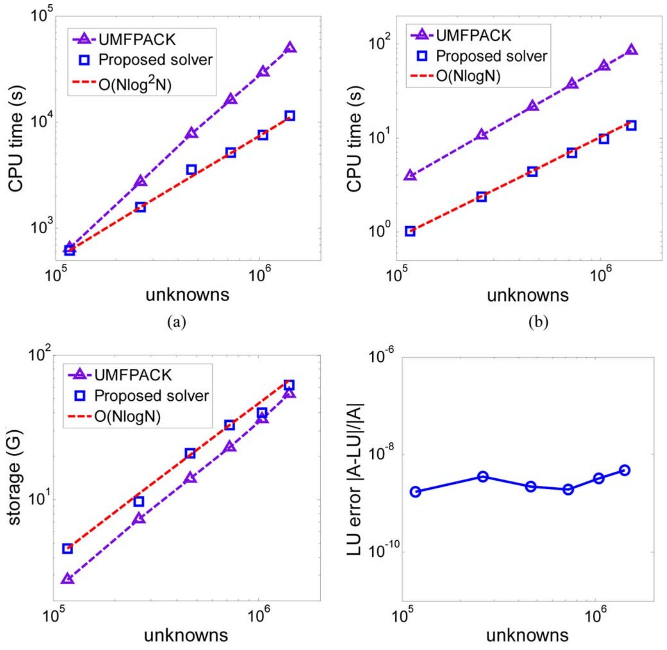

• T=1.00, OD = 1000 um, W = 100 um, PS (port separation) = 50 um
• Metal conductivity 5.8e+7
PEC
OD
650mu, eps=3.4
PS
15mu
W
30mu, eps=3.4
15 mu
W
The bottom is backed by PEC
x 10^-3
y
x
x 10^-3
0
1
2
3
4
5
6
7
8

(a) (b)

Fig. 6. Illustration of an inductor array. (a) Geometrical and material detail of one inductor. (b) A 7 × 7 inductor array.

TABLE I
RANK DISTRIBUTION ACROSS THE TREE LEVEL FOR A 7 × 7 INDUCTOR ARRAY THAT HAS 1,415,127 UNKNOWNS

Tree Level	Minimum k	Maximum k
1	No admissible blocks	
2		
3		
4	2	3
5	2	8
6	1	9
7	3	18
8	3	20
9	6	9
10	2	16
11	1	12
12	1	74
13	1	70
14	1	20
15	1	20
16	1	20
17	1	20
18	1	31
19	1	35
20	4	20
21	5	20



(a) (b) (c) (d)

Fig. 7. Performance of the proposed LU-based direct solver for simulating an inductor array. (a) CPU time for LU factorization. (b) CPU time for solving one right hand side. (c) Storage. (d) Accuracy.

respect to tree level observed in the simulation of the 7×7 inductor example that involved more than 1 million unknowns. As can be seen from Table I, the rank k fluctuates across all the tree levels. In Table I, the smaller the tree level, the closer it is to the root cluster, which is at level 0. The minimum rank denotes the smallest rank present in the admissible block in a tree level; and the maximum rank denotes the largest rank present in the admissible block in the same level. It can be seen that even in the same tree level, the required rank for each admissible block is different to achieve the same level of accuracy. However, overall, the rank k is a small number compared to the number of unknowns. We also compared the rank distribution between different problem sizes. For example, for a 3×3 inductor array, with the same ε, the maximum rank was 83, which appeared at level 11. The minimum rank was 1.

In Fig. 7(a), we plot LU factorization time cost by the proposed direct solver, and that cost by UMFPACK 5.0 with respect to the number of unknowns. The proposed solver demonstrates a complexity of $O(N \log^2 N)$, which agrees very well with theoretical analysis, whereas UMFPACK has a much higher complexity. In Fig. 7(b), we plot the matrix solution time of the proposed direct solver, and that of UMFPACK for one right hand side. Once again, the proposed direct solver outperforms UMFPACK. In addition, the proposed direct solver is shown to have an $O(N \log N)$ complexity in matrix solution (backward and forward substitution). In Fig. 7(c), we plot the storage requirement of the proposed direct solver and that of UMFPACK in simulating this example. Even though the storage of the proposed solver is shown to be a little bit higher than that of UMFPACK, the complexity of the proposed solver is lower, and hence for larger number of unknowns, the proposed solver will outperform UMFPACK in storage. In Fig. 7(d), we plot the relative error of the proposed $\mathcal{H}$-LU-based direct solver accelerated by nested dissection. Good accuracy is observed in the entire range.

VIII. Conclusions

In this work we introduced the $\mathcal{H}$ matrix as a mathematical framework to develop fast solvers for direct FEM-based analysis of electromagnetic problems. We proved the existence of the $\mathcal{H}$-matrix-based representation of the FEM matrix and its inverse for electrodynamic problems, thus laid a theoretical foundation for developing error-controlled $\mathcal{H}$-based solutions for fast direct FEM-based analysis of electrodynamic problems.

Both inverse-and LU-based direct solutions were developed. Accuracy was controlled by adaptively determining the rank k for each admissible block based on required accuracy. The computation and storage complexity were shown to be $O(k^2 N \log^2 N)$, and $O(kN \log N)$ respectively by both theoretical analysis and numerical experiments. Since k is a small parameter that is adaptively determined by accuracy requirement, we have observed $O(N \log^2 N)$ time complexity and $O(N \log N)$ memory complexity with a constant order of accuracy across a wide range of unknowns and electric sizes. The LU-based solution was further accelerated by nested dissection based ordering. A comparison with the state-of-the-art direct FEM solution that employs the most advanced sparse matrix solver such as UMFPACK has shown a clear advantage of the proposed solver. Moreover, existing sparse solvers such as UMFPACK cannot afford to computing a direct inverse because storing each column of the inverse is not feasible for large matrices, whereas the proposed solver can store the dense inverse in $O(N \log N)$ units.

The proposed direct FEM solver of almost linear complexity and controlled accuracy is applicable to general problems involving arbitrarily-shaped geometries and non-uniform materials. It has been successfully applied to both electrostatic and electrodynamic problems involving millions of unknowns. More electrodynamic applications will be explored in the future.

References

[1] A. George, "Nested dissection of a regular finite element mesh," *SIAM J. Numerical Analysis*, vol. 10, no. 2, pp. 345–363, Apr. 1973.

[2] J.-S. Choi, T. C. Kramer, R. J. Adams, and F. X. Canning, "Factorization of finite element matrices using overlapped localizing LOGOS modes," in *IEEE Int. Symp. Antennas and Propagation*, 2008, pp. 4–4.

[3] J. Choi, R. J. Adams, and F. X. Canning, "Sparse factorization of finite element matrices using overlapped localizing solution modes," *Microw. Opt. Technol. Lett.*, vol. 50, no. 4, pp. 1050–1054, 2008.

[4] W. Hackbusch and B. Khoromaskij, "A Sparse Matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices," *Computing*, vol. 62, pp. 89–108, 1999.

[5] W. Hackbusch and B. N. Khoromskij, "A sparse $\mathcal{H}$-matrix arithmetic. Part II: Application to multi-dimensional problems," *Computing*, vol. 64, pp. 21–47, 2000.

[6] S. Borm, L. Grasedyck, and W. Hackbusch, *Hierarchical Matrices*, Lecture note 21 of the Max Planck Institute for Mathematics in the Sciences, 2003.

[7] L. Grasedyck and W. Hackbusch, "Construction and arithmetics of $\mathcal{H}$-matrices," *Computing*, vol. 70, no. 4, pp. 295–344, Aug. 2003.

[8] M. Bebendorf and W. Hackbusch, "Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with $L\infty$-coefficients," *Numerische Mathematik*, vol. 95, pp. 1–28, 2003.

[9] H. Liu, W. Chai, and D. Jiao, "An $\mathcal{H}$-matrix-based fast direct solver for finite-element-based analysis of electromagnetic problems," in *2009 Int. Annu. Rev. Progress in Applied Computational Electromagnetics (ACES)*, Mar. 2009, pp. 5–5.

[10] H. Liu and D. Jiao, "A direct finite-element-based solver of significantly reduced complexity for solving large-scale electromagnetic problems," in *Int. Microwave Symp. (IMS)*, Jun. 2009, p. 4.

[11] H. Liu and D. Jiao, "Performance analysis of the H-matrix-based fast direct solver for finite-element-based analysis of electromagnetic problems," in *Proc. IEEE Int. Symp. Antennas and Propagation*, Jun. 2009, p. 4.

[12] UMFPACK5.0 [Online]. Available: http://www.cise.ufl.edu/research/sparse/umfpack/

[13] J. M. Jin, *The Finite Element Method in Electromagnetics*, 2nd ed. New York: Wiley, 2002, pp. 442–442.

[14] W. Chai and D. Jiao, "$\mathcal{H}$-and $\mathcal{H}^2$-matrix-based fast integral-equation solvers for large-scale electromagnetic analysis," in *IET Microw., Antennas Propagat*, 2009 [Online]. Available: http://cobweb.ecn.purdue.edu/~djiao/publications.html#journal

[15] 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 Propagat.*, vol. 57, pp. 3147–3159, Oct. 2009.

[16] W. Chai and D. Jiao, "An $\mathcal{H}^2$-matrix-based integral-equation solver of linear-complexity for large-scale full-wave modeling of 3-D circuits," in *Proc. IEEE 17th Conf. Electrical Performance of Electronic Packaging (EPEP)*, Oct. 2008, pp. 283–286.

[17] J. Shaeffer, "Direct solve of electrically large integral equations for problem sizes to 1 M unknowns," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 8, pp. 2306–2313, Aug. 2008.

[18] V. Rokhlin, "Rapid solution of integral equations of classic potential theory," *J. Comput. Phys.*, vol. 60, pp. 187–207, Sep. 1985.

[19] W. Chew, J. Jin, C. Lu, E. Michiielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*, W. C. Chew, J. M. Jin, E. Michielssen, and J. M. Song, Eds. Norwood, MA: Artech House, 2001.

[20] K. Nabors and J. White, "FastCap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, pp. 1447–1459, Nov. 1991.

[21] W. Shi, J. Liu, N. Kakani, and J. Yu, "A fast hierarchical algorithm for three-dimensional capacitance extraction," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 3, pp. 330–336, Mar. 2002.

[22] W. Hackbusch, B. Khoromskij, and S. Sauter, "On $\mathcal{H}^2$-matrices," in *Lecture on Applied Mathematics*, H. Bungartz, R. Hoppe, and C. Zenger, Eds. Munich, Germany: Springer, 2000, pp. 9–29.

[23] P. Lancaster, "Lambda-matrices and vibrating systems," in *Ordinary Differential Equations With Constant Coefficients*. Oxford, U.K.: Pergamon Press, 1966, ch. 6, pp. 100–114.

[24] P. André, *Vibration Control of Active Structures: An Introduction*, 2nd ed. , The Netherlands: Kluwer Academic Publishers, 2002, pp. 18–19.

[25] F. Tisseur and K. Meerbergen, "The quadratic eigenvalue problem," *SIAM Review*, vol. 43, no. 2, pp. 235–286, 2001.

[26] H. Liu and D. Jiao, "Layered $\mathcal{H}$-matrix based direct matrix inversion of significantly reduced complexity for finite-element-based large-scale electromagnetic analysis," in *Proc. 2010 IEEE Int. Symp. Antennas and Propagation*, Jul. 2010, p. 4.

Haixin Liu received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006. He is currently pursuing the Ph.D. degree in the School of Electrical and Computer Engineering and works in the On-Chip Electromagnetics Group at Purdue University, West Lafayette, IN.

His research interests include computational electromagnetics, high-performance VLSI CAD, fast numerical methods for very large scale IC and package problems.

Dan Jiao (S'00–M'02–SM'06) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign in 2001.

She then worked at Technology CAD Division at the Intel Corporation until September 2005 as Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. In September 2005, she joined Purdue University, West Lafayette, IN, as an Assistant Professor in the School of Electrical and Computer Engineering. In 2009, she was promoted to Associate Professor with tenure. She has authored two book chapters and over 100 papers in refereed journals and international conferences. Her current research interests include computational electromagnetics, high-frequency digital, analogue, mixed-signal, and RF IC design and analysis, high-performance VLSI CAD, modeling of micro-and nano-scale circuits, applied electromagnetics, fast and high-capacity numerical methods, fast time domain analysis, scattering and antenna analysis, RF, microwave, and millimeter wave circuits, wireless communication, and bio-electromagnetics.

Dr. Jiao received the Ruth and Joel Spira Outstanding Teaching Award in 2010. In 2008, she received the NSF CAREER Award. In 2006, she received the Jack and Cathie Kozik Faculty Start-up Award, which recognizes an outstanding new faculty member in Purdue ECE. She also received an ONR Award through Young Investigator Program in 2006. In 2004, she received the Best Paper Award from Intel's annual corporate-wide technology conference (Design and Test Technology Conference) for her work on generic broadband model of high-speed circuits. In 2003, she won the Intel Logic Technology Development (LTD) Divisional Achievement Award in recognition of her work on the industry-leading BroadSpice modeling/simulation capability for designing high-speed microprocessors, packages, and circuit boards. She was also awarded the Intel Technology CAD Divisional Achievement Award for the development of innovative full-wave solvers for high-frequency IC design. In 2002, she was awarded by Intel Components Research the Intel Hero Award (Intel-wide, she was the tenth recipient) for the timely and accurate two- and three-dimensional full-wave simulations. She also won the Intel LTD Team Quality Award for her outstanding contribution to the development of the measurement capability and simulation tools for high-frequency on-chip cross-talk. She was the winner of the 2000 Raj Mittra Outstanding Research Award given her by the University of Illinois at Urbana-Champaign. She has served as a reviewer for many IEEE journals and conferences.