Fast Structure-Aware Direct Time-Domain Finite-Element Solver for the Analysis of Large-Scale On-Chip Circuits

Woochan Lee and Dan Jiao, *Senior Member, IEEE*

***Abstract*— A fast time-domain finite-element algorithm is developed for the analysis and the design of very large-scale on-chip circuits. The structure specialty of on-chip circuits, such as Manhattan geometry and layered permittivity, is preserved in the proposed algorithm. As a result, the large-scale matrix solution encountered in the 3-D circuit analysis is turned into a simple scaling of the solution of a small 1-D tridiagonal matrix, which can be obtained in linear (optimal) complexity with negligible cost. Furthermore, the time step size is not sacrificed, and the total number of time steps to be simulated is also significantly reduced, thus achieving a total cost reduction in the CPU time. Applications to the simulation of very large-scale on-chip circuit structures on a single core have demonstrated the superior performance of the proposed method.**

***Index Terms*— DC analysis, fast solvers, on-chip circuits, on-die power grids, time-domain finite-element method (TDFEM), transient analysis.**

I. Introduction

THE analysis and the design of on-chip circuits are of critical importance to the overall performance of integrated circuits (IC) and systems [1]–[13]. Many on-chip circuit structures, such as a power grid and a clock grid, constitute a very large-scale problem in modeling and simulation, the size of which also continuously grows with the advancement of the processing technology. A straightforward solution of a global on-chip interconnect network would result in a numerical system that is beyond the capability of existing most powerful computational resources.

Different from many other engineering problems, the structure of an on-chip circuit is special in the sense that its geometry is of Manhattan type and its dielectrics are layered. Not taking advantage of these specialties would unnecessarily slow down the computation; however, preserving the structure specialties and taking advantage of them in a numerical solution is not a straightforward task either. Take the layered dielectrics as an example. If one employs a frequency-domain solver [7], [12], [15] to analyze the on-chip circuits, the resultant system matrix is composed of permittivity-, conductivity-, and permeability-related terms. Since conductivity and permeability are not layered, the layered property of the permittivity cannot be preserved and taken advantage of in the solution of the frequency-domain system matrix. The same is true for an implicit time-domain method [6], [11]. If one employs an explicit time-domain method [1], [2], [8], [13], although only the weighted sum of the permittivity- and conductivity-related matrices needs to be solved, since the conductivity is not layered as the layout structure is different in different dielectric layers, the layered structure of the permittivity is also lost in the numerical solution of an explicit time-domain method.

There have been structure-preserving algorithms developed to exploit the structure specialty of the on-chip circuits. In [1], a time-domain layered finite-element reduction-recovery (LAFE-RR) method [1] was developed to solve large-scale IC design problems at high frequencies. In this method, the layered property of dielectrics is employed to perform system reduction analytically from multiple layers to single layer, which is a 2-D numerical system. Considering the Manhattan geometry, the 2-D single-layer system is further reduced to single line that is 1-D in the hierarchical FE-RR method [8]. However, since the conductivity is not layered, in [1] and [8], the conductivity-related term is moved to the right-hand side of the system matrix equation to enable the analytical system reduction. This makes the time step required for a stable time-domain simulation much smaller than that permitted by a traditional explicit time-domain method, and hence slowing down the overall computation. In [2], a fast marching method is developed to address the time step issue. In this method, the conductivity-related term is kept to the left-hand side of the marching equation in time, and a preconditioner that permits an LAFE-RR solution is constructed to iteratively solve the system matrix with an expedited convergence. The preconditioner is built by replacing some metal layers with solid metal planes, the performance of which is problem-dependent for accelerating the original matrix solution. In [9], an efficient structure-aware method was developed to preserve the layered permittivity and the Manhattan-type geometry in an explicit time-domain finite-element method (TDFEM) for analyzing very large-scale IC problems. Different from [2], the method is a direct solution that avoids the common problems of an iterative solution.

Manuscript received December 15, 2014; revised April 24, 2015 and July 30, 2015; accepted August 11, 2015. Date of current version October 2, 2015. This work was supported in part by the National Science Foundation through the Division of Computing and Communication Foundations under Award 1065318, in part by the Intel Corporation, and in part by the Defense Advanced Research Projects Agency under Award N00014-10-1-0482B. Recommended for publication by Associate Editor M. S. Tong upon evaluation of reviewers' comments.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: lee1231@purdue.edu; 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/TCPMT.2015.2472403

However, the method requires the computation of a matrix exponential. This matrix exponential can be computed from the sum of a finite number of terms with guaranteed convergence, irrespective of the choice of time step. However, numerically, for the sum to converge fast with a fewer number of terms, the time step used in [9], though much larger than that in [1], is still smaller than that allowed by a traditional central-difference-based explicit TDFEM method. As a consequence, the overall computational efficiency is compromised.

In this paper, we present an efficient direct time-domain finite-element solver that preserves the layered property of the permittivity and the Manhattan-type structure present in the on-chip circuits without sacrificing the time step size. As a result, at each time step, the matrix solution of a large-scale 3-D circuit problem becomes a simple scaling of the solution of a small 1-D tridiagonal matrix, which can be obtained in linear (optimal) complexity with negligible cost. Meanwhile, the time step is not reduced compared with that of the central-difference-based explicit TDFEM. Furthermore, the total number of time steps to be simulated is also significantly reduced, thereby achieving a total cost reduction in the CPU time. Comparisons with conventional solutions have demonstrated the obvious advantages of the proposed method in computational capacity and efficiency.

II. Proposed Method

A. General Idea

A time-domain FEM solution of the second-order vector-wave equation for an IC problem results in the following linear system of equations [14]:

$$\mathbf{T}\ddot{u}(t)+\mathbf{R}\dot{u}(t)+\mathbf{S}u(t)=\dot{I}(t) \tag{1}$$

in which $\mathbf{T}$ is a mass matrix, $\mathbf{R}$ is associated with conductivity, $\mathbf{S}$ is a stiffness matrix, u is the field solution vector, and I is a vector of current sources. The single dot above a letter denotes a first-order time derivative, while the double dots denote a second-order time derivative. $\mathbf{T}$, $\mathbf{R}$, and $\mathbf{S}$ are assembled from their elemental contributions as follows:

$$\mathbf{T}^e=\mu_0\varepsilon\langle\mathbf{N}_i,\mathbf{N}_j\rangle \tag{2}$$

$$\mathbf{R}^e=\mu_0\sigma\langle\mathbf{N}_i,\mathbf{N}_j\rangle \tag{3}$$

$$\mathbf{S}^e=\mu_r^{-1}\langle\nabla\times\mathbf{N}_i,\nabla\times\mathbf{N}_j\rangle \tag{4}$$

where ε is the permittivity, σ is the conductivity, μ_0 is the free-space permeability, μ_r is the relative permeability, $\mathbf{N}$ is the vector basis employed to represent electric field $\mathbf{E}$, and $\langle\cdot,\cdot\rangle$ denotes an inner product.

A central-difference-based discretization of (1) in time results in the following explicit updating equation:

$$\left(\mathbf{T}+\frac{\Delta t}{2}\mathbf{R}\right)u^{n+1}=2\mathbf{T}u^n-\mathbf{T}u^{n-1}-\Delta t^2\mathbf{S}u^n+\frac{\Delta t}{2}\mathbf{R}u^{n-1}-\Delta t^2\dot{I}^n \tag{5}$$

where the field solution at the most advanced time step, u^{n+1}, is obtained from the field solutions at the previous two time steps, u^n and u^{n-1}, step by step. Obviously, the updating of (5) in time requires a matrix solution of $(\mathbf{T}+(\Delta t/2)\mathbf{R})$. This matrix does not have a layered property, since $\mathbf{R}$, as shown in (3), is related to conductivity, and conductivity is not layered. If one moves the $\mathbf{R}$-based term from the left-hand side of (5) to the right-hand side, i.e., let $\mathbf{R}$ be associated with the field value at the previous time step, the resultant time step for a stable simulation is too small to be used for an efficient simulation [2]. To fully exploit the layered property of the permittivity distribution, we propose a fast algorithm that turns a matrix solution to a simple scaling without sacrificing the time step size as follows.

We begin by rewriting (5) as

$$\mathbf{K}u^{n+1}=\tilde{b} \tag{6}$$

in which

$$\mathbf{K}=\mathbf{I}+\frac{\Delta t}{2}\mathbf{T^{-1}R} \tag{7}$$

$$\tilde{b}=2u^n-u^{n-1}-(\Delta t)^2\mathbf{T}^{-1}\left(\mathbf{S}u^n-\frac{1}{2\Delta t}\mathbf{R}u^{n-1}+\dot{I}^n\right). \tag{8}$$

To take advantage of the layered property of materials, we propose to obtain the inverse of $\mathbf{K}$ from the following series expansion:

$$\mathbf{K^{-1}}=(\mathbf{I}+\mathbf{A})^{-1}=\mathbf{I}-\mathbf{A}+\mathbf{A^2}-\mathbf{A^3}+\mathbf{A^4}-\mathbf{A^5}+\cdots \tag{9}$$

where

$$\mathbf{A}=\frac{\Delta t}{2}\mathbf{T^{-1}R}. \tag{10}$$

However, series (9) would not converge unless the following condition is satisfied:

$$\|\mathbf{A}\|<1 \tag{11}$$

that is, the norm of $\mathbf{A}$ is <1. Unfortunately, this condition is generally not satisfied in the on-chip circuits, with a large time step Δt used in a central-difference-based TDFEM scheme like (5). This is because the metal conductivity of the on-chip circuits is high, the typical value of which is in the order of 10^7 S/m, and $\|\mathbf{T^{-1}R}\|$ is proportional to σ/ε as can be seen from (2) and (3). With a high conductivity σ, one has to use a very small time step Δt to make $\|\mathbf{A}\|<1$, rendering the time-domain simulation of (1) computationally expensive.

To overcome the aforementioned difficulty, we propose to reduce the conductivity σ and increase the permittivity ε, such that $\|\mathbf{T^{-1}R}\|$ is reduced to such a value that (11) is satisfied. Apparently, this change is not feasible, because the material parameters are altered, and hence, the original structure is completely changed. However, based on the fact that the on-chip circuit's response is dominated by static modes in a fairly wide range of frequencies from zero to a few gigahertz [15] and the space distribution of static fields does not change when the permittivity and conductivity are scaled, we can modify permittivity and conductivity, so that Δt can be enlarged to a desired value while (11) is still satisfied. To explain, the solution of (1) is governed by the following quadratic eigenvalue problem:

$$(\lambda^2\mathbf{T}+\lambda\mathbf{R}+\mathbf{S})V=0 \tag{12}$$

in which λ is the eigenvalue and V is the eigenvector. The field solution of (1) at any time is nothing but a linear superposition of the eigenvectors in (12). The eigenvectors whose eigenvalues are zero are termed dc modes or static modes. They represent one kind of fundamental space variations of the fields in the given structure, which satisfies Maxwell's equations as well as all the boundary conditions such as those at the material interface and on the truncation boundary. As quantitatively analyzed in [15], for small electrical sizes (true for many on-chip structures), the solution of (1) is dominated by dc modes, whereas the contributions from full-wave modes are negligible. To see this point more clearly, the solution of (1) in frequency domain can be written as

$$u(\omega) = \mathbf{V}(\Lambda + j\omega)^{-1}\mathbf{V}^{-1}\tilde{I}$$

where $\mathbf{V}$ is the eigenvector matrix of (12), Λ is the diagonal matrix of the eigenvalues in (12), and $\tilde{I}$ is the current source in frequency domain. The smallest nonzero eigenvalue of (12), which corresponds to the first full-wave mode, has a magnitude proportional to $\pi c/l_{\max}$, where $l_{\max}$ is the largest geometrical dimension of the object and c is the speed of light. For the on-chip circuits, $l_{\max}$ is, in general, less than 1 mm, resulting in the smallest nonzero eigenvalue no less than $3\mathrm{e}+11$ rad/s. Hence, for ω ranging from zero to a few gigahertz, the effects of full-wave modes are not obvious, and the field solution is dominated by the dc modes whose eigenvalues are zero.

Since dc modes have eigenvalues $\lambda = 0$, they satisfy

$$\mathbf{S}V = 0. \tag{13}$$

In other words, the space distribution of the dc eigenmode V makes $\mathbf{S}V$ vanish, which also agrees with physics, since the curl of static $\mathbf{E}$ fields is zero. Since $\mathbf{S}$ is only related to permeability as can be seen from (4), the field distributions of dc modes do not depend on the specific values of permittivity and conductivity. Hence, we can use this fact to change the material parameters of the original structure, so that the time step Δt can be significantly enlarged. Although a different problem is simulated, the dc mode of the new problem is the same as that in the original problem.

It should also be noted here that letting $\mathbf{R} = 0$, i.e., removing the lossy part from the entire structure, cannot produce the correct dc modes of the original problem that has lossy conductors. This is because there are two kinds of dc modes [15] that satisfy (13). One represents the capacitance (C) effect. This mode is the nullspace eigenvector of the following generalized eigenvalue problem:

$$\mathbf{S}_{\text{oo}}V = \lambda\mathbf{T}_{\text{oo}}V \tag{14}$$

where $\mathbf{S}_{\text{oo}}$ and $\mathbf{T}_{\text{oo}}$, respectively, denote $\mathbf{S}$ and $\mathbf{T}$ formed by the unknowns outside conductors, with the conductor surface serving as the perfect conductor boundary condition of the dielectric region. Clearly, by scaling permittivity, and hence $\mathbf{T}$, the eigenvector of (14) stays the same. The other dc mode carries the resistance (R) effect. This mode is the nullspace eigenvector of the other generalized eigenvalue problem

$$\mathbf{S}_{\text{ii}}V = \lambda\mathbf{R}_{\text{ii}}V \tag{15}$$

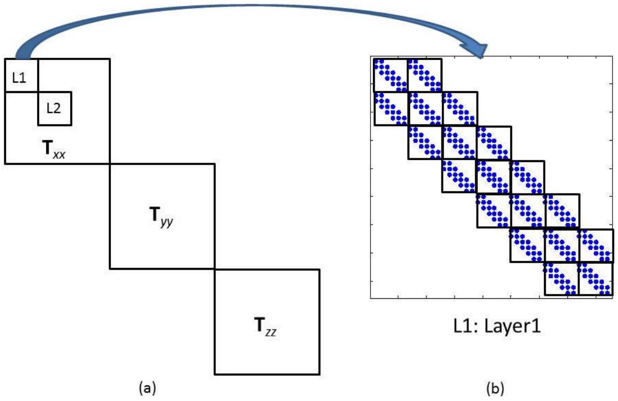


Fig. 1. Structure of $\mathbf{T}$ matrix. (a) Overall $\mathbf{T}$ matrix structure. (b) Structure of each diagonal block (layer) in $\mathbf{T}_{\xi\xi}$ ($\xi = x, y, z$), which is a block tridiagonal matrix with each tridiagonal block linearly proportional to each other.

where $\mathbf{S}_{\text{ii}}$ and $\mathbf{R}_{\text{ii}}$, respectively, denote $\mathbf{S}$ and $\mathbf{R}$ formed by the unknowns inside and on the surface of the standalone conductors, without being superposed by the contribution from the unknowns outside conductors. Again, by scaling the conductivity, $\mathbf{R}$ is scaled by a single number, but the eigenvectors in (15) do not change. By setting $\mathbf{R} = 0$, the conductor is changed to a dielectric material, and thus, the dc modes of such a dielectric problem are not governed by (14) and (15) any more. Therefore, even though a lossless treatment has a lot of advantages in computation, to obtain a complete set of dc modes for a general lossy problem, $\mathbf{R}$ part cannot be set to zero. However, for any conductor whose conduction current is larger than the displacement current by two orders of magnitude or above, (14) and (15) would hold true for accurately representing dc modes. Therefore, we have a wide range of conductivity to choose from.

Based on the above finding, we choose σ and ε in such a way so that $\|\mathbf{A}\| < 1$ with a central-difference-based time step, and hence the inverse of $\mathbf{K}$, can be obtained based on (9). Since (8) and (9) only require the computation of $\mathbf{T}^{-1}$, we turn the solution of $\mathbf{K}$ to the solution of $\mathbf{T}$. Since $\mathbf{T}$ is only related to permittivity, the layered property of permittivity can be fully exploited to further turn the solution of $\mathbf{T}$ to a simple scaling, which is detailed in Section II-B.

B. Fast Structure-Aware T's Solution Leveraging Manhattan-Type Geometry and Layered Permittivity

Since on-chip circuits are the Manhattan-type structures, a brick-element-based discretization is ideal to use without sacrificing the accuracy in geometrical modeling. The $\mathbf{T}$ matrix with a brick-element-based discretization can be naturally decomposed into $\mathbf{T}_{xx}$, $\mathbf{T}_{yy}$, and $\mathbf{T}_{zz}$ diagonal blocks due to the orthogonality of x-, y-, and z-directions, as illustrated in Fig. 1(a). With proper ordering of unknowns, each of $\mathbf{T}_{xx}$, $\mathbf{T}_{yy}$, and $\mathbf{T}_{zz}$ can further be structured to a block diagonal matrix consisting of L1, L2, and so on, shown in Fig. 1(a). This is because the matrix blocks in different layers are fully decoupled, where the *layer* here refers to the region where the x-, y-, and z-orientated unknowns reside. The x-unknown, y-unknown, and z-unknown layers are, respectively, normal to

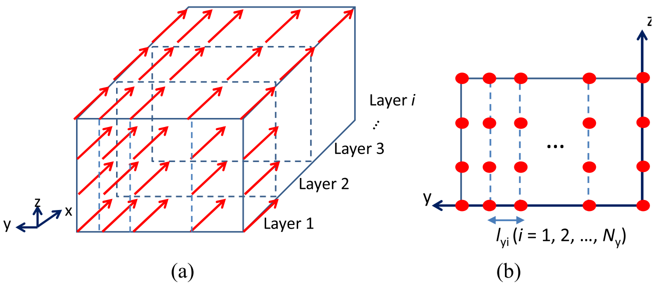


Fig. 2. Illustration of layers of x-unknowns (y-unknowns and z-unknowns can be visualized using the same figure by switching the coordinates). (a) 3-D view. (b) Cross-sectional view. Red dots: unknowns perpendicular to the cross section.

the x-, y-, and z-directions. To see this more clearly, Fig. 2(a) illustrates a 3-D view of the multiple layers of x-unknowns, with its cross-sectional view shown in Fig. 2(b), where each red dot represents one x-unknown. The layers of y-unknowns can be visualized by replacing the x-, y-, and z-coordinates in Fig. 2 by y, z, and x, respectively. Similarly, the layers of z-unknowns can be visualized by replacing x, y, and z in Fig. 2 by z, x, and y, respectively.

From Fig. 2(a), it can be clearly seen that the matrix formed for x-unknowns in each layer is completely decoupled from that formed in another layer, since the unknowns are internal to the layer. As a result, $\mathbf{T}_{xx}$ in Fig. 1(a) is a block diagonal matrix, with each block representing $\mathbf{T}_{xx}$ in one layer. The same is true for $\mathbf{T}_{yy}$ and $\mathbf{T}_{zz}$, as is evident by treating the red arrows/dots in Fig. 2 as y- and z-unknowns, respectively.

In each layer, if we order the unknowns line by line, i.e., order all the unknowns along one vertical (or horizontal) line, as shown in Fig. 2(b), and then move along the horizontal (or vertical) direction to the unknowns on the next line, we will obtain a block tridiagonal matrix, as shown in Fig. 1(b), which can be written as

$$\begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_0 & & & & & \\ \mathbf{B}_0 & \mathbf{A}_0+\mathbf{A}_1 & \mathbf{B}_1 & & & & \\ & \mathbf{B}_1 & \mathbf{A}_1+\mathbf{A}_2 & \mathbf{B}_2 & & & \\ & & \mathbf{B}_2 & \mathbf{A}_2+\mathbf{A}_3 & & & \\ & & & & \ddots & & \\ & & & & & \ddots & \\ & & & & & \mathbf{A}_{L-2}+\mathbf{A}_{L-1} & \mathbf{B}_{L-1} \\ & & & & & \mathbf{B}_{L-1} & \mathbf{A}_{L-1} \end{bmatrix} \tag{16}$$

where each value of $\mathbf{A}_i$ ($i = 0, 1, 2, \ldots, L-1$) represents a sparse matrix formed on a single line. Due to the layered permittivity and the Manhattan geometry leveraged by the brick-element-based discretization, $\mathbf{A}_i$ for x-, y-, and z-unknowns, respectively, can be written as

$$\begin{aligned} \mathbf{A}_{i,x} &= \frac{l_{y,i}}{l_{y,0}}\mathbf{A}_{0,x}, \quad i = 0, 1, 2, \ldots, N_y \\ \mathbf{A}_{i,y} &\sim \frac{l_{x,i}}{l_{x,0}}\mathbf{A}_{0,y}, \quad i = 0, 1, 2, \ldots, N_x \\ \mathbf{A}_{i,z} &\sim \frac{l_{x,i}}{l_{x,0}}\mathbf{A}_{0,z}, \quad i = 0, 1, 2, \ldots, N_x \end{aligned} \tag{17}$$

where N_x and N_y are, respectively, the number of unknowns along the x- and y-directions; $l_{\xi,i}$ ($\xi = x, y, z$) is the ith segment length along ξ-direction; $\mathbf{A}_{0,\xi}$ ($\xi = x, y, z$) is the mass matrix $\mathbf{T}$ formed by the unknowns along the first line in the cross section that is perpendicular to the x-, y-, and z-directions, respectively. For y-orientated unknowns, the line direction for ordering unknowns is chosen as z instead of x, because in this way, each region in between two lines has the same permittivity configuration. We can also choose x as the line direction. The only change is to incorporate a permittivity ratio into $\mathbf{A}_{i,y}$. Equation (17) gives $\mathbf{A}_i$ matrix components in each layer shown in (16). For different layers, $\mathbf{A}_{i,x}$ and $\mathbf{A}_{i,y}$ are the same, but $\mathbf{A}_{i,z}$ needs to be scaled by a ratio of permittivity, since the permittivity in each z-layer (one dielectric stack) is different. As for $\mathbf{B}_i$, it is nothing but

$$\mathbf{B}_i = 0.5\mathbf{A}_i \tag{18}$$

for unknowns along any direction. As a result, all the $\mathbf{A}_i$ and $\mathbf{B}_i$ blocks in (16) are linearly proportional to each other. Furthermore, each of $\mathbf{A}_i$ and $\mathbf{B}_i$ is tridiagonal. Hence, the solution of $\mathbf{T}$ becomes a simple scaling of the solution of a small tridiagonal matrix as follows.

For the matrix shown in (16), we perform an analytical line reduction (the union of the number of unknowns in each block $\mathbf{A}_i$ forms a line) to a single line based on the proportionality of the blocks, from which we recover the solution anywhere else. The reduced matrix $\tilde{\mathbf{A}}_i$ can be expressed by

$$\begin{aligned} \tilde{\mathbf{A}}_{i,\xi} &= \mathbf{A}_{i-1,\xi} + \mathbf{A}_{i,\xi} - \mathbf{B}_{i-1,\xi}\tilde{\mathbf{A}}_{i-1,\xi}^{-1}\mathbf{B}_{i-1,\xi} = s_\xi(i) \cdot \mathbf{A}_{0,\xi} \\ &\qquad\qquad\qquad\qquad\qquad\qquad i = 1, 2, \ldots, L-1 \\ \tilde{\mathbf{A}}_{i,\xi} &= \mathbf{A}_{i-1,\xi} - \mathbf{B}_{i-1,\xi}\tilde{\mathbf{A}}_{i-1,\xi}^{-1}\mathbf{B}_{i-1,\xi} = s_\xi(i) \cdot \mathbf{A}_{0,\xi}, \quad i = L \end{aligned} \tag{19}$$

where $L = N_y$, N_x, and N_x, respectively, are along the x-, y-, and z-directions. As can be seen, no inverse and matrix–matrix products are needed for the computation of $\tilde{\mathbf{A}}_{i,\xi}$, since the blocks involved are all linearly proportional to $\mathbf{A}_{0,\xi}$. We only need to calculate the scaling coefficients s_ξ based on the edge length ratio as follows:

$$\begin{aligned} s_\xi(i) &= \text{length\_ratio\_}\xi[i-1] + \text{length\_ratio\_}\xi[i] \\ &\quad - 0.25 \cdot (\text{length\_ratio\_}\xi[i-1])^2 \cdot [s_\xi(i-1)]^{-1} \ (i < L) \\ s_\xi(i) &= \text{length\_ratio\_}\xi[i-1] \\ &\quad - 0.25 \cdot (\text{length\_ratio\_}\xi[i-1])^2 \cdot [s_\xi(i-1)]^{-1} \ (i = L) \end{aligned} \tag{20}$$

where the length ratio length_ratio$_\xi$ ($\xi = x, y, z$), based on (17), is given by

$$\begin{aligned} \text{length\_ratio\_}x[i] &= \frac{l_{y,i}}{l_{y,0}} \\ \text{length\_ratio\_}y[i] = \text{length\_ratio\_}z[i] &= \frac{l_{x,i}}{l_{x,0}}. \end{aligned} \tag{21}$$

The right-hand side b is also analytically reduced to a single-line-based right-hand side as follows:

$$\begin{aligned} \tilde{b}_{i,\xi} &= b_{i,\xi} - \mathbf{B}_{i-1,\xi}\tilde{\mathbf{A}}_{i-1,\xi}^{-1}\tilde{b}_{i-1,\xi} \\ &= b_{i,\xi} - 0.5 \times \text{length\_ratio\_}\xi[i-1] \\ &\quad \times [s_\xi(i-1)]^{-1} \cdot \tilde{b}_{i-1,\xi}. \end{aligned} \tag{22}$$

Algorithm 1 Solving ξ ($\xi = x, y, z$)-Unknowns

```
1. for $i = 1, 2, ..., L$
  1.1. $b[i] = b[i] - 0.5 \cdot length\_ratio\_\xi[i-1]$
         $\cdot (s\_\xi[i-1])^{-1} \cdot b[i-1]$
2. inv_uv($u_\xi$, $v_\xi$, $b[L]$, $temp$)
3. $x[L] = (s\_\xi[L])^{-1} \cdot temp$
4. for $i = (L-1), \ldots 1, 0$
  4.1. inv_uv($u\_\xi, v\_\xi, b[i], temp$)
  4.2. $x[i] = (s\_\xi[i])^{-1} \cdot temp$
         $- 0.5 \cdot (s\_\xi[i])^{-1} \cdot (length\_ratio\_\xi[i]) \cdot x[i+1]$
```

in which $i = 1, 2, \ldots, L, \xi = x, y, z$. After the unknowns in the last line are solved from the reduced single-line system

$$x_i = \tilde{\mathbf{A}}_{i,\xi}^{-1}\tilde{b}_{i,\xi} = [s_\xi(i)]^{-1}\mathbf{A}_{0,\xi}^{-1}\tilde{b}_{i,\xi}, \quad i = L. \tag{23}$$

The unknowns on other lines are obtained recursively from the following:

$$\begin{aligned} x_i &= \tilde{\mathbf{A}}_{i,\xi}^{-1}(\tilde{b}_{i,\xi} - \mathbf{B}_{i,\xi}x_{i+1}) \\ &= [s_\xi(i)]^{-1}\mathbf{A}_{0,\xi}^{-1}\tilde{b}_{i,\xi} - 0.5[s_\xi(i)]^{-1}\text{length\_ratio\_}\xi[i]x_{i+1} \\ &\qquad (i = L-1, \ldots 2, 1, 0). \end{aligned} \tag{24}$$

The aforementioned algorithm for recovering the $\xi = x, y, z$ direction unknowns is shown in Algorithm 1. In this algorithm, Step 1 is to generate the right-hand side vector shown in (22), and Step 2 produces the result of $\mathbf{A}_{0,\xi}^{-1}\tilde{b}_{i,\xi}$ with i being the last line index. u_ξ and v_ξ are the precomputed UV factors of tridiagonal matrix $\mathbf{A}_{0,\xi}$, and the solution is stored in vector "temp." Step 3 generates the solution of the last line shown in (23). Finally, Step 4 is to compute the solutions on all the other lines in (24).

The overall computation is the solution of one tridiagonal matrix, and all the other steps are simple algebraic operations whose computational cost is negligible. The tridiagonal matrix is the $\mathbf{A}_{0,\xi}$ ($\xi = x, y, z$) matrix, whose size is 1-D. Its solution can be accomplished by UV factorization for tridiagonal matrices in linear complexity with negligible cost [16].

C. Fast DC-Mode Extraction From a New Problem With Modified Material Parameters

We perform time marching of (6) but with modified $\mathbf{T}$ and $\mathbf{R}$ matrices

$$\mathbf{T}_{\text{new}} = a_t\mathbf{T}, \quad \mathbf{R}_{\text{new}} = a_r\mathbf{R} \tag{25}$$

by scaling all the conductivity values down by a_r, and increasing all the permittivity values by a_t. By doing so, (6) can be performed fast with a large time step while (9) can still be converged in a few terms. Since (6) is based on a central-difference time discretization, the time step needs to satisfy the stability condition for a central-difference-based time marching. Thus, we enlarge time step through (25), but we do not exceed the time step allowed by the central-difference-based time marching for a stable simulation.

As for the detailed choice of material parameters, take the conductivity as an example, we choose conductivity σ in the range of $(\sigma_{\min}, \sigma_{\max})$, where the smallest value $\sigma_{\min}$ is determined in such a way that the resultant material is still a conductor. Hence, we choose $\sigma_{\min} = 100\ \omega\varepsilon$, so that the conduction current is larger than the displacement current by two orders of magnitude. Considering the prevailing on-chip circuit operating frequencies, $\sigma_{\min}$ can be chosen as 100. $\sigma_{\max}$ is determined from the maximum conductivity that makes the norm of $\mathbf{A} = (\Delta t/2)\mathbf{T}^{-1}\mathbf{R}$ less than 1, and hence ensuring the convergence of (9). Since the norm of $\mathbf{T}^{-1}\mathbf{R}$ is proportional to σ/ε, $\mathbf{A}$'s norm is approximately $0.5\Delta t(\sigma/\varepsilon)$. For this norm to be less than 1, the largest conductivity $\sigma_{\max}$ that can be chosen is obviously $2\varepsilon/\Delta t$. On-chip circuits have smallest feature sizes at micrometer level or even smaller, yielding a time step of 1e-16 s or smaller. Hence, $\sigma_{\max}$ is at the level of $1e + 5$ or larger. In the range of $(\sigma_{\min}, \sigma_{\max})$, every σ can be chosen without breaking down the proposed method. However, there is a better choice that makes the total CPU time smaller. This will be studied in Section III.

It is worth mentioning that the time step of the central-difference scheme in the proposed method is also larger than that in the original problem, because permittivity is increased. To explain, from the stability analysis of a central-difference TDFEM scheme [14], it is known that the time step needs to satisfy $\Delta t < 2/(\rho(\mathbf{T}^{-1}\mathbf{S}))^{1/2}$ for a stable time marching, where $\rho(\mathbf{T}^{-1}\mathbf{S})$ denotes the spectral radius of $\mathbf{T}^{-1}\mathbf{S}$. Thus, with (25), the new time step can be enlarged by a factor of $\sqrt{a_t}$. In addition, by modifying both conductivity and permittivity, the number of terms used in (9) can be further reduced. The reasons are: 1) Δt of $\mathbf{A}$ in (10) is proportional to $\sqrt{a_t}$ and 2) $\|\mathbf{T}^{-1}\mathbf{R}\|$ of $\mathbf{A}$ is proportional to $1/a_t$. Hence, overall, the norm of $\mathbf{A}$ in (10) is reduced by $1/\sqrt{a_t}$, which results in a smaller number of terms for the convergence of (9), thereby more speedup.

We only need to perform the simulation of (6) for a small time window to reveal the dc modes, like the preprocessing algorithm given in [17]. The detailed algorithm is as follows. We start the solution of (6). At every p step, we sample the solution of (6) and add it as one column vector in $\mathbf{X}$, which is initialized to be zero. The sample interval p is generally chosen as $\Delta t_{\text{accuracy}}/\Delta t_{\text{stability}}$, where $\Delta t_{\text{accuracy}}$ is the time step required by sampling accuracy for the input spectrum and $\Delta t_{\text{stability}}$ is the time step determined by stability condition. When adding a new solution vector into $\mathbf{X}$, we orthogonalize it with the column vectors that have already been stored in $\mathbf{X}$.

With $\mathbf{X}$, whose column dimension is denoted by k, we transform the original large quadratic eigenvalue problem of size N in (12) to a small eigenvalue problem of size k

$$\mathbf{B}_r\Phi_r = \lambda\mathbf{A}_r\Phi_r \tag{26}$$

where

$$\mathbf{A}_r = \begin{bmatrix} \mathbf{R}_r & \mathbf{T}_r \\ \mathbf{T}_r & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_r = \begin{bmatrix} -\mathbf{S}_r & 0 \\ 0 & \mathbf{T}_r \end{bmatrix} \tag{27}$$

in which

$$\mathbf{T}_r = \mathbf{X^H}\mathbf{T}\mathbf{X}, \ \mathbf{R}_r = \mathbf{X^H}\mathbf{R}\mathbf{X}, \quad \mathbf{S}_r = \mathbf{X^H}\mathbf{S}\mathbf{X}. \tag{28}$$

At early time, we observe the eigenvalues of large magnitude from (26). The dc modes appear later, which can be identified by their small values compared with other eigenvalues.

Once dc modes are identified from (26), we can terminate the time marching of (6). Let the dc modes extracted from (26) be $\tilde{\Phi}_{\rm dc}$ The dc modes of the original problem (12) can be obtained as

$$\mathbf{V}_{\rm dc} = \mathbf{X}_{N\times k}\tilde{\Phi}_{{\rm dc},k\times k_{\rm dc}} \tag{29}$$

where the subscripts denote the matrix dimensions and $k_{\rm dc}$ is the number of dc modes.

In (25), if we increase permittivity too much, the electrical size of the structure will be greatly enlarged, which will enlarge the time window to be simulated to identify dc modes. This is because for an electrically larger problem, more modes are involved in the field solution. In particular, the first higher order mode would appear at a lower frequency. The frequency at which to observe the dc mode, thus, becomes lower. As a result, in time domain, one has to wait for a longer time before the dc mode becomes nonnegligible in the field solution. Therefore, a_t cannot be chosen too large. In general, we choose it to be no greater than 10. Similarly, if we reduce the conductivity too much, the metal would be changed to a dielectric. The physical behavior of the dc modes, which is dominated by RC-effects, cannot be captured. In view of this, we reduce the conductivity in such a way that the resultant material is still a metal. In general, the conductivity is chosen to be no less than 1000 S/m.

D. Synthesizing Solution of the Original Problem

Since the original problem and the new modified problem share the same dc modes in common, the field solution of the original problem (1) can be accurately expanded into the dc modes extracted from the modified problem as follows:

$$u(t) = \mathbf{V}y(t). \tag{30}$$

with $\mathbf{V} = \mathbf{V}_{\rm dc}$, as shown in (29). Then, we then substitute (30) into (1) and multiply the resultant by $\mathbf{V}^H$ on both the sides, obtaining

$$\mathbf{T}_r\ddot{y}(t) + \mathbf{R}_r\dot{y}(t) + \mathbf{S}_r y(t) = \tilde{I}(t) \tag{31}$$

where $\mathbf{T}_r = \mathbf{V}^H\mathbf{T}\mathbf{V}$, $\mathbf{R}_r = \mathbf{V}^H\mathbf{R}\mathbf{V}$, $\mathbf{S}_r = \mathbf{V}^H\mathbf{S}\mathbf{V}$, and $\tilde{I}(t) = \mathbf{V}^H\dot{I}(t)$. The dimension of (31) is of O (1), which is much smaller than the original size of (1). In addition, the time step used for simulating (31) can be solely determined by accuracy, thereby much larger than that of the conventional explicit TDFEM. This is because the modes that cannot be stably simulated by such a large time step are not included in $\mathbf{V}$, as analyzed in [17]. As a result of small size and large time step, (31) can be simulated with negligible time.

It is also worth mentioning that the input pulse used for the dc-mode extraction can be different from the real pulse used in the final simulation. For example, a higher frequency input pulse can be employed in the step of dc-mode extraction, so that the CPU time can be further reduced.

III. Numerical Results

A. Two On-Chip Interconnect Structures

First, we simulate an on-chip interconnect structure to validate the proposed algorithms. The structure is illustrated in Fig. 3. The length, width, and height of the structure are 120, 30, and 3.192 μm, respectively. The top and bottom planes are truncated by a perfect electric conductor (PEC) boundary condition, while the front and back planes are terminated by absorbing boundary condition, and the other two boundaries are left open. The permittivity and conductivity distribution of the structure is shown in Fig. 3(b). The input current sources have a Gaussian derivative pulse of $I(t) = 2(t - t_0\exp(-(t - t_0)^2/\tau^2)$, with $\tau = 3 \times 10^{-8}$ s and $t_0 = 4\tau$. They are launched from the bottom metal layer to the inner conductor, and from the upper metal layer to the inner conductor, as shown in Fig. 3(a) (red arrows).

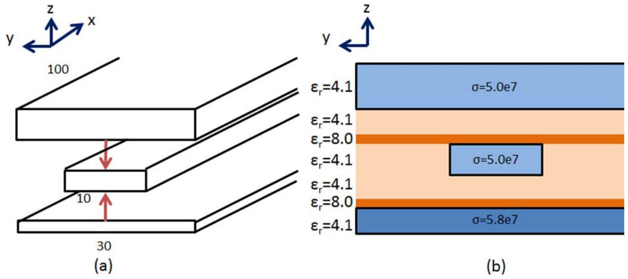


Fig. 3. Illustration of an on-chip interconnect. (a) 3-D view of the structure. (b) yz plane view of the structure.

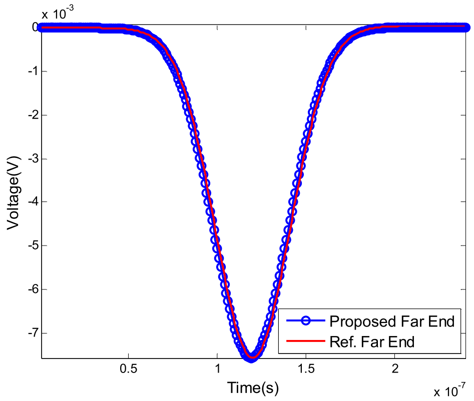


Fig. 4. Simulation of an on-chip interconnect.

The modified problem for fast dc-mode extraction has a conductivity reduced by 1×10^{-4}, and permittivity increased by 10, which results in a time step of 3.3×10^{-15} s used in time marching for dc-mode extraction. Due to the increase in permittivity by 10, the resultant time step is $\sqrt{10}$ larger than that required by the original problem for stability. In the stage of dc-mode extraction, ten solutions are sampled with a sampling interval of $p = 303\,030$, before the dc mode is accurately identified. The number of terms used in the series expansion (9) is 9. After the dc-mode extraction from the modified problem, the transient solution of the original problem is synthesized. The voltage obtained at the far end of the structure is plotted in Fig. 4. Excellent agreement is observed between the proposed method and the conventional TDFEM scheme [14]. With the same computer (Intel XEON E5410 2.33-GHz processor), the speedup of the proposed method over the conventional TDFEM is shown to be 4.824,

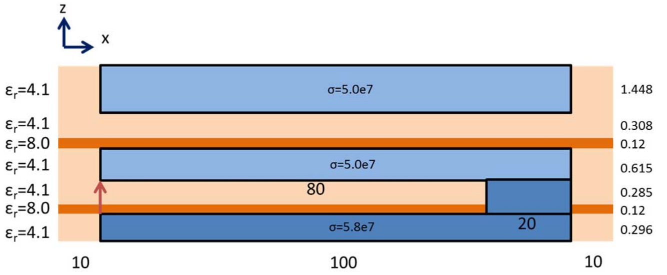
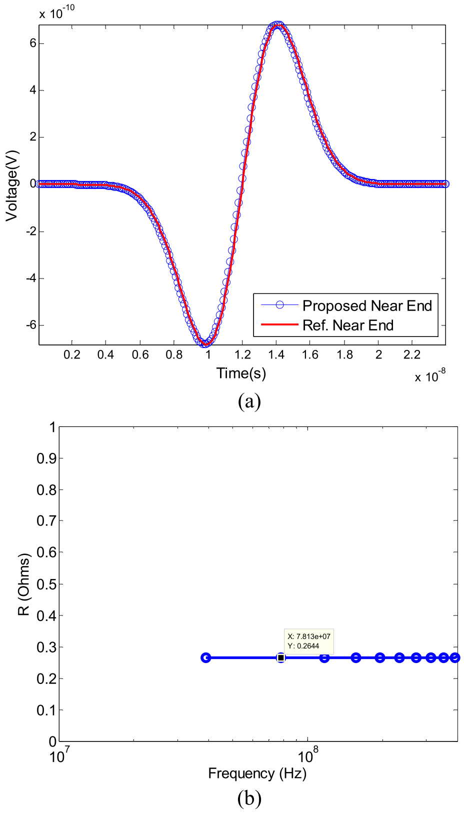


Fig. 5. Side view of a shorted on-chip interconnect.

which includes the CPU time at every step from the time-domain simulation of a modified problem for fast dc-mode extraction to the synthesis of the solution of the original problem. In contrast, with the method in [9], although the challenge of matrix solution is also overcome, the total CPU time is still longer than that of a conventional TDFEM due to a reduced time step.

The structure simulated in the above is dominated by capacitance effects at relatively low frequencies. To examine the accuracy of the proposed method in handling R-dominant circuits, we simulate the same structure, but let the far end shorted to the bottom plane by a metal contact, as shown in Fig. 5. Different from the setting in the previous structure, the input current source is launched from the bottom metal layer to the inner conductor only, and it has a Gaussian derivative pulse with $\tau = 3 \times 10^{-9}$ s. The modified problem has a conductivity reduced by 1×10^{-4}. The time step used in time marching for dc-mode extraction is 1×10^{-15} s, which is the same as that permitted by a traditional TDFEM-based simulation of the original problem. The sampling interval p is 100 000, and seven terms are used in (9). The CPU time of the conventional TDFEM is 3.729×10^4 s, while the proposed method is 1.835 times faster. Same as that in the previous example, we can also modify permittivity to enlarge time step as well as reduce the number of terms required for the convergence of (9). The modified problem has a conductivity reduced by 1×10^{-4} and permittivity increased by 10. Again, due to the increase in permittivity by 10, the resultant time step is enlarged to 3.3×10^{-15} s, and hence $p = 30\,303$. In total, ten solutions are sampled before the dc mode is accurately identified. The number of terms is five for the convergence of series expansion, which is smaller than that in the previous setting where permittivity is not changed. The speedup of the proposed method over the conventional TDFEM is hence increased to 8.095. In Fig. 6, we plot the near-end voltage of the proposed method in comparison with the reference TDFEM solution. Excellent agreement is observed.

In this example, we have also extracted the resistance R from the time-domain results obtained from the proposed method. Since the current source is launched from the ground plane to the strip line at the center metal layer, as shown in Fig. 5 (arrow), the voltage obtained at the current source location divided by the input current source will yield the input impedance of the structure. First, we correlate the voltage shown in Fig. 6(a) with the current source in the time domain, and find the that the voltage has the same waveform as the current source, but scaled by a factor of 0.2644 Ω. This tells us clearly that the input impedance is dominated by an R-based effect, and R is 0.2644 Ω. Then, we do a more detailed frequency-domain analysis. We perform a Fourier transform on both the voltage and the current source, and then divide the voltage by the current in frequency domain. Fig. 6(b) plots the resistance we obtain in the input spectrum. Clearly, the resistance is shown to be 0.2644 Ω. This resistance agrees well with the analytical data of 0.2602 Ω, which is the resistance of an 80-μm-long metal of the cross-sectional area of 10 μm × 0.615 μm with conductivity 5e + 7 S/m. In fact, the resistance extracted from our numerical simulation should be more accurate than the analytical estimation, since it considers the corner and fringing effects.



(a)

(b)

Fig. 6. Accuracy validation of the proposed algorithm in simulating a far-end shorted on-chip interconnect. (a) Voltage. (b) Extracted resistance.

B. On-Chip Power Grid

An on-chip power grid structure, as shown in Fig. 7, is simulated. The red lines denote power rails, while blue ones are ground rails. There is a vertical via connecting two metal wires, wherever the two wires of same polarity cross each other in the top view. The size of the structure is 7 μm × 7 μm × 7.6 μm. The top and bottom planes are set to be PEC, and the other four sides are left open. The number of unknowns in this example is 1101. The permittivity is layered, as shown in Fig. 7(c), and the conductivity of the metal is 5×10^7 S/m.

Since the proposed method allows for the use of a different pulse in the dc-mode extraction stage compared with the real one required in the simulation stage, we employ a higher frequency input than the original input to achieve an even

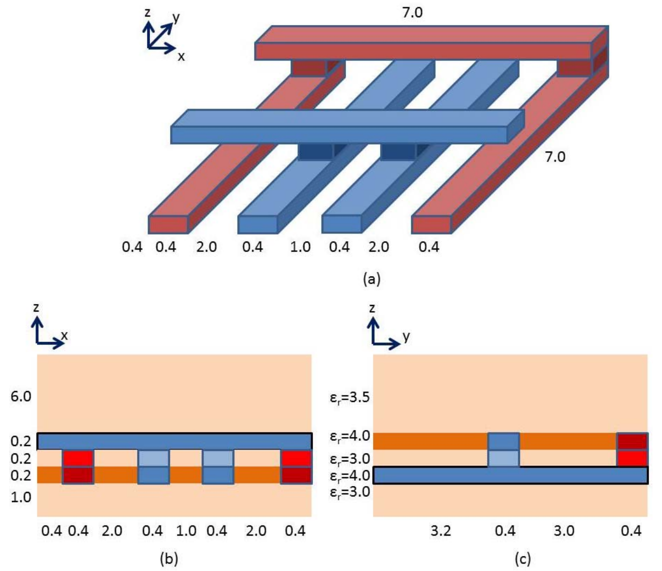


Fig. 7. Illustration of the structure of an on-chip power grid. (a) 3-D view. (b) xz plane view. (c) yz plane view.

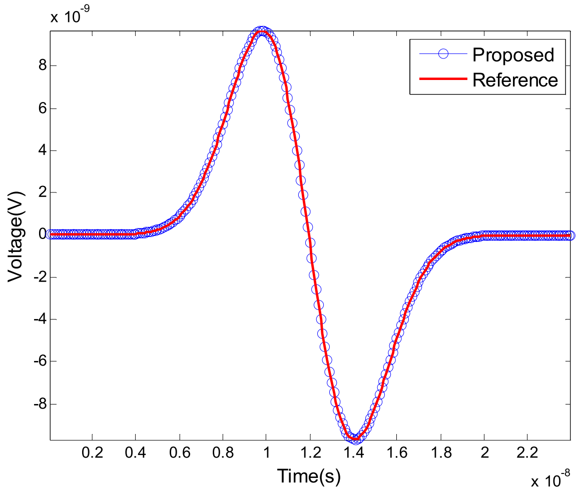


Fig. 8. Accuracy validation of the proposed algorithm in power grid simulation.

better speedup. The original Gaussian derivative pulse is $I(t) = 2(t - t_0)\exp(-(t - t_0)^2/\tau^2)$, where $\tau = 3 \times 10^{-9}$ s and $t_0 = 4\tau$. The pulse we use in the stage of dc-mode extraction has $\tau = 3 \times 10^{-11}$ s, and hence, its maximum frequency is 100 times larger than before. The time step used in time marching for the dc-mode extraction is 5×10^{-16} s. The solutions are sampled every 2000 steps, i.e., $p = 2000$. In total, 15 solutions are sampled before the dc mode is accurately identified. The modified problem has a conductivity reduced by 1×10^{-4}. The permittivity is kept the same as before. The number of terms is five for the convergence of series expansion shown in (9). The current source is launched between one power rail and one ground rail in the lower metal layer, and the voltage between the two rails is sampled and plotted in Fig. 8. Again, excellent agreement with the reference TDFEM solution is observed. With the same computer, the CPU time of the conventional central-difference TDFEM is 8.950×10^4 s, whereas the CPU time of the proposed method, including all steps, is 3.555×10^2 s, thus a speedup of 251.7.

TABLE I

PERFORMANCE AS A FUNCTION OF CONDUCTIVITY

Conductivity scaling factor	DC-extraction sample number	Number of expansion terms	Total CPU time (s)
1e-5	120	3	2026
0.2e-4	80	3	1270
0.5e-4	25	4	478
1e-4	15	5	356
2e-4	10	5	260
5e-4	8	7	262
1e-3	8	12	418

In this example, we have also varied the scaling factor a_r from 1e-5 to 1e-3 to examine the choice of conductivity on the performance of the proposed solver. We find that all of the conductivity choices produce accurate results. However, different choices can result in different CPU times. In Table I, we list the total CPU time and the resultant simulation parameters as a function of a_r, where the second column refers to the number of samples used in the dc-mode extraction step, while the third column shows the number of expansion terms used in (9). It is evident that if a larger a_r, and hence a larger conductivity is chosen, the norm of $\mathbf{A}$ shown in (10) is larger. As a result, the number of terms used to obtain the inverse of the system matrix $\mathbf{K}$ is larger, degrading the computational efficiency. However, the time interval required for extracting the dc modes is shorter, as evidenced by the smaller number of solution samples. If a smaller a_r is chosen, since the material is less conductive, the time we have to wait to extract accurate dc modes is longer. However, the number of expansion terms used for matrix solution is smaller due to a smaller norm of A. Therefore, neither the largest ($\sigma_{\max}$) nor the smallest conductivity ($\sigma_{\min}$) results in the smallest CPU run time. Instead, a choice in between is a better choice in terms of minimizing CPU run time.

C. Rectangular Spiral Inductor

Then, we simulate an on-chip spiral inductor to validate the proposed algorithms. The structure along with its permittivity configuration is illustrated in Fig. 9. The entire computational domain occupies a region of 1200 μm × 1000 μm × 500 μm. The top and bottom planes are truncated by a PEC boundary condition, while all the other boundaries are left open. The number of unknowns in this example is 14 286. The permittivity is layered and the conductivity of the conducting wire is 5×10^7 S/m. The input current source has a Gaussian derivative pulse of $I(t) = 2(t - t_0)\exp(-(t - t_0)^2/\tau^2)$, with $\tau = 3 \times 10^{-8}$ s and $t_0 = 4\tau$. It is launched from the bottom PEC plane to the left port of the inductor, as shown in Fig. 9 (red arrow). The time step used in time marching for dc-mode extraction is 5×10^{-14} s. The solutions are sampled every 20 000 steps, i.e., $p = 20\,000$, since the time step

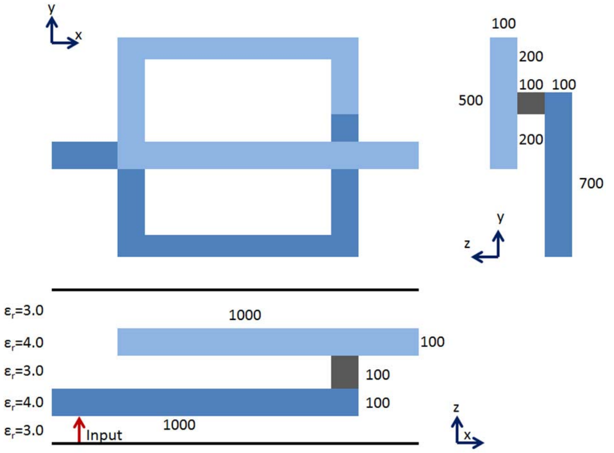


Fig. 9. Illustration of a rectangular spiral inductor structure.

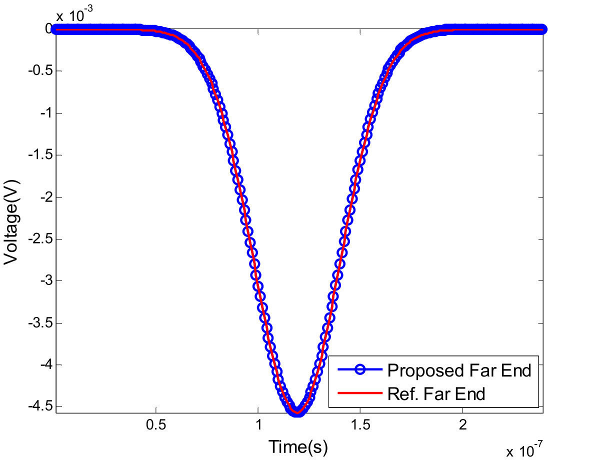


Fig. 10. Accuracy validation of the proposed algorithm for the simulation of a rectangular spiral inductor with $dt = 1.5 \times 10^{-13}$.

required by accuracy is 1×10^{-9} s. In total, six solutions are sampled before the dc mode is accurately identified. The modified problem has a conductivity reduced by 1×10^{-5}. The number of terms is seven for the convergence of series expansion shown in (9). The CPU time of the conventional TDFEM is 1.734×10^5 s, whereas the proposed method is 8.874 times faster.

Furthermore, we increase the permittivity by 10. As a result, the time step is enlarged by a factor of $\sqrt{10}$, yielding a time step of 1.5×10^{-13} s used in time marching for dc-mode extraction. The solutions are sampled every 6666 steps. In total, six solutions are sampled before the dc mode is accurately identified. The speedup of the proposed method is 20.736. The voltage simulated at the right port of the inductor is plotted in Fig. 10. Excellent agreement is observed between the proposed method and the conventional central-difference-based TDFEM scheme.

D. Suite of Large-Scale On-Chip Power Grids

In the last example, we simulate a suite of large-scale on-chip power grids, as shown in Fig. 11. The first one has a unit block size of 7.2 μm × 7.2 μm × 7.6 μm, which is then extended ten times along both x- and y-directions. The top and bottom planes are set to be PEC, and the other four sides are left open. The number of unknowns in this example is 118 715. The permittivity is layered as shown in Fig. 11(c), and the conductivity of the metal is 5×10^7 S/m.

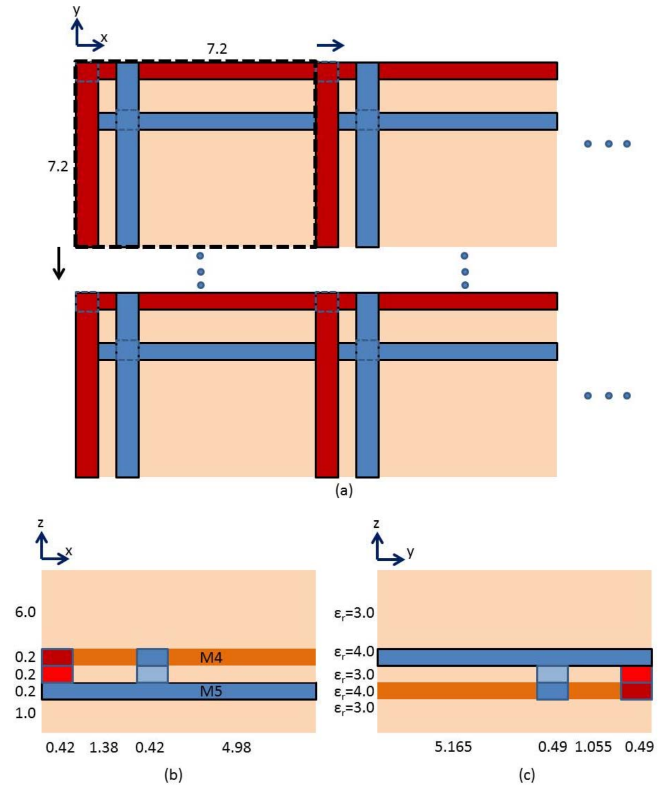


Fig. 11. Illustration of a larger on-chip power grid structure. (a) xy plane view. (b) xz plane view. (c) yz plane view.

In the dc-mode extraction step, we employ a higher frequency input than the original input. The original Gaussian derivative pulse is $I(t) = 2(t - t_0)\exp(-(t - t_0)^2/\tau^2)$, where $\tau = 3 \times 10^{-9}$ s and $t_0 = 4\tau$. The pulse we use in the step of dc-mode extraction has $\tau = 3 \times 10^{-11}$ s, and hence a maximum frequency 100 times larger than before. The time step used in time marching for dc-mode extraction is 5×10^{-16} s. The solutions are sampled every 2000 steps, i.e., $p = 2000$. In total, 11 solutions are sampled before the dc mode is accurately identified. The modified problem has a conductivity reduced by 1×10^{-4}. The permittivity is kept the same as before. The number of terms is five for the convergence of series expansion shown in (9). The current source is launched between one power rail and one ground rail in the upper metal layer, and the voltage between the two rails is sampled and plotted in Fig. 12 in comparison with the reference TDFEM solution. With the same computer, the CPU time of the conventional central-difference TDFEM is 4.896×10^6 s, whereas the CPU time of the proposed method, including all steps, is 3.248×10^4 s, thus a speedup of 150.7.

Then, we increase the structure simulated in the above progressively along both x- and y-directions. The chip area is increased from 72 × 72 μm^2, 144 × 44 μm^2, 288 × 288 μm^2, 360 × 360 μm^2, 576 × 360 μm^2, 565 × 576 μm^2, 720 × 720 μm^2, and 1440 × 720 μm^2 to 1440 × 1440 μm^2,

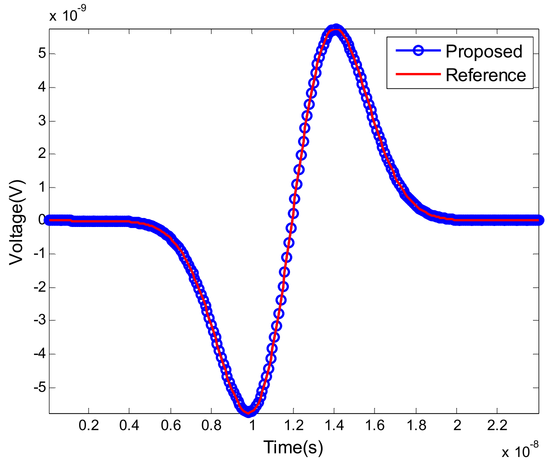
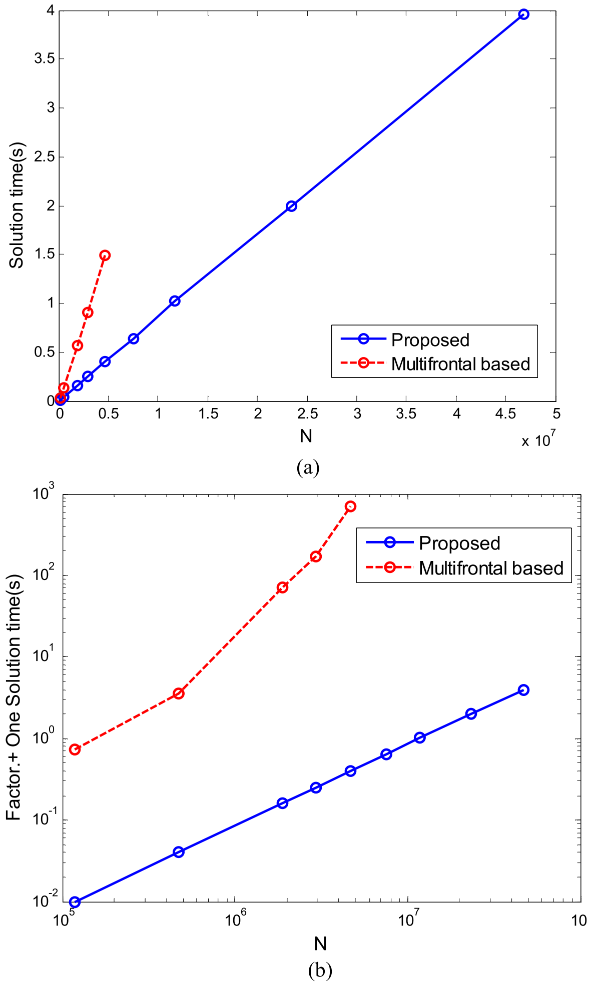


Fig. 12. Accuracy validation of the proposed algorithm in simulating a larger on-chip power grid.



(a)

(b)

Fig. 13. CPU time versus N for simulating a suite of on-chip power grids. (a) One solution time. (b) Factorization and one solution time.

resulting in 118 715, 471 425, 1 878 845, 2 933 555, 4 691 255, 7 501 685, 11 717 105, 23 426 105, and 46 834 205 unknowns, respectively. Using this suite of power grid structures, we compare the performance of the matrix solution in the proposed method with that of the conventional TDFEM, which employs a multifrontal-based direct solver [18]. For large cases, conventional *UV* factorization of tridiagonal matrices is not satisfactory, because U and V coefficients grow exponentially with the number of unknowns. Thus, the *UV* factorization is only used for cases with a small number of unknowns, and the ratio-based *DS* factorization [19] is employed to solve the tridiagonal matrix with linear complexity in negligible time for large unknown cases.

The solution time for one right-hand side is shown in Fig. 13(a), while the sum of the factorization and one solution time is shown in Fig. 13(b). As shown in Fig. 13, the solution time of the proposed solver is much less than that of the conventional solver. Moreover, the solution time has a clear linear scaling with the number of unknowns N. In contrast, the conventional solver has a complexity much higher than linear, and the number of unknowns the conventional solver can handle on the same computer is much fewer than that solved by the proposed method. For example, the proposed solver has no difficulty in simulating the last case in the suite of power grid structures, which has over 46 million unknowns, on a single core, whereas the conventional solver cannot go beyond a few million unknowns on the same computer due to their high memory complexity in addition to high time complexity. The lines for the multifrontal-based UMFPACK solver are only plotted for five data points, because it runs out of memory and fails to finish the matrix solutions for the remaining four test structures.

IV. Conclusion

In this paper, a fast structure-aware direct time-domain finite-element solver is developed for the analysis and the design of very large-scale on-chip circuits. The structure specialty of on-chip circuits, such as Manhattan geometry and layered permittivity, is preserved in the proposed numerical solution, and the resulting disadvantage in time step present in [1], [2], and [9] is overcome. As a result, the computational challenge of solving a very large-scale matrix encountered in the large-scale circuit analysis is removed, since the matrix solution at each time step is converted into a simple scaling regardless of the matrix size, and the total number of time steps to be simulated is also significantly reduced. The proposed method can be used for not only fast transient analysis but also IR-drop analysis and frequency-domain analysis of the on-chip circuits in a fairly wide range of frequencies, where full-wave effects are not yet dominant.

References

[1] H. Gan and D. Jiao, "A time-domain layered finite element reduction recovery (LAFE-RR) method for high-frequency VLSI design," *IEEE Trans. Antennas Propag.*, vol. 55, no. 12, pp. 3620–3629, Dec. 2007.

[2] H. Gan and D. Jiao, "A fast-marching time-domain layered finite-element reduction-recovery method for high-frequency VLSI design," *IEEE Trans. Antennas Propag.*, vol. 57, no. 2, pp. 577–581, Feb. 2009.

[3] M. Ha, K. Srinivasan, and M. Swaminathan, "Transient chip-package cosimulation of multiscale structures using the Laguerre-FDTD scheme," *IEEE Trans. Adv. Packag.*, vol. 32, no. 4, pp. 816–830, Nov. 2009.

[4] K. Sun, Q. Zhou, K. Mohanram, and D. C. Sorensen, "Parallel domain decomposition for simulation of large-scale power grids," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2007, pp. 54–59.

[5] G. Antonini and A. E. Ruehli, "Waveform relaxation time domain solver for subsystem arrays," *IEEE Trans. Adv. Packag.*, vol. 33, no. 4, pp. 760–768, Nov. 2010.

[6] R. Wang and J.-M. Jin, "A symmetric electromagnetic-circuit simulator based on the extended time-domain finite element method," *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 12, pp. 2875–2884, Dec. 2008.

[7] A. Rong and A. C. Cangellaris, "Generalized PEEC models for three-dimensional interconnect structures and integrated passives of arbitrary shapes," in *Proc. IEEE 10th Topical Meeting Elect. Perform. Electron. Packag.*, Oct. 2001, pp. 225–228.

[8] H. Gan and D. Jiao, "Hierarchical finite-element reduction-recovery method for large-scale transient analysis of high-speed integrated circuits," *IEEE Trans. Adv. Packag.*, vol. 33, no. 1, pp. 276–284, Feb. 2010.

[9] W. Lee and D. Jiao, "Structure-aware time-domain finite-element method for efficient simulation of VLSI circuits," in *Proc. IEEE Int. Symp. Antennas Propag.*, Jul. 2014, pp. 2254–2255.

[10] Q. He, D. Chen, J. Zhu, and D. Jiao, "Minimal-order circuit model based fast electromagnetic simulation," in *Proc. IEEE 22nd Conf. Elect. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2013, pp. 219–222.

[11] L. E. Tobón, Q. Ren, and Q. H. Liu, "Spectral-prism element for multi-scale layered package-chip co-simulations using the discontinuous Galerkin time-domain method," *Electromagnetics*, vol. 34, nos. 3–4, pp. 270–285, 2014.

[12] W. Yu and X. Wang, *Advanced Field-Solver Techniques for RC Extraction of Integrated Circuits*. Berlin, Germany: Springer-Verlag, 2014.

[13] D. Chen and D. Jiao, "Time-domain orthogonal finite-element reduction-recovery method for electromagnetics-based analysis of large-scale integrated circuit and package problems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1138–1149, Aug. 2009.

[14] J. Jin, "Finite element analysis in the time domain," in *The Finite Element Method in Electromagnetics*. New York, NY, USA: Wiley, 2002, pp. 529–584.

[15] J. Zhu and D. Jiao, "A rigorous solution to the low-frequency breakdown in full-wave finite-element-based analysis of general problems involving inhomogeneous lossless/lossy dielectrics and nonideal conductors," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 12, pp. 3294–3306, Dec. 2011.

[16] 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, 1992.

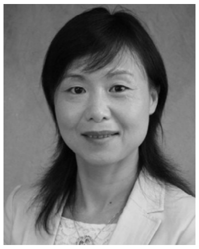
[17] Q. He, H. Gan, and D. Jiao, "Explicit time-domain finite-element method stabilized for an arbitrarily large time step," *IEEE Trans. Antennas Propag.*, vol. 60, no. 11, pp. 5240–5250, Nov. 2012.

[18] [Online]. Available: http://www.cise.ufl.edu/research/sparse/umfpack/UMFPACK-5.7.0.tar.gz

[19] J. Jain, H. Li, S. Cauley, C.-K. Koh, and V. Balakrishnan, "Numerically stable algorithms for inversion of block tridiagonal and banded matrices," Dept. Elect. Comput. Eng., Purdue Univ., West Lafayette, IN, USA, Tech. Rep. 357, 2007.

Woochan Lee received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2002 and 2005, respectively. He is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA.

He was commissioned as a full-time Lecturer and first Lieutenant with the Korea Military Academy, Seoul, from 2005 to 2008. He has been the Deputy Director and a Patent Examiner with Korean Intellectual Property Office, Daejeon, Korea, since 2004. His current research interests include computational electromagnetics for very large-scale electromagnetic analysis.



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

She was with the Technology Computer-Aided Design (CAD) Division, Intel Corporation, Santa Clara, CA, USA, until 2005, as a Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. In 2005, she joined the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, as an Assistant Professor, where she is currently a Professor. She has authored three book chapters and over 220 papers in refereed journals and international conferences. Her current research interests include computational electromagnetics, high-frequency digital, analog, mixed-signal, and RF integrated circuit (IC) design and analysis, high-performance very large-scale integration CAD, modeling of microscale and nanoscale circuits, applied electromagnetics, fast and high-capacity numerical methods, fast time-domain analysis, scattering and antenna analysis, RF, microwave, and millimeter-wave circuits, wireless communication, and bioelectromagnetics.

Dr. Jiao received the 2013 S. A. Schelkunoff Prize Paper Award of the IEEE Antennas and Propagation Society, which recognizes the best paper published in the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION during the previous year. She was a recipient of the 2010 Ruth and Joel Spira Outstanding Teaching Award, the 2008 National Science Foundation CAREER Award, the 2006 Jack and Cathie Kozik Faculty Start Up Award (which recognizes an outstanding new faculty member of the School of Electrical and Computer Engineering, Purdue University), the 2006 Office of Naval Research Award under the Young Investigator Program, the 2004 Best Paper Award at the Intel Corporation's Annual Corporate-Wide Technology Conference (Design and Test Technology Conference) for her work on generic broadband model of high-speed circuits, the 2003 Intel Corporation's Logic Technology Development (LTD) Divisional Achievement Award in recognition of her work on the industry-leading BroadSpice modeling/simulation capability for designing high-speed microprocessors, packages, and circuit boards, the Intel Corporation's Technology CAD Divisional Achievement Award for the development of innovative full-wave solvers for high frequency IC design, the 2002 Intel Corporation's Components Research Intel Hero Award (Intel-wide, she was the tenth recipient) for the timely and accurate 2-D and 3-D full-wave simulations, the Intel Corporation's LTD Team Quality Award for her outstanding contribution to the development of the measurement capability and simulation tools for high frequency on-chip crosstalk, and the 2000 Raj Mittra Outstanding Research Award from the University of Illinois at Urbana–Champaign. She has served as a Reviewer for many IEEE journals and conferences. She is an Associate Editor of the IEEE TRANSACTIONS ON COMPONENTS, PACKAGING, AND MANUFACTURING TECHNOLOGY. She was among the 85 engineers selected throughout the nation for the National Academy of Engineering's 2011 U.S. Frontiers of Engineering Symposium.