INVITED PAPER

Direct Matrix Solution of Linear Complexity for Surface Integral-Equation-Based Impedance Extraction of Complicated 3-D Structures

The authors of this paper develop a low-complexity matrix solution to solve a surface integral equation for extracting the impedance of 3-D nonideal conductors embedded in a dielectric material.

By Wenwen Chai and Dan Jiao, *Senior Member IEEE*

ABSTRACT | We develop a linear-complexity direct matrix solution for the surface integral equation (IE)-based impedance extraction of arbitrarily shaped 3-D nonideal conductors embedded in a dielectric material. A direct inverse of a highly irregular system matrix composed of both dense and sparse matrix blocks is obtained in $O(N)$ complexity with N being the matrix size. It outperforms state-of-the-art impedance solvers, be they direct solvers or iterative solvers, with fast central processing unit (CPU) time, modest memory consumption, and without sacrificing accuracy, for both small and large number of unknowns. The inverse of a 2.68-million-unknown matrix arising from the extraction of a large-scale 3-D interconnect having 128 buses, which is a matrix solution for 2.68 million right-hand sides, was obtained in less than 1.5 GB memory and 1.3 h on a single CPU running at 3 GHz.

KEYWORDS | Direct solvers; electromagnetic analysis; fast integral equation (IE) solvers; impedance extraction; interconnects

Manuscript received June 15, 2011; revised October 31, 2011 and January 20, 2012; accepted February 26, 2012. Date of publication June 1, 2012; date of current version January 16, 2013. This work was supported by a grant from SRC (Task 1292.073), and grants from the National Science Foundation (NSF) under Awards 0747578 and 0702567.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: wchai@purdue.edu; djiao@purdue.edu).

Digital Object Identifier: 10.1109/JPROC.2012.2190577

I. INTRODUCTION

With the increase in the processing power of the central processing unit (CPU), the memory and system interconnect links connected to a CPU need to have an exponentially increased bandwidth in order to fully utilize the computing power. This leads to higher speed signals on each data line as well as an increase in the number of data lines. It also becomes necessary to move chips closer to each other by revolutionary technologies such as 3-D stacking via through silicon vias (TSVs), etc. Enabling higher bandwidth brings significant challenges to the analysis and design of interconnects. To address these challenges, a full-wave modeling technology is required that can rapidly characterize the interaction between a large number of I/Os in the face of large problem sizes.

Existing fast solvers for solving large-scale circuit problems are, in general, iterative solvers since traditional direct solvers are computationally expensive. Among these iterative solvers, representative are fast multipole-based methods [1], [2], [34], fast low-rank compression methods [3], [4], hierarchical algorithms [5]–[7], and fast Fourier transform (FFT)-based methods [8], [9], which have dramatically reduced the memory and CPU time of the iterative solution of the dense system matrix resulting from an integral equation (IE)-based analysis.

The optimal complexity of an iterative solver is $O(N_{\mathrm{rhs}}N_{\mathrm{it}}N)$, where N_{rhs} is the number of right-hand

sides, N_{it} is the number of iterations, and N is the matrix size. To analyze the interaction between high-bandwidth interconnects, the number of right-hand sides is proportional to the I/O count. When the I/O count is large, iterative solvers become inefficient since an entire iteration procedure has to be repeated for each I/O (port).

There has been much recent progress in direct solvers [10]–[13], [17], [35], [36] for solving both circuit problems that typically are electrically small or moderate and scattering problems that typically are electrically large. In [10], an IE-based dense matrix is rendered sparse by applying multilevel multipole expansions, and the resultant sparse matrix is solved by a conventional sparse matrix solver for circuit parameter extraction of microelectronic structures. In [35], [36], a characteristic basis function method is developed for reducing the size of the system matrix arising in the analysis of monolithic microwave integrated circuits and scattering problems. In [12], a local-global solution (LOGOS) framework is proposed to develop efficient factorization algorithms for IE methods for electromagnetic analysis. In [11], an adaptive cross approximation (ACA)-based method is used to compress and directly solve IE-based dense matrices arising from scattering problems. It successfully solved electrically large IEs for problem sizes up to 1-M unknowns. In [17], the fast $\mathcal{H}$-matrix arithmetic was introduced to solve large-scale electrodynamic problems. The cost of the fast $\mathcal{H}$-matrix-based computation was further reduced without sacrificing accuracy. The resultant direct solver successfully solved dense matrices that involve more than 1 million unknowns associated with electrodynamic problems of 96 wavelengths in fast CPU time (less than 20 h in LU factorization, 85 s in LU solution), modest memory consumption, and with prescribed accuracy satisfied on a single CPU running at 3 GHz. Other recent developments on direct solvers can be found in [13]. These direct solvers have significantly reduced the computational cost of their traditional counterparts. However, no linear complexity (optimal complexity) has been achieved for the analysis of general 3-D problems.

The focus of this paper is circuit parameter extraction of arbitrarily shaped 3-D lossy conductors embedded in a dielectric material. This class of problems typically has small or moderate electric sizes. For electrically small or moderate problems, no linear-complexity direct matrix solution was achieved prior to the work reported in [14]–[16], where an $\mathcal{H}^2$-matrix-based mathematical framework [18]–[21] was introduced to reduce the computational complexity of direct matrix solutions. Although the $\mathcal{H}^2$-matrix enables a highly efficient computation of dense matrices, under this mathematical framework, no linear complexity has been established for matrix inversion. In [14], it is established for the first time that an $\mathcal{H}^2$-based inverse can be performed in linear complexity. The detailed inverse algorithm and a theoretical analysis on its complexity and accuracy are given in [15]. In [16], it is shown that an $\mathcal{H}^2$-based LU factorization can also be performed in linear complexity. Based on the linear-time direct matrix solution, in [14]–[16], the dense system of linear equations arising from an IE-based analysis was directly solved for the capacitance extraction of arbitrarily shaped 3-D structures embedded in inhomogeneous materials. However, the impedance extraction developed in [14], [16] was only for ideal conductors in a uniform material. To the best of our knowledge, the impedance extraction of 3-D structures that involve nonideal conductors has not been accomplished with a linear-complexity direct solution.

The contribution of this paper is the development of a linear-complexity direct solution for the surface IE-based impedance extraction involving arbitrarily shaped 3-D nonideal conductors embedded in a dielectric material. A surface integral formulation [22], [34] is attractive for impedance extraction compared to a volume integral formulation since the number of unknowns is greatly reduced. The surface integral formulation used in this work is based on a full-wave formulation given in [22] that rigorously handles skin effects, and meanwhile supports an electric potential-based excitation. This formulation is shown to be accurate and robust over a broad band of frequencies. Although a full-wave kernel is considered in this work, it is worth mentioning that the electric size of the problem being considered is not large since if the electric size is large, there is no need to consider fields inside the nonideal conductors as skin depth is negligible and conductors can be accurately treated as perfect conductors. The unknowns solved by the surface integral formulation in [22] are tangential $\mathbf{E}$, tangential $\mathbf{H}$, scalar potential, and charge density on the conducting surfaces. The resultant system matrix is composed of both dense and sparse matrix blocks. Some of these blocks are even not square matrices. Although such a complicated matrix structure does not create an additional challenge to the fast computation of a matrix–vector multiplication, it does render the fast computation of a matrix inverse or LU particularly challenging. The entire system matrix cannot be represented as one $\mathcal{H}^2$ matrix. The same is true for its inverse. As a result, the method developed in [14], [15] is not directly applicable to the impedance extraction concerned in this work. In the following sections, we establish a general $O(N)$ direct matrix solution for rapidly solving a highly irregular system matrix composed of both dense and sparse blocks, to extract impedances of arbitrarily shaped 3-D nonideal conductors embedded in a dielectric material. The proposed $O(N)$ inverse is the matrix solution for N right-hand sides. Moreover, its accuracy can be controlled to any desired order. The basic idea of this paper has been presented in conference paper [23]. Limited by space, many details were omitted in [23]. They are fully discussed in this paper along with an enriched section of numerical results.

The remainder of this paper is organized as follows. In Section II, we present a mathematical background of the $\mathcal{H}^2$-matrix framework. In Section III, we construct a system matrix from a surface IE-based formulation for impedance extraction of arbitrarily shaped 3-D lossy conductors immersed in a dielectric material. In Section IV, we present the proposed linear-complexity direct matrix solution. Moreover, we provide a theoretical analysis on the complexity and accuracy of the proposed direct matrix solution. In Section V, we present numerical results to demonstrate the performance of the proposed direct solver for impedance extraction. The solver demonstrates a clear linear scaling in both CPU time and memory consumption with controlled accuracy. It is capable of inverting a matrix involving 2.68 million unknowns resulting from surface IE-based impedance extraction in 1.3 h and 1.5-GB memory on a single 3-GHz CPU. Comparisons with state-of-the-art impedance solvers such as FastHenry [32] and FastImp [9], [31] have demonstrated a clear advantage of the proposed direct solver. Section VI relates to our conclusions.

II. MATHEMATICAL BACKGROUND

The $\mathcal{H}$ (hierarchical)-matrix is a general mathematical framework [18], [24]–[26], which enables a highly compact representation and efficient numerical computation of dense matrices. Both storage requirements and matrix–vector multiplications using $\mathcal{H}$ matrices are of complexity $O(N\log^{\alpha}N)$. $\mathcal{H}^2$-matrices, which are a specialized subclass of hierarchical matrices, were later introduced in [19]–[21]. Compared to an $\mathcal{H}$-matrix, an $\mathcal{H}^2$ matrix has a reduced computational complexity. The nested structure is the key difference between $\mathcal{H}$-matrices and $\mathcal{H}^2$-matrices. In what follows, we give a basic overview of the $\mathcal{H}^2$-matrix-based mathematical framework.

A. $\mathcal{H}^2$-Matrix Definition

An $\mathcal{H}^2$ matrix is generally associated with a strong admissibility condition [18, p. 145]. Denoting the full index set of all the basis functions used in a numerical discretization by $\mathcal{I}=\{1,2,\ldots,N\}$, where N is the total number of basis functions, consider two subsets t and s of the $\mathcal{I}$, the strong admissibility condition is defined as

$$(t,s)\text{ are admissible} = \begin{cases}\text{true}, & \text{if } \max\{\operatorname{diam}(\Omega_t),\operatorname{diam}(\Omega_s)\}\le\eta\operatorname{dist}(\Omega_t,\Omega_s)\\ \text{false}, & \text{otherwise}\end{cases} \tag{1}$$

where Ω_t and Ω_s are the supports of the union of all the basis functions in t and s, respectively, $\operatorname{diam}(.)$ is the Euclidean diameter of a set, $\operatorname{dist}(.,.)$ is the Euclidean distance between two sets, and η is a positive parameter that can be used to control the admissibility condition. Generally speaking, it is not practical to directly measure the Euclidean diameter and Euclidean distance between Ω_t and Ω_s. An axis-parallel bounding box $Q_t\supseteq\Omega_t$, which is the tensor product of intervals [18, pp 46–48], is thus used to represent the support of the union of all the basis functions in t, and similarly in s. If subsets t and s satisfy (1), they are admissible; otherwise, they are inadmissible.

As can be seen from (1), if subsets t and s are admissible, they are sufficiently far from each other. Given an IE-based dense matrix $\mathbf{Z}$, denoting the matrix block formed by t and s by $\mathbf{Z}^{t,s}$, if all the blocks $\mathbf{Z}^{t,s}$ formed by the admissible (t,s) in matrix $\mathbf{Z}$ can be represented by a factorized form

$$\begin{aligned}&\mathbf{Z}^{t,s}:=\mathbf{V}^t\mathbf{S}^{t,s}\mathbf{V}^{s^{\mathrm{T}}}\quad \mathbf{V}^t\in\mathbb{C}^{\#t\times k}\\ &\mathbf{S}^{t,s}\in\mathbb{C}^{k\times k}\quad \mathbf{V}^s\in\mathbb{C}^{\#s\times k}\end{aligned} \tag{2}$$

where $\mathbf{V}^t(\mathbf{V}^s)$ is nested, then $\mathbf{Z}$ is an $\mathcal{H}^2$ matrix. In (2), $\mathbf{V}^t$ and $\mathbf{V}^s$ are called a cluster basis associated with t and s, respectively, $\mathbf{S}^{t,s}$ is called a coupling matrix, k is the rank of $\mathbf{V}^t(\mathbf{V}^s)$, and "#" denotes the cardinality of a set.

For an efficient computation, the hierarchical dependence of the unknowns is stored in a tree structure. Each node of the tree is called a cluster, which represents a subset of the entire unknown set $\mathcal{I}$. The cluster basis $\mathbf{V}^t$ is nested if across the cluster tree, it satisfies

$$\mathbf{V}^t=\begin{pmatrix}\mathbf{V}^{t1}\mathbf{E}^{t1}\\ \mathbf{V}^{t2}\mathbf{E}^{t2}\end{pmatrix}=\begin{pmatrix}\mathbf{V}^{t1} & \\ & \mathbf{V}^{t2}\end{pmatrix}\begin{pmatrix}\mathbf{E}^{t1}\\ \mathbf{E}^{t2}\end{pmatrix} \tag{3}$$

where $t1,t2\in\operatorname{children}(t)$, which are the two children clusters of t. In (3), $\mathbf{E}^{t1}$ and $\mathbf{E}^{t2}$ are called transfer matrices associated with a nonleaf cluster t and they are used to build a relationship between t and its two children.

B. $\mathcal{H}^2$-Matrix-Based Fast Arithmetics

The nested property enables linear-time arithmetics of $\mathcal{H}^2$ matrices. In mathematical literature, it is shown that for frequency independent kernels, storage requirements, matrix–vector multiplications, and matrix-matrix multiplications using $\mathcal{H}^2$-matrices are all of complexity $O(N)$ [21]. However, no linear-complexity direct matrix solutions have been reported in the mathematical literature. In [14]–[16], it is shown that $\mathcal{H}^2$-based inverse and LU can also be performed in linear complexity for static problems and electrically moderate problems.

The challenge of this paper is the development of a linear-complexity direct matrix solution for a highly irregular matrix system arising from the impedance extraction of arbitrarily shaped 3-D lossy conductors embedded in a dielectric material. The system matrix is mixed with both

dense and sparse matrix blocks. The entire system matrix cannot be represented as one $\mathcal{H}^2$ matrix. The same is true for its inverse. In what follows, we will show step by step how this challenge is overcome.

III. FORMULATION OF SYSTEM MATRIX

A. Surface Integral Formulation for Full-Wave-Based Impedance Extraction

Consider a union of conductors of finite conductivity σ immersed in a dielectric material characterized by permittivity ε and permeability μ. We employ the surface IE-based formulation derived in [22] to extract the impedance matrix of the conductor network in a broad band of frequencies. The formulation comprises the following five equations:

$$-\frac{1}{2}\vec{E}(\vec{r}) = Z_C \cdot \left[\frac{j}{K_C}\iint_S ds'\nabla'\cdot(\hat{n}\times\vec{H})\nabla G_0 + jK_C\iint_S ds'(\hat{n}\times\vec{H})G_0\right] + \iint_S ds'(\hat{n}\times\vec{E})\times\nabla G_0 + \nabla\varphi(\vec{r}) \quad (4)$$

$$\frac{1}{2}\vec{H}(\vec{r}) = -\left[\frac{j}{K_C}\iint_S ds'\nabla'\cdot(\hat{n}\times\vec{E})\nabla G_1 + jK_C\iint_S ds'(\hat{n}\times\vec{E})G_1\right]\Big/ Z_C + \iint_S ds'(\hat{n}\times\vec{H})\times\nabla G_1 \quad (5)$$

$$\varphi(\vec{r}) = \iint_S G_0\rho(\vec{r}')/\varepsilon\, ds' \quad (6)$$

$$\nabla\cdot(\hat{n}\times\vec{H})(\vec{r}_{nc}) = K_C^2\rho(\vec{r}_{nc})/\sigma\mu \quad (7)$$

$$\varphi(\vec{r}_c) = \psi_C \quad (8)$$

where $\vec{E}$ is electric field intensity, $\vec{H}$ is magnetic field intensity, $\hat{n}$ is a unit vector normal to the conductor surface and pointing away from the conductor, G_0 and G_1 are the full-wave Green's function in the background material, and the conducting region, respectively. They are given by

$$G_0(\vec{r},\vec{r}') = \frac{e^{-jk|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|}$$
$$G_1(\vec{r},\vec{r}') = \frac{e^{-jK_C|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|} \quad (9)$$

with

$$k = \omega\sqrt{\mu\varepsilon}, \qquad K_C = \sqrt{\omega^2\mu\varepsilon - j\omega\mu\sigma} \quad (10)$$

where ω is the angular frequency. In (4)–(9), $Z_C = \omega\mu/K_C$, $\vec{r}$ denotes an observation point, $\vec{r}'$ denotes a source point, S is the conducting surface, φ is electric scalar potential, ρ is charge density, $\vec{r}_c$ denotes a point on the contact surfaces where the voltage source ψ_C is supplied, and $\vec{r}_{nc}$ denotes a point on the noncontact surface. The first equation is an electric field integral equation (EFIE) that describes the interaction of equivalent electric and magnetic currents on the conductor surfaces via the background material; the second IE is a magnetic field integral equation (MFIE) formulated for what is inside each conductor. The third equation describes the potential-charge relationship. The fourth equation reveals the relationship between charges on the noncontact surfaces and tangential $\vec{H}$. The last equation describes a voltage-source-based excitation. The introduction of electric potential φ and subsequently the introduction of (6) and (7) facilitate the incorporation of a voltage-source-based excitation into a field-based EFIE-MFIE system.

The unknowns involved in (4)–(8) are

$$(\hat{n}\times\vec{E}, \hat{n}\times\vec{H}, \varphi, \rho)$$

on the conducting surfaces. After $\hat{n}\times\vec{H}$ is solved, we can use it to compute the current and the impedance of the conductor as follows:

$$I = \sigma\iint_{S_C} dsE_n = -\sigma/(\sigma + j\omega\varepsilon)\iint_{S_C} ds\nabla\cdot(\hat{n}\times\vec{H})$$
$$Z = \frac{V}{I} \quad (11)$$

where S_C represents the contact surface. For a conductor network, its admittance matrix can be computed as

$$\mathbf{Y}_{ij} = \frac{I_i}{V_j}, \qquad (V_i = 0,\ i \neq j)$$

where I_i is the current computed at the ith port with the jth port excited by $\varphi = V_j$ and all the other ports grounded. From the admittance matrix, one can obtain other network parameters such as $\mathbf{Z}$-parameters and $\mathbf{S}$(cattering)-parameters as well.

B. Construction of the System Matrix

We discretize the conducting surface into triangular elements to accurately model arbitrarily shaped conductors. In each triangular element, the equivalent magnetic current $\hat{n} \times \vec{E}$ and the equivalent electric current $\hat{n} \times \vec{H}$ are expanded by using RWG vector basis functions $\vec{f}_n$ [29] as follows:

$$\hat{n} \times \vec{H} = \sum_{n=1}^{N} I_n \vec{f}_n \quad \text{and} \quad \hat{n} \times \vec{E} = \sum_{n=1}^{N} II_n \vec{f}_n \tag{12}$$

where I_n and II_n are unknown coefficients.

The charge density ρ and the potential φ are expanded by scalar pulse basis function in each element. We denote the total number of RWG bases by N, and the total number of triangular panels by N_T. The Galerkin method is applied to test (4) and (5). The centroid collocation method is applied to test (6) and (7). The resultant system of linear equations can be written as

$$Z_C \mathbf{L}_0[I_n] - \mathbf{K}_0[II_n] + \boldsymbol{\varphi}_{\text{sp}}[\varphi] = 0 \tag{13}$$

$$\mathbf{K}_1[I_n] + \mathbf{L}_1/Z_C[II_n] = 0 \tag{14}$$

$$[\varphi] = \mathbf{P}_0[\rho] \tag{15}$$

$$\mathbf{J}_{\text{sp}}[I_n] = [\rho]_{NC} \tag{16}$$

$$[\varphi]_C = \psi_C \tag{17}$$

where the subscripts "C" and "NC" denote the quantities on the contact surfaces and noncontact surfaces, respectively. The $\mathbf{L_0}, \mathbf{K_0}, \mathbf{L_1}, \mathbf{K_1}, \mathbf{P_0}$ in (13)–(17) are dense matrices. Their elements are given by

$$\begin{aligned}
(\mathbf{L}_i)_{mn} &= jK_C \iint_{S_m} ds \vec{f}_m(\vec{r}) \cdot \iint_{S_n} ds' \vec{f}_n(\vec{r}') G_i \\
&\quad + \left(\frac{1}{jK_C}\right) \iint_{S_m} ds \nabla \cdot \vec{f}_m(\vec{r}) \iint_{S_n} ds' \nabla' \cdot \vec{f}_n(\vec{r}') G_i \\
(\mathbf{K}_i)_{mn} &= \iint_{S_m} ds \vec{f}_m(\vec{r}) \cdot \iint_{S_n} ds' \nabla G_i \times \vec{f}_n(r') \\
&\quad - 0.5 \iint_{S_m} ds \left(\hat{n} \times \vec{f}_m(\vec{r})\right) \cdot \vec{f}_n(\vec{r}) \\
\mathbf{P}_0 &= \left(\frac{1}{\varepsilon}\right) \iint_{S} G_0(\vec{r}, \vec{r}')\, ds'
\end{aligned} \tag{18}$$

with $i = 0, 1$. The $\boldsymbol{\varphi}_{\text{sp}}$ in (13) is a sparse matrix of dimension $N \times N_T$. For each $(\boldsymbol{\varphi}_{\text{sp}})_{mn}$, m is the index of the RWG basis function, whose degree of freedom is assigned to each edge, and n is the index of the pulse basis function, whose degree of freedom is assigned to each triangular panel. The $\mathbf{J}_{\text{sp}}$ has a dimension of $NC \times N$, which is also sparse. The NC is the number of triangular panels on the noncontact surfaces. The elements of $\boldsymbol{\varphi}_{\text{sp}}$ and $\mathbf{J}_{\text{sp}}$ are given by

$$\begin{aligned}
(\boldsymbol{\varphi}_{\text{sp}})_{mn} &= -\iint_{S_m} ds \nabla \cdot \vec{f}_m(\vec{r}_n) \\
(\mathbf{J}_{\text{sp}})_{mn} &= \sigma\mu / K_C^2 \nabla \cdot \vec{f}_n(\vec{r}_m).
\end{aligned} \tag{19}$$

The total number of unknowns in the system (13)–(17) is $2N + 2N_T$, where $2N$ unknowns are associated with vector-based RWG basis functions, and $2N_T$ unknowns are associated with scalar-based pulse basis functions.

C. Derivation of a Reduced System Matrix that Involves Equivalent Electric Currents Only

The matrix system shown in (13)–(17) consists of both sparse and dense blocks, which cannot be represented as an $\mathcal{H}^2$ matrix as a whole. To develop a linear-complexity direct solution of (13)–(17), our strategy is to eliminate all the $[II_n]$, $[\rho]$, and $[\varphi]$ unknowns from (13)–(17) to reduce the system to a small one that only involves $[I_n]$.

Using (14) to eliminate $[II_n]$, (13) becomes

$$\begin{aligned}
Z_C\left(\mathbf{L}_0 + \mathbf{K}_0 \cdot \mathbf{L}_1^{-1} \cdot \mathbf{K}_1\right)[I_n] + \boldsymbol{\varphi}_{\text{sp}}[\varphi] &= 0 \\
\Rightarrow \mathbf{H_{LK}}[I_n] + \boldsymbol{\varphi}_{\text{sp}}[\varphi] &= 0
\end{aligned} \tag{20}$$

with $\mathbf{H_{LK}} = Z_C(\mathbf{L}_0 + \mathbf{K}_0 \cdot \mathbf{L}_1^{-1} \cdot \mathbf{K}_1)$.

Based on (15), the charge $[\rho]$ on the noncontact surface can be represented by $[\varphi]$ as

$$\begin{aligned}
[\rho]_{NC} &= \left(\mathbf{P}_0^{-1}\right)_{NC\times C} \cdot \psi_C + \left(\mathbf{P}_0^{-1}\right)_{NC\times NC} [\varphi]_{NC} \\
&= \left(\mathbf{P}_0^{-1}\right)_{NC\times C} \cdot \psi_C + \left[\left(\widehat{\mathbf{P}}_0\right)_{NC\times NC}\right]^{-1} [\varphi]_{NC}
\end{aligned} \tag{21}$$

with $\left(\widehat{\mathbf{P}}_0\right)_{NC\times NC} = (\mathbf{P}_0)_{NC\times NC} - (\mathbf{P}_0)_{NC\times C} \times (\mathbf{P}_0)_{C\times C}^{-1} \times (\mathbf{P}_0)_{C\times NC}$, which is the Schur complement of the matrix block $(\mathbf{P}_0)_{NC\times NC}$.

Using (16) and (21), we can directly relate $[\varphi]$ to $[I_n]$ by

$$[\varphi]_{NC} = \left(\widehat{\mathbf{P}}_0\right)_{NC\times NC} \cdot \left(\mathbf{J}_{\text{sp}}[I_n] - \left(\mathbf{P}_0^{-1}\right)_{NC\times C} \cdot \psi_C\right). \tag{22}$$

By separating the contact-surface potential from the noncontact-surface potential, (20) can be rewritten as

$$\mathbf{H_{LK}}[I_n] + (\boldsymbol{\varphi}_{\text{sp}})_{N\times C} \cdot \psi_C + (\boldsymbol{\varphi}_{\text{sp}})_{N\times NC} \cdot [\varphi]_{NC} = 0. \tag{23}$$

Substituting (22) into (23), we obtain

$$\mathbf{Z}[I_n] = \mathbf{rhs} \tag{24}$$

where

$$\begin{aligned}\mathbf{Z} &= \mathbf{H_{LK}} + \mathbf{H_{SP}} \\ \mathbf{rhs} &= \left((\boldsymbol{\varphi}_{\mathrm{sp}})_{N\times NC}\cdot(\widehat{\mathbf{P}}_0)_{NC\times NC}\cdot(\mathbf{P_0^{-1}})_{NC\times C}\right. \\ &\quad \left.-(\boldsymbol{\varphi}_{\mathrm{sp}})_{N\times C}\right)\cdot\psi_C\end{aligned}$$

in which

$$\begin{aligned}\mathbf{H_{LK}} &= Z_C\left(\mathbf{L}_0 + \mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1\right) \\ \mathbf{H_{SP}} &= (\boldsymbol{\varphi}_{\mathrm{sp}})_{N\times NC}\cdot(\widehat{\mathbf{P}}_0)_{NC\times NC}\cdot\mathbf{J}_{\mathrm{sp}}.\end{aligned} \tag{25}$$

Electro-Magneto-Quasi-Static (EMQS) and Magneto-Quasi-Static (MQS) Analysis: The system matrix shown in (13)–(17) can be further reduced for MQS analysis and EMQS analysis in a relatively low frequency regime. For EMQS analysis, since displacement current is negligible, the wave number k is 0 in the Green's function of the background material given in (9), which further simplifies the system matrix (24) by removing the $e^{jk|\vec{r}-\vec{r}'|}$ term. For MQS analysis, in addition to $k = 0$, the surface charge density ρ can be ignored, which reduces (16) to

$$\mathbf{J}_{\mathrm{sp}}[I_n] = 0. \tag{26}$$

Based on (20), we obtain

$$[I_n] = -\mathbf{H_{LK}^{-1}}\boldsymbol{\varphi}_{\mathrm{sp}}[\varphi]. \tag{27}$$

From (26) and (27), we can obtain the following system for MQS analysis

$$\mathbf{J}_{\mathrm{sp}}\mathbf{H_{LK}^{-1}}\boldsymbol{\varphi}_{\mathrm{sp}}[\varphi] = 0. \tag{28}$$

By separating the potential on contact surfaces from that on noncontact surfaces, we obtain

$$\left(\mathbf{J}_{\mathrm{sp}}\mathbf{H_{LK}^{-1}}(\boldsymbol{\varphi}_{\mathrm{sp}})_{N\times NC}\right)[\varphi]_{NC} = -\mathbf{J}_{\mathrm{sp}}\mathbf{H_{LK}^{-1}}(\boldsymbol{\varphi}_{\mathrm{sp}})_{N\times C}\cdot\psi_C \tag{29}$$

from which $[\varphi]$ on noncontact surfaces can be computed. After $[\varphi]$ is computed, $[I_n]$ can be solved from (27).

In (24), $\mathbf{Z}$ is obtained by eliminating $[II_n]$, $[\rho]$, and $[\varphi]$ unknowns. In general, the unknown elimination has a cubic complexity. In other words, the matrix operations shown in (25) have a cubic complexity. In this work, we ensure that the elimination of the $[II_n]$, $[\rho]$, and $[\varphi]$ unknowns, i.e. the construction of $\mathbf{Z}$, is performed in linear complexity, and also the reduced $[I_n]$-system is solved in linear complexity, holding the complexity of the entire solution to linear. The detailed procedure of the proposed linear-complexity direct solution is given in the following section.

IV. LINEAR-COMPLEXITY DIRECT SOLUTION

In this section, we present a complete procedure of linear-complexity direct solution to the system (13)–(17) for full-wave-based impedance extraction. The entire flow is shown in Fig. 1. First, we construct an efficient $\mathcal{H}^2$ partition that fully takes advantage of the surface IE-based formulation to generate clusters that are 2-D for efficient computation of 3-D problems. Based on the proposed efficient $\mathcal{H}^2$ partition, we build $\mathcal{H}^2$-representations of the dense matrices $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, $\mathbf{K_1}$, and $\mathbf{P_0}$ with controlled accuracy, and in linear complexity. We also build $\mathcal{H}^2$-representations of the sparse matrices $\boldsymbol{\varphi}_{\mathrm{sp}}$ and $\mathbf{J}_{\mathrm{sp}}$, which does not involve any approximation. Since the matrices involved in (13)–(17) do not share the same cluster basis in common, to facilitate an efficient computation, we unify the cluster bases. We then use the new unified cluster bases to update the $\mathcal{H}^2$ representations of the dense matrices $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, $\mathbf{K_1}$, and $\mathbf{P_0}$ as well as sparse matrices $\boldsymbol{\varphi}_{\mathrm{sp}}$ and $\mathbf{J}_{\mathrm{sp}}$. After that, we perform a linear-complexity

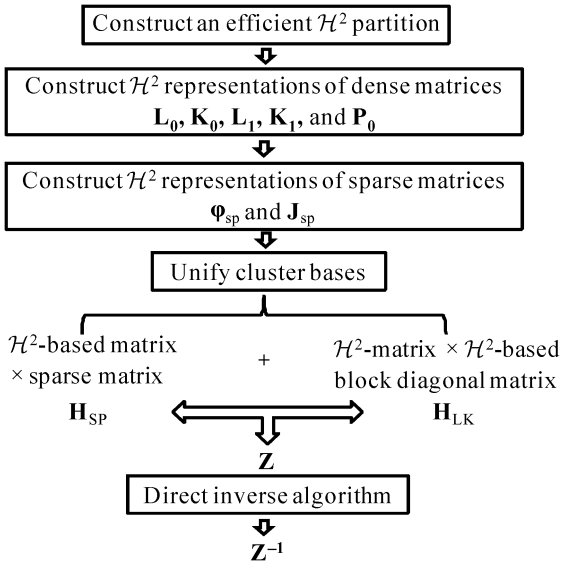


Fig. 1. *Overall procedure of the proposed direct solution.*

multiplication of the $\mathcal{H}^2$-based sparse matrix $\mathbf{J}_{\mathrm{sp}}(\varphi_{\mathrm{sp}})$ with an $\mathcal{H}^2$-based dense matrix, from which we obtain an $\mathcal{H}^2$-based $\mathbf{H}_{\mathrm{sp}}$. Moreover, we perform a linear-complexity multiplication of an $\mathcal{H}^2$-based dense matrix with an $\mathcal{H}^2$-based block diagonal matrix to obtain an $\mathcal{H}^2$-based $\mathbf{H}_{\mathrm{LK}}$. With $\mathbf{H}_{\mathrm{sp}}$ and $\mathbf{H}_{\mathrm{LK}}$ obtained, $\mathbf{Z}$ is obtained. Because each of the aforementioned steps has a linear complexity, the construction of $\mathbf{Z}$ has a linear complexity. Similarly, the **rhs** in (25) is computed in linear time by sparse matrix–vector multiplications and $\mathcal{H}^2$-based matrix–vector multiplications. After $\mathbf{Z}$ and **rhs** are obtained, the linear-complexity inverse algorithm developed in [15] can be employed to directly solve (24) in linear time. In the following subsections, we detail the algorithm in each step.

A. New $\mathcal{H}^2$ Partition

An $\mathcal{H}^2$-based partition is to separate a matrix into admissible blocks that have a factorized form shown in (2) and inadmissible blocks that have a full-matrix representation. This is generally done by a cardinality-based splitting method [18]. Although this method is general, it is not efficient in the context of the surface integral-based impedance extraction. In this work, we propose a new partition method that significantly improves the efficiency of the resultant $\mathcal{H}^2$-based computation for wideband impedance extraction.

We first show the proposed scheme for constructing a cluster tree, from which we build an $\mathcal{H}^2$-partition. To help understanding, we use a simple example to explain the construction procedure without loss of generality. Consider an impedance system made of eight arbitrarily shaped conductors. Assuming (t_1, t_2, t_3) to be a coordinate system, the arbitrarily shaped conductors can be orientated in any direction in the 3-D space. We first find out the direction along which the structure being simulated has the maximal size, we then split the entire system into two subsystems along this direction. We repeat the process, during which each conductor is treated as the smallest splitting-unit, and we do not split any single conductor. We continue to split in this way until one conductor is left in each subsystem. After that, we switch to another strategy to split a single conductor. We separate the single conductor into three groups that, respectively, contain the panels on t_1t_2, t_2t_3, t_1t_3 surfaces, and place each group as a top cluster of the conductor. We then use the conventional splitting method to construct the descendant clusters of the three top clusters. We keep such a splitting until the number of bases involved in each cluster is less than or equal to *leafsize*, which is a parameter to control the tree depth. The aforementioned scheme generates a cluster tree shown in Fig. 2. The major advantage of such an approach to constructing a cluster tree is that each cluster is made two dimensional, and hence fully taking advantage of the surface-based formulation to speed up the $\mathcal{H}^2$-based computation of 3-D problems.

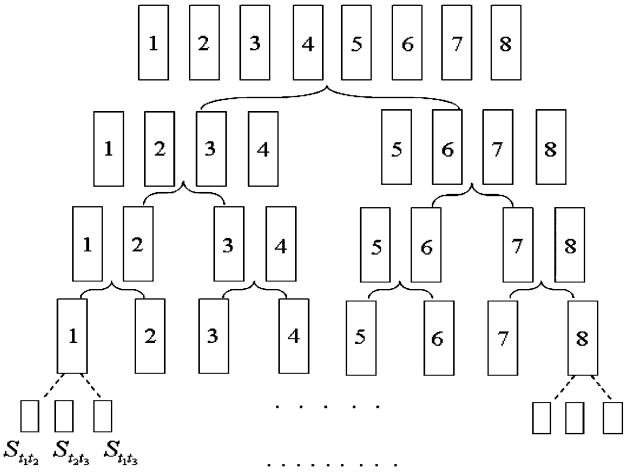


Fig. 2. *A cluster tree.*

We denote the cluster tree as T. The first node of the tree is called root cluster, denoted by $\mathrm{Root}(T)$. Clusters with indices no more than *leafsize* are leaves. Each nonleaf cluster has two children. Based on the cluster tree T and the admissibility condition (1), we construct an $\mathcal{H}^2$ partition. We start from $\mathrm{Root}(T)$ and $\mathrm{Root}(T)$, and test the admissibility condition between clusters $t \in T$ and $s \in T$ level by level. Once two clusters t and s are found to be admissible, the block formed by them is called an admissible block, and we do not check the admissibility condition for the combination of their children. If clusters t and s are both leaf clusters but not admissible, we call the block formed by them inadmissible block. The aforementioned process results in an $\mathcal{H}^2$ partition composed of both admissible blocks and inadmissible blocks.

The above procedure can be used to generate $\mathcal{H}^2$ partitions for the impedance extraction system described in Section III. As shown in Section III-B, two different basis functions are used for discretization: one is RWG basis and the other is triangular-panel-based pulse basis. Therefore, we construct a cluster tree-R and a cluster tree-T for RWG bases and pulse bases, respectively. The cluster tree-R is used to construct an $\mathcal{H}^2$ partition of $\mathbf{L}_0, \mathbf{K}_0, \mathbf{L}_1, \mathbf{K}_1$, and the cluster tree-T is used to construct the $\mathcal{H}^2$ partition of $\mathbf{P}_0$. Fig. 3(a) shows one example of such an $\mathcal{H}^2$ partition.

Although $\mathbf{L}_0$, $\mathbf{K}_0$, $\mathbf{L}_1$, and $\mathbf{K}_1$ share the same $\mathcal{H}^2$ partition, the $\mathcal{H}^2$ structure of $\mathbf{L}_1$ and $\mathbf{K}_1$ is different from that of $\mathbf{L}_0$ and $\mathbf{K}_0$. This difference lies in the fact that (5) is only satisfied in each conductor, and hence the matrix block formed for one conductor does not couple with that formed for another conductor, which is physically disconnected. Thus, only the diagonal blocks that are formed by the same conductor cluster are nonzero and all the other blocks are zero as shown in Fig. 3(b). In addition, if two conductors c_1 and c_2 have the same shape and size,

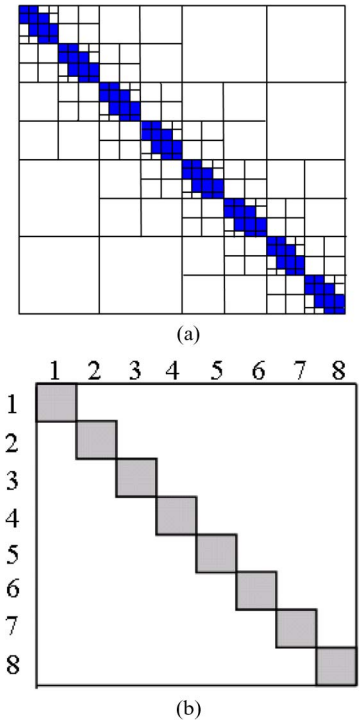


Fig. 3. *An $\mathcal{H}^2$-matrix partition: (a) Partition for $\mathbf{L_0}, \mathbf{K_0}$ (■ full matrix block, □ admissible block). (b) Diagonal partition for $\mathbf{L_1}, \mathbf{K_1}$: gray blocks represent nonzero blocks, white blocks represent zero blocks.*

their corresponding diagonal blocks are the same, which simplifies the computational cost associated with these blocks. For example, when evaluating the inverse of $(\mathbf{L}_1)_{c1,c1}$ and $(\mathbf{L}_1)_{c2,c2}$, once one of them is computed, the other can be directly obtained. By doing so, the computational efficiency can be further improved.

B. $\mathcal{H}^2$ Representations of Dense Matrices $\boldsymbol{L}_0, \boldsymbol{K}_0, \boldsymbol{L}_1, \boldsymbol{K}_1, \boldsymbol{P}_0$

To construct $\mathcal{H}^2$ representations of the dense matrices $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, $\mathbf{K_1}$, and $\mathbf{P_0}$ involved in (13)–(17), for an admissible block (t, s), we replace the G_1 and G_0 in (18) by a degenerate approximation based on Lagrange interpolation method

$$\tilde{G}_{0,1}^{t,s}(\vec{r},\vec{r}') = \sum_{\nu\in K^t}\sum_{\mu\in K^s} G\left(\xi_\nu^t,\xi_\mu^s\right) L_\nu^t(\vec{r}) L_\mu^s(\vec{r}') \tag{30}$$

where $(\xi_\nu^t)_{\nu\in K^t}$ and $(\xi_\mu^s)_{\mu\in K^s}$ are Lagrange interpolation points chosen, respectively, in t and s; and $(L_\nu^t)_{\nu\in K^t}$ and $(L_\mu^s)_{\mu\in K^s}$ are the corresponding Lagrange polynomials. The number of interpolation points in each cluster is p^d, where p is the number of interpolation points along each dimension, and d is problem dimension, which is reduced from 3 to 2 by the proposed new $\mathcal{H}^2$ partition although the problem to be solved is 3-D in nature. The accuracy of (30) can be controlled to any order as proved in [28].

By substituting (30) into (18), the double integrals in (18) can be separated into two single integrals

$$\begin{aligned}
\left(\tilde{\mathbf{L}}_{0,1}^{t,s}\right)_{ij} &= \sum_{\nu\in K^t}\sum_{\mu\in K^s}\Bigg(\iint_{S_i} \vec{J}_i(\vec{r}) L_\nu^t(\vec{r})\, ds \\
&\quad \cdot jK_C G_{0,1}\left(\xi_\nu^t,\xi_\mu^s\right)\cdot\iint_{S_j}\vec{J}_j(\vec{r}\,') L_\mu^s(\vec{r}\,')\, ds' \\
&\quad + \iint_{S_i}\nabla\cdot\vec{J}_i(\vec{r}) L_\nu^t(\vec{r})\, ds\cdot\frac{1}{jK_C} G_{0,1}\left(\xi_\nu^t,\xi_\mu^s\right) \\
&\quad \cdot\iint_{S_j}\nabla'\cdot\vec{J}_j(\vec{r}\,') L_\mu^s(\vec{r}\,')\, ds'\Bigg) \\
\left(\tilde{\mathbf{K}}_{0,1}^{t,s}\right)_{ij} &= \sum_{\nu\in K^t}\sum_{\mu\in K^s}\iint_{S_i}\vec{J}_i(\vec{r}) L_\nu^t(\vec{r})\, ds\cdot\nabla G_{0,1}\left(\xi_\nu^t,\xi_\mu^s\right) \\
&\quad \times\iint_{S_j}\vec{J}_j(\vec{r}\,') L_\mu^s(\vec{r}\,')\, ds' \\
\left(\tilde{\mathbf{P}}_0^{t,s}\right)_{ij} &= \sum_{\nu\in K^t}\sum_{\mu\in K^s} L_\nu^t(\vec{r}_i)\cdot G_0\left(\xi_\nu^t,\xi_\mu^s\right)\Big/\varepsilon\cdot\iint_{S_j} L_\mu^s(\vec{r}\,')\, ds'
\end{aligned}$$

from which we obtain $\mathcal{H}^2$-representations of $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, $\mathbf{K_1}$, and $\mathbf{P_0}$ as

$$\begin{aligned}
\tilde{\mathbf{L}}_{0,1}^{t,s} &= \mathbf{V}_{\mathbf{L}}^t \mathbf{S}_{\mathbf{L}_{0,1}}^{t,s} \mathbf{V}_{\mathbf{L}}^{s^{\mathrm{T}}} \\
\tilde{\mathbf{K}}_{0,1}^{t,s} &= \mathbf{V}_{\mathbf{K}}^t \mathbf{S}_{\mathbf{K}_{0,1}}^{t,s} \times \mathbf{V}_{\mathbf{K}}^{s^{\mathrm{T}}} \\
\tilde{\mathbf{P}}_0^{t,s} &= \left(\mathbf{V}_{p1}^T\right)^t \mathbf{S}_{\mathbf{P}_0}^{t,s}\left(\mathbf{V}_{p2}^T\right)^{s\mathrm{T}}
\end{aligned} \tag{31}$$

where the cluster bases in a given cluster t are

$$\begin{aligned}
\left(\mathbf{V}_{\mathbf{K}}^t\right)_{iv} &= \iint_{S_i}\vec{J}_i(\vec{r}) L_\nu^t(\vec{r})\, ds \\
\left(\mathbf{V}_S^t\right)_{iv} &= \iint_{S_i}\left(\nabla\cdot\vec{J}_i(\vec{r})\right) L_\nu^t(\vec{r})\, ds \\
\mathbf{V}_{\mathbf{L}}^t &= \left[\mathbf{V}_{\mathbf{K}}^t \quad \mathbf{V}_S^t\right] \\
\left(\mathbf{V}_{p1}^T\right)_{iv}^t &= L_\nu^t(\vec{r}_i), \quad \left(\mathbf{V}_{p2}^T\right)_{iv}^t = \iint_{S_i} L_\nu^t(\vec{r})\, ds.
\end{aligned} \tag{32}$$

The coupling matrices shown in (31) are shown in

$$\begin{cases} \mathbf{S}^{t,s}_{\mathbf{L}_{0,1}} = \begin{bmatrix} jK_C\mathbf{S}_{G_{0,1}} & \\ & \left(\frac{1}{jK_C}\right)\mathbf{S}_{G_{0,1}} \end{bmatrix}, \\ \left(\mathbf{S}_{G_{0,1}}\right)_{\nu\mu} = G_{0,1}\left(\xi^t_\nu, \xi^s_\mu\right) \\ \left(\mathbf{S}^{t,s}_{\mathbf{K}_{0,1}}\right)_{\nu\mu} = \nabla G_{0,1}\left(\xi^t_\nu, \xi^s_\mu\right), \left(\mathbf{S}^{t,s}_{\mathbf{P}0}\right)_{\nu\mu} = G_0\left(\xi^t_\nu, \xi^s_\mu\right)\Big/\varepsilon \end{cases} \tag{33}$$

with $i \in t, j \in s, \nu \in K^t, \mu \in K^s$, and $\mathbf{V}^t \in \mathbb{R}^{\#t\times\#K^t}$, $\mathbf{S}^{t,s} \in \mathbb{C}^{\#K^t\times\#K^s}$. In (31)–(33), $\#K^{t(s)} \ll \#t$ $(\#s)$ is the rank in the cluster t or s, which is determined by the number of interpolation points. $\mathbf{V}_{\mathbf{L},\mathbf{K},S}$ denotes cluster basis formed by the cluster tree-R, and $(\mathbf{V}^{\mathbf{T}})_{p1,p2}$ denotes cluster basis formed by the cluster tree-T. Since the same space of Lagrange polynomials is used, cluster bases $\mathbf{V}_{\mathbf{L},\mathbf{K},S}$ and $(\mathbf{V}^{\mathbf{T}})_{p1,p2}$ are all nested cluster bases. From (30) to (33), with the $\mathcal{H}^2$-representation found for the dense matrices $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, $\mathbf{K_1}$, and $\mathbf{P_0}$, we prove the existence of the $\mathcal{H}^2$-representation of the dense matrix blocks involved in impedance extraction.

As can be seen from (31) and (32), $\mathbf{L}_0$ and $\mathbf{L}_1$ share the same cluster basis $\mathbf{V_L}$, and they are only different in coupling matrix $\mathbf{S}$; similarly, $\mathbf{K}_0$ and $\mathbf{K}_1$ share the same cluster basis $\mathbf{V_K}$ but with different coupling matrix $\mathbf{S}$. However, $\mathbf{V_L}$ is different from $\mathbf{V_K}$. The $\mathbf{P_0}$ even has different row and column cluster bases. Therefore, for efficient computation of (24), in Section IV-D, we propose an algorithm to unify different cluster bases to build a common set of cluster basis with nested property preserved.

C. $\mathcal{H}^2$ Representations of Sparse Matrices φ_{sp} and J_{sp}

As can be seen from (25), sparse matrices $\boldsymbol{\varphi}_{\rm sp}$ and $\mathbf{J}_{\rm sp}$ are involved in the matrix-matrix multiplication for computing $\mathbf{H}_{\rm sp}$. In order to make use of the linear-time matrix-matrix multiplication algorithm of $\mathcal{H}^2$ matrices, we represent both sparse matrices $\boldsymbol{\varphi}_{\rm sp}$ and $\mathbf{J}_{\rm sp}$ as $\mathcal{H}^2$ matrices. To do so, we first construct an $\mathcal{H}^2$ partition for $\boldsymbol{\varphi}_{\rm sp}$. As can be seen from (19), the row cluster of $\boldsymbol{\varphi}_{\rm sp}$ is formed by RWG bases, and the column cluster of $\boldsymbol{\varphi}_{\rm sp}$ is formed by triangular panel-based pulse bases. Hence, we choose the cluster tree-R as its row cluster tree and cluster tree-T as its column cluster tree, and build an $\mathcal{H}^2$ partition based on the two cluster trees and the admissibility condition (1). We then fill in the matrix elements in each $\mathcal{H}^2$-block. From (19), it can be seen that for an edge i associated with an RWG basis, only $(\boldsymbol{\varphi}_{\rm sp})_{i,t1}$ and $(\boldsymbol{\varphi}_{\rm sp})_{i,t2}$ are nonzero, and all the other $(\boldsymbol{\varphi}_{\rm sp})$ elements are zero, where t_1 and t_2 denote the two triangular panels that share edge i in common. In other words, only adjacent edges and triangular panels can have interactions in $\boldsymbol{\varphi}_{\rm sp}$. Therefore, all the admissible blocks become zero since they are sufficiently far from each other. Only inadmissible blocks are nonzero. Furthermore, in each nonzero block, only several elements are nonzero. Thus, we only need to record all the nonzero elements and their locations in the $\mathcal{H}^2$ representation. In sparse matrix $\mathbf{J}_{\rm sp}$, only $(\mathbf{J}_{\rm sp})_{t1,i}$, $(\mathbf{J}_{\rm sp})_{t1,j}$, and $(\mathbf{J}_{\rm sp})_{t1,k}$ for the triangle element t_1 are nonzero, while all the other elements are zero, where i, j, k are the three edges in triangular element t_1. Therefore, only adjacent edges and triangle panels have interactions in $\mathbf{J}_{\rm sp}$. As a result, similar to $\boldsymbol{\varphi}_{\rm sp}$, the $\mathcal{H}^2$ representation of $\mathbf{J}_{\rm sp}$ can be formed. Since all the blocks that satisfy the admissibility condition are not only low rank, but also zero in $\mathbf{J}_{\rm sp}$ and $\boldsymbol{\varphi}_{\rm sp}$, the $\mathcal{H}^2$ representations of $\mathbf{J}_{\rm sp}$ and $\boldsymbol{\varphi}_{\rm sp}$ are exact.

D. Unifying Cluster Bases

We propose to use orthogonalization to unify the cluster bases. The goal is to unify the cluster bases of $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, and $\mathbf{K_1}$ to be the same set of cluster basis $\mathbf{V}$ with $\mathbf{V}^{\rm T}\mathbf{V} = \mathbf{I}$ satisfied for each cluster. Take $\mathbf{V_L}$ shown in (32) as an example, it is composed of a vector-based $\mathbf{V_K}$ and a scalar-based $\mathbf{V}_S$. Given a cluster t, we first expand the vector-based $\mathbf{V}^t_{\mathbf{L}}$ into a scalar-based form as the following:

$$\mathbf{V}^t_{\mathbf{L}\rm expand} = \begin{bmatrix} \mathbf{V}^t_{\mathbf{K}x} & \mathbf{V}^t_{\mathbf{K}y} & \mathbf{V}^t_{\mathbf{K}z} & \mathbf{V}^t_S \end{bmatrix}.$$

We then orthogonalize $\mathbf{V}^t_{\mathbf{L}\rm expand}$, which results in an orthogonal cluster $\tilde{\mathbf{V}}^t_{\mathbf{L}}$ that contains all the independent column vectors of the original $\mathbf{V}^t_{\mathbf{L}}$. We use the orthogonalization algorithm in [33] to perform this task, which has a linear complexity. With $\tilde{\mathbf{V}}^t_{\mathbf{L}}$ obtained, the cluster bases of $\mathcal{H}^2$-based $\tilde{\mathbf{L}}_{\mathbf{0}}$, $\tilde{\mathbf{K}}_{\mathbf{0}}$, $\tilde{\mathbf{L}}_{\mathbf{1}}$, and $\tilde{\mathbf{K}}_{\mathbf{1}}$ can be accurately unified by $\tilde{\mathbf{V}}^t_{\mathbf{L}}$. For example, based on $\tilde{\mathbf{V}}^t_{\mathbf{L}}$, $\tilde{\mathbf{L}}_{\mathbf{0}}$ can be updated based on the following equation:

$$\begin{aligned} \tilde{\mathbf{L}}^{t,s}_0 &= \left(\tilde{\mathbf{V}}^t_{\mathbf{L}}\tilde{\mathbf{V}}^{t^{\rm T}}_{\mathbf{L}} \cdot \mathbf{V}^t_{\mathbf{L}}\right) \cdot \mathbf{S}^{t,s}_{\mathbf{L}_0} \cdot \left(\tilde{\mathbf{V}}^s_{\mathbf{L}}\tilde{\mathbf{V}}^{s^{\rm T}}_{\mathbf{L}} \cdot \mathbf{V}^s_{\mathbf{L}}\right)^{\rm T} \\ &= \tilde{\mathbf{V}}^t_{\mathbf{L}}\left(\tilde{\mathbf{V}}^{t^{\rm T}}_{\mathbf{L}}\mathbf{V}^t_{\mathbf{L}}\mathbf{S}^{t,s}_{\mathbf{L}_0}\mathbf{V}^{s^{\rm T}}_{\mathbf{L}}\tilde{\mathbf{V}}^s_{\mathbf{L}}\right)\tilde{\mathbf{V}}^{s^{\rm T}}_{\mathbf{L}} \\ &= \tilde{\mathbf{V}}^t_{\mathbf{L}}\Bigg(\tilde{\mathbf{V}}^{t^{\rm T}}_{\mathbf{L}}\mathbf{V}^t_{\mathbf{K}x}\left(jK_C\mathbf{S}^{t,s}_{G_0}\right)\left(\tilde{\mathbf{V}}^{s^{\rm T}}_{\mathbf{L}}\mathbf{V}^s_{\mathbf{K}x}\right)^{\rm T} \\ &\quad + \tilde{\mathbf{V}}^{t^{\rm T}}_{\mathbf{L}}\mathbf{V}^t_{\mathbf{K}y}\left(jK_C\mathbf{S}^{t,s}_{G_0}\right)\left(\tilde{\mathbf{V}}^{s^{\rm T}}_{\mathbf{L}}\mathbf{V}^s_{\mathbf{K}y}\right)^{\rm T} \\ &\quad + \tilde{\mathbf{V}}^{t^{\rm T}}_{\mathbf{L}}\mathbf{V}^t_{\mathbf{K}z}\left(jK_C\mathbf{S}^{t,s}_{G_0}\right)\left(\tilde{\mathbf{V}}^{s^{\rm T}}_{\mathbf{L}}\mathbf{V}^s_{\mathbf{K}z}\right)^{\rm T} \\ &\quad + \tilde{\mathbf{V}}^{t^{\rm T}}_{\mathbf{L}}\mathbf{V}^t_S\left(\frac{\mathbf{S}_{G_0}}{jK_C}\right)\left(\tilde{\mathbf{V}}^{s^{\rm T}}_{\mathbf{L}}\mathbf{V}^s_S\right)^{\rm T}\Bigg)\tilde{\mathbf{V}}^{s^{\rm T}}_{\mathbf{L}} \\ &= \tilde{\mathbf{V}}^t_{\mathbf{L}}\Big(jK_C\Big(\mathbf{W}^t_x\mathbf{S}^{t,s}_{G_0}\mathbf{W}^{s^{\rm T}}_x + \mathbf{W}^t_y\mathbf{S}^{t,s}_{G_0}\mathbf{W}^{s^{\rm T}}_y \\ &\qquad + \mathbf{W}^t_z\mathbf{S}^{t,s}_{G_0}\mathbf{W}^{s^{\rm T}}_z\Big) \\ &\qquad + \mathbf{W}^t_S\mathbf{S}^{t,s}_{G_0}\mathbf{W}^{s^{\rm T}}_S/(jK_C)\Big)\tilde{\mathbf{V}}^{s^{\rm T}}_{\mathbf{L}} \end{aligned}$$

with

$$\mathbf{W}^t_{x,y,z} = \tilde{\mathbf{V}}^{t^{\mathrm{T}}}_{\mathbf{L}} \mathbf{V}^t_{\mathbf{K}x,y,z}. \tag{34}$$

$\tilde{\mathbf{L}}_\mathbf{1}$ can be updated in a similar way.

Next, we show how to update $\tilde{\mathbf{K}}_\mathbf{0}$. The $\mathbf{S}^{t,s}_{\mathbf{K}}$ is a vector-based coupling matrix, as can be seen from (33). It can be expanded as

$$\mathbf{S}^{t,s}_{\mathbf{K}\text{expand}} = \begin{bmatrix} \mathbf{S}^{t,s}_{\mathbf{K}x} & \mathbf{S}^{t,s}_{\mathbf{K}y} & \mathbf{S}^{t,s}_{\mathbf{K}z} \end{bmatrix}.$$

Since $\tilde{\mathbf{V}}^t_{\mathbf{L}}$ keeps all the independent vectors of $\mathbf{V}^t_{\mathbf{L}}$, and $\mathbf{V}^t_{\mathbf{K}}$ is part of $\mathbf{V}^t_{\mathbf{L}}$ as shown in (32), $\mathbf{V}^t_{\mathbf{K}}$ is contained in $\tilde{\mathbf{V}}^t_{\mathbf{L}}$. Hence, $\tilde{\mathbf{K}}_\mathbf{0}$ can be updated accurately based on $\tilde{\mathbf{V}}^t_{\mathbf{L}}$ as

$$\begin{aligned} \tilde{\mathbf{K}}^{t,s}_0 &= \tilde{\mathbf{V}}^t_{\mathbf{L}} \tilde{\mathbf{V}}^{t^{\mathrm{T}}}_{\mathbf{L}} \cdot \mathbf{V}^t_{\mathbf{K}} \mathbf{S}^{t,s}_{\mathbf{K}_0} \times \left(\tilde{\mathbf{V}}^s_{\mathbf{L}} \tilde{\mathbf{V}}^{s^{\mathrm{T}}}_{\mathbf{L}} \cdot \mathbf{V}^s_{\mathbf{K}} \right)^{\mathrm{T}} \\ &= \tilde{\mathbf{V}}^t_{\mathbf{L}} \Big[\left(\mathbf{W}^t_x \mathbf{S}^{t,s}_{\mathbf{K}y} \mathbf{W}^{s^{\mathrm{T}}}_z - \mathbf{W}^t_z \mathbf{S}^{t,s}_{\mathbf{K}y} \mathbf{W}^{s^{\mathrm{T}}}_x \right) \\ &\quad + \left(\mathbf{W}^t_z \mathbf{S}^{t,s}_{\mathbf{K}x} \mathbf{W}^{s^{\mathrm{T}}}_y - \mathbf{W}^t_y \mathbf{S}^{t,s}_{\mathbf{K}x} \mathbf{W}^{s^{\mathrm{T}}}_z \right) \\ &\quad + \left(\mathbf{W}^t_y \mathbf{S}^{t,s}_{\mathbf{K}z} \mathbf{W}^{s^{\mathrm{T}}}_x - \mathbf{W}^t_x \mathbf{S}^{t,s}_{\mathbf{K}z} \mathbf{W}^{s^{\mathrm{T}}}_y \right) \Big] \tilde{\mathbf{V}}^{s^{\mathrm{T}}}_{\mathbf{L}}. \end{aligned} \tag{35}$$

$\tilde{\mathbf{K}}_\mathbf{1}$ can be updated in a similar way. From (34) and (35), it can be seen that the cluster bases and coupling matrices of updated $\tilde{\mathbf{L}}_\mathbf{0}$, $\tilde{\mathbf{K}}_\mathbf{0}$, $\tilde{\mathbf{L}}_\mathbf{1}$, and $\tilde{\mathbf{K}}_\mathbf{1}$ all become scalar-based ones. Furthermore, they share the same cluster basis $\tilde{\mathbf{V}}_\mathbf{L}$.

The row cluster basis $\mathbf{V}^T_{p1}$ and column cluster basis $\mathbf{V}^T_{p2}$ of matrix $\mathbf{P}_0$ also need to be updated to one set of cluster basis. We collect $\mathbf{V}^T_{p1}$ and $\mathbf{V}^T_{p2}$ into one cluster basis in a form $[\mathbf{V}^T_{p1}\ \mathbf{V}^T_{p2}]$, and then orthogonalize it to obtain $\tilde{\mathbf{V}}^T_p$. By doing so, $\tilde{\mathbf{V}}^T_p$ contains all the independent column vectors of $\mathbf{V}^T_{p1}$ and $\mathbf{V}^T_{p2}$, and hence the $\mathcal{H}^2$-representation of $\mathbf{P}_0$ can be accurately updated by $\tilde{\mathbf{V}}^T_p$.

E. Computing H_{LK} and H_{SP}

In this subsection, we show how to compute $\mathbf{H}_{\mathbf{LK}}$ and $\mathbf{H}_{\mathbf{SP}}$ in (25) and obtain their $\mathcal{H}^2$-matrix representations.

Computing $\mathbf{H}_{\mathbf{LK}}$*:* In order to compute $\mathbf{H}_{\mathbf{LK}}$, we need to know $\mathbf{L}_\mathbf{0}$, $\mathbf{L}_\mathbf{1}^{-1}$, $\mathbf{K}_\mathbf{0}$, $\mathbf{K}_\mathbf{1}$ as can be seen from (25). The $\mathcal{H}^2$ representations of $\mathbf{L}_\mathbf{0}, \mathbf{K}_\mathbf{0}, \mathbf{L}_\mathbf{1}, \mathbf{K}_\mathbf{1}$ have been obtained as $\tilde{\mathbf{L}}_\mathbf{0}$, $\tilde{\mathbf{L}}_\mathbf{1}$, $\tilde{\mathbf{K}}_\mathbf{0}$, and $\tilde{\mathbf{K}}_\mathbf{1}$ in previous subsections. The $\mathbf{L}_\mathbf{1}^{-1}$ can be computed directly in linear complexity by the inverse algorithm in [15], and stored in an $\mathcal{H}^2$ matrix with $O(N)$ units. In addition, due to the block diagonal nature of $\mathbf{L}_\mathbf{1}$, we only need to compute the inverse of each diagonal block. For conductors having the same shape and size, their corresponding block inverses are the same, and hence only one block inverse needs to be computed. A discussion of the linear-time inverse algorithm in [15] will be given in the following Section IV-F.

With the $\mathcal{H}^2$ representations of $\mathbf{L}_\mathbf{0}$, $\mathbf{L}_\mathbf{1}^{-1}$, $\mathbf{K}_\mathbf{0}$, and $\mathbf{K}_\mathbf{1}$ known, the $\mathcal{H}^2$ representation of $\mathbf{H}_{\mathbf{LK}}$ shown in (25) can be constructed based on an $\mathcal{H}^2$-based matrix-matrix multiplication algorithm given in [21]. To be specific, by performing an $\mathcal{H}^2$-based matrix-matrix multiplication in the diagonal blocks, we can obtain an $\mathcal{H}^2$-based representation of $\tilde{\mathbf{L}}_\mathbf{1}^{-1}\tilde{\mathbf{K}}_\mathbf{1}$. By performing $\tilde{\mathbf{K}}_\mathbf{0} \cdot (\tilde{\mathbf{L}}_\mathbf{1}^{-1}\tilde{\mathbf{K}}_\mathbf{1})$ and adding it upon $\tilde{\mathbf{L}}_\mathbf{0}$, we can obtain an $\mathcal{H}^2$-based representation of $\mathbf{H}_{\mathbf{LK}}$. It is worth mentioning that due to the diagonal structure of both $\tilde{\mathbf{L}}_\mathbf{1}$ and $\tilde{\mathbf{K}}_\mathbf{1}$, matrix-matrix multiplications are only performed on the nonzero diagonal blocks. All these operations have a linear complexity since an $\mathcal{H}^2$-based matrix-matrix multiplication as well as addition has a linear complexity.

Computing $\mathbf{H}_{\mathbf{SP}}$*:* In order to compute $\mathbf{H}_{\mathbf{SP}}$, we need to first compute $(\hat{\mathbf{P}}_0)_{NC\times NC}$, which is the Schur complement of $(\mathbf{P}_0)_{NC\times NC}$, as can be seen from (25). For a given matrix $\mathbf{A}$, the Schur complement of its bottom rightmost block $\mathbf{A}_{22}$ can be evaluated during the computation of $\mathbf{A}$'s inverse based on Matrix Inversion Lemma [30]. In addition, the evaluation of $\mathbf{rhs}$ in (25) requires the inverse computation of $\mathbf{P}_0$ to obtain $(\mathbf{P}_\mathbf{0}^{-1})_{NC\times C}$. If we compute the inverse of $\mathbf{P}_0$, we can simultaneously obtain the Schur complement of $(\mathbf{P}_0)_{NC\times NC}$ and $(\mathbf{P}_\mathbf{0}^{-1})_{NC\times C}$ during the inverse process. The inverse of $\mathbf{P}_0$ can be directly computed using the linear-time inverse algorithm in [15]. Furthermore, since $\mathbf{P}_0$ as well as its inverse is symmetric, we only need to compute half of it.

Next, in order to represent $\mathbf{H}_{\mathbf{SP}}$ in (25) as an $\mathcal{H}^2$ matrix, we need to compute an $\mathcal{H}^2$-based product based on a sparse matrix φ_{sp} (or $\mathbf{J}_{\mathrm{sp}}$) and an $\mathcal{H}^2$ matrix $\hat{\mathbf{P}}_0$. The computation of such a matrix-matrix multiplication is much more efficient than that between two $\mathcal{H}^2$-based dense matrices. The underlying operation is equivalent to a few matrix–vector multiplications. When computing an $\mathcal{H}^2$-based matrix-matrix multiplication, we choose the partition formed by the first matrix's row cluster and the second matrix's column cluster as the partition of the matrix product.

F. Linear-Time Inverse-Based Direct Solution of Z

With the steps described in previous sections completed, we obtain an $\mathcal{H}^2$ matrix $\tilde{\mathbf{Z}}$ to represent the $\mathbf{Z}$ matrix in (24). We then employ the $O(N)$ $\mathcal{H}^2$-inverse algorithm [15] to compute $\tilde{\mathbf{Z}}^{-1}$ and employ the $O(N)$ $\mathcal{H}^2$-based matrix–vector multiplication algorithm [21], [28] to compute $\mathbf{rhs}$ as well as $[I_n] = \tilde{\mathbf{Z}}^{-1} \cdot \mathbf{rhs}$. Based on $[I_n]$, the current and the network parameters like impedance parameters can be extracted. If multiple right-hand sides are involved, instead of performing a matrix–vector multiplication for each right-hand side, we store the multiple right-hand sides in a sparse matrix and perform an $\mathcal{H}^2$-based matrix-matrix multiplication to obtain the solution in $O(N)$ time.

It is worth mentioning that although a matrix inverse and a matrix-matrix multiplication [21] share the same number of block matrix multiplications, there is a major difference that prevents one from using a fast matrix-matrix multiplication algorithm to achieve a linear complexity in inverse. The major difference is that in the level-by-level computation of the inverse, at each level, the computation is performed based on updated matrix blocks obtained from the computation at previous level instead of the original matrix. In contrast, in the level-by-level computation of the matrix-matrix multiplication, at each level, the computation is always performed based on the original matrix, which is never updated. This difference would render the inverse complexity higher than linear if one does not address it properly. In [15], we develop three new algorithms to render the total cost of an inverse linear. The first algorithm is an instantaneous collect operation for generating the auxiliary admissible block forms; the second algorithm is a modified block matrix multiplication algorithm; and the third one is an instantaneous split operation.

The main operation in the inverse algorithm is to perform fast block matrix multiplications based on orthogonalized and nested cluster basis. For example, after the orthogonalization, $\mathbf{V}^{s\mathrm{T}}\mathbf{V}^{s}=\mathbf{I}$ is satisfied for each cluster s, and hence an admissible block-based matrix multiplication encountered in the inverse procedure can be done based on

$$\begin{aligned}\mathbf{V}^{t}\mathbf{S}_1\mathbf{V}^{s^{\mathrm{T}}}\times\mathbf{V}^{s}\mathbf{S}_2\mathbf{V}^{r^{\mathrm{T}}}&=\mathbf{V}^{t}\mathbf{S}_1\left(\mathbf{V}^{s^{\mathrm{T}}}\times\mathbf{V}^{s}\right)\mathbf{S}_2\mathbf{V}^{r^{\mathrm{T}}}\\&=\mathbf{V}^{t}\mathbf{S}_1\mathbf{I}\mathbf{S}_2\mathbf{V}^{r^{\mathrm{T}}}\\&=\mathbf{V}^{t}(\mathbf{S}_1\times\mathbf{S}_2)\mathbf{V}^{r^{\mathrm{T}}}\end{aligned}$$

where only $\mathbf{S}_1\times\mathbf{S}_2$ needs to be computed, the cost of which is $O(k^3)$. The main purpose of the instantaneous collect operation and the instantaneous split operation is to keep the cost of each block matrix multiplication constant. Since each $\mathcal{H}^2$-block is computed from $O(C_{sp})$ block matrix multiplications and there are in total $O(C_{sp}N)$ blocks in an $\mathcal{H}^2$ matrix, the total cost of the inverse is $O(C_{sp}^2k^3)N$, which is linear. For nonstatic problems whose electric size is moderate, the rank can be increased by a rank function developed in [28] with the prescribed accuracy satisfied. The total cost of the resultant direct matrix solution is still linear as proved in [16].

G. Complexity Analysis

The storage of an $\mathcal{H}^2$-matrix is $O(N)$. The total computational cost for solving the impedance system (24) includes two parts: one is the $\mathcal{H}^2$-based construction of $\mathbf{Z}$ that includes the computation of $\mathbf{P}_0^{-1}$, $\mathbf{L}_1^{-1}$, $\mathbf{H_{SP}}$ and $\mathbf{H_{LK}}$, which is the cost of eliminating unknowns $[II_n]$, $[\rho]$, and $[\varphi]$ from (13)–(17), and the other is the direct solution of (24) by computing $\tilde{\mathbf{Z}}^{-1}$, each of which is performed in linear complexity as analyzed in the subsections above. As a result, the total complexity is linear.

H. Accuracy Analysis

We analyze the accuracy of the proposed direct matrix solution step by step based on the overall procedure shown in Fig. 1.

The first step is to construct an efficient $\mathcal{H}^2$ partition. The accuracy of this step can be controlled by η in (1).

The $\mathcal{H}^2$ representation of dense matrices $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, $\mathbf{K_1}$, and $\mathbf{P_0}$ is obtained by a degenerate approximation based on Lagrange interpolation given in (30). The error bound of this approximation is derived in [28]. It is shown that the error of (30) can be controlled to any desired order by the number of interpolation points p and parameter η used for $\mathcal{H}^2$ partition. For impedance extraction of 3-D lossy conductors considered in this work, the electric size of the underlying problem is not large, and hence a very small p is sufficient for achieving a good accuracy. This is also numerically demonstrated, as can be seen from examples given in the section of numerical results.

The $\mathcal{H}^2$ representation of sparse matrices φ_{sp} and $\mathbf{J}_{\mathrm{sp}}$ is exact, as explained in Section IV-C, and hence this step does not introduce any error.

Unifying the cluster bases is performed by orthogonalization. We do not reduce the rank of the cluster basis during the orthogonalization procedure, and hence no error is introduced.

To compute $\mathbf{H_{LK}}$, we first need to obtain $\mathbf{L_1^{-1}}$. It is obtained by the inverse algorithm developed in [15] with controlled accuracy. In this algorithm, we theoretically proved the existence of the $\mathcal{H}^2$ matrix representation of the inverse of the dense system matrix arising from a capacitance extraction problem, and revealed the relationship between the block cluster tree of the original matrix and that of its inverse. By using a similar proof, it can be shown that there exists an $\mathcal{H}^2$-representation of $\mathbf{L_1^{-1}}$. Moreover, $\mathbf{L_1^{-1}}$ shares the same block cluster tree as $\mathbf{L_1}$. Hence, the $\mathcal{H}^2$-partition developed for $\mathbf{L_1}$ is equally applicable to $\mathbf{L_1^{-1}}$. This fact can also be understood physically. Consider an admissible block $(\mathbf{L_1})^{t,s}$ formed by two subsets t and s of the entire unknown set. It represents the response (equivalent voltage) in subset t with the source (equivalent current) located in subset s. This block can be admissible because the admissibility condition is satisfied, i.e. the response is sampled in subset t that is sufficiently far from the source. Based on the same reasoning, $(\mathbf{L_1^{-1}})^{t,s}$ should be also admissible because it also represents the response in subset t that is sufficiently far from the source. The only difference is that in the inverse, the physical quantity that represents the source and the physical quantity that represents the response are reversed from those in the original matrix. For example, in $(\mathbf{L_1^{-1}})^{t,s}$, the source becomes an equivalent voltage, while the response is an equivalent current.

With $\mathbf{L_1^{-1}}$ obtained with controlled accuracy, the $\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$ is computed by an $\mathcal{H}^2$-based block-diagonal matrix multiplied by an $\mathcal{H}^2$-based block-diagonal matrix with controlled accuracy. The resultant matrix is also block diagonal. The $\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$ is then obtained from a formatted matrix-matrix multiplication between an $\mathcal{H}^2$-based dense matrix $\mathbf{K}_0$ and an $\mathcal{H}^2$-based block-diagonal matrix $\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$, with the $\mathcal{H}^2$ partition of the product matrix $(\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1)$ assumed to be the same as $\mathbf{K}_0$, and hence $\mathbf{L_0}$. This assumption is true because if a block is admissible in $\mathbf{K}_0$, it is also admissible in $\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$. This can be analyzed as follows. Consider a (t,s) block that is admissible in $\mathbf{K}_0$, since $\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$ is a block diagonal matrix, to compute the (t,s) block in $\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$, only $(s1,s)$ and $(s2,s)$ blocks in $\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$ will participate in the computation since all the other blocks in the columns representing s are zero, where $s1$ is the row cluster that forms an admissible block with column cluster s in the same conductor cluster, and $s2$ is the row cluster that forms an inadmissible block with s. The multiplication of $\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$ for obtaining the (t,s) block hence becomes

$$\left(\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1\right)^{t,s}=\mathbf{K}_0^{t,s1}\left(\mathbf{L}_1^{-1}\cdot\mathbf{K}_1\right)^{s1,s}+\mathbf{K}_0^{t,s2}\left(\mathbf{L}_1^{-1}\cdot\mathbf{K}_1\right)^{s2,s}. \quad (36)$$

In the first term in the right-hand side, although $\mathbf{K}_0^{t,s1}$ may not be admissible, the $(\mathbf{L}_1^{-1}\cdot\mathbf{K}_1)^{s1,s}$ is admissible, and hence the first term has a low rank form shown in (2); in the second term, since $(s2,s)$ is inadmissible, $s2$ must be physically close to s; since $\mathbf{K}_0^{t,s}$ is admissible, t must be sufficiently far from s, and hence $s2$. As a result, $\mathbf{K}_0^{t,s2}$ is admissible, and thereby the second term in (36) also has a low rank form. Hence, the resultant $(\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1)^{t,s}$ has a low-rank representation. Therefore, the (t,s) block that is admissible in $\mathbf{K}_0$ is also admissible in $\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$. Similarly, we can prove that an inadmissible block in $\mathbf{K}_0$ is also inadmissible in $\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$. As a result, the $\mathcal{H}^2$ partition constructed for $\mathbf{K}_0$ and $\mathbf{L_0}$ is equally applicable to $\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$. Thus, based on this partition, the resultant formatted $\mathcal{H}^2$-based matrix-matrix multiplication has a controlled accuracy. The addition of $\mathbf{L_0}$ and $\mathbf{K}_0\cdot\mathbf{L}_1^{-1}\cdot\mathbf{K}_1$ is then performed based on a straightforward $\mathcal{H}^2$-based matrix-matrix addition without introducing additional error.

To compute $\mathbf{H_{sp}}$, we first need to obtain $(\widehat{\mathbf{P}}_0)_{NC\times NC}$. This is obtained together with $(\mathbf{P_0^{-1}})_{NC\times C}$ (required in the computation of $\mathbf{rhs}$) by computing $\mathbf{P_0^{-1}}$. The $\mathbf{P_0^{-1}}$ is obtained with controlled accuracy by the inverse algorithm in [15]. The matrix product $(\varphi_{\rm sp})_{N\times NC}\cdot(\widehat{\mathbf{P}}_0)_{NC\times NC}\cdot\mathbf{J}_{\rm sp}$ is performed based on an $\mathcal{H}^2$-based formatted matrix-matrix multiplication. The product tree is formed between the row cluster tree of $\varphi_{\rm sp}$ and the column cluster tree of $\mathbf{J}_{\rm sp}$, both of which is the cluster tree constructed for the RWG vector basis functions. In the formatted matrix-matrix multiplication, we assume that the $\mathcal{H}^2$ partition of the product matrix $(\varphi_{\rm sp})_{N\times NC}\cdot(\widehat{\mathbf{P}}_0)_{NC\times NC}\cdot\mathbf{J}_{\rm sp}$ is the same as that of $\mathbf{L}_i$ and $\mathbf{K}_i$ $(i=0)$ matrices. This again can be proved to be true as follows. Consider a (t,s) block in the partition formed based on the RWG-basis-based cluster tree, if (t,s) satisfies the admissibility condition, we can prove that the corresponding block in $(\varphi_{\rm sp})_{N\times NC}\cdot(\widehat{\mathbf{P}}_0)_{NC\times NC}\cdot\mathbf{J}_{\rm sp}$ has a low rank form. This is because

$$\begin{aligned}\left((\varphi_{\rm sp})_{N\times NC}\cdot(\widehat{\mathbf{P}}_0)_{NC\times NC}\cdot\mathbf{J}_{\rm sp}\right)^{t,s}&=(\varphi_{\rm sp})_{N\times NC}^{t,t'}(\widehat{\mathbf{P}}_0)_{N\times NC}^{t',s'}\mathbf{J}_{\rm sp}^{s',s}\\&=(\varphi_{\rm sp})_{N\times NC}^{t,t'}\mathbf{V}^{t'}\mathbf{S}^{t',s'}\mathbf{V}^{s'T}\mathbf{J}_{\rm sp}^{s',s}\end{aligned}$$

which is low rank. In the above, we only consider the (t,t') block in $\varphi_{\rm sp}$, where t' includes basis functions that directly interact with t, because $\varphi_{\rm sp}$ is a sparse matrix whose nonzero elements only appear in the close interaction. Similarly, in $\mathbf{J}_{\rm sp}$, we only need to consider the (s,s')-block, where s' includes basis functions that directly interact with s. As a consequence, in $\widehat{\mathbf{P}}_0$, only the (t',s') block is involved in the computation. Since (t,s) satisfies the admissibility condition, and t' is close to t, and s' is close to s, (t',s') also satisfies the admissibility condition. Thus the (t',s') block is admissible in $\widehat{\mathbf{P}}_0$. Therefore, it can be replaced by a factorized low-rank form, from which it can be seen clearly that the (t,s) block has a low rank form in $(\varphi_{\rm sp})_{N\times NC}\cdot(\widehat{\mathbf{P}}_0)_{NC\times NC}\cdot\mathbf{J}_{\rm sp}$, and hence is also admissible.

The addition of $\mathbf{H_{LK}}$ and $\mathbf{H_{SP}}$ is then performed based on a straightforward $\mathcal{H}^2$-based matrix-matrix addition without introducing additional error, from which $\mathbf{Z}$ is obtained. The error controlled inverse algorithm [15] is then employed to obtain $\mathbf{Z}^{-1}$.

From the aforementioned analysis, it can be seen that the error is well controlled at each step of the proposed direct matrix solution. Therefore, the accuracy of the entire solution can be controlled to a desired order.

V. NUMERICAL RESULTS

For all the examples simulated in this work, we choose *leafsize* $=8$ and $\eta=1$ for the construction of $\mathcal{H}^2$ partition; and we use $p=3$ for the interpolation along each dimension. All the simulation is done on a single 3 GHz CPU. A surrounding medium with relative permittivity equal to 1 is considered in all examples. The mesh size is not restricted by the skin depth because of the underlying surface-based formulation that rigorously accounts for skin effects. The mesh size is chosen to capture the field variation with space on the conducting surface. Whether the mesh is fine enough or not is numerically judged by checking the convergence of the impedance result.

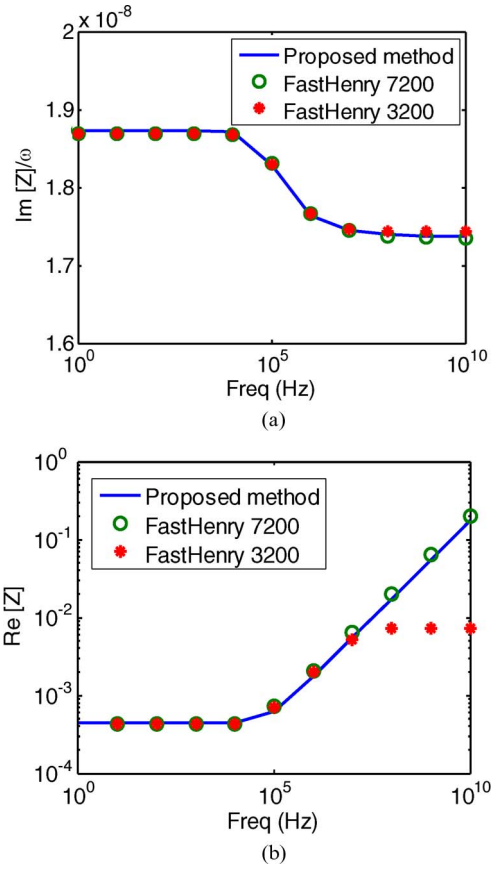


Fig. 4. *Impedance of a straight conductor wire from 1 Hz to 10 GHz. (a) Imaginary part. (b) Real part.*

A. A Straight Conductor Wire

We first use a simple conductor wire to test the accuracy of the proposed method for wideband impedance extraction. The wire has a cross section of 1 mm by 1 mm and a length of 25 mm. The conductivity of the wire is 5.8e + 7 S/m. The discretisation results in 630 triangular panels. Both real part and imaginary part of the impedance computed by the proposed method are compared with those extracted by FastHenry [32], which is based on a volume IE method accelerated by fast multipole algorithm. The inductances generated by both methods agree with each other very well as can be seen from Fig. 4(a). For the real part of the impedance as shown in Fig. 4(b), FastHenry has to use a dense mesh with 7200 filaments to capture the skin effect in the higher frequency band, whereas the proposed surface-based method can still capture the frequency dependency with a 630-panel-based discretization.

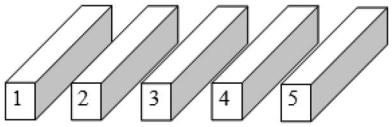


Fig. 5. *A 5-bus structure.*

The total solution time of the proposed solver across the entire frequency band is 22.2 s, while the total solution time of FastHenry across the entire frequency band is 434.5 s.

B. A 1 × 5 Bus Structure

We next simulated a 5-bus structure. Each bus has a size of 2 μm × 2 μm × 20 μm as shown in Fig. 5. The distance between two adjacent conductors is 5 μm. Each conductor has the same conductivity as that used in the previous example. The frequency point for extraction is 10 GHz. The discretization of the problem generates 1440 triangular panels. The impedances (in Ω) extracted by the proposed method are shown in the equation at the bottom of the page. The error of the impedance matrix is 0.42%, which is computed by $\|\mathbf{Z}_{H2} - \mathbf{Z}_{FH}\|_F / \|\mathbf{Z}_{FH}\|_F$, where $\mathbf{Z}_{H2}$ is obtained by the proposed $\mathcal{H}^2$-based direct solver and $\mathbf{Z}_{FH}$ is obtained by FastHenry with 12 000 filaments.

The total solution time of the proposed solver is 8.1 s, while the total solution time of FastHenry is 458.3 s.

C. Large-Scale Spiral Inductor Array

In this example, we consider an inductor array composed of rectangular spirals, each of which has 4 full turns. The width, thickness, and spacing of the rectangular spiral are 1, 1, and 1 μm, respectively, and the inner diameter of the rectangular spiral is 10 μm. The array includes $2 \times 2, 2 \times 4, 2 \times 8, 2 \times 16$ rectangular spirals, as shown in Fig. 6(a), which, respectively, results in 72 692, 145 384, 290 768, and 581 536 number of unknowns. For a fair comparison, a similar discretisation is used in FastImp [31] and the residual error of GMRES used by FastImp is set as 10^{-2}. The frequency is 1 GHz. Fig. 6(b) and (c) show the memory and the total solution time for both methods. As can be clearly seen, the total solution time of the proposed method scales linearly with

$$Z = \begin{bmatrix} 9.87 + i71.23 & 0.062 + i26.78 & -0.105 + i16.09 & -0.091 + i11.37 & -0.076 + i8.74 \\ 0.062 + i26.78 & 9.98 + i71.10 & 0.114 + i26.73 & -0.080 + i16.08 & -0.091 + i11.37 \\ -0.105 + i16.10 & 0.113 + i26.73 & 9.99 + i71.18 & 0.113 + i26.73 & -0.105 + i16.10 \\ -0.091 + i11.37 & -0.080 + i16.08 & 0.114 + i26.73 & 9.98 + i71.10 & 0.062 + i26.78 \\ -0.076 + i8.74 & -0.091 + i11.37 & -0.105 + i16.10 & 0.062 + i26.78 & 9.88 + i71.24 \end{bmatrix} \times 10^{-2}.$$

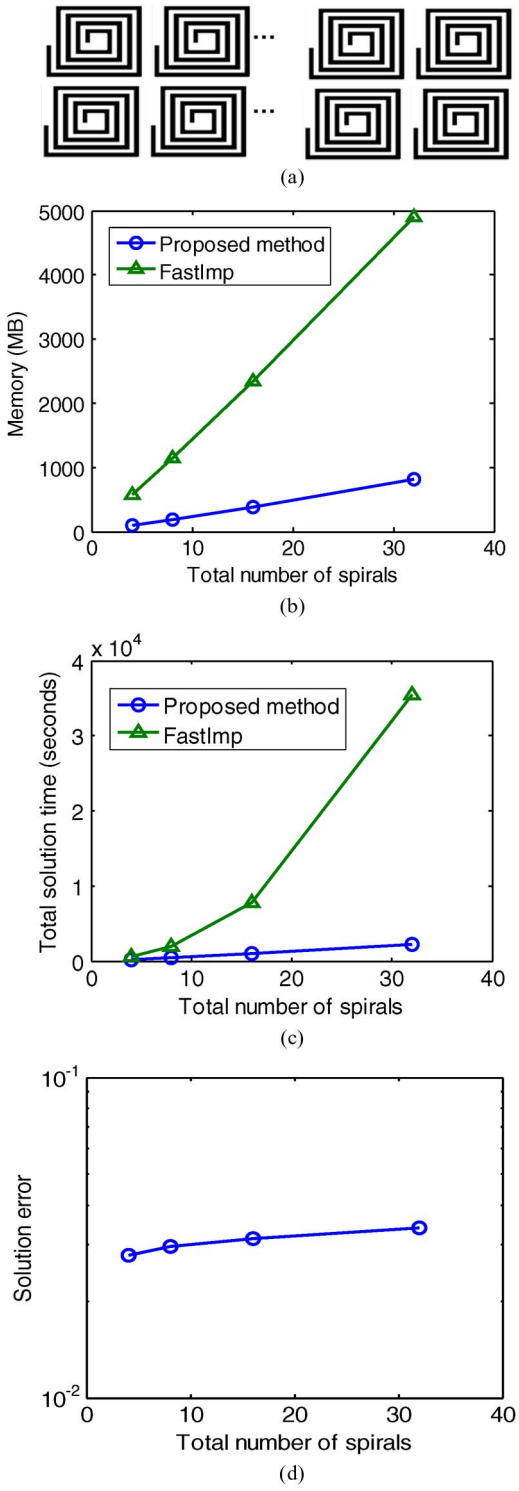


Fig. 6. *Simulation of an array composed of rectangular spirals. (a) Inductor Array. (b) Memory. (c) Total solution time. (d) Solution error.*

the number of unknowns, whereas the total solution time of FastImp is affected by the number of iterations and the number of right-hand sides, and hence FastImp failed to

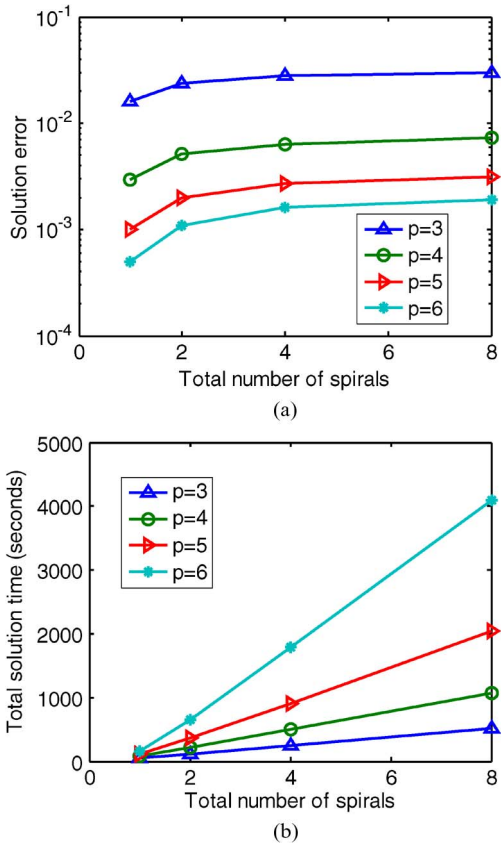


Fig. 7. *Performance of the proposed solver in achieving a higher order of accuracy. (a) Solution error. (b) Total solution time.*

show linear complexity although its matrix–vector multiplication achieved almost linear complexity. In Fig. 6(d), we plot the solution error measured by $\|\tilde{\mathbf{Z}}I - \mathbf{rhs}\|/\|\mathbf{rhs}\|$ of the proposed method. Good accuracy is observed in the entire unknown range.

Next, we tested the capability of the proposed solver in achieving a higher order of accuracy. The structure was the spiral inductor array shown in Fig. 6(a). The arrays simulated had 1, 2, 4, and 8 rectangular spirals, respectively. The simulation parameters $n_{\min}$ and η remained the same, but the number of interpolation points p was increased from 3 to 6. Fig. 7(a) and (b), respectively, show the solution error measured by $\|\tilde{\mathbf{Z}}I - \mathbf{rhs}\|/\|\mathbf{rhs}\|$ and the total solution time of the proposed solver. As can be seen clearly, by increasing p, the solution error is reduced without sacrificing the linear time complexity.

D. Large-Scale 3-D Buses in Multiple Layers

The last example is a crossover bus structure in two layers and each layer has m conductors. Each conductor has a size of 1 μm $\times$ 1 μm $\times$ $(2m+1)$ μm and a conductivity of 5.8e + 7 S/m. The m is chosen as 1, 2, 4, 8, 16, 32, 64, which, respectively, results in 800, 4480, 14 080, 48 640, 179 200, 686 080, 2 682 880 number of unknowns.

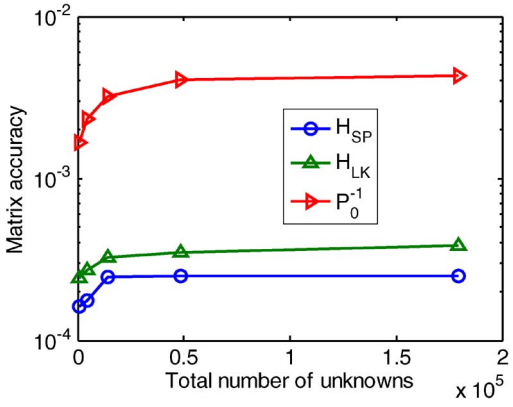


Fig. 8. *Matrix accuracy.*

The impedance is extracted at 1 GHz. First, we test the accuracy of the matrices shown in (25) by evaluating $\|\mathbf{H_{SP}} - \tilde{\mathbf{H}}_{\mathbf{SP}}\|_F/\|\mathbf{H_{SP}}\|_F$, $\|\mathbf{H_{LK}} - \tilde{\mathbf{H}}_{\mathbf{LK}}\|_F/\|\mathbf{H_{LK}}\|_F$, and $\|\tilde{\mathbf{P}}_0^{-1}\mathbf{P}_0 - \mathbf{I}\|_F/\|\mathbf{I}\|_F$, where $\mathbf{H_{SP}}, \mathbf{H_{LK}}$ are both full matrix representations while $\tilde{\mathbf{H}}_{\mathbf{SP}}, \tilde{\mathbf{H}}_{\mathbf{LK}}$ are their corresponding $\mathcal{H}^2$ representations, $\tilde{\mathbf{P}}_{\mathbf{0}}^{-1}$ is the $\mathcal{H}^2$-based inverse of $\mathbf{P}_0$, and $\mathbf{I}$ is an identity matrix. Good accuracy of $\tilde{\mathbf{H}}_{\mathbf{SP}}$, $\tilde{\mathbf{H}}_{\mathbf{LK}}$ and $\tilde{\mathbf{P}}_0^{-1}$ is achieved in the entire unknown range, as can be seen from Fig. 8, and the worst error is 2.55E-4, 3.81E-4 and 4.39E-3, respectively. Since the assessment of matrix accuracy requires the computation of full matrices, we only tested the accuracy of the matrix for m up to 16. For larger cases, we used reference solutions to test the solution accuracy of the proposed method as can be seen from the following test.

We examined the efficiency of the proposed direct solver and compared its performance with FastImp. For a fair comparison, a similar discretization was used in both methods. The relative residual in GMRES used by FastImp was set as 10^{-3}. The FastImp was only used to simulate the bus structure with m up to 32 since the advantage of the proposed solver was already obvious. Fig. 9(a) shows the memory consumption by both methods. It can be seen that the memory required by the proposed method is eight times less than that required by FastImp. The time for one matrix–vector multiplication is plotted in Fig. 9(b), which shows that the proposed scheme is about 18 times faster than FastImp. Fig. 9(c) shows the total solution time of the proposed method that includes the cost of each step in Fig. 1. As can be seen clearly, the total solution time of the proposed method scales linearly with the number of unknowns. It is also much faster than FastImp even though the proposed method computes the entire inverse, whereas FastImp only solves for $2m$ right-hand sides. In addition, we tested the accuracy of the extracted impedance. One column of the impedance matrix was extracted. We tested the solution accuracy of both the proposed method and FastImp based on $\|\mathbf{Z}_1 - \mathbf{Z}_1'\|/\|\mathbf{Z}_1\|$, where $\mathbf{Z}_1$ is a vector of

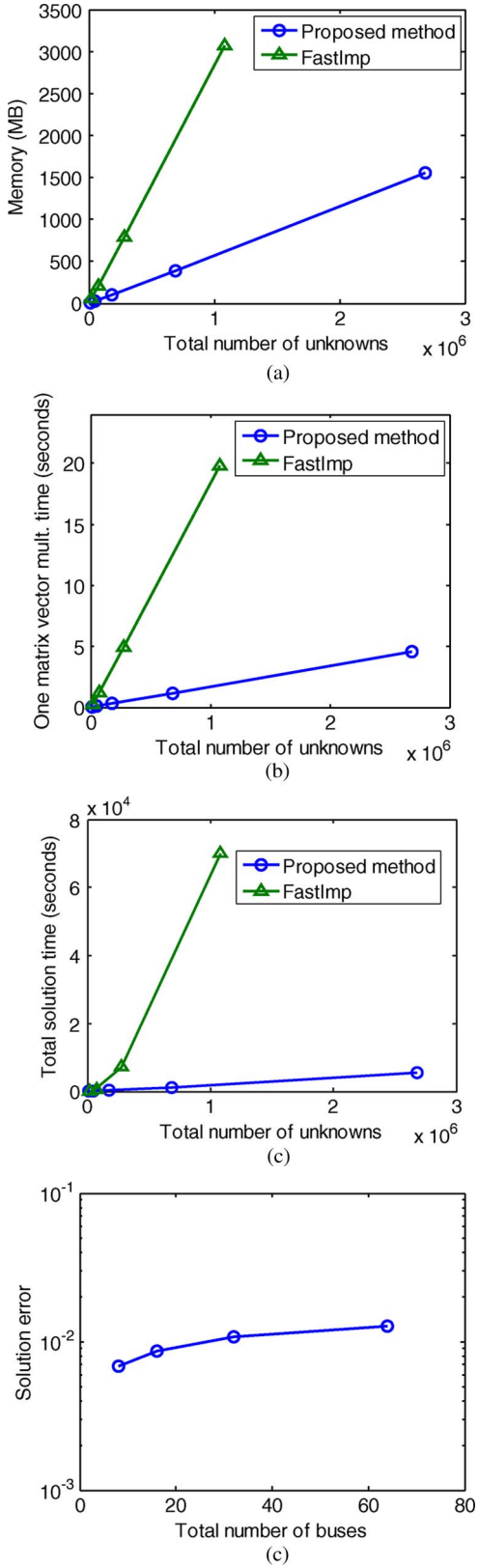


Fig. 9. *Simulation of an $m \times m$ 3-D bus. (a) Memory. (b) Time for one matrix vector product. (c) Total solution time. (d) Impedance error compared to FastImp.*

impedances between port 1 and other ports computed by a *full-matrix*-based direct solver, while $\mathbf{Z}_1'$ is computed by either the proposed method or FastImp. Because the full-matrix-based direct solver is very expensive, we only used this approach to assess error for small bus structures. It is shown that the accuracy of the proposed method is 0.32%, and 0.47%, respectively, for the bus structure with $m = 1$ and $m = 2$. For the same two structures, the accuracy of FastImp is 0.70%, and 0.89%, respectively. Because both methods achieved very good accuracy, we used FastImp to benchmark the accuracy of the proposed method for larger buses based on $\|\mathbf{Z}_{\text{this}} - \mathbf{Z}_{\text{fastimp}}\|/\|\mathbf{Z}_{\text{fastimp}}\|$, where $\mathbf{Z}_{\text{this}}$ is computed by the proposed method, and $\mathbf{Z}_{\text{fastimp}}$ is computed by FastImp. As can be seen from Fig. 9(d), excellent accuracy is observed in the entire unknown range.

VI. CONCLUSION

A direct solver of linear complexity is developed for the surface integral-based impedance extraction of arbitrarily shaped nonideal 3-D conductors embedded in a dielectric material. It successfully solves a highly irregular system matrix mixed with both dense and sparse blocks in linear CPU time and memory consumption, with controlled accuracy. Numerical results demonstrate its superior performance.

Impedance parameters ($\mathbf{Z}$-parameters) are one kind of network parameters to characterize a circuit. From the impedance parameters or the voltage and current results obtained from the proposed solver, one can also obtain any other circuit parameters of interest such as admittance ($\mathbf{Y}$) parameters and scattering ($\mathbf{S}$) parameters. ■

REFERENCES

[1] K. Nabors and J. White, "FastCap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Comput.-Aided Design*, vol. 10, no. 11, pp. 1447–1459, Nov. 1991.

[2] Y. C. Pan, W. C. Chew, and L. X. Wan, "A fast multipole-method based calculation of the capacitance matrix for multiple conductors above stratified dielectric media," *IEEE Trans. Microw. Theory Tech.*, vol. 49, no. 3, pp. 480–490, Mar. 2001.

[3] S. Kapur and D. E. Long, "IES^3: A fast integral equation solver for efficient 3-dimensional extraction," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 1997, pp. 448–455.

[4] D. Gope and V. Jandhyala, "Efficient solution of EFIE via low-rank compression of multilevel predetermined interactions," *IEEE Trans. Antennas Propag.*, vol. 53, no. 10, pp. 3324–3333, Oct. 2005.

[5] W. Shi, J. Liu, N. Kakani, and T. Yu, "A fast hierarchical algorithm for 3-D capacitance extraction," *IEEE Trans. Comput.-Aided Design*, vol. 21, no. 3, pp. 330–336, 2002.

[6] Y. Yi, P. Li, V. Sarin, and W. Shi, "A Preconditioned hierarchical algorithm for impedance extraction of three-dimensional structures with multiple dielectrics," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 11, pp. 1918–1927, Nov. 2008.

[7] R. Jiang, Y.-H. Chang, and C. C.-P. Chen, "ICCAP: A Linear time sparsification and reordering algorithm for 3D BEM capacitance extraction," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 7, pp. 3060–3068, Jul. 2006.

[8] J. R. Phillips and J. White, "A precorrected FFT method for capacitance extraction of complicated 3-D structures," in *Proc. Int. Conf. Comput.-Aided Design*, 1994, pp. 268–271.

[9] Z. Zhu, B. Song, and J. White, "Algorithms in FastImp: A fast and wideband impedance extraction program for complicated 3-D geometries," in *Proc. 40th ACM/EDAC/IEEE Design Automat. Conf*, 2003, pp. 712–717.

[10] D. Gope, I. Chowdhury, and V. Jandhyala, "DiMES: Multilevel fast direct solver based on multipole expansions for parasitic extraction of massively coupled 3D microelectronic structures," *Proc. ACM/EDAC/IEEE Design Autom. Conf.*, 2005, pp. 159–162.

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

[12] R. J. Adams, Y. Xu, X. Xu, S. D. Gedney, and F. X. Canning, "Modular fast direct electromagnetic analysis using local-global solution modes," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2427–2441, Aug. 2008.

[13] L. Greengard, D. Gueyffier, P.-G. Martinnson, and V. Rokhlin, "Fast direct solvers for integral equations in complex three-dimensional domains," *Acta Numerica*, vol. 18, pp. 243–275, May 2009.

[14] W. Chai, D. Jiao, and C. C. Koh, "A Direct integral-equation solver of linear complexity for large-scale 3D capacitance and impedance extraction," in *Proc. 46th ACM/EDAC/IEEE Design Automat. Conf.*, Jul. 2009, pp. 752–757.

[15] W. Chai and D. Jiao, "Dense matrix inversion of linear complexity for integral-equation based large-scale 3-D capacitance extraction," *IEEE Trans. Microw. Theory Tech*, vol. 59, no. 10, pp. 2404–2421, Oct. 2011.

[16] W. Chai and D. Jiao, "An LU decomposition based direct integral equation solver of linear complexity and higher-order accuracy for large-scale interconnect extraction," *IEEE Trans. Adv. Packag.*, vol. 33, no. 4, pp. 794–803, Nov. 2010.

[17] W. Chai and D. Jiao, "A complexity-reduced $\mathcal{H}$-matrix based direct integral equation solver with prescribed accuracy for large-scale electrodynamic analysis," in *Proc. IEEE Int. Symp. Antennas Propag.*, Jun. 2010, DOI: 10.1109/APS.2010.5561966.

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

[19] S. Börm, "$\mathcal{H}^2$-matrices—Multilevel methods for the approximation of integral operators," *Comput. Vis. Sci.*, vol. 7, pp. 173–181, 2004.

[20] S. Börm and W. Hackbusch, "$\mathcal{H}^2$-matrix approximation of integral operators by interpolation," *Appl. Numer. Math.*, vol. 43, pp. 129–143, 2002.

[21] S. Börm, "$\mathcal{H}^2$-matrix arithmetics in linear complexity," *Computing*, vol. 77, pp. 1–28, 2006.

[22] B. Song, Z. Zhu, J. Rockway, and J. White, "A new surface integral formulation for wideband impedance extraction of 3-D structures," in *Proc. Int. Conf. Comput.-Aided Design*, 2003, pp. 843–847.

[23] W. Chai and D. Jiao, "Direct matrix solution of linear complexity for surface integral-equation based impedance extraction of high bandwidth interconnects," in *Proc. 48th ACM/EDAC/IEEE Design Autom. Conf.*, Jun. 2011, pp. 206–211.

[24] W. Hackbusch and B. Khoromskij, "A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices," *Computing*, vol. 62, pp. 89–108, 1999.

[25] W. Hackbusch and B. Khoromskij, "A sparse matrix arithmetic. Part II: Application to multi-dimensional problems," *Computing*, vol. 64, pp. 21–47, 2000.

[26] S. Börm, "Introduction to hierarchical matrices with applications," *Eng. Anal. Boundary Elements (EABE)*, vol. 27, pp. 405–422, 2003.

[27] K. Zhao, M. N. Vouvakis, and J. Lee, "The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems," *IEEE Trans. Electromagn. Compat.*, vol. 47, no. 4, pp. 763–773, Nov. 2005.

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

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

[30] Wikipedia, *Invertible Matrix*, 1923. [Online]. Available: http://en.wikipedia.org/wiki/Invertible_matrix

[31] *FastImp: A Fast Impedance Tool*. [Online]. Available: http://www.mit.edu/people/zhzhu/fastImp.html

[32] Computational Prototyping Group. [Online]. Available: http://www.rle.mit.edu/cpg/research_codes.htm

[33] S. Börm, "Approximation of integral operators by $\mathcal{H}^2$-matrices with adaptive bases," *Computing*, vol. 74, pp. 249–271, 2005.

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

[35] S. J. Kwon, K. Du, and R. Mittra, "Characteristic basis function method: A numerically efficient technique for analyzing microwave and RF circuits," *Microw. Opt. Technol. Lett.*, vol. 38, no. 6, pp. 444–448, Jul. 2003.

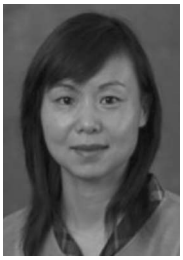
[36] G. Bianconi, C. Pelletti, R. Mittra, K. Du, and A. Monorchio, "An efficient technique for the evaluation of the reduced matrix in the context of the CBFM for layered media," *IEEE Antenna Wireless Propag. Lett.*, vol. 10, pp. 674–677, 2011.

ABOUT THE AUTHORS

Wenwen Chai received the B.S. degree from the University of Science and Technology of China, Hefei, Anhui, China, in 2004 and the M.S. degree from the Chinese Academy of Sciences, Beijing, China, in 2007, both in electrical engineering. She is currently working toward the Ph.D. degree in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

She works in the On-Chip Electromagnetics Group, Purdue University. Her research is focused on computational electromagnetics, high-performance very large scale integration (VLSI) computer-aided design (CAD), and fast and high-capacity numerical methods.

Ms. Chai received the IEEE Antennas and Propagation Society Doctoral Research Award for 2009–2010.



Dan Jiao (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, in October 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 160 papers in refereed journals and international conferences. Her current research interests include computational electromagnetics, high-frequency digital, analog, mixed-signal, and radio-frequency integrated circuit (RFIC) design and analysis, high-performance very large scale integration (VLSI) computer-aided design (CAD), modeling and simulation of micro- and nanoscale 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 bioelectromagnetics.

Dr. Jiao was among 100 engineers chosen for the National Academy of Engineering's 2011 U.S. Frontiers of Engineering Symposium. She received the Ruth and Joel Spira Outstanding Teaching Award in 2010. In 2008, she received the National Science Foundation (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 Office of Naval Research (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 crosstalk. She was the winner of the 2000 Raj Mittra Outstanding Research Award given her by the University of Illinois at Urbana-Champaign.