

# Fast Reduction Algorithms in the Frequency-Domain Layered Finite Element Method for the Electromagnetic Analysis of Large-Scale High-Frequency Integrated Circuits

Feng Sheng and Dan Jiao, *Senior Member, IEEE*

**Abstract**—In this paper, fast algorithms are proposed for an efficient reduction of a 3-D layered system matrix to a 2-D layered one in the framework of the frequency-domain layered finite element method. These algorithms include: 1) an effective preconditioner  $P$  that can converge the iterative solution of the volume-unknown-based matrix equation in a few iterations; 2) a fast direct computation of  $P^{-1}$  in linear complexity in both CPU run time and memory consumption; and 3) a fast evaluation of  $P^{-1}b$  in linear complexity, with  $b$  being an arbitrary vector. With these fast algorithms, the volume-unknown-based matrix equation is solved in linear complexity with a small constant in front of the number of unknowns, and hence significantly reducing the complexity of the 3-D to 2-D reduction. The algorithms are rigorous without making any approximation. They apply to any arbitrarily-shaped multilayer structure. Numerical and experimental results are shown to demonstrate the accuracy and efficiency of the proposed algorithms.

**Index Terms**—Electromagnetic modeling, finite element method, frequency domain, large-scale, on-chip.

## I. INTRODUCTION

AS IC DESIGN SCALES into the deep submicron regime (and the nanometer regime), electromagnetics (EM)-based analysis has increasingly become essential for four major reasons.

- 1) Reduced feature sizes: at the 45 nm processing technology node and beyond, the IC industry will have to print features that are several times less than the wavelength of light (193 nm) being used in optical lithography. In this regime, light has to be modeled as a wave rather than a ray approximation.
- 2) Increased clock frequency: as it is necessary to analyze the signal response with harmonics five times the clock frequency, static RLC-based simulations are no longer adequate.

Manuscript received July 21, 2008; revised December 31, 2008. First published October 30, 2009; current version published February 26, 2010. This work was supported in part by a grant from the Office of Naval Research under Award N00014-06-1-0716, in part by a grant from Intel Corporation, and in part by a grant from the National Science Foundation under Award 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.

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.2009.2014353

- 3) The transition from single core to multicore: active power management of multicore processors requires EM analysis of the global power supply network in order to accurately model current variations and transient power drops and ground bounces.
- 4) Increased level of integration: integrating RF, analog, and digital circuitry on a same chip leads often to undesirable coupling and sometimes to system failure. For high-frequency mixed-signal design, an electromagnetic solution is required to overcome the fundamental limits of circuit-based solutions.

In recent years, researchers in both circuits and fields have initiated the development of innovative computational EM techniques for on-chip problems [1]–[12]. In [4] and [5], a layered finite element method (LFEM) was developed for electromagnetic analysis of large-scale high-frequency integrated circuits. In this method, the integrated circuit is discretized into many layers by using triangular prism elements, with the prism axis being the *layer-growth* direction. The vector prism basis functions are used to expand the unknown electric field. After discretization, the system matrix of the original 3-D problem is reduced to that of 2-D layers. The system matrix of 2-D layers is then reduced to that of a single layer. The cost of reduction scales linearly with single-layer computational complexity. The entire numerical procedure is rigorous without making any approximation. In a realistic on-chip structure, one can encounter a large number of layers when choosing the layer-growth direction, i.e., slicing the structure, along either  $x$ - or  $z$ -direction, assuming  $y$  is the dielectric stack-growth direction. (Note that in current processing technology of integrated circuits, the dielectric stack is grown layer by layer.) Hence, the number of single-layer unknowns is generally much less than that of total unknowns. For example, the former is 2270 while the latter is 3.04 million in a test-chip interconnect [16] we have measured before. 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. Although the reduction has certain cost, the cost can be amortized over many design iterations.

To further improve the efficiency of the LFEM method, fast reduction algorithms are proposed here to efficiently reduce the system matrix of the original 3-D problem to that of 2-D layers. The volume-unknown-based matrix is first structured to be a

block tridiagonal matrix. This reduces the complexity of the 3-D to 2-D reduction from  $O(M^3)$  to  $O(M^{2.5})$ , with  $M$  being the number of volume unknowns in a single layer (this notation is used throughout this paper). The complexity of  $O(M^{2.5})$  is further reduced to  $O(M^2)$  by developing a number of fast algorithms. These algorithms include: 1) an effective preconditioner  $\mathbf{P}$  that can converge the iterative solution of the volume-unknown-based matrix equation in a few iterations; 2) a fast direct computation of  $\mathbf{P}^{-1}$  in linear complexity in both CPU run time and memory consumption; and 3) a fast evaluation of  $\mathbf{P}^{-1}\mathbf{b}$  in linear complexity, with  $\mathbf{b}$  being an arbitrary vector. With these fast algorithms, the volume-unknown-based matrix equation is solved in linear complexity. In addition, the constant in front of the number of unknowns is small (less than 25). The accuracy and efficiency of the proposed fast reduction algorithms have been demonstrated by numerical results and measurements.

The remainder of this paper is organized as follows. In Section II, the LFEM method is briefly reviewed. In Section III, the proposed fast reduction algorithms are presented. In Section IV, numerical and experimental results are given to demonstrate the accuracy and efficiency of the proposed algorithms. Section V relates to our conclusions.

## II. BRIEF REVIEW OF THE LFEM METHOD

Consider 3-D integrated circuit problems. The circuit can be a global on-chip interconnect network, a package, and a mixed-signal IC. Generally, these circuits are multilayer structures. They are embedded in a multilayer dielectric medium backed by silicon, GaAs, InP or other substrates. A Manhattan-type on-chip circuit is even layered in any of the  $x$ -,  $y$ -, and  $z$ -directions. Inside these circuits, the electric field  $\mathbf{E}$  satisfies the second-order vector wave equation

$$\nabla \times [\mu_r^{-1} \nabla \times \mathbf{E}] - k_0^2 \bar{\epsilon}_r \mathbf{E} = -jk_0 Z_0 \mathbf{J} \text{ in } V \quad (1)$$

subject to certain boundary conditions. In (1),  $\mu_r$  is the relative permeability;  $\bar{\epsilon}_r$  denotes a complex permittivity that comprises both permittivity and conductivity;  $k_0$  and  $Z_0$  are free-space wave number and impedance, respectively;  $\mathbf{J}$  is the current source;  $V$  is the computational domain that encloses the circuit. To solve (1), a numerical algorithm is formulated to obtain the  $\mathbf{E}$  field or  $\mathbf{H}$  field inside the computational domain, from which the design parameters of interest are obtained. Due to the computational complexity of on-chip circuit problems, the resultant numerical system is generally large even for a small circuitry.

To overcome the large problem size, in [4], [5], a layered finite element method was developed. This method consists of a number of steps. The first step is to reduce a 3-D layered system matrix to a 2-D layered one. This is equivalent to eliminating all the volume unknowns (unknowns that are along layer-growth direction). For instance, the volumetric unknowns in layer 1, which are  $N_{1v}$ , can be eliminated by using the procedure illustrated in Fig. 1, where  $N_{1s}$  and  $N_{2s}$  are the surface unknowns on the top and bottom surfaces of layer 1. In Fig. 1, matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$ ,  $\mathbf{E}$ , and  $\mathbf{F}$  are formed by the vector basis functions associated with corresponding unknowns. For example, matrix  $\mathbf{A}$  is formed by the bases associated with upper surface unknowns,  $\mathbf{B}$  is formed by the bases relating to volume unknowns, and  $\mathbf{D}$

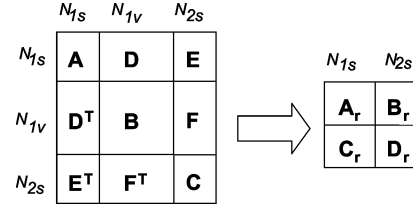


Fig. 1. Illustration of eliminating volume unknowns.

is formed between the bases pertaining to the upper surface unknowns and volume ones. These matrices are assembled from their elemental contributions as follows:

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

in which  $\mathbf{W}$  is the edge basis function ([17], pp. 234–237),  $\xi$  is the node basis function [17, pp. 80],  $\Omega$  denotes the support of a triangular element, and  $l$  is the height of a prism element, which is the layer thickness.

As shown in [4], [5], the reduced matrices can be obtained from the original matrices as follows:

$$\begin{aligned} \mathbf{A}_r &= \mathbf{A} - \mathbf{D}\mathbf{B}^{-1}\mathbf{D}^T \\ \mathbf{B}_r &= \mathbf{E} - \mathbf{D}\mathbf{B}^{-1}\mathbf{F} \\ \mathbf{C}_r &= \mathbf{B}_r^T \\ \mathbf{D}_r &= \mathbf{C} - \mathbf{F}^T\mathbf{B}^{-1}\mathbf{F}. \end{aligned} \quad (3)$$

Due to the matrix properties such as  $\mathbf{F} = -\mathbf{D}^T$  and  $\mathbf{A} = \mathbf{C}$  [4], [5], (3) can be evaluated as

$$\begin{aligned} \mathbf{A}_r &= \mathbf{A} - \mathbf{D}\mathbf{B}^{-1}\mathbf{D}^T \\ \mathbf{B}_r &= \mathbf{E} + \mathbf{D}\mathbf{B}^{-1}\mathbf{D}^T \\ \mathbf{C}_r &= \mathbf{B}_r^T \\ \mathbf{D}_r &= \mathbf{A} - \mathbf{D}\mathbf{B}^{-1}\mathbf{D}^T. \end{aligned} \quad (4)$$

Therefore, the major computation task of the reduction becomes the evaluation of  $\mathbf{D}\mathbf{B}^{-1}\mathbf{D}^T$ . Since  $\mathbf{B}$  is sparse, in [4], [5],  $\mathbf{D}\mathbf{B}^{-1}\mathbf{D}^T$  is evaluated as follows.

- Step 1) An LU factorization of  $\mathbf{B}$  is performed by using an advanced sparse solver such as a multifrontal method [13].
- Step 2)  $\mathbf{B}^{-1}\mathbf{D}^T$  is calculated by forward and backward substitution.
- Step 3)  $\mathbf{D}\mathbf{B}^{-1}\mathbf{D}^T$  is obtained.

The above procedure is referred to as *direct reduction* in this sequel. It can be expensive when the dimension of  $\mathbf{B}$  is large.

The computational cost of Step 1 is  $O(M^3)$  in CPU and  $O(M^2)$  in memory. The cost of Step 2 is  $O(M^3)$  in CPU run time and  $O(M^2)$  in memory since there exist  $O(M)$  columns in  $\mathbf{D}^T$ . The cost of Step 3 is  $O(M^2)$  in CPU run time and memory (note that  $\mathbf{D}$  is sparse). As a result, the total cost of evaluating  $\mathbf{DB}^{-1}\mathbf{D}^T$ , and hence the cost of reducing a 3-D layered system matrix to a 2-D layered one is dominated by Step 2 and Step 1, which is  $O(M^3)$  in CPU run time. This is expensive when  $M$  is large. In the next section, fast reduction algorithms are proposed to reduce the computational cost by  $O(M)$  times. In these algorithms,  $\mathbf{B}$  is first structured to be block-tridiagonal. The cost of LU factorization is hence brought down to  $O(M^2)$  from  $O(M^3)$ , and the cost of backward and forward substitution is brought down to  $O(M^{1.5})$  from  $O(M^2)$ . Then a number of fast algorithms are developed to bring the complexity further down to  $O(M)$ .

### III. FAST REDUCTION ALGORITHMS

The key to evaluate  $\mathbf{DB}^{-1}\mathbf{D}^T$  efficiently is to solve the following matrix equation efficiently:

$$\mathbf{B}x = d \quad (5)$$

in which  $d$  is one of the column vectors of  $\mathbf{D}^T$ . Here, a preconditioned system

$$\mathbf{P}^{-1}\mathbf{B}x = \mathbf{P}^{-1}d \quad (6)$$

is solved iteratively to obtain  $x$ . It is well known that to solve (6) efficiently, three requirements need to be satisfied. First, the preconditioner  $\mathbf{P}$  must be effective so that the number of iterations can be minimized in a solution process. Second, it should be computationally inexpensive to obtain the inverse or the factorization of  $\mathbf{P}$ . And third, it should be computationally inexpensive to obtain  $\mathbf{P}^{-1}b$ , with  $b$  being an arbitrary vector. The proposed fast reduction algorithms fulfill all the three requirements. With them, (6) is solved with a linear complexity, which is the optimal complexity one can hope for to solve (6), and hence (5). In addition, the constant multiplier in front of the number of unknowns is very small.

The fast reduction algorithms encompass five components: 1) a structuring of matrix  $\mathbf{B}$ , 2) an effective preconditioner  $\mathbf{P}$ , 3) an iterative solution of (6) and a proof of its convergence, 4) a fast direct computation of  $\mathbf{P}^{-1}$  in linear complexity, and 5) a fast evaluation of  $\mathbf{P}^{-1}b$  in linear complexity, each of which is elaborated in the following subsections respectively.

#### A. A Structuring of Matrix $\mathbf{B}$

Matrix  $\mathbf{B}$  is structured to be a block tridiagonal matrix. To ease the explanation of the structuring of  $\mathbf{B}$ , we give an example of a typical on-chip structure in Fig. 2(a). As shown in this figure, the dielectric stack is grown vertically layer by layer along  $y$ . This direction is called as *stack-growth* direction in this paper. In the LFEM, one can choose the *layer-growth* (prism-axis) direction the same as the stack-growth direction, i.e.,  $y$ -direction in the specific example shown in Fig. 2(a), or different from the stack-growth direction. Here, without loss of generality, we choose  $z$ -direction as the layer-growth direction,

i.e., the direction to grow the layers of prism elements. Then, the volume unknowns are assigned along  $z$  direction. Recall that matrix  $\mathbf{B}$  is formed between volume unknowns in each layer. Fig. 2(b) depicts an  $x - y$  cross-sectional view of these volume unknowns, in which each dot denotes a volume unknown. If the unknowns are ordered from line 1 to line 2 to line 3, and to line  $N_x$ , a  $\mathbf{B}$  matrix having a pattern shown in Fig. 2(c) will be generated. It is a block tridiagonal matrix. More important, each diagonal block is a *tridiagonal* matrix. The off-diagonal blocks are very sparse. For example, for the mesh shown in Fig. 2(b), each row has only two nonzero elements (or even less), one is the diagonal one, and the other is the super- or sub-diagonal one.

The lines 1, 2, 3, ..., and  $N_x$  in the mesh shown in Fig. 2(b) are used to form a block-tridiagonal matrix. Apparently, these lines impose an additional constraint on the meshing. However, in a realistic on-chip circuit, these lines do exist physically, which are the material interfaces between the conducting region and the dielectric region as can be seen from Fig. 2(a). Because of a large number of conductors present in an on-chip circuit (these conductors are used to connect millions of transistors), there exist many of these lines. Even one does not introduce these lines into the mesh, these lines exist, along which the mesh must be partitioned.

#### B. An Effective Preconditioner $\mathbf{P}$

$\mathbf{B}$  is split as the following:

$$\mathbf{B} = \mathbf{P} - \mathbf{L} - \mathbf{U} \quad (7)$$

with  $\mathbf{P}$  being the block diagonal matrix,  $-\mathbf{L}$  the strict lower triangular part, and  $-\mathbf{U}$  the strict upper triangular part. The matrix patterns of  $\mathbf{P}$ ,  $\mathbf{L}$ , and  $\mathbf{U}$  are shown in Fig. 2(d).  $\mathbf{L}$  and  $\mathbf{U}$  are sparse matrices. For the mesh shown in Fig. 2(b),  $\mathbf{L}$  and  $\mathbf{U}$  are bidiagonal matrices.

The matrix  $\mathbf{P}$  serves as an effective preconditioner of  $\mathbf{B}$ . It was shown by numerical experiments that with  $\mathbf{P}$ , the iterative process of solving (5) converges in two iterations for realistic on-chip examples we have tested.

#### C. Iterative Solution and A Proof of Its Convergence

By using  $\mathbf{P}$  as the preconditioner, (5) is solved iteratively as

$$\mathbf{P}x^{(k+1)} = (\mathbf{L} + \mathbf{U})x^{(k)} + d, \quad k \geq 0 \quad (8)$$

with  $k$  denoting the iteration number. When (8) converges,  $x^{(k+1)} = x^{(k)}$ , and hence  $(\mathbf{P} - \mathbf{L} - \mathbf{U})x^{(k+1)} = d$ . From (7), it can be seen that  $x^{(k+1)}$  is the solution of  $\mathbf{B}x = d$ . The following proof demonstrates that the iteration in (8) converges for any right-hand side  $d$  and any initial vector  $x^{(0)}$ .

Define a sequence of iterates of the form

$$x^{(k+1)} = \mathbf{G}x^{(k)} + f \quad (9)$$

in which  $\mathbf{G}$  is a square matrix. It is known that if the spectral radius of  $\mathbf{G}$ , denoted by  $\rho(\mathbf{G})$ , is less than 1, then  $\mathbf{I} - \mathbf{G}$  is nonsingular and the iteration in (9) converges for any  $f$  and  $x^{(0)}$  [14].

From (8), it can be seen that  $\mathbf{G}$  in our iteration scheme is

$$\mathbf{G} = \mathbf{P}^{-1}(\mathbf{L} + \mathbf{U}). \quad (10)$$

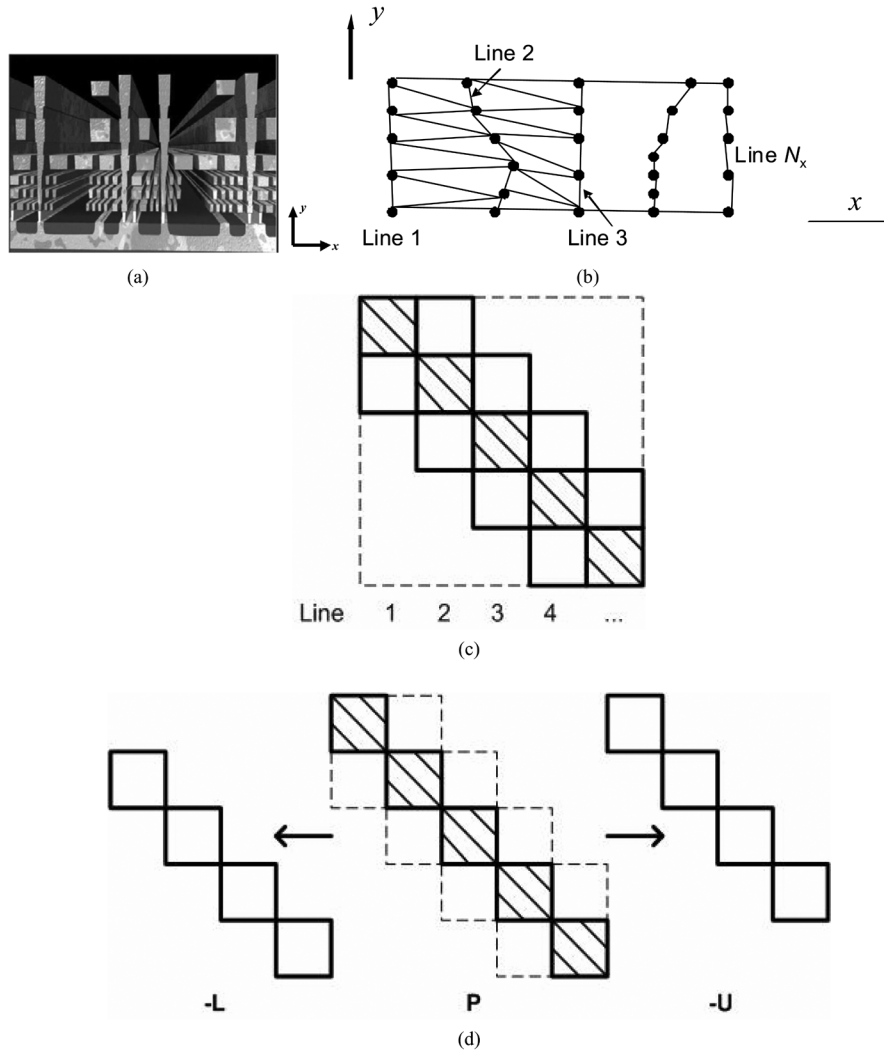


Fig. 2. (a) Example of on-chip structures; (b) mesh and ordering; (c) matrix pattern for (b); and (d) the splitting of (b).

Denoting the eigenvalues of  $\mathbf{G}$  by  $\lambda$ , we have

$$\mathbf{G}v = \lambda v, \quad (11)$$

in which  $v$  denotes the eigenvector. From (10) and (11), we obtain

$$(\mathbf{L} + \mathbf{U})v = \lambda \mathbf{P}v. \quad (12)$$

Taking a norm on both sides of (12), we obtain

$$|\lambda| = \frac{\|(\mathbf{L} + \mathbf{U})v\|}{\|\mathbf{P}v\|} \quad (13)$$

in which  $\|(\cdot)\|$  denotes the norm of  $(\cdot)$ . Without loss of generality, the 1-norm of a vector is used, which is

$$\|g\|_1 = \sum_{i=1}^m |g_i| \quad (14)$$

in which  $m$  is the length of vector  $g$ . If  $\|(\mathbf{L} + \mathbf{U})v\|$  is less than  $\|\mathbf{P}v\|$ , then the modulus of  $\lambda$  is less than 1, then  $\rho(\mathbf{G}) < 1$ , and then the convergence of (8) is proved.

Matrices  $\mathbf{L}$  and  $\mathbf{U}$  are the off-diagonal blocks of  $\mathbf{B}$ , and  $\mathbf{P}$  is the diagonal block. As can be seen from (2), the matrix elements of  $\mathbf{B}$  can be written as

$$\mathbf{B}_{mn}^e = \text{Re}[\mathbf{B}_{mn}^e] + j\text{Im}[\mathbf{B}_{mn}^e] \quad (15)$$

in which

$$\begin{aligned} \text{Re}[\mathbf{B}_{mn}^e] &= l[-k_0^2 \epsilon_r \langle \xi_m, \xi_n \rangle_\Omega + \mu_r^{-1} \langle \nabla \xi_m, \nabla \xi_n \rangle_\Omega] \\ \text{Im}[\mathbf{B}_{mn}^e] &= \omega \mu_0 \sigma l \langle \xi_m, \xi_n \rangle_\Omega \end{aligned} \quad (16)$$

where

$$\langle \xi_m, \xi_n \rangle_\Omega = \begin{cases} \frac{\Delta}{6}, & m = n \\ \frac{\Delta}{12}, & m \neq n \end{cases} \quad (17)$$

and

$$\langle \nabla \xi_m, \nabla \xi_n \rangle_\Omega = \begin{cases} \frac{a_m^2}{4\Delta}, & m = n \\ \frac{-a_m a_n \cos \theta_{mn}}{4\Delta}, & m \neq n. \end{cases} \quad (18)$$

In (17) and (18),  $\Delta$  is the area of a triangular element,  $a_m$  ( $m = 1, 2, 3$ ) is the length of edge  $m$  that is opposite to node  $m$  in the

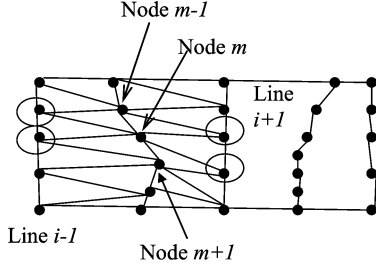


Fig. 3. Illustration of an arbitrary node  $m$  and its connection with other nodes.

triangular element, and  $\theta_{mn}$  is the angle formed between edge  $m$  and edge  $n$ .

It is known that the frequency band of on-chip circuits is from DC to tens of gigahertz. The typical geometrical dimension of on-chip circuits is at  $1 \mu\text{m}$  level. This makes the first part in the square bracket in (16) negligible when compared to the second part. For example, consider 50 GHz and  $0.1 \mu\text{m}$  mesh size

$$k_0^2 \epsilon_r \langle \xi_m, \xi_n \rangle_\Omega \sim 10^{-9}. \quad (19)$$

In contrast, the second part in (16), which is  $\mu_r^{-1} \langle \nabla \xi_m, \nabla \xi_n \rangle_\Omega$ , is an  $O(1)$  quantity as can be seen from (18). Hence,

$$\mathbf{B}_{mn}^e \sim l [\mu_r^{-1} \langle \nabla \xi_m, \nabla \xi_n \rangle_\Omega + j\omega\mu_0\sigma \langle \xi_m, \xi_n \rangle_\Omega] \quad (20)$$

which can be denoted by

$$\mathbf{B}_{mn}^e \sim \begin{cases} s_d - jt & m = n \\ s_o - \frac{jt}{2} & m \neq n \end{cases} \quad (21)$$

in which

$$\begin{aligned} s_d &= l\mu_r^{-1} \frac{a_m^2}{4\Delta} \\ s_o &= -l\mu_r^{-1} \frac{a_m a_n \cos \theta_{mn}}{4\Delta} \\ t &= l\omega\mu_0\sigma \frac{\Delta}{6}. \end{aligned} \quad (22)$$

With the matrix elements of  $\mathbf{B}$  known,  $\|(\mathbf{L} + \mathbf{U})v\|$  and  $\|\mathbf{P}v\|$  can be derived. Since  $\mathbf{P}$  is tridiagonal and symmetric,  $\|\mathbf{P}v\|$  can be written as

$$\|\mathbf{P}v\| = \sum_{m=1}^M |\mathbf{B}_{m,m-1}v_{m-1} + \mathbf{B}_{m,m}v_m + \mathbf{B}_{m,m+1}v_{m+1}|. \quad (23)$$

Consider an arbitrary volume unknown denoted by node  $m$  in Fig. 3, which is a more detailed plot of Fig. 2(b). Assuming that the mesh is reasonably good, and hence the sizes of all the elements are similar, we obtain

$$\mathbf{B}_{m,m} \sim n_e s_d - jn_e t \quad (24)$$

in which  $n_e$  is the number of triangular elements that own node  $m$  as one of their vertices. In (24),  $\mathbf{B}_{m,m}$  is obtained by assembling the contribution from  $n_e$  elements. Based on the unknown

ordering scheme illustrated in Section III-A, node  $m-1$ , node  $m$ , and node  $m+1$  all reside on the same line as shown in Fig. 3. Since node  $m$  and node  $m-1$  can only be simultaneously shared by two triangular elements, and so are node  $m$  and node  $m+1$ ,  $\mathbf{B}_{m,m-1}$ , and  $\mathbf{B}_{m,m+1}$  are assembled from two elemental contributions as

$$\begin{aligned} \mathbf{B}_{m,m-1} &\sim 2s_o - jt \\ \mathbf{B}_{m,m+1} &\sim 2s_o - jt. \end{aligned} \quad (25)$$

Therefore, (23) can be written as

$$\|\mathbf{P}v\| = \sum_{m=1}^M \left| (2s_o - jt)v_{m-1} + (n_e s_d - jn_e t)v_m + (2s_o - jt)v_{m+1} \right|. \quad (26)$$

Since nodes  $m-1$  and  $m+1$  are adjacent to node  $m$ , the eigenfield  $v$  at these two nodes can be estimated by  $v$  at node  $m$ . Hence

$$\begin{aligned} \|\mathbf{P}v\| &\sim \sum_{m=1}^M \left| (2s_o - jt) + (n_e s_d - jn_e t) \right. \\ &\quad \left. + (2s_o - jt) \right| |v_m| \\ &= \sum_{m=1}^M \sqrt{(n_e s_d + 4s_o)^2 + (n_e + 2)^2 t^2} |v_m|. \end{aligned} \quad (27)$$

As shown in Fig. 3, node  $m$  not only interacts with the adjacent two nodes residing on the same line, but also interacts with nodes residing on the line left to it and the line right to it. The interaction with the nodes residing on the left line is characterized by  $\mathbf{L}$  and that with the nodes residing on the right line is characterized by  $\mathbf{U}$ . Therefore,  $\mathbf{L}$  and  $\mathbf{U}$  consist of matrix elements  $\mathbf{B}_{m,n}$  in which node  $n$  resides on the adjacent lines and connects directly to node  $m$  through one edge. Assuming that the number of nodes  $n$  is  $n_p$ ,  $\|(\mathbf{L} + \mathbf{U})v\|$  can be evaluated by

$$\begin{aligned} \|(\mathbf{L} + \mathbf{U})v\| &= \sum_{m=1}^M \left| \sum_{n=1}^{n_p} \mathbf{B}_{mn} v_n \right| \\ &= \sum_{m=1}^M \left| \sum_{n=1}^{n_p} (2s_o - jt)v_n \right|. \end{aligned} \quad (28)$$

Since the  $n_p$  nodes are adjacent to node  $m$ , (28) can be estimated as

$$\|(\mathbf{L} + \mathbf{U})v\| \sim \sum_{m=1}^M n_p \sqrt{4s_o^2 + t^2} |v_m|. \quad (29)$$

Take the mesh shown in Fig. 3 as an example,  $n_p = 4$  (the number of circled nodes) and  $n_e = 6$ , hence, from (27) and (29),  $\|(\mathbf{L} + \mathbf{U})v\| < \|\mathbf{P}v\|$ . In general on-chip applications, since the dielectric stack is multi-layered, the mesh must be partitioned at the dielectric interface (along  $y$ ) as shown in Fig. 4, i.e., no elements can go beyond a dielectric layer. Since each layer is so thin that it generally sustains only one layer of elements, the

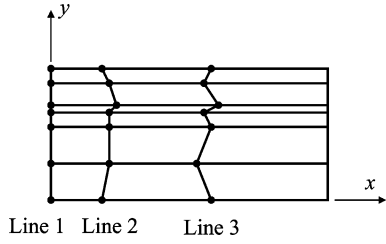


Fig. 4. Illustration of the mesh partition at the dielectric interfaces.

maximum number of nodes in the adjacent lines that a single node can directly interact with is 6 as shown in Fig. 4. Therefore

$$n_p \leq 6. \quad (30)$$

Even if the on-chip dielectric stack is not considered, (30) is what is generally needed to make a good mesh. In addition, there exists the following relationship between  $n_p$  and  $n_e$

$$n_e = 2 \left( \frac{n_p}{2} + 1 \right) = n_p + 2. \quad (31)$$

Hence, from (27) and (29), the following inequality always holds true

$$\|(\mathbf{L} + \mathbf{U})v\| < \|\mathbf{P}v\|. \quad (32)$$

Therefore, from (13),  $\rho(\mathbf{G}) < 1$ . And hence, the iteration in (8) converges for any  $d$  and any initial vector  $x^{(0)}$ .

From (9), it can be seen that

$$x^{(k+1)} - x^{(k)} = \mathbf{G}^k (x^{(1)} - x^{(0)}) \quad (33)$$

hence

$$\begin{aligned} \|x^{(k+1)} - x^{(k)}\| &= \|\mathbf{G}^k (x^{(1)} - x^{(0)})\| \\ &< \|\mathbf{G}\|^k \|x^{(1)} - x^{(0)}\|. \end{aligned} \quad (34)$$

Clearly, the smaller  $\|\mathbf{G}\|$  is, the faster the convergence is. From (10)

$$\|\mathbf{G}\| = \|\mathbf{P}^{-1}(\mathbf{L} + \mathbf{U})\| < \|\mathbf{P}^{-1}\| \|\mathbf{L} + \mathbf{U}\| = c \frac{\|\mathbf{L} + \mathbf{U}\|}{\|\mathbf{P}\|} \quad (35)$$

in which  $c$  is the condition number of matrix  $\mathbf{P}$ . When estimating  $\|\mathbf{G}\|$ ,  $c$  can be treated as a constant. This is because  $\mathbf{P}$  is block diagonal, and each diagonal block is formed by volume unknowns residing on one line along the stack-growth direction ( $y$ -direction) as shown in Fig. 2(b). Hence, each diagonal block of  $\mathbf{P}$  is mainly determined by the material and geometrical information of the stack. For on-chip circuits, the stack is fixed for each processing technology node. Hence, the condition number of matrix  $\mathbf{P}$  can be estimated as a constant. As a result, from (35), the ratio of  $\|\mathbf{L} + \mathbf{U}\|$  to  $\|\mathbf{P}\|$  serves as a good measure of the convergence rate of the proposed iterative solution.

Since  $\mathbf{B}$  is a block tridiagonal matrix, (8) can be simplified to

$$\mathbf{P}_{ii} x_i^{(k+1)} = \mathbf{L}_{i,i-1} x_{i-1}^{(k)} + \mathbf{U}_{i,i+1} x_{i+1}^{(k)} + d_i, \quad 1 \leq i \leq n \quad (36)$$

in which  $i$  denotes the line index,  $\mathbf{P}_{ii}$  denotes the  $i$ th diagonal block of  $\mathbf{P}$ , which is formed by volume unknowns on line  $i$ . The operations in (36) involve an inverse of each block diagonal matrix,  $\mathbf{P}_{ii}$ , and two sparse matrix vector multiplications. Since  $\mathbf{L}_{i,i-1}$  and  $\mathbf{U}_{i,i+1}$  are extremely sparse (for example, two nonzero elements per row), the matrix vector multiplication in (36) can be obtained efficiently in linear complexity with a constant multiplier generally less than 5. The remaining task is to compute  $\mathbf{P}_{ii}^{-1}b$  efficiently, with  $b$  being the right-hand side of (36).

#### D. Fast Direct Computation of $\mathbf{P}^{-1}$ In Linear Complexity

Matrix  $\mathbf{P}$  is the block diagonal matrix of  $\mathbf{B}$ . It is composed of  $N_x$  block matrices  $\mathbf{P}_{ii}$  ( $i = 1, 2, \dots, N_x$ ) as shown in Fig. 2(c) and (d). Each  $\mathbf{P}_{ii}$  is tridiagonal because each volume unknown residing on one line has only crosstalk with itself and its adjacent two neighbors. The inverse of a tridiagonal matrix belongs to the class of semiseparable matrices. Assuming  $\mathbf{P}_{ii}$  is of order  $N_y$ , there exist two sequences  $\{u_i\}$ ,  $\{v_i\}$ ,  $i = 1, 2, \dots, N_y$  such that  $\mathbf{P}_{ii}^{-1}$  can be written as follows [15]:

$$\mathbf{P}_{ii}^{-1} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & & u_1 v_{N_y} \\ u_1 v_2 & u_2 v_2 & & u_2 v_{N_y} \\ u_1 v_3 & u_2 v_3 & & u_3 v_{N_y} \\ & \vdots & \ddots & \vdots \\ u_1 v_{N_y} & u_2 v_{N_y} & & u_{N_y} v_{N_y} \end{bmatrix}. \quad (37)$$

Denoting  $\mathbf{P}_{ii}$  as  $\text{tri}(a_{1:N_y}, -b_{1:N_y-1})$ , the sequences  $\{u_i\}$ ,  $\{v_i\}$  can be generated in  $O(N_y)$  operations as

$$\begin{aligned} d_{N_y} &= a_{N_y}, & d_i &= a_i - \frac{b_{i+1}^2}{d_{i+1}}, & i &= N_y - 1, \dots, 1 \\ v_1 &= \frac{1}{d_1}, & v_i &= v_{i-1} \frac{b_i}{d_i}, & i &= 2, \dots, N_y \\ \chi_1 &= a_1, & \chi_i &= a_i - \frac{b_i^2}{\chi_{i-1}}, & i &= 2, \dots, N_y \\ u_{N_y} &= \frac{1}{\chi_{N_y} v_{N_y}}, & u_{N_y-i} &= \frac{b_{N_y-i+1}}{\chi_{N_y-i}} u_{N_y-i+1}, \\ & & i &= 1, \dots, N_y - 1. \end{aligned} \quad (38)$$

Therefore,  $\mathbf{P}_{ii}^{-1}$ , and hence  $\mathbf{P}^{-1}$ , can be obtained in linear complexity. In addition,  $\mathbf{P}_{ii}^{-1}$  is stored in  $\{u_i\}$ ,  $\{v_i\}$ ,  $i = 1, 2, \dots, N_y$  sequences rather than a full matrix of size  $N_y^2$  in (37). Hence, although  $\mathbf{P}_{ii}^{-1}$  is dense, the memory consumption of storing  $\mathbf{P}_{ii}^{-1}$ , and hence  $\mathbf{P}^{-1}$ , is linearly proportional to the matrix size.

#### E. Fast Evaluation of $\mathbf{P}^{-1}b$ in Linear Complexity

Since the inverse of  $\mathbf{P}_{ii}$  is dense, apparently the evaluation of  $\mathbf{P}_{ii}^{-1}b$  requires  $O(N_y^2)$  operations. In fact, it can be obtained in  $O(N_y)$  operations. The matrix vector multiplication  $\mathbf{P}_{ii}^{-1}b$  is evaluated as shown in (39). Consider the upper triangular part. The underlying terms are those that do not need to be computed, as they can be reused from the previous step if one starts from the last row. As a result, for each row, there is only one multiplication  $v_i b_i$  ( $i$  is the row index) that needs to be calculated together with one summation and one multiplication with  $u_i$ . The same is true for the strict lower triangular part. The underlying terms

TABLE I  
COMPARISON OF THE S-PARAMETERS

Frequency	1GHz	12.8GHz	50GHz
S-parameters (Direct Reduction)	S11 = -0.2853-0.4620j S12 = 0.5909-0.4746j	S11 = -0.8530-0.4075j S12 = -0.7621-0.1167j	S11 = -0.8295-0.2627j S12 = 0.7780+0.5780j
S-parameters (Fast Reduction)	S11 = -0.2857-0.4622j S12 = 0.5905-0.4747j	S11 = -0.8531-0.4094j S12 = -0.7635-0.1165j	S11 = -0.8297-0.2687j S12 = 0.7770+0.5497j

do not need to be recomputed if one starts from the second row and proceed to the following rows. For each row, there is only one multiplication  $u_{i-1}b_{i-1}$  that needs to be calculated together with one summation with the previous value and one multiplication with  $v_i$ . Hence, the cost of  $\mathbf{P}_{ii}^{-1}b$  scales as  $O(N_y)$  with a constant multiplier equal to 7. See (39) at the bottom of the page.

As a result, the computational cost of solving  $\mathbf{B}x = d$  scales linearly with the number of unknowns. Hence, the total cost of computing  $\mathbf{B}^{-1}\mathbf{D}^T$  scales as  $O(M^2)$  because there exist  $O(M)$  columns in  $\mathbf{D}^T$ . As a result, the cost of 3-D to 2-D reduction scales as  $O(M^2)$ , speeding up the direct reduction by  $O(M)$  times.

#### IV. NUMERICAL RESULTS

The performance of the proposed fast reduction algorithms was tested on a test-chip 3-D interconnect structure [5], [16]. The structure is of 300  $\mu\text{m}$  width and 2000  $\mu\text{m}$  length. Table I compares the accuracy of the proposed fast reduction with that of the direct reduction. S-parameters sampled at three different frequency points, 1 GHz, 12.8 GHz, and 50 GHz, are chosen for comparison. As shown in Table I, good agreement is observed.

The number of single-layer volume unknowns,  $M$ , is 300 in this simulation. The initial vector of the iteration in (8) is chosen as 0. The iteration converges in two steps with the relative error in  $x$  less than 0.001. The relative error is defined by

$$\text{Relative Error} = \frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|}$$

$$\text{where, } \|x^{(k)}\| = \sum_i |x_i^{(k)}|. \quad (40)$$

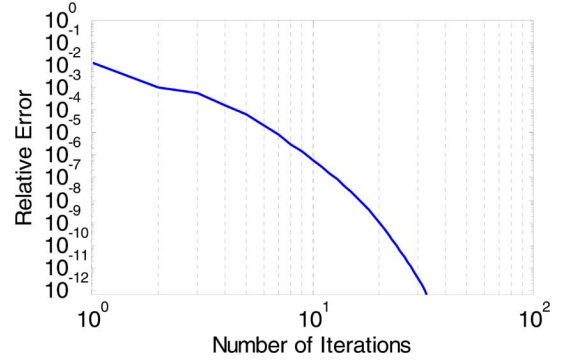


Fig. 5. Relative error versus the number of iterations for a test-chip interconnect structure.

In (40), the difference between  $x^{k+1}$  and  $x^k$  is used to assess the error instead of that between  $x^{k+1}$  and true  $x$ . This is because when  $x^{k+1}$  is equal to  $x^k$ ,  $x^{k+1}$  is the solution of  $\mathbf{B}x = d$  as can be seen from (7) and (8).

In Fig. 5, the relative error of  $x$  is plotted as a function of the number of iterations at 12.8 GHz. Fast convergence can be observed. The accuracy of  $10^{-6}$  is achieved within 10 iterations. The  $\|\mathbf{L} + \mathbf{U}\|$  and  $\|\mathbf{P}\|$  were also numerically evaluated. It was shown that  $\|\mathbf{L} + \mathbf{U}\|$  was  $5.084241 \times 10^{14}$  and  $\|\mathbf{P}\|$  was  $1.759374 \times 10^{17}$ , which contributed to the fast convergence of the proposed iterative scheme. The 1-norm of a matrix is used which is the maximum sum of the absolute values in each row/column.

In Fig. 6, the S-parameters were plotted over the entire frequency band, which revealed an excellent agreement between the proposed fast reduction and the direct reduction as well as the measured data. When performing the reduction, all the columns of  $\mathbf{D}^T$  were processed. It was shown that the number

$$\mathbf{P}_{ii}^{-1}b = \begin{pmatrix} u_1v_1u_1v_2u_1v_3 & \cdots & u_1v_{N_y} \\ u_1v_2u_2v_2u_2v_3 & \cdots & u_2v_{N_y} \\ u_1v_3u_2v_3u_3v_3 & \cdots & u_3v_{N_y} \\ \vdots & \ddots & \vdots \\ u_1v_{N_y}u_2v_{N_y}u_3v_{N_y} & \cdots & u_{N_y}v_{N_y} \end{pmatrix} \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{N_y} \end{Bmatrix}$$

$$= \begin{Bmatrix} u_1(v_1b_1 + v_2b_2 + v_3b_3 + v_4b_4 + v_5b_5 + \cdots + v_{N_y}b_{N_y}) \\ v_2u_1b_1 + u_2(v_2b_2 + v_3b_3 + v_4b_4 + v_5b_5 + \cdots + v_{N_y}b_{N_y}) \\ v_3(u_1b_1 + u_2b_2) + u_3(v_3b_3 + v_4b_4 + v_5b_5 + \cdots + v_{N_y}b_{N_y}) \\ \cdots \cdots \cdots \\ v_{N_y}(u_1b_1 + u_2b_2 + \cdots + u_{N_y-2}b_{N_y-2} + u_{N_y-1}b_{N_y-1}) + u_{N_y}v_{N_y}b_{N_y} \end{Bmatrix} \quad (39)$$

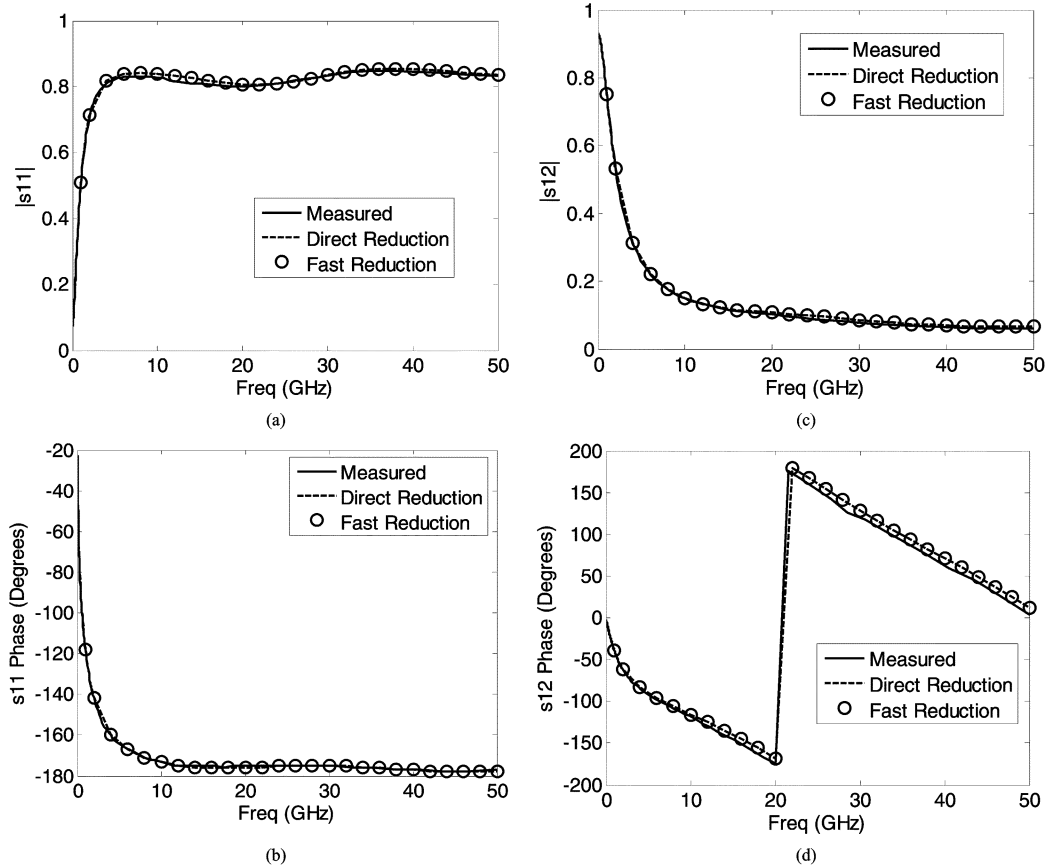


Fig. 6. S-parameters of an on-chip interconnect structure of length  $2000 \mu\text{m}$ . (a) S11 magnitude. (b) S11 phase. (c) S12 magnitude. (d) S12 phase.

of iterations required by different columns to reach the same accuracy could be different. But within 2 iterations, the S-parameters shown in Table I and Fig. 6 can be obtained. There exist some columns that require a few more iterations. However, it was observed that the norm of these columns was orders-of-magnitude less than that of the other columns, and the error in these columns has little influence on the accuracy of the reduced matrix in (3).

The test-chip interconnect structure was then modified to examine whether the same performance could be observed in a different structure. The original metal plane in metal-3 layer, which was continuous along the  $2000 \mu\text{m}$  length, was broken every other  $100 \mu\text{m}$  along the length. As a result, the metal-3 layer was filled by 10 metal plates, each of which was  $100 \mu\text{m}$  long. The spacing between two adjacent plates was  $100 \mu\text{m}$ . The proposed fast reduction and the direct reduction were used to simulate this structure. In Table II, the S-parameters sampled at three different frequency points are given, which reveal a very good agreement. Again, only two iterations were used to generate the results shown in Table II. In Fig. 7, the relative error with respect to the number of iterations was plotted at 1 GHz, 12.8 GHz, and 50 GHz, respectively. Fast convergence can be observed. The ratio of  $\|(\mathbf{L} + \mathbf{U})\|$  to  $\|\mathbf{P}\|$  at the three different frequency points was given in Table III. Clearly, the ratio increases when frequency increases, which explains the slower convergence at the high frequency end observed in Fig. 7. With the accuracy validated, the CPU time of the proposed fast reduction was com-

pared against that of the direct reduction. In Table IV, the CPU time for solving (5) for 1000 right-hand sides is listed with respect to the number of unknowns for both schemes. It is clear that the proposed technique is faster than the direct one. In addition, the speedup increases with the number of unknowns  $M$ , i.e., the larger the problem size, the more efficient the proposed technique. This is because the time complexity of the proposed technique in solving (5) is strict  $O(M)$  as can be seen from the third row in Table IV. In contrast, existing advanced sparse solvers such as a multifrontal method has an increased slope in the complexity curve, i.e., the slope increases when the number of unknowns increases. For example, when  $M$  was increased to 1 188 288, it took the multifrontal-method-based direct reduction more than 1 h to solve (5) for 1000 right-hand sides, while the proposed technique took about 6 min. Asymptotically, the complexity of the multifrontal solver reaches  $O(M^2)$  when solving a block-tridiagonal matrix. The computing platform used was an IBM system x3550 with two Intel Woodcrest dual core processors.

In Fig. 8, the relative error versus the number of iterations was plotted for  $M = 2664$  case, compared to Fig. 5, similar convergence could be observed. In Fig. 9, the relative error versus the number of iterations was plotted for  $M = 24048$  case. Again, fast convergence is observed. Here, we observe that the convergence rate of the proposed iterative solution has little dependence on the number of unknowns. This is because in on-chip problems, the stack is fixed. Take the computational



TABLE II  
COMPARISON OF THE S-PARAMETERS FOR A MODIFIED TEST-CHIP INTERCONNECT STRUCTURE

Frequency	1GHz	12.8GHz	50GHz
S-parameters (Direct Reduction)	S11 = -2.7860e-1- 4.6041e-1j S12 = 5.9861e-1- 4.7209e-1j	S11 = -8.4876e-1- 2.0240e-2j S12 = -8.5060e-2- 1.3239e-1j	S11 = -8.1403e-1- 5.3027e-2j S12 = 8.1759e-2- 5.1072e-2j
S-parameters (Fast Reduction)	S11 = -2.7729e-1- 4.5979e-1j S12 = 5.9985e-1- 4.7167e-1j	S11 = -8.4866e-1- 1.9134e-2j S12 = -8.4631e-2- 1.3277e-1j	S11 = -8.1180e-1- 4.8064e-2j S12 = 8.3086e-2- 4.9891e-2j

TABLE III  
RATIO OF  $\|(\mathbf{L} + \mathbf{U})\|$  TO  $\|\mathbf{P}\|$  AT THE THREE DIFFERENT FREQUENCY POINTS

Frequency	1GHz	12.8GHz	50GHz
$\ (\mathbf{L} + \mathbf{U})\  / \ \mathbf{P}\ $	2.96e-4	2.9e-3	1.0e-2

TABLE IV  
COMPARISON OF THE CPU TIME

$M$	2664	4248	12168	24048
CPU time (Direct Reduction)	2.45s	5.18s	16.15s	40.89s
CPU time (Fast Reduction)	0.49s	0.78s	2.25s	4.85s
Speedup	5.0	6.64	7.18	8.43

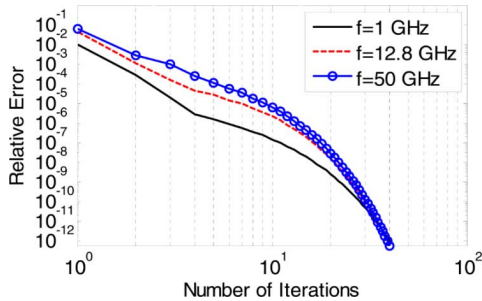


Fig. 7. Relative error versus the number of iterations for a modified test-chip interconnect structure.

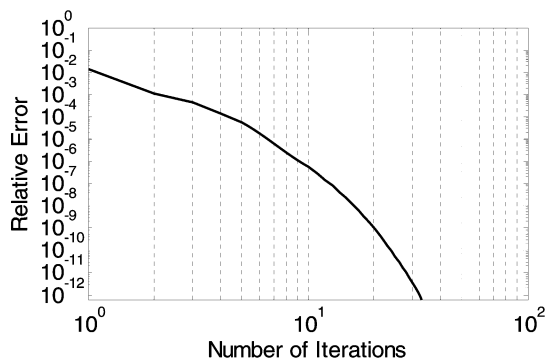


Fig. 8. Relative error versus the number of iterations for the test-chip interconnect structure ( $M = 2664$ ).

domain shown in Fig. 4 as an example, the discretization along the stack-growth direction ( $y$ -direction) is fixed, when the problem size is increased, the problem size is increased along the  $x$  direction only. Based on the definition of  $\mathbf{P}$ ,  $\mathbf{L}$ , and  $\mathbf{U}$ , the  $\|(\mathbf{L} + \mathbf{U})\|$  and  $\|\mathbf{P}\|$  are determined by the matrix

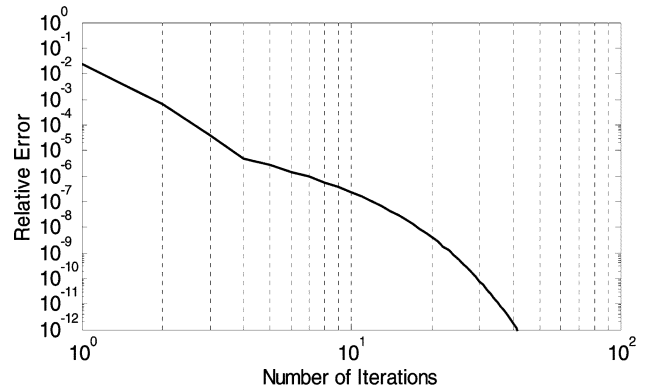


Fig. 9. Relative error versus the number of iterations for the test-chip interconnect structure ( $M = 24048$ ).

characteristics in two adjacent segments along  $x$  direction. The dielectric stack is the same in all the segments along  $x$ . What is different in each segment is the conductor location. But where to place conductors is also constrained in an on-chip circuit. In all the layers present in a stack, only metal layers can be used to place conductors. Hence, similar conductor and dielectric configuration is encountered no matter the problem size is small or large when the problem size is increased along  $x$ . Hence, the ratio of  $\|(\mathbf{L} + \mathbf{U})\|$  to  $\|\mathbf{P}\|$  is at the same level (at each frequency point), which results in a similar convergence rate irrespective of the number of unknowns.

## V. CONCLUSION

In this paper, fast reduction algorithms were developed to efficiently reduce a 3-D layered system matrix to a 2-D layered one in the framework of the frequency-domain layered finite element method. These algorithms include a structuring

of the volume-unknown-based matrix, an effective preconditioner  $\mathbf{P}$ , a direct solution of  $\mathbf{P}^{-1}$  in linear complexity, and a fast computation of  $\mathbf{P}^{-1}\mathbf{b}$  in linear complexity. With these fast algorithms, the iterative solution of the volume-unknown-based matrix equation converges in a few iterations in realistic on-chip structures. In addition, in each iteration the computational cost scales linearly with the number of unknowns. As a result, the volume-unknown-based matrix equation is solved in linear complexity with a small constant (less than 25) in front of the number of unknowns. The reduction from a 3-D layered system to a 2-D layered one is hence accelerated significantly. In addition, the proposed fast reduction algorithms apply to any arbitrarily-shaped multilayer structure.

#### ACKNOWLEDGMENT

The authors would like to thank Dr. M. J. Kobrinsky and Dr. S. Chakravarty at Intel Corporation for providing measurement data.

#### REFERENCES

- [1] 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. Computer-Aided Design*, vol. 20, no. 7, pp. 877–887, Jul. 2001.
- [2] A. Rong, A. C. Cangellaris, and L. Dong, "Comprehensive broadband electromagnetic modeling of on-chip interconnects with a surface discretization-based generalized PEEC model," in *Proc. IEEE 12th Topical Meeting Electrical Performance Electron. Packag. (EPEP)*, 2003, pp. 367–370.
- [3] Z. H. Zhu, B. Song, and J. K. White, "Algorithms in fastimp: A fast and wideband impedance extraction program for complicated 3D geometries," *IEEE Trans. Computer-Aided Design*, vol. 24, no. 7, pp. 981–998, Jul. 2005.
- [4] D. Jiao, S. Chakravarty, and C. Dai, "A layered finite element method for high-frequency modeling of large-scale three-dimensional on-chip interconnect structures," in *Proc. IEEE 15th Topical Meeting Electrical Performance Electronic Packag. (EPEP)*, Oct. 2006, pp. 313–316.
- [5] 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 Propag.*, vol. 55, no. 2, pp. 422–432, Feb. 2007.
- [6] 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 *Proc. Int. Conf. Simulation Semiconductor Processes Devices*, 2003, pp. 39–42.
- [7] S. Kapur and D. E. Long, "Large-scale full-wave simulation," in *Proc. Design Automation Conf.*, 2004.
- [8] D. Gope, A. E. Ruehli, C. Yang, and V. Jandhyala, "(S)PEEC: Time- and frequency-domain surface formulation for modeling conductors and dielectrics in combined circuit electromagnetic simulations," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 6, pp. 2453–2464, Jun. 2006.
- [9] Z. G. Qian, J. Xiong, L. Sun, I. T. Chiang, W. C. Chew, L. J. Jiang, and Y. H. Chu, "Crosstalk analysis by fast computational algorithms," in *Proc. IEEE 14th Topical Meeting Electrical Performance Electron. Packag.*, 2005, pp. 367–370.
- [10] F. Ling, V. I. Okhamtovski, W. Harris, S. McCracken, and A. Dengi, "Large-scale broadband parasitic extraction for fast layout verification of 3D RF and mixed-signal on-chip structures," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 1, pp. 264–273, Jan. 2005.
- [11] H. Gan and D. Jiao, "A fast and high-capacity electromagnetic solution for high-speed IC design," in *Proc. IEEE/ACM 2007 Int. Conf. Computer-Aided Design (ICCAD)*, 2007, pp. 1–6.
- [12] Z. Y. Yuan, Z. F. Li, and M. L. Zou, "Computer-aided analysis of on-chip interconnects near semiconductor substrate for high-speed VLSI," *IEEE Trans. Comput. Aided Des.*, vol. 19, no. 9, pp. 990–998, Sep. 2000.
- [13] UMFPAK [Online]. Available: <http://www.cise.ufl.edu/research/sparse/umfpack/>
- [14] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA: SIAM, 2003.
- [15] G. Meurant, "A review on the inverse of symmetric tridiagonal and block tridiagonal matrices," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 3, pp. 707–728, Jul. 1992.
- [16] 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. Adv. Packag.*, vol. 28, no. 1, pp. 57–62, Feb. 2005.
- [17] J. M. Jin, *The Finite Element Method in Electromagnetics*, 1st ed. New York: Wiley, 1993.



**Feng Sheng** received the B.S. degree in electronic and information engineering from Zhejiang University, Hangzhou, China, in 2006. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

In August 2006, he 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.



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

She then worked in the Technology CAD Division, Intel Corporation, until September 2005 as a Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. In September 2005, she joined Purdue University, West Lafayette, IN, as an Assistant Professor in the School of Electrical and Computer Engineering. She has authored two book chapters and over 90 papers in refereed journals and international conferences. Her current research interests include high frequency digital, analog, 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 the Jack and Cathie Kozik Faculty Start-up Award, which recognizes an outstanding new faculty member in Purdue ECE. She also received an ONR award through Young Investigator Program in 2006. In 2004, she received the Best Paper Award from Intel's annual corporate-wide technology conference (Design and Test Technology Conference) for her work on generic broadband model of high-speed circuits. In 2003, she won the Intel Logic Technology Development (LTD) Divisional Achievement Award in recognition of her work on the industry-leading BroadSpice modeling/simulation capability for designing high-speed microprocessors, packages, and circuit boards. She was also awarded the Intel Technology CAD Divisional Achievement Award for the development of innovative full-wave solvers for high frequency IC design. In 2002, she was awarded by Intel Components Research the Intel Hero Award (Intel-wide she was the tenth recipient) for the timely and accurate two- and three-dimensional full-wave simulations. She also won the Intel LTD Team Quality Award for her outstanding contribution to the development of the measurement capability and simulation tools for high frequency on-chip cross-talk. She was the winner of the 2000 Raj Mittra Outstanding Research Award given her by the University of Illinois at Urbana-Champaign. She has served as the reviewer for many IEEE journals and conferences.