A Linear Complexity Direct Volume Integral Equation Solver for Full-Wave 3-D Circuit Extraction in Inhomogeneous Materials

Saad Omar, *Member, IEEE*, and Dan Jiao, *Senior Member, IEEE*

***Abstract*—An $\mathcal{H}^2$-matrix based linear complexity direct matrix solution is developed for the volume integral equation (VIE) based broadband full-wave extraction of general 3-D circuits. Such circuits are in general electrically small or moderate, but contain arbitrarily shaped lossy conductors immersed in inhomogeneous dielectrics with ports located anywhere in the physical layout of the circuit. In the proposed direct solver, we first develop a well-conditioned VIE formulation without sacrificing the rigor and the advantages of the prevailing formulations. This formulation facilitates a robust direct solution of good accuracy even with a rank-1 representation. We then overcome the numerical challenge of solving the resultant highly unstructured system matrix mixed with both square and rectangular dense and sparse matrices by developing a fast linear complexity direct solution. This direct solution is capable of inverting dense matrices involving over 2 million unknowns in less than 1 h on a single CPU core running at 3 GHz. Numerical simulations of large-scale 3-D circuits and comparisons with state-of-the-art linear complexity iterative VIE solvers have demonstrated the accuracy, efficiency, and linear complexity of the proposed direct VIE solver.**

***Index Terms*—Circuit modeling, fast solvers, full-wave analysis, impedance extraction, linear complexity solvers, S-parameter extraction, 3-D structures, volume integral equations (VIEs).**

I. Introduction

MANY real-world problems are complicated in geometry and inhomogeneous in materials, which are also exposed to ambient conditions. The volume integral equation (VIE) based methods [1]–[9] have a great flexibility in modeling both complicated geometry and inhomogeneous materials in open-region settings. The new VIE formulation in [10]–[12] has an added capability of permitting a simultaneous circuit-scattering analysis of circuits exposed to external fields. It also allows for voltage ports to be assigned at an arbitrary point in the physical layout of the circuits. Notable applications of such a simultaneous circuit-scattering analysis include the design of microwave and RF circuits in communication satellites and military integrated circuits operating in the battlefield.

From a practical point of view, the advantages of the VIE-based analyses can only be fully accentuated if they can be performed with low computational complexity. Prevailing fast integral-equation solvers are mainly iterative solvers [4], [13]–[20] that accelerate dense matrix–vector multiplications. The best complexity that can be achieved by an iterative solver is $O(N_{\rm rhs}N_{\rm it}N)$, with $N_{\rm rhs}$ being the number of right-hand sides, $N_{\rm it}$ the number of iterations, and N the number of unknowns. This complexity can be high when the number of ports of the circuit under analysis and/or the number of iterations are large.

There have been significant contributions in fast direct solvers [21]–[34] for the analysis of problems ranging from circuits to scattering problems. A direct solver has a potential of achieving the optimal complexity, which is linear complexity $O(N)$ for solving a problem of N unknowns. This complexity is independent of the number of right-hand sides. Such an optimal complexity has been achieved in [24], [26]–[28], and [30] for the surface IE based capacitance, as well as full-wave extraction of arbitrarily shaped 3-D circuits. With such a linear complexity direct solver, a dense matrix resulting from the surface IE based circuit extraction involving over 3.71 million unknowns has been inverted in fast CPU time (1.6 h) with modest memory consumption (4.4 GB) [30]. Comparisons with state-of-the-art fast solvers, including the fast multipole method (FMM), have shown a clear advantage of the $O(N)$ direct surface-IE solvers in both CPU time and memory consumption [24], [26]–[28], [30].

As compared to a surface IE solver, a VIE solver is flexible in modeling arbitrary inhomogeneity. However, no $O(N)$ direct VIE solver has been accomplished for the full-wave extraction of 3-D circuits with arbitrarily shaped nonideal conductors embedded in inhomogeneous dielectrics. The contribution of this paper is such a direct VIE solver. In this direct solver, in view of the ill-conditioning issue of the full-wave analysis of circuits from low to high frequencies [35], we first propose a well-conditioned VIE formulation. This formulation is then cast into a well-conditioned reduced form facilitating a linear complexity numerical inverse. The proposed well-conditioned formulation retains all the inherent advantages of the VIE formulation developed in [12]. It thus possesses first-principles based accuracy, and permits circuit parameter extraction at ports located

Manuscript received May 01, 2014; revised July 26, 2014, September 26, 2014, December 10, 2014, and January 08, 2015; accepted January 12, 2015. Date of publication February 09, 2015; date of current version March 03, 2015. This work was supported by the National Science Foundation (NSF) under Grant 0747578, by the SRC under Grant Task 292.073, and by the Defense Advanced Projects Research Agency (DARPA) under Grant HR0011-14-1-0057.

S. Omar was with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. He is now with Schlumberger-Doll Research, Cambridge, MA 02139 USA.

D. Jiao is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: 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.2015.2396494

anywhere in the physical structure of the circuit. More importantly, the new formulation reduces the condition number of the VIE system matrix to be inverted by many orders of magnitude. We then develop an $O(N)$ direct solution to solve the resulting unstructured system matrix composed of both square and rectangular matrices that among themselves can either be dense or sparse. Numerical experiments validate the accuracy, superior performance, and capability of the proposed new VIE solver. The advantages of the proposed well-conditioned VIE formulation and its fast linear complexity direct solution pay dividends while inverting a dense VIE matrix of over 1 million unknowns in as low as 5 GB of memory and 4 min on a single CPU running at 3 GHz. This paper is a significant theoretical and performance enhancing extension of our conference paper [36]. An engineering based explanation of the underlying mathematical $\mathcal{H}^2$-matrix framework and the implementation details are also presented to aid novice readers.

This paper is organized as follows. In Section II, we review the mathematical background of the $\mathcal{H}^2$-matrix framework and provide an engineering view to this framework. We also review the direct matrix solutions we have developed in the past and describe the research problems and challenges encountered in this work. In Section III, we present our proposed well-conditioned VIE formulation for broadband circuit modeling in inhomogeneous materials. In Section IV, we elaborate how to generate a well-conditioned numerical system from the proposed formulation, which can also be readily represented in $\mathcal{H}^2$-matrix format. In Section V, the proposed linear-complexity direct matrix solution is described. In Section VI, a theoretical analysis is presented on the complexity and accuracy of the proposed direct solution. In Section VII, numerical results are given to demonstrate the performance of the proposed direct solver for circuit modeling. The solver demonstrates a clear linear complexity in both CPU time and memory consumption with controllable accuracy. Comparisons with linear complexity iterative solvers have demonstrated a clear advantage of the proposed direct solver. Section VIII concludes this paper.

II. Background and Problem Description

A. Mathematical Background

The $\mathcal{H}$ (hierarchical)-matrix is a general mathematical framework [37]–[40] for compact representation and efficient computation of dense matrices. The $\mathcal{H}^2$-matrix [41]–[44] is a specialized subclass of hierarchical matrices. It has a much reduced computational complexity due to the use of a nested structure. A basic overview of this framework can be found from [28, Sec. II]. Here, for completeness of this paper, we provide a few mathematical definitions.

Consider two subsets t and s of the entire unknown set $\mathcal{I} := \{1, 2, \ldots, N\}$. The two sets are said to be admissible if they satisfy the following strong admissibility condition [37]

$$\max\{\mathrm{diam}(\Omega_t), \mathrm{diam}(\Omega_s)\} \leq \eta \mathrm{dist}(\Omega_t, \Omega_s) \tag{1}$$

where η is a positive parameter, Ω_t, and Ω_s are the supports of t and s in space, $\mathrm{diam}(.)$ denotes the Euclidean diameter of a set, and $\mathrm{dist}(.,.)$ stands for the Euclidean distance between two sets. In an $\mathcal{H}^2$-matrix, an admissible block $\mathbf{Z}^{t,s}$ is represented by the following factorized form:

$$\begin{aligned} \mathbf{Z}^{t,s} &= \mathbf{V}^t \mathbf{S}^{t,s} \mathbf{V}^{sT} \\ \mathbf{V}^t &\in \mathbb{C}^{\#t \times K^t} \\ \mathbf{S}^{t,s} &\in \mathbb{C}^{K^t \times K^s} \\ \mathbf{V}^s &\in \mathbb{C}^{\#s \times K^s} \end{aligned} \tag{2}$$

where $\mathbf{V}^t(\mathbf{V}^s)$ is called a cluster basis associated with $t(s)$, $\mathbf{S}^{t,s}$ is called a coupling matrix, $K^{t(s)}$ denotes the rank of $\mathbf{V}^t(\mathbf{V}^s)$, and $\#$ denotes the cardinality of a set. The hierarchical dependence of the unknowns is stored in a tree structure with each node of the tree called a cluster, and the tree a cluster tree. In an $\mathcal{H}^2$-matrix, the cluster basis $\mathbf{V}^t$ is nested, which can be represented by its two children's cluster bases as the following:

$$\mathbf{V}^t = \begin{bmatrix} \mathbf{V}^{t_1}\mathbf{T}^{t_1} \\ \mathbf{V}^{t_2}\mathbf{T}^{t_2} \end{bmatrix} = \begin{bmatrix} \mathbf{V}^{t_1} & 0 \\ 0 & \mathbf{V}^{t_2} \end{bmatrix} \begin{bmatrix} \mathbf{T}^{t_1} \\ \mathbf{T}^{t_2} \end{bmatrix} \tag{3}$$

where t_1 and $t_2 \in \mathrm{children}(t)$ are the two children clusters of t, and $\mathbf{T}^{t_1}$ and $\mathbf{T}^{t_2}$ are transfer matrices.

It is shown in [44] that the $\mathcal{H}^2$-based storage, matrix–vector multiplication, and matrix–matrix multiplication are all of linear complexity for constant-rank cases. However, no linear complexity inversion exists in mathematical literature.

B. Engineering View of the $\mathcal{H}^2$-Based Mathematical Framework

We view the $\mathcal{H}^2$-matrix as a mathematical framework rather than one specific method. Many existing fast solvers can be interpreted in this framework. This framework also provides a flexible infrastructure for continuous algorithm innovations to further accelerate the computation of dense matrices. For example, many fast solvers share the following in common, which is also one fundamental approach incorporated into the $\mathcal{H}^2$-based framework—the off-diagonal blocks in the system matrix, which characterize the far-field interaction between two separated groups of unknowns, are compactly stored and efficiently computed, whereas the near-field blocks keep their original representations and computations. As can be seen from (1), the blocks of far-field interactions are called admissible blocks in an $\mathcal{H}^2$-matrix. Depending on the specific solver used, the compact representation, or *data-sparse* representation, of the original dense matrix can be generated either algebraically [15], [16], [19], [22], [31] or in a kernel-dependent way [13], [17], [18], [20], [21], resulting in different rank, storage requirements, and time complexity. Most of the existing fast methods accelerate the matrix–vector multiplication used in an iterative solver. In contrast, we study how to invert or factorize a dense matrix fast by exploiting the nested low-rank representation inherent in the off-diagonal blocks.

C. Our Previous Work on Linear Complexity Direct Matrix Solutions

The linear complexity of an $\mathcal{H}^2$-based direct matrix inversion and LU-factorization does not exist in the mathematical literature prior to the work developed in [24], [26], and [27], in which we show that new algorithms can be developed to reduce the complexity of both $\mathcal{H}^2$-based inverse and LU factorization

to linear $O(N)$ complexity for electrically small and moderate problems. By the algorithms developed in [26, Sec. VI], each of the admissible, inadmissible, and nonleaf blocks in the inverse can be computed through a constant number of multiplications, and each multiplication has an $O(k^3)$ complexity, where k is the rank. As a result, we have $O(N)$ complexity for inverse. As compared to an LU factorization based direct matrix solution, a single inverse provides the solutions for N right-hand sides comprising an identity matrix. For different right-hand sides of interest, after the inverse is obtained, a single $O(N)$ $\mathcal{H}^2$-based matrix–matrix multiplication gives the solution for all right-hand sides, and the solutions for all these right-hand sides can also be compactly stored in $O(N)$ units.

D. On the Rank of Integral Operators

For electrically small problems, a constant rank is sufficient to achieve a desired order of accuracy irrespective of the problem size in representing the original dense matrix, as well as its inverse. For electrically moderate problems, a rank function is used to increase rank in a finite electric size range without compromising the linear complexity of the LU factorization and solution, as shown in [27]. For electrically large problems, it is proven in [45] that for any prescribed accuracy, the minimal asymptotic rank, for capturing the interaction between two separated geometry blocks in an integral operator, is a constant for 1-D distributions of source and observation points, grows very slowly with electrical size as square root of the logarithm for 2-D distributions, and scales linearly with the electrical size of the block diameter for 3-D distributions. Since the number of unknowns in a surface- and volume-IE based analysis scales with electrical size in a quadratic, and cubic way, respectively, the existence of an error-bounded low-rank representation is proved for both surface- and volume-based integral operators for electromagnetic analysis, irrespective of the electric size and object shape.

Certainly, due to the rank's growth with electric size, the same linear complexity algorithm developed for constant-rank cases would not achieve the same linear complexity for analyzing electrically large problems. However, the methods in an $\mathcal{H}^2$-matrix framework should not be stereotyped to be static approaches. In fact, the FMM developed for electrodynamic problems can also be viewed in the $\mathcal{H}^2$-based mathematical framework because the matrix structure resulting from an FMM based algorithm is the same nested structure as that in an $\mathcal{H}^2$-matrix, although the asymptotic rank of an FMM based representation is full rank. This full asymptotic rank in FMM is due to a source–observer separated nonminimal rank representation of the Green's function. The work in [45] provides a theoretical basis for developing more advanced algorithms to exploit the nested low-rank structure of the IE operators, thereby further reducing the complexity of electrically large computations. A recent demonstration can be seen from [46]. However, the focus of this paper is the integrated circuit extraction problem with existing circuit operating frequencies. Such a problem is electrically small, which permits a constant-rank representation for achieving prescribed accuracy.

E. Problem Description

The fast direct solutions we have developed in the past are for surface IE based solvers. The problem considered in this work is how to develop a linear complexity direct matrix solution for the VIE based circuit-parameter extraction of arbitrarily shaped 3-D lossy conductors embedded in inhomogeneous dielectric materials. Compared to a surface integral operator, a volume IE is structurally more complicated and computationally more expensive [47]. Furthermore, for circuit extraction problem, the VIE based system matrix is ill conditioned, which will become clear in Section III-A. Without a well-conditioned numerical system, it is difficult to achieve controlled accuracy in a numerical inverse whose performance thrives on approximate representations. We thus propose a well-conditioned formulation to address this problem. Though well-conditioned, the resultant system matrix is mixed with both dense and sparse matrix blocks that can also be either square or rectangular. The entire system matrix cannot be represented as a single $\mathcal{H}^2$-matrix. The existing $\mathcal{H}^2$-based algorithms are not directly applicable to the rectangular matrices encountered in the VIE system for matrix–matrix and matrix–vector multiplications. These challenges have been overcome by new algorithms developed in this work.

It is also worth mentioning that although generally speaking a VIE is an IE of the second kind, but the key difference here is that in the presence of *conductors*, the *diagonal-type* total-field contribution becomes negligibly small as compared to the other terms. In essence, for problems involving conductors, the VIE also approaches the IEs of first kind.

III. Proposed Well-Conditioned Full-Wave VIE Formulation

A. Brief Review of the VIE Formulation in [12] for Circuit Analysis

Consider a general 3-D circuit with arbitrarily shaped conductors of finite conductivity embedded in inhomogeneous dielectrics that can be lossless, lossy, and dispersive, as depicted in Fig. 1. Both conductors and dielectrics are characterized by space-dependent complex permittivity $\bar{\epsilon}(\vec{r}) = (\epsilon_r(\vec{r})\epsilon_0 - j\sigma(\vec{r})/\omega)$, with ϵ_r being dielectric constant, ϵ_0 being free-space permittivity, σ being conductivity, ω being angular frequency, and $\vec{r}$ denoting an arbitrary point in a 3-D space. The background material is assumed to be free space having permittivity ϵ_0. It can also be another uniform material. For the extraction of circuit parameters such as impedance $(\mathbf{Z})$- and scattering $(\mathbf{S})$-parameters at ports located anywhere in a circuit, we developed a potential-based source model in [12] to overcome the limitations of a commonly used delta-gap voltage source model. With the potential-based source model, a port can be flexibly assigned at an arbitrary point, either internal or external to the circuit. In contrast, in a delta-gap voltage source model, two conductors are needed to set up a port, the gap between the two conductors is required to be small, and the gap needs to be filled by a perfect conductor, each of which imposes a limitation in port assignments.

Now consider an arbitrary 3-D circuit attached to a potential source. In such a setting, the total potential ϕ is known at the

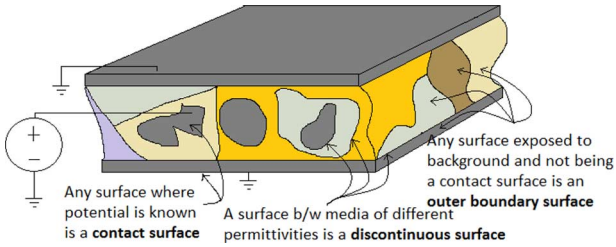

Fig. 1. Generic problem setup and associated terminology.

contact points $\vec{r}_c$ where the potential source is attached. The total field $\vec{E}(\vec{r})$ at any point $\vec{r}$ in space is given by

$$\vec{E}(\vec{r}) = -\nabla\phi(\vec{r}) - j\omega\vec{A}(\vec{r}) \tag{4}$$

subject to

$$\phi(\vec{r}_c) = \Psi_c \tag{5}$$

where Ψ_c is the applied potential. Equation (4) can further be written as

$$\frac{\vec{D}(\vec{r})}{\bar{\epsilon}(\vec{r})} - \mu\omega^2 \int_V \kappa(\vec{r'})\vec{D}(\vec{r'})g(\vec{r},\vec{r'})dv' + \nabla\phi(\vec{r}) = 0 \tag{6}$$

where Green's function $g(\vec{r},\vec{r'}) = e^{-jk_0|\vec{r}-\vec{r'}|}/4\pi|\vec{r}-\vec{r'}|$, k_0 is the free-space wavenumber, κ is the contrast ratio defined as

$$\kappa(\vec{r}) = \frac{\bar{\epsilon}(\vec{r}) - \epsilon_0}{\bar{\epsilon}(\vec{r})}$$

and $\vec{D}(\vec{r})$ is given by the relation

$$\vec{D}(\vec{r}) = \bar{\epsilon}(r)\vec{E}$$

which is also related to the equivalent volume current by $\vec{J}_e(\vec{r}) = j\omega\kappa(\vec{r})\vec{D}(\vec{r})$. The ϕ in (6) is generated by equivalent charges that include the equivalent volume charges (volume polarization charges) and the equivalent surface charges (surface polarization charges) at the material discontinuity where $\kappa(\vec{r})$ is discontinuous. As shown in [12], the density of the equivalent surface charges on the contact surfaces, ρ_{cs}, is unknown due to the source attachment, while the density of other surface charges $\rho_s|_{s\notin S_c}$ and the density of the equivalent volume charges ρ_v have a known relationship with $\vec{D}$ as follows:

$$\rho_v(\vec{r'}) = -\kappa(\vec{r'})\nabla'\cdot\vec{D}(\vec{r'}) \tag{7}$$

$$\rho_s|_{s\notin S_c}(\vec{r'}) = (\kappa^+(\vec{r'}) - \kappa^-(\vec{r'}))\vec{D}(\vec{r'})\cdot\hat{n} \tag{8}$$

in which S_c represents the contact surface where the potential source is attached, superscripts $-$ and $+$ are the indices of the two materials at a material discontinuity, and $\hat{n}$ denotes a unit vector normal to the material interface and pointing from material $+$ to material $-$. As a result, ρ_{cs} is the additional unknown to be solved together with $\vec{D}$ to analyze a circuit attached to a potential source. The work in [12] solves ρ_{cs} and $\vec{D}$ simultaneously to analyze circuits with an arbitrary port configuration. Its first-principles-based formulation also allows for a simultaneous circuit-scattering analysis.

Despite the full-wave rigor and the modeling flexibility provided by the VIE formulation in [12], the formulation is ill conditioned for analyzing realistic integrated circuits where conductors have finite conductivity and the operating frequency spans a wide range from zero to high frequencies. This is because, for relatively small electric sizes, the magnetic vector potential term becomes negligible as compared to the scalar potential term, and the scalar potential term is rank deficient [35]. Even though there is an additional term contributed by the total electric field in the VIE formulation, for large conductivity associated with metals, this term can also be overwhelmed by the scalar potential term. Although this is not a problem for a brute-force direct solution as long as the condition number is still within the machine precision, it can degrade the accuracy greatly for a fast solution thriving on error-controllable approximated numerical system and computations.

B. Proposed Well-Conditioned Full-Wave VIE Formulation

To build a well-conditioned VIE formulation, first, we explicitly enforce the following condition:

$$\nabla\cdot\vec{D}(\vec{r}) = 0 \quad \vec{r}\in V \tag{9}$$

which states that the divergence of $\vec{D}$ is zero in a source-free region. This condition is not explicitly enforced, but implicitly satisfied in the classical VIE formulations like [1], as well as the formulation in [12]. The vector basis functions employed to expand $\vec{D}$ such as the Schaubert–Wilton–Glisson (SWG) bases have a nonzero divergence too. Enforcing (9) has a direct implication on the equivalent volume charges in the system, making $\rho_v(\vec{r'}) = 0$ in (7) and reducing the expression for total potential to

$$\phi(\vec{r}) = \int_{S\notin S_c} \frac{(\kappa^+(\vec{r'}) - \kappa^-(\vec{r'}))\vec{D}\cdot\hat{n}}{\epsilon_0} g(\vec{r},\vec{r'})ds' + \int_{S_c} \frac{\rho_{cs}(\vec{r'})}{\epsilon_0} g(\vec{r},\vec{r'})ds' \tag{10}$$

such that

$$\phi(\vec{r}_c) = \Psi_c. \tag{11}$$

In other words, with the ρ_v vanished in each volume element filled by a single material, now the potential ϕ is only due to the equivalent surface charges. Apparently, ρ_{cs} in (10) seems to be an additional set of unknowns, but the $\phi(\vec{r}_c) = \Psi_c$ relation is utilized to eliminate it. As a result of explicitly enforcing (9), we avoid the nullspace arising from the divergence of $\vec{D}$ term in the VIE formulation. From an eigenvalue perspective, this nullspace has made the smallest eigenvalue modulus approach zero for the VIE system involving lossy conductors, thus being a major contributor to the ill-conditioning issue.

Secondly, we explicitly introduce ϕ as unknowns. This does not mean we introduce approximations since we will let ϕ satisfy the same ϕ-charge relationship as that satisfied in the original full-wave VIE formulations. To explain, in the original VIE formulations, the ϕ is directly represented in terms of $\vec{D}$ by using the relationship between ϕ and equivalent charges and thereby $\vec{D}$, as shown in (8). Here, instead of doing so directly, we introduce ϕ as an intermediate unknown, and write ϕ-charge (and hence, ϕ-$\vec{D}$), relationship as a separate equation to be simultaneously solved with (6). In this way, on one hand, we do

not lose the rigor of the original full-wave VIE formulation since the new formulation is mathematically equivalent to the original one. On the other hand, the new formulation provides an alternative way of solving the VIE system of equations, which is well conditioned. To be more specific, the Gaussian elimination can be performed to generate a well-conditioned matrix to solve ϕ first, and then solve $\vec{D}$. In this procedure, we avoid inverting a matrix mixed with both the magnetic vector potential term and the ϕ term, thus avoid suffering from the ill-conditioning of the VIE numerical system. The detailed procedure will become clear in Section IV.

To summarize, the proposed well-conditioned VIE formulation for circuits attached to a potential source Ψ_c is as follows:

$$\frac{\vec{D}(\vec{r})}{\bar{\epsilon}(\vec{r})} - \mu\omega^2 \int_V \kappa(\vec{r'})\vec{D}(\vec{r'})g(\vec{r},\vec{r'})dv' + \nabla\phi(\vec{r}) = 0 \tag{12}$$

$$\phi(\vec{r}_c) = \Psi_c \tag{13}$$

$$\nabla\cdot\vec{D}(\vec{r}) = 0 \; \vec{r}\in V \tag{14}$$

$$\phi(\vec{r}) = \int_{S\notin S_c} \frac{(\kappa^+(\vec{r'}) - \kappa^-(\vec{r'}))\vec{D}\cdot\hat{n}}{\epsilon_0} g(\vec{r},\vec{r'})ds' + \int_{S_c} \frac{\rho_{cs}(\vec{r'})}{\epsilon_0} g(\vec{r},\vec{r'})ds'. \tag{15}$$

The above formulation can also be readily modified to the case when the structure being simulated is excited by an external incident field $\vec{E}^{\rm inc}$ such as a plane wave. In this case, (13) does not exist, and the ρ_{cs} in (15), like $\rho_s|_{s\notin S_c}$, has a known relationship with $\vec{D}$. Hence, we solve the following equations simultaneously:

$$\frac{\vec{D}(\vec{r})}{\bar{\epsilon}(\vec{r})} - \mu\omega^2 \int_V \kappa(\vec{r'})\vec{D}(\vec{r'})g(\vec{r},\vec{r'})dv' + \nabla\phi(\vec{r}) = \vec{E}^{\rm inc}(\vec{r}) \tag{16}$$

$$\nabla\cdot\vec{D}(\vec{r}) = 0 \quad \vec{r}\in V \tag{17}$$

$$\phi(\vec{r}) = \int_S \frac{(\kappa^+(\vec{r'}) - \kappa^-(\vec{r'}))\vec{D}\cdot\hat{n}}{\epsilon_0} g(\vec{r},\vec{r'})ds'. \tag{18}$$

As can be seen from the above derivation, the proposed well-conditioned formulation is analytically equivalent to the original VIE formulation. Numerically, however, the new formulation can reduce the condition number of the underlying VIE system matrix to be inverted by many orders of magnitude, thus sustaining an accurate direct solution for circuit extraction with even a rank-1 representation.

IV. Discretization and System Matrix Construction

We discretize the computational domain into tetrahedral elements. In each tetrahedral element, the unknown electric displacement $\vec{D}(\vec{r})$ is expanded into SWG basis functions $\vec{f}_n(\vec{r})$ [1], the coefficient of which is denoted by D_n. The SWG basis functions inherently have nonzero divergence. This introduces nonphysical fictitious volume charges and a source of numerical cancellation errors. It is also worth pointing out that with (17) this numerical cancellation is also taken care of. The potential, ϕ is expanded by pulse basis functions.

A. System Matrix

We test (12) using Galerkin's method with vector basis $\vec{f}_m(\vec{r})$, while employing the Centroid Collocation method to test (13)–(15). When the $\nabla\phi(\vec{r})$ term in (12) is tested by vector basis $\vec{f}_m$, we obtain the following:

$$\int_V \vec{f}_m(\vec{r})\cdot\nabla\phi(\vec{r})dv = \int_{S\in ob} \phi\vec{f}_m(\vec{r})\cdot\hat{n}_m ds - \int_V \phi\nabla\cdot\vec{f}_m(\vec{r})dv \tag{19}$$

where ϕ in the surface integral resides on the outermost boundary of the computational domain, which is denoted by ϕ_{ob}; and the ϕ in the volume integral on the right-hand side of (19) is denoted by ϕ_v in the following formulations.

The system of linear equations obtained after discretization and testing can be written as

$$\begin{aligned} \mathbf{Z}[D_n] + \mathbf{V^1}[\phi_v] + \mathbf{V^2}[\phi_{ob}] &= K_1 \\ \mathbf{\Sigma}[D_n] + \mathbf{0}[\phi_v] + \mathbf{0}[\phi_{ob}] &= 0 \\ -\mathbf{P_0^{ob}}[D_n] + \mathbf{0}[\phi_v] + \mathbf{I}[\phi_{ob}] &= K_2 \end{aligned} \tag{20}$$

where $\mathbf{I}$ is an identity matrix, $\mathbf{Z}$ and $\mathbf{P_0^{ob}}$ are dense, whereas $\mathbf{V^1}$, $\mathbf{V^2}$, and $\mathbf{\Sigma}$ are sparse matrices. The first row of equations in (20) is the discretized and tested equivalent of (12), the second row is out of (14), whereas the last row is the discretized and tested version of (15) subject to (13).

The expressions of the matrix elements in $\mathbf{V^1}$ and $\mathbf{V^2}$ are given as

$$\begin{aligned} \mathbf{V}^1_{mn} &= -\int_{V_m} \nabla\cdot\vec{f}_m(\vec{r})dv \\ \mathbf{V}^2_{mn} &= \int_{S_m\in ob} (\vec{f}_m(\vec{r})\cdot\hat{n}_m)ds \end{aligned} \tag{21}$$

which are nonzero only in the nth columns, where n denotes the index of the tetrahedron or boundary patch where the $\vec{f}_m$ basis is located. The contribution from all contact-surface patches goes into the right-hand-side vector K_1

$$K_{1_m} = -\int_{S_m\in S_c} (\vec{f}_m(\vec{r})\cdot\hat{n})\Psi_c ds \tag{22}$$

since the potential on contact patches is known according to (13). The sparse matrix $\mathbf{\Sigma}$ has a dimension of $N_V \times N$, where N_V and N are the number of tetrahedrons, and the total number of SWG bases, respectively. The $\mathbf{\Sigma_{mn}}$ is nonzero only for those four bases that constitute the mth tetrahedron and given by

$$[\mathbf{\Sigma}]_{\mathbf{mn}} = \int_{V_m} \nabla\cdot\vec{f}_n(\vec{r'})dv' = \int_{S_m} \vec{f}_n(\vec{r'})\cdot\hat{n}ds' \tag{23}$$

which is the same as the transpose of $\mathbf{V^1}$.

The dense matrix $\mathbf{Z}$ is of dimension $N\times N$ and can be written as a sum of a sparse matrix and a dense matrix $\mathbf{A}$ as follows:

$$\mathbf{Z}_{mn} = \int_{V_m} \frac{\vec{f}_m(\vec{r})}{\bar{\epsilon}(\vec{r})}\cdot\vec{f}_n(\vec{r})dv - \mu\omega^2\mathbf{A}_{mn} \tag{24}$$

where

$$\mathbf{A}_{mn} = \int_{V_m} \int_{V_n} \kappa(\vec{r'}) \vec{f}_m(\vec{r}) \cdot \vec{f}_n(\vec{r'}) g(\vec{r}, \vec{r'}) dv' dv. \quad (25)$$

Since the dense nature of the $\mathbf{Z}$ matrix is imparted by the $\mathbf{A}$ matrix, in the remainder of this paper, the dense matrix $\mathbf{A}$ would not be referred to, but rather would be implied by $\mathbf{Z}$, and vice versa.

The $\mathbf{P_0^{ob}}$ matrix is of dimension $N_{ob} \times N$, where N_{ob} is the number of outermost boundary patches. It has the following form:

$$\mathbf{P_0^{ob}} = \mathbf{P_0}|_{\mathbf{ob,disc}} - \mathbf{P_0}|_{\mathbf{ob,cs}} \mathbf{P_0}|_{\mathbf{cs,cs}}{}^{-1} \mathbf{P_0}|_{\mathbf{cs,disc}} \quad (26)$$

where the subscripts "$\mathbf{ob}$," "$\mathbf{cs}$," and "$\mathbf{disc}$" denote the quantities on the outermost boundary, contact surfaces, and interfaces between discontinuous media, respectively. The $\mathbf{P_0}$ is the discretized equivalent of (15),

$$[\mathbf{P_0}|_{\mathbf{r_{cm}, S_n}}]_{\mathbf{mn}} = \frac{1}{\epsilon_0} \int_{S_n} \rho(\vec{r'}) g(\vec{r_{cm}}, \vec{r'}) ds' \quad (27)$$

in which the first subscript $\vec{r}_{cm}$ describes the centroid of the mth observation surface, and $\rho(\vec{r'})$ includes both $\rho_s|_{s \notin S_c}$ shown in (8) and contact-surface charges ρ_{cs}. The $\mathbf{P_0^{ob}}$ shown in (26) is the Schur complement corresponding to ϕ_{ob}. It is obtained by eliminating the contact-surface charges ρ_{cs}, based on (13), from the $\mathbf{P_0}$ based $\phi - \rho$ system of equations. Lastly, K_2 is the contribution from the contact surfaces given as

$$K_{2m} = \mathbf{P_0}|_{\mathbf{ob,cs}} \mathbf{P_0}|_{\mathbf{cs,cs}}{}^{-1} \Psi_c|_m. \quad (28)$$

For the incident field excitation case, K_1 is related to the incident field $\vec{E}^{\text{inc}}$, K_2 is 0, and $\mathbf{P_0^{ob}}$ is simply $\mathbf{P_0}|_{\mathbf{ob,disc}}$, while other matrix blocks remain the same.

B. Derivation of a Well-Conditioned Reduced System Matrix Suitable for $\mathcal{H}^2$-Matrix Representation

The system matrix as derived in Section IV-A involves both sparse and dense blocks, which, as a whole, cannot be represented in an $\mathcal{H}^2$-matrix format. We thus propose to reduce the original numerical system to the one involving ultimately only D_n as final unknowns by elimination. The elimination order, i.e., which unknown to eliminate first and which next, is not unique. To ensure a well-conditioned numerical system, we choose an order such that we can invert $\mathbf{Z}$ in as much an original form as possible, i.e., avoiding the inverse of a sum of the $\mathbf{Z}$ and the $\mathbf{P_0^{ob}}$ matrix, which is ill conditioned. This is because the $\mathbf{P_0^{ob}}$ term carries the contribution from the electric scalar potential, which can dominate $\mathbf{Z}$. This is especially true when lossy conductors, with finitely large conductivity, are considered since their presence diminishes the total electric field contribution. The detailed procedure is as follows.

Starting from the first equation of (20), we have

$$[D_n] = \mathbf{Z^{-1}}(K_1 - \mathbf{V^1}[\phi_v] - \mathbf{V^2}[\phi_{ob}]) \quad (29)$$

which is then substituted into the second row of (20). We obtain

$$\mathbf{\Sigma}[\mathbf{Z^{-1}}(K_1 - \mathbf{V^1}[\phi_v] - \mathbf{V^2}[\phi_{ob}])] = 0$$

which, in turn, gives us the expression of ϕ_v as

$$[\phi_v] = \mathbf{T}(K_1 - \mathbf{V^2}[\phi_{ob}]) \quad (30)$$

where

$$\mathbf{T} = [\mathbf{\Sigma Z^{-1} V^1}]^{-1} \mathbf{\Sigma Z^{-1}}.$$

This expression for ϕ_v is then substituted alongside the expression of D_n into the last row of (20), which helps to solve for ϕ_{ob},

$$[\phi_{ob}] = \mathbf{H_{full}^{-1}}(K_2 + \mathbf{F}K_1) \quad (31)$$

where

$$\mathbf{H_{full}} = [\mathbf{I} + \mathbf{F V^2}]$$
$$\mathbf{F} = \mathbf{P_0^{ob}} \mathbf{Z^{-1}}(\mathbf{I} - \mathbf{V^1 T}). \quad (32)$$

Now for the solution to the system given by (20), once (31) is solved for ϕ_{ob}, it is substituted into (30) to solve for ϕ_v, and finally the solution D_n is obtained from (29).

It can be seen from (29)–(31) that none involves an inversion of a matrix comprising contributions from both magnetic vector potential and electric scalar potential. In (29), we invert $\mathbf{Z}$, which only involves the magnetic vector potential matrix and the sparse matrix representing the total field. It is well conditioned, having a condition number orders of magnitude smaller than the original VIE system matrix that includes the scalar potential term. In (30), the computation of $\mathbf{T}$ involves the inverse of $\mathbf{T'} = [\mathbf{\Sigma Z^{-1} V^1}]$, which is well conditioned since the sparse matrix $\mathbf{\Sigma}$ and its transpose $\mathbf{V^1}$ are well conditioned as well as $\mathbf{Z}$. In (32), $\mathbf{P_0^{ob}}$ and thereby $\mathbf{F}$ are well-conditioned matrices. As a result, $\mathbf{H_{full}}$ is well conditioned. The identity matrix $\mathbf{I}$ is associated with the diagonal block, whose computation does not involve any approximation. Regardless of the dominance of the other term that is subtracted from or added upon the $\mathbf{I}$, its computation is exact, and hence, accurate up to machine precision. The above elimination order is essentially to solve ϕ first, and $\vec{D}$ next. For any other elimination order, we always run into the issue of inverting a matrix that has contributions from both magnetic vector and electric scalar potentials. The concurrent presence of these contributions results in an ill-conditioned matrix at relatively low frequencies, degrading the accuracy of the underlying numerical inverse.

The elimination process discussed above, involving matrix inversions and matrix–matrix multiplications, generally require cubic operations with respect to the number of unknowns involved. This work allows for the above elimination of the ϕ_v, ϕ_{ob}, and finally D_n unknowns to be performed in linear complexity. It means to say that both matrix inversions and matrix–matrix multiplications are performed in linear complexity, rendering the complexity of the entire solution linear. The detailed procedure is given in Section V.

V. Linear Complexity Direct Solution

We describe each step in the proposed direct solution in this section. First, we construct an $\mathcal{H}^2$-partition that is centered upon making the computation efficient. We will detail the techniques on how to tackle rectangular matrices present in (20). Based on

the proposed efficient $\mathcal{H}^2$-partition, we construct $\mathcal{H}^2$-representations of the dense matrices $\mathbf{Z}$ and $\mathbf{P_0^{ob}}$ in linear complexity with controlled accuracy. We also build $\mathcal{H}^2$-representations of the sparse matrices $\mathbf{\Sigma}$, $\mathbf{V^1}$, and $\mathbf{V^2}$, which are exact involving no approximation. Since some matrices in (20) do not have the same cluster basis, we unify the cluster bases to facilitate an efficient computation. We then update the $\mathcal{H}^2$-representations of the dense ($\mathbf{Z}$ and $\mathbf{P_0^{ob}}$) and sparse matrices ($\mathbf{\Sigma}$, $\mathbf{V^1}$ and $\mathbf{V^2}$) based on the new unified cluster basis. The linear complexity inverses of the multiple matrices involved in (29)–(31) are then performed to obtain the desired solution. The procedure of doing a matrix–matrix multiplication with rectangular matrices is also presented.

A. Efficient $\mathcal{H}^2$-Partition of VIE Matrices

An $\mathcal{H}^2$-based partition is to separate a matrix into two kinds of blocks: admissible blocks that have a factorized representation shown in (1) and inadmissible blocks that keep their original full-matrix form. The performance of the subsequent computational procedures relies heavily on this initial partitioning. Extra efforts need to be made to promote efficient computations.

We recursively split the 3-D computational domain that is composed of specific basis functions (e.g., the SWG basis functions for D_n, pulse basis functions for ϕ_v and ϕ_{ob}) into sub-domains until the number of unknowns in each sub-domain becomes less than or equal to the *leafsize* ($n_{\min}$). Typical values of $n_{\min}$ can be as small as 2 to as large as 100, and essentially $n_{\min}$ controls the depth of the tree. While doing the splitting operation, special care is taken to adaptively make sure that at each nonleaf tree level, every cluster has two children of similar size, we call this partitioning as a balanced partition. Such a splitting facilitates most efficient computational cases in the proceeding numerical procedures.

For the VIE based numerical system shown in (20), in total, we construct five different trees, namely, $T_{\rm SWG}$ for SWG bases, T_Σ for pulse bases representing ϕ_v, T_{ob} for pulse bases expanding ϕ_{ob}, T_{cs} for pulse bases representing ϕ on the contact surfaces denoted by ϕ_{cs}, and $T_{\rm disc}$ for pulse bases representing $\phi_{\rm disc}$, the ϕ at the material discontinuity. For computational efficiency of the matrix–matrix multiplication, discussed later in this paper, the structure of $T_{\rm disc}$ is kept the same as that of $T_{\rm SWG}$.

Now, we elaborate the VIE based construction of an admissible *block cluster tree* from the cluster trees $T_\mathcal{I}$ and $T_\mathcal{J}$ (in accordance with the admissibility condition). In a VIE based formulation, $\mathcal{I}$ is not necessarily the same as $\mathcal{J}$. Before we explain the general procedure adopted, we would like to point out that meanwhile the parameter "level" is numbered 0 from the root of the tree, the parameter "height" is numbered 0 from the leaf level. "Level-consistent partitioning" is one in which admissibility criterion is checked only for two clusters at the same level number, whereas for "height-consistent partitioning" it is done for two clusters at the same height. Due to the missing exploitation of the adaptive balanced splitting previously, only level-consistent $\mathcal{H}^2$-partitioning was relied upon. Though for square matrices, both level- and height-consistent partitioning is essentially the same, for rectangular matrices, performance-wise, height-consistent partitioning promotes generation of more admissible blocks than its counterpart. This is because for a general setting, height-consistency guarantees admissibility checking of geometrically and cardinally same sized blocks, thereby increasing the admissibility block count. Implementation-wise, however, $\mathcal{H}^2$-arithmetics demand novel algorithms to deal with rectangular cases, which are not covered in the open literature before and are shown in Table I.

TABLE I
HEIGHT-CONSISTENT RECTANGULAR MATRIX–MATRIX MULTIPLICATIONS

```
Procedure MultiplyBlock(A^{t,s}, B^{s,r}, C^{t,r})
  if height(t) = height(r)
    if #children(s) < 0
      if height(t) < height(s)
        ∀s' ∈ children(s)
          MultiplyBlock(A^{t,s'}, B^{s',r}, C^{t,r})
      else
        RecursiveMultiply(A^{t,s}, B^{s,r}, C^{t,r})
    else
      RecursiveMultiply(A^{t,s}, B^{s,r}, C^{t,r})
  else if height(t) > height(r)
    if #children(s) > 0
      if height(t) < height(s)
        ∀s' ∈ children(s)
          MultiplyBlock(A^{t,s'}, B^{s',r}, C^{t,r})
      else
        RecursiveMultiply(A^{t,s}, B^{s,r}, C^{t,r})
    else
      ∀t' ∈ children(t)
        MultiplyBlock(A^{t',s}, B^{s,r}, C^{t',r})
  else
    if #children(s) > 0
      if height(r) < height(s)
        ∀s' ∈ children(s)
          MultiplyBlock(A^{t,s'}, B^{s',r}, C^{t,r})
      else
        RecursiveMultiply(A^{t,s}, B^{s,r}, C^{t,r})
    else
      ∀r' ∈ children(r)
        MultiplyBlock(A^{t,s}, B^{s,r'}, C^{t,r'})
```

In this VIE solver, the cluster tree $T_{\rm SWG} = T_\mathcal{I} = T_\mathcal{J}$ is used to construct the matrix $\mathbf{Z}$, and the cluster tree $T_{cs} = T_\mathcal{I} = T_\mathcal{J}$ is used to construct the matrix $\mathbf{P_0^{ob}}|_{\mathbf{cs,cs}}$. As for the rectangular sparse matrices, $T_\mathcal{I} = T_{\rm SWG}$ and $T_\mathcal{J} = T_\Sigma$ are used for $\mathbf{V^1}$, and $T_\mathcal{I} = T_{\rm SWG}$ and $T_\mathcal{J} = T_{ob}$ are used for $\mathbf{V^2}$. For the dense rectangular matrices, $T_\mathcal{I} = T_{cs}$ and $T_\mathcal{J} = T_{\rm disc}$ are used to construct the matrix $\mathbf{P_0^{ob}}|_{\mathbf{cs,disc}}$, $T_\mathcal{I} = T_{ob}$ and $T_\mathcal{J} = T_{cs}$ are used to construct $\mathbf{P_0^{ob}}|_{\mathbf{ob,cs}}$, while $T_\mathcal{I} = T_{ob}$ and $T_\mathcal{J} = T_{\rm disc}$ are employed for $\mathbf{P_0^{ob}}|_{\mathbf{ob,disc}}$.

B. $\mathcal{H}^2$-Representation of Dense Matrices $\mathbf{Z}$ and $\mathbf{p_0^{ob}}$

To generate an $\mathcal{H}^2$-representation of the dense matrices $\mathbf{Z}$ and $\mathbf{P_0^{ob}}$, for an admissible block (t, s), we replace the underlying kernel function i.e., the Green's function $g(\vec{r}, \vec{r'})$ in (20) by a degenerate approximation based on the Lagrange interpolation method

$$\tilde{g}^{t,s} = \sum_{\nu \in K^t} \sum_{\mu \in K^s} g(\xi_\nu^t, \xi_\mu^s) L_\nu^t(\vec{r}) L_\mu^s(\vec{r'}) \tag{33}$$

where $K := \{\nu \in \mathbb{N}^3 : \nu_i \leq p, \forall i \in \{1,\ldots,3\}\} = \{1,\ldots,p\}^3$; p is the number of interpolation points along each dimension; $(\xi_\nu^t)_{\nu\in K^t}$ is a family of interpolation points in t; $(\xi_\mu^s)_{\mu\in K^s}$ is a family of interpolation points in s; and $(L_\nu^t)_{\nu\in K^t}$ and $(L_\mu^s)_{\mu\in K^s}$ are the corresponding Lagrange polynomials. The sparse matrix component of the $\mathbf{Z}$ matrix, due to the total field term, only appears in the inadmissible blocks, whereas the admissible blocks (t,s) after the degenerate approximation can be written as

$$\tilde{\mathbf{Z}}_{mn}^{\mathbf{t,s}} = \sum_{\nu\in K^t}\sum_{\mu\in K^s}\int_{V_m}\vec{f}_m(\vec{r})L_\nu^t(\vec{r})dv\cdot[-\mu\omega^2 g(\xi_\nu^t,\xi_\mu^s)]\cdot \int_{V_n}\kappa(\vec{r'})\vec{f}_n(\vec{r'})L_\mu^s(\vec{r'})dv'.$$

Similarly, the $\mathbf{P_0}|_{\mathbf{r_{cm}},\mathbf{S_n}}$ matrix after the degenerate approximation can be written as

$$\tilde{\mathbf{P}}_\mathbf{0}|_{mn} = \sum_{\nu\in K^t}\sum_{\mu\in K^s}L_\nu^t(\vec{r}_{cm})\frac{g(\xi_\nu^t,\xi_\mu^s)}{\epsilon_0}\int_{S_n}L_\mu^s(\vec{r'})ds' \quad (34)$$

from which we obtain the $\mathcal{H}^2$ representations of the dense matrices $\mathbf{Z}$ and $\mathbf{P_0^{ob}}$ written as

$$\begin{aligned}\tilde{\mathbf{Z}}^{\mathbf{t,s}} &= \mathbf{V}_{z1}^t\mathbf{S_z}^{t,s}\mathbf{V}_{z2}^{sT}\\ \tilde{\mathbf{P}}_\mathbf{0}^{\mathbf{t,s}} &= \mathbf{V}_{p1}^t\mathbf{S_p}^{t,s}\mathbf{V}_{p2}^{sT}\end{aligned} \quad (35)$$

where the cluster bases in a given cluster t are

$$\begin{aligned}\mathbf{V}_{z1}^t &= \int_{V_m}\vec{f}_m(\vec{r})L_\nu^t(\vec{r})dv\\ \mathbf{V}_{z2}^t &= \int_{V_m}\kappa(\vec{r})\vec{f}_m(\vec{r})L_\nu^t(\vec{r})dv\\ \mathbf{V}_{p1}^t &= L_\nu^t(\vec{r})\\ \mathbf{V}_{p2}^t &= \int_{S_m}L_\nu^t(\vec{r})ds.\end{aligned} \quad (36)$$

A close inspection of (36) reveals that although $\mathbf{V}_{p1}^t$ is set to fit in all the forms of row clusters ($\mathbf{P_0}|_{\mathbf{cs},\cdot}$ and $\mathbf{P_0}|_{\mathbf{ob},\cdot}$) for the matrices we need to represent, but $\mathbf{V}_{p2}^t$ is slightly complicated. For example, it can be directly applied to the case of representing the column cluster bases for contact surfaces in $\mathbf{P_0}|_{\cdot,\mathbf{cs}}$. But when it comes to the case of representing the column cluster bases for discontinuous surface patches in $\mathbf{P_0}|_{\cdot,\mathbf{disc}}$, in order to match dimensions for proceeding matrix–matrix multiplications, the column cluster bases dimension needs to be made the same as the number of SWG bases. Special care has to be taken to make sure that only those bases corresponding to the discontinuous surfaces are filled according to (36). This also calls upon the need for unification of SWG bases with this type of bases before doing the matrix–matrix multiplication. The details are discussed in Section V-D. The coupling matrices shown in (35) are given by

$$\begin{aligned}\mathbf{S_z}^{t,s}{}_{\nu\mu} &= -\mu\omega^2 g(\xi_\nu^t,\xi_\mu^s)\\ \mathbf{S_p}^{t,s}{}_{\nu\mu} &= \frac{g(\xi_\nu^t,\xi_\mu^s)}{\epsilon_0}.\end{aligned} \quad (37)$$

In (35)–(37), $\nu \in (1,K^t)$, $\mu \in (1,K^s)$, $\mathbf{V}^t \in \mathbb{R}^{\#t\times K^t}$, and $\mathbf{S}^{t,s} \in \mathbb{C}^{K^t\times K^s}$, and $K^{t(s)} \ll \#t(\#s)$ is the rank of the cluster basis for t or s, determined by the number of interpolation points in an interpolation based $\mathcal{H}^2$-representation. Since the same space of Lagrange polynomials is used, each of the cluster bases is a nested cluster basis. Another important observation that can be made regarding the cluster bases is that both row and column cluster bases in the above $\mathcal{H}^2$-representations are different. To make the computation efficient for such cases, where we have multiple cluster bases for the same cluster, we unify their contributions into one single cluster basis keeping the nested property intact.

C. $\mathcal{H}^2$-Representation of Sparse Matrices $\mathbf{\Sigma}$, $\mathbf{V^1}$, and $\mathbf{V^2}$

Not only the dense matrices are needed to be represented in an efficient format, but also the sparse matrices. This is because these matrices are also involved in the computation when we need to do matrix–matrix multiplications, for instance. Linear complexity $\mathcal{H}^2$-matrix arithmetic would only come handy if both the operands are represented in an $\mathcal{H}^2$-matrix format. Another important task is to form corresponding cluster bases for these sparse matrices. It becomes of pivotal importance when the target matrix (the result of a matrix–matrix multiplication) shares the same structure as the sparse matrix, but is no longer sparse, and thus needs a cluster basis for the representation of admissible (i.e., off-diagonal) blocks. In the following, we present the $\mathcal{H}^2$-matrix representation of each sparse matrix.

1) $\mathcal{H}^2$-Representation of $\mathbf{\Sigma}$ and $\mathbf{V^1}$: The row cluster of $\mathbf{\Sigma}$ is from tetrahedral pulse bases and the column cluster is from the SWG bases. We hence choose the column cluster bases to be the same as used for representing the $\mathbf{Z}$ matrix. As for the row cluster bases,

$$\mathbf{V}_\Sigma^t = \int_{V_m}L_\nu^t(\vec{r})dv. \quad (38)$$

Therefore, we choose T_Σ as its row tree and T_{SWG} as its column tree, and construct an $\mathcal{H}^2$ partition. We then fill the matrix entries by keeping in view that $\mathbf{\Sigma_{mn}}$ is nonzero only for those four bases/patches that constitute the mth tetrahedron, and all other $N-4$ entries are zeros. Since for any tetrahedral element all of its surface patches can never satisfy the admissibility condition, all the admissible blocks are zeros. The presence of nonzero cluster bases needs the coupling matrices to be made zero for all the admissible blocks. For the full matrix blocks, only a few of the entries are nonzero. The sparse matrix $\mathbf{V^1}$ is simply the transpose of $\mathbf{\Sigma}$.

2) $\mathcal{H}^2$-Representation of $\mathbf{V^2}$: The row cluster of $\mathbf{V^2}$ is built by the SWG bases and the column cluster is from the outer boundary surface bases. We thus choose the column cluster bases to be the same as (36), while for the row cluster bases, we re-utilize our unified cluster bases used for representing the $\mathbf{Z}$ matrix. Like sparse matrices $\mathbf{\Sigma}$ and $\mathbf{V^1}$, all the admissible blocks are zero, while inadmissible blocks are filled with the nonzero elements of $\mathbf{V^2}$.

D. Unify and Orthogonalize Cluster Bases

The construction of an $\mathcal{H}^2$-representation of each matrix involved in (20) is done based on the matrix's expression. This can lead to different row and column cluster bases (as in the case of $\mathbf{Z}$ and $\mathbf{P_0^{ob}}$), a mismatch in the type of cluster bases while doing

matrix–matrix multiplications (as in the case of $\mathbf{P_0^{ob}Z^{-1}}$), and the cluster bases by themselves can be vector based, thus affecting efficiency. To address these issues, we propose to do an orthogonalization based unification of the cluster bases so as to get a unified cluster basis, e.g., $\mathbf{V}$ with $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ satisfied for each cluster. A close inspection of the cluster bases constructed in previous sections reveal that only those used to construct $\mathbf{Z}$ and $\mathbf{P_0^{ob}}$ need to be unified and orthogonalized while all the rest are just needed to be orthogonalized using [44]. We explain this by taking the example of the cluster bases used in the construction of $\mathbf{Z}$. In order to overcome the cluster basis mismatch that can occur while doing the multiplication of the type $\mathbf{P_0^{ob}} \times \mathbf{Z}^{-1}$, we introduce a new cluster basis

$$\mathbf{V}_{pz}^t = \int_{S_m} L_\nu^t(\vec{r})ds \qquad \forall S_m.$$

Given a cluster t, we then assemble all the cluster bases associated with it as

$$\mathbf{V}_{Z_{\text{expand}}}^t = [\mathbf{V_{z1x}^t V_{z1y}^t V_{z1z}^t V_{z2x}^t V_{z2y}^t V_{z2z}^t V_{pz}^t}]$$

where $\mathbf{V_{z1x}}$ denotes the x-component of basis $\mathbf{V_{z1}}$ shown in (36), and other subscripts can be understood similarly. We then orthogonalize $\mathbf{V}_{Z_{\text{expand}}}^t$, which results in the orthogonal cluster basis $\tilde{\mathbf{V}}_Z^t$ that contains all the linearly independent columns of the above unification. All this is done in linear complexity using [44]. This orthogonalization should be proceeded in a one-cluster-basis-at-a-time fashion if cluster bases of different norms are to be unified.

Now based on this unified and orthogonal cluster basis $\tilde{\mathbf{V}}_Z^{\mathbf{t}}$, we update the matrix $\mathbf{Z}$ as follows:

$$\begin{aligned}\tilde{\mathbf{Z}}^{\mathbf{t,s}} &= \tilde{\mathbf{V}}_Z^{\mathbf{t}}\tilde{\mathbf{V}}_Z^{\mathbf{tT}}\mathbf{V}_{z1}{}^t\mathbf{S}_{\mathbf{z}}^{t,s}\mathbf{V}_{z2}{}^{sT}\tilde{\mathbf{V}}_Z^{\mathbf{s}}\tilde{\mathbf{V}}_Z^{\mathbf{sT}} \\ &= \tilde{\mathbf{V}}_Z^{\mathbf{t}}[\tilde{\mathbf{V}}_Z^{\mathbf{tT}}\mathbf{V}_{z1}{}^t]\mathbf{S}_{\mathbf{z}}^{t,s}[\mathbf{V}_{z2}{}^{sT}\tilde{\mathbf{V}}_Z^{\mathbf{s}}]\tilde{\mathbf{V}}_Z^{\mathbf{sT}}. \end{aligned} \tag{39}$$

It is worth appreciating the fact that, due to the nested property, the cluster bases product given by square bracketed terms can be calculated in $O(k^3)$ operations where k is the rank of the cluster bases. Since k is a constant for circuit applications, overall, the update procedure also takes $O(N)$ computational resources. The row cluster basis $\mathbf{V}_{p1}^t$ and the column cluster basis $\mathbf{V}_{p2}^t$ of matrix $\mathbf{P_0^{ob}}$ also needs to be unified and orthogonalized to obtain $\tilde{\mathbf{V}}_p^t$. With $\tilde{\mathbf{V}}_p^t$, the $\mathcal{H}^2$ representation of $\mathbf{P_0^{ob}}$ can also be updated accurately.

E. Linear Complexity $\mathcal{H}^2$-Matrix Based Direct VIE Inverse

We now obtain an $\mathcal{H}^2$-matrix for each of the matrices involved in the solution of the system given by (20). There are in total four inverses involved to arrive at the final solution. For each of these inverses, we perform an $O(N)$ $\mathcal{H}^2$-inverse algorithm developed in [26].

The main operation in the inversion is to perform block matrix multiplications. There are a total of 27 cases that an $\mathcal{H}^2$-arithmetics based block matrix multiplication can have. Since we have three types of $\mathcal{H}^2$-matrix blocks (namely, admissible, inadmissible, and nonleaf blocks), these combinatorially possible 27 multiplication cases do practically occur if no special care is taken. Not only do these include extra implementation cases, but also lead to computationally less efficient cases and projection based approximations. A quick derivation reflects that the balanced partitioning of Section V-A can shrink these cases to 13. For example, with this partition, it is not feasible to have an admissible block getting multiplied by an inadmissible block with the target being a nonleaf block. By allowing inadmissible blocks at the leaf-level only, we further bring the 13 cases down to the most efficient seven cases. These seven cases are: 1) admissible $\times$ admissible $=$ admissible; 2) admissible $\times$ nonleaf $=$ admissible; 3) nonleaf $\times$ nonleaf $=$ admissible; 4) inadmissible $\times$ inadmissible $=$ inadmissible; 5) admissible $\times$ nonleaf $=$ nonleaf; 6) nonleaf $\times$ nonleaf $=$ nonleaf; and 7) admissible $\times$ admissible $=$ nonleaf. Cases 1) and 4)–7) are done without any approximation, while Cases 2) and 3) involve a collect operation, the accuracy of which is controlled by the rank of cluster bases. These seven cases of multiplications also benefit the matrix–matrix multiplications involved in the proposed direct solver.

F. Updated Matrix–Vector and Matrix–Matrix Multiplication

For a single excitation, the right-hand sides K_1 and K_2 are vectors. When doing the final evaluations to find ϕ_v, ϕ_{ob} and finally D_n, one can use an $O(N)$ $\mathcal{H}^2$ based matrix–vector multiplication. However, special care needs to be taken to accommodate all possible cases involving the rectangular $\mathcal{H}^2$-matrices. For general $\mathcal{H}^2$-rectangular matrices, the difference is in recursively going down the $\mathcal{H}^2$ block cluster tree until the level consistency is approached. For multiple-port excitations, the right-hand sides K_1 and K_2 take the form of matrices. These matrices due to their inherent data-sparse nature can be cast into $\mathcal{H}^2$-matrices. In addition, during the elimination procedure, we have a number of matrix–matrix multiplications to perform. While performing these multiplications, we need to take care of the general cases one might confront while dealing with general height-consistent rectangular $\mathcal{H}^2$ matrices. As mentioned before, while height-consistent $\mathcal{H}^2$ matrices offer better performance; from the implementation point of view, novel algorithms need to be developed, as the open literature relies only on level-consistent $\mathcal{H}^2$ matrices. A pseudo-code is given in Table I, summarizing all the possible scenarios alongside the basic case (RecursiveMultiply $(\mathbf{A^{t,s}}, \mathbf{B^{s,r}}, \mathbf{C^{t,r}})$) of the $\mathcal{H}^2$-matrix multiplications in [44].

VI. Complexity and Accuracy Analysis

A. Complexity Analysis

The storage of all the matrices involved is done in the $\mathcal{H}^2$-matrix format, which has $O(N)$ complexity. The total computational cost for solving the VIE based system (20) includes the following three parts.

1) The $\mathcal{H}^2$ based construction of the matrices, the complexity of which has already been analyzed to be linear.
2) Direct $\mathcal{H}^2$-matrix based recursive inverses of four matrices, each of which is performed in linear complexity.
3) Matrix–vector and matrix–matrix multiplications involved in the elimination process, all of which have a linear complexity as shown in [28].

As a result, the total complexity of the proposed solver is linear.

B. Accuracy Analysis

The accuracy of the proposed direct solver in each step is analyzed as follows.

The $\mathcal{H}^2$-representation of dense matrices $\mathbf{Z}$ and $\mathbf{P_0^{ob}}$ is obtained from (33). It is shown in [26] that the error of (33) can be controlled to any desired order by the number of interpolation points p and parameter η used for the $\mathcal{H}^2$-partition. For full-wave circuit extraction concerned in this work, the electric size of the underlying problem is not large, and hence, a very small p is sufficient for achieving good accuracy. The $\mathcal{H}^2$-representation of $\mathbf{\Sigma}$, $\mathbf{V^1}$, and $\mathbf{V^2}$ is exact, as they are sparse matrices.

Unification of the cluster bases is achieved by orthogonalization, during which the rank of the cluster basis is not reduced, and hence, no additional error is introduced.

To compute $\mathbf{P_0^{ob}}$, we first obtain $\mathbf{P_{cs,cs}^{-1}}$ by the inverse algorithm developed in [26]. The next steps involve pre-multiplication by $\mathbf{P_0}|_{\mathbf{ob,cs}}$ and post multiplication by $\mathbf{P_0}|_{\mathbf{cs,disc}}$. These matrix–matrix multiplications are error-controlled to the same accuracy as used in representing them [44]. Next we add $\mathbf{P_0}|_{\mathbf{ob,disc}}$ to obtain the desired $\mathbf{P_0^{ob}}$. The addition does not introduce new errors since both the operands and the target share the same cluster bases and $\mathcal{H}^2$-partition.

Similarly, we approach at finding the inverse of the $\mathbf{Z}$ matrix. The $\mathbf{Z}$ to be inverted in this VIE solver only consists of the total field term and the magnetic vector potential term, as shown in (24). It can be compactly written as

$$\mathbf{Z} = \mathbf{D} - \omega^2 \mathbf{A}' \tag{40}$$

where $\mathbf{D}$ denotes the matrix associated with the total field term. The solution of $\mathbf{Z}$ is governed by the eigenvectors of the following generalized eigenvalue problem [35]

$$\mathbf{D}v = \lambda \mathbf{A}' v \tag{41}$$

where v denotes an eigenvector, and λ denotes an eigenvalue. To solve

$$\mathbf{Z}x = b \tag{42}$$

with b being an arbitrary right-hand side, we can expand x in the space of $\mathbf{V}$, the column space formed by all the eigenvectors. Thus, $x = \mathbf{V}y$, where y is the unknown coefficient vector. Substituting $x = \mathbf{V}y$ into (42), and multiplying both sides by $\mathbf{V}^T$, we obtain

$$(\mathbf{V}^T\mathbf{D}\mathbf{V} - \omega^2 \mathbf{V}^T\mathbf{A}'\mathbf{V})y = \mathbf{V}^T b. \tag{43}$$

From (41), we have $\mathbf{V}^T\mathbf{D}\mathbf{V} = \mathbf{V}^T\mathbf{A}'\mathbf{V}\mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is the diagonal matrix containing all the eigenvalues. Using this fact, (43) becomes

$$(\mathbf{V}^T\mathbf{A}'\mathbf{V})(\mathbf{\Lambda} - \omega^2)y = \mathbf{V}^T b \tag{44}$$

from which we obtain

$$y = \frac{1}{\mathbf{\Lambda} - \omega^2}(\mathbf{V}^T\mathbf{D}\mathbf{V})^{-1}\mathbf{V}^T b. \tag{45}$$

Therefore, the solution of (42) is

$$x = \mathbf{V}y = \mathbf{V}\frac{1}{\mathbf{\Lambda} - \omega^2}(\mathbf{V}^T\mathbf{A}'\mathbf{V})^{-1}\mathbf{V}^T b. \tag{46}$$

When b in (42) is an identity matrix, we obtain the inverse of $\mathbf{Z}$ as the following:

$$\mathbf{Z}^{-1} = \mathbf{V}\frac{1}{\mathbf{\Lambda} - \omega^2}(\mathbf{V}^T\mathbf{A}'\mathbf{V})^{-1}\mathbf{V}^T \tag{47}$$

which can further be compactly written as $\mathbf{VSW}$. As can be seen, the weight of the ith mode V_i in the inverse is proportional to $1/(\lambda_i - \omega^2)$. For any frequency ω, only those eigenvectors whose eigenvalues are close to ω^2 have a large weight, while those that are far away from the given frequency can be truncated based on a prescribed accuracy. Therefore, $\mathbf{Z}$'s inverse has a low-rank representation. For electrically small cases, i.e., low-frequency cases, the rank k is a bounded constant for achieving any prescribed accuracy. Now consider an arbitrary off-diagonal block of the inverse formed by arbitrarily selected m rows and n columns. From (47), this block can be written as $\mathbf{V}_{m\times k}\mathbf{S}_{k\times k}\mathbf{W}_{k\times n}$, where $\mathbf{V}_{m\times k}$ is the matrix containing the m rows of $\mathbf{V}$, and $\mathbf{W}_{k\times n}$ contains the n columns of $\mathbf{W}$. As a result, it is low rank, and hence, the $\mathcal{H}^2$-representation of the inverse of $\mathbf{Z}$ exists, and we can apply the algorithm in [26] to obtain $\mathbf{Z^{-1}}$.

Next, we evaluate the matrix $\mathbf{T}$, which needs an inverse of the matrix $\mathbf{T}' = [\mathbf{\Sigma}\mathbf{Z^{-1}}\mathbf{V^1}]$ to be found out. Since there exists an $\mathcal{H}^2$-representation of $\mathbf{Z^{-1}}$, $\mathbf{\Sigma}$, and $\mathbf{V^1}$, respectively, the product of the three matrices also has an error-controlled $\mathcal{H}^2$-matrix representation based on which an $\mathcal{H}^2$-inverse can be performed.

The last major step is to find an inverse of the matrix $\mathbf{H_{full}}$. It is worth noting that the computation of $\mathbf{H_{full}}$ involves sparse operations and dense matrix–matrix multiplications. All these operations are performed accurately. The associated matrix–vector operation required to evaluate $K_2 + \mathbf{F}K_1$ and all other elimination steps are again accurately performed [26].

VII. Numerical Results

The first two of the four examples are presented in order to validate the proposed VIE direct solver and its formulation. The last two examples, which include a suite of large-scale on-chip bus structures (involving over 1 million D_n unknowns) and large-scale irregularly shaped spiral inductor arrays (involving over 2 million D_n unknowns) demonstrate the performance benefits that can be achieved with the proposed direct solver. Both these large-scale examples show a clear linear scaling of memory and time utilization for inverting and solving the proposed VIE system matrix.

A. Impedance Extraction of a 1 × 5 On-Chip Bus

The first example is a 1 × 5 bus structure having typical on-chip dimensions in a single material (air). As illustrated in Fig. 2, each bus has a dimension of 2 μm × 2 μm × 20 μm, while the spacing between two adjacent buses is 5 μm. The metal conductivity is 5.8×10^7 S/m, and the frequency for extraction is 10 GHz resulting in $\bar{\epsilon}(\vec{r}) = \epsilon_0(1.0 - j6.55 \times 10^8)$. The discretization results in 6000 tetrahedrons. The rank used in the $\mathcal{H}^2$-representation is 1 obtained from an interpolation based method. The impedance parameter matrix extracted from the

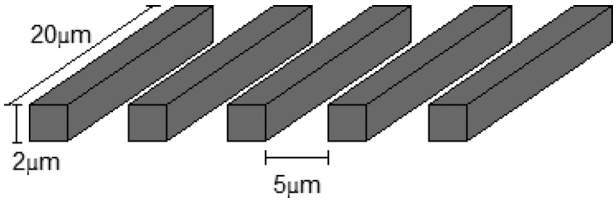

Fig. 2. Geometrical details of a 1 × 5 bus structure.

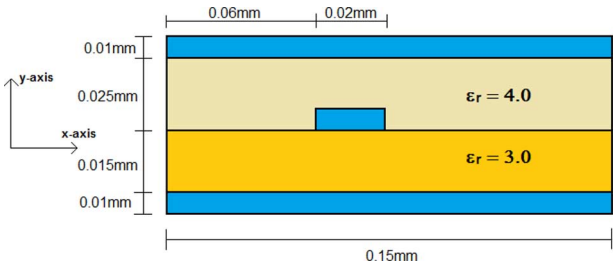

Fig. 3. Cross-sectional view of a multiple-dielectric package interconnect.

proposed well-conditioned VIE formulation is given as follows at the bottom of this page. All impedances are in 10^{-2} Ω. The relative difference of the above impedance matrix is shown to be 0.29%, computed from $\|\mathbf{Z} - \mathbf{Z}_{\rm FH}\|_F / \|\mathbf{Z}_{\rm FH}\|_F$, where $\mathbf{Z}_{\rm FH}$ is obtained from FastHenry [3] with 12 000 filaments, $\mathbf{Z}$ is from the proposed full-wave VIE direct solver, and Frobenius norm is used. FastHenry is a fast-multipole accelerated iterative IE solver, which performs a dense matrix–vector multiplication in linear complexity. In contrast, the proposed direct solver computes a dense matrix inverse in linear complexity. The solution time for the proposed direct solver is 84.8 s as compared to 458.3 s taken by FastHenry [28].

We also used this example to examine the condition number of the traditional VIE system matrix as compared to the proposed one. For the 1 × 5 bus structure, the condition number of the original VIE matrix including the scalar potential contribution is 2.863748×10^7, whereas the condition number of the proposed one is 3.775967, a reduction of seven orders of magnitude.

B. S-Parameter Extraction of a Package Interconnect With Multiple Dielectrics

A 3-D package interconnect with multiple dielectrics, whose detailed geometrical and material data are shown in Fig. 3, is extracted in a broad band of frequencies from 1 to 30 GHz. The length of the structure is 1 cm. The conductivity of the metal is 5.8×10^7 S/m.

In Fig. 4, we plot the S-parameters extracted from the proposed well-conditioned VIE direct solver in comparison with the reference data provided by the Intel Corporation. Good agreement is observed in the entire frequency band for all S-parameters.

C. Large-Scale On-chip Bus Structures

A suite of large-scale on-chip bus structures from a 4 × 4 array to a 64 × 64 array is then simulated at 10 GHz. The simulated scenario for a 16 × 16 array is illustrated in Fig. 5. The dimensions of each bus are 2 μm × 2 μm × 20 μm. The distance between the centers of two neighboring buses on the same horizontal axis is 20 μm, while on the vertical axis it is 40 μm. The conductivity of the metal is 5.8×10^7 S/m. Each of the bus element is discretized into 248 D_n unknowns.

The cluster trees $T_{\rm SWG}$, T_Σ, T_{ob}, T_{cs}, and $T_{\rm disc}$ are built with leafsizes 80, 40, 30, 10, and 80, respectively. For the $\mathcal{H}^2$-partition, the admissibility parameter is set as $\eta = 0.55$, and for computational efficiency, once the leaf level is reached the admissibility is not checked and the blocks are allowed to remain as full matrices. The performance of the proposed solver is demonstrated by simulating arrays of 4 × 4 to 64 × 64 resulting in D_n unknowns from 3968 to 1 015 808, where at each simulation point the number of buses along each dimension is doubled. The total number of unknowns including both D_n and ϕ unknowns ranges from 6528 to 1 671 168. The percentage of the admissible blocks, calculated as the ratio of the number of matrix elements in the admissible blocks to the total N^2 matrix elements, ranges from 75% to 99% in this example. The performance results obtained from these simulations are summarized in Fig. 5, where the horizontal axis denotes the number of D_n, and hence, equivalent current unknowns, N. We can see that the error of representing the $\mathbf{Z}$ matrix of the VIE formulations remains well controlled even with a single point interpolation scheme, thus rank-1 representation. The relative inverse error with respect to identity, for all four matrices involved in the VIE system matrix, is also well controlled in and below the 10^{-2} range. The relative inverse error for the final matrix $\mathbf{H}_{\rm full}$ is shown to be in the 10^{-9} range. The CPU cost of the most expensive operation, i.e., the inversion of the $\mathbf{Z}$ matrix, the total solution time, and the memory cost are shown in Fig. 5, each of which exhibits a clear linear scaling with respect to the number of unknowns.

We also simulated the same bus structure of 1 015 808 unknowns when the horizontal spacing between adjacent buses is enlarged to 40 μm. Since more admissible blocks are involved in this case, we are able to finish the inversion of $\mathbf{Z}$ of 1 015 808 unknowns in 4 min using less than 5-GB memory.

$$\begin{bmatrix} 10.128+71.349i & 0.105+26.699i & -0.052+16.037i & -0.034+11.322i & -0.019+8.706i \\ 0.105+26.699i & 10.258+71.229i & 0.153+26.657i & -0.032+16.019i & -0.034+11.322i \\ -0.052+16.037i & 0.153+26.657i & 10.272+71.217i & 0.153+26.657i & -0.052+16.037i \\ -0.034+11.322i & -0.032+16.019i & 0.153+26.657i & 10.258+71.229i & 0.105+26.699i \\ -0.019+8.706i & -0.034+11.322i & -0.052+16.037i & 0.105+26.699i & 10.128+71.349i \end{bmatrix}$$

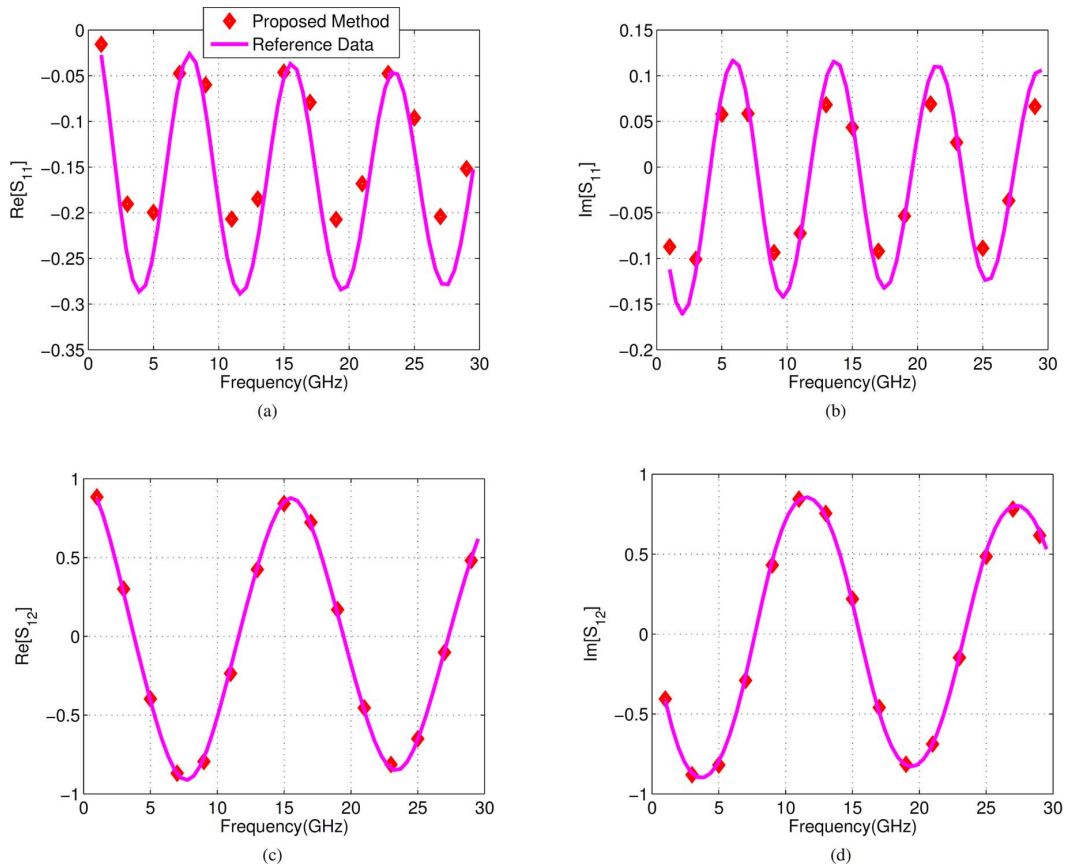

Fig. 4. S-parameters of a 3-D package interconnect with nonuniform materials simulated by the proposed well-conditioned VIE solver in comparison with reference data. (a) $\mathrm{Re}[S_{11}]$. (b) $\mathrm{Im}[S_{11}]$. (c) $\mathrm{Re}[S_{12}]$. (d) $\mathrm{Im}[S_{12}]$.

In addition, we simulated the same arrays of 4×4 to 64×64 buses, but with a smaller spacing of 10 μm between the adjacent buses. For $\mathcal{H}^2$-partitioning, the admissibility parameter of 1.5, leaf-size of 80 for SWG bases, and the interpolation scheme to achieve an accuracy of 10^{-4} for both $\mathbf{Z}$ and $\mathbf{P_0}$ is chosen. The resultant average rank is 11 for all the admissible blocks. In Fig. 6(a), we plot the matrix representation errors and the inverse errors as a function of D_n unknowns. We can see that both errors are small. It is worth pointing out that as compared to the previously simulated examples with larger inter-element spacing, the inter-element spacing has reduced significantly in this example, resulting in much stronger coupling in the off-diagonal blocks not only in the original matrices, but also in the inverse matrices. As a result, the rank required to achieve a good accuracy has increased. From a practical stand-point of extracted parameters, thresholds of 10^{-2} produce accurate results. For better accuracy, it is obvious that one needs to increase the rank of the initial representation.

The CPU cost of the most expensive operations, i.e., the inversion of the $\mathbf{Z}$ matrix, the total solution time, and the memory cost are shown in Fig. 6(b) in comparison with that of the previously simulated example having inter-element spacing of 20 μm along horizontal and 40 μm along the vertical direction. The solid lines for each of the three plots are for the case of 10-μm spacing while the dotted lines are for the 20-μm case. It is evident that each of the plot again exhibits a clear linear scaling with respect to the number of unknowns and that for lesser admissible blocks, i.e., for the 10-μm spacing case, the total cost is larger than that of the previously simulated example. It is also worth mentioning here that this would not be an issue if we have a procedure that allows the same level of admissible blocks as in the previous case. The increase in the interpolation rank can then be compressed to achieve performance benefits, which can be seen from [48].

D. Large-Scale Arrays of Irregularly Shaped Spiral Inductors

A suite of large-scale arrays of irregularly shaped inductors is then simulated at 10 GHz. This large-scale structure is simulated to demonstrate the flexibility of the proposed VIE solver in geometrical modeling while keeping the same order of performance irrespective of the shape of the structures being simulated. The geometrical details of each of the array element are illustrated in Fig. 7. The outer radius of each inductor is 10 μm. The thickness of the metallic wire, the via height Δv, and the port length Δt are all 1 μm. The conductivity of the metal is 5.8×10^7 S/m. Port 1 is located from $\tilde{\phi} = 0$ to $\tilde{\phi} = 20^o$ at the lower layer, where $\tilde{\phi}$ denotes the azimuthal angle. Port 2 is located from $\tilde{\phi} = 320^o$ to $\tilde{\phi} = 340^o$ in the upper layer. The distance between the centers of two neighboring inductors is 100 μm. Each of the array element is discretized into 498 D_n unknowns.

The cluster trees T_{SWG}, T_{Σ}, T_{ob}, T_{cs}, and T_{disc} are built with leafsizes 80, 30, 30, 4, and 80, respectively. For $\mathcal{H}^2$ partitioning, the admissibility parameter is set as $\eta = 0.95$, and

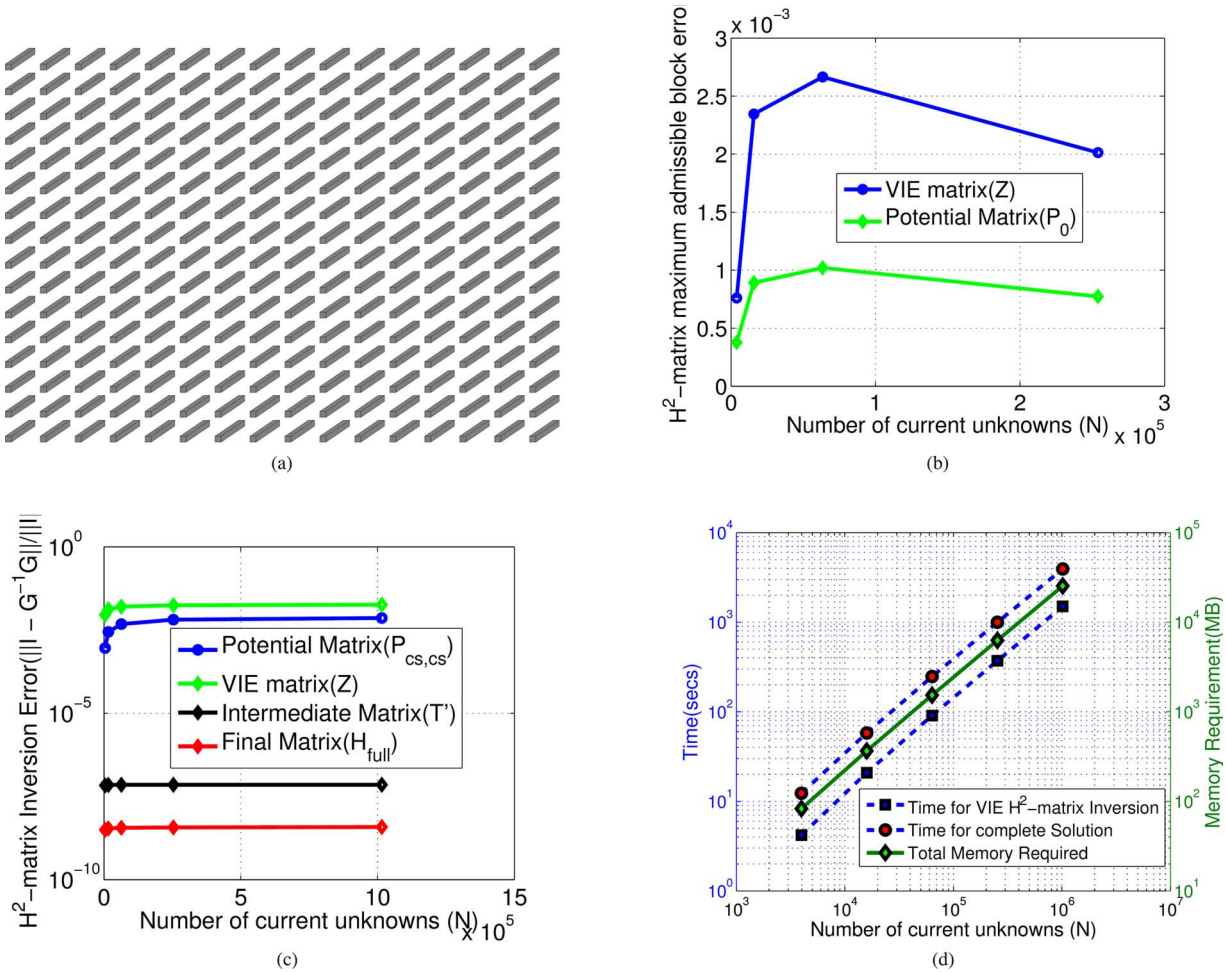

Fig. 5. Simulation results for large scale bus arrays of 4 × 4 to 64 × 64 elements. (a) 3-D view of a 16 × 16 array. (b) $\mathcal{H}^2$-matrix representation errors. (c) $\mathcal{H}^2$-matrix inverse errors. (d) CPU time and memory.

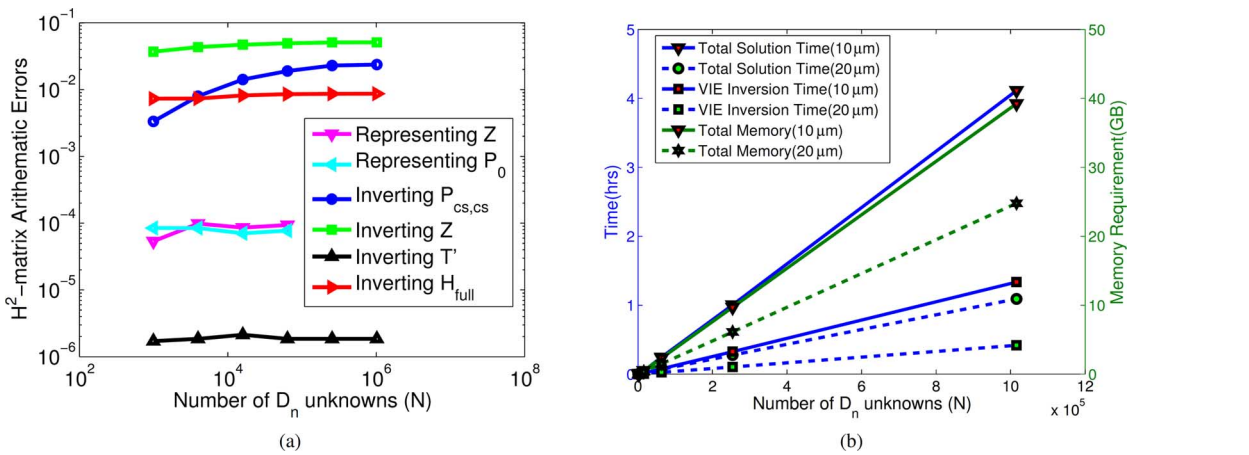

Fig. 6. Simulation results for large scale bus arrays of 4 × 4 to 64 × 64 elements with a smaller inter-element spacing of 10 μm. (a) $\mathcal{H}^2$-matrix representation and inverse errors. (b) CPU time and memory.

again for performance reasons, once the leaf level is reached, the admissibility is not checked and the blocks are allowed to remain as full matrices. The performance of the proposed solver is demonstrated by simulating arrays of 2 × 2 to 64 × 64, resulting in D_n unknowns from 1992 to 2 039 808. The total number of unknowns including both D_n and ϕ unknowns increases from 3744 to 3 833 856. The performance results obtained from these simulations are summarized in Fig. 7. Again, we can see that the error of representing the **Z** matrix of the VIE formulations remains small even with a single-point interpolation scheme. The relative inverse error with respect to identity is also in the 4×10^{-3} range. The total memory requirement for the complete solve, the cost of the most expensive operation, i.e., the inversion of the **Z** matrix, as well as the total solution

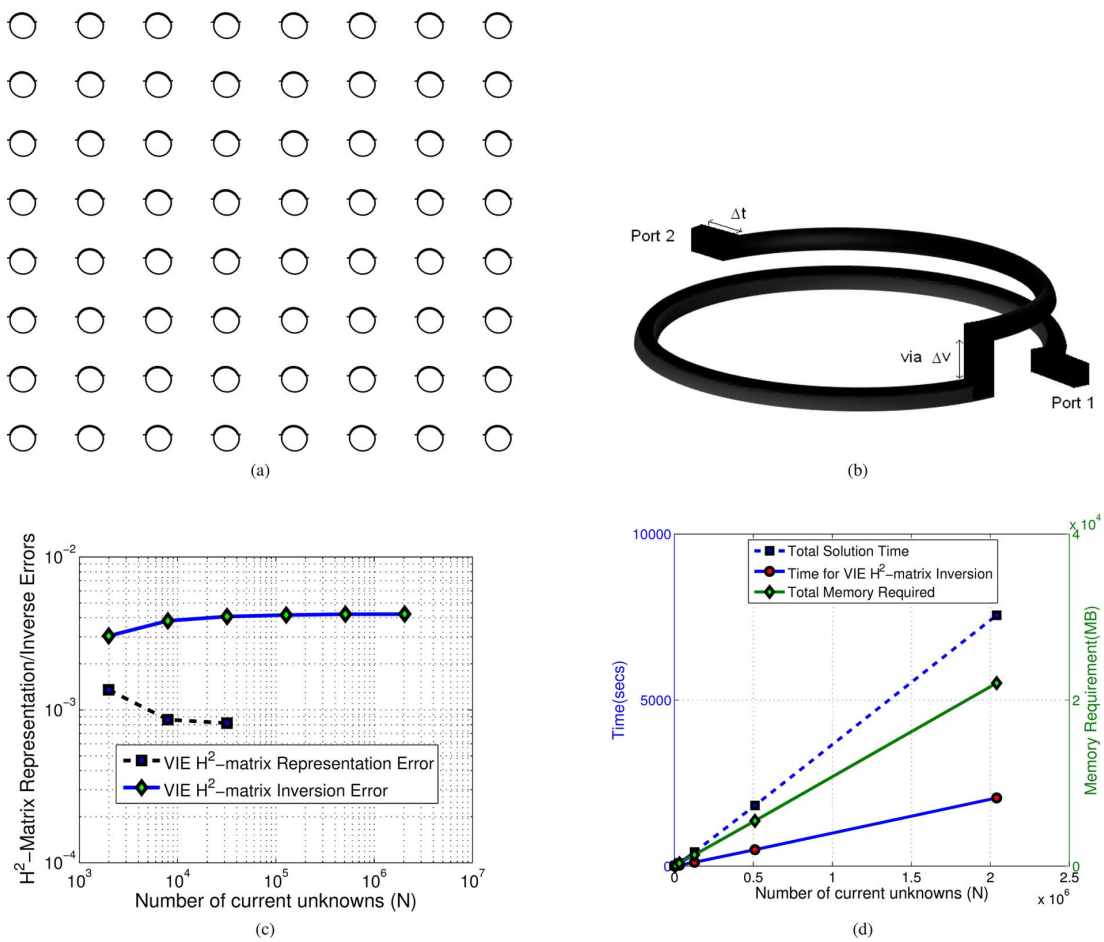

Fig. 7. Simulation results for a suite of larg-scale arrays of irregularly shaped inductors from 2×2 to 64×64 elements. (a) Top view of an 8×8 array. (b) 3-D view of a single irregularly shaped inductor . (c) VIE $\mathcal{H}^2$-matrix and inverse errors. (d) CPU time and memory.

time show a clear linear scaling with respect to the number of unknowns.

VIII. Conclusions

An $\mathcal{H}^2$-matrix based linear complexity direct inverse for the VIE has been developed for the broadband full-wave extraction of general 3-D circuits, containing arbitrarily shaped lossy conductors immersed in inhomogeneous dielectrics, with ports located anywhere in the physical layout of the circuit. A well-conditioned VIE formulation has been proposed, which is then cast into a well-conditioned reduced form, facilitating a fast linear complexity inverse of good accuracy. An efficient $\mathcal{H}^2$-partition and modified algorithms for the resulting highly irregular matrices are also presented, which successfully solve the VIE irregular system in $O(N)$ complexity. The well-conditioned VIE formulation has been presented for both potential source and incident field based excitations. Numerical experiments have validated the accuracy, performance, and linear complexity of the proposed direct VIE solver. Recently, this work has also been extended to a rank-minimized direct VIE solver [48].

References

[1] D. H. Schaubert, D. R. Wilton, and A. W. Glisson, "A tetrahedral modeling method for electromagnetic scattering by arbitrarily shaped inhomogeneous dielectric bodies," *IEEE Trans. Antennas Propag.*, vol. AP-32, no. 1, pp. 77–85, Jan. 1984.

[2] A. E. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Trans. Microw. Theory Techn.*, vol. MTT-22, no. 3, pp. 216–221, Mar. 1974.

[3] M. Kamon, M. J. Tsuk, and J. K. White, FastHenry. Res. Lab. Electron., MIT, Cambridge, MA, USA [Online]. Available: http://www.rle.mit.edu/cpg/research_codes.htm

[4] M. Kamon, M. J. Tsuk, and J. K. White, "FASTHENRY: A multipole-accelerated 3-D inductance extraction program," *IEEE Trans. Microw. Theory Techn.*, vol. 42, no. 9, pp. 1750–1758, Sep. 1994.

[5] M. M. Botha, "Solving the volume integral equations of electromagnetic scattering," *J. Comput. Phys.*, vol. 218, no. 1, pp. 141–158, 2006.

[6] M. I. Sancer, K. Sertel, J. L. Volakis, and P. Van Alstine, "On volume integral equations," *IEEE Trans. Antennas Propag.*, vol. 54, no. 5, pp. 1488–1495, May 2006.

[7] L. E. Sun and W. C. Chew, "A novel formulation of the volume integral equation for general large electromagnetic scattering problems," in *IEEE Int. Antennas Propag. Symp.*, San Diego, CA, USA, 2008, 4 pp.

[8] C.-C. Lu, P. Yla-Oijala, M. Taskinen, and J. Sarvas, "Comparison of two volume integral equation formulations for solving electromagnetic scattering by inhomogeneous dielectric objects," in *IEEE Int. Antennas Propag. Symp.*, Charleston, SC, USA, 2009, 4 pp.

[9] C. Pelletti, G. Bianconi, R. Mittra, and A. Monorchio, "Volume integral equation analysis of thin dielectric sheet using sinusoidal macro-basis functions," *IEEE Antennas Wireless Propag. Lett.*, vol. 12, pp. 441–444, Jul. 2009.

[10] S. Omar and D. Jiao, "A new volume integral equation formulation for analyzing 3-D circuits in inhomogeneous dielectrics exposed to external fields," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Seattle, WA, USA, 2013, 3 pp.

[11] S. Omar and D. Jiao, "A new volume integral formulation for fullwave extraction of 3-D circuits in inhomogeneous dielectrics exposed to external fields," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, Jul. 2013, pp. 730–731.

[12] S. Omar and D. Jiao, "A new volume integral formulation for broadband 3-D circuit extraction in inhomogeneous materials with and without external electromagnetic fields," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 12, pp. 4302–4312, Dec. 2013.
[13] W. C. Chew, J. M. Jin, E. Michielssen, and J. M. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Norwood, MA, USA: Artech House, 2001.
[14] K. Sertel and J. L. Volakis, "Multilevel fast multipole method solution of volume integral equations using parametric geometry modeling," *IEEE Trans. Antennas Propag.*, vol. 52, no. 7, pp. 1686–1692, Jul. 2004.
[15] S. Kurz, O. Rain, and S. Rjasanow, "The adaptive cross-approximation technique for the 3-D boundary-element method," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 421–424, Mar. 2002.
[16] S. Kapur and D. E. Long, "IES^3: A fast integral equation solver for efficient 3-dimensional extraction," in *IEEE/ACM Int. Conf. Comput.-Aided Design Dig.*, Nov. 1997, pp. 448–455.
[17] H. Gan and W. C. Chew, "A discrete BCG-FFT algorithm for solving 3D inhomogeneous scatterer problems," *J. Electromagn. Waves Appl.*, vol. 9, no. 10, pp. 1339–1357, 1995.
[18] Z. Zhu, B. Song, and J. White, "Algorithms in fastimp: A fast and wide-band impedance extraction program for complicated 3-D geometries," in *40th ACM/EDAC/IEEE Design Automat. Conf.*, 2003, pp. 712–717.
[19] W. Shi, J. Liu, N. Kakani, and T. Yu, "A fast hierarchical algorithm for 3-D capacitance extraction," *IEEE Trans. Comput.-Aided Des.*, vol. 21, no. 3, pp. 330–336, Mar. 2002.
[20] N. A. Ozdemir, "Method of moments solution of a non-conformal volume integral equation via the IE-FFT algorithm for electromagnetic scattering from penetrable objects" Ph.D. dissertation, The Ohio State University, Columbus, OH, USA, 2007 [Online]. Available: https://etd.ohiolink.edu/
[21] 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," in *42th ACM/EDAC/IEEE Des. Automat. Conf.*, 2005, pp. 159–162.
[22] 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.
[23] R. J. Adams, Y. Xu, X. Xu, J. Choi, 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.
[24] 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.
[25] L. Greengard, D. Gueyffier, P.-G. Martinnson, and V. Rokhlin, "Fast direct solvers for integral equations in complex three-dimensional domains," *Acta Numer.*, vol. 18, pp. 243–275, May 2009.
[26] 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 Techn.*, vol. 59, no. 10, pp. 2404–2421, Oct. 2011.
[27] 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.
[28] W. Chai and D. Jiao, "Direct matrix solution of linear complexity for surface integral-equation based impedance extraction of complicated 3-D structures," *Proc. IEEE*, vol. 101, no. 2, pp. 372–388, Feb. 2013.
[29] 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, 4 pp.
[30] W. Chai and D. Jiao, "Linear-complexity direct and iterative integral equation solvers accelerated by a new rank-minimized-representation for large-scale 3-D interconnect extraction," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 8, pp. 2792–2805, Aug. 2013.
[31] A. Heldring, J. M. Rius, J. M. Tamayo, J. Parron, and E. Ubeda, "Multiscale compressed block decomposition for fast direct solution of method of moments linear system," *IEEE Trans. Antennas Propag.*, vol. 59, no. 2, pp. 526–536, Feb. 2011.
[32] A. Freni, P. D. Vita, P. Pirinoli, L. Matekovits, and G. Vecchi, "Fast-factorization acceleration of MoM compressive domain-decomposition," *IEEE Trans. Antennas Propag.*, vol. 59, no. 12, pp. 4588–4599, Dec. 2011.
[33] D. Gonzalez-Ovejero, F. Mesa, and C. Craeye, "Accelerated macro basis functions analysis of finite printed antenna arrays through 2D and 3D multipole expansions," *IEEE Trans. Antennas Propag.*, vol. 61, no. 2, pp. 707–717, Feb. 2013.
[34] S. Omar and D. Jiao, "An $\mathcal{H}^2$-matrix based fast direct volume integral equation solver for electrodynamic analysis," in *Proc. Int. Annu. Rev. Progr. Appl. Comput. Electromagn.*, Columbus, OH, USA, 2012, 6 pp.
[35] J. Zhu, S. Omar, and D. Jiao, "Solution of the electric field integral equation when it breaks down," *IEEE Trans. Antennas Propag.*, vol. 62, no. 8, pp. 4122–4134, Aug. 2014.
[36] S. Omar and D. Jiao, "A linear complexity $\mathcal{H}^2$-matrix based direct volume integral solver for broadband 3-D circuit extraction in inhomogeneous materials," in *IEEE MTT-S Int. Microw. Symp. Dig.*, 2014, 3 pp.
[37] S. Borm, L. Grasedyck, and W. Hackbusch, "Hierarchical matrices," Max Planck Inst. Math. Sci., Leipzig, Germany, Lecture Note 21, 2003.
[38] W. Hackbusch and B. Khoromskij, "A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: introduction to $\mathcal{H}$-matrices," *Computing*, vol. 62, pp. 89–108, 1999.
[39] W. Hackbusch and B. Khoromskij, "A sparse matrix arithmetic. Part II: Application to multi-dimensional problems," *Computing*, vol. 64, pp. 21–47, 2000.
[40] S. Börm, "Introduction to hierarchical matrices with applications," *Eng. Anal. Boundary Elements*, vol. 27, pp. 405–422, 2003.
[41] S. Börm, "Efficient numerical methods for non-local operators: $\mathcal{H}^2$-matrix compression algorithms, analysis," European Math. Soc., Zurich, Switzerland, 2010.
[42] S. Börm, "$\mathcal{H}^2$-matrices multilevel methods for the approximation of integral operators," *Comput. Vis. Sci.*, vol. 7, pp. 173–181, 2004.
[43] S. Börm and W. Hackbusch, "$\mathcal{H}^2$-matrix approximation of integral operators by interpolation," *Appl. Numer. Math.*, vol. 43, pp. 129–143, 2002.
[44] S. Börm, "$\mathcal{H}^2$-matrix arithmetics in linear complexity," *Computing*, vol. 77, pp. 1–28, 2006.
[45] W. Chai and D. Jiao, "Theoretical study on the rank of integral operators for broadband electromagnetic modeling from static to electrodynamic frequencies," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 3, no. 12, pp. 2113–2126, Dec. 2013.
[46] S. Omar and D. Jiao, "$O(N)$ iterative and $O(N\log N)$ direct volume integral equation solvers for large-scale electrodynamic analysis," in *Int. Electromagn. Adv. Appl. Conf.*, Aug. 2014, pp. 593–596.
[47] S. Omar and D. Jiao, "An $\mathcal{H}^2$-matrix based fast volume integral equation solver for electrodynamic analysis," *IET Microw., Antennas, Propag.*, vol. 7, no. 14, pp. 1145–1153, 2013.
[48] S. Omar and D. Jiao, "An $O(N)$ direct volume IE solver with a rank-minimized $\mathcal{H}^2$-representation for large-scale 3-D circuit extraction in inhomogeneous materials," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, Jul. 2014, 2 pp.

Saad Omar (S'13–M'14) received the B.S.E.E degree (with highest distinction) from the University of Engineering and Technology, Lahore, Pakistan, in 2009, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 2014.

Since 2014, he has been affiliated with Schlumberger-Doll Research, Cambridge, MA, USA. His current research interests include computational and applied electromagnetics, direct integral-equation solvers, inverse scattering problems, fast and high-capacity numerical methods, high-performance very large scale integration (VLSI) computer-aided design (CAD) tools, high-frequency VLSI circuit design and analysis, microwave and millimeter wave circuits and bioelectromagnetics.

Dr. Omar is a member of IEEE Microwave Theory and Techniques Society (IEEE MTT-S), IEEE Antennas and Propagation Society (AP-S), and Golden Key International Honour Society. He was the recipient of the IEEE Antennas and Propagation Society Doctoral Research Award for 2013–2014. He was also the recipient of Pakistan's most prestigious Presidential Award, 15 Gold Medals, and the National Talent Scholarship.

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

From 2001 to September 2005, she was a Senior Computer-Aided Design (CAD) Engineer, Staff Engineer, and Senior Staff Engineer with the Technology Computer-Aided Design (CAD) Division, Intel Corporation. In September 2005, she joined Purdue University, West Lafayette, IN, USA, as an Assistant Professor with the School of Electrical and Computer Engineering, where she is currently a Professor. She has authored two book chapters and over 200 papers in refereed journals and international conferences. Her current research interests include computational electromagnetics, high-frequency digital, analog, mixed-signal, and RF integrated circuit (IC) design and analysis, high-performance very large scale integration (VLSI) CAD, modeling of microscale and nanoscale circuits, applied electromagnetics, fast and high-capacity numerical methods, fast time-domain analysis, scattering and antenna analysis, RF, microwave, and millimeter-wave circuits, wireless communication, and bio-electromagnetics.

Dr. Jiao has been a reviewer for many IEEE journals and conferences. She is an Associate Editor for the IEEE TRANSACTIONS ON COMPONENTS, PACKAGING, AND MANUFACTURING TECHNOLOGY. She was the recipient of the 2013 S. A. Schelkunoff Prize Paper Award of the IEEE Antennas and Propagation Society, which recognizes the Best Paper published in the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION during the previous year. She has been named a University Faculty Scholar by Purdue University since 2013. She was among 85 engineers selected throughout the nation for the National Academy of Engineering's 2011 U.S. Frontiers of Engineering Symposium. She was the recipient of the 2010 Ruth and Joel Spira Outstanding Teaching Award, the 2008 National Science Foundation (NSF) CAREER Award, the 2006 Jack and Cathie Kozik Faculty Start-up Award (which recognizes an outstanding new faculty member of the School of Electrical and Computer Engineering, Purdue University), a 2006 Office of Naval Research (ONR) Award under the Young Investigator Program, the 2004 Best Paper Award presented at the Intel Corporation's annual corporate-wide technology conference (Design and Test Technology Conference) for her work on generic broadband model of high-speed circuits, the 2003 Intel Corporation's Logic Technology Development (LTD) Divisional Achievement Award, the Intel Corporation's Technology CAD Divisional Achievement Award, the 2002 Intel Corporation's Components Research, the Intel Corporation's Hero Award (Intel-wide she was the tenth recipient), the Intel Corporation's LTD Team Quality Award, and the 2000 Raj Mittra Outstanding Research Award presented by the University of Illinois at Urbana-Champaign.