# Linear-Complexity Direct and Iterative Integral Equation Solvers Accelerated by a New Rank-Minimized $\mathcal{H}^2$-Representation for Large-Scale 3-D Interconnect Extraction

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

***Abstract*—We develop a new rank-minimized $\mathcal{H}^2$-matrix-based representation of the dense system matrix arising from an integral-equation (IE)-based analysis of large-scale 3-D interconnects. Different from the $\mathcal{H}^2$-representation generated by the existing interpolation-based method, the new $\mathcal{H}^2$-representation minimizes the rank in nested cluster bases and all off-diagonal blocks at all tree levels based on accuracy. The construction algorithm of the new $\mathcal{H}^2$-representation is applicable to both real- and complex-valued dense matrices generated from scalar and/or vector-based IE formulations. It has a linear complexity, and hence, the computational overhead is small. The proposed new $\mathcal{H}^2$-representation can be employed to accelerate both iterative and direct solutions of the IE-based dense systems of equations. To demonstrate its effectiveness, we develop a linear-complexity preconditioned iterative solver as well as a linear-complexity direct solver for the capacitance extraction of arbitrarily shaped 3-D interconnects in multiple dielectrics. The proposed linear-complexity solvers are shown to outperform state-of-the-art $\mathcal{H}^2$-based linear-complexity solvers in both CPU time and memory consumption. A dense matrix resulting from the capacitance extraction of a 3-D interconnect having 3.71 million unknowns and 576 conductors is inverted in fast CPU time (1.6 h), modest memory consumption (4.4 GB), and with prescribed accuracy satisfied on a single core running at 3 GHz.**

***Index Terms*—Direct solver, fast solvers, $\mathcal{H}^2$-matrix, integral-equation (IE)-based methods, interconnect extraction, iterative solver.**

## I. Introduction

THE INCREASED level of integration and higher signal speeds have made the analysis and design of integrated circuits and packages increasingly challenging. Existing fast integral-equation (IE) solvers for solving large-scale circuit problems are, in general, iterative solvers [1]–[6], the optimal complexity of which is $O(N_{\rm rhs}N_{\rm it}N)$, where $N_{\rm rhs}$ is the number of right-hand sides, $N_{\rm it}$ is the number of iterations, and $N$ is the matrix size. When $N_{\rm rhs}$ or $N_{\rm it}$ is large, iterative solvers become inefficient. Recently, an $\mathcal{H}^2$-matrix-based mathematical framework [7]–[11] was introduced and further developed to reduce the computational complexity of IE-based solutions [12]–[16]. In [13] and [14], a linear-complexity $\mathcal{H}^2$-based inverse was obtained for IE-based capacitance extraction with arbitrary 3-D geometry and nonuniform materials. In [15], it is shown that an $\mathcal{H}^2$-based LU factorization can also be accomplished in linear complexity. Later in [16], a more general linear-complexity direct solution is developed to solve a highly irregular system matrix mixed with both dense and sparse blocks for impedance extraction of arbitrarily shaped 3-D nonideal conductors in a dielectric medium.

In [12]–[16], the $\mathcal{H}^2$-matrix-based representations of IE-based dense matrices are all generated by an interpolation-based method. The rank of each admissible block is determined by the number of interpolation points. Due to the limitation of an interpolation-based method, the resultant rank for each admissible block is not the minimal rank required by accuracy. If the rank can be minimized based on accuracy, the linear-complexity solutions reported in [13]–[16] can be further accelerated. In mathematical literature, a total cluster basis algorithm [19] has been developed to construct optimal cluster bases. Local Schur decompositions have also been used to eliminate redundant functions from an original $\mathcal{H}^2$-approximation [17], [18].

In this work, we develop a new algorithm to generate a rank-minimized $\mathcal{H}^2$-representation for IE-based large-scale 3-D interconnect extraction. Interconnect extraction refers to the circuit model generation of interconnects including capacitance ($C$), resistance ($R$), inductance ($L$), impedance parameter, $S$-parameter, and other network-parameter extraction. This new algorithm is built upon the mathematical framework of [17]–[19], but goes beyond [17]–[19]. In addition to solving complicated 3-D interconnect problems; algorithm wise, it has the following differences and merits.

- It compresses all the off-diagonal blocks (including off-diagonal inadmissible blocks) in an $\mathcal{H}^2$ matrix to admissible blocks having a minimal rank for the prescribed accuracy.
- It minimizes the rank of both nested cluster bases and coupling matrices for the prescribed accuracy.
- It is applicable to vector-based integral formulations and complex-valued system matrices in addition to

Manuscript received May 17, 2013; accepted June 03, 2013. Date of publication July 02, 2013; date of current version August 02, 2013. This work was supported by the Semiconductor Research Corporation (SRC) under a grant (Task 1292.073) and by the National Science Foundation (NSF) under Award 0747578 and Award 1065318.

W. Chai was with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. She is now with the PrimeRail Group, Synopsys Inc., Mountain View, CA 94043 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.2013.2268741

scalar-based IE formulations and real-valued system matrices.

- It has a linear complexity for 3-D interconnect extraction, and hence, the computational overhead is small.
- It is worth mentioning that both equations and pseudocodes of the proposed algorithm are original, which cannot be found anywhere else.

The proposed rank-minimized $\mathcal{H}^2$-representation can be employed to accelerate both iterative and direct solutions of the IE-based dense systems of equations arising from 3-D interconnect extraction. To demonstrate its effectiveness, in this work, both linear-complexity iterative and direct IE solvers with the new $\mathcal{H}^2$-representation are developed for capacitance extraction with arbitrary 3-D geometry and nonuniform materials. We show that they significantly outperform the linear-complexity IE solvers constructed based on the original interpolation-based $\mathcal{H}^2$-representation [12]–[16].

To be specific, with the new $\mathcal{H}^2$-representation, a linear-complexity iterative solver has been developed.

- It has significantly reduced storage and CPU time for one matrix-vector product.
- An effective preconditioner is built from the $\mathcal{H}^2$-based inverse of the new $\mathcal{H}^2$-representation in linear complexity. It further accelerates the iterative solver by greatly reducing the number of iterations.

With the new $\mathcal{H}^2$-representation, a linear-complexity direct solver is also completed.

- It outperforms the state-of-the-art linear-complexity direct IE solver [13]–[16].
- It is capable of inverting a dense matrix involving 3.71 million unknowns associated with a large-scale 3-D on-chip interconnect, embedded in multiple dielectrics, having 576 conductors, in fast CPU time (1.6 h) and modest memory consumption (4.4 GB) with prescribed accuracy satisfied, on a single 8222SE AMD Opteron processor running at 3 GHz.

The basic idea of the proposed work has been presented in [22], while this paper completes [22] from the perspectives of both algorithm development and numerical experiments.

## II. Background

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

Consider a dense system of linear equations

$$\mathbf{G}x = b. \tag{1}$$

The $\mathcal{H}^2$-matrix representation of dense matrix $\mathbf{G}$ shown in (1) is generally associated with a strong $\eta$-admissibility condition [7]. We denote the index set of the basis functions used in the discretization of (1) by $\mathcal{I} = \{1, 2, \ldots, N\}$, where $N$ is the total number of unknowns. Two subsets $t$ and $s$ of $\mathcal{I}$ are admissible if they satisfy the following strong admissibility condition [7, p. 145]

$$(t,\, s) \text{ are admissible} \\ = \begin{cases} \text{True}, & \text{if } \max\{\mathrm{diam}(\Omega_t), \mathrm{diam}(\Omega_s)\} \le \eta \mathrm{dist}(\Omega_t, \Omega_s) \\ \text{False}, & \text{otherwise} \end{cases} \tag{2}$$

where $\Omega_t$ and $\Omega_s$ are the supports of the union of all the basis functions in $t$ and $s$ respectively, $\mathrm{diam}(.)$ is the Euclidean diameter of a set, $\mathrm{dist}(.,.)$ is the Euclidean distance between two sets, and $\eta$ is a positive parameter that can be used to control the admissibility condition. When the admissibility condition is satisfied, the submatrix $\mathbf{G}^{t,s}$ in (1) can be written in a factorized form as

$$\tilde{\mathbf{G}}^{t,s} := \mathbf{V}^t \mathbf{S}^{t,s} \mathbf{V}^{s^{\mathrm{T}}} \quad \mathbf{V}^t \in \mathbb{C}^{m \times k} \quad \mathbf{S}^{t,s} \in \mathbb{C}^{k \times k} \quad \mathbf{V}^s \in \mathbb{C}^{n \times k} \tag{3}$$

where $\mathbf{V}^t$ (or $\mathbf{V}^s$) is called a cluster basis, subsets $t$ and $s$ are called clusters of a cluster tree [7], and $\mathbf{S}^{t,s}$ is called a coupling matrix; $m$ and $n$ are the cardinality of $t$ and $s$, respectively; $k \in \mathbb{N}$ is the rank of $\tilde{\mathbf{G}}^{t,s}$, which is smaller than $m$ and $n$. In an interpolation-based $\mathcal{H}^2$ representation, $k$ is $p^d$, where $d = 1, 2, 3$ is the dimension of the problem and $p$ is the number of interpolation points along each dimension. The cluster basis $\mathbf{V}^t$ only needs to be stored in the leaf clusters since $\mathbf{V}^t$ is nested, which can be written as

$$\mathbf{V}^t = \begin{pmatrix} \mathbf{V}^{t1}\mathbf{E}^{t1} \\ \mathbf{V}^{t2}\mathbf{E}^{t2} \end{pmatrix} = \begin{pmatrix} \mathbf{V}^{t1} & \\ & \mathbf{V}^{t2} \end{pmatrix} \begin{pmatrix} \mathbf{E}^{t1} \\ \mathbf{E}^{t2} \end{pmatrix} \tag{4}$$

where $t_1, t_2 \in$ children$(t)$, $\mathbf{E}^{t1}$, and $\mathbf{E}^{t2}$ are transfer matrices associated with a nonleaf cluster $t$, which are used to build a connection between $t$ and its two children. If two subsets $t$ and $s$ of $\mathcal{I}$ do not satisfy the admissibility condition given in (2), matrix block $\mathbf{G}^{t,s}$ is inadmissible and represented in its original full matrix form. The low-rank approximation of $\mathbf{G}$ given in (3) together with the nested property shown in (4) constitutes an $\mathcal{H}^2$-matrix representation of $\mathbf{G}$.

It is shown in [7]–[11] that the total storage of an $\mathcal{H}^2$ matrix is $k^2 C_{\mathrm{sp}} O(N)$, where $C_{\mathrm{sp}}$ (sparsity constant) is the maximal number of matrix blocks formed by one cluster in a block cluster tree [7]. An $\mathcal{H}^2$-based matrix-vector multiplication has the same complexity as the storage. In [13]–[15], we show that $\mathcal{H}^2$-based inverse and LU factorization both have a complexity of $k^3 C_{\mathrm{sp}}^2 O(N)$.

### B. IE Formulations for 3-D-Interconnect Extraction

*1) Capacitance Extraction:* Consider a multi-conductor structure embedded in an inhomogeneous material, an IE-based solution to capacitance extraction results in the following dense system of equations [1], [6]:

$$\mathbf{G}q = v \tag{5}$$

where $\mathbf{G} = \begin{bmatrix} \mathbf{P}_{cc} & \mathbf{P}_{cd} \\ \mathbf{E}_{dc} & \mathbf{E}_{dd} \end{bmatrix}$, $q = \begin{bmatrix} q_c \\ q_d \end{bmatrix}$, and $v = \begin{bmatrix} v_c \\ 0 \end{bmatrix}$, in which $q_c$ and $q_d$ are the charge vectors of the conductor panels and the dielectric–dielectric interface panels, respectively, and $v_c$ is the potential vector associated with the conductor panels. The entries of $\mathbf{P}$ and $\mathbf{E}$ are

$$\begin{aligned} \mathbf{P}_{ij} &= \frac{1}{a_i}\frac{1}{a_j}\int_{S_i}\int_{S_j} g^c(\mathbf{r}_i, \mathbf{r}_j) d\mathbf{r}_i d\mathbf{r}_j \\ \mathbf{E}_{ij} &= (\varepsilon_a - \varepsilon_b)\frac{\partial}{\partial n_a}\frac{1}{a_i}\frac{1}{a_j}\int_{S_i}\int_{S_j} g^c(\mathbf{r}_i, \mathbf{r}_j) d\mathbf{r}_i d\mathbf{r}_j \end{aligned} \tag{6}$$

where $a_i$ and $a_j$ are the areas of panel $S_i$ and $S_j$, respectively, $g^c$ is static Green's function, and $\varepsilon_a$ and $\varepsilon_b$ are the permittivity of two adjacent regions $a$ and $b$, and $n_a$ is normal to the dielectric-interface pointing to dielectric $a$. The diagonal entries of $\mathbf{E}_{dd}$ are $e_{ij} = (\varepsilon_a + \varepsilon_b)/(2a_i\varepsilon_0)$.

As can be seen from (5) and (6), the IE formulation for capacitance extraction is a scalar-based one, and the matrices $\mathbf{P}$ and $\mathbf{E}$ are both real-valued matrices.

*2) Impedance Extraction:* The impedance extraction includes both resistance–inductance *(RL)*-extraction and full-wave impedance extraction. From impedance parameters, one can also obtain any other network circuit parameters such as $S$-parameters. The impedance extraction involves vector-based IE formulations and complex-valued system matrices.

In [13], we extracted the impedance of a 3-D structure with ideal conductors. In [16], we developed a direct IE solver of linear complexity to extract the impedance of 3-D lossy conductors in a dielectric medium. In both cases, one essential component in the IE formulation is the electric field integral equation (EFIE), which is a vector-based formulation. In the following, we use EFIE as an example to explain the $\mathcal{H}^2$-representation of an IE-based dense matrix arising from a vector-based formulation. The EFIE can be written as

$$\begin{aligned}\mathbf{E}_i|_{\tan} = &\iint_S [j\omega\mu\mathbf{J}_s(\mathbf{r}')g^i(\mathbf{r},\mathbf{r}')]_{\tan} dS' \\ &-\iint_S \left[\frac{j}{\omega\varepsilon}(\nabla'\cdot\mathbf{J}_s(\mathbf{r}'))\nabla' g^i(\mathbf{r},\mathbf{r}')\right]_{\tan} dS' \qquad (7)\end{aligned}$$

where $\omega$ is the angular frequency, $\mathbf{J}_s$ is the surface current density, $g^i(\mathbf{r},\mathbf{r}') = \exp(-j\kappa|\mathbf{r}-\mathbf{r}'|)/4\pi|\mathbf{r}-\mathbf{r}'|$ is the full-wave Green's function with $\kappa$ being the wavenumber, and the subscript $\tan$ denotes a tangential component. By expanding the unknown $\mathbf{J}_s$ using RWG basis functions [21], and applying Galerkin's method to (7), we obtain

$$\mathbf{G}I = V \qquad (8)$$

where the matrix elements of $\mathbf{G}$ can be seen in [12] and [16].

### C. Interpolation-Based $\mathcal{H}^2$-Representation of IE-Based Dense System Matrices

In [13]–[16], the dense system matrices $\mathbf{G}$ shown in (5) and (8) are represented by $\mathcal{H}^2$ matrices using an interpolation-based method. The details are summarized below.

If two subsets $t$ and $s$ of $\mathcal{I}$ satisfy the strong admissibility condition (2), the original kernel function $g$ in (6) and (7) can be replaced by a degenerate approximation with a controlled accuracy

$$\tilde{g}^{t,s}(\mathbf{r},\mathbf{r}') = \sum_{v\in K^t}\sum_{\mu\in K^s} g(\xi_v^t,\xi_\mu^s)L_v^t(\mathbf{r})L_\mu^s(\mathbf{r}') \qquad (9)$$

where $(\xi_v^t)_{v\in K^t}$ is a family of interpolation points in $t$, $(\xi_\mu^s)_{\mu\in K^s}$ is a family of interpolation points in $s$, and $(L_v^t)_{v\in K^t}$ and $(L_\mu^s)_{\mu\in K^s}$ are corresponding Lagrange polynomials satisfying $L_v(\xi_\tau) = \delta_{v,\tau}$ for all $v,\tau\in K^t$, and $L_\mu(\xi_\tau) = \delta_{\mu,\tau}$ for all $\mu,\tau\in K^s$.

With (9), the submatrix $\tilde{\mathbf{G}}^{t,s}$ in (5) can be written in a factorized form as

$$\begin{aligned}&\tilde{\mathbf{G}}^{t,s} \\ &:= \mathbf{V}_c^t\mathbf{S}_c^{t,s}\mathbf{V}_c^{s^{\mathrm{T}}} \quad \mathbf{V}_c^t\in\mathbb{C}^{t\times K^t} \quad \mathbf{S}_c^{t,s}\in\mathbb{C}^{K^t\times K^s} \quad \mathbf{V}_c^s\in\mathbb{C}^{s\times K^s} \qquad (10)\end{aligned}$$

where

$$\begin{aligned}(\mathbf{V}_c^t)_{iv} &= \left(\frac{1}{a_i}\right)\int_{S_i} L_v^t(\mathbf{r}_i)d\mathbf{r} \\ (\mathbf{V}_c^s)_{j\mu} &= \left(\frac{1}{a_j}\right)\int_{S_j} L_\mu^s(\mathbf{r}_j)d\mathbf{r}' \\ (\mathbf{S}_c^{t,s})_{v\mu} &= \begin{cases} g^c(\xi_v^t,\xi_\mu^s) \\ \quad (t \text{ is composed of conductor panels}) \\ (\varepsilon_a-\varepsilon_b)\frac{\partial g^c(\xi_v^t,\xi_\mu^s)}{\partial n_a} \\ \quad (t \text{ is composed of dielectric panels}) \end{cases} \\ &\text{for } i\in t,\ j\in s,\ v\in K^t, \text{ and } \mu\in K^s. \qquad (11)\end{aligned}$$

If we use the same space of polynomials for all clusters, then $\mathbf{V}_c$ is nested. As can be seen from (11), $\mathbf{V}_c$ is a scalar-based cluster basis, and $\mathbf{S}$ is a real-valued coupling matrix.

Similarly, with (9), the submatrix $\tilde{\mathbf{G}}^{t,s}$ in (8) can be represented as

$$\begin{aligned}&\tilde{\mathbf{G}}^{t,s} := \overrightarrow{\mathbf{V}}^t\left(\mathbf{S}_1^{t,s}\right)\overrightarrow{\mathbf{V}}^{s\mathrm{T}} + \mathbf{V}_d^t\left(\mathbf{S}_2^{t,s}\right)\mathbf{V}_d^{s^{\mathrm{T}}} \\ &\overrightarrow{\mathbf{V}}^t, \mathbf{V}_d^t\in\mathbb{C}^{t\times K^t} \quad \mathbf{S}_{1,2}^{t,s}\in\mathbb{C}^{K^t\times K^s} \quad \overrightarrow{\mathbf{V}}^s, \mathbf{V}_d^s\in\mathbb{C}^{s\times K^s} \\ &\qquad (12)\end{aligned}$$

where

$$\begin{aligned}\left(\overrightarrow{\mathbf{V}}^t\right)_{mv} &= \iint_{S_m}\vec{J}_m(\vec{r})L_v^t(\vec{r})ds \\ (\mathbf{V}_d^t)_{mv} &= \iint_{S_m}(\nabla\cdot\vec{J}_m(\vec{r}))L_v^t(\vec{r})ds \\ \left(\overrightarrow{\mathbf{V}}^s\right)_{n\mu} &= \iint_{S_n}\vec{J}_n(\vec{r}')L_\mu^s(\vec{r}')ds' \\ (\mathbf{V}_d^s)_{n\mu} &= \iint_{S_n}(\nabla'\cdot\vec{J}_n(\vec{r}'))L_\mu^s(\vec{r}')ds' \\ \left(\mathbf{S}_1^{t,s}\right)_{v\mu} &= j\omega\mu g^i(\xi_v^t,\xi_\mu^s) \\ \left(\mathbf{S}_2^{t,s}\right)_{v\mu} &= \frac{-j}{\omega\varepsilon}g^i(\xi_v^t,\xi_\mu^s)\end{aligned}$$

with $m\in t$, $n\in s$, $v\in K^t$, $\mu\in K^s$, and $\vec{J}$ being RWG basis functions.

The $\mathcal{H}^2$-representation shown in (12) is composed of a vector-based cluster basis $\overrightarrow{\mathbf{V}}$, a scalar-based cluster basis $\mathbf{V}_d$, and complex-valued coupling matrices $\mathbf{S}_1$ and $\mathbf{S}_2$. If we expand the vector-based $\overrightarrow{\mathbf{V}}$ as

$$\overrightarrow{\mathbf{V}} = [\mathbf{V}_x, \mathbf{V}_y, \mathbf{V}_z]$$

with

$$(\mathbf{V}_{x,y,z})_{mv} = \iint_{S_m}(J_{x,y,z})_m(\vec{r})L_v^t(\vec{r})ds$$

then the vector-based (12) can be decomposed into a scalar-based one,

$$\begin{aligned}\tilde{\mathbf{G}}^{t,s} &= \tilde{\mathbf{G}}_x^{t,s} + \tilde{\mathbf{G}}_y^{t,s} + \tilde{\mathbf{G}}_z^{t,s} + \tilde{\mathbf{G}}_d^{t,s} \\ &= \mathbf{V}_x^t \left(\mathbf{S}_x^{t,s}\right) \mathbf{V}_x^{s^{\mathrm{T}}} + \mathbf{V}_y^t \left(\mathbf{S}_y^{t,s}\right) \mathbf{V}_y^{s^{\mathrm{T}}} \\ &\quad + \mathbf{V}_z^t \left(\mathbf{S}_z^{t,s}\right) \mathbf{V}_z^{s^{\mathrm{T}}} + \mathbf{V}_d^t \left(\mathbf{S}_2^{t,s}\right) \mathbf{V}_d^{s^{\mathrm{T}}}\end{aligned}$$

with

$$\mathbf{S}_{x,y,z}^{t,s} = \mathbf{S}_1^{t,s} \text{ and } \mathbf{V}_{x,y,z,d}^t \in \mathbb{C}^{t \times K^t} \quad \mathbf{V}_{x,y,z,d}^s \in \mathbb{C}^{s \times K^s}. \tag{13}$$

### D. Interpolation-Based $\mathcal{H}^2$-Representation is not Rank-Minimized $\mathcal{H}^2$-Representation

For a given admissibility condition, and hence, $\mathcal{H}^2$-partition, the sparsity constant $C_{\mathrm{sp}}$ is a fixed constant. Thus, theoretically speaking, to minimize the storage and computational cost associated with an $\mathcal{H}^2$-based matrix for a given $\mathcal{H}^2$-partition, the $\mathcal{H}^2$-representation must possess a minimal rank in both cluster bases and coupling matrices for a prescribed accuracy requirement.

Given an accuracy requirement $\varepsilon$, it has been proven that the rank-$r$ representation $(\mathbf{R})$ generated from singular value decomposition (SVD) is a minimal rank approximation of the original matrix $\mathbf{M}$ that fulfills $\|\mathbf{M} - \mathbf{R}\|_2 \leq \varepsilon$ [23]. The SVD-based minimal-rank approximation *does not separate observation and source coordinates*. It treats the entire matrix as a whole and finds a minimal number of vectors, and hence, rank, to represent the matrix with prescribed accuracy. In contrast, an interpolation-based method separates source and observation coordinates in approximating Green's function, as can be seen from (9). As a result, the Green's function $g(|\vec{r} - \vec{r}'|)$, which only depends on the *distance* between source $\vec{r}'$ and observer $\vec{r}$, is approximated by $L_v^t(\vec{r}) L_\mu^s(\vec{r}')$, thus becoming a function of the full coordinates of $\vec{r}'$ and $\vec{r}$. Once it becomes a function of complete source and observation coordinates, the accuracy of (9) is always restricted by the Nyquist sampling theorem, hence leading to a much larger rank than that obtained from an SVD-based minimal-rank representation for the same accuracy. A detailed rank study of electromagnetics-based IE operators can be found in [24].

To find a minimal rank based on accuracy for a single matrix block $\mathbf{G}^{t,s}$, as shown in (10), we can compute its SVD directly. We can also compute the Gram matrix of $\mathbf{G}^{t,s}$, and then perform Schur decomposition of the Gram matrix to obtain a minimal rank based on prescribed accuracy [19]. Denoting the Gram matrix of $\mathbf{G}^{t,s}$ by $\mathbf{G}_2^{t,s}$, which is $\mathbf{G}_2^{t,s} = \mathbf{G}^{t,s} \mathbf{G}^{t,s^{\mathrm{H}}}$. Since $\mathbf{G}_2^{t,s}$ is Hermitian, the Schur decomposition of $\mathbf{G}_2^{t,s}$ yields

$$\mathbf{G}_2^{t,s} \overset{\text{by Schur decomposition}}{=} \mathbf{P}\mathbf{D}_2\mathbf{P}^{\mathrm{H}} \tag{14}$$

where the superscript H denotes a conjugate and transpose operation, $\mathbf{P}$ is a unitary matrix, and the diagonal matrix $\mathbf{D}_2 = \mathbf{D}^2$ with $\mathbf{D} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min})$ is comprised of all singular values of $\mathbf{G}^{t,s}$ with $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_{\min} \geq 0$. Hence, given an accuracy requirement $\varepsilon$, one can generate a low-rank representation of $\mathbf{G}^{t,s}$ with rank $k$ determined from the following to satisfy the prescribed accuracy:

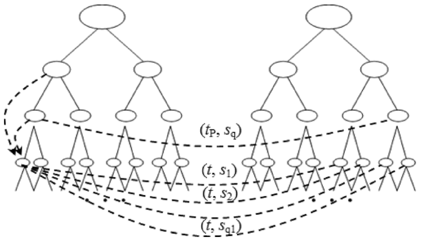


Fig. 1. Block cluster tree formed between a cluster tree and itself.

$$\frac{\left\|\mathbf{G}_2^{t,s} - \mathbf{P}_k \mathbf{P}_k^{\mathrm{T}} \mathbf{G}_2^{t,s}\right\|_{\mathrm{F}}}{\left\|\mathbf{G}_2^{t,s}\right\|_{\mathrm{F}}} = \frac{\sqrt{\sum_{i=k+1}^{\min} \sigma_i^2}}{\sqrt{\sum_{i=1}^{\min} \sigma_i^2}} < \varepsilon \tag{15}$$

where $\mathbf{P}_k$ denotes the first $k$ columns of $\mathbf{P}$. The Schur decomposition is computationally expensive for large matrices. In Section III, we present a detailed algorithm to compute the minimal rank of each matrix block in an $\mathcal{H}^2$ matrix based on the Schur decomposition of Gram matrices in linear complexity, from which we construct a new $\mathcal{H}^2$-rerpesentation with its rank minimized for the prescribed accuracy.

## III. Linear-Complexity Construction of a New Rank-Minimized $\mathcal{H}^2$-Represențaion

### A. Formulate Gram Matrices of an $\mathcal{H}^2$-Based Representation

Consider a general cluster tree T, shown in the left half of Fig. 1, which depicts a multi-level partition of the entire unknown set $\mathcal{I} = \{1, 2, \ldots, N\}$. This tree is built level by level from the root level of the complete unknown set $\mathcal{I} := \{1, 2, \ldots, N\}$ to the leaf level where the number of unknowns in each node is no greater than a predetermined constant *leafsize*. Each node in the cluster tree is called a cluster. It denotes a subset of unknowns. The nodes in the leaf (bottom) level are called leaf clusters, while others are called nonleaf clusters. It is evident that the total number of clusters in the cluster tree is $O(N)$.

The *block* cluster tree that characterizes the interaction between unknowns is built between T and itself when Galerkin's method is used for discretizing the IE operator. A cluster $t$ in a cluster tree, based on the admissibility condition (2), forms admissible blocks $(t, s_1), (t, s_2), \ldots, (t, s_{q1})$ in its tree level, as illustrated by the links shown in Fig. 1. The cluster basis $\mathbf{V}^t$ associated with cluster $t$ is used for the representation of all the admissible blocks formed by $t$. Due to nested property shown in (4), $\mathbf{V}^t$ is also indirectly used for all the admissible blocks $(t_P, s_q)$ formed by $t$'s ancestors $t_P$, as shown in Fig. 1. Therefore, based on the total cluster basis algorithm [19], for each cluster $t$ in the cluster tree, regardless of leaf level or nonleaf level, we collect all the $t$-related matrix blocks including those horizontally formed by the cluster $t$ in the same tree level, as well as those vertically formed by the $t$'s ancestors across the tree level. All these blocks are stored in a matrix $\mathbf{G}^t$ as the following:

$$\mathbf{G}^t = \left[\tilde{\mathbf{G}}^{t,s1}, \tilde{\mathbf{G}}^{t,s2}, \ldots, \tilde{\mathbf{G}}_t^{tp,sq}\right] \tag{16}$$

where each matrix block in (16) is a single matrix block that is either admissible or inadmissible, and $tp$ denotes $t$'s ancestors. Note that $\tilde{\mathbf{G}}_t^{tp,sq}$ is not the entire block formed by $tp$, but a part of the block corresponding to $t$. Based on the nested property shown in (4), we can derive $\tilde{\mathbf{G}}_t^{tp,sq}$ as the following:

$$\tilde{\mathbf{G}}_t^{tp,sq} = \mathbf{V}^t\mathbf{E}^t\mathbf{S}^{tp,sq}\mathbf{V}^{sq^{\mathrm{T}}} \tag{17}$$

which is only the $t$-related part of the admissible block formed by $tp$ and $sq$. We next compute the Gram matrix of $\mathbf{G}^t$ as

$$\mathbf{G}_2^t = \mathbf{G}^t\mathbf{G}^{t^{\mathrm{H}}} \tag{18}$$

which can be efficiently evaluated as the following:

$$\begin{aligned}
\mathbf{G}_2^t =& \sum_{s\in col(t)} \tilde{\mathbf{G}}^{(t,s)}\tilde{\mathbf{G}}^{(t,s)^{\mathrm{H}}} + \sum_{s\in col(tp)} \tilde{\mathbf{G}}^{(tp,s)}\tilde{\mathbf{G}}^{(tp,s)^{\mathrm{H}}} \\
=& \sum_{s\in col^+(t)} \mathbf{V}^t\mathbf{S}^{t,s}\mathbf{V}^{s^{\mathrm{T}}}\overline{\mathbf{V}}^s\mathbf{S}^{t,s^{\mathrm{H}}}\mathbf{V}^{t^{\mathrm{H}}} + \sum_{s\in col^-(t)} \mathbf{F}^{(t,s)}\mathbf{F}^{(t,s)^{\mathrm{H}}} \\
&+ \sum_{s\in col(tp)} \mathbf{V}^t\mathbf{E}^t\mathbf{S}^{tp,s}\mathbf{V}^{s^{\mathrm{T}}}\overline{\mathbf{V}}^s\mathbf{S}^{tp,s^{\mathrm{H}}}\left(\mathbf{V}^t\mathbf{E}^t\right)^{\mathrm{H}} \\
=& \mathbf{V}^t\left(\sum_{s\in col^+(t)} \mathbf{S}^{t,s}\mathbf{B}^s\mathbf{S}^{t,s^{\mathrm{H}}}\right. \\
&\left. + \mathbf{E}^t\left(\sum_{s\in col(tp)} \mathbf{S}^{tp,s}\mathbf{B}^s\mathbf{S}^{tp,s^{\mathrm{H}}}\right)\mathbf{E}^{t^{\mathrm{H}}}\right)\mathbf{V}^{t^{\mathrm{H}}} \\
&+ \sum_{s\in col^-(t)} \mathbf{F}^{(t,s)}\mathbf{F}^{(t,s)^{\mathrm{H}}} \\
=& \mathbf{V}^t\underbrace{\left(\overline{\mathbf{S}}_{\mathrm{sum}}^t + \mathbf{E}^t\overline{\mathbf{S}}_{\mathrm{sum}}^{tp}\mathbf{E}^{t^{\mathrm{H}}}\right)}_{\mathbf{S}_{\mathrm{sum}}^t}\mathbf{V}^{t^{\mathrm{H}}} + \underline{\sum_{s\in col^-(t)} \mathbf{F}^{(t,s)}\mathbf{F}^{(t,s)^{\mathrm{H}}}}
\end{aligned} \tag{19}$$

where

$$\begin{aligned}
\mathbf{B}^s &= \mathbf{V}^{s^{\mathrm{T}}}\overline{\mathbf{V}}^s \\
\overline{\mathbf{S}}_{\mathrm{sum}}^t &= \sum_{s\in col^+(t)} \mathbf{S}^{t,s}\mathbf{B}^s\mathbf{S}^{t,s^{\mathrm{H}}} \\
\mathbf{S}_{\mathrm{sum}}^t &= \overline{\mathbf{S}}_{\mathrm{sum}}^t + \mathbf{E}^t\overline{\mathbf{S}}_{\mathrm{sum}}^{tp}\mathbf{E}^{t^{\mathrm{H}}}
\end{aligned} \tag{20}$$

$\overline{\mathbf{V}}^s$ is the conjugate of $\mathbf{V}^s$, $col(t)$, $col^+(t)$, and $col^-(t)$ denote a set of column clusters $s$ that can form matrix blocks, admissible blocks, and inadmissible blocks, respectively, with $t$. A full matrix block $\tilde{\mathbf{G}}^{t,s}$ is denoted by $\mathbf{F}^{t,s}$ in (19).

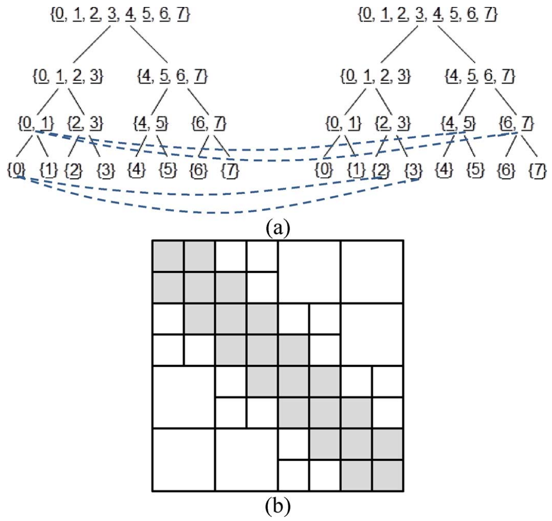


Fig. 2. $\mathcal{H}^2$-matrix example. (a) Block cluster tree built between cluster tree T and itself. (b) Matrix structure.

To give an example, consider a problem of $N = 16$ unknowns, the cluster tree of which, T, has four levels, as shown in the left half of Fig. 2(a), with leafsize being 2. In Fig. 2(a), each link represents an admissible block formed between a row cluster and a column cluster. Due to space limit, not all the admissible links are shown in Fig. 2(a). The $\mathcal{H}^2$-matrix structure corresponding to Fig. 2(a) is shown in Fig. 2(b), where each shaded block is a full-matrix block, while each white block is an admissible block that has a low-rank form. Now, consider a leaf cluster $t := \{0\}$ shown in cluster tree T. $\mathbf{G}^t$ in (16) is $\mathbf{G}^{\{0\}}$, which is the union of all the admissible blocks and inadmissible blocks formed by cluster $\{0\}$ in the same tree level, as well as the admissible blocks formed by cluster $\{0\}$'s ancestors at all the other levels. As can be seen from Fig. 2, cluster $\{0\}$ forms an inadmissible block with clusters $\{0\}$ and $\{1\}$, respectively, in the same tree level. Meanwhile, cluster $\{0\}$ forms an admissible block with clusters $\{2\}$, and $\{3\}$, respectively, in the same tree level. In addition, cluster $\{0\}$'s ancestor $\{0,1\}$ forms an admissible block with clusters $\{4,5\}$, and $\{6,7\}$, respectively, at one level above the leaf level, and beyond that level, no more admissible blocks are formed by cluster $\{0\}$'s ancestors. Hence,

$$\begin{aligned}
\mathbf{G}^{\{0\}} = \Big[&\tilde{\mathbf{G}}^{\{0\},\{0\}}, \tilde{\mathbf{G}}^{\{0\},\{1\}}, \tilde{\mathbf{G}}^{\{0\},\{2\}}, \\
&\tilde{\mathbf{G}}^{\{0\},\{3\}}, \tilde{\mathbf{G}}_0^{\{0,1\},\{4,5\}}, \tilde{\mathbf{G}}_0^{\{0,1\},\{6,7\}}\Big].
\end{aligned}$$

As shown in (17), for the block formed at nonleaf levels by cluster $t$'s ancestors $tp$, we take the part corresponding to cluster $t$ into $\mathbf{G}^t$ using the nested property of cluster bases. Now, $t := \{0\}$ and $tp := \{0,1\}$. Hence,

$$\begin{aligned}
\tilde{\mathbf{G}}_{\{0\}}^{\{0,1\},\{4,5\}} &= \mathbf{V}^{\{0\}}\mathbf{E}^{\{0\}}\mathbf{S}^{\{0,1\},\{4,5\}}\mathbf{V}^{\{4,5\}^{\mathrm{T}}} \\
\tilde{\mathbf{G}}_{\{0\}}^{\{0,1\},\{6,7\}} &= \mathbf{V}^{\{0\}}\mathbf{E}^{\{0\}}\mathbf{S}^{\{0,1\},\{6,7\}}\mathbf{V}^{\{6,7\}^{\mathrm{T}}}.
\end{aligned}$$

From the components of $\mathbf{G}^{\{0\}}$, it can be clearly seen that $col(t)$, $col^{+}(t)$, and $col^{-}(t)$ for cluster $t := \{0\}$ in (19) are, respectively,

$$\begin{aligned} col(t) &= col(\{0\}) = [\{0\}, \{1\}, \{2\}, \{3\}] \\ col^{+}(t) &= col^{+}(\{0\}) = [\{2\}, \{3\}] \\ col^{-}(t) &= col^{-}(\{0\}) = [\{0\}, \{1\}] \\ col(tp) &= [\{4,5\}, \{6,7\}]. \end{aligned}$$

```
Procedure Comp_$\overline{\mathbf{S}}_{sum}$ ($b$) ($b$ is the input $\mathcal{H}^2$ partition)
    if $b$ is a non-leaf matrix block
        for ($i$=0;$i$<children;$i$++)
            if $b$ ($i$) is an admissible block
                $\overline{\mathbf{S}}^{t}_{sum} = \overline{\mathbf{S}}^{t}_{sum} + \mathbf{S}^{t,s}\mathbf{B}^{s}\mathbf{S}^{t,s^{\mathrm{H}}}$
            if $b$ ($i$) is an inadmissible block                    (21)
                $\mathbf{G}^{t}_{2} = \mathbf{G}^{t}_{2} + \mathbf{F}^{t,s}\mathbf{F}^{t,s^{\mathrm{H}}}$
            if $b$ ($i$) is a non-leaf block
                Comp_$\overline{\mathbf{S}}_{sum}$($b$($i$))
```

### *B. Computing New Nested Cluster Bases With Rank Minimized Based on Accuracy*

To find a new set of nested cluster basis with its rank minimized for a prescribed accuracy, we perform the computation level by level across the cluster tree from bottom leaf level to top root level.

We start with the leaf clusters. Consider the general cluster tree $\mathrm{T}$ shown in the left half of Fig. 1. For a leaf cluster $t$, we can directly compute out its Gram matrix $\mathbf{G}_2^t$ given in (19), then perform a Schur decomposition of $\mathbf{G}_2^t$ based on (14), which is then truncated to rank $k$, as shown in (15), based on prescribed accuracy $\varepsilon$. The resultant $\mathbf{P}_k$ is the new cluster basis $\tilde{\mathbf{V}}^t$ that has a minimal rank required by accuracy for the leaf cluster $t$.

However, before we compute (19), we should prepare for $\mathbf{S}^t_{\mathrm{sum}}$ for every cluster $t$ in the cluster tree first. To prepare for $\mathbf{S}^t_{\mathrm{sum}}$ for every cluster $t$, we need to compute $\overline{\mathbf{S}}^t_{\mathrm{sum}}$ first, as can be seen from (20). A pseudo-code for computing $\overline{\mathbf{S}}^t_{\mathrm{sum}}$ is given in (21), from which we obtain $\overline{\mathbf{S}}^t_{\mathrm{sum}}$ for all clusters. We then compute $\mathbf{S}^t_{\mathrm{sum}}$ , the pseudo-code of which is given in (22). After $\mathbf{S}^t_{\mathrm{sum}}$ is obtained for every cluster, the $\mathbf{G}_2^t$ shown in (19) can be readily computed from $\mathbf{V}^t\mathbf{S}^t_{\mathrm{sum}}\mathbf{V}^{t^{\mathrm{H}}} + \sum_{s\in col^{-}(t)} \mathbf{F}^{(t,s)}\mathbf{F}^{(t,s)^{\mathrm{H}}}$. The Schur decomposition of $\mathbf{G}_2^t$ based on accuracy requirement $\varepsilon$ will produce $\mathbf{P}^t$, which is the new cluster basis $\tilde{\mathbf{V}}^t$, whose rank is minimized by accuracy.

For a nonleaf cluster $t$, we cannot directly use the aforementioned procedure since the new cluster basis is required to preserve the nested property in addition to having a minimal rank required by accuracy. As can be seen from (16), when constructing the cluster basis for a child cluster $t$, we have already collected all the upper-level matrix blocks formed by $t$'s ancestors $t_p$. Therefore, we can utilize the child cluster basis to accurately project the cluster basis of the upper-level blocks. For example, assuming $t_1$ and $t_2$ are two children of a nonleaf cluster $t$, since the minimal rank cluster bases are constructed from bottom level to top level, the minimal-rank cluster bases $\tilde{\mathbf{V}}_{t1}$ and $\tilde{\mathbf{V}}_{t2}$ have been computed. Hence, the matrix block $\tilde{\mathbf{G}}^{t,s}$ can be accurately projected as the following:

$$\tilde{\mathbf{G}}^{t,s}_{\mathrm{proj}(t)} = \begin{bmatrix} \tilde{\mathbf{V}}^{t1}\tilde{\mathbf{V}}^{t1^{\mathrm{H}}} & \\ & \tilde{\mathbf{V}}^{t2}\tilde{\mathbf{V}}^{t2^{\mathrm{H}}} \end{bmatrix} \tilde{\mathbf{G}}^{t,s}. \tag{23}$$

```
Procedure Comp_$\mathbf{S}_{sum}$ ($t$) ($t$ is the input cluster tree)
    if $t$ has children
        for ( $i$=0; $i$ <children; $i$++)
            $\overline{\mathbf{S}}^{ti}_{sum} = \overline{\mathbf{S}}^{ti}_{sum} + \mathbf{E}^{ti}\overline{\mathbf{S}}^{t}_{sum}\mathbf{E}^{ti^{\mathrm{H}}}$          (22)
            Comp_$\mathbf{S}_{sum}$ ( $t_i$ )
```

We then update $\mathbf{G}^t$ for a nonleaf $t$ in (16) by replacing all the admissible blocks $\tilde{\mathbf{G}}^{t,s}$ with $\tilde{\mathbf{G}}^{t,s}_{\mathrm{proj}(t)}$. This update does not affect the inadmissible-block-based computation shown in the second underlined part of (19) since all the full-matrix blocks are only formed by leaf clusters. Based on (23) and nested property shown in (4), for a nonleaf cluster $t$, the computation of (18) becomes

$$\begin{aligned} \mathbf{G}_2^t &= \begin{bmatrix} \tilde{\mathbf{V}}^{t1}\tilde{\mathbf{V}}^{t1^{\mathrm{H}}} & \\ & \tilde{\mathbf{V}}^{t2}\tilde{\mathbf{V}}^{t2^{\mathrm{H}}} \end{bmatrix} \mathbf{G}^t\mathbf{G}^{t^{\mathrm{H}}} \begin{bmatrix} \tilde{\mathbf{V}}^{t1}\tilde{\mathbf{V}}^{t1^{\mathrm{H}}} & \\ & \tilde{\mathbf{V}}^{t2}\tilde{\mathbf{V}}^{t2^{\mathrm{H}}} \end{bmatrix}^{\mathrm{H}} \\ &= \begin{bmatrix} \tilde{\mathbf{V}}^{t1} & \\ & \tilde{\mathbf{V}}^{t2} \end{bmatrix} \cdot \underbrace{\left( \begin{bmatrix} \tilde{\mathbf{B}}^{t1}\mathbf{E}^{t1} \\ \tilde{\mathbf{B}}^{t2}\mathbf{E}^{t2} \end{bmatrix} \left(\mathbf{S}^t_{\mathrm{sum}}\right) \begin{bmatrix} \tilde{\mathbf{B}}^{t1}\mathbf{E}^{t1} \\ \tilde{\mathbf{B}}^{t2}\mathbf{E}^{t2} \end{bmatrix}^{\mathrm{H}} \right)}_{\mathbf{G}^t_{2,\mathrm{proj}}} \\ &\quad \cdot \begin{bmatrix} \tilde{\mathbf{V}}^{t1} & \\ & \tilde{\mathbf{V}}^{t2} \end{bmatrix}^{\mathrm{H}} \end{aligned} \tag{24}$$

with $\tilde{\mathbf{B}}^t = \tilde{\mathbf{V}}^{t^{\mathrm{H}}}\mathbf{V}^t$, and the underlined part

$$\mathbf{G}^t_{2,\mathrm{proj}} = \begin{bmatrix} \tilde{\mathbf{B}}^{t1}\mathbf{E}^{t1} \\ \tilde{\mathbf{B}}^{t2}\mathbf{E}^{t2} \end{bmatrix} \left(\mathbf{S}^t_{\mathrm{sum}}\right) \begin{bmatrix} \tilde{\mathbf{B}}^{t1}\mathbf{E}^{t1} \\ \tilde{\mathbf{B}}^{t2}\mathbf{E}^{t2} \end{bmatrix}^{\mathrm{H}}.$$

Next, we only need to perform a Schur decomposition of $\mathbf{G}^t_{2,\mathrm{proj}}$, the size of which is $O(k \times k)$ that is much smaller than the size of $\mathbf{G}_2^t$. This generates an orthogonal space $\mathbf{P}^t$. The rank-$k$ column space of $\mathbf{G}_2^t$ can then be written as

$$\tilde{\mathbf{V}}^t = \begin{bmatrix} \tilde{\mathbf{V}}^{t1} & \\ & \tilde{\mathbf{V}}^{t2} \end{bmatrix} \mathbf{P}^t = \begin{bmatrix} \tilde{\mathbf{V}}^{t1} & \\ & \tilde{\mathbf{V}}^{t2} \end{bmatrix} \begin{bmatrix} \mathbf{P}^{t1} \\ \mathbf{P}^{t2} \end{bmatrix}. \tag{25}$$

Clearly, (25) has an exact form of (4) with the new transfer matrices $\tilde{\mathbf{E}}^{t1}$ and $\tilde{\mathbf{E}}^{t2}$ being $\mathbf{P}^{t1}$ and $\mathbf{P}^{t2}$, respectively. Hence, the nested property is preserved. The pseudo-code for computing the minimal-rank cluster bases for both leaf clusters and nonleaf clusters are shown in (26).

To help understand the aforementioned procedure for nonleaf clusters better, consider the example shown in Fig. 2. To compute the nested cluster basis for nonleaf cluster $t := \{0, 1\}$, we do not need to directly compute the Gram matrix (18) using

$\mathbf{G}^t = \mathbf{G}^{\{0,1\}}$ since the dimension of $\mathbf{G}^t$ can be large, especially for $t$ closer to the root level of the cluster tree.

```
Procedure Mini_Rank_clusters ($t$, $\varepsilon$) ($t$ is the input cluster, $\varepsilon$ is the prescribed accuracy)
if $t$ has children
    for ( i=0; i <children; i++){
            Mini_Rank_clusters ($t_i$, $\varepsilon$)
            Compute $\tilde{\mathbf{B}}^{ti}\mathbf{E}^{ti} \rightarrow be^{ti}$
    }
    $\mathbf{G}^t_{2,proj} = \mathbf{G}^t_{2,proj} + \left( \begin{bmatrix} be^{t1} \\ be^{t2} \end{bmatrix} (\mathbf{S}^t_{sum}) \begin{bmatrix} be^{t1} \\ be^{t2} \end{bmatrix}^{\mathrm{H}} \right)$          (26)
    Perform Schur decomposition on $\mathbf{G}^t_{2,proj}$ based on $\varepsilon$
    $\rightarrow \mathbf{G}^t_{2,proj} = pdp^{\mathrm{H}}$
    $\tilde{\mathbf{E}}^{t1} = p^{t1}$, $\tilde{\mathbf{E}}^{t2} = p^{t2}$, and $\tilde{\mathbf{B}}^t = \tilde{\mathbf{E}}^{t1^{\mathrm{H}}} be^{t1} + \tilde{\mathbf{E}}^{t2^{\mathrm{H}}} be^{t2}$
else
    $\mathbf{G}^t_2 = \mathbf{G}^t_2 + \mathbf{V}^t\mathbf{S}^t_{sum}\mathbf{V}^{t^{\mathrm{H}}}$
    Perform Schur decomposition on $\mathbf{G}^t_2$ based on $\varepsilon$
    $\rightarrow \mathbf{G}^t_2 = pdp^{\mathrm{H}}$
    $\tilde{\mathbf{V}}^t = p^t$ and $\tilde{\mathbf{B}}^t = \tilde{\mathbf{V}}^{t^{\mathrm{H}}}\mathbf{V}^t$
```

Instead, the $\mathbf{G}^t_2$ in (18) has been transformed to (24). Thus, we only need to compute the underlined part $\mathbf{G}^t_{2,\mathrm{proj}} = \mathbf{G}^{\{0,1\}}_{2,\mathrm{proj}}$ in (24), which is a matrix of size $2k$ by $2k$ regardless of the tree level of $t$. To obtain $\mathbf{G}^{\{0,1\}}_{2,\mathrm{proj}}$, it only costs $O(k^3)$ operations since $\tilde{\mathbf{B}}^{t1} = \tilde{\mathbf{B}}^{\{0\}} = \tilde{\mathbf{V}}^{\{0\}^{\mathrm{H}}}\mathbf{V}^{\{0\}}$, $\tilde{\mathbf{B}}^{t2} = \tilde{\mathbf{B}}^{\{1\}} = \tilde{\mathbf{V}}^{\{1\}^{\mathrm{H}}}\mathbf{V}^{\{1\}}$, $\mathbf{E}^{t1} = \mathbf{E}^{\{0\}}$, $\mathbf{E}^{t2} = \mathbf{E}^{\{1\}}$, and $\mathbf{S}^t_{\mathrm{sum}}$ each is of size $k$ by $k$. After $\mathbf{G}^{\{0,1\}}_{2,\mathrm{proj}}$ is obtained, due to its size, performing its Schur decomposition also only costs $O(k^3)$ operations. The resultant $\mathbf{P}^t$ split into $\mathbf{P}^{t1}$ and $\mathbf{P}^{t2}$ provides the transfer matrices of the cluster basis for nonleaf cluster $\{0,1\}$. In addition, for nonleaf clusters, as shown in pseudo-code (26), $\tilde{\mathbf{B}}^t$ is obtained by adding the contributions from $t$'s one-level-down children as $\tilde{\mathbf{B}}^t = \tilde{\mathbf{E}}^{t1^{\mathrm{H}}} be^{t1} + \tilde{\mathbf{E}}^{t2^{\mathrm{H}}} be^{t2}$ instead of computing $\tilde{\mathbf{V}}^{t^{\mathrm{H}}}\mathbf{V}^t$ directly. Hence, the cost of computing $\tilde{\mathbf{B}}^t$ is also $O(k^3)$ for any cluster $t$.

### *C. Computing New Coupling Matrices With Rank Minimized Based on Accuracy*

After obtaining the minimal-rank cluster basis for a prescribed accuracy, we can use it to update the coupling matrices. If the matrix block is an admissible block, we compute

$$\tilde{\mathbf{G}}^{t,s}_{\mathrm{new}} = \tilde{\mathbf{V}}^t\tilde{\mathbf{V}}^{t^{\mathrm{H}}}\tilde{\mathbf{G}}^{t,s}(\tilde{\mathbf{V}}^s\tilde{\mathbf{V}}^{s^{\mathrm{H}}})^{\mathrm{T}} = \tilde{\mathbf{V}}^t\mathbf{S}^{t,s}_{\mathrm{new}}\tilde{\mathbf{V}}^{s^{\mathrm{T}}} \tag{27}$$

where the $\mathbf{S}^{t,s}_{\mathrm{new}}$ is computed by

$$\mathbf{S}^{t,s}_{\mathrm{new}} = \tilde{\mathbf{V}}^{t^{\mathrm{H}}}\tilde{\mathbf{G}}^{t,s}\overline{\tilde{\mathbf{V}}}^s = \tilde{\mathbf{V}}^{t^{\mathrm{H}}}\mathbf{V}^t\mathbf{S}^{t,s}\mathbf{V}^{s^{\mathrm{T}}}\overline{\tilde{\mathbf{V}}}^s = \tilde{\mathbf{B}}^t\mathbf{S}^{t,s}\tilde{\mathbf{B}}^{s^{\mathrm{T}}}. \tag{28}$$

If the matrix block is an off-diagonal inadmissible block, we compute

$$\tilde{\mathbf{G}}^{t,s}_{\mathrm{new}} = \tilde{\mathbf{V}}^t\tilde{\mathbf{V}}^{t^{\mathrm{H}}}\mathbf{F}^{t,s}(\tilde{\mathbf{V}}^s\tilde{\mathbf{V}}^{s^{\mathrm{H}}})^{\mathrm{T}} = \tilde{\mathbf{V}}^t\mathbf{S}^{t,s}_{\mathrm{new}}\tilde{\mathbf{V}}^{s^{\mathrm{T}}}$$

with

$$\mathbf{S}^{t,s}_{\mathrm{new}} = \tilde{\mathbf{V}}^{t^{\mathrm{H}}}\mathbf{F}^{t,s}\overline{\tilde{\mathbf{V}}}^s \tag{29}$$

where $\overline{\tilde{\mathbf{V}}}^s$ denotes the conjugate of $\tilde{\mathbf{V}}^s$.

The pseudo-code to compute the minimal-rank admissible blocks and inadmissible blocks are shown in (30) as follows:

```
Procedure Mini_Rank_blocks ($b$) (with $b$=($t$, $s$))
    if $b$ is a non-leaf matrix block
        for (i=0;i<children;i++)
            Mini_Rank_blocks ( $b(i)$ )
    else if $b$ is an admissible block
        $\mathbf{S}^{t,s}_{new} = \tilde{\mathbf{B}}^t\mathbf{S}^{t,s}\tilde{\mathbf{B}}^{s^{\mathrm{T}}}$          (30)
    else if $b$ is an off-diagonal inadmissible block
        $\mathbf{S}^{t,s}_{new} = \tilde{\mathbf{V}}^{t^{\mathrm{H}}}\mathbf{F}^{t,s}\overline{\tilde{\mathbf{V}}}^s$
```

Based on the procedure in (26) and (30), a new $\mathcal{H}^2$-representation is generated with a minimal rank in both cluster bases and coupling matrices for a given accuracy requirement $\varepsilon$. The resultant new $\tilde{\mathbf{V}}$ is unitary with $\tilde{\mathbf{V}}^{t\mathbf{H}}\tilde{\mathbf{V}}^t = \mathbf{I}$ satisfied for each cluster $t$. In addition, the proposed procedure generates an efficient $\mathcal{H}^2$-matrix partition with only diagonal blocks being full matrix and all the other blocks being admissible since all the off-diagonal inadmissible blocks have been compressed as admissible blocks, as shown in (29).

It is worth mentioning that if the original system matrix $\mathbf{G}$ is not symmetric, its row cluster basis and column cluster basis are different. In such cases, we construct new cluster bases for row clusters and column clusters, respectively, and then use both to update the coupling matrices based on the procedure given in the above.

### *D. Techniques for Handling Vector-Based IE Formulations*

In a vector-based IE formulation, each admissible block in the $\mathcal{H}^2$ representation is composed of multiple sub-blocks, as can be seen from (13). This complicates the operations associated with this block such as matrix–vector multiplications and matrix–matrix multiplications. To overcome this problem, we develop a procedure to merge multiple matrix blocks associated with $(t,s)$ to a single admissible block, and make the resultant block have a minimal rank in both cluster basis and coupling matrix. The detailed procedure is as follows.

First, we minimize the rank of each sub-block of $\tilde{\mathbf{G}}^{t,s}$ by the method described in Sections III-A–III-C; we then merge the first two sub-blocks into a single block. We continue to merge the resultant block with the third sub-block. Such a procedure continues until only one matrix block is left for each $(t,s)$. Take (13) as an example, this process is shown in the equation shown at the bottom of the following page.

In order to efficiently merge two submatrix blocks, we perform

$$\mathbf{V}^t_x\left(\mathbf{S}^{t,s}_x\right)\mathbf{V}^{s^{\mathrm{T}}}_x + \mathbf{V}^t_y\left(\mathbf{S}^{t,s}_y\right)\mathbf{V}^{s^{\mathrm{T}}}_y = \begin{bmatrix} \mathbf{V}^t_x & \mathbf{V}^t_y \end{bmatrix} \begin{bmatrix} \mathbf{S}^{t,s}_x & \\ & \mathbf{S}^{t,s}_y \end{bmatrix} \begin{bmatrix} \mathbf{V}^s_x & \mathbf{V}^s_y \end{bmatrix}^{\mathrm{T}}.$$

We then minimize the rank of the above $\mathcal{H}^2$ representation with $\begin{bmatrix} \mathbf{V}_x & \mathbf{V}_y \end{bmatrix}$ being its cluster basis at a leaf level, $\begin{bmatrix} \mathbf{E}_x & \\ & \mathbf{E}_y \end{bmatrix}$ being its transfer matrix at a nonleaf level, and $\begin{bmatrix} \mathbf{S}_x & \\ & \mathbf{S}_y \end{bmatrix}$ its coupling matrix, by the rank minimization procedure described in Sections III-A–III-C. In addition, we derive the following fast techniques to speed up the computation.

*1) Fast Computation of* $\mathbf{B}^{s} = {\mathbf{V}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}^{s}$ *in (20):* Since $\mathbf{V}_{x}$ and $\mathbf{V}_{y}$ are all orthogonal, we can make use of this property to speed up the computation of $\mathbf{B}^{s}$ in (20). For a leaf cluster $s$,

$$\begin{aligned}\mathbf{B}^{s} &= {\mathbf{V}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}^{s}\\ &= \begin{bmatrix}\mathbf{V}_{x}^{s} & \mathbf{V}_{y}^{s}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\overline{\mathbf{V}}_{x}^{s} & \overline{\mathbf{V}}_{y}^{s}\end{bmatrix}\\ &= \begin{bmatrix}{\mathbf{V}_{x}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{x}^{s} & {\mathbf{V}_{x}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{y}^{s}\\ {\mathbf{V}_{y}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{x}^{s} & {\mathbf{V}_{y}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{y}^{s}\end{bmatrix}\\ &= \begin{bmatrix}\overline{{\mathbf{V}_{x}^{s}}^{\mathrm{H}}\mathbf{V}_{x}^{s}} & {\mathbf{V}_{x}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{y}^{s}\\ {\mathbf{V}_{y}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{x}^{s} & \overline{{\mathbf{V}_{y}^{s}}^{\mathrm{H}}\mathbf{V}_{y}^{s}}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{I} & {\mathbf{V}_{x}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{y}^{s}\\ {\mathbf{V}_{y}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{x}^{s} & \mathbf{I}\end{bmatrix}\\ &= \begin{bmatrix}c^{s}\mathbf{I} & {\mathbf{V}_{x}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{y}^{s}\\ {\mathbf{V}_{y}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}_{x}^{s} & c^{s}\mathbf{I}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{B}_{xx}^{s} & \mathbf{B}_{xy}^{s}\\ \mathbf{B}_{yx}^{s} & \mathbf{B}_{yy}^{s}\end{bmatrix}\end{aligned}\tag{31}$$

where $c^{s}$ is a constant associated with the cluster $s$. When $s$ is a leaf cluster, $c^{s} = 1$. Therefore, the diagonal blocks $\mathbf{B}_{xx}^{s}$ and $\mathbf{B}_{yy}^{s}$ can be directly obtained based on the orthogonality of $\mathbf{V}_{x}$ and $\mathbf{V}_{y}$. In addition, $\mathbf{B}_{yx}^{s} = \left(\mathbf{B}_{xy}^{s}\right)^{\mathrm{H}}$ is satisfied in (32). Hence, we only need to compute one of them. For a nonleaf cluster $s$ with two children $s_1$ and $s_2$, $\mathbf{B}^{s}$ can be computed as follows:

$$\begin{aligned}\mathbf{B}^{s} &= {\mathbf{V}^{s}}^{\mathrm{T}}\overline{\mathbf{V}}^{s}\\ &= \begin{bmatrix}\mathbf{V}^{s1}\mathbf{E}^{s1}\\ \mathbf{V}^{s2}\mathbf{E}^{s2}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\overline{\mathbf{V}^{s1}\mathbf{E}^{s1}}\\ \overline{\mathbf{V}^{s2}\mathbf{E}^{s2}}\end{bmatrix}\\ &= {\mathbf{E}^{s1}}^{\mathrm{T}}\mathbf{B}^{s1}\overline{\mathbf{E}}^{s1} + {\mathbf{E}^{s2}}^{\mathrm{T}}\mathbf{B}^{s2}\overline{\mathbf{E}}^{s2}\\ &= \begin{bmatrix}\mathbf{E}_{x}^{s1} & \\ & \mathbf{E}_{y}^{s1}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{B}_{xx}^{s1} & \mathbf{B}_{xy}^{s1}\\ \mathbf{B}_{yx}^{s1} & \mathbf{B}_{yy}^{s1}\end{bmatrix}\begin{bmatrix}\overline{\mathbf{E}}_{x}^{s1} & \\ & \overline{\mathbf{E}}_{y}^{s1}\end{bmatrix}\\ &\quad + \begin{bmatrix}\mathbf{E}_{x}^{s2} & \\ & \mathbf{E}_{y}^{s2}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{B}_{xx}^{s2} & \mathbf{B}_{xy}^{s2}\\ \mathbf{B}_{yx}^{s2} & \mathbf{B}_{yy}^{s2}\end{bmatrix}\begin{bmatrix}\overline{\mathbf{E}}_{x}^{s2} & \\ & \overline{\mathbf{E}}_{y}^{s2}\end{bmatrix}\\ &= \begin{bmatrix}{\mathbf{E}_{x}^{s1}}^{\mathrm{T}}\left(c^{s1}\mathbf{I}\right)\overline{\mathbf{E}}_{x}^{s1} & {\mathbf{E}_{x}^{s1}}^{\mathrm{T}}\mathbf{B}_{xy}^{s1}\overline{\mathbf{E}}_{y}^{s1}\\ {\mathbf{E}_{y}^{s1}}^{\mathrm{T}}\mathbf{B}_{yx}^{s1}\overline{\mathbf{E}}_{x}^{s1} & {\mathbf{E}_{y}^{s1}}^{\mathrm{T}}\left(c^{s1}\mathbf{I}\right)\overline{\mathbf{E}}_{y}^{s1}\end{bmatrix}\\ &\quad + \begin{bmatrix}{\mathbf{E}_{x}^{s2}}^{\mathrm{T}}\left(c^{s2}\mathbf{I}\right)\overline{\mathbf{E}}_{x}^{s2} & {\mathbf{E}_{x}^{s2}}^{\mathrm{T}}\mathbf{B}_{xy}^{s2}\overline{\mathbf{E}}_{y}^{s2}\\ {\mathbf{E}_{y}^{s2}}^{\mathrm{T}}\mathbf{B}_{yx}^{s2}\overline{\mathbf{E}}_{x}^{s2} & {\mathbf{E}_{y}^{s2}}^{\mathrm{T}}\left(c^{s2}\mathbf{I}\right)\overline{\mathbf{E}}_{y}^{s2}\end{bmatrix}\\ &= \begin{bmatrix}\left(c^{s1}+c^{s2}\right)\mathbf{I} & {\mathbf{E}_{x}^{s1}}^{\mathrm{T}}\mathbf{B}_{xy}^{s1}\overline{\mathbf{E}}_{y}^{s1} + {\mathbf{E}_{x}^{s2}}^{\mathrm{T}}\mathbf{B}_{xy}^{s2}\overline{\mathbf{E}}_{y}^{s2}\\ {\mathbf{E}_{y}^{s1}}^{\mathrm{T}}\mathbf{B}_{yx}^{s1}\overline{\mathbf{E}}_{x}^{s1} + {\mathbf{E}_{y}^{s2}}^{\mathrm{T}}\mathbf{B}_{yx}^{s2}\overline{\mathbf{E}}_{x}^{s2} & \left(c^{s1}+c^{s2}\right)\mathbf{I}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{B}_{xx}^{s} & \mathbf{B}_{xy}^{s}\\ \mathbf{B}_{yx}^{s} & \mathbf{B}_{yy}^{s}\end{bmatrix}.\end{aligned}\tag{32}$$

As can be seen, when $s$ is at a nonleaf level, we only need to get $c_{s1}$ and $c_{s2}$ stored in the children level and directly use them to obtain the diagonal blocks, as shown in (32). $\mathbf{B}_{yx}^{s} = \left(\mathbf{B}_{xy}^{s}\right)^{\mathrm{H}}$ is also satisfied, and hence, we only need to compute one of them.

*2) Fast Computation of* $\overline{\mathbf{S}}_{\mathrm{sum}}^{t}$ *Shown in (20):* Based on the fast computation of $\mathbf{B}^{s}$, we can further accelerate the computation of $\overline{\mathbf{S}}_{\mathrm{sum}}^{t}$ as follows:

$$\begin{aligned}\overline{\mathbf{S}}_{\mathrm{sum}}^{t} &= \mathbf{S}^{t,s}\mathbf{B}^{s}{\mathbf{S}^{t,s}}^{\mathrm{H}}\\ &= \begin{bmatrix}\mathbf{S}_{x}^{t,s} & 0\\ 0 & \mathbf{S}_{y}^{t,s}\end{bmatrix}\begin{bmatrix}c^{s}\mathbf{I} & \mathbf{B}_{xy}^{s}\\ \mathbf{B}_{yx}^{s} & c^{s}\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{S}_{x}^{t,s} & 0\\ 0 & \mathbf{S}_{y}^{t,s}\end{bmatrix}^{\mathrm{H}}\\ &= \begin{bmatrix}c^{s}\mathbf{S}_{x}^{t,s}{\mathbf{S}_{x}^{t,s}}^{\mathrm{H}} & \mathbf{S}_{x}^{t,s}\mathbf{B}_{xy}^{s}{\mathbf{S}_{y}^{t,s}}^{\mathrm{H}}\\ \mathbf{S}_{y}^{t,s}\mathbf{B}_{yx}^{s}{\mathbf{S}_{x}^{t,s}}^{\mathrm{H}} & c^{s}\mathbf{S}_{y}^{t,s}{\mathbf{S}_{y}^{t,s}}^{\mathrm{H}}\end{bmatrix}.\end{aligned}\tag{33}$$

Similarly, $(\overline{\mathbf{S}}_{\mathrm{sum}}^{t})_{21} = {(\overline{\mathbf{S}}_{\mathrm{sum}}^{t})_{12}}^{\mathrm{H}}$ is satisfied, and hence, only one of them needs to be computed.

The aforementioned procedure generates a new $\mathcal{H}^2$ representation that has the following advantages over the conventional one in (13). First, multiple matrix blocks are merged into a single matrix block, which greatly simplifies the computation associated with $(t, s)$ block. Second, we remove linearly dependent vectors in the space of $\tilde{\mathbf{G}}_{x}^{t,s}$, $\tilde{\mathbf{G}}_{y}^{t,s}$, $\tilde{\mathbf{G}}_{z}^{t,s}$, and $\tilde{\mathbf{G}}_{d}^{t,s}$ in (13) by merging them into a single admissible block. The rank of the resultant single block is minimized based on the prescribed accuracy.

### *E. Treatment of Complex-Valued Cluster Bases*

The original cluster bases obtained from an interpolation-based method are real valued, as shown in (10) and (12), while the cluster bases resulting from the aforementioned rank minimization procedure, such as $\tilde{\mathbf{V}}$ in (27) and (29), become complex valued since they take kernel functions into account. In many applications, it is desirable to keep the cluster bases real valued. For example, the basic operation involved in a matrix inverse is the block matrix multiplication

$$\begin{aligned}\mathbf{V}^{t}\mathbf{S}_{1}{\mathbf{V}^{s}}^{\mathrm{T}} \times \mathbf{V}^{s}\mathbf{S}_{2}{\mathbf{V}^{r}}^{\mathrm{T}} &= \mathbf{V}^{t}\mathbf{S}_{1}({\mathbf{V}^{s}}^{\mathrm{T}}\mathbf{V}^{s})\mathbf{S}_{2}{\mathbf{V}^{r}}^{\mathrm{T}}\\ &= \mathbf{V}^{t}(\mathbf{S}_{1}\mathbf{S}_{2}){\mathbf{V}^{r}}^{\mathrm{T}}\end{aligned}\tag{34}$$

the fast equality of which utilizes the orthogonality of cluster bases, i.e., ${\mathbf{V}^{s}}^{\mathrm{T}}\mathbf{V}^{s} = \mathbf{I}$. If $\mathbf{V}$ is a complex-valued orthogonal matrix, we have ${\mathbf{V}^{s}}^{\mathrm{H}}\mathbf{V}^{s} = \mathbf{I}$ instead of ${\mathbf{V}^{s}}^{\mathrm{T}}\mathbf{V}^{s} = \mathbf{I}$. Therefore, the product ${\mathbf{V}^{s}}^{\mathrm{T}}\mathbf{V}^{s}$ cannot be eliminated from (34), and the block matrix multiplication will involve cluster-basis associated computation, which can slow down the computation. To overcome this problem, in the following, we develop a method to convert complex-valued cluster-bases to real-valued cluster bases.

$$\tilde{\mathbf{G}}^{t,s} = \underline{\underline{\underline{\mathbf{V}_{x}^{t}\left(\mathbf{S}_{x}^{t,s}\right){\mathbf{V}_{x}^{s}}^{\mathrm{T}} + \mathbf{V}_{y}^{t}\left(\mathbf{S}_{y}^{t,s}\right){\mathbf{V}_{y}^{s}}^{\mathrm{T}}} + \mathbf{V}_{z}^{t}\left(\mathbf{S}_{z}^{t,s}\right){\mathbf{V}_{z}^{s}}^{\mathrm{T}}} + \mathbf{V}_{d}^{t}\left(\mathbf{S}_{2}^{t,s}\right){\mathbf{V}_{d}^{s}}^{\mathrm{T}}}$$

First, we expand the new $\mathcal{H}^2$ block $\tilde{\mathbf{G}}^{t,s}_{\mathrm{new}}$ shown in (27) as

$$\begin{aligned}\tilde{\mathbf{G}}^{t,s}_{\mathrm{new}} &= \tilde{\mathbf{V}}^{t}\mathbf{S}^{t,s}_{\mathrm{new}}\tilde{\mathbf{V}}^{s^{\mathrm{T}}}\\ &= \left(\tilde{\mathbf{V}}^{t}_{\mathrm{re}} + j\tilde{\mathbf{V}}^{t}_{\mathrm{im}}\right)\mathbf{S}^{t,s}_{\mathrm{new}}\left(\tilde{\mathbf{V}}^{s}_{\mathrm{re}} + j\tilde{\mathbf{V}}^{s}_{\mathrm{im}}\right)^{\mathrm{T}} \qquad (35)\end{aligned}$$

where $\tilde{\mathbf{V}}_{\mathrm{re}}$ and $\tilde{\mathbf{V}}_{\mathrm{im}}$ represent the real and imaginary parts of $\tilde{\mathbf{V}}$, respectively. We put the real and imaginary parts into one matrix $\mathbf{V}_{\mathrm{comb}} = \left[\tilde{\mathbf{V}}^{t}_{\mathrm{re}} \quad \tilde{\mathbf{V}}^{t}_{\mathrm{im}}\right]$, which is a real-valued cluster basis. The transfer matrix for $\mathbf{V}_{\mathrm{comb}}$ at a nonleaf level can be obtained using the following formula:

$$\begin{aligned}\tilde{\mathbf{V}}\tilde{\mathbf{E}} &= \left(\tilde{\mathbf{V}}_{\mathrm{re}} + j\tilde{\mathbf{V}}_{\mathrm{im}}\right)\left(\tilde{\mathbf{E}}_{\mathrm{re}} + j\tilde{\mathbf{E}}_{\mathrm{im}}\right)\\ &= \left(\tilde{\mathbf{V}}_{\mathrm{re}}\tilde{\mathbf{E}}_{\mathrm{re}} - \tilde{\mathbf{V}}_{\mathrm{im}}\tilde{\mathbf{E}}_{\mathrm{im}}\right) + j\left(\tilde{\mathbf{V}}_{\mathrm{re}}\tilde{\mathbf{E}}_{\mathrm{im}} + \tilde{\mathbf{V}}_{\mathrm{im}}\tilde{\mathbf{E}}_{\mathrm{re}}\right)\\ &\quad \text{(Put the above into one matrix)}\\ &\rightarrow \left[\tilde{\mathbf{V}}_{\mathrm{re}}\tilde{\mathbf{E}}_{\mathrm{re}} - \tilde{\mathbf{V}}_{\mathrm{im}}\tilde{\mathbf{E}}_{\mathrm{im}} \quad \tilde{\mathbf{V}}_{\mathrm{re}}\tilde{\mathbf{E}}_{\mathrm{im}} + \tilde{\mathbf{V}}_{\mathrm{im}}\tilde{\mathbf{E}}_{\mathrm{re}}\right]\\ &= \left(\tilde{\mathbf{V}}_{\mathrm{re}}\tilde{\mathbf{V}}_{\mathrm{im}}\right)\underline{\begin{pmatrix}\tilde{\mathbf{E}}_{\mathrm{re}} & \tilde{\mathbf{E}}_{\mathrm{im}}\\ -\tilde{\mathbf{E}}_{\mathrm{im}} & \tilde{\mathbf{E}}_{\mathrm{re}}\end{pmatrix}} \qquad (36)\end{aligned}$$

where the underlined part of (36) is the new transfer matrix for $\mathbf{V}_{\mathrm{comb}}$ at a nonleaf level. Next, we use the linear-time orthogonalization procedure shown in [11] to orthogonalize $\mathbf{V}_{\mathrm{comb}}$, which removes the redundant information carried by $\tilde{\mathbf{V}}^{t}_{\mathrm{re}}$ and $\tilde{\mathbf{V}}^{t}_{\mathrm{im}}$, and obtain the minimal rank representation of $\tilde{\mathbf{V}}^{t}_{\mathrm{re}}$ and $\tilde{\mathbf{V}}^{t}_{\mathrm{im}}$ for a prescribed accuracy. This procedure generates an orthogonal cluster basis $\tilde{\mathbf{V}}_{\mathrm{comb}}$. $\tilde{\mathbf{V}}_{\mathrm{comb}}\tilde{\mathbf{V}}^{\mathrm{T}}_{\mathrm{comb}}\tilde{\mathbf{V}}^{t}_{\mathrm{re}}$ and $\tilde{\mathbf{V}}_{\mathrm{comb}}\tilde{\mathbf{V}}^{\mathrm{T}}_{\mathrm{comb}}\tilde{\mathbf{V}}^{t}_{\mathrm{im}}$ are the best approximation of $\tilde{\mathbf{V}}^{t}_{\mathrm{re}}$ and $\tilde{\mathbf{V}}^{t}_{\mathrm{im}}$ in the space of the new cluster bases, respectively. Next, we perform

$$\begin{aligned}&\tilde{\mathbf{V}}^{t}_{\mathrm{comb}}\tilde{\mathbf{V}}^{t^{\mathrm{T}}}_{\mathrm{comb}}\tilde{\mathbf{G}}^{t,s}_{\mathrm{new}}(\tilde{\mathbf{V}}^{s}_{\mathrm{comb}}\tilde{\mathbf{V}}^{s^{\mathrm{T}}}_{\mathrm{comb}})\\ &\qquad = \tilde{\mathbf{V}}^{t}_{\mathrm{comb}}\left(\tilde{\mathbf{V}}^{t^{\mathrm{T}}}_{\mathrm{comb}}\tilde{\mathbf{G}}^{t,s}_{\mathrm{new}}\tilde{\mathbf{V}}^{s}_{\mathrm{comb}}\right)\tilde{\mathbf{V}}^{s^{\mathrm{T}}}_{\mathrm{comb}} \qquad (37)\end{aligned}$$

to update the coupling matrices, where the original complex-valued unitary cluster basis $\tilde{\mathbf{V}}$ is replaced by the new real-valued orthogonal cluster basis $\tilde{\mathbf{V}}_{\mathrm{comb}}$.

### *F. Complexity and Accuracy Analysis*

Before analyzing the complexity, we summarize the basic steps of the proposed algorithm as follows.

Step 1) Prepare $\mathbf{B}^s = \mathbf{V}^{s^{\mathrm{T}}}\overline{\mathbf{V}}^s$ in (20) for all clusters $s$ in linear time by using the cluster basis product algorithm in [11, p. 12.]. Take Fig. 2 as an example, cluster $s$ refers to eight leaf clusters at the bottom level and seven nonleaf clusters at other levels.

Step 2) Compute $\overline{\mathbf{S}}^{t}_{\mathrm{sum}}$ for all clusters $t$ based on pseudo-code given in (21). Again, for the example shown in Fig. 2, there are 15 clusters for which we compute $\overline{\mathbf{S}}^{t}_{\mathrm{sum}}$.

Step 3) Compute $\mathbf{S}^{t}_{\mathrm{sum}}$ for all clusters $t$ based on pseudo-code given in (22).

Step 4) Compute new nested cluster bases for all clusters with rank minimized by accuracy, based on pseudo-code given in (26).

- For all leaf clusters $t$, compute $\mathbf{G}^{t}_{2}$ and perform the Schur decomposition of $\mathbf{G}^{t}_{2}$ for a given accuracy. The resulting new cluster basis is $\tilde{\mathbf{V}}^{t}$. This step corresponds to the section of code after "*else*" in (26). Take Fig. 2 as an example, this step is done for all leaf clusters $\{1\}, \{2\}, \ldots,$ and $\{8\}$.
- For all nonleaf clusters $t$, as shown in (24), we only need to compute the underlined part $\mathbf{G}^{t}_{2,\mathrm{proj}}$ and perform the Schur decomposition of $\mathbf{G}^{t}_{2,\mathrm{proj}}$. This step corresponds to the section of code after "*if t has children*" in (26).

Step 5) Compute new coupling matrices with rank minimized for accuracy based on the pseudo-code given in (30).

With the aforementioned steps, now we can analyze the complexity of the proposed algorithm step by step. Step 1) is computed for all clusters by using the cluster basis product algorithm in [11, p. 12]. This algorithm has linear complexity, as shown in [11].

In Step 2), we compute $\overline{\mathbf{S}}^{t}_{\mathrm{sum}}$ based on the pseudo-code given in (21). Since $\mathbf{B}^s$ is obtained from Step 1 already, the computation for each admissible block $b$ in (21) only involves the computation of $\mathbf{S}^{t,s}\mathbf{B}^{s}\mathbf{S}^{t,s^{\mathrm{H}}}$, which costs $O(k^3)$ operations, where $k$ denotes the rank of each block. This is because each $\mathbf{B}^s$ is of size $k$ by $k$, and so is the coupling matrix $\mathbf{S}^{t,s}$. For each inadmissible block, we compute $\mathbf{F}^{t,s}\mathbf{F}^{t,s^{\mathrm{H}}}$, as shown in (21), the cost of which is constant since the inadmissible block is of dimension *leafsize* by *leafsize*, and *leafsize* is a predetermined constant. Since there are $O(N)$ blocks in an $\mathcal{H}^2$-matrix, either admissible or inadmissible, and each block is associated with a constant computational cost in Step 2), the computational complexity of Step 2) is linear.

In Step 3), we compute $\mathbf{S}^{t}_{\mathrm{sum}}$ for all clusters $t$ based on the pseudo-code given in (22). This procedure is essentially a top-down traversal of the cluster tree. For each cluster $t$, we compute $\mathbf{E}^{ti}\overline{\mathbf{S}}^{t}_{\mathrm{sum}}\mathbf{E}^{ti^{\mathrm{H}}}$ for its one-level-down children, add it upon $\overline{\mathbf{S}}^{ti}_{\mathrm{sum}}$, and then go down to the next tree level. The same procedure is repeated until we reach leaf level. Since transfer matrix $\mathbf{E}^{ti}$ is of size $k$ by $k$, and so is $\overline{\mathbf{S}}^{t}_{\mathrm{sum}}$. In (22), the computation of $\mathbf{E}^{ti}\overline{\mathbf{S}}^{t}_{\mathrm{sum}}\mathbf{E}^{ti^{\mathrm{H}}}$ for each cluster $t$ in a cluster tree is $O(k^3)$. The total number of clusters in a cluster tree is $O(N)$, and each cluster is only associated with $O(k^3)$ operations in Step 3). Thus, the computational complexity of Step 3) is linear.

In Step 4), described in Section III-B, we compute the rank-minimized cluster bases based on (26). In (26), $\mathbf{S}^{t}_{\mathrm{sum}}$ is known from Step 3), and hence, we only need to compute $\mathbf{G}^{t}_{2,\mathrm{proj}}$, the size of which is $2k \times 2k$, and perform the Schur decomposition of $\mathbf{G}^{t}_{2,\mathrm{proj}}$ for all nonleaf clusters, while computing $\mathbf{G}^{t}_{2}$ of size $k$ by $k$ and perform the Schur decomposition of $\mathbf{G}^{t}_{2}$ for all leaf clusters. The computation of $\mathbf{G}^{t}_{2,\mathrm{proj}}$ or $\mathbf{G}^{t}_{2}$ for each cluster $t$ costs at most $O(k^3)$, and the computation of the Schur decomposition of $\mathbf{G}^{t}_{2,\mathrm{proj}}$ or $\mathbf{G}^{t}_{2}$ for each cluster $t$ also costs $O(k^3)$ operations. Since the total number of clusters in a cluster tree is $O(N)$, and each cluster is associated with $O(k^3)$ operations, the complexity of Step 4) is linear.

In Step 5), described in Section III-C, we update the coupling matrices based on (30). The update of each matrix block shown in (30) only requires $O(k^3)$ operations for each block since either $\tilde{\mathbf{B}}^t\mathbf{S}^{t,s}\tilde{\mathbf{B}}^{s^{\mathrm{T}}}$ for each admissible block or $\tilde{\mathbf{V}}^{t^{\mathrm{H}}}\mathbf{F}^{t,s}\overline{\tilde{\mathbf{V}}}^{s}$ for the inadmissible block costs just $O(k^3)$ operations since the total number of matrix blocks in an $\mathcal{H}^2$-matrix is $O(N)$. The total computational complexity of Step 5) is also linear.

By adding up the cost of all five steps, we obtain the total computation cost of the proposed method for constructing the new $\mathcal{H}^2$-representation, which is linear.

As for the accuracy of the proposed algorithm, among the five steps, Steps 1), 2), 3), and 5) involve no approximation. Step 4) is performed based on a prescribed accuracy. For example, if the prescribed accuracy is $\varepsilon$, the truncation of Schur decomposition in (26) is done to satisfy the prescribed accuracy $\varepsilon$ based on truncation error shown in (15). As a result, the proposed algorithm is error controllable.

To construct the new $\mathcal{H}^2$-representation of $\mathbf{G}$ in a vector-based IE formulation, which originally contains multiple submatrix blocks in each $(t, s)$, we use the procedure shown in Section III-D, which is to merge multiple subblocks into a rank-minimized single block. As shown in Section III-D, the merge operation is performed by using the proposed rank-minimization procedure described in Sections III-A–III-C. As a result, the procedure has the same linear complexity and the accuracy is controlled to satisfy the prescribed accuracy. In Section III-E, the complex-valued cluster bases are converted to real-valued cluster bases. The cost of conversion is the same as the cost of orthogonalization, which is linear, as shown in [11], without loss of accuracy.

## IV. Linear-Complexity Iterative and Direct IE Solvers Accelerated by the New $\mathcal{H}^2$-Representation

The proposed rank-minimized new $\mathcal{H}^2$-representation can be used to accelerate both iterative and direct solutions of the IE-based dense system of equations.

### *A. Linear- Complexity Inverse-Based Direct Solver*

After the rank-minimized $\mathcal{H}^2$ representation of $\mathbf{G}$ is constructed, we compute its inverse $\mathbf{G}^{-1}$ by using the linear-complexity inverse algorithm developed in [14]. It is worth mentioning that although a matrix inverse and a matrix–matrix multiplication [11] share the same number of block matrix multiplications, there is a major difference that prevents one from using a fast matrix–matrix multiplication algorithm to achieve a linear complexity in inverse. The major difference is that in the level-by-level computation of the inverse, at each level, the computation is performed based on updated matrix blocks obtained from the computation at the previous level instead of the original matrix. In contrast, in the level-by-level computation of the matrix–matrix multiplication, at each level, the computation is performed based on the original matrix, which is never updated. This difference would render the inverse complexity higher than linear if one does not address it properly. In [14], we develop three new algorithms to render the total cost of an inverse linear. The first algorithm is an instantaneous collect operation for generating the auxiliary admissible block forms; the second algorithm is a modified block matrix multiplication algorithm; and the third one is an instantaneous split operation. The accuracy of the aforementioned linear-complexity inverse has been both theoretically proved and numerically verified in [14].

### *B. Linear-Complexity Preconditioned Iterative Solver*

With the new $\mathcal{H}^2$-representation, the memory and computational cost of an $\mathcal{H}^2$-based iterative solver can also be significantly reduced. In addition, in the framework of the proposed method, it is very convenient to construct an effective preconditioner for accelerating the iterative solution. We can set a lower order of accuracy and use the resultant less-accurate cluster basis $\tilde{\mathbf{V}}^P$ to build an approximate $\mathcal{H}^2$-repesentation. This representation serves as an effective preconditoner of the original system matrix. Its inverse can be obtained by the linear-complexity $\mathcal{H}^2$-inverse algorithm [14]. Since the rank of $\tilde{\mathbf{V}}^P$ used for the preconditioner is minimized based on a lower order accuracy, the inverse of the preconditioner can be efficiently obtained. In addition, the effectiveness of the preconditioner can be adaptively controlled by the accuracy prescribed for constructing the approximate cluster bases $\tilde{\mathbf{V}}^P$. Certainly, when the accuracy setting is not low, the resultant inverse is a good representation of the inverse of the original matrix. The solver then becomes the direct solver described in Section IV-A.

## V. Numerical Results

In this section, we first examine the effectiveness of the proposed new $\mathcal{H}^2$-repesentation in minimizing rank for given accuracy for both capacitance extraction and full-wave impedance extraction of interconnects. We then proceed to examine the performance of both the linear-complexity iterative solver and the linear-complexity direct solver with the proposed new $\mathcal{H}^2$-repesentation by extracting capacitances of a number of interconnects from a small number of unknowns to 3.71 million unknowns involving up to 576 conductors. The computer used has an 8222 SE AMD Opteron processor running at 3 GHz.

### *A. Effectiveness of the Proposed New $\mathcal{H}^2$-Representation in Reducing Rank for Prescribed Accuracy*

The first example is a package interconnect [20] composed of $m$ lossy conductors embedded in a uniform dielectric, as shown in Fig. 3. The $m$ is chosen as 4, 8, 16, and 32, respectively. The simulation parameters are leafsize $= 20$ and $\eta = 2$. The dense system of equations for extracting impedances is formulated based on [16]. In the proposed new $\mathcal{H}^2$-representation, $\varepsilon = 10^{-4}$ is used in Schur decomposition. In the interpolation-based $\mathcal{H}^2$-representation, $p = 3$ is used.

In Fig. 4(a), we compare the original rank determined by the interpolation method and the rank determined by the proposed method with respect to $N$. Since the rank may not be the same in each matrix block, we measure the rank by using the average rank defined as $\sqrt{\sum_{i=1}^{nk} k_i^2/nk}$, where $k_i$ is the rank of the $i$th admissible block, $nk$ is the number of admissible blocks, and the summation represents the total storage of the admissible blocks. In Fig. 4(b), we plot the accuracy of the proposed rank-minimized $\mathcal{H}^2$-representation with respect to $N$. The matrix accuracy is measured by $\|\mathbf{G} - \tilde{\mathbf{G}}\|_{\mathrm{F}}/\|\mathbf{G}\|_{\mathrm{F}}$, where $\tilde{\mathbf{G}}$ is the new

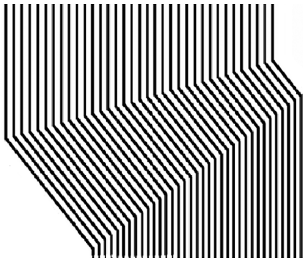
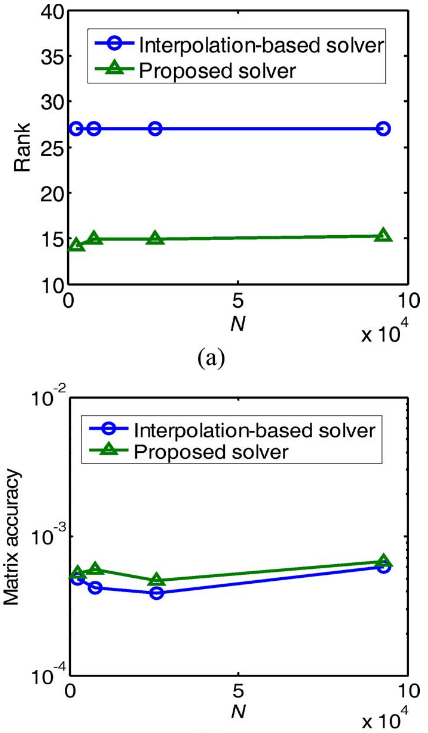


Fig. 3. Package interconnect composed of $m$-lossy conductors.



(a)

(b)

Fig. 4. Performance of the proposed method for analyzing a package interconnect structure. (a) Rank. (b) Matrix accuracy.

$\mathcal{H}^2$-representation, and $\mathbf{G}$ is the original full matrix. For comparison, the accuracy of the interpolation-based $\mathcal{H}^2$-representation is also plotted. As can be seen from Fig. 4(a) and (b), the proposed method significantly reduces the rank without sacrificing the accuracy.

The second example is a four-layer bus structure in a uniform dielectric, as shown in Fig. 5. In each layer, there are $m$ conductors, and each conductor has a dimension of $1 \times 1 \times (2m + 1)$ m$^3$. We simulate a suite of such structures with $m$ chosen as 5, 10, 20, and 40, respectively. The parameters used in the simulations are *leafsize* $= 20$, $\eta = 2.5$, $\varepsilon = 10^{-4}$ for Schur decomposition in the proposed method, while $p = 4$ in the interpolation method.

In Fig. 6(a), we compare the original rank determined by the interpolation method and the rank determined by the proposed method with respect to $N$. In Fig. 6(b), we plot the accuracy of the proposed rank-minimized $\mathcal{H}^2$-representation with respect to $N$. As can be seen from Fig. 6(a) and (b), the rank is again significantly reduced without sacrificing the accuracy.

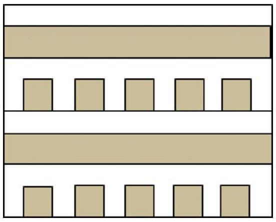
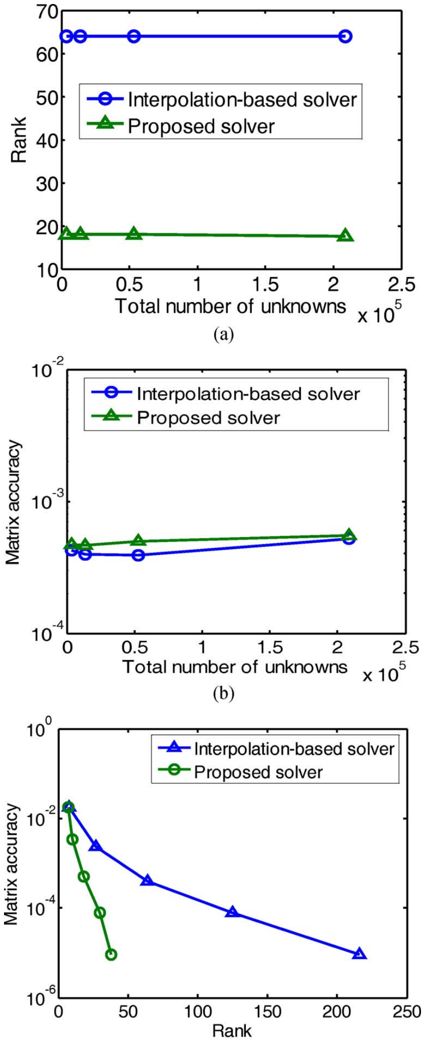


Fig. 5. Four-layer bus structure in a uniform dielectric with $m$ (the number of conductors per layer) ranging from 5 to 40.



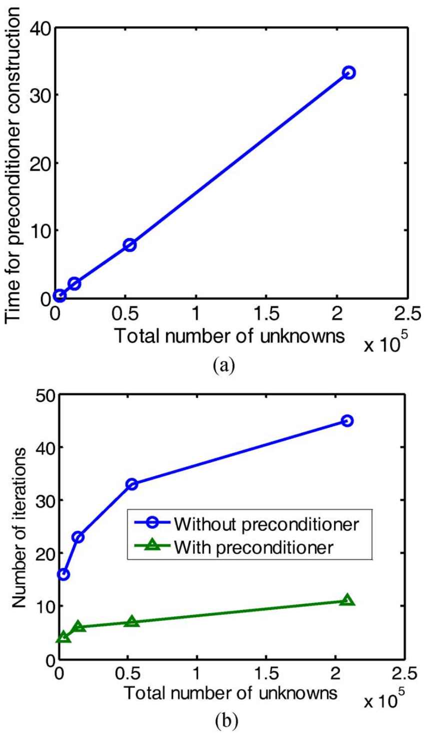
(a)

(b)

(c)

Fig. 6. Performance of the proposed rank-minimization method for analyzing a suite of four-layer bus structures from $m = 4$ to $m = 40$ in a uniform dielectric. (a) Rank versus $N$. (b) Matrix accuracy versus $N$. (c) Rank obtained for different accuracy for $m = 20$ case with $N = 53120$.

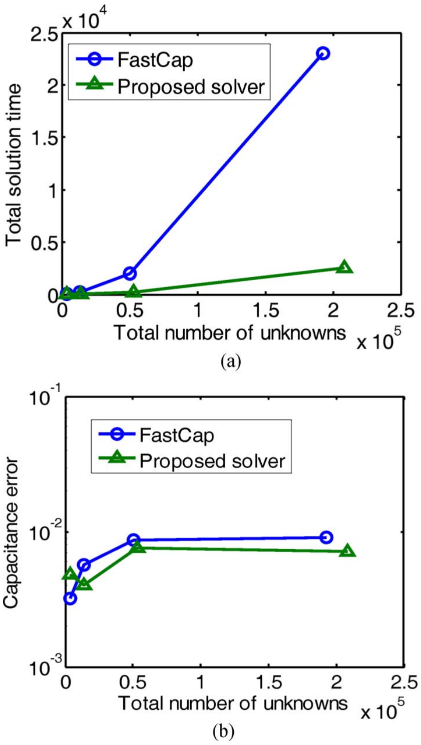
The proposed method for generating a rank-minimized $\mathcal{H}^2$-representation is error controllable, and also for any given accuracy, the rank of the proposed $\mathcal{H}^2$-representation is smaller than that of the interpolation-based $\mathcal{H}^2$-representation. To demonstrate this point, we re-simulate one of the above four-layer bus structures (the one with $m = 20$) under various accuracy settings. In Fig. 6(c), we plot the matrix accuracy with respect to rank from the proposed method in comparison with the same generated by the interpolation-based method. It is evident that first, the error of the proposed method is controllable; second, for the same accuracy, the rank of the proposed new $\mathcal{H}^2$-representation is much smaller than that of the interpolation-based $\mathcal{H}^2$-representation.



(a)

(b)

Fig. 7. Performance of the proposed preconditioner for analyzing a four-layer bus structure in a uniform dielectric. (a) Construction time of the proposed preconditioner (seconds). (b) Number of iterations.



(a)

(b)

Fig. 8. Performance of the proposed preconditioned iterative solver. (a) Total solution time (seconds). (b) Capacitance error.

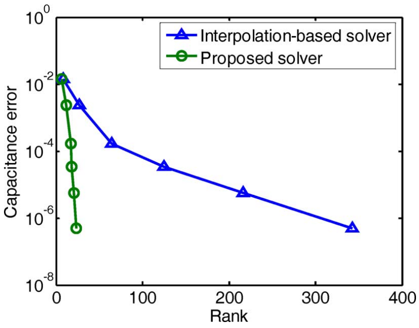


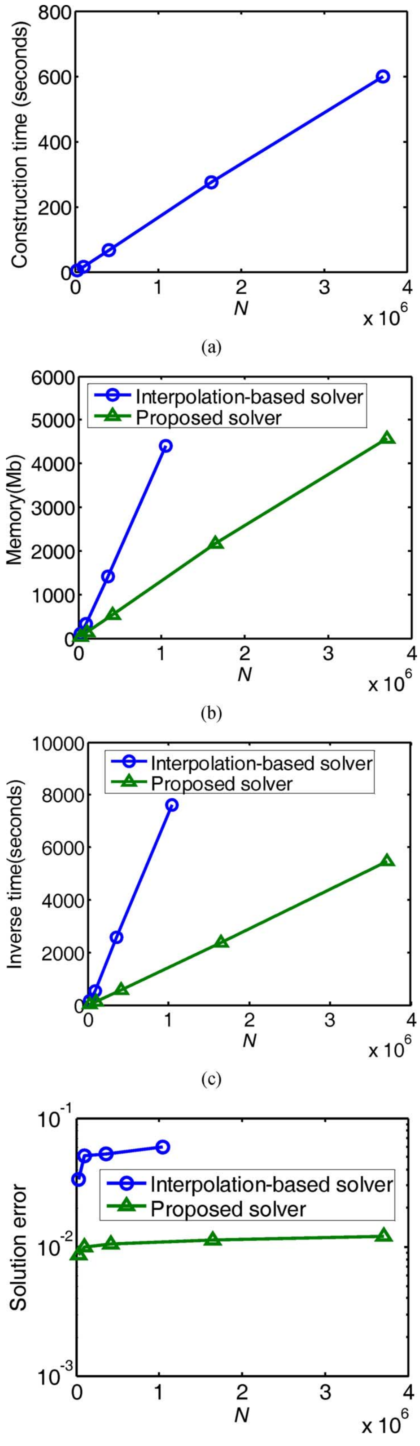
Fig. 9. Capacitance error versus rank of the proposed method in comparison with that of the interpolation-based solver.

## *B. Linear-Complexity Preconditioned Iterative Solver With the Proposed New $\mathcal{H}^2$-Representation*

Next, we show the performance of the proposed preconditioned iterative solution for analyzing the suite of four-layer bus structures shown in Fig. 5. We use the GMRES method with the proposed preconditioner to iteratively solve the system matrix. The convergence criterion of the GMRES iteration is set as $10^{-3}$. The accuracy requirement is chosen as $10^{-2}$ for constructing orthogonal cluster bases $\tilde{\mathbf{V}}^P$ described in Section IV-B. In Fig. 7(a), we plot the CPU time for the construction of the proposed preconditioner. A clear linear scaling can be observed. Once the proposed preconditioner is constructed, it can be used for all right-hand sides. The number of iterations used in the proposed preconditioned iterative method is shown in Fig. 7(b). For comparison, the number of iterations required by the GMRES method without using the proposed preconditioner is also plotted. As can be seen clearly from Fig. 7(b), the number of iterations is significantly reduced by using the proposed preconditioner. It can be further reduced by setting a higher order of accuracy for constructing orthogonal sparse cluster bases $\tilde{\mathbf{V}}^P$.

In Fig. 8(a), we plot the total solution time of the proposed preconditioned iterative solver including the time of constructing the new $\mathcal{H}^2$-representation, constructing the preconditioner, and iteratively solving the matrix system for all conductors. A clear linear scaling can be observed. We also simulate the same bus structure using FastCap2.0, which is a state-of-the-art preconditioned iterative capacitance solver available in the public domain. The expansion order of FastCap2.0 is chosen as 2, and the convergence tolerance is set as $10^{-3}$. A similar number of unknowns is used for comparison with FastCap2.0. As can be seen from Fig. 8, the proposed iterative solver is much faster without sacrificing accuracy. The capacitance error shown in Fig. 8(b) is measured by $\|\mathbf{C} - \mathbf{C}'\|_{\mathrm{F}}/\|\mathbf{C}\|_{\mathrm{F}}$, where $\mathbf{C}$ is the capacitance matrix obtained from FastCap2.0 with expansion order $= 3$, and $\mathbf{C}'$ is that generated by the proposed iterative solver or by FastCap2.0 with expansion order $= 2$. Excellent accuracy can be observed in both solvers.

To demonstrate the fact that the proposed solver is error controllable and its rank is much reduced for achieving a prescribed accuracy as compared to an interpolation-based $\mathcal{H}^2$-solver, we select one of the four-layer bus structures simulated in the above, whose $m$ is 5. We compute the capacitance matrix error in reference to that generated from a *full-matrix*-based computation that does not involve any approximation. Full-matrix computation is used here as the reference because as a fast solver, FastCap, utilizes approximations. In Fig. 9, we plot the capacitance matrix error of the proposed method with respect



(a)

(b)

(c)

(d)

Fig. 10. Performance of the proposed direct solver. (a) Construction time of the new $\mathcal{H}^2$-representation. (b) Memory. (c) Inverse time. (d) Solution error ($p = 1$ used in the interpolation-based solver).

to the rank. It is evident that the capacitance error can be well controlled by the rank used in the proposed method, which is, in turn, controlled by $\varepsilon$ in (15), which is one input used in pseudo-code (26). In Fig. 9, we also plot the capacitance matrix error of the interpolation-based solver versus rank. It is clear that to achieve the same capacitance accuracy, the rank required by the proposed method is much smaller than that of the interpolation-based $\mathcal{H}^2$-solver.

### C. Linear-Complexity Direct Solver With the Proposed New $\mathcal{H}^2$-Representation

Next, we show the performance of the proposed linear-complexity direct solver with the new $\mathcal{H}^2$-representation. The example considered is a multilayer 3-D on-chip interconnect structure [6], [14]. The relative permittivity of the interconnect structure is 3.9 in M1, 2.5 from M2 to M6, and 7.0 from M7 to M8. The structure involves 48 conductors. To test the large-scale modeling capability of the proposed direct solver, the 48-conductor structure is duplicated horizontally, resulting in 576 conductors, the discretization of which leads to 3.71 million unknowns. The parameters used in the proposed method are $leafsize = 20$, $\eta = 2$, and $\varepsilon = 10^{-4}$ for Schur decomposition, with $p = 3$ used in the interpolation-based $\mathcal{H}^2$-representation. Fig. 10(a)–(c) shows the new $\mathcal{H}^2$-representation construction time, memory consumption, and inverse time, respectively. A clear linear scaling can be observed in all the figures. In Fig. 10(d), we plot the solution accuracy measured by relative residual. Excellent accuracy can be observed. In addition, in Fig. 10(b)–(d), we overlay the cost and accuracy of the proposed direct solver with those of the linear-complexity direct solver in [13] and [14] that uses an interpolation-based $\mathcal{H}^2$-representation with $p = 1$. The advantage of the proposed direct solver can be clearly seen. The proposed solver achieves higher order accuracy with even less computational cost.

## VI. Conclusions

In this paper, we have developed a method to generate a new $\mathcal{H}^2$-matrix-based representation with its rank minimized for a given $\mathcal{H}^2$-partition and a given accuracy for an IE-based analysis of large scale 3-D interconnects. Such a representation features a minimized rank in both nested cluster bases and coupling matrices with prescribed accuracy satisfied. The new $\mathcal{H}^2$-representation is constructed in linear complexity, and hence, the computational overhead is small. It is applicable to both real and complex-valued dense system matrices generated from scalar- and/or vector-based IE formulations.

The proposed new rank-minimized $\mathcal{H}^2$-representation can be used to accelerate both iterative and direct solutions of IE-based dense systems of equations. To demonstrate this point, both $O(N)$ complexity direct and preconditioned iterative solvers are developed with the proposed new $\mathcal{H}^2$-representation for 3-D capacitance extraction. They are shown to outperform state-of-the-art linear-complexity capacitance solvers in both memory and CPU consumption. As for the comparison between the proposed linear-complexity direct solver and the proposed linear-complexity preconditioned iterative solver, the former is clearly the solver of choice when the number of right-hand sides is large. The proposed algorithm for constructing a rank-minimized $\mathcal{H}^2$-representation, which compresses all the off-diagonal matrix blocks to have a minimal rank based on accuracy, can also be applied to accelerate the solution of full-wave and electrically large IEs, which will be explored in the future.

## REFERENCES

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

[2] J. R. Phillips and J. White, "A precorrected FFT method for capacitance extraction of complicated 3-D structures," in *Proc. ICCAD*, 1994, pp. 268–271.

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

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

[5] Z. Zhu, B. Song, and J. White, "Algorithms in FastImp: A fast and wideband impedance extraction program for complicated 3-D geometries," *Design Autom. Conf.* , pp. 712–717, 2003.

[6] S. Yan, V. Saren, and W. Shi, "Sparse transformations and preconditioners for hierarchical 3-D capacitance extraction with multiple dielectrics," *Design Autom. Conf.* , pp. 788–793, 2004.

[7] S. Borm, L. Grasedyck, and W. Hackbusch, "Hierarchical matrices," Max Planck Inst. Math. Sci., Leipzig, Germany, Lecture Note 21, 2003.

[8] W. Hackbusch, B. Khoromskij, and S. Sauter, "On $\mathcal{H}^2$–matrices," Lecture Appl. Math. H. Bungartz, R. Hoppe, and C. Zenger, Eds., 2000, pp. 9–29.

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

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

[11] S. Börm, "$\mathcal{H}^2$-matrix arithmetics in linear complexity," Max Planck Inst. Math., Lecture Note, 2005.

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

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

[14] 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.

[15] 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.

[16] 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 (Special Issue)*, vol. 101, no. 2, pp. 372–388, Feb. 2013.

[17] S. Börm and W. Hackbusch, "Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices," *Computing*, vol. 69, pp. 1–35, 2002.

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

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

[20] H. Braunisch, "Guided waves in high-performance microprocessor packaging," in *URSI Radio Sci. Meeting Conf. Dig.*, Ottawa, ON, Canada, Jul. 2007.

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

[22] W. Chai and D. Jiao, "A linear-complexity direct integral equation solver accelerated by a new rank-minimized $\mathcal{H}^2$-representation for large-scale 3-D interconnect extraction," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2012, 3 pp.

[23] C. F. Van Loan and G. H. Golub, *Matrix Computations*. London, U.K.: The Johns Hopkins Univ. Press, 1996.

[24] W. Chai and D. Jiao, "A theoretical study on the rank of integral operators for broadband electromagnetic modeling from static to electrodynamic frequencies," *IEEE Trans. Compon., Packag., Manuf. Technol.*, 2013, (see also Purdue ECE Tech. Rep. [Online]. Available: http://docs.lib.purdue.edu/ecetr/436/), accepted for publication.

**Wenwen Chai** (S'10) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2004, the M.S. degree in electrical engineering from the Chinese Academy of Sciences, Beijing, China, in 2007, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2012.

From 2007 to 2012, she was a Research Assistant with the On-Chip Electromagnetics Group, Purdue University. In 2012, she joined the PrimeRail Group, Synopsys Inc., Mountain View, CA, USA, as a Senior Research and Development Engineer. Her research interests include computational electromagnetics, high-performance very large scale integration (VLSI) computer-aided design (CAD), and fast and high-capacity numerical methods.

Dr. Chai was the recipient of the IEEE Antennas and Propagation Society Doctoral Research Award (2009–2010) and the Synopsys IG Division Individual Recognition Award for her contribution to the performance improvement of the large-scale power grid analysis tool (2012).

**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.

She then joined the Technology Computer-Aided Design (CAD) Division, Intel Corporation, until September 2005, where she was a Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. 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 180 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 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 publications 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 the 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 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 Intel Hero Award, the Intel Corporation's LTD Team Quality Award, and the 2000 Raj Mittra Outstanding Research Award of the University of Illinois at Urbana-Champaign.