Fast Iterative Solution Algorithms in the Frequency-Domain Layered Finite Element Method for Analyzing Integrated Circuits

Feng Sheng, Houle Gan, *Student Member, IEEE*, and Dan Jiao, *Senior Member, IEEE*

***Abstract*—Fast algorithms are developed in this work for solving the system matrix resulting from a frequency-domain layered finite element based analysis of integrated circuits. The frequency-domain layered finite element method represents a 3-D layered system by a 2-D layered system, and further by a single-layered one. The reduced system matrix is generally denser than the original sparse matrix. In this paper, we show that 1) the dense matrix–vector multiplication can be performed in linear complexity; in addition, the reduction cost can be bypassed, 2) an effective preconditioner can be developed to converge the iterative solution of the reduced system matrix in a small number of iterations, and 3) the preconditioner can be solved in linear complexity. As a result, the reduced system matrix can be solved efficiently. The algorithms are rigorous without making any approximation. They apply to any arbitrarily-shaped multilayer structure. Numerical results demonstrated the accuracy, effectiveness, and efficiency of the proposed algorithms in analyzing on-chip circuits.**

***Index Terms*—Electromagnetic modeling, finite element method, frequency domain, iterative solver, on chip, preconditioner.**

I. Introduction

The move towards integrating analog, mixed-signal, and radio-frequency (RF) circuitry on a single chip has made the analysis and design of microelectronic systems increasingly challenging. To sustain the continued scaling and integration of integrated circuits, there is a critical need for the electromagnetically accurate modeling and simulation of the integrated circuits. However, there are many modeling challenges associated with on-chip circuits. These challenges include large problem size, a large number of nonuniform dielectric stacks with strong nonuniformity, a large number of nonideal conductors, the presence of silicon substrate, highly-skewed aspect ratios, etc. The importance and challenges of electromagnetic-based analysis of integrated circuit (IC) problems have stimulated new developments in computational electromagnetic methods [1]–[11]. In [6], a frequency-domain layered finite element method (LFEM) was developed for electromagnetic analysis of large-scale high-frequency integrated circuits. In this method, first, the system matrix of the original 3-D problem is reduced to that of 2-D layers. Second, the system matrix of 2-D layers is reduced to that of a single layer. The cost of reduction scales linearly with single-layer computational complexity. In a realistic on-chip structure, one can encounter a large number of layers when growing the layer along either x or z direction, assuming y is the stack-growth direction. Hence, the number of single-layer unknowns is generally much less than that of total unknowns. Take a typical on-chip interconnect as an example, the former is 2270, whereas the latter is 3.04 million. Therefore, the LFEM method is capable of handling large-scale multilayer structures. Designers can deal with a much smaller system produced by the LFEM to perform design optimization.

The layered finite element method can be viewed as a fast Gaussian elimination procedure which fully takes advantage of the layered property present in today's integrated circuits and package structures. The underlying mesh used in the layered finite element method is a triangular-prism-element based mesh, which allows for irregular structures in the transverse plane, and meanwhile capturing the layered geometry along the third dimension. For layered structures, a triangular-prism-element based mesh is indeed natural for choice. In the layered finite-element method, a number of important matrix properties associated with a prism-element-based mesh are identified or constructed to speed up the process of Gaussian elimination.

The efficiency of the layered finite element method hinges upon two factors: 1) how efficiently the unknowns can be reduced and 2) how efficiently the reduced system matrix can be solved. In [7] and [8], we developed fast reduction algorithms to efficiently reduce the system matrix. In this work, we present fast algorithms for solving the reduced system matrix efficiently.

This paper is organized in the following manner. In Section II, an overview of the LFEM and the problem statement are given. In Section III, the proposed fast solution algorithms are described. In Section IV, numerical results are presented to validate the proposed algorithms. In Section V, conclusions are drawn.

II. Overview of the LFEM and Problem Statement

Consider 3-D circuit problems. The circuit can be a global on-chip interconnect network, a package, and a mixed-signal IC circuit. Generally, these circuits are multilayered structures. They are embedded in a multilayer dielectric medium backed

Manuscript received March 21, 2009; revised July 09, 2009. First published October 13, 2009; current version published May 05, 2010. This work was supported in part by the Office of Naval Research under Grant N00014-06-1-0716, in part by a grant from Intel Corporation, and in part by the National Science Foundation under Grant 0747578. This work was recommended for publication by Associate Editor J. Tan upon evaluation of the reviewers comments.

The authors are 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/TADVP.2009.2029561

by silicon, GaAs, InP or other substrates. A Manhattan-type on-chip circuit is even layered in any of x, y, and z directions. Inside these circuits, the electric field $\mathbf{E}$ satisfies the second-order vector wave equation

$$\nabla \times \left[\mu_r^{-1} \nabla \times \mathbf{E}\right] - k_0^2 \bar{\varepsilon}_r \mathbf{E} = -jk_0 Z_0 \mathbf{J} \quad \text{in } V \tag{1}$$

subject to certain boundary conditions. A finite-element solution of (1) and its boundary condition results in the following matrix equation:

$$\mathbf{K}u = b \tag{2}$$

in which $\mathbf{K}$ and b are assembled from their elemental counterparts:

$$\begin{aligned} \mathbf{K}^e =& \mu_r^{-1} \langle \nabla \times \mathbf{N}_i, \nabla \times \mathbf{N}_j \rangle_V - k_0^2 \bar{\varepsilon}_r \langle \mathbf{N}_i, \mathbf{N}_j \rangle_V \\ & + jk_0 \langle \hat{n} \times \mathbf{N}_i, \hat{n} \times \mathbf{N}_j \rangle_{S_o} \\ b^e =& -jk_0 Z_0 \langle \mathbf{N}_i, \mathbf{J} \rangle_V \end{aligned} \tag{3}$$

where $\bar{\varepsilon}_r = \varepsilon_r(1 + (\sigma / j\omega\varepsilon_r\varepsilon_0))$ is the complex relative permittivity, and $\mathbf{N}$ is the vector basis function used to expand $\mathbf{E}$. In the LFEM method, $\mathbf{N}$ is chosen as triangular prism vector bases [6, eq. (5)]. In deriving (3), the first-order absorbing boundary condition is assumed for simplicity.

In a layered finite-element based solution of (2), the unknowns are ordered layer by layer, resulting in a 3-D layered system matrix. In Fig. 1(a), we show the system matrix $\mathbf{K}$ in a single layer, where N_{1s} denotes the surface unknowns on the upper surface of layer 1, N_{1V} denotes the volume unknowns in layer 1, and N_{2s} denotes the surface unknowns on the upper surface of layer 2. Matrix $\mathbf{A}$ is formed between unknowns N_{1s} and N_{1s}; $\mathbf{B}$ is formed between unknowns N_{1V} and N_{1V}; and $\mathbf{C}$ is formed between unknowns N_{2s} and N_{2s}. The matrix elements of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{E}$, and $\mathbf{F}$ are given by

$$\begin{aligned} \mathbf{A}_{ij}^e =& \mu_r^{-1} \left[\frac{l}{3} \langle \nabla \times \mathbf{W}_i, \nabla \times \mathbf{W}_j \rangle_\Omega + \frac{1}{l} \langle \mathbf{W}_i, \mathbf{W}_j \rangle_\Omega \right] \\ & - k_0^2 \bar{\varepsilon}_r \frac{l}{3} \langle \mathbf{W}_i, \mathbf{W}_j \rangle_\Omega \\ \mathbf{B}_{ij}^e =& -k_0^2 \bar{\varepsilon}_r l \langle \xi_i, \xi_j \rangle_\Omega + \mu_r^{-1} l \langle \nabla \xi_i, \nabla \xi_j \rangle_\Omega, \quad \mathbf{C}_{ij}^e = \mathbf{A}_{ij}^e, \\ \mathbf{D}_{ij}^e =& \mu_r^{-1} \langle \mathbf{W}_i, \nabla \xi_j \rangle_\Omega \\ \mathbf{E}_{ij}^e =& \mu_r^{-1} \left[\frac{l}{6} \langle \nabla \times \mathbf{W}_i, \nabla \times \mathbf{W}_j \rangle_\Omega - \frac{1}{l} \langle \mathbf{W}_i, \mathbf{W}_j \rangle_\Omega \right] \\ & - k_0^2 \bar{\varepsilon}_r \frac{l}{6} \langle \mathbf{W}_i, \mathbf{W}_j \rangle_\Omega, \quad \mathbf{F}_{ij}^e = -\left(\mathbf{D}_{ij}^e\right)^T \end{aligned} \tag{4}$$

in which $\mathbf{W}$ is the edge basis function [12, pp. 234–237], ξ is the node basis function [12, p. 80], Ω denotes the support of a triangular element, and l is the height of a prism element, which is the layer thickness.

The 3-D layered system matrix is then reduced to a 2-D layered one. This is equivalent to eliminating the volume unknowns, the procedure of which is shown in Fig. 1. The relationship between the reduced matrices and the original matrices can be written as

$$\begin{aligned} \mathbf{A_r} =& \mathbf{A} - \mathbf{DB^{-1}D^T} \qquad \mathbf{B_r} = \mathbf{E} - \mathbf{DB^{-1}F} \\ \mathbf{C_r} =& \mathbf{B_r^T} \qquad \mathbf{D_r} = \mathbf{C} - \mathbf{F^T B^{-1} F}. \end{aligned} \tag{5}$$

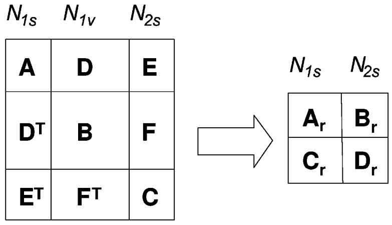


Fig. 1. Illustration of volume unknown elimination. (a) System matrix $\mathbf{K}$ in a single layer. (b) Surface unknown based system $\mathbf{K}_s$.

Due to the matrix properties such as $\mathbf{F} = -\mathbf{D^T}$ and $\mathbf{A} = \mathbf{C}$ shown in [6], (5) becomes

$$\begin{aligned} \mathbf{A_r} =& \mathbf{A} - \mathbf{DB^{-1}D^T} \qquad \mathbf{B_r} = \mathbf{E} + \mathbf{DB^{-1}D^T} \\ \mathbf{C_r} =& \mathbf{B_r^T} \qquad \mathbf{D_r} = \mathbf{A} - \mathbf{DB^{-1}D^T}. \end{aligned} \tag{6}$$

Fast reduction algorithms have been developed in [7] and [8] to perform the computation in (6) in an optimal computational complexity. The 2-D layered system can be further reduced to a single-layer system. The remaining task is to solve the reduced system efficiently. However, due to the reduction process, the reduced system matrix formed by surface unknowns becomes much denser than the original sparse matrix. Hence, solving it can be computationally intensive when the matrix size is large. In the following, we propose fast iterative solution algorithms to solve the reduced system efficiently.

III. Fast Solution Algorithms

In this section, we first present fast algorithms for solving the 2-D layered system matrix, we then give fast algorithms for solving the single-layered system. We also show how the computational cost of the reduction from a 3-D layered system to a 2-D layered system, and to a single-layered system can be bypassed.

A. Fast Solution of the 2-D Layered System Matrix

The reduced 2-D layered system of equations can be written as

$$\mathbf{K}_s u_s = b_s \tag{7}$$

where u_s denotes surface unknowns in all layers, b_s is the corresponding right-hand side, and $\mathbf{K}_\mathrm{s}$ is the reduced system matrix shown in Fig. 2, where $\mathbf{A}_\mathrm{ri}$, $\mathbf{B}_\mathrm{ri}$, $\mathbf{C}_\mathrm{ri}$, and $\mathbf{D}_\mathrm{ri}$ are $\mathbf{A}_\mathrm{r}$, $\mathbf{B}_\mathrm{r}$, $\mathbf{C}_\mathrm{r}$, and $\mathbf{D}_\mathrm{r}$ in the ith layer, the expressions of which are given in (6).

Equation (7) can be solved iteratively using the method of generalized minimal residual (GMRES) [13]. To solve a system of equations $\mathbf{G}\,x = b$, an m-step GMRES process generates an orthonormal basis $\{v_j\}_{j=1}^m$ of the Krylov subspace $\kappa_m(v_1, \mathbf{G})$ of a reduced dimension m spanned by $v_1, \mathbf{G}v_1, \cdots, \mathbf{G}^{m-1}v_1$, where v_1 is an initial unit norm vector. The projected matrix of $\mathbf{G}$ onto $\kappa_m(v_1, \mathbf{G})$ is represented by an $m \times m$ upper Hessenberg matrix $\mathbf{H}_m$. A least square problem is then solved to

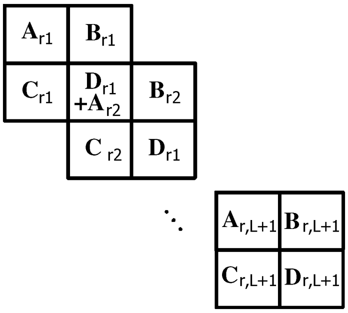


Fig. 2. Surface-unknown-based 2-D layered system $\mathbf{K}_s$.

minimize the norm of the residual $r_m = b - \mathbf{G}x_m$. The convergence of a GMRES procedure is theoretically guaranteed when the number of iterations m is the same as the dimension of $\mathbf{G}$.

A representative restarted GMRES procedure for solving (7) is given below:

```
$x_0 = arbitrary; \mathbf{P}r_0 = b - \mathbf{K}_s x_0; \beta = \|r_0\|_2; v_1 = \frac{r_0}{\beta};$
$for\ j = 1\ to\ m$ ($m$ is the internal iteration number)
  $\mathbf{P}w_j = \mathbf{K}_s v_j;$   (8.1)
    $for\ i = 1\ to\ j$
    $h_{i,j} = v_i^T w_j;$
    $w_j = w_j - h_{i,j} v_i;$
  $end$
  $h_{j+1,j} = \|w_j\|_2$
  $v_{j+1} = \frac{w_j}{h_{j+1,j}}$
$end$
$compute\ y_m\ that\ minimizes\ \|\beta e_1 - \tilde{\mathbf{H}}_m y_m\|_2;$
    $x_m = x_0 + v_m y_m$
$if\ not\ convergent,$
  $restart\ the\ above\ procedure\ with\ x_0 = x_m$
(The number of restarted times is the external
    iteration number.)                                   (8)
```

where $\mathbf{P}$ is a preconditioner, x_0 is the initial guess, m denotes the internal iteration number, and the number of times to restart the GMRES is called as the external iteration number. Clearly, the efficient computation of (8) relies on an efficient matrix–vector multiplication $\mathbf{K}_s v_j$ and an effective preconditioner $\mathbf{P}$ that can minimize the iteration number m. In addition, the solution of $\mathbf{P}$ needs to be computationally efficient. In the following, we show three fast algorithms that can fulfill the aforementioned three requirements.

1) Linear-Complexity Matrix–Vector Multiplication: The optimal complexity of computing $\mathbf{K}_s v$ is $O(N)$ complexity (Note that the ratio of N_s to N is approximately 0.75 in a triangular prism element based mesh). However, as shown in Fig. 2, $\mathbf{K}_\mathrm{s}$ is dense because it is made of $\mathbf{A}_\mathrm{r}$, $\mathbf{B}_\mathrm{r}$, $\mathbf{C}_\mathrm{r}$, and $\mathbf{D}_\mathrm{r}$ in each layer, and each of $\mathbf{A}_\mathrm{r}$, $\mathbf{B}_\mathrm{r}$, $\mathbf{C}_\mathrm{r}$, and $\mathbf{D}_\mathrm{r}$ is dense due to the elimination of volume unknowns. Apparently, it is not feasible to perform $\mathbf{K}_s v$ in $O(N)$ complexity. This difficulty can be overcome by the following approach. As shown in (6), $\mathbf{A_r}$ is $\mathbf{A} - \mathbf{DB^{-1}D^T}$, $\mathbf{B_r}$ is $\mathbf{E} + \mathbf{DB^{-1}D^T}$, $\mathbf{C_r}$ is $\mathbf{B_r^T}$, and $\mathbf{D_r}$ is $\mathbf{A} - \mathbf{DB^{-1}D^T}$. Therefore, instead of forming $\mathbf{A}_\mathrm{r}$, $\mathbf{B}_\mathrm{r}$, $\mathbf{C}_\mathrm{r}$, and $\mathbf{D}_\mathrm{r}$, and multiplying them by corresponding vectors, we can multiply $\mathbf{A} - \mathbf{DB^{-1}D^T}$ and $\mathbf{E} + \mathbf{DB^{-1}D^T}$ by corresponding vectors. Since $\mathbf{D}$, $\mathbf{A}$, and $\mathbf{E}$ are all sparse, multiplying $\mathbf{D}$, $\mathbf{A}$, and $\mathbf{E}$ by any vector can be performed in linear complexity. In addition, multiplying $\mathbf{B^{-1}}$ by any vector can be performed in linear complexity by employing the fast technique we developed in [7] and [8]. Basically, $\mathbf{B}$ is structured to be a block tridiagonal matrix. It is then split into $\mathbf{B} = \mathbf{D} - \mathbf{L} - \mathbf{U}$, with $\mathbf{D}$ being the diagonal block, $\mathbf{L}$ and $\mathbf{U}$ lower and upper off-diagonal blocks respectively. Due to the matrix property of $\mathbf{B}$, we showed in [7] and [8] that $\mathbf{D}$ can be used as an effective preconditioner to converge the iterative solution of $\mathbf{B}$ in a few iterations. As a result, the entire matrix–vector multiplication $\mathbf{K}_s v$ can be carried out in linear complexity. In addition, we bypass the need of computing $\mathbf{A}_\mathrm{r}$, $\mathbf{B}_\mathrm{r}$, $\mathbf{C}_\mathrm{r}$, and $\mathbf{D}_\mathrm{r}$ in each layer, i.e., *the reduction cost is avoided.*

2) Effective Preconditioner: With the efficient matrix–vector multiplication achieved, next we develop an effective preconditioner that can reduce the number of iterations in the iterative solution of (7). In addition, the solution of this preconditioner is constructed to be computationally efficient.

To elaborate, the mass matrix part in (3) can be split into two parts:

$$-k_0^2 \bar{\varepsilon}_r \langle \mathbf{N}_i, \mathbf{N}_j \rangle_V = -k_0^2 \mathbf{T} + j\omega\mu_0 \mathbf{R}, \tag{9}$$

where

$$\mathbf{T} = \varepsilon_r \langle \mathbf{N}_i, \mathbf{N}_j \rangle_V, \quad \mathbf{R} = \sigma \langle \mathbf{N}_i, \mathbf{N}_j \rangle_V. \tag{10}$$

The matrix structure of $\mathbf{T}$ and $\mathbf{R}$ in each layer is shown in Fig. 3, where N_{ls} denotes the upper surface unknowns in the lth layer, N_{lv} denotes the volume unknowns in the lth layer, and $N_{(l+1)s}$ denotes the upper surface unknowns in the $(l+1)$th layer, respectively. The matrix $\mathbf{T}$ possesses two important properties. First, the matrix block formed by volume unknowns (center matrix block) is decoupled from those formed between surface unknowns. Second, the matrix blocks formed between N_{ls} themselves (upper left blocks) in different layers are linearly proportional to each other; and the matrix blocks formed between N_{ls} and $N_{(l+1)s}$ (upper right blocks) in different layers are also linearly proportional to each other. Moreover, these two matrix blocks (upper left block and upper right block) are also mutually proportional to each other in each layer. The linear proportionality holds true for any multilayered integrated circuit structure and any choice of layer-growth direction due to the layered structure and the vector basis functions used in the layered finite element method. For example, assuming y is the stack-growth direction along which permittivity ε is layered. If one chooses the y direction as the layer-growth direction, the linear proportionality is obvious due to layered permittivity. If one chooses x or z direction as the layer-growth direction, the linear proportionality still holds true because although each layer does not have a constant permittivity, the configuration of the permittivity is the same across all layers.

$$\begin{array}{c|c|c|c}
 & N_{ls} & N_{lV} & N_{(l+l)s} \\ \hline
N_{ls} & \frac{l}{3}\varepsilon_r\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega & 0 & \frac{l}{6}\varepsilon_r\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega \\ \hline
N_{lV} & 0 & l\varepsilon_r\left\langle \xi_\mathrm{i},\xi_\mathrm{j}\right\rangle_\Omega & 0 \\ \hline
N_{(l+l)s} & \frac{l}{6}\varepsilon_r\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega & 0 & \frac{l}{3}\varepsilon_r\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega
\end{array}$$

(a)

$$\begin{array}{c|c|c|c}
 & N_{ls} & N_{lV} & N_{(l+l)s} \\ \hline
N_{ls} & \frac{l}{3}\sigma\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega & 0 & \frac{l}{6}\sigma\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega \\ \hline
N_{lV} & 0 & l\sigma\left\langle \xi_\mathrm{i},\xi_\mathrm{j}\right\rangle_\Omega & 0 \\ \hline
N_{(l+l)s} & \frac{l}{6}\sigma\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega & 0 & \frac{l}{3}\sigma\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega
\end{array}$$

(b)

Fig. 3. The structure of $\mathbf{T}$ and $\mathbf{R}$ in each layer. (a) Matrix structure of $\mathbf{T}$ in the lth layer. (b) Matrix structure of $\mathbf{R}$ in the lth layer.

$$\begin{array}{c|c|c|c}
 & N_{ls} & N_{lV} & N_{(l+l)s} \\ \hline
N_{ls} & \frac{l}{3}\sigma_c\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega & 0 & \frac{l}{6}\sigma_c\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega \\ \hline
N_{lV} & 0 & l\sigma_c\left\langle \xi_\mathrm{i},\xi_\mathrm{j}\right\rangle_\Omega & 0 \\ \hline
N_{(l+l)s} & \frac{l}{6}\sigma_c\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega & 0 & \frac{l}{3}\sigma_c\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega
\end{array}$$

Fig. 4. The structure of matrix $\mathbf{Rc}$ in each layer.

$$\begin{array}{c|c|c|c}
 & N_{ls} & N_{lV} & N_{(l+l)s} \\ \hline
N_{ls} & \frac{l}{3}\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega & 0 & \frac{l}{6}\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega \\ \hline
N_{lV} & 0 & l\left\langle \xi_\mathrm{i},\xi_\mathrm{j}\right\rangle_\Omega & 0 \\ \hline
N_{(l+l)s} & \frac{l}{6}\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega & 0 & \frac{l}{3}\left\langle \mathrm{W_i},\mathrm{W_j}\right\rangle_\Omega
\end{array}$$

Fig. 5. The structure of matrix $\mathbf{S}_1$ in the lth layer.

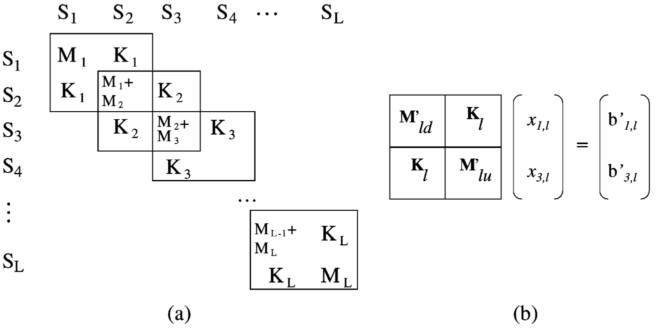


Fig. 6. (a) Matrix pattern of $\mathbf{P}$. (b) The reduced single-layer system matrix.

Matrix $\mathbf{R}$ as shown in Fig. 3(b) has a similar structure as $\mathbf{T}$. However, the distribution of conductivity σ in an integrated circuit is very different from that of permittivity ε. As can be seen from Fig. 3(b), although in each layer, the matrix block formed between N_{ls} and N_{ls} and that formed between N_{ls} and $N_{(l+1)s}$ are linearly proportional to each other, this property does not hold true across different layers. Therefore, in the following, we construct a conductivity-related matrix $\mathbf{R}_\mathrm{c}$, which processes the same property as $\mathbf{T}$.

The matrix structure of $\mathbf{R}_\mathrm{c}$ is shown in Fig. 4, in which σ_c is artificially assigned in each element, which satisfies

$$\sigma_c^e = \begin{cases} \sigma^e, & if\ \sigma^e > 0 \\ \sigma^e\ or\ 0, & if\ \sigma^e = 0 \end{cases} \tag{11}$$

where σ^e is the physical conductivity in element e. Basically, if the physical conductivity is nonzero in element e, σ_c is chosen the same as the physical conductivity; if the physical conductivity is zero in element e, σ_c is chosen either as the physical conductivity or 0 whichever that can make $\mathbf{R}_\mathrm{c}$ possess the same property as $\mathbf{T}$. A general approach of constructing $\mathbf{R}_\mathrm{c}$ is to replace each metal layer in a multilayered circuit by a solid metal plane. Thus, the conductivity σ_c has the same property as the permittivity ε_r, i.e., they both are layered.

The stiffness matrix part in (3) can also be split into two parts

$$\frac{1}{\mu_r}\langle \nabla \times \mathbf{N}_i, \nabla \times \mathbf{N}_j\rangle_V = \frac{1}{\mu_r}(\mathbf{S}_1 + \mathbf{S}_2) \tag{12}$$

where $\mathbf{S}_1$ possesses a similar structure as $\mathbf{T}$ and $\mathbf{R_c}$ as shown in Fig. 5.

The preconditioner $\mathbf{P}$ is constructed from $\mathbf{T}$, $\mathbf{R}_\mathrm{c}$, and $\mathbf{S}_1$ as the following

$$\mathbf{P} = -k_0^2\mathbf{T} + j\omega\mu_0\mathbf{R}_\mathrm{c} + \frac{1}{\mu_r}\mathbf{S}_1 \tag{13}$$

The matrix $\mathbf{P}$ serves as an effective preconditioner of the reduced system matrix. It was shown by our numerical experiments that with $\mathbf{P}$, the iterative process of solving the reduced system matrix converges in a small number of iterations for realistic on-chip circuits we have tested.

3) Efficient Computation of $\mathbf{P}^{-1}b$ *in Linear Complexity:* With the preconditioner $\mathbf{P}$ developed, the computing task of solving (7) becomes solving $\mathbf{P}w_j = \mathbf{K}_s v_j$ at each iteration as shown in Step (8.1) in (8). In the following, we show an efficient solution of $\mathbf{P}x = b$, with b being an arbitrary right-hand side.

The surface unknown related part in $\mathbf{P}$ has a matrix pattern shown in Fig. 6(a), in which $\mathbf{M}_i$ $(i = 1, 2, \ldots, L)$ are linearly proportional to each other, $\mathbf{K}_i$ $(i = 1, 2, \ldots, L)$ are lineary proportional to each other, and $\mathbf{M}_i$ $(i = 1, 2, \ldots, L)$ are also mutually proportional to $\mathbf{K}_i$ $(i = 1, 2, \ldots, L)$. To explain, $\mathbf{M}_i$ is made of $\mathbf{T}$, $\mathbf{R}_\mathrm{c}$, and $\mathbf{S}_1$ as can be seen from (13). If one chooses the layer growth direction as the two directions that are different from the stack-growth direction, then each layer has the same permittivity and conductivity σ_c configuration. As a result, $\mathbf{M}_i$ can be written as $\mathbf{M}_i = (-k_0^2\tilde{\mathbf{T}} + j\omega\mu_0\tilde{\mathbf{R}}_\mathrm{c} + (1/\mu_r)\tilde{\mathbf{S}}_1)l = \mathbf{M}l$ with $\mathbf{M}$ being the same across all the layers, and $\mathbf{K}_i = 0.5\mathbf{M}_i$. Hence, $\mathbf{M}_i$ $(i = 1, 2, \ldots, L)$ and $\mathbf{K}_i$ $(i = 1, 2, \ldots, L)$ are linearly proportional to each other. Note that the prism element is made in one layer and extruded along the layer growth direction.

Due to the linear proportionality of $\mathbf{M}_i$ and $\mathbf{K}_i$ $(i = 1, 2, \ldots, L)$, the matrix solution we developed in layered finite-element reduction-recovery method (LAFE-RR) [9], [10] can be used to compute $\mathbf{P}^{-1}b$ in linear complexity. Essentially, the matrix system can be analytically reduced to a single-interface system, which is then solved in linear complexity. From the solution of the single-interface system, the solution in other layers can be recovered in linear complexity. To elaborate, assuming that layer l is of interest, the surface-based $\mathbf{P}$ shown in Fig. 6(a) can be analytically reduced to a single-layer system matrix shown in Fig. 6(b). The matrix $\mathbf{M}'_{ld}$ carries the contribution from all the layers above layer l to layer l, while matrix $\mathbf{M}'_{lu}$ carries the contribution from all the layers below layer l to layer l. These two matrices can be obtained recursively and analytically from

$$\begin{aligned}\mathbf{M}'_2 &= \mathbf{M}_1 + \mathbf{M}_2 - \mathbf{K}_1\mathbf{M}_1^{-1}\mathbf{K}_1\\ \mathbf{M}'_3 &= \mathbf{M}_2 + \mathbf{M}_3 - \mathbf{K}_2\mathbf{M}_2'^{-1}\mathbf{K}_2\\ &\vdots\\ \mathbf{M}'_{ld} &= \mathbf{M}_{l-1} + \mathbf{M}_l - \mathbf{K}_{l-1}\mathbf{M}_{l-1}'^{-1}\mathbf{M}_{l-1}\end{aligned} \tag{14}$$

and

$$\begin{aligned}\mathbf{M}'_{L-1} &= \mathbf{M}_{L-1} + \mathbf{M}_L - \mathbf{K}_L\mathbf{M}_L^{-1}\mathbf{K}_L\\ \mathbf{M}'_{L-2} &= \mathbf{M}_{L-2} + \mathbf{M}_{L-1} - \mathbf{K}_{L-1}\mathbf{M}_{L-1}'^{-1}\mathbf{K}_{L-1}\\ &\vdots\\ \mathbf{M}'_{lu} &= \mathbf{M}_l + \mathbf{M}_{l+1} - \mathbf{K}_{l+1}\mathbf{M}_{l+1}'^{-1}\mathbf{M}_{l+1}.\end{aligned} \tag{15}$$

Matrices $\mathbf{M}'_{ld}$ and $\mathbf{M}'_{lu}$ can be obtained analytically because in (14) and (15), there is no need to compute the matrix inverse and matrix–matrix multiplication because of the linear proportionality. As a result, the reduction shown in Fig. 6, which is mathematically represented by (14) and (15), is achieved with minimal numerical calculation. To be more specific, if $\mathbf{M}_1$, which is $\mathbf{M}$ in the first layer, is chosen as the reference, then $\mathbf{M}'_{ld}$ and $\mathbf{M}'_{lu}$ can be obtained instantly by scaling $\mathbf{M}_1$ by a certain coefficient. Once the unknowns in the lth layer are solved, the unknowns in other layers can be obtained recursively as follows:

$$\begin{aligned}x_i &= \mathbf{M}_i'^{-1}\left[b'_i - \mathbf{K}_i x_{i+1}\right], \quad i = l-1,\ldots,2,1\\ x_{i+1} &= \mathbf{M}_i'^{-1}\left[b'_{i+1} - \mathbf{K}_i x_i\right], \quad i = l+1,2,\ldots,L-2\\ x_{i+1} &= \mathbf{M}_i^{-1}[b_{i+1} - \mathbf{K}_i x_i], \quad i = L-1.\end{aligned} \tag{16}$$

Once again, there is no need to compute $\mathbf{M}_i'^{-1}\mathbf{K}_i$ since the two matrices are linearly proportional to each other.

B. Fast Solution of the Single Layered System Matrix

The 2-D layered system $\mathbf{K}_{\mathrm{s}}$ shown in Fig. 2 can be further reduced to a single-layered system as demonstrated in [6]. This single-layered system can be made of any two interfaces, the crosstalk between ports on which is of interest. The single-layered system can be solved iteratively using the preconditioner proposed in the above section. Basically, the preconditioner $\mathbf{P}$ shown in Fig. 6 is reduced in the same fashion as the system matrix $\mathbf{K}_{\mathrm{s}}$, which is then used to solve the reduced single-layered system. It is shown by numerical experiments that the convergence of the proposed preconditioner on a reduced single-layered system is even faster than that on a 2-D layered system due to clustered eigenvalues, and hence a reduced spectrum. In general, the iterative solution based on the proposed preconditioner can converge in a few iterations. In addition, the reduction of the preconditioner $\mathbf{P}$ from a 2-D layered system to a single-layered one is analytical, and hence involving no computational cost.

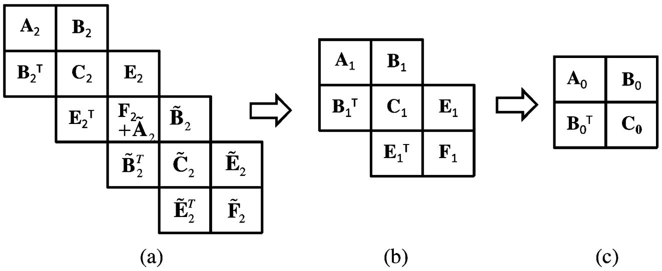


Fig. 7. Illustration of the reduction from a 2-D layered system to a single layered one. (a) A 2-D layered system consisting of 4 layers. (b) A reduced two-layer system. (c) A reduced single-layer system.

The remaining computational task is to perform the matrix–vector multiplication efficiently, and meanwhile bypass the 2-D-to-single-layer reduction cost. It should be noted that the cost of 2-D-to-single-layer reduction scales linearly with single-layer computational complexity. It is a one-time cost that can be amortized over many design iterations. Therefore, one can first pre-generate the reduced single-layered system and use it for many design iterations. However, if there is a need of minimizing the reduction cost, the following numerical procedure can be performed.

To ease the explanation of this procedure, in Fig. 7, we give an example of a reduction process from a four-layer system to a single-layered one, which is to be solved iteratively. Take the matrix block $\mathbf{A}_0$ shown in Fig. 7(c) as an example. $\mathbf{A}_0$ is obtained by a level-by-level reduction from Fig. 7(a)–(c). Our task is to compute $\mathbf{A}_0 x$ efficiently, with x being an arbitrary vector.

Based on [6, eq. (15)], $\mathbf{A}_0$ can be written as

$$\mathbf{A}_0 = \mathbf{A}_1 - \mathbf{B}_1\mathbf{C}_1^{-1}\mathbf{B}_1^T \tag{17}$$

where $\mathbf{A}_1$, $\mathbf{B}_1$, and $\mathbf{C}_1$ are the matrices shown in Fig. 7(b), which are one level above the single-layered matrices. By tracing back to another level above, $\mathbf{A}_0$ can be written as

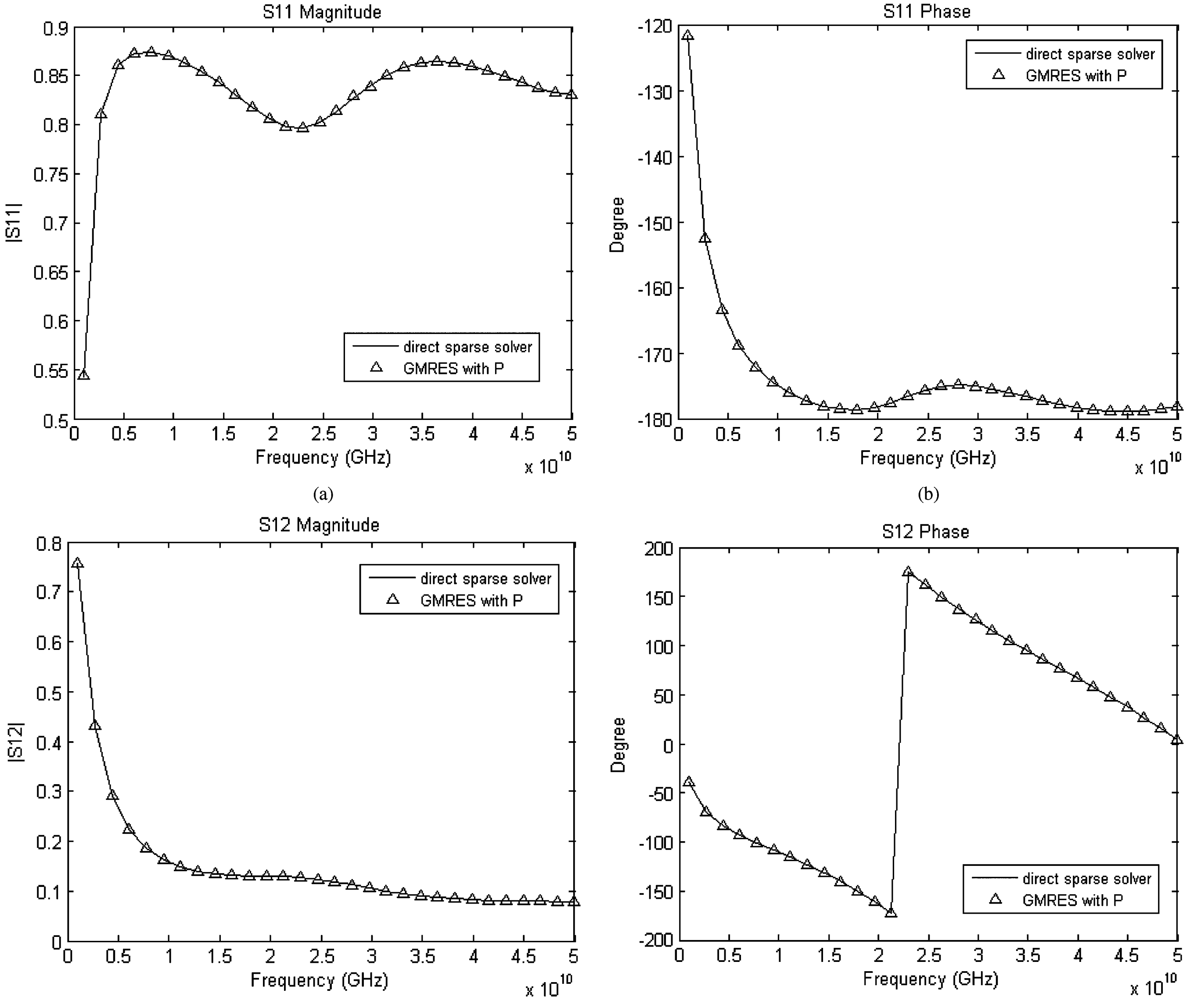
$$\begin{aligned}\mathbf{A}_0 = &\left(\mathbf{A}_2 - \mathbf{B}_2\mathbf{C}_2^{-1}\mathbf{B}_2^T\right)\\ &- \left(-\mathbf{B}_2\mathbf{C}_2^{-1}\mathbf{E}_2\right)\mathbf{C}_1^{-1}\left(-\mathbf{E}_2^T\mathbf{C}_2^{-1}\mathbf{B}_2^T\right)\end{aligned} \tag{18}$$

where $\mathbf{A}_2$, $\mathbf{B}_2$, $\mathbf{C}_2$, and $\mathbf{E}_2$ are the matrices shown in Fig. 7(a), which are matrices in a 2-D-layered system.

Instead of constructing $\mathbf{A}_0$ and computing $\mathbf{A}_0 x$ directly, we can use (18) to compute $\mathbf{A}_0 x$, i.e., do the following:

$$\left(\mathbf{A}_2 - \mathbf{B}_2\mathbf{C}_2^{-1}\mathbf{B}_2^T\right)x - \left(-\mathbf{B}_2\mathbf{C}_2^{-1}\mathbf{E}_2\right)\mathbf{C}_1^{-1}\left(-\mathbf{E}_2^T\mathbf{C}_2^{-1}\mathbf{B}_2^T\right)x. \tag{19}$$

Since $\mathbf{A}_2$, $\mathbf{B}_2$, and $\mathbf{E}_2$ are all matrices in the 2-D-layered system, based on the scheme developed in Section III-A-1, multiplying $\mathbf{A}_2$, $\mathbf{B}_2$, and $\mathbf{E}_2$ by any vector can be performed in linear complexity. As for multiplying $\mathbf{C}_2^{-1}$ by any vector v, it is equivalent to solving y from $\mathbf{C}_2 y = v$. Again the preconditioner proposed in Section III-A can be used to converge the iterative solution of $\mathbf{C}_2 y = v$ quickly. As a result, the computing task becomes the evaluation of $\mathbf{C}_2 y$, which again can be performed in linear complexity because $\mathbf{C}_2$ is a 2-D-layered system matrix block. The remaining task is to compute $\mathbf{C}_1^{-1} v$ with v being an arbitrary vector. Again it is equivalent to solving y from $\mathbf{C}_1 y = v$. The preconditioner proposed in Section III-A again can be used to converge the iterative



(a)

(b)

(c)

(d)

Fig. 8. *S*-parameters of an on-chip interconnect structure of length 2000 μm. (a) S11 magnitude. (b) S11 phase. (c) S12 magnitude. (d) S12 phase.

solution of $\mathbf{C}_1 y = v$ quickly. The computing task hence again becomes the computing of a matrix–vector multiplication $\mathbf{C}_1 y$. Matrix $\mathbf{C}_1$ can be written as

$$\mathbf{C}_1 = \mathbf{F}_2 - \mathbf{E}_2^T \mathbf{C}_2^{-1} \mathbf{E}_2 + \tilde{\mathbf{A}}_2 - \tilde{\mathbf{B}}_2 \tilde{\mathbf{C}}_2^{-1} \tilde{\mathbf{B}}_2^T. \tag{20}$$

Clearly, all the matrices involved in (20) are the 2-D layered matrix blocks, and hence based on the analysis above, multiplying $\mathbf{C}_1$ by any vector can be performed in linear complexity. As a result, the single-layered matrix–vector multiplication can be performed in linear complexity, and meanwhile the reduction cost from a 3-D layered system to a single-layered system is bypassed. In summary, the fast iterative solution proposed in Section III-A can be recursively used at each level shown in Fig. 7 to reduce the computational cost.

IV. Numerical Results

The performance of the proposed fast iterative solution was tested on a number of 3-D on-chip interconnect structures. For all these examples, the initial guess used in the iterative solution was set to be zero.

The first example was a test-chip 3-D interconnect structure. This was the first example simulated in [6], and also the example shown in [11, Fig.5]. The structure was 300 μm wide and 2000 μm long. It involved a 10-μm-wide strip in metal 2 (M2) layer, one ground plane in metal 1 (M1) layer, and one ground plane in metal 3 (M3) layer. The distance of this strip to the M2 returns at the left- and right-hand sides was 50 μm, respectively. Along the length, the structure was divided into 22 layers, resulting in 7175 unknowns. After representing the system matrix (2) as (7) by removing volume unknowns, only 5635 surface unknowns are remained. Note that no computational cost is incurred by rewriting (2) as (7) since $\mathbf{A}_\mathrm{r}$, $\mathbf{B}_\mathrm{r}$, $\mathbf{C}_\mathrm{r}$, and $\mathbf{D}_\mathrm{r}$ in $\mathbf{K}_\mathrm{s}$ are not actually formed. The proposed preconditioner was then used to solve (7) to extract *S*-parameters. In Fig. 8, we plot the *S*-parameters simulated over the entire frequency band. The *S*-parameters show an excellent agreement with those obtained

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT PRECONDITIONERS FOR SOLVING THE REDUCED 2-D LAYERED SYSTEM

	External Iteration Number	Internal Iteration Number	S-Parameters

Direct sparse solver	NA	NA	S11 = -0.2880-0.4638j
			S12 = 0.5884-0.4742j

GMRES without a preconditioner	2	20	S11 = -0.9999+2.010(e-7)j
			S12 = 0.2869-0.4720j

GMRES with **P'**	2	20	S11 = -0.9999+5.5408(e-6)j
			S12 = 0.2869-0.4720j

GMRES with **P**	2	20	S11 = -0.2891-0.4627j
			S12 = 0.5877-0.4752j

from a direct solver [14], which was shown to agree well with the measured data [6].

Next, we compared the performance of the proposed preconditioner with other preconditioning techniques. One preconditioning technique is to use the combined $\mathbf{T}$ and $\mathbf{R}_c$ matrix without $\mathbf{S}_1$ as shown below

$$\mathbf{P}' = -k_0^2\mathbf{T} + j\omega\mu_0\mathbf{R}_c. \tag{21}$$

The preconditioner $\mathbf{P}'$ also permits an efficient solution described in Section III-A3. In Table I, we listed S-parameters simulated with $\mathbf{P}$, $\mathbf{P}'$, and without any preconditioner. The frequency simulated was 1 GHz. As shown in Table I, within 40 iterations, the proposed iterative solver converged to the reference S-parameters obtained by a direct sparse solver [14]; whereas the other two preconditioning techniques failed to converge. In Table I, we list both external iteration number and internal iteration number. The total iteration number is the product of the two iteration numbers.

The convergence of the proposed iterative solver for solving the reduced 2-D layered system (7) is plotted in Fig. 9. The external iteration number in GMRES, denoted by *maxit*, was set to be 2, and the internal iteration number, denoted by *restart*, was allowed to change. The relative residual $(relres)$ was defined as

$$relres = \frac{\left\|\mathbf{P}^{-1}\left(b - \mathbf{K}u^{(k)}\right)\right\|_2}{\|\mathbf{P}^{-1}b\|_2} \tag{22}$$

where $u^{(k)}$ is the solution at the kth step. The log function used in plotting Fig. 9 is a natural logarithm function. Fast convergence can be observed from Fig. 9.

The CPU time cost by the proposed iterative solver was then examined. To test the scaling of the proposed solver with respect to unknown number N, the size of the original problem was enlarged by adding more layers to generate a range of unknowns from 4025 to 13 475. The CPU time per iteration for step (8.1) is listed in Table II with respect to unknown number. A linear scaling with respect to unknown number can be observed, which verifies the proposed linear-time matrix–vector multiplication and linear-time solution for the preconditioner $\mathbf{P}$.

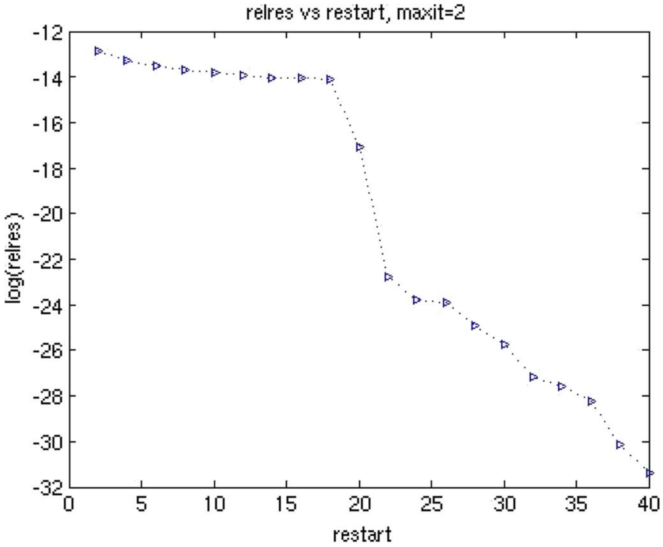


Fig. 9. Relative residual versus number of iterations for solving a reduced 2-D layered system. (Natural logarithm is used.)

TABLE II
CPU COST FOR SOLVING A REDUCED 2-D LAYERED SYSTEM

Unknown Number	4025	7175	10325	13475
CPU time Per Iteration	33.3ms	66.7ms	100.1 ms	133.3 ms

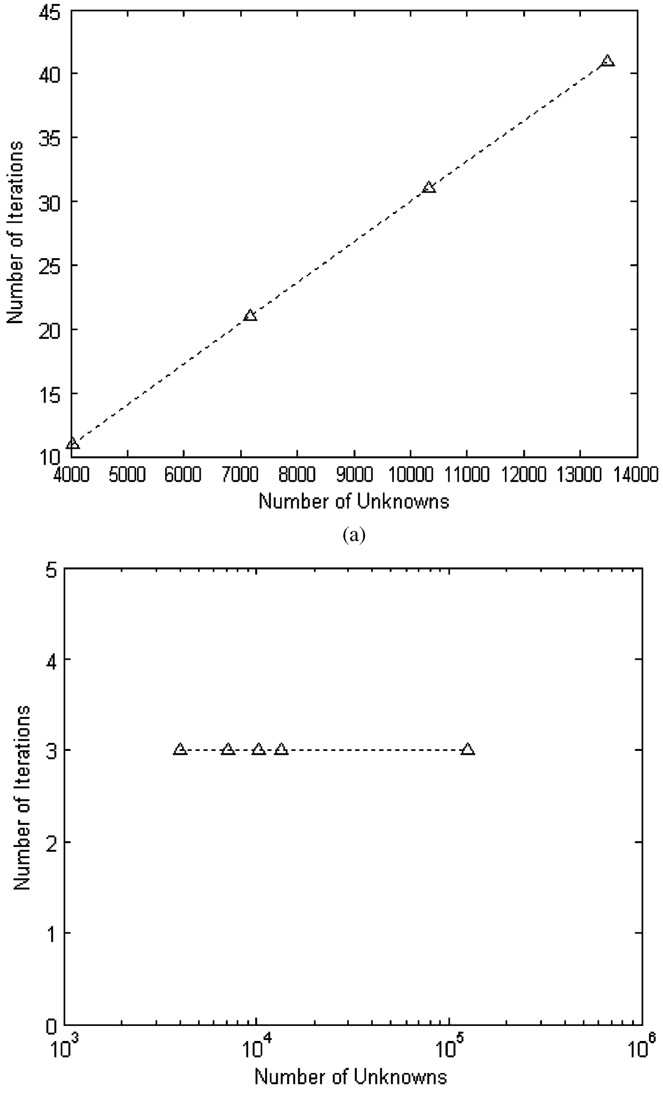
TABLE III
CPU TIME COMPARISON

No. of unknowns	4025	7175	10325	13475
Direct solution time for solving (2)	2.10 s	7.73 s	11.69 s	16.09 s
Direct solution time for solving (7)	7.21 s	12.26 s	17.13 s	22.18 s
CPU time of the proposed iterative solution for solving (7)	0.58 s	1.51 s	3.19 s	5.40 s

We then compared the CPU time of the proposed iterative solver with the layered finite element analysis based on the multifrontal-based direct sparse solver [14]. The CPU time cost by the direct solver for solving the original matrix (2) as well as the cost for solving the 2-D layered system (7) is listed in Table III. As shown in Table III, the proposed iterative solver costs less time due to the efficiency of the proposed preconditioner. It should be noted that there is no reduction cost involved in solving (7) in the proposed iterative solution because the reduced matrix is actually not formed. Their analytical expressions are used to carry out efficient matrix–vector multiplication as described in Section III-A1.

In Fig. 10, we plot the number of iterations versus the number of unknowns for both 2-D-based reduced system and a single-layer based reduced system. The relative residual was set as 1e-4. As can be observed from Fig. 10(a), the iteration number grows with unknown number for a reduced 2-D system. However, for the single-layered system, the iteration number required to reach convergence remains as a constant, as can be seen from Fig. 10(b). To confirm this, we simulated a case involving 0.1 million unknowns. Once again, we observe a constant number of iterations as can be seen from Fig. 10(b). The relative residual used for Fig. 10(b) was 1e-6.

The second structure simulated was an on-chip bus structure as shown in Fig. 11, where the unit of geometrical dimensions



(a)

(b)

Fig. 10. Number of iterations versus number of unknowns required for solving (a) a 2-D layered system and (b) a single-layered system.

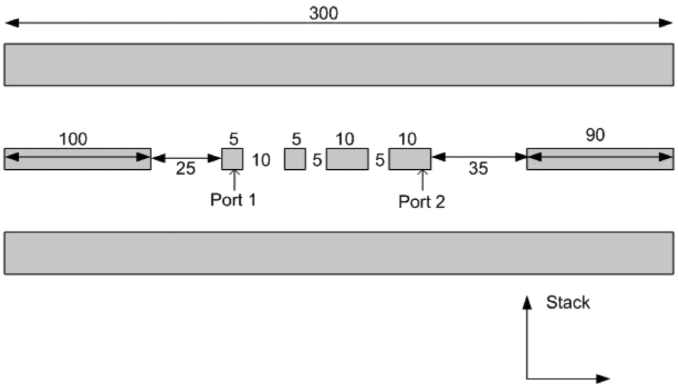


Fig. 11. The cross-sectional view of an on-chip bus structure.

is μm. The dielectric stack was kept the same as that in previous example.

The preconditioner $\mathbf{P}$ and $\mathbf{P}'$ were constructed for performance comparison. The layer-growth direction was chosen as the length direction z. Along z, the structure was discretized into 22 layers. The 22-layer system was then reduced to a three-layer system by the layered finite element method. The preconditioners $\mathbf{P}$ and $\mathbf{P}'$ were then applied to solving the reduced three-layer system respectively, with the first and last layer used for absorbing outgoing waves. As shown in Table IV, the proposed preconditioner converged the iterative solution in 40 iterations, whereas $\mathbf{P}'$ failed to converge the iterative solution. In addition, the iterative solution without a preconditioner also failed to converge in 40 iterations.

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT PRECONDITIONERS (EXAMPLE 2)

	External Iteration Number	Internal Iteration Number	S-Parameter

Direct sparse solver	NA	NA	S11 = 0.1839-0.9138j
			S12 = -3.1877(e-7)+1.4962(e-7)j

GMRES without a preconditioner	2	20	S11 = -0.9999+8.4904(e-8)j
			S12 = -8.2203(e-8)+2.7542(e-7)j

GMRES with **P'**	2	20	S11 = -0.9999+4.879(e-6)j
			S12 = -8.2202(e-8)+2.7542(e-7)j

GMRES with **P**	2	20	S11 = 0.1839-0.9138j
			S12 = -3.0481(e-7)+1.5770(e-7)j

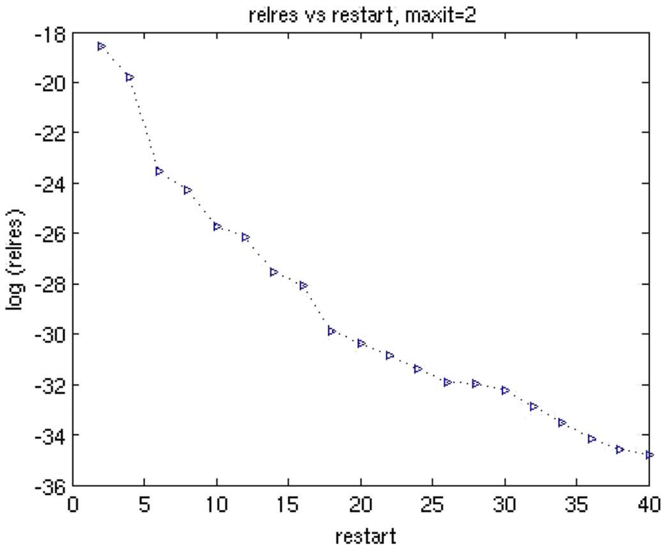


Fig. 12. Relative residual versus iteration number for solving a reduced three-layer system (natural logarithm is used).

The relative residual (relres) versus the inner iteration number is plotted in Fig. 12, with external iteration number chosen to be 2. Again, fast convergence is observed. In addition, compared to the 2-D layered system, the reduced single-layered system is shown to converge even faster. This is mainly because the reduction procedure helps cluster eigenvalues, and hence expediting convergence.

We then compared the CPU time of the proposed iterative solver for the reduced system with the layered finite element analysis based on the multifrontal-based direct sparse solver. As can be seen from Table V, the proposed iterative solution is shown to be faster.

The third structure was a 3-D bus structure with orthogonal returns as shown in Fig. 13. The structure is of length 2000 μm. The spacing between the orthogonal wires as well as the width of each orthogonal wire is 100 $\mu\mathrm{m}$. The 3-D layered system was first reduced to a single-layer one. The proposed preconditioner

TABLE V
CPU COST FOR SOLVING A FURTHER REDUCED SYSTEM THAT HAS A SINGLE-LAYER INTERCONNECT

	Iteration	CPU time (s)	S-Parameter

LFEM based on the direct sparse solver (LU decomposition)	NA	20.3	S11 = 0.1839-0.9138j
			S12 = -3.1877(e-7)+1.4962(e-7)j

LFEM based on the proposed iterative solution	2	8.2	S11 = 0.1840-0.9138j
			S12 = -4.4168(e-7)+2.0941(e-7)j

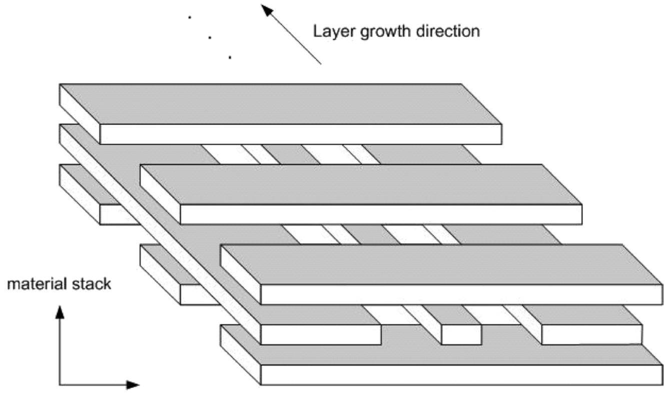


Fig. 13. A 3-D bus structure with orthogonal returns.

TABLE VI
PERFORMANCE OF THE PROPOSED PRECONDITIONER (EXAMPLE 3)

	External Iteration Number	Internal Iteration Number	Relative Residual	S-Parameter

Direct sparse solver	NA	NA	NA	S11 = -0.1460-0.3939j
				S12 = 0.7306-0.4066j

GMRES with **P**	1	2	7.8832e-9	S11 = -0.1460-0.3940j
				S12 = 0.7307-0.4066j

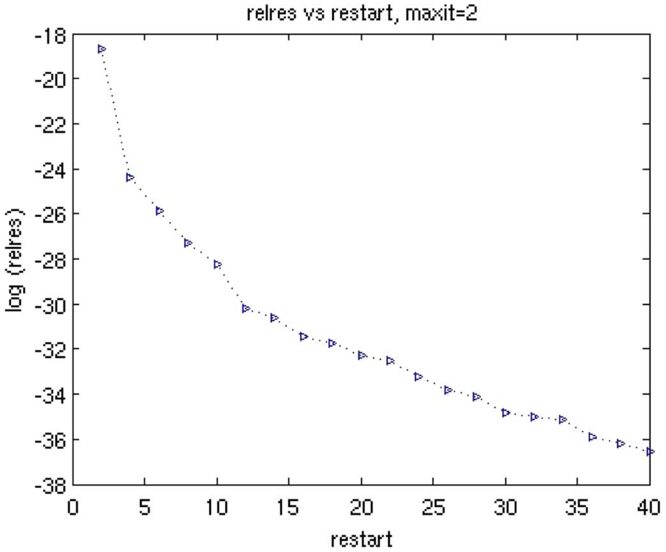


Fig. 14. Relative residual versus iteration number for solving a reduced single-layer system (natural logarithm is used.)

$\mathbf{P}$ was then applied to solving the single-layer system. The matrix $\mathbf{R}_c$ in $\mathbf{P}$ was constructed by replacing orthogonal returns in the top and bottom metal layers by a solid metal plane. With $\mathbf{P}$, the iterative solution converged in two iterations as shown in Table VI. The convergence of the proposed iterative solution is plotted in Fig. 14. Fast convergence can be observed.

V. CONCLUSION

Fast iterative solution algorithms are developed in this work for solving the system matrix resulting from a layered finite element based analysis of integrated circuits. The algorithms include fast dense matrix–vector multiplication that can bypass the reduction cost, an effective preconditioner, and an efficient solution of the preconditioner. Numerical experiments on both 2-D layered systems and further reduced single-layered systems have demonstrated the fast convergence of the proposed iterative solution.

REFERENCES

[1] J. Ihm and A. C. Cangellaris, "Modeling of semiconductor substrate on on-chip power grid switching," in *IEEE 13th Topical Meeting on Electrical Performance of Electronic Packaging (EPEP)*, 2004, pp. 265–268.

[2] P. J. Restle, A. E. Ruehli, S. G. Walker, and G. Papadopoulos, "Full-wave PEEC time-domain method for the modeling of on-chip interconnects," *IEEE Trans. CAD*, vol. 20, no. 7, pp. 877–887, July 2001.

[3] W. C. Chew, "Toward a more robust and accurate fast integral solver for microchip applications," in *IEEE 12th Topical Meeting on Electrical Performance of Electronic Packaging (EPEP)*, 2003.

[4] S. Kapur and D. E. Long, "Large-scale full-wave simulation," in *41th ACM/IEEE Design Automation Conference*, 2004, pp. 806–809.

[5] D. Jiao, M. Mazumder, S. Chakravarty, C. Dai, M. Kobrinsky, M. Harmes, and S. List, "A novel technique for full-wave modeling of large-scale three-dimensional high-speed on/off-chip interconnect structures," in *International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, 2003, pp. 39–42.

[6] D. Jiao, S. Chakravarty, and C. Dai, "A layered finite-element method for electromagnetic analysis of large-scale high-frequency integrated circuits," *IEEE Trans. Antennas Propagat.*, vol. 55, no. 2, pp. 422–432, Feb. 2007.

[7] F. Sheng, S. Chakravarty, and D. Jiao, "An efficient 3-D-to-2-D reduction technique for frequency-domain layered finite element analysis of large-scale high-frequency integrated circuits," in *IEEE 16th Topical Meeting on Electrical Performance of Electronic Packaging (EPEP)*, 2007, pp. 295–298.

[8] F. Sheng and D. Jiao, "Fast reduction algorithms in the frequency-domain layered finite element method for the electromagnetic analysis of large-scale high-frequency integrated circuits," *IEEE Trans. Advanced Packaging*, 2009, to appear.

[9] H. Gan and D. Jiao, "A time-domain Layered Finite Element Reduction Recovery (LAFE-RR) method for high-frequency VLSI design," *IEEE Trans. Antennas Propagat.*, vol. 55, no. 12, pp. 3620–3629, Dec. 2007.

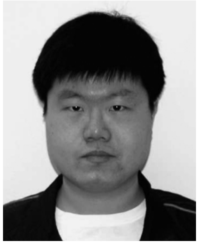
[10] H. Gan and D. Jiao, "Hierarchical finite element reduction recovery method for large-scale transient analysis of high-speed integrated circuits," *IEEE Trans. Advanced Packaging*, 2009, to appear.

[11] M. J. Kobrinsky, S. Chakravarty, D. Jiao, M. C. Harmes, S. List, and M. Mazumder, "Experimental validation of crosstalk simulations for on-chip interconnects using S-parameters," *IEEE Trans. Advanced Packaging*, vol. 28, no. 1, pp. 57–62, Feb. 2005.

[12] J. M. Jin*, The Finite Element Method in Electromagnetics*, 1st ed. New York: John Wiley & Sons, 1993.

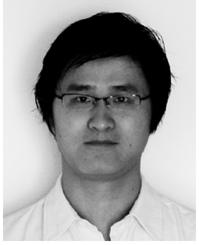
[13] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA: SIAM, 2003.

[14] UMFPACK [Online]. Available: http://www.cise.ufl.edu/research/sparse/umfpack/



Feng Sheng received his B.S. degree in Electronic & Information Engineering from Zhejiang University, Hangzhou, China, in 2006. He is now working towards his Ph.D. degree in the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

In August 2006, Feng Sheng joined the On-Chip Electromagnetics Group, Purdue University, as a Research Assistant. His current research interest is computational electromagnetics for large-scale high-frequency integrated circuit design.



Houle Gan (S'08) received the B.S. and M.S. degrees in information science and electronic engineering from Zhejiang University, Hangzhou, China, in 2003 and 2006, respectively. He is now working toward the Ph.D. degree in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

In 2006, He was a System Engineer in Realsil Microelectronics Inc., Suzhou, China. In September 2006, he joined On-Chip Electromagnetics Group, Purdue University, as a Research Assistant. His current research interest is computational electromagnetics for large-scale high-frequency integrated circuit design.

Mr. Gan was one of the three recipients of the IEEE Antennas and Propagation Society Ph.D. Research Award for 2008–2009 selected worldwide.

Dan Jiao (S'00–M'02–SM'06) received her Ph.D. degree in Electrical Engineering from the University of Illinois at Urbana-Champaign in October 2001. She then worked at Technology CAD Division at the Intel Corporation until September 2005 as Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. In September 2005, she joined Purdue University as an Assistant Professor in the School of Electrical and Computer Engineering. Since Aug. 2009, she has been an Associate Professor. She has authored two book chapters and over 100 papers in refereed journals and international conferences. Her current research interests include high frequency digital, analogue, mixed-signal, and RF IC design and analysis, high-performance VLSI CAD, modeling of micro- and nano-scale circuits, computational electromagnetics, applied electromagnetics, fast and high-capacity numerical methods, fast time domain analysis, scattering and antenna analysis, RF, microwave, and millimeter wave circuits, wireless communication, and bio-electromagnetics.

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