

RWCap: A Floating Random Walk Solver for 3-D Capacitance Extraction of Very-Large-Scale Integration Interconnects

Wenjian Yu, *Senior Member, IEEE*, Hao Zhuang, Chao Zhang, Gang Hu, and Zhi Liu

Abstract—A floating random walk (FRW) solver, called RWCap, is presented for the capacitance extraction of very-large-scale integration (VLSI) interconnects. An approach, including the numerical characterization of the cross-interface transition probability and weight value, is proposed to accelerate the extraction of structures with multiple dielectric layers. A comprehensive variance reduction scheme based on the importance sampling and stratified sampling is proposed to improve the convergence rate of the FRW algorithm. Finally, the space management technique using an octree data structure and the parallel computing technique are presented to further improve the efficiency. Numerical experiments are carried out with the test cases generated under the 180 and 45-nm process technologies. They demonstrate that the proposed multidielectric FRW algorithm achieves up to 160× speedup over the FRW algorithm using spherical transition domains to cross dielectric interface, with very small memory overhead. The variance reduction techniques further bring 3× or more speedup without memory overhead and the loss of accuracy. The RWCap also outperforms other existing FRW algorithm and fast boundary element method solvers in terms of computational time or scalability. The experiments on an 8-core CPU machine show that the parallel RWCap is over 6× faster than its serial-computing version.

Index Terms—Capacitance extraction, floating random walk (FRW), interconnect modeling, multidielectric structure, parallel computing, variance reduction.

I. INTRODUCTION

UNDER THE nanometer process technology, accurate extraction of interconnect capacitance with 3-D field-

Manuscript received May 20, 2012; revised August 18, 2012; accepted September 29, 2012. Date of current version February 14, 2013. This work was supported in part by the National Natural Science Foundation of China Research Project under Grant 61076034, the Tsinghua National Laboratory for Information Science and Technology Basic Research Project, the MOST of China Project 2011DFA60290, and the Tsinghua University Initiative Scientific Research Program. This paper was recommended by Associate Editor H. Qian.

W. Yu and C. Zhang are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: yu-wj@tsinghua.edu.cn; eric.3zc@gmail.com).

H. Zhuang was with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China, and the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. He is now with the Department of Computer Science and Engineering, University of California, San Diego, CA 92182 USA (e-mail: zhuanghoward@gmail.com).

G. Hu is with Columbia University, New York, NY 10027 USA (e-mail: henry.hu.sh@gmail.com).

Z. Liu is with Murex SAS, New York, NY 10019 USA (e-mail: liuzhi0210@gmail.com).

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/TCAD.2012.2224346

solver algorithm becomes increasingly important for high-performance integrated circuit (IC) design. The field-solver algorithm for capacitance extraction can be classified into two categories: 1) the conventional deterministic algorithms based on boundary element method (BEM) [1]–[6], finite element method (FEM) [7], [8], etc., and 2) the floating random walk (FRW) algorithm with stochastic nature [9]–[20]. The deterministic algorithms are fast and accurate, but not suitable for large-scale structures due to the large demand of computational time and the bottleneck of memory usage.

The FRW algorithm for capacitance extraction, presented as a 2-D version, was proposed in 1992 [9]. Its basic idea is to convert the calculation of conductor charge to the Monte Carlo (MC) integration performed with floating random walks. In the FRW algorithm, each walk starts from a point on a Gaussian surface enclosing the master conductor and terminates on a conductor surface after some successive hops. For each hop of a walk, a conductor-free square (or cube in 3-D problem) centered at current location is constructed and the hop reaches a random point on the boundary of square. The spatial transition of hop obeys a probability distribution (called surface Green's function) on the square boundary. In a general 3-D problem, if the cubic transition domain encloses a homogeneous dielectric, the surface Green's function for the hop can be calculated analytically. The transition probability is usually calculated and tabulated offline, and then recalled during the actual walk. Therefore, the FRW procedure can be performed very quickly. The FRW algorithm does not rely on assembling any linear equation system, and has several computational advantages over the deterministic methods [1]–[8]: lower memory usage, more scalability for large structures, tunable accuracy, and better parallelism. The 3-D FRW algorithm for capacitance extraction has been developed and applied to the design and analysis of very-large-scale integration (VLSI) circuits [10]–[12]. An academic FRW-based capacitance extractor CAPEM was also developed [13]. In 2005, Batterywala *et al.* proposed several techniques to reduce the variance of MC procedure in the FRW-based capacitance extraction [14], and further reduce the total computing time. The FRW algorithm was also extended to handle the floating dummy fills [15].

In 2008, a technique based on the FRW algorithm was proposed to enable fast incremental variational capacitance extraction [16]. A general FRW algorithm was also proposed in [16] for arbitrary dielectric configuration, where the whole

problem domain was covered by a set of cubic transition subdomains for which the transition probability is numerically calculated online, rather than offline. This technique largely reduces the number of hops for a FRW walk, with the overhead of calculating and storing the transition probabilities for a lot of transition domains. The general FRW algorithm can be very time-consuming for a large-scale 3-D problem. A hierarchical FRW (HFRW) algorithm was later proposed for a fabric-aware extraction problem [17], where the topological variation rather than the common nontopological variation was considered. Note that the HFRW is not suitable for the general problem of capacitance extraction, because an arbitrary structure cannot be regarded as the composition of predefined motif structures. Different from most of FRW-based capacitance solvers which employ the cubic transition domain to suit the Manhattan geometry of VLSI interconnects, the technique using spherical transition domains was investigated in [18].

Although the FRW algorithm is able to handle the interconnect structure embedded in actual VLSI multielectric technology [12], [13], there is little literature that reveals the algorithm details of the 3-D FRW for multielectric capacitance extraction. Because the transition probability for a cubic domain with inhomogeneous dielectric cannot be derived analytically, extending the single-dielectric algorithm [9], [11] to the multielectric structure is not straightforward. An approach to handle the multielectric structure is using the spherical transition domain [19], because the transition probability for the spherical domain with two hemispheres having different permittivities is analytically available [21]. However, for nanometer VLSI interconnects embedded in up to 10 layers of dielectrics, this strategy will largely sacrifice the efficiency because the walk stops frequently at the dielectric interface. Another approach is to numerically characterize the transition probability for the transition domain across dielectric interface. This was first discussed in [22] for a Dirichlet problem (not capacitance extraction), where a stochastic algorithm was used to characterize the spherical transition domains with arbitrary position of dielectric interface. It should be pointed out that in most situations the sphere transition domain does not suit the capacitance extraction of VLSI interconnects. With the similar finite difference method (FDM) as presented in [16], an efficient scheme was recently proposed to characterize both the multielectric surface Green's function and the weight value [20].

The aim of this paper is to present the algorithms employed in a FRW-based 3-D capacitance solver, called RWCap. The solver is able to efficiently simulate the structures with Manhattan geometry and multilayered dielectrics, which is the major scenario for the capacitance extraction of VLSI interconnects. The main contributions of this paper are as follows.

- 1) An efficient approach is proposed for the capacitance extraction with multilayered dielectrics, which utilizes the numerically characterized surface Green's function and weight value for the cubic transition domain with two dielectric layers. With the precharacterization procedure for a given process technology, the proposed approach accelerates the FRW-based extraction for up to 160× with very small memory overhead.

- 2) A comprehensive variance reduction approach is proposed to accelerate the convergence rate of the FRW algorithm. The approach includes an importance sampling based technique to minimize the variance of weight value distribution, and a scheme combining it and the stratified sampling (SS) technique. This approach brings 3× or more speedup to the proposed FRW algorithm for multielectric capacitance extraction, without memory overhead and the loss of accuracy.
- 3) A parallel FRW algorithm on the multicore/multi-CPU platform is proposed. A space management technique and other implementation skills are presented to make the FRW-based solver efficient while handling structures with large complexity. Numerical results show the parallel RWCap achieves more than 6× speedup on a machine with 8 CPU cores.

The rest of this paper is organized as follows. The basic FRW algorithm for multielectric problem [19] and the FDM for characterizing the surface Green's function for cubic transition domain are briefly introduced in Section II. The proposed approach for multielectric capacitance extraction is presented in Section III, while the importance sampling technique and the comprehensive variance reduction approach is proposed in Section IV. The space management technique and the technique for parallelization are presented in Section V. The numerical results are given in Section VI, which validate the efficiency of proposed techniques and compare RWCap with a fast BEM-based solver. Finally, we draw the conclusions. Some preliminary results of this paper were presented in [20]. We extend it with the second contribution, more numerical results and comparisons, and more technical details.

II. BASIC FLOATING RANDOM WALK ALGORITHMS

A. FRW Algorithm for Single-Dielectric Problem

The fundamental formula of the FRW algorithm is [9]

$$\phi(\mathbf{r}) = \oint_S P(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} \quad (1)$$

where $\phi(\mathbf{r})$ is the electric potential at point \mathbf{r} , and S is a closed surface surrounding \mathbf{r} . $P(\mathbf{r}, \mathbf{r}^{(1)})$ is called the surface Green's function. For a fixed \mathbf{r} , $P(\mathbf{r}, \mathbf{r}^{(1)})$ can be regarded as the probability density function (PDF) for selecting a random point $\mathbf{r}^{(1)}$ on S . In this sense, $\phi(\mathbf{r})$ can be estimated by the mean value of $\phi(\mathbf{r}^{(1)})$, providing sufficient large number of sample points $\mathbf{r}^{(1)}$ on S are evaluated. If S is the surface of a homogeneous cube centered at \mathbf{r} , $P(\mathbf{r}, \mathbf{r}^{(1)})$ only depends on the relative position of $\mathbf{r}^{(1)}$, and is not related to the size of cube [9], [11]. More importantly, this Green's function can be derived analytically [11], and precalculated and stored as the discrete probabilities for jumping to the discretized cells of the cube surface.

In the situation that $\phi(\mathbf{r}^{(1)})$ is unknown, we apply (1) recursively to obtain the following nested integral formula:

$$\phi(\mathbf{r}) = \oint_{S^{(1)}} P^{(1)}(\mathbf{r}, \mathbf{r}^{(1)}) \oint_{S^{(2)}} P^{(2)}(\mathbf{r}^{(1)}, \mathbf{r}^{(2)}) \dots \oint_{S^{(k+1)}} P^{(k+1)}(\mathbf{r}^{(k)}, \mathbf{r}^{(k+1)}) \phi(\mathbf{r}^{(k+1)}) d\mathbf{r}^{(k+1)} \dots d\mathbf{r}^{(2)} d\mathbf{r}^{(1)} \quad (2)$$

where $S^{(i)}, (i=1, \dots, k+1)$ is the surface of the i th cube centered at $\mathbf{r}^{(i-1)}$. $P^{(i)}(\mathbf{r}^{(i-1)}, \mathbf{r}^{(i)})$, $(i=1, \dots, k+1)$, are the surface Green's functions relating the potentials at $\mathbf{r}^{(i-1)}$ to $\mathbf{r}^{(i)}$. This can be interpreted as a floating random walk procedure; for the i th hop of a walk, the maximum conductor-free cube centered at $\mathbf{r}^{(i-1)}$ is constructed and then a point $\mathbf{r}^{(i)}$ is randomly selected on the cube surface according to the discrete probabilities obtained with $P^{(i)}(\mathbf{r}^{(i-1)}, \mathbf{r}^{(i)})$. Note that, to obtain the probabilities, we only need to consider the normalized unit-size cube for the i th cube, and the corresponding positions of $\mathbf{r}^{(i-1)}$ and $\mathbf{r}^{(i)}$ in the unit-size cube. The walk terminates after k hops if the potential at point $\mathbf{r}^{(k)}$ is known, e.g., it is on a conductor surface in the problem of capacitance extraction. With the surface Green's function and derived sampling probabilities for a unit-size cube calculated in advance [11], the major cost of random walk is for geometric operations. After performing many walks, the mean value of these estimates approximates $\phi(\mathbf{r})$ very well.

For extracting capacitances among conductors, the relationship between conductor charge and potential is needed. So, a Gaussian surface G_j is constructed to enclose conductor j (the master conductor), and according to the Gauss theorem

$$Q_j = \oint_{G_j} \vec{D}(\mathbf{r}) \bullet \hat{\mathbf{n}}(\mathbf{r}) d\mathbf{r} = \oint_{G_j} F(\mathbf{r})(-\nabla\phi(\mathbf{r})) \bullet \hat{\mathbf{n}}(\mathbf{r}) d\mathbf{r} \quad (3)$$

where Q_j is the charge on conductor j , $F(\mathbf{r})$ is the dielectric permittivity at point \mathbf{r} , and $\hat{\mathbf{n}}(\mathbf{r})$ is the normal direction of G_j at \mathbf{r} . With (1) substituted to (3), the following formula can be derived

$$Q_j = \oint_{G_j} F(\mathbf{r})g \oint_{S^{(1)}} \omega(\mathbf{r}, \mathbf{r}^{(1)})P^{(1)}(\mathbf{r}, \mathbf{r}^{(1)})\phi(\mathbf{r}^{(1)})d\mathbf{r}^{(1)}d\mathbf{r} \quad (4)$$

where the weight value

$$\omega(\mathbf{r}, \mathbf{r}^{(1)}) = -\frac{\nabla_r P^{(1)}(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})}{gP^{(1)}(\mathbf{r}, \mathbf{r}^{(1)})} \quad (5)$$

where ∇_r is the gradient operator with respect to \mathbf{r} , and the constant g satisfies $\oint_{G_j} F(\mathbf{r})g d\mathbf{r} = 1$. Now the first integral in (4) can be interpreted as a stochastic sampling procedure on G_j , and the second integral can be calculated with the above FRW procedure based on (2). The only difference is the weight value (5), which contributes to the estimated value of Q_j , and relates Q_j to the conductor potentials (voltages) through the FRW procedure. Thus, averaging the weight values from many walks produces the self- and coupling-capacitances of conductor j . Note that the calculation of weight value can also be accelerated with a precharacterization process as that for the transition probability.

The above deduction derives the FRW algorithm for capacitance extraction (see Algorithm 1 in the next subsection). Note that the total computing time of FRW algorithm is roughly

$$T_{\text{total}} = N_{\text{walk}} \cdot N_{\text{hop}} \cdot T_{\text{hop}} \quad (6)$$

where N_{walk} is the number of random walks/paths, N_{hop} is the average number of hops in a walk, and T_{hop} is the average computing time for a hop. The techniques proposed in this paper reduce N_{walk} , N_{hop} and T_{hop} so that they produce an efficient FRW algorithm for capacitance extraction.

Algorithm 1 Basic FRW for multielectric problem

- 1: Load the precomputed probabilities and weight values for single-dielectric cubic transition domain;
- 2: Construct the Gaussian surface enclosing master conductor j ;
- 3: $C_{ji} := 0, \forall i; n_{\text{path}} := 0$;
- 4: **Repeat**
- 5: $n_{\text{path}} = n_{\text{path}} + 1$;
- 6: Pick a point $\mathbf{r}^{(0)}$ on Gaussian surface, and then generate a single-dielectric cubic transition domain T centered at it; pick a point $\mathbf{r}^{(1)}$ on the surface of T according to the precomputed probabilities, and then calculate the weight value ω with the help of the precomputed weight values;
- 7: **While** the current point is not on a conductor **do**
- 8: **If** the current point is on dielectric interface, construct a sphere transition domain, **otherwise** construct a single-dielectric cubic domain;
- 9: Pick a point on the domain surface, according to the probabilities of cubic domain or sphere domain;
- 10: **End**
- 11: $C_{ji} := C_{ji} + \omega$; // the current point is on conductor i
- 12: **Until** the convergence criterion is met
- 13: $C_{ji} := C_{ji}/n_{\text{path}}, \forall i$;

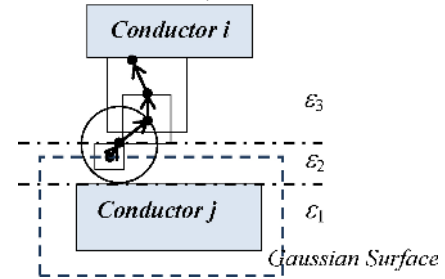


Fig. 1. Shortest random walk in the multielectric structure with the usage of spherical transition domain.

B. Basic FRW Algorithm for Multielectric Problem

The analytical surface Green's function for the spherical transition domain with different hemispheres can help the FRW to handle the situation with multiple dielectrics [19], [21]. Since the single-dielectric FRW relies on that the transition cube is within a single dielectric, the sphere transition domain is used to continue the walk stopping at dielectric interface. The resulting FRW algorithm is described as Algorithm 1.

However, while extracting the actual interconnect structure with multiple dielectrics, Algorithm 1 would induce many hops in a walk, and thus is not efficient. Fig. 1 illustrates an example of walk in the basic FRW algorithm.

C. Numerical Technique to Calculate Multielectric Surface Green's Function

With the FDM, the transition probabilities can be calculated for a transition cube with multiple dielectrics [16]. Suppose there is a cubic domain with multiple dielectric layers, as shown in Fig. 2. After discretization, the cube surface is dissected into small panels. The transition probability we want is the relationship of the potentials at the center point and a surface panel.

In each homogeneous subdomain, the Laplace equation holds

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0. \quad (7)$$

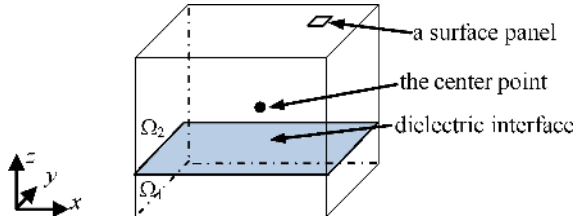


Fig. 2. Transition cube with multiple dielectrics.

At the dielectric interface, there is the continuity condition

$$\varepsilon^+ \frac{\partial \phi}{\partial z^+} = \varepsilon^- \frac{\partial \phi}{\partial z^-} \quad (8)$$

where ε^+ and ε^- are the permittivities of up and down dielectrics, respectively. Using the FDM, the following matrix equation is derived from (7) and (8):

$$\begin{bmatrix} E_{11} & E_{12} & E_{13} \\ \mathbf{0} & \mathbf{I}_2 & \mathbf{0} \\ E_{31} & \mathbf{0} & D_{33} \end{bmatrix} \begin{bmatrix} \phi_I \\ \phi_B \\ \phi_F \end{bmatrix} = \begin{bmatrix} 0 \\ f_B \\ 0 \end{bmatrix} \quad (9)$$

where ϕ_I , ϕ_B , and ϕ_F are the potential unknowns on inner grid points, the surface panels, and dielectric interfaces, respectively. f_B is the boundary potentials, which may be regarded as the Dirichlet boundary condition. The E matrices are sparse finite difference matrices discretizing the Laplace operator. The third block row of (9) is derived from (8), and therefore D_{33} is a diagonal matrix. And, \mathbf{I}_2 is an identity matrix. We then have

$$\phi_I = -(\mathbf{E}_{11} - \mathbf{E}_{13} \mathbf{D}_{33}^{-1} \mathbf{E}_{31})^{-1} \mathbf{E}_{12} f_B. \quad (10)$$

It expresses the relationship between the inner points and the boundary points. Suppose the center of the cube is the k th inner grid points. Then, the potential of center point is

$$\phi_k = \mathbf{e}_k^T \phi_I = -((\mathbf{E}_{11} - \mathbf{E}_{13} \mathbf{D}_{33}^{-1} \mathbf{E}_{31})^{-T} \mathbf{e}_k)^T \mathbf{E}_{12} f_B \quad (11)$$

where \mathbf{e}_k denotes a column vector where the k th element is 1 and otherwise 0. From (11), we can see that the row vector

$$\mathbf{P}_k = -((\mathbf{E}_{11} - \mathbf{E}_{13} \mathbf{D}_{33}^{-1} \mathbf{E}_{31})^{-T} \mathbf{e}_k)^T \mathbf{E}_{12} \quad (12)$$

represents the discrete probabilities for transition from the center point to the boundary panels, and is what we want.

The numerical technique can be generalized to consider different boundary conditions, such as the Neumann boundary, floating potential boundary, etc. [16]. However, it is impractical to precompute and tabulate the transition probabilities for the maximum conductor-free transition cubes with all possible dielectric configurations. The question is how many transition domains should be precharacterized to well balance the memory usage and computational efficiency of the multidielectric capacitance extraction.

III. MULTIDIELECTRIC FRW ALGORITHM WITH THE PRECHARACTERIZED PROBABILITIES AND WEIGHT VALUES

In this section, we first propose an approach to precharacterize the transition cubes with two dielectric layers. Then, the details of the FDM techniques for the precharacterization are presented. The multidielectric FRW algorithm is finally given.

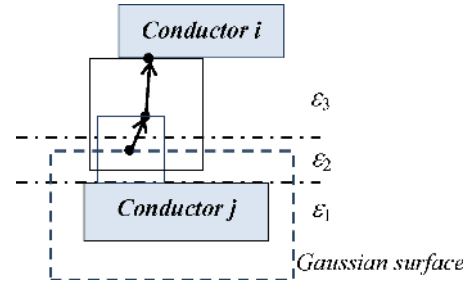


Fig. 3. Shortest random walk in the multidielectric structure with the usage of multidielectric transition probability and weight value.

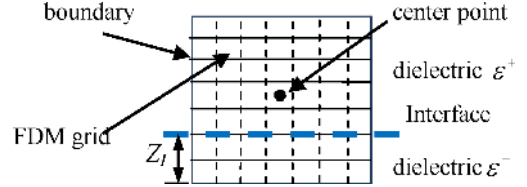


Fig. 4. Cubic transition domain for precharacterizing the multidielectric process technology (2-D cross-section view).

A. Basic Idea

Modern VLSI process technology involves multiple layers of metal wires and dielectrics. The coupling capacitance mainly exists between the wires at the same layer or at two adjacent layers, due to the densely routed wires. If the hop in FRW walk is able to cross one dielectric interface, the number of hops in a walk may not increase much as compared with that for single-dielectric circumstance. Therefore, numerically characterizing the surface Green's function for the cubic transition domain with two dielectric layers may bring sufficient benefit to the extraction of multidielectric VLSI structures. Fig. 3 shows an example of walk under the assumption that the hop is able to cross one dielectric interface. Comparing Figs. 1 and 3, we can see that the number of hops is largely reduced. On the other hand, for a specified technology with n_d dielectric layers (adjacent layers with same permittivity are regarded as one layer), there are no more than n_d adjacent dielectric pairs with distinct pair of permittivities. For each pair of permittivities, the surface Green's function is precharacterized as Green's function tables (GFTs), and the weight value is precharacterized as weight value tables (WVTs). The computing time for the precharacterization procedure and the memory usage for recalling the GFTs and WVTs should be moderate for a given process technology.

In the precharacterization procedure, we only consider the unit-size cubic transition domain, which is discretized with FDM grids. The cube includes two dielectric layers, and the position of dielectric interface is set to conform to the FDM grid (see Fig. 4 for a cross-section view). For each dielectric configuration (permittivity pair and interface position), the FDM technique based on Section II-C is employed to derive the corresponding GFT and WVT.

B. Details of the Precharacterization Procedure

For precharacterizing the surface Green's function accurately, some details of FDM should be considered. First, to avoid the potential unknown setting on the edge or corner of the cube, each surface unknown is set at the center of the

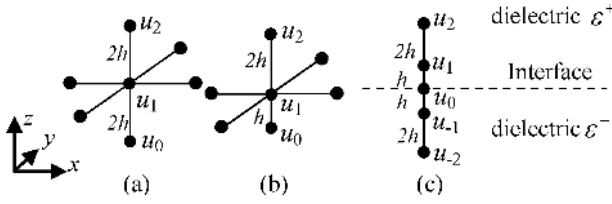


Fig. 5. Illustration of finite difference schemes for (a) inner grid point, (b) grid point near boundary, and (c) point on interface.

boundary panel. Also note that there is an unknown at the center of cube. To connect these unknowns with a FDM grid, we first divide the cube into $N \times N \times N$ equal-sized cells and then attach one unknown to each cell's center. Here N should be an odd number (see Fig. 4). These N^3 unknowns form the ϕ_I in (9) for inner grid points. The boundary unknowns attached to panel's center point form ϕ_B . Because the height of interface Z_I has the value of $1/N, 2/N, \dots, (N-1)/N$, extra unknowns ϕ_F are defined on the interface surface to connect inner points upward and downward.

Different finite difference (FD) schemes are used to generate the coefficient matrix blocks in (9). For the uniform inner grid point [see Fig. 5(a)], the standard seven-point differential scheme with second-order accuracy is used. Fig. 5(b) shows the grid points near the boundary, where the grid step varies (we assume the inner grid size is $2h$). For this nonuniform grid, the Lagrange interpolation is used to derive the differential scheme with second-order accuracy. For example, the z -direction derivative for the point in Fig. 5(b) is approximated with

$$\frac{\partial^2 \phi}{\partial z^2} \approx \frac{2u_0}{3h^2} - \frac{u_1}{h^2} + \frac{u_2}{3h^2}. \quad (13)$$

Here u denotes the potential on grid point. For the point on interface, in order to be consistent with the standard seven-point differential scheme for inner point, the formula with second-order accuracy is derived to approximate (8). As shown in Fig. 5(c), on each side of interface the potentials of two points are used to approximate $\partial\phi/\partial z^+$ or $\partial\phi/\partial z^-$. For this example, the derived difference equation for (8) is

$$\varepsilon^+ \frac{-8u_0 + 9u_1 - u_2}{6h} = \varepsilon^- \frac{-8u_0 - 9u_{-1} + u_{-2}}{6h}. \quad (14)$$

It should be pointed out that, if using the first-order difference equation for (8), we find out that the obtained surface Green's function would include some obvious error.

In the FDM solution, the height of interface Z_I cannot be $1/2$, which omits an important situation when the walk stops at a dielectric interface. A special treatment should be taken. As shown in Fig. 6(a), which is a side view, the cube edge should be divided into even segments, so that there is no boundary panel across the interface. Therefore, the center point is not an inner grid point. As shown in Fig. 6(b), the potential at the center point can be approximated by its eight neighbors, that is

$$u_c = \frac{\varepsilon^+ u_9 + \varepsilon^- u_{10}}{\varepsilon^+ + \varepsilon^-} = \frac{\varepsilon^+(u_1 + u_2 + u_3 + u_4) + \varepsilon^-(u_5 + u_6 + u_7 + u_8)}{4(\varepsilon^+ + \varepsilon^-)} \quad (15)$$

where u_c denotes the potential of the center point. In practice, a vector e'_k is constructed to record the positions of nodes

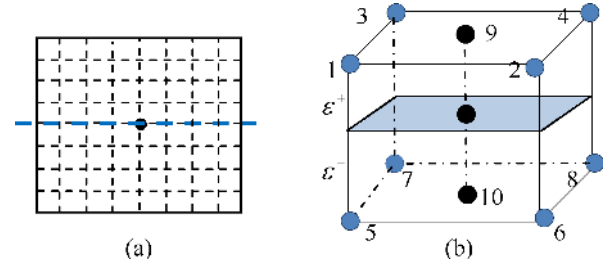


Fig. 6. Situation where the height of interface $Z_I=1/2$. (a) Side view of the unit-size cube. (b) FDM grids at the center of cube.

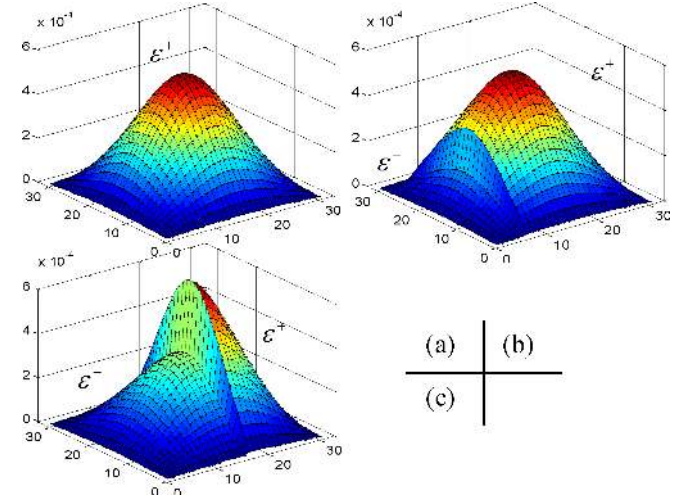


Fig. 7. GFT distributions on cube boundaries when $\varepsilon^- = 2.6$ and $\varepsilon^+ = 5$ (a) on the top face, (b) on the side wall, for $Z_I = 1/5$, and (c) on the side wall, for $Z_I = 1/2$.

from 1 to 8 in the vector ϕ_I and their contributions to u_c , according to (15). A similar formula to (12) can be used to efficiently calculate the discrete probabilities for the transition cube whose dielectric interface is at height $1/2$

$$P'_k = -((E_{11} - E_{13} D_{33}^{-1} E_{31})^{-T} e'_k)^T E_{12}. \quad (16)$$

Fig. 7 shows the distributions of the precharacterized GFT for the configuration: $\varepsilon^- = 2.6$, $\varepsilon^+ = 5$. Note that the plot in Fig. 7(a) is similar to that of single-dielectric surface Green's function in [11], and the plots in Fig. 7(b) and (c) demonstrate that there is much larger probability to hop toward the dielectric region with larger permittivity. By setting $\varepsilon^- = \varepsilon^+$, we have used the single-dielectric Green's function to validate the accuracy of the numerical GFTs obtained with FDM.

To calculate the weight value in (5), we first derive

$$\omega(\mathbf{r}, \mathbf{r}^{(1)}) = -\frac{1}{g \cdot L} \cdot \frac{\frac{\partial P(\mathbf{r}, \mathbf{r}^{(1)})}{\partial x} n_x + \frac{\partial P(\mathbf{r}, \mathbf{r}^{(1)})}{\partial y} n_y + \frac{\partial P(\mathbf{r}, \mathbf{r}^{(1)})}{\partial z} n_z}{P(\mathbf{r}, \mathbf{r}^{(1)})} \quad (17)$$

where L is the edge length of the first cube, and we assume $\hat{n}(\mathbf{r}) = [n_x, n_y, n_z]^T$. Here $P(\mathbf{r}, \mathbf{r}^{(1)})$ denotes the surface Green's function for the unit-size cube, and \mathbf{r} and $\mathbf{r}^{(1)}$ is the corresponding points in the unit-size cube. Since we have obtained the GFT representing $P(\mathbf{r}, \mathbf{r}^{(1)})$, the finite difference formula for partial derivative should be employed to calculate the weight value. For example, $\partial P/\partial x$ stands for the sensitivity of the probabilities with respect to the x -axis coordinate of the center point of cube. Since P_k in (11) is the GFT associated with the center point, we denote the GFTs associated with its

six neighbor points to be \mathbf{P}_{k+1} , \mathbf{P}_{k-1} , \mathbf{P}_{k+N} , \mathbf{P}_{k-N} , \mathbf{P}_{k+N^2} , and \mathbf{P}_{k-N^2} . Then, with the centered difference formula we have

$$\frac{\partial \mathbf{P}_k}{\partial x} \approx \frac{\mathbf{P}_{k+1} - \mathbf{P}_{k-1}}{4h} = -((\mathbf{E}_{11} - \mathbf{E}_{13} \mathbf{D}_{33}^{-1} \mathbf{E}_{31})^{-T} \tilde{\mathbf{e}}_1)^T \mathbf{E}_{12} \quad (18)$$

$$\frac{\partial \mathbf{P}_k}{\partial y} \approx \frac{\mathbf{P}_{k+N} - \mathbf{P}_{k-N}}{4h} = -((\mathbf{E}_{11} - \mathbf{E}_{13} \mathbf{D}_{33}^{-1} \mathbf{E}_{31})^{-T} \tilde{\mathbf{e}}_N)^T \mathbf{E}_{12} \quad (19)$$

$$\frac{\partial \mathbf{P}_k}{\partial z} \approx \frac{\mathbf{P}_{k+N^2} - \mathbf{P}_{k-N^2}}{4h} = -((\mathbf{E}_{11} - \mathbf{E}_{13} \mathbf{D}_{33}^{-1} \mathbf{E}_{31})^{-T} \tilde{\mathbf{e}}_{N^2})^T \mathbf{E}_{12}. \quad (20)$$

Here the vector $\tilde{\mathbf{e}}_i$, ($i = 1, N, N^2$) is defined as

$$\tilde{\mathbf{e}}_i(j) = \begin{cases} 1/4h, & j = k + i \\ -1/4h, & j = k - i. \\ 0, & \text{otherwise} \end{cases} \quad (21)$$

Equations (18)–(20) are the formulas for calculating the partial derivatives in (17). With a small value of h , they have sufficient accuracy. In practice, the values of $\frac{\partial P/\partial x}{P}$, $\frac{\partial P/\partial y}{P}$, and $\frac{\partial P/\partial z}{P}$ for each boundary panel (the position of $\mathbf{r}^{(1)}$) are precomputed and stored as the WVTs. With the WVTs, the weight value (17) in actual walk can be calculated quickly.

It should be pointed out that precharacterizing the WVT for the multidielectric cube enlarges the size of the first transition domain, and is very important for improving the computational efficiency. Since ω is inversely proportional to the size of the first cube L , and without the multidielectric WVT the first cube is bounded by the nearest dielectric interface, the value of ω would be very large if there is a thin dielectric layer near the Gaussian surface. This large sample value would further slow down the convergence of MC procedure. Therefore, the multidielectric WVT can enlarge the size of first transition cube and thus prominently reduces the number of walks to attain a certain accuracy goal [20].

For a predefined value of N (segment number along cube edge), N positions of dielectric interface are considered. Totally, $4N$ times of equation solution with the coefficient matrix of $(\mathbf{E}_{11} - \mathbf{E}_{13} \mathbf{D}_{33}^{-1} \mathbf{E}_{31})^T$ (N for GFTs and $3N$ for WVTs) are involved for a pair of permittivities. As for the memory usage, $6N^2$ real numbers are needed to present a single GFT. For the $4N$ GFTs and WVTs which characterize a dielectric pair, the memory space is needed to store $24N^3$ about numbers. If $N = 31$ and the double-precision number is used, the storage is about 5.5 MB. If three dielectric layers are allowed in the precharacterized transition cube, the memory storing the three-layer GFTs and WVTs would be about $N/2$ times of that for two-layer GFTs and WVTs, due to the two dielectric interfaces. This will make the extra memory for the two-layer, three-layer GFTs and WVTs approach 1 GB, if a process technology, including 10 three-consecutive-dielectric configurations. Besides, allowing three dielectric layers in the precharacterized transition domain will increase the complexity of implementation, such as matching the interface heights in actual transition cube and those in the precharacterized. Therefore, in this paper we only precharacterize the domain with two dielectric layers to well balance the memory usage and computational efficiency.

C. FRW Algorithm with Multidielectric GFTs and WVTs

Algorithm 2 describes the FRW algorithm utilizing the multidielectric GFTs and WVTs.

Algorithm 2 FRW using multidielectric GFTs and WVTs

- 1: Load the precomputed transition probabilities, weight values for single-dielectric cubic transition domain;
- 2: Load the precomputed multidielectric GFTs and WVTs;
- 2: Construct the Gaussian surface enclosing master conductor j ;
- 3: $C_{ji} := 0, \forall i; npath := 0$;
- 4: **Repeat**
- 5: $npath := npath + 1$;
- 6: Pick a point $\mathbf{r}^{(0)}$ on Gaussian surface, and then generate a cubic transition domain (may contain two dielectrics) T centered at it; pick a point $\mathbf{r}^{(1)}$ on the surface of T according to the precomputed probabilities, and then calculate the weight value ω with the help of the WVTs;
- 7: **While** the current point is not on a conductor **do**
- 8: Construct the largest conductor-free cubic domain containing at most two dielectrics;
- 9: Pick a point on the domain surface, according to the transition probabilities of cubic domain;
- 10: **End**
- 11: $C_{ji} := C_{ji} + \omega$; // the current point is on conductor i
- 12: **Until** the stopping criterion is met
- 13: $C_{ji} := C_{ji}/npath, \forall i$;

In Algorithm 2, a shrinking operation for the cube may be performed after generating the transition cube (in step 6 and step 8). Its aim is to adjust the size of the transition cube to make the height of dielectric interface to be i/N ($i = 1, 2, \dots, N - 1$), or $1/2$ of the cube size. Therefore, the precharacterized GFTs and WVTs can be used without loss of accuracy.

IV. TECHNIQUES FOR VARIANCE REDUCTION

In this section, we first review the basic ideas of importance sampling and SS, followed by the variance reduction techniques proposed in [14]. Then, we propose the technique to minimize the variance of weight value distribution in the FRW, and a comprehensive variance reduction scheme.

A. Background

The MC method can be used to calculate a multidimensional integral

$$I = \int_{\Omega} f(\mathbf{x}) d\mathbf{x}. \quad (22)$$

If we pick n random samples \mathbf{x}_i uniformly distributed in Ω

$$I_n = A \cdot \frac{\sum_{i=1}^n f(\mathbf{x}_i)}{n} \quad (23)$$

becomes an estimate for I , where $A = \int_{\Omega} d\mathbf{x}$. Note that I_n is the mean value of $A \cdot f(\mathbf{x}_i)$, and is also a random quantity due to the randomness of \mathbf{x}_i . The variance of function $A \cdot f(\mathbf{x})$ can be approximated by the variance of the discrete estimates $A \cdot f(\mathbf{x}_i)$

$$Var_n = \frac{A^2 \sum_{i=1}^n (f(\mathbf{x}_i))^2 - n \cdot I_n^2}{n}. \quad (24)$$

Due to the central limit theorem [23], I_n obeys the normal probability distribution, and its standard deviation (Std) can be estimated with

$$err_n \approx \frac{\sqrt{Var(A \cdot f(\mathbf{x}))}}{\sqrt{n}} \approx \frac{\sqrt{Var_n}}{\sqrt{n}} \quad (25)$$

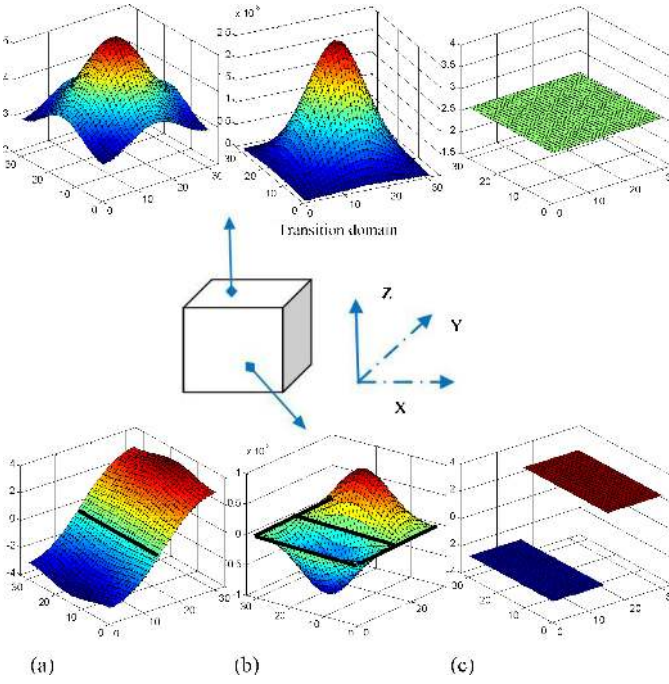


Fig. 8. Values in WVT for $\mathbf{r}^{(1)}$ on top and side faces of the cube for the (a) ω of the standard FRW, (b) ω_c of the variant FRW, and (c) $\bar{\omega}$ using the importance sampling.

where $\text{Var}(A \cdot f(\mathbf{x}))$ denotes the variance of function $A \cdot f(\mathbf{x})$. err_n in (25) is also called the $1-\sigma$ error of I_n . With (25), it can be observed that the error of I_n is inversely proportional to the square root of the number of samples. Since I_n conforms to the normal distribution, the $1-\sigma$ error err_n means the error bound at the confidence level of 68%. And, it has the confidence level of 99.7% that the result is within the error of $3 \cdot \text{err}_n$.

According to the above discussion, reducing the variance of the integrand $f(\mathbf{x})$ brings smaller err_n , and therefore I_n with higher accuracy, for a given number of sample points. The idea of importance sampling (IS) is to find a positive valued function $q(\mathbf{x})$ such that $\int_{\Omega} q(\mathbf{x}) d\mathbf{x} = 1$ and $q(\mathbf{x})$ is nearly proportional to the function $f(\mathbf{x})$. Then, (22) is converted to

$$I = \int_{\Omega} \frac{f(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x}. \quad (26)$$

We can use $q(\mathbf{x})$ as a probability density function for the sampling of Ω , and evaluate $f(\mathbf{x}_i)/q(\mathbf{x}_i)$ as the estimate in the MC procedure. This means that $f(\mathbf{x})/q(\mathbf{x})$ is regarded as the integrand function in the MC integration. Since $q(\mathbf{x})$ is nearly proportional to $f(\mathbf{x})$, the variance of the new integrand is much less than that of $f(\mathbf{x})$. Therefore, the MC procedure corresponding to (26) will converge much faster for a given error threshold.

Another variance reduction technique is the SS, whose idea is to divide the original integral to several integrals on its subdomains (also called strata), and then calculate them separately. For example, dividing the Ω in (22) into two equal-sized subdomains Ω_a and Ω_b , we have

$$I = I_a + I_b = \int_{\Omega_a} f(\mathbf{x}) d\mathbf{x} + \int_{\Omega_b} f(\mathbf{x}) d\mathbf{x}. \quad (27)$$

If the MC method is used to calculate I_a and I_b separately, the formula (23) can be substituted with

$$I'_n = I_{na}^{(a)} + I_{nb}^{(b)} = \frac{A}{2} \cdot \frac{\sum_{i=1}^{n_a} f(\mathbf{x}_{a,i})}{n_a} + \frac{A}{2} \cdot \frac{\sum_{i=1}^{n_b} f(\mathbf{x}_{b,i})}{n_b} \quad (28)$$

where n_a and n_b are the sampling numbers in Ω_a and Ω_b , respectively ($n_a + n_b = n$). Due to the stochastic independence of $I_{na}^{(a)}$ and $I_{nb}^{(b)}$, the variance of I'_n is the sum of the variances of $I_{na}^{(a)}$ and $I_{nb}^{(b)}$. It is proved that [23]

$$\text{Var}(I'_n) < \text{Var}(I_n) \quad (29)$$

if $f(\mathbf{x})$ has different mean value in subdomains Ω_a and Ω_b . Furthermore, $\text{Var}(I'_n)$ achieves its minimum value if $(n_a/n_b)^2$ equals to the ratio of variances of function $f(\mathbf{x})$ in Ω_a and Ω_b . Because the $\text{Var}(I_n)$ is the square of the Std in (25), calculating (28) with the idea of SS produces the estimation to the original integral with less $1-\sigma$ standard error. Thus, we need fewer MC samples to achieve a specified accuracy criterion.

The above theory of statistics is applicable to the FRW algorithm, because (4) is an integral. With the formula for the variance of capacitance estimates, similar to (24), the $1-\sigma$ error of capacitance result is predictable [9]–[11]. So, usually the accuracy criterion is set as the termination condition of FRW algorithm. In practice, the algorithm checks the error after every certain number of walks. Once the $1-\sigma$ error is below the specified accuracy goal, the algorithm will terminate.

In [14], the variance reduction techniques were proposed for the FRW-based capacitance extraction. First, the capacitance was regarded as the integral of normal electric field intensity on the Gaussian surface. Based on that the electric field diminishes with the inverse of distance, the function $q(\mathbf{x})$ for IS was simply defined to be proportional to $1/D(\mathbf{r})$, where $D(\mathbf{r})$ was the distance of point \mathbf{r} to the conductor inside the Gaussian surface. The SS was also applied, which decomposed the first transition cube's surface into 16 pieces (strata), and the FRW calculations were performed for each strata separately. It should be pointed out that a variant FRW formula, instead of (4) (referred to as the standard FRW formula), was used in [14]

$$Q_j = \oint_{G_j} F(\mathbf{r}) g \oint_{S^{(1)}} \omega_c(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r}. \quad (30)$$

The variant weight value is

$$\omega_c(\mathbf{r}, \mathbf{r}^{(1)}) = -\frac{\nabla_r P^{(1)}(\mathbf{r}, \mathbf{r}^{(1)}) \cdot \hat{\mathbf{n}}(\mathbf{r})}{g}. \quad (31)$$

Accordingly, the sampling scheme in $S^{(1)}$ obeys the uniform probabilities, rather than the probabilities derived by the surface Green's function. In the next two subsections, we will show that the variant FRW formula (30) causes worse convergence behavior than the standard FRW (4), and the variance reduction techniques were not fully exploited in [14].

B. Importance Sampling with the Weight Values Averaged

In the FRW algorithm, the weight value (5) or (17), is the estimate of capacitance in the MC procedure. From another viewpoint, it is the estimation to the integrand function (the normal electric field intensity) of the integral (4). Although the integrand is unknown, we can instead reduce the variance of weight values by transforming the FRW formula with the IS technique. It is noticed that with the help of WVT,

we actually know the possible values of the weight value. Suppose \mathbf{r} is on the top surface of the Gaussian surface, which means $\hat{\mathbf{n}}(\mathbf{r})$ is along the z -direction. In Fig. 8, we draw the distributions of the opposite of weight value for the $\mathbf{r}^{(1)}$ on top and side faces of the cube. Fig. 8(a) corresponds to the ω in the standard FRW, while Fig. 8(b) corresponds to the ω_c in the variant FRW formula (31).

With Fig. 8(a), (b) and (17), we know that ω and ω_c are nonpositive for $\mathbf{r}^{(1)}$ on the top face, while they may approach to zero for $\mathbf{r}^{(1)}$ on the side face (outlined with bold lines). This kind of value distribution demonstrates that the weight value has a certain degree of variance. And, the variance of ω_c seems larger than that of ω , which is verified by the numerical experiments showing a slower convergence rate of the variant FRW.

We now seek a function $q(\mathbf{x})$ to modify the integral formula (4) so as to reduce the variance of resulting weight value, based on the precomputed WVTs and GFTs. For the transition cube with a specific dielectric configuration, we first calculate

$$K = \oint_S |\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})| d\mathbf{r}^{(1)} \quad (32)$$

where S and $P(\mathbf{r}, \mathbf{r}^{(1)})$ are the surface and the surface Green's function of the unit-size cube, respectively. Since \mathbf{r} is the center of cube, K is a positive constant. We transform (4) to

$$\begin{aligned} Q_j &= \oint_{G_j} F(\mathbf{r})g \oint_S -\frac{\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})}{L \cdot g} \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r} \\ &= \oint_{G_j} F(\mathbf{r})g \oint_S \frac{-K \cdot \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})}{L \cdot g |\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})|} \cdot \\ &\quad \frac{|\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})|}{K} \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r} \end{aligned} \quad (33)$$

where L is the size of the first cube, with which the integral domain is converted from $S^{(1)}$ to the unit-size cube S .

We define function $q(\mathbf{x})$ to be

$$q(\mathbf{r}, \mathbf{r}^{(1)}) = \frac{|\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})|}{K} = \frac{|\omega(\mathbf{r}, \mathbf{r}^{(1)})| P(\mathbf{r}, \mathbf{r}^{(1)})}{K}. \quad (34)$$

Here $\omega(\mathbf{r}, \mathbf{r}^{(1)})$ denotes the weigh value for the unit-size cube, which is a little bit different from (5). According to (32), $q(\mathbf{r}, \mathbf{r}^{(1)})$ can serve as the PDF for sampling S . Then

$$Q_j = \oint_{G_j} F(\mathbf{r})g \oint_S \frac{-K \cdot \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})}{L \cdot g |\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})|} \cdot q(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r}. \quad (35)$$

This suggests a new sampling scheme, where the weight value is

$$\tilde{\omega}(\mathbf{r}, \mathbf{r}^{(1)}) = \begin{cases} -K/(L \cdot g), & \text{if } \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) > 0 \\ K/(L \cdot g), & \text{if } \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) < 0. \end{cases} \quad (36)$$

Note that it is impossible that $\mathbf{r}^{(1)}$ has a value such that $\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) = 0$, according to the sampling probability function (34). In Fig. 8(c), we draw the distribution of the opposite of the new weight value (36). It is almost constant, for cubes with same size and dielectric configuration. Thus, the new FRW scheme may exhibit much less variance.

Now, the point sampling on the first cube obeys the new probabilities obtained from $q(\mathbf{r}, \mathbf{r}^{(1)})$, which can be generated

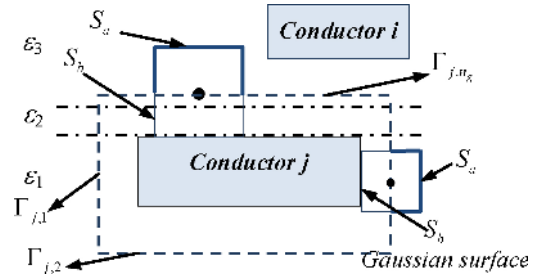


Fig. 9. 2-D view illustrating the n_g faces of a Gaussian surface, and the two half surfaces of the first transition cube.

by multiplying the original GFT and WVT, element by element. Similar to the derivation of (17), the new probabilities $q(\mathbf{r}, \mathbf{r}^{(1)})$ differ with the different direction of $\hat{\mathbf{n}}(\mathbf{r})$. We tabulate it with tables called GFT_ISS, which have the same storage size as the original WVTs. The new weight values (36) are also precomputed and tabulated, but require much less storage.

C. Comprehensive Variance Reduction Scheme

Although the weight values are averaged with formula (35), they still fluctuate when the first cube varies with different positions of \mathbf{r} . This kind of variance can be further handled with the SS technique. Since for a given cube the weight value has only two nonzero values of (36), decomposing the cube's surface into 16 stratas as in [14] is unnecessary. We just divide the cube's surface into two stratas, one with positive weight value and the other with negative weight value. Instead, the Gaussian surface is decomposed to capture the different variances of weight values on its different faces. Suppose the Gaussian surface has n_g faces (see Fig. 9). The original (4) is converted to

$$\begin{aligned} Q_j &= \sum_{k=1}^{n_g} \int_{\Gamma_{j,k}} F(\mathbf{r})g \oint_S -\frac{\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})}{L \cdot g} \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r} \\ &= \sum_{k=1}^{n_g} \int_{\Gamma_{j,k}} F(\mathbf{r})g \int_{S_a} -\frac{\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})}{L \cdot g} \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r} \\ &\quad + \sum_{k=1}^{n_g} \int_{\Gamma_{j,k}} F(\mathbf{r})g \int_{S_b} -\frac{\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})}{L \cdot g} \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r} \end{aligned} \quad (37)$$

where $\Gamma_{j,k}$ stands for the k th face of the Gaussian surface G_j . S_a stands for the surface of the scaled half cube outside G_j , and S_b stands for the surface of the scaled half cube inside G_j . With this partition, we observe that $\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})$ always has positive value for $\mathbf{r}^{(1)}$ on S_a , and negative value for $\mathbf{r}^{(1)}$ on S_b (see Lemma 1 in the Appendix). Now, Q_j becomes the summation of $2n_g$ integrals. Each integral can be calculated separately. For example

$$I_k = \int_{\Gamma_{j,k}} F(\mathbf{r})g \int_{S_a} -\frac{\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})}{L \cdot g} \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r}, (1 \leq k \leq n_g). \quad (38)$$

Because $\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})$ does not change sign on S_a , we define

$$K_a = \int_{S_a} \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) d\mathbf{r}^{(1)}. \quad (39)$$

Then

$$q_a(\mathbf{r}, \mathbf{r}^{(1)}) = \frac{\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})}{K_a} \quad (40)$$

becomes a PDF for the second integral in (38). We derive

$$I_k = A_k \int_{\Gamma_{j,k}} \frac{F(\mathbf{r})g}{A_k} \int_{S_a} -\frac{K_a}{L \cdot g} q_a(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r}, (1 \leq k \leq n_g) \quad (41)$$

Algorithm 3 FRW using the variance reduction techniques

```

1: For each dielectric configuration occurring in the test
   case, load the available GFT, GFT_ISs and WVTs.
2: Construct the Gaussian surface enclosing master
   conductor j;
3:  $C_{ji} := 0, \forall i; n_k := 0, \forall k;$ 
4: Repeat
5: Pick a point  $r^{(0)}$  on Gaussian surface, and then generate
   a cubic transition domain (may contain two dielectrics)
   T centered at it; pick a point  $r^{(1)}$  on the surface of T
   according to the precomputed GFT_IS, and then calcu-
   late the weight value  $\tilde{\omega}_k$  with (46) considering the posi-
   tion of  $r^{(0)}$  and  $r^{(1)}$ ;
6: While the current point is not on a conductor do
7:   Construct the largest conductor-free cubic domain
   containing at most two dielectrics;
8:   Pick a point on the domain surface, according to the
   precomputed GFT of the cubic domain;
9: End
10: Register  $\tilde{\omega}_k$  to strata  $k$  for  $C_{ji}$ ; //the current point is on
   conductor  $i$ 
11:  $n_k := n_k + 1;$ 
12: Calculate the standard error with formula similar to
   (45);
13: Until the stopping criterion is met
14: Calculate  $C_{ji}, \forall i$  with formula similar to (44).

```

where $A_k = \int_{\Gamma_{j,k}} F(\mathbf{r})g d\mathbf{r}$. This can be interpreted by a FRW procedure, where the sampling on S_a obeys the PDF of $q_a(\mathbf{r}, \mathbf{r}^{(1)})$ and the corresponding weight value is

$$\tilde{\omega}_k(\mathbf{r}, \mathbf{r}^{(1)}) = \frac{-K_a}{L \cdot g}, (1 \leq k \leq n_g). \quad (42)$$

Similar derivation can be applied to

$$I_k = \int_{\Gamma_{j,k-n_g}} F(\mathbf{r})g \int_{S_b} -\frac{\nabla_{\mathbf{r}} P(\mathbf{r}, \mathbf{r}^{(1)}) \cdot \hat{\mathbf{n}}(\mathbf{r})}{L \cdot g} \phi(\mathbf{r}^{(1)}) d\mathbf{r}, (k > n_g) \quad (43)$$

provided that K_b and $q_b(\mathbf{r}, \mathbf{r}^{(1)})$ are defined similarly to K_a and $q_a(\mathbf{r}, \mathbf{r}^{(1)})$, respectively. And for $k > n_g$, we let A_k equal to A_{k-n_g} . Suppose n_k walks are used to calculate I_k , ($k = 1, \dots, 2n_g$), and the weight values returned are $\tilde{\omega}_{k,1}, \dots, \tilde{\omega}_{k,n_k}$. The estimation for (37) with the SS technique is computed as

$$\tilde{Q}_j = \sum_{k=1}^{2n_g} \theta_k = \sum_{k=1}^{2n_g} \left(\frac{A_k}{n_k} \sum_{i=1}^{n_k} \tilde{\omega}_{k,i} \right). \quad (44)$$

And, the corresponding variance is computed piecewise as

$$\text{Var}(\tilde{Q}_j) = \sum_{k=1}^{2n_g} \left(\frac{A_k^2}{n_k^2} \sum_{i=1}^{n_k} \tilde{\omega}_{k,i}^2 - \frac{\theta_k^2}{n_k} \right). \quad (45)$$

It should be pointed out that (44) and (45) need to be explained for calculating the capacitances and variance of capacitance estimates. Comparing (39), (40) with (32), (34), we conclude that the new sampling scheme is compatible with the scheme in the last subsection. Thus, the precomputed GFT_ISs can still be used. Because K_a equals the half of K in (32) (see Theorem 2 in the Appendix), we have

$$\tilde{\omega}_k(\mathbf{r}, \mathbf{r}^{(1)}) = \begin{cases} \frac{-K}{2L \cdot g}, & 1 \leq k \leq n_g \\ \frac{K}{2L \cdot g}, & n_g < k \leq 2n_g. \end{cases} \quad (46)$$

So, the weight value tabulated in last subsection is still valid.

In our implementation, the values of n_k , ($k = 1, \dots, 2n_g$) are not predefined. Instead, the FRW procedure corresponding to the IS technique in last subsection is performed. The actual walks are registered to the $2n_g$ stratas, followed by calculating the capacitance value and error estimation with formulas similar to (44) and (45). Finally, the IS technique in [14] can also be applied to handle the first integral. All these

Algorithm 4 Insert to octree (conductor B, node T)

```

1: If T is a leaf node then
2:   For each b in the candidate list of T do
3:     If b dominate B then return;
4:     Elseif B dominate b then
5:       Remove b from the candidate list of T;
6:     Endif
7:   Endfor
8:   Add B to the candidate list of T;
9:   If the size of T's candidate list is larger than
   threshold then
10:    Divide T into 8 child nodes:  $T_1, \dots, T_8;$ 
11:    For each b in the candidate list of T do
12:      For  $i = 1$  to 8,  $\text{InsertToOctree}(b, T_i);$  EndFor
13:    EndFor
14:    Endif
15:  Else
16:    For each child c of T do
17:       $\text{InsertToOctree}(B, c);$ 
18:    Endfor
19:  Endif

```

consist of a comprehensive variance reduction scheme, and the resulting FRW algorithm is described as Algorithm 3.

V. SPACE MANAGEMENT TECHNIQUE AND PARALLEL IMPLEMENTATION

In this section, the space management technique is first presented to reduce the computing time for structures with a large amount of conductors. Then, the parallel technique is presented to boost the computational speed of FRW-based capacitance extraction on the multicore/multi-CPU platform.

A. Space Management Technique

For each FRW hop, the distance from the current point to conductors should be measured to construct the maximum conductor-free transition cube. Its computing time should be reduced as short as possible because millions of hops are involved in the FRW algorithm. In [27], several data structures were discussed to speed up the distance queries in the FRW algorithm. Here we present a space management technique based on the octree spatial data structure [24] to organize the conductors in the problem such that the distance is calculated for only a few of conductors for each hop. This technique is required for the case involving many conductors. Each octree node represents a cubic spatial domain, which may be decomposed into eight subdomains and each one corresponds to a child node in the tree. For each node of octree, we want to have a number of possible nearest conductors (candidate list) for the spatial points it represents. To build such an octree, we first define the dominant relationship.

Definition 1: T is a cubic domain, B_1 and B_2 are conductors. B_1 dominate B_2 regarding T, iff. $\forall P \in T, d(P, B_1) \leq d(P, B_2)$, where $d(\cdot)$ is the function of the ∞ -norm distance in 3-D space.

The octree is constructed through a procedure of inserting conductors. First, the octree only has one node (root node), representing the whole problem space. Then, all conductors in the problem are inserted to this node, one by one, with Algorithm 4.

Note that a threshold is predefined as the maximum size of the candidate list. In step 12 and step 17 of Algorithm 4, there are recursive calls. The dominant relationship of conductor blocks can be judged with their distances to the cubic domain of current node.

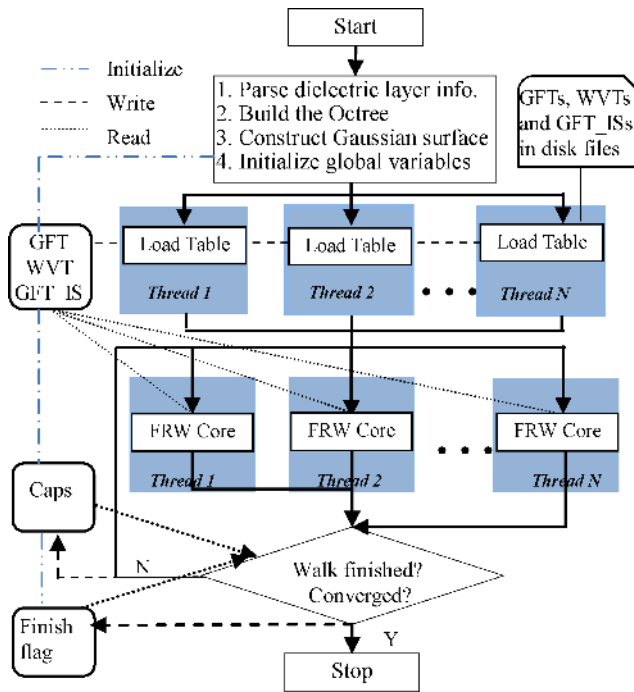


Fig. 10. Flowchart of the parallel FRW on multicore platform.

Once the octree is established for the extracted structure, in each hop only the conductors in the candidate list of the leaf node that the current point belongs to are enquired to calculate distance. So, the time complexity of each hop is about $O(h_o + t_o)$, where h_o is the height of octree and t_o is the threshold. We have tested two structures with 201 and 402 wires respectively. Numerical results show the running time of FRW algorithm is reduced by 72% or 88%, with negligible memory overhead.

B. Parallel Implementation

The FRW algorithm is very suitable for parallelization, since the walks are independent to each other. Fig. 10 shows the flowchart of the parallel FRW algorithm on a multicore/multi-CPU platform. Multiple threads are allocated to execute the random walk procedure (steps 5–9 in Algorithm 3; denoted by FRW core in Fig. 10). In order to reduce the expense of communication, a unique random number generator [25] is kept for each thread. The lock operation only happens when updating the value of capacitance and checking the program termination criteria with total number of walks or estimated error. This check is performed for every m walks ($m = 1000$). The work of loading GFTs, WVTs, and GFT_ISs is also parallelized without difficulty. The pthread APIs are used in our implementation.

For large structure including thousands of conductor blocks, serially building the octree (step 2 in Fig. 10) may cost a lot of time. To reduce this expense, the construction of octree can also be parallelized. The basic idea is that, in the insertion operation every conductor is independent to each other and the operation on every tree node is also independent to each other.

VI. NUMERICAL RESULTS

We have developed a C++ program RWCcap with the proposed techniques. To evaluate the efficiency of different techniques, four subversions of RWCcap are defined.

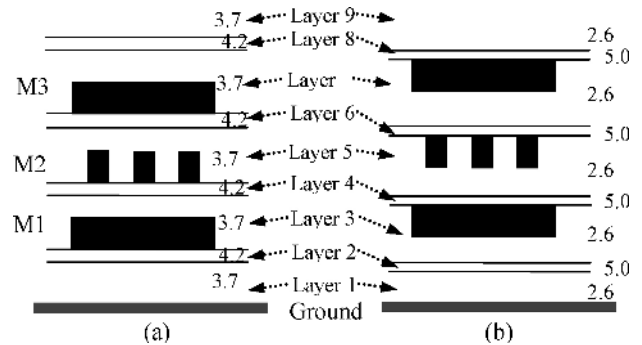


Fig. 11. Cross section of process technology used in the test cases. (a) 180-nm technology, where layers 2, 4, 6, 8 are thin dielectrics with thickness of 200-nm, and the thickness of layers 3, 5, 7 is 1180-nm. (b) 45-nm technology, where layers 2, 4, 6, 8 are thin dielectrics with thickness of 40-nm, and the thickness of layers 3, 5, 7 is 220-nm.

- 1) RWCcap(O): The basic multidielectric FRW (Algorithm 1).
- 2) RWCcap(M): The FRW using the multidielectric GFTs and WVTs (Algorithm 2).
- 3) RWCcap(IS): The FRW using the IS technique proposed in Section IV-B.
- 4) RWCcap(R): The FRW using all variance reduction techniques (Algorithm 3).

The termination criteria of RWCcap is that the $1 - \sigma$ error of self-capacitance becomes smaller than 1% of the mean value. This means that the relative error is within 3% (in a 99.7% confidence level). The FDM-based method to generate the multidielectric GFTs and WVTs is implemented in MATLAB, using the functions for sparse matrix. Unless otherwise stated, the value of N (segment number along cube edge) is set to 31.

As revealed in [14], the placement of Gaussian surface has an impact on the performance of FRW algorithm. In that work, the Gaussian surface was constructed to enclose the master conductor in such a way that points on it are equi-distant from the conductor and from other adjacent conductors. However, this strategy is not valid for the case with rare conductors, where along some direction the equidistant position becomes very far away from the master conductor. This worsens the convergence behavior of the FRW procedure. So, we adopt a different strategy which first finds the six axis-direction equidistant positions from the master, and then use their minimum to construct the Gaussian surface enclosing the master conductor.

The experiments are carried out on a Linux server with Intel Xeon E5620 8-core CPU of 2.40 GHz. The computational time listed does not include that for loading the multidielectric GFTs, WVTs, and GFT_ISs, because this can be performed only once for a process technology. We store these data with binary format files, and the time for loading them is less than 0.2 s for each tested process technology.

A. Test Cases

The typical 180- and 45-nm technology described in [26] are considered. The cross sections of the test structures with three metal layers are shown in Fig. 11, where the relative permittivity of each dielectric layer is labeled. For the 180-nm technology, the width and height of each wire are 300 and 530-nm. While in the 45-nm technology, the width and height are

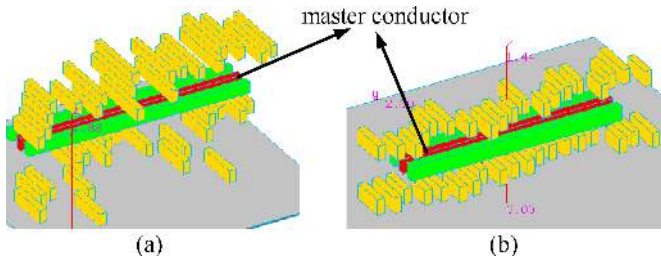


Fig. 12. 3-D views of the test cases. (a) Case 2. (b) Case 5.

TABLE I

DESCRIPTION OF SIX TEST CASES

	Technology	Description
Case 1		Three parallel wires. Wire length is 10 000-nm, and the spacing is 300-nm to the left and 500-nm to the right.
Case 2	180-nm technology	41 wires in three metal layers. Wire width is 300-nm.
Case 3		5 × 5 bus structure. Wire length is 3000-nm, and the spacing is 300-nm.
Case 4	45-nm technology	Three parallel wires. Wire length is 2000-nm, and the wire spacing is 70-nm.
Case 5		41 wires in 3 metal layers. Wire width is 70-nm.
Case 6		5 × 5 bus structure. Wire length is 700-nm, and the spacing is 70-nm.

70 and 140-nm, respectively. The first test structure includes three parallel wires in M2 layer. For the second structure, two sets of 19 wires are added to M1 and M3 layers, respectively, with random width and spacing (see Fig. 12). The third structure is a 5 × 5 bus structure distributed in M2 and M3 layers. Combining with the two process technologies, we obtain six test cases as listed in Table I. For each case, the conductor in the middle is set as the master conductor, and the self- and coupling capacitances of the master conductor are extracted.

We first validate the proposed techniques with experiments of serial computing. Then, the efficiency of the parallel RW-Cap is demonstrated.

B. Validating the Multidielectric FRW Algorithm

The six test cases are extracted with RWCap(O) and RW-Cap(M), whose results are compared in Table II. Besides the self-capacitance C_{self} , we also list a major coupling capacitance as C_{c1} which is between the master and its right-side neighbor wire. $\text{Std}(C_{c1})$ in Table II is the ratio of this coupling capacitance's Std to its mean value. Note that the Std of C_{self} equals 1% of its mean value, which is the termination criterion.

The capacitance values in Table II validate the accuracy of RWCap(M), i.e. Algorithm 2. For several cases, the Std of the coupling capacitance is even reduced with the RWCap(M). The values of $\text{Std}(C_{c1})$ less than 2% also demonstrate that the FRW algorithm is able to calculate the major coupling capacitance accurately. The data of CPU time suggest that the technique of utilizing the multidielectric GFTs and WVTs speeds up the original FRW algorithm by several tens times to more than one hundred times. And, the memory overhead is only 12 MB, which is negligible. The benefit brought by the multidielectric WVTs is actually larger than that of GFTs, because the former avoids the first transition domain with very small size, and therefore largely reduces the number of walks for convergence. This is more prominent for the 45-nm

TABLE II

COMPUTATIONAL RESULTS OF RWCap(O) AND RWCap(M)
(CAPACITANCE IN UNIT OF 10^{-16} F, MEMORY IN UNIT OF MB)

Case	RWCap	C_{self}	C_{c1}	$\text{Std}(C_{c1})$ (%)	No. of walks	Memory	Time (s)	Speedup
1	O	18.90	6.42	1.6	6098k	1.6	1468.55	–
	M	18.61	6.32	1.5	206k	13.4	17.46	84
2	O	19.50	5.22	1.9	5512k	1.7	368.88	–
	M	19.40	5.33	1.7	179k	13.6	5.13	72
3	O	7.25	2.61	2.0	4180k	1.6	533.88	–
	M	7.34	2.64	1.7	132k	13.5	6.13	87
4	O	3.60	1.62	1.5	8075k	1.5	2554.09	–
	M	3.64	1.61	1.5	160k	13.4	15.93	160
5	O	3.95	1.36	1.9	6195k	1.6	516.48	–
	M	3.94	1.35	1.8	122k	13.5	4.12	125
6	O	1.43	0.525	1.7	7417k	1.6	1261.99	–
	M	1.42	0.504	1.7	150k	13.5	9.59	132

technology, because there is a 40-nm thickness thin dielectric layer just above the master conductor. We find out with the experiments that the speedup brought by the multidielectric GFTs solely is from $2 \times$ to $3 \times$.

We have also solved case 4 with FastCap [1] and CAPEM [13]. CAPEM is a binary-code program, whose technical details are not published yet. Fastcap's result is 3.55×10^{-16} F, with time of 34.7 s and 1.8 GB memory. It validates the accuracy of RWCap (with discrepancy of 2.5%), and shows that FastCap is inferior even for this smaller case due to a lot of panels caused by the discretization of dielectric interfaces. With 100 000 walks, CAPEM's result is 3.62 ± 0.18 (5%), while consuming 773 s. To compare fairly, we also run RW-Cap(M) for 100k walks, which produces C_{self} of 3.49 ± 0.045 (1.3%) after 9.5 s. This means RWCap(M) is $81 \times$ faster than CAPEM and has better convergence rate.

CAPEM also has the function of precharacterizing the transition cube with multilayered dielectrics. While setting different values of N (segment number along cube edge), we have compared the computational times of generating the GFTs and WVTs for a dielectric configuration of cube with CAPEM and our program in Fig. 13. The figure demonstrates that the precharacterization procedure with the presented techniques is faster than that of CAPEM. For one tested process technology, there are two permittivity pairs and the precharacterization time for RWCap(M) is about 524 s. In the precharacterization, generating GFTs and generating WVTs cost about 1/4 and 3/4 of the total time, respectively. Also note that while performing the precharacterization for RWCap(IS) and RWCap(R), the computational time will not increase, as we have discussed at the end of Section IV-B.

Comparing the CPU times in Table II, it is revealed that the time is not scaled with the complexity of structure. This is the distinct property of the FRW algorithm, as compared with the conventional algorithms. For example, Case 4 has the most complex structure including 41 wires [see Fig. 12(b)], but it costs the least CPU time. The reason is that, for a case with dense environment conductors the walk will terminate quickly.

C. Validating the Variance Reduction Techniques

To demonstrate the efficiency of the proposed variance reduction techniques, RWCap(M) is compared with RWCap(IS)

TABLE III
COMPUTATIONAL RESULTS OF RWCAP(R) AND ITS COMPARISON WITH OTHER FRW-BASED ALGORITHMS

Case	RWCAP(M)		RWCAP(IS)		RWCAP(R)							Variant FRW		Variant FRW(R)	
	No. of Walks	Time (s)	No. of Walks	Time (s)	No. of Walks	$C_{self}(10^{-16} F)$	$C_{c1}(10^{-16} F)$	Std (C_{c1}) (%)	Time (s)	Speedup ^a	Speedup ^b	No. of Walks	Time (s)	No. of Walks	Time (s)
1	206k	17.46	151k	11.80	61k	18.40	6.41	1.9	4.82	3.6	2.7	348k	29.07	145k	12.83
2	179k	5.13	131k	3.72	56k	19.51	5.47	2.1	1.51	3.4	2.4	307k	8.46	128k	3.65
3	132k	6.13	96k	4.22	46k	7.28	2.69	1.8	2.10	2.9	2.4	246k	11.54	110k	5.04
4	160k	15.93	114k	10.75	51k	3.65	1.63	1.6	4.93	3.3	2.8	292k	30.01	130k	13.95
5	122k	4.12	90k	2.89	44k	3.91	1.36	1.9	1.47	2.8	2.4	232k	7.77	100k	3.52
6	150k	9.59	104k	6.13	51k	1.44	0.517	1.9	3.13	3.1	2.6	267k	16.65	131k	8.21

^aThe speedup ratio to RWCAP(M).

^bThe speedup ratio to the variance reduction accelerated variant FRW algorithm in [14], i.e. Variant FRW(R).

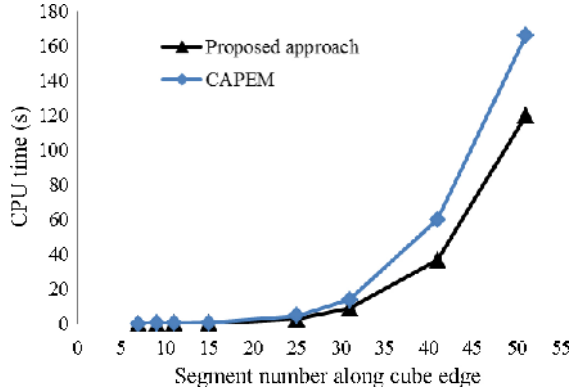


Fig. 13. Computing time for generating the multidielectric GFTs and WVTs with various segment numbers along the cube's edge. The results are obtained on a Linux server with Intel Xeon 3.0GHz CPU.

and RWCAP(R). The results are listed in Table III. From this table we can see that the importance sampling and SS technique largely reduce the number of walks for the same accuracy criterion. For the six cases, they bring the speedup ratios from $2.8\times$ to $3.6\times$ (**$3.2\times$** on average). We have also implemented the variant FRW formula (30), and the corresponding variance reduction technique in [14] with our program, which is referred to as Variant FRW and Variant FRW(R) in Table III. Compared with them, our technique also shows more than $2\times$ speedup, and RWCAP(IS) using the technique of importance sampling is comparable or superior to the Variant FRW(R). Note that the SS in [14] only considers the distribution of single-dielectric weight value, which makes its speedup not prominent for the multidielectric cases.

The capacitance values in Table III partially validate the accuracy of RWCAP(R). For the complete validation, we run RWCAP(R) for 3000 times, and then plot the distribution of the extracted C_{self} . Fig. 14 shows this distribution for case 1, accompanied by the distribution obtained with RWCAP(M). Both plots approximate to the normal distribution, and the calculated Std ($<1\%$ of mean value) suggests that the accuracy of extracted C_{self} is not degraded with the variance reduction. Besides, we compare the $Std(C_{c1})$ in Tables II and III, and find out that RWCAP(R) also preserves the accuracy of coupling capacitance.

D. Comparing with the Fast Boundary Element Method

Experiments are carried out to compare RWCAP(R) with a fast boundary method utilizing the quasi-multiple medium (QMM) technique [2]. The QMM accelerated BEM (called QBEM)¹ is suitable for multidielectric cases, and has exhibited

¹The program QBEM is shared on the website of the first author: <http://learn.tsinghua.edu.cn:8080/2003990088/software.htm>

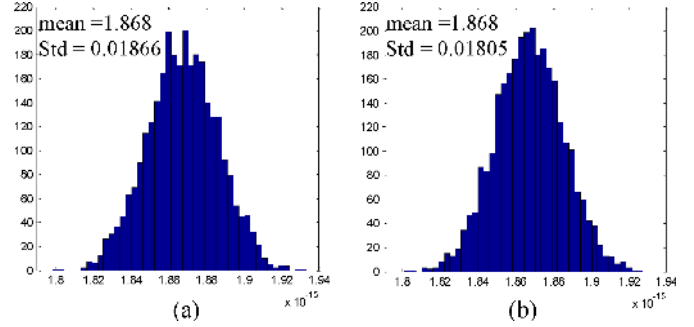


Fig. 14. Distribution of the C_{self} extracted with (a) RWCAP(M), and (b) RWCAP(R), for 3000 runs.

TABLE IV
COMPARISON OF RWCAP(R) AND QBEM FOR THE 45-NM-TECHNOLOGY CASES (CAPACITANCE IN UNIT OF $10^{-16} F$, TIME IN UNIT OF SECOND)

Case	QBEM				RWCAP(R)					
	No. of panels	C_{self}	C_{c1}	Time	C_{self}	Dis. (%)	C_{c1}	Dis. (%)	Time	Speedup
4	9334	3.63	1.61	2.79	3.65	0.6	1.67	3.7	4.93	0.57
5	12211	3.81	1.32	2.44	3.95	3.7	1.36	3.0	1.47	1.7
6	13883	1.38	0.504	4.09	1.44	4.3	0.517	2.6	3.13	1.3
7 ^a	54980	50.4	20.7	180	52.1	3.4	18.5	-10.6	4.96	36

^aA 3-layer structure with totally 403 wires, under the 45-nm technology.

TABLE V
COMPUTATIONAL TIME OF THE PARALLEL RWCAP(R) FOR CASE 1

	No. of Thread	Time (s)	Speedup
Walk-number criterion (100k)	1	8.05	-
	2	4.06	1.98
	4	2.03	3.97
	8	1.16	6.94
Accuracy criterion (1%)	1	4.79	-
	8	0.70	6.82

about $10\times$ speedup over the FastCap. The 45-nm-technology cases are tested, and the computational results are listed in Table IV. To compare their performance for large-scale structure, the 7th test case is constructed, which includes 403 wires in the three metal layers (with similar structure to case 5, but has 200 crossing wires in each neighbor layer).

For the extraction with QBEM, the Neumann boundary is needed to surround the extracted structure, while the FRW algorithm assumes the Dirichlet boundary far from the structure. So, the self-capacitance in the problem for FRW is always larger than that in the problem for QBEM. Based on this analysis, we find out that the discrepancy of QBEM and RWCAP(R) on C_{self} (see Table IV) is fairly small. As for the coupling capacitance, the discrepancy of the both solvers on C_{c1} is no more than 10%. From the table, we also see that QBEM is faster than RWCAP(R) for the smallest case, but

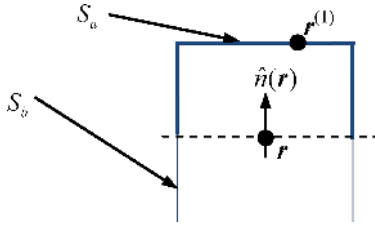


Fig. 15. 2-D view illustrating the unit-size cube and its two half parts of surface.

this advantage is lost for more complex cases. For the largest 403-wire case, RWCap(R) becomes $36\times$ faster than QBEM.

E. Validating the Efficiency of Parallel Computing

On the 8-core machine, experiments are carried out to validate the parallel implementation of RWCap(R). We first set 100 000 walks to be the termination criterion of RWCap, and run it with different number of threads. The computing time for Case 1 is listed in Table V, which shows the high efficiency of the parallelization. For all cases, nearly $7\times$ speedup can be achieved on the 8-core machine. While setting the standard error of 1% as the termination criterion, the parallel speedup of RWCap decreases a little because more walks than required are assigned to balance the workload. However, it still achieves over $6\times$ speedup on the 8-core machine, as shown in Table V. The experiment is also performed for the 7th test case with 403 wires, whose result show the parallel RWCap costs about 0.81 s with the 1% termination criterion. Comparing with the serial-computing time in Table IV, this means a $6.1\times$ speedup.

VII. CONCLUSION

By numerically precharacterizing the surface Green's function and the weight value for cubic transition domain with two dielectric layers, an efficient FRW algorithm was proposed for multidielectric capacitance extraction. The algorithm was further accelerated with a new variance reduction scheme based on the importance sampling and the stratified sampling. Along with a space management technique to reduce the computing time for each hop in the FRW algorithm and the parallel computing technique, an efficient FRW solver called RWCap was developed.

Numerical results on test structures with nine dielectric layers under the 180 and 45-nm process technologies validated the efficiency of the proposed techniques. While comparing with other existing FRW algorithms, RWCap showed several tens times speedup over CAPEM [13], and more than $2\times$ speedup over the technique proposed in [14]. The comparisons with FastCap and QBEM showed that the proposed FRW algorithm has comparable computing speed as the fast BEM solvers for small cases, but for large case was several tens times more efficient in CPU time and memory usage.

It should be pointed out that the proposed variance reduction technique is also applicable to the extraction problem with a single dielectric. Numerical experiments have been carried out on the single-dielectric cases, which validate the similar efficiency as on the multidielectric cases, but are omitted due to the limit of space. In the future, the RWCap will be extended for the chip-level capacitance extraction tasks. The RWCap program has been shared on the website of the first author.

APPENDIX

In the Appendix, we prove the properties about the weight value and its integral on the transition cube to validate the variance reduction scheme, as proposed in Section IV-C.

Lemma 1:

$$\begin{cases} \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) > 0, & \text{if } \mathbf{r}^{(1)} \in S_a \\ \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) < 0, & \text{if } \mathbf{r}^{(1)} \in S_b. \end{cases} \quad (\text{A.1})$$

Here, $P(\mathbf{r}, \mathbf{r}^{(1)})$ is the surface Green's function of the unit-size cube centered at point \mathbf{r} . $\hat{\mathbf{n}}(\mathbf{r})$ is a normal direction. S_a is the half of the cube's surface complying with $\hat{\mathbf{n}}(\mathbf{r})$, while S_b is the other half complying with the negative direction of $\hat{\mathbf{n}}(\mathbf{r})$.

Proof: Fig. 15 illustrates the meaning of related symbols. Since only the Manhattan geometry is considered in this paper, we assume $\hat{\mathbf{n}}(\mathbf{r})$ is along the z -axis direction without loss of generality. Then

$$\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) = \frac{\partial P(\mathbf{r}, \mathbf{r}^{(1)})}{\partial z} \quad (\text{A.2})$$

where \mathbf{r} has coordinates $[x, y, z]^T$. Assuming a small perturbation of $\Delta \mathbf{z} = [0, 0, \Delta z]^T$ along the $\hat{\mathbf{n}}(\mathbf{r})$ direction, we consider

$$\Delta P = P(\mathbf{r} + \Delta \mathbf{z}, \mathbf{r}^{(1)}) - P(\mathbf{r}, \mathbf{r}^{(1)}).$$

Because $P(\mathbf{r}, \mathbf{r}^{(1)})$ means the correlation (or contribution) coefficient of the potential at point $\mathbf{r}^{(1)}$ to the potential at point \mathbf{r} , a movement of \mathbf{r} toward $\mathbf{r}^{(1)}$ makes this correlation stronger. Thus, ΔP is positive for $\mathbf{r}^{(1)}$ on S_a , which suggests the limit of $\Delta P/\Delta z$, i.e. $\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) > 0$

² On the contrary, for $\mathbf{r}^{(1)}$ on S_b , $\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) < 0$.

Considering the formula of weight value (5), (17), and that the surface Green's function $P(\mathbf{r}, \mathbf{r}^{(1)})$ is always positive, we conclude Theorem 1 about the weight value. ■

Theorem 1: The weight value in the standard FRW algorithm is negative if $\mathbf{r}^{(1)}$ falls onto the surface of half cube outside G_j , and is positive if $\mathbf{r}^{(1)}$ falls onto the surface of half cube inside G_j . Here G_j is the Gaussian surface surrounding the master conductor j .

Lemma 2:

$$\oint_S \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) d\mathbf{r}^{(1)} = 0 \quad (\text{A.3})$$

where $P(\mathbf{r}, \mathbf{r}^{(1)})$ is the surface Green's function of the unit-size cube centered at point \mathbf{r} , S is the surface of cube, and $\hat{\mathbf{n}}(\mathbf{r})$ is a normal direction.

Proof: The surface Green's function has the following property:

$$\mathcal{F}(\mathbf{r}) = \int_S P(\mathbf{r}, \mathbf{r}^{(1)}) d\mathbf{r}^{(1)} = 1 \quad (\text{A.4})$$

for any point \mathbf{r} in the space surrounded by S . This can be derived from (1), if we consider a conductor shell whose surface potential $\phi(\mathbf{r}^{(1)}) = 1$. Then, the potential at anywhere in the shell should be 1. This proves (A.4). Because ■

$$\oint_S \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) d\mathbf{r}^{(1)} = \nabla_r \mathcal{F}(\mathbf{r}) \bullet \hat{\mathbf{n}}(\mathbf{r}) \quad (\text{A.5})$$

which is the normal partial derivative of $\mathcal{F}(\mathbf{r})$, it must be zero based on the fact that $\mathcal{F}(\mathbf{r})$ is a constant. This proves (A.3).

²Because the sampling probability function for $\mathbf{r}^{(1)}|_S \propto |\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r})|/K$ (34), it is impossible that $\mathbf{r}^{(1)}$ has a value such that $\nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) = 0$.

With (A.3), and the definition of K_a and K_b in Section IV-C

$$\oint_S \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) d\mathbf{r}^{(1)} = K_a + K_b = 0. \quad (\text{A.6})$$

(A.1) suggests that $K_a > 0$ and $K_b < 0$. So,

$$\begin{aligned} K &= \oint_S \left| \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) \right| d\mathbf{r}^{(1)} \\ &= \oint_{S_a} \left| \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) \right| d\mathbf{r}^{(1)} + \oint_{S_b} \left| \nabla_r P(\mathbf{r}, \mathbf{r}^{(1)}) \bullet \hat{\mathbf{n}}(\mathbf{r}) \right| d\mathbf{r}^{(1)} \\ &= K_a - K_b \\ &= 2K_a. \end{aligned} \quad (\text{A.7})$$

This proves Theorem 2.

Theorem 2: K_a equals the half of K , where K is defined by (32) and K_a defined by (39), in Section IV.

REFERENCES

- [1] K. Nabors and J. White, "FastCap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 10, no. 11, pp. 1447–1459, Nov. 1991.
- [2] W. Shi, J. Liu, N. Kakani, and T. Yu, "A fast hierarchical algorithm for three-dimensional capacitance extraction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 3, pp. 330–336, Mar. 2002.
- [3] W. Yu and Z. Wang, "Enhanced QMM-BEM solver for three-dimensional multiple-dielectric capacitance extraction within the finite domain," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 2, pp. 560–566, Feb. 2004.
- [4] S. Yan, V. Sarin, and W. Shi, "Sparse transformations and preconditioners for 3-D capacitance extraction," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 9, pp. 1420–1426, Sep. 2005.
- [5] W. Yu, M. Zhang, and Z. Wang, "Efficient 3-D extraction of interconnect capacitance considering floating metal-fills with boundary element method," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 25, no. 1, pp. 12–18, Jan. 2006.
- [6] W. Chai, D. Jiao, and C.-K. Koh, "A direct integral-equation solver of linear complexity for large-scale 3-D capacitance and impedance extraction," in *Proc. DAC*, Jul. 2009, pp. 752–757.
- [7] N. P. van der Meijjs and A. J. van Genderen, "An efficient finite element method for submicron IC capacitance extraction," in *Proc. DAC*, Jun. 1989, pp. 678–681.
- [8] G. Chen, H. Zhu, T. Cui, Z. Chen, X. Zeng, and W. Cai, "ParAFEMCap: A parallel adaptive finite-element method for 3-D VLSI interconnect capacitance extraction," *IEEE Trans. Microw. Theory Tech.*, vol. 60, no. 2, pp. 218–231, Feb. 2012.
- [9] Y. Le Coz and R. B. Iverson, "A stochastic algorithm for high speed capacitance extraction in integrated circuits," *Solid State Electron.*, vol. 35, no. 7, pp. 1005–1012, Jul. 1992.
- [10] Y. Le Coz, H. J. Greub, and R. B. Iverson, "Performance of random walk capacitance extractors for IC interconnects: A numerical study," *Solid-State Electron.*, vol. 42, no. 4, pp. 581–588, Apr. 1998.
- [11] R. B. Iverson and Y. Le Coz, "A floating random-walk algorithm for extracting electrical capacitance," *Math. Comput. Simul.*, vol. 55, pp. 59–66, 2001.
- [12] M. Kamon and R. Iverson, "High-accuracy parasitic extraction," in *EDA for IC Implementation, Circuit Design, and Process Technology*, L. Lavagno, L. Scheffer, and G. Martin, Eds. Boca Raton, FL: CRC Press/Taylor and Francis, 2006.
- [13] M. P. Desai. (1998). *The Capacitance Extraction Tool* [Online]. Available: <http://www.ee.iitb.ac.in/~microel/download>
- [14] S. H. Batterywala and M. P. Desai, "Variance reduction in Monte Carlo capacitance extraction," in *Proc. 18th Int. Conf. VLSI Design*, Jan. 2005, pp. 85–90.
- [15] S. H. Batterywala, R. Ananthakrishna, Y. Luo, and A. Gyure, "A statistical method for fast and accurate capacitance extraction in the presence of floating dummy fills," in *Proc. 19th Int. Conf. VLSI Design*, Jan. 2006, pp. 129–134.
- [16] T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "A capacitance solver for incremental variation-aware extraction," in *Proc. ICCAD*, Nov. 2008, pp. 662–669.
- [17] T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "A hierarchical floating random walk algorithm for fabric-aware 3-D capacitance extraction," in *Proc. ICCAD*, Nov. 2009, pp. 752–758.
- [18] A. Brambilla and P. Maffezzoni, "A statistical algorithm for 3-D capacitance extraction," *IEEE Microw. Guided Wave Lett.*, vol. 10, no. 8, pp. 304–306, Aug. 2000.
- [19] G. Hu, W. Yu, H. Zhuang, and S. Zeng, "Efficient floating random walk algorithm for interconnect capacitance extraction considering multiple dielectrics," in *Proc. IEEE Int. Conf. ASIC*, Oct. 2011, pp. 896–899.
- [20] H. Zhuang, W. Yu, G. Hu, Z. Liu, and Z. Ye, "Fast floating random walk algorithm for multi-dielectric capacitance extraction with numerical characterization of Green's functions," in *Proc. ASP-DAC*, Jan. 2012, pp. 377–382.
- [21] G. M. Royer, "A Monte Carlo procedure for potential theory problems," *IEEE Trans. Microw. Theory Tech.*, vol. 19, no. 10, pp. 813–818, Oct. 1971.
- [22] J. N. Jere and Y. Le Coz, "An improved floating-random-walk algorithm for solving the multi-dielectric Dirichlet problem," *IEEE Trans. Microw. Theory Tech.*, vol. 41, no. 2, pp. 325–329, Feb. 1993.
- [23] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.
- [24] H. Samet, *Applications of Spatial Data Structure*. Reading, MA: Addison-Wesley, 1990.
- [25] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Modeling Comput. Simul.*, vol. 8, no. 1, pp. 3–30, Jan. 1998.
- [26] D. Deschacht, S. de Rivaz, A. Farcy, T. Lacrevez, and B. Flechet, "Keep on shrinking interconnect size: Is it still the best solution?" in *Proc. IEMT*, Nov. 2010, pp. 1–4.
- [27] N. Bansal, "Randomized algorithms for capacitance estimation," Indian Instit. Technol. Bombay, Mumbai, India, Tech. Rep., Apr. 1999.



Wenjian Yu (S'01–M'04–SM'10) received the B.S. and Ph.D. degrees in computer science from Tsinghua University, Beijing, China, in 1999 and 2003, respectively.

In 2003, he joined Tsinghua University, where he is currently an Associate Professor with the Department of Computer Science and Technology. He has visited the Computer Science and Engineering Department, University of California, San Diego (UCSD), several times, from September 2005 to January 2008. His current research interests include

parasitic extraction, modeling and simulation of interconnects, and a broad range of numerical methods.

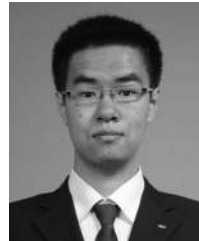
Dr. Yu was the recipient of the distinguished Ph.D. Award from Tsinghua University in 2003. He was a Technical Program Committee Member of the ACM/IEEE Asia South-Pacific Design Automation Conference in 2005, 2007, 2008, and 2013.



Hao Zhuang received the M.S. degree in microelectronics and solid-state electronics from Peking University, Beijing, China, in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of California, San Diego.

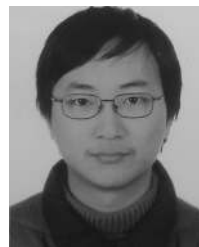
From 2011 to 2012, he was a Visiting Student with the Department of Computer Science and Technology, Tsinghua University, Beijing. His current research interests include circuit simulation, interconnect parasitic extraction, and design automation for microelectronic circuits.

Mr. Zhuang was the recipient of the Powell Fellowship from the Jacobs School of Engineering, University of California, San Diego, in 2012.



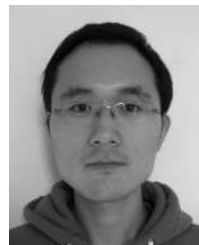
Chao Zhang received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 2012, where he is currently pursuing the M.S. degree in computer science with the Department of Computer Science and Technology.

His current research interests include interconnect capacitance extraction.



Gang Hu received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 2011, and the M.S. degree in computer science from Columbia University, New York, in 2012, where he is currently pursuing the Ph.D. degree in computer science.

His current research interests include making reliable and secure software systems, including software testing and verifying.



Zhi Liu received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 2010, and the M.S. degree in computer engineering from Télécom ParisTech, Paris, France, in 2012.

He is currently a Software Engineer with Murex S.A.S., New York.