A Parallel and Incremental Extraction of Variational Capacitance with Stochastic Geometric Moments

Fang Gong, *Student Member, IEEE,* Hao Yu, *Member, IEEE,* Lingli Wang, *Member, IEEE* and Lei He, *Senior Member, IEEE,*

***Abstract*—This paper presents a parallel and incremental solver for stochastic capacitance extraction. The random geometrical variation is described by stochastic geometrical moments, which lead to a densely augmented system equation. To efficiently extract the capacitance and solve the system equation, a parallel fast-multipole-method (FMM) is developed in the framework of stochastic geometrical moments. This can efficiently estimate the stochastic potential interaction and its matrix-vector product (MVP) with charge. Moreover, a generalized minimal residual (GMRES) method with incremental update is developed to calculate both the nominal value and the variance. Our overall extraction flow is called *piCAP*. A number of experiments show that piCAP efficiently handles a large-scale on-chip capacitance extraction with variations. Specifically, a parallel MVP in piCAP is up to 3X faster than a serial MVP, and an incremental GMRES in piCAP is up to 15X faster than non-incremental GMRES methods.**

***Index Terms*—Process variation, Capacitance extraction, Fast multipole method.**

I. Introduction

AS IC designs are approaching processes below 45nm, there exist large uncertainties from chemical mechanical polishing (CMP), etching, and lithography [1], [2], [3], [4], [5], [6], [7]. As a result, the fabricated interconnect and dielectric can show a significant difference from the nominal shape. The value of an extracted capacitance thereby can differ from the nominal value by a large margin, which may further lead to significant variability for the timing analysis. For example, as shown in [1], the variation of interconnect can cause as much as 25% variation in the clock skew. Therefore, accurately extracting the capacitance with consideration of the stochastic process variation becomes a necessity.

To avoid discretizing the entire space, the boundary element method (BEM) is used to evaluate capacitance by discretizing the surface into panels on the boundary of the conductor and the dielectric [8], [9], [10], [11]. Though this results in a discretized system with a small dimension, the discretized system under BEM is dense. FastCap [8] solves such a dense system by a generalized minimal residual (GMRES) method. Instead of performing the expensive LU decomposition, the GMRES iteratively reaches the solution with the use of the matrix-vector multiplication. The computational cost of the matrix-vector-product (MVP) can be reduced by either a fast-multipole-method (FMM) [8], a low-rank approximation [9], and a hierarchical-tree decomposition [10]. As a result, the complexity of the fast full-chip extractions generally comes from two parts: the evaluation of MVP and the preconditioned GMRES iteration.

A few recent works [3], [4], [5] discuss interconnect extraction considering process variation. The variation is represented by the stochastic orthogonal polynomial (SOP) [12], [13] when calculating a variational capacitance. Since the interconnect length and cross-area are at different scales, the variational capacitance extraction is quite different between the on-chip [4], [5] and the off-chip [3]. The on-chip interconnect variation from the geometrical parameters, such as width length of one panel and distance between two panels, is more dominant [4], [5] than the rough surface effect seen from the off-chip package trace. However, it is unknown how to leverage the stochastic process variation into the MVP by FMM [3], [4], [5]. Similar to deal with the stochastic analog mismatch for transistors [14], a cost-efficient full-chip extraction needs to explore an explicit relation between the stochastic variation and the geometrical parameter such that the electrical property can show an explicit dependence on geometrical parameters. Moreover, the expansion by SOP with different collocation schemes [12], [13], [4], [5] always results in an augmented and dense system equation. This significantly increases the complexity when dealing with a large-scale problem. The according GMRES thereby needs to be designed in an incremental fashion to consider the update from the process variation. As a result, a scalable extraction algorithm similar to [8], [9], [10] is required to consider the process variation with the new MVP and GMRES developed accordingly as well.

To address the aforementioned challenges, this paper contributes as follows. First, to reveal an explicit dependence on geometrical parameters, the potential interaction is represented by a number of geometrical moments. As such, the process variation can be further included by expanding the geometrical moments with use of stochastic orthogonal polynomials, called *stochastic geometrical moments* in this paper. Next, with the use of the stochastic geometric moment, the process variation can be incorporated into a modified FMM algorithm that evaluates the MVP in parallel. Finally, an incremental GMRES method is introduced to update the preconditioner with different variations. Such a parallel and incremental

F. Gong and L. He are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (E-mail: fang08@ee.ucla.edu; lhe@ee.ucla.edu).

H. Yu is with Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Dr. Hao Yu is sponsored by the NRF2010NRF-POC001-001 and MOE Tier-1 RG 26/10 (M52040146) from Singapore. (E-mail: haoyu@ntu.edu.sg)

L. Wang is with State Key Laboratory of Application Specific Integrated Circuits and Systems, Fudan University, Shanghai, P.R. China (E-mail: llwang@fudan.edu.cn).

Manuscript received July 15, 2010; revised January 31, 2011.

full chip capacitance extraction considering the stochastic variation is called *piCAP*. Parallel and incremental analysis are the two effective techniques in reducing computational cost. Experiments show that our method with stochastic polynomial expansion is hundreds of times faster than the Monte-Carlo based method while maintaining a similar accuracy. Moreover, the parallel MVP in our method is up to 3X faster than the serial method, and the incremental GMRES in our method is up to 15X faster than non-incremental GMRES methods.

The rest of the paper is organized in the following manner. We first review the background of the capacitance extraction and fast multipole method (FMM) in Section II. We introduce the concept of the stochastic geometrical moment in Section III, and illustrate a parallel FMM method based on the stochastic geometrical moment in Section IV. We further propose a novel incremental GMRES method in Section V and present experiment results in Section VI. Finally, the paper is concluded in Section VII.

II. Background

A. Boundary Element Method (BEM)

The boundary element method (BEM), used in most fast capacitance extractions [8], [9], [10], starts with an integral equation

$$\phi(r) = \int_{r' \in a'} \frac{\rho(r')}{4\pi\epsilon_0 |r - r'|} da', \tag{1}$$

where $\phi(r)$ is the potential at the observer metal, $\rho(r')$ is the surface-charge density at the source metal, da' is an incremental area at the surface of the source metal, and the source r' is on da'.

By discretizing the metal surface into N panels sufficiently such that the charge-density is uniform at each panel, a linear system equation can be obtained by the *point-collocation* [8]:

$$Pq = b, \tag{2}$$

where P is an $N \times N$ matrix of potential coefficients (or potential interactions), q is an N vector of panel charges, and b is an N vector of panel potentials. By probing b iteratively with one volt at each panel in the form of [0,...,1,...,0], the solved vector q is one column of the capacitance matrix.

Note that each entry P_{ij} in the potential matrix P represents the potential observed at the *observer panel* a_j due to the charge at the *source panel* a_i:

$$P_{ij} = \frac{1}{a_i} \int_{r_i \in a_i} \frac{1}{4\pi\epsilon_0 |r_i - r_j|} da_i. \tag{3}$$

When panel i and panel j are well-separated by definition, P_{ij} can be well approximated by $\frac{1}{4\pi\epsilon_0 |r_i - r_j|}$ [8], [9], [10], [4], [5].

The resulting potential coefficient matrix P is usually dense in the BEM method. As such, directly solving (2) would be computationally expensive. FastCap [8] applies an iterative GMRES method [15] to solve (2). Instead of performing an expensive LU decomposition of the dense P, GMRES first forms a preconditioner W such that $W^{-1} \cdot P$ has a smaller condition number than P, which can accelerate the convergence of iterative solvers [16]. Take the left preconditioning as an example:

$$(W^{-1} \cdot P)q = W^{-1} \cdot b.$$

Then, using either multipole-expansion [8], low-rank approximation [9] or the hierarchical-tree method [10] to efficiently evaluate the matrix-vector-product (MVP) for $(W^{-1} \cdot P)q_i$ (q_i is the solution for i-th iteration), the GMRES method minimizes below residue error iteratively till converged.

$$min: \ ||W^{-1} \cdot b - (W^{-1} \cdot P)q_i||$$

Clearly, GMRES requires a well-designed preconditioner and a fast matrix-vector-product (MVP). In fact, fast multipole method (FMM) is able to accelerate the evaluation of MVP with $O(N)$ time complexity where N is the number of variables. We will introduce FMM first as what follows.

B. Fast Multipole Method (FMM)

The fast multipole method was initially proposed to speed up the evaluation of long-ranged particle forces in the N-body problem [17], [18]. It can also be applied to the iterative solvers by accelerating calculation of matrix-vector-product [8]. Let's take the capacitance extraction problem as an example to introduce the operations in the FMM. In general, the FMM discretizes the conductor surface into panels and forms a cube with a finite height containing a number of panels. Then, it builds a hierarchical oct-tree of cubes and evaluates the potential interaction P at different levels.

Specifically, the FMM first assigns all panels to leaf cells/cubes, and computes the multipole expansions for all panels in each leaf cell. Then, FMM calculates the multipole expansion of each parent cell using the expansions of its children cells (called M2M operations in Upward Pass). Next, the local field expansions of the parent cells can be obtained by adding multipole expansions of well-separated parent cells at the same levels (called M2L operations). After that, FMM descends the tree structure to calculate the local field expansion of each panel based on the local expansion of its parent cell (called L2L in Downward Pass). All these operations are illustrated within Fig. (1).

In order to further speed up the evaluation of MVP, our stochastic extraction has a parallel evaluation Pq with variations, which is discussed in Section IV, and an incremental preconditioner, which is discussed in Section V. Both of these features depend on how an explicit dependence between the stochastic process variation and the geometric parameters can be found, which will be discussed in Section III.

III. Stochastic Geometrical Moment

With Fast Multipole Method, the complexity of MVP Pq evaluation can be reduced to $O(N)$ during the GMRES iteration. Since the spatial decomposition in FMM is geometrically dependent, it is helpful to express P using geometrical moments with an explicit geometry-dependence. As a result, this can lead to an efficient recursive update (M2M, M2L, L2L) of P on the oct-tree. The geometry-dependence is also one key

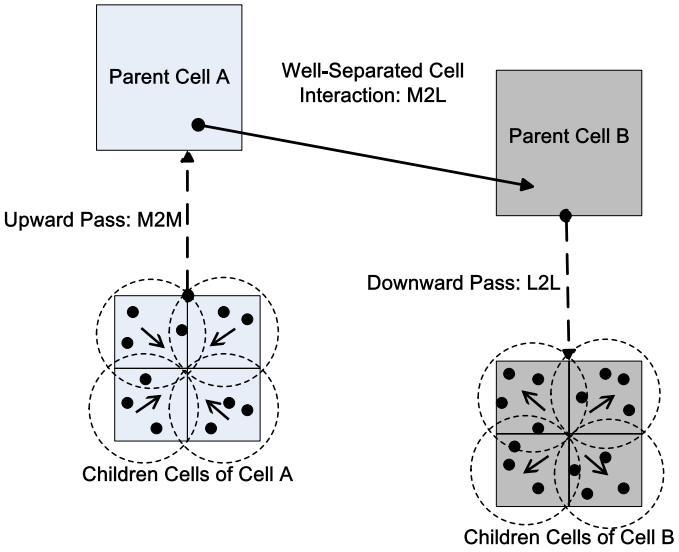


Fig. 1. Multipole Operations Within the FMM Algorithm

property to preserve in presence of the stochastic variation. In this section, we first derive the geometrical moment and then expand it by stochastic orthogonal polynomials to calculate the potential interaction with variations.

A. Geometrical Moment

In this paper, we focus on local random variations, or stochastic variations. Without loss of generality, two primary geometrical parameters with stochastic variation are considered for illustration purpose: panel-distance (d) and panel-width (h). Due to the local random variation, the width of the discretized panel, as well as the distance between panels, may show random deviations from the nominal value. With expansions in Cartesian coordinates, we can relate the potential interaction with the geometry parameter through *geometrical moments* (GMs) that can be extended to consider stochastic variations.

Let the center of an observer-cube be r_0 and the center of a source-cube be r_c. We assume that the distance between the i-th source-panel and r_c is a vector $\mathbf{r}$

$$\mathbf{r} = r_x\overrightarrow{x} + r_y\overrightarrow{y} + r_z\overrightarrow{z}$$

with $|\mathbf{r}| = r$, and the distance between r_0 and r_c is a vector $\mathbf{d}$

$$\mathbf{d} = d_x\overrightarrow{x} + d_y\overrightarrow{y} + d_z\overrightarrow{z}$$

with $|\mathbf{d}| = d$.

In Cartesian coordinates ($x-y-z$), when the observer is outside the source region ($d > r$), a *multipole expansion* (ME) [19], [20] can be defined as

$$\begin{aligned}\frac{1}{|\mathbf{r}-\mathbf{d}|} &= \sum_{p=0}\frac{(-1)^p}{p!}(\underbrace{\mathbf{r}\cdots\mathbf{r}}_{p})\underbrace{\times\cdots\times}_{p}(\underbrace{\nabla\cdots\nabla}_{p}\frac{1}{d}) \\ &= \sum_{p=0}M_p = \sum_{p=0}l_p(d)m_p(r)\end{aligned} \tag{4}$$

by expanding r around r_c, where

$$\begin{aligned}&l_0(d) = \frac{1}{d},\ m_0(r) = 1 \\ &l_1(d) = \frac{d_k}{d^3},\ m_1(r) = -r_k \\ &l_2(d) = \frac{3d_kd_l}{d^5},\ m_2(r) = \frac{1}{6}(3r_kr_l - \delta_{kl}r^2) \\ &\cdots \\ &l_p(d) = \underbrace{\nabla\cdots\nabla}_{p}\frac{1}{d},\ m_p(r) = \frac{(-1)^p}{p!}(\underbrace{\mathbf{r}\cdots\mathbf{r}}_{p}).\end{aligned} \tag{5}$$

Note that d_k, d_l are the coordinate components of vector $\mathbf{r}$ in Cartesian coordinates. The same is true for r_k and r_l. ∇ is the Laplace operator to take the spatial difference, δ_{kl} is the Kronecker delta function, and $(\mathbf{r}\cdots\mathbf{r})$ and $(\nabla\cdots\nabla\frac{1}{d})$ are rank-p tensors with $x^\alpha, y^\beta, z^\gamma$ ($\alpha+\beta+\gamma = p$) components.

Assume that there is a spatial shift at the source-cubic center r_c for example, change one child's center to its parent's center by $\mathbf{h}$ ($|h| = c\cdot h$), where c is a constant and h is the panel width. This leads to the following transformation for m_p in (5)

$$\begin{aligned}m_p' &= \underbrace{((\mathbf{r}+\mathbf{h})\cdots(\mathbf{r}+\mathbf{h}))}_{p} \\ &= m_p + \sum_{q=0}^{p}\frac{p!}{q!(p-q)!}(\underbrace{\mathbf{h}\cdots\mathbf{h}}_{j})m_{p-j}.\end{aligned} \tag{6}$$

Moreover, when the observer is inside the source region ($d < r$), a *local expansion* (LE) under Cartesian coordinates is simply achieved by exchanging d and h in (4)

$$\frac{1}{|\mathbf{r}-\mathbf{h}|} = \sum_{p=0}L_p = \sum_{p=0}m_p(h)l_p(r). \tag{7}$$

Also, when there is a spatial shift of the observer-cubic center r_0, the shift of moments $l_p(r)$ can be derived similarly to (6).

Clearly, both M_p, L_p and their spatial shifts show an explicit dependence on the panel-width h and panel-distance d. For this reason, we call M_p and L_p *geometrical moments*. As such, we can also express the potential coefficient

$$4\pi\epsilon_0\cdot P(h,d) \simeq \begin{cases}\sum_{p=0}M_p & if\ d > r \\ \sum_{p=0}L_p & otherwise\end{cases} \tag{8}$$

as a geometrical-dependence function $P(h,d)$ via geometrical moments.

Moreover, assuming that local random variations are described by two random variables. ξ_h for the panel-width h, and ξ_d for the panel-distance d, the stochastic forms of M_k and L_k become

$$\begin{aligned}\hat{M}_p(\xi_h,\xi_d) &= M_p(h_0+h_1\xi_h, d_0+d_1\xi_d) \\ \hat{L}_p(\xi_h,\xi_d) &= L_p(h_0+h_1\xi_h, d_0+d_1\xi_d)\end{aligned} \tag{9}$$

where h_0 and d_0 are the nominal values and h_1 as well as d_1 define the perturbation range (% of nominal). Similarly, the stochastic potential interaction becomes $\hat{P}(\xi_h,\xi_d)$.

B. SOP Expansion

By expanding the stochastic potential interaction $\hat{P}(\xi_h, \xi_d)$ with stochastic orthogonal polynomials (SOPs), we can further derive the *stochastic geometric moments* (SGMs) below.

Assuming that there is one random distribution ξ related to one stochastic geometric variation, its related stochastic orthogonal polynomial is $\Phi(\xi)$. For example, for a Gaussian random distribution, $\Phi_i(\xi)$ is a Hermite polynomial [12], [13]

$$\Phi(\xi) = [1,\ \xi,\ \ \xi^2 - 1, \cdots,]^T. \tag{10}$$

As such, we can get the n-th order expansion of a potential coefficient matrix with $n+1$ Hermite polynomials by

$$\begin{aligned} \hat{P}(\xi) &= P_0\Phi_0(\xi) + P_1\Phi_1(\xi) + \cdots + P_n\Phi_n(\xi) \\ &= \sum_{k=0}^{n} P_k\Phi_k(\xi). \end{aligned} \tag{11}$$

Accordingly, the charge-density $\hat{q}(\xi)$ becomes:

$$\hat{q}(\xi) = \sum_{j=0}^{n} q_j\Phi_k(\xi). \tag{12}$$

By applying an inner-product with $\Phi_k(\xi)(k = 0, 1, ..., n)$

$$\langle \Phi_k,\ \hat{P}(\xi)\hat{q}(\xi) - b\rangle = 0 \tag{13}$$

to minimize the residue, we can derive an augmented linear system equation

$$\mathcal{P} \times \mathcal{Q} = \mathcal{B}. \tag{14}$$

The augmented P is calculated by

$$\mathcal{P} = (W_0 \otimes P_0 + W_1 \otimes P_1 + \cdots + W_n \otimes P_n). \tag{15}$$

Note that $\otimes$ represents a tensor product, and

$$W_k = \begin{pmatrix} w_{k,0,0} & w_{k,0,1} & \cdots & w_{k,0,n} \\ w_{k,1,0} & w_{k,1,1} & \cdots & w_{k,1,n} \\ \vdots & \vdots & w_{k,i,j} & \vdots \\ w_{k,n,0} & w_{k,n,1} & \cdots & w_{k,n,n} \end{pmatrix},$$

where $w_{k,i,j} =< \Phi_k\Phi_i\Phi_j >$ is the inner product of Hermite polynomials Φ_k, Φ_i, and Φ_j.

In addition, the augmented $\mathcal{Q}$, $\mathcal{B}$ and b_i become

$$\mathcal{Q} = \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_n \end{bmatrix},\ \mathcal{B} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}, b_i = \sum_{k=0}^{n}\sum_{j=0}^{n} w_{k,i,j} \times P_i \times q_j.$$

By further defining

$$\mathcal{P}_{i,j} = \sum_{k=0}^{n} w_{k,i,j} \cdot P_k,$$

The augmented system equation illustrated in Eq.(14) will have an explicit block-structure as shown below

$$\begin{pmatrix} \mathcal{P}_{0,0} & \mathcal{P}_{0,1} & \cdots & \mathcal{P}_{0,n} \\ \mathcal{P}_{1,0} & \mathcal{P}_{1,1} & \cdots & \mathcal{P}_{1,n} \\ \vdots & \vdots & \mathcal{P}_{i,j} & \vdots \\ \mathcal{P}_{n,0} & \mathcal{P}_{n,1} & \cdots & \mathcal{P}_{n,n} \end{pmatrix} \times \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}. \tag{16}$$

We use $n = 1$ as an example to illustrate the above general expression. First, the potential coefficient matrix $\hat{P}$ can be expanded with the first two Hermite polynomials by

$$\hat{P}(\xi) = P_0\Phi_0(\xi) + P_1\Phi_1(\xi) = P_0 + P_1\xi.$$

Then, the W_k $(k = 0, 1)$ matrix becomes

$$W_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ W_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix},$$

and the newly augmented coefficient system can be written as

$$\begin{aligned} \mathcal{P} &= W_0 \otimes P_0 + W_1 \otimes P_1 \\ &= \begin{pmatrix} P_0 & 0 & 0 \\ 0 & P_0 & 0 \\ 0 & 0 & P_0 \end{pmatrix} + \begin{pmatrix} 0 & P_1 & 0 \\ P_1 & 0 & 2P_1 \\ 0 & 2P_1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} P_0 & P_1 & 0 \\ P_1 & P_0 & 2P_1 \\ 0 & 2P_1 & P_0 \end{pmatrix}. \end{aligned} \tag{17}$$

By solving q_0, q_1, $\cdots$ and q_n, the Hermite polynomial expansion of charge-density can be obtained. Especially, the mean and the variance can be obtained from

$$E(q(\xi_d)) = q_0$$
$$Var(q(\xi_d)) = q_1^2 Var(\xi_d) + q_2^2 Var(\xi_d^2 - 1) = q_1^2 + 2q_2^2.$$

Considering that the dimension of $\hat{P}$ is further augmented, the complexity to solve the augmented system in Eq.(16) would be expensive. To mitigate this problem, we present a parallel FMM to reduce the cost of MVP evaluations in Section IV and an incremental preconditioner to reduce the cost of GMRES evaluation in Section V.

IV. Parallel Fast Multipole Method with SGM

Although the parallel fast multipole method has been discussed before such as [21], the extension to deal with stochastic variation for capacitance extraction needs to be addressed in the content of stochastic geometric moments (SGMs). In the following, we illustrate the parallel FMM considering the process variation.

The first step of a parallel FMM evaluation is to hierarchically subdivide space in order to from the clusters of panels. This is accomplished by using a tree-structure to represent each subdivision. We assume that there are N panels at the finest (or bottom) level. Providing depth H, we build an octtree with $H = \lceil log_8\frac{N}{n}\rceil$ by assigning n panels in one cube. In other words, there are 8^h cubes at the bottom level. A parallel FMM further distributes a number of cubes into different processors to evaluate $\mathcal{P}$. In the following steps, the stochastic $\mathcal{P}\times Q$ is evaluated in two passes: an upward pass for multipole-expansions (MEs) and a downward pass for local-expansions (LEs), both of which are further illustrated with details below.

A. Upward Pass

The upward-pass accumulates the multipole-expanded near-field interaction starting from the bottom level ($l = 0$). For each child cube (leaf) without variation (nominal contribution

to P_0) at the bottom level, it first evaluates the stochastic geometrical moment with (4) for all panels in that cube. If each panel experiences a variation ξ_d or ξ_h, it calculates $P_i(\xi) \times q(i \neq 0, \xi = \xi_d, \xi_h)$ by adding perturbation $h_i\xi_h$ or $d_i\xi_d$ to consider different variation sources, and then evaluates the stochastic geometric moments with (9).

After building the MEs for each panel, it transverses to the upper level to consider the contribution from parents. The moment of a parent cube can be efficiently updated by summing the moments of its 8 children via a M2M operation. Based on Eq.(6), the M2M translates the children's $\hat{M}_p$ into their parents.

The M2M operations at different parents are performed in parallel since there is no data-dependence. Each processor builds its own panels' stochastic geometric moments while ignoring the existence of other processors.

B. Downward Pass

The potential evaluation for the observer is managed during a downward pass. At l-th level ($l > 0$), two cubes are said to be *adjacent* if they have at least one common vertex. Two cubes are said to be *well separated* if they are not adjacent at level l but their parent cubes are adjacent at level $l-1$. Otherwise, they are said to be *far* from each other. The list of all the well-separated cubes from one cube at level l is called the *interaction list* of that cube.

From the top level $l = H - 1$, interactions from the cubes on the interaction list to one cube are calculated by a M2L operation at one level (M2L operation at top level). Assuming that a source-parent center r_c is changed to an observer-parent's center r_0, this leads to a LE (7) using the ME (4) when exchanging the r and d. As such, the M2L operation translates the source's $\hat{M}_p$ into the observer's $\hat{L}_p$ for a number of source-parents on the interaction list of one observer-parent at the same level. Due to the use of the interaction list, the M2L operations have the data-dependence that introduces overhead for a parallel evaluation.

After the M2L operation, interactions are further recursively distributed down to the children from their parents by a L2L operation (converse of the upward pass). Assume that the parent's center r_0 is changed to the child's center r_0' by a constant $\mathbf{h}$. Identical to the M2M update by (6), a L2L operation updates $\mathbf{r}$ by $\mathbf{r}' = \mathbf{r} + \mathbf{h}$ for all children's $\hat{L}_k$s. In this stage, all processors can perform the same M2L operation at the same time on different data. This perfectly employs the parallelism.

Finally, the FMM sums the L2L results for all leaves at the bottom level ($l = 0$) and tabulates the computed products $P_i \times q_j$ ($i, j = 0, 1, \cdots, n$). By summing up the products in order, the FMM returns the product $\mathcal{P} \times Q^{(i)}$ in (16) for the next GMRES iteration.

C. Data Sharing and Communication

The total runtime complexity for the parallel FMM using stochastic geometrical moments can be estimated by $O(N/B)+O(log_8 B)+C(N,B)$, where N is the total number of panels and B is the number of used processors. The $C(N,B)$ implies communication or synchronization overhead. Therefore, it is desired to minimize the overhead of data sharing and communication.

We notice that data dependency mainly comes from the interaction list during M2L operations. In this operation, a local cube needs to know the ME moments from cubes in its interaction list. To design a task distribution with small latency between computation and communication, our implementation uses a complement interaction list and prefetch operation.

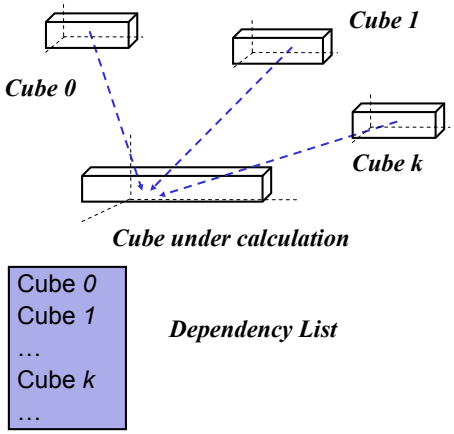


Fig. 2. Prefetch operation in M2L.

As shown in Figure(2), the complement interaction list (or dependency list) for the cube under calculation records cubes that require its ME moments to be listed within the shaded area. As such, it first anticipates which ME moments will be needed by other dependent cubes (such as Cube 0, $\cdots$, Cube k shown in Figure(2)) and distributes the required ME moments prior to the computation. From the point of view of these dependent cubes, they can "prefetch" the required ME moments. Therefore, the communication overhead can be significantly reduced.

V. Incremental GMRES

The parallel FMM presented in Section IV provides a fast matrix-vector-product for the fast GMRES iteration. As discussed in Section II and III, another critical factor for a fast GMRES is the construction of a good preconditioner. In this section, to improve the convergence of GMRES iteration, we first present a deflated power iteration to improve convergence during the extraction. Then, we introduce an incremental precondition in the framework of the deflated power iteration.

A. Deflated Power Iteration

The convergence of GMRES can be slow in the presence of degenerated small eigen values of the potential matrix $\mathcal{P}$, such as the case for most extraction problems with fine meshes. Constructing a preconditioner W to shift the eigen value distribution (spectrum) of a preconditioned matrix $W \cdot \mathcal{P}$ can significantly improve the convergence [22]. This is one of the so called *deflated GMRES* methods [23].

To avoid fully decomposing $\mathcal{P}$, an implicitly restarted Arnoldi method by ARPACK[1] can be applied to find its first K eigen values $[\lambda_1, ..., \lambda_K]$ and its Kth-order Krylov subspace composed by the first K eigen vectors $V_K = [v_1, ..., v_K]$, where

$$\mathcal{P}V_K = V_K D_K, \quad V_K^T V_K = I. \tag{18}$$

Note that D_K is a diagonal matrix composed of the first K eigen values

$$D_K = V_K^T A V_K = diag[\lambda_1, ..., \lambda_K]. \tag{19}$$

Then, an according spectrum preconditioner is formed

$$W = I + \sigma(V_K D_K^{-1} V_K^T), \tag{20}$$

which leads to a shifted eigen-spectrum using

$$(W \cdot \mathcal{P})v_i = (\sigma + \lambda_i)v_i \ i = 1, ..., K. \tag{21}$$

Note that σ is the shifting value that leads to a better convergence. This method is called *deflated power iteration*. Moreover, as discussed below, the spectral preconditioner W can be easily updated in an incremental fashion.

B. Incremental Precondition

The essence of the deflated GMRES is to form a preconditioner that shifts degenerated small eigen values. For a new $\mathcal{P}'$ with updated $\delta\mathcal{P}$, the distribution of the degenerated small eigen values change accordingly. Therefore, given a preconditioner W for the nominal system with the potential matrix $\mathcal{P}^{(0)}$, it would be expensive for another native Arnoldi iteration to form a new preconditioner W' for a new $\mathcal{P}'$ with updated $\delta\mathcal{P}$ from $\mathcal{P}^{(1)}$, $\cdots$, $\mathcal{P}^{(n)}$. Instead, we show that W can be incrementally updated as follows.

If there is a perturbation $\delta\mathcal{P}$ in $\mathcal{P}$, the perturbation δv_i of ith eigen vectors v_i $(k = 1, ..., K)$ can be given by [24]

$$\delta v_i = V_i B_i^{-1} V_i^T \delta\mathcal{P} v_i. \tag{22}$$

Note that V_i is the subspace composed of

$$[v_1, ..., v_j, ..., v_K]$$

and B_i is the perturbed spectrum

$$diag[\lambda_i - \lambda_1, ..., \lambda_i - \lambda_j, ..., \lambda_i - \lambda_K]$$

($j \neq i$, $i, j = 1, ..., K$). As a result, δV_K can be obtained similarly for K eigen vectors.

Assume that the perturbed preconditioner is W'

$$\begin{aligned} W' &= (I + \sigma V_K'(D_K')^{-1}(V_K')^T) \\ &= W + \delta W \end{aligned} \tag{23}$$

where

$$V_K' = V_K + \delta V_K, \quad D_K' = (V_K')^T P V_K'. \tag{24}$$

After expanding V_K' by V_K and δV_K, the incremental change in the preconditioner W can be obtained by

$$\delta W = \sigma(E_K - V_K D_K^{-1} F_K D_K^{-1} V_K). \tag{25}$$

[1]http://www.caam.rice.edu/software/ARPACK/

where

$$E_K = \delta V_K D_K^{-1} V_K^T + (\delta V_K D_K^{-1} V_K^T)^T. \tag{26}$$

and

$$F_K = \delta V_K^T V_K D_K + (\delta V_K^T V_K D_K)^T. \tag{27}$$

Note that all the above inverse operations only deal with the diagonal matrix D_K and hence the computational cost is low.

Since there is only one Arnoldi iteration to construct a nominal spectral preconditioner W, it can only be efficiently updated when $\delta\mathcal{P}$ changes. For example, $\delta\mathcal{P}$ is different when one alters the perturbation range h_1 of panel-width or changes the variation type from panel-width h to panel-distance d. We call this deflated GMRES method with the incremental precondition an *iGMRES method.*

For our problem in Eq.(16), we first analyze an augmented nominal system with

$$\begin{aligned} \mathbf{W} &= diag[W,\ W, \cdots,\ W] \\ \mathbf{P} &= diag[\mathcal{P}^{(0)},\ \mathcal{P}^{(0)}, \cdots,\ \mathcal{P}^{(0)}] \\ \mathbf{D}_K &= diag[D_K,\ D_K, \cdots,\ D_K] \\ \mathbf{V}_K &= diag[V_K,\ V_K, \cdots,\ V_K], \end{aligned}$$

which are all block diagonal with n blocks. Hence there is only one preconditioning cost from the nominal block $\mathcal{P}^{(0)}$. In addition, the variation contributes to the perturbation matrix by

$$\delta\mathcal{P} = \begin{pmatrix} 0 & \mathcal{P}_{0,1} & \cdots & \mathcal{P}_{0,n} \\ \mathcal{P}_{1,0} & 0 & \cdots & \mathcal{P}_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{P}_{n,0} & \mathcal{P}_{n,1} & \cdots & 0 \end{pmatrix}. \tag{28}$$

VI. Experiment Results

Based on the proposed algorithm, we have developed a program *piCap* using C++ on Linux network servers with Xeon processors (2.4GHz CPU and 2GB memory). In this section, we first validate the accuracy of stochastic geometrical moments by comparing them with the Monte-Carlo integral. Then, we study the parallel runtime scalability when evaluating the potential interaction using MVP with charge. In addition, the incremental GMRES preconditioner is verified when compared to its non-incremental counterpart with total runtime.

A. Accuracy Validation

To validate the accuracy of *SGM* by first-order and second-order expansions, we use two distant square panels. The nominal center-to-center distance d is d_0, and nominal panel width h is h_0.

1) Incremental Analysis: One possible concern is about the accuracy of incremental analysis, which considers independent variation sources separately and combines their contributions to get the total variable capacitance. In order to validate this, we first introduce panel width variation (Gaussian distribution with perturbation range h_1), and calculate the variable capacitance distribution. Then, panel distance variation d_1 is added and the same procedure is conducted. As such, according to incremental analysis, we can obtain the total

TABLE I
INCREMENTAL ANALYSIS VS. MONTE CARLO METHOD

2 panels, $d_0 = 10\mu m, h_0 = 2\mu m$, $d_1 = 30\%d_0$, $h_1 = 30\%h_0$			
	Incremental Analysis (fF)	Monte Carlo (fF)	Error (%)
$\mu_{C_{ij}}$	-1.1115	-1.1137	0.19
$\sigma_{C_{ij}}$	0.11187	0.11211	0.21
2 panels, $d_0 = 25\mu m, h_0 = 5\mu m$, $d_1 = 20\%d_0$, $h_1 = 20\%h_0$			
	Incremental Analysis (fF)	Monte Carlo (fF)	Error (%)
$\mu_{C_{ij}}$	-2.7763	-2.7758	0.018
$\sigma_{C_{ij}}$	0.19477	0.194	0.39

TABLE II
ACCURACY AND RUNTIME(S) COMPARISON BETWEEN MC(3000), *piCap*.

2 panels, $d_0 = 7.07\mu m, h_0 = 1\mu m$, $d_1 = 20\%d_0$		
	MC	piCAP
$C_{ij}(fF)$	-0.3113	-0.3056
Runtime (s)	2.6965	0.008486
2 panels, $d_0 = 11.31\mu m, h_0 = 1\mu m$, $d_1 = 10\%d_0$		
	MC	piCAP
$C_{ij}(fF)$	-0.3861	-0.3824
Runtime (s)	2.694	0.007764
2 panels, $d = 4.24\mu m, h_0 = 1\mu m$, $d_1 = 20\%d_0, h_1 = 20\%$		
	MC	piCAP
$C_{ij}(fF)$	-0.2498	-0.2514
Runtime (s)	2.7929	0.008684

capacitance as a superposition of nominal capacitance and both variation contributions. Moreover, we introduce the Monte Carlo simulations (10000 times) as the baseline, where both variations are introduced simultaneously. The comparison is shown in Table I, and we can observe that the results from incremental analysis can achieve high accuracy.

Actually, it is ideal to consider all variations simultaneously, but the dimension of system can increase exponentially with the number of variations and thus the complexity is prohibited. As a result, when the variation sources are independent, it is possible and necessary to separate them by solving the problem with each variation individually.

2) Stochastic Geometrical Moments: Next, the accuracy of proposed method based on stochastic geometrical moments (SGM) is verified with the same two panel examples. To do so, we introduce a set of different random variation ranges with Gaussian distribution for their distance d and width h. For this example, Monte-Carlo method is used to validate the accuracy of stochastic geometrical moments.

First, Monte-Carlo method calculates their C_{ij} 3000 times and each time the variation with a normal distribution is introduced to distance d randomly.

Then, we introduce the same random variation to geometric moments in (9) with stochastic polynomial expansion. Because of an explicit dependence on geometrical parameters according to (4), we can efficiently calculate $\hat{C}_{ij}$. Table II shows the C_{ij} value and runtime using the aforementioned two approaches. The comparison in Table II shows that stochastic geometric moments can not only keep high accuracy, which yields an average error of 1.8%, but also are up to ~ 347 faster than the Monte-Carlo method.

TABLE III
MVP RUNTIME (SECONDS)/SPEEDUP COMPARISON FOR FOUR DIFFERENT EXAMPLES

#wire	20	40	80	160
#panels	12360	10320	11040	12480
1 proc	0.737515/1.0	0.541515/1.0	0.605635/1.0	0.96831/1.0
2 procs	0.440821/1.7X	0.426389/1.4X	0.352113/1.7X	0.572964/1.7X
3 procs	0.36704/2.0X	0.274881/2.0X	0.301311/2.0X	0.489045/2.0X
4 procs	0.273408/2.7X	0.19012/2.9X	0.204606/3.0X	0.340954/2.8X

B. Speed Validation

In this part, we study the runtime scalability using a few large examples to show both the advantage of the parallel FMM for MVP and the advantage of the deflated GMRES with incremental preconditions.

1) Parallel Fast Multipole Method: The four large examples are comprised of $20, 40, 80$ and 160 conductors, respectively. For the two-layer example with 20 conductors, each conductor is of size $1\mu m \times 1\mu m \times 25\mu m$ (width$\times$thick$\times$length), and *piCap* employs a uniform $3\times3\times50$ discretization. Fig.(3) shows its structure and surface discretization.

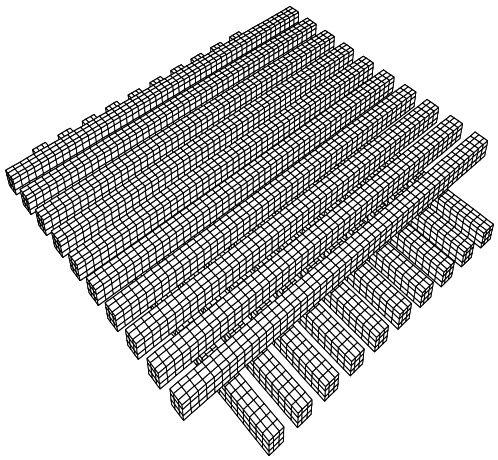


Fig. 3. The structure and discretization of two-layer example with 20 conductors.

For each example, we use a different number of processors to calculate the MVP of $P \times q$ by the parallel FMM. Here we assume that only d has a 10% perturbation range with Gaussian distribution. As shown in Table III, the runtime of the parallel MVP decreases evidently when more processors are involved. Due to the use of the complement interaction list, the latency of communication is largely reduced and the runtime shows a good scalability versus the number of processors. Moreover, the total MVP runtime with four processors is about $3X$ faster on average than runtime with a single processor.

It is worth mentioning that MVP needs to be performed many times in the iterative solver such as GMRES. Hence, even a small reduction of MVP runtime can lead to an essential impact on the total runtime of the solution, especially when the problem size increases rapidly.

2) Deflated GMRES: piCap has been used to perform analysis for three different structures as shown in Fig.(4). The

first is a plate with size $32\mu m \times 32\mu m$ and discretized as 16×16 panels. The other two examples are Cubic capacitor and Bus2x2 cross-over structures. For each example, we can obtain two stochastic equation systems in (17) by considering variations separately from width h of each panel and from the centric distance d between two panels, both with 20% perturbation ranges from their nominal values which should obey the Gaussian distribution.

To demonstrate the effectiveness of the deflated GMRES with a spectral preconditioner, two different algorithms are compared in Table IV. In the baseline algorithm (column "diagonal prec."), it constructs a simple preconditioner using diagonal entries. As the fine mesh structure in the extraction usually introduces degenerated or small eigen values, such a preconditioning strategy within the traditional GMRES usually needs much more iterations to converge. In contrast, since the deflated GMRES employs the spectral preconditioner to shift the distribution of non-dominant eigen values, it accelerates the convergence of GMRES leads to a reduced number of iterations. As shown by Table IV, the deflated GMRES consistently reduces the number of iterations by 3X on average.

TABLE IV
RUNTIME AND ITERATION COMPARISON FOR DIFFERENT EXAMPLES.

	#panel	#variable	diagonal prec.		spectral prec.	
			# iter	time	# iter	time
single plate	256	768	29	24.594	11	8.625
cubic	864	2592	32	49.59	11	19.394
cross-over	1272	3816	41	72.58	15	29.21

3) Incremental Preconditioner: With the spectral preconditioner, an incremental GMRES can be designed easily to update the preconditioner when considering different stochastic variations. It quite often happens that a change occurs in the perturbation range of one geometry parameter or in the variation type from one geometry parameter to the other. As the system equation in (17) is augmented to 3X larger than the nominal system, it becomes computationally expensive to apply any non-incremental GMRES methods whenever there is a change from the variation. As shown by the experiments, the incremental preconditioning in the deflated GMRES can reduce the computation cost dramatically.

As described in Section V, iGMRES needs to perform the precondition only one time for the nominal system and to update the preconditioner with perturbations from matrix block $P^{(1)}$. In order to verify the efficiency of such an incremental preconditioner strategy, we apply two different perturbation ranges for h_1 for panels of the two-layer 20 conductors shown in Fig. (3). Then, we compare the total runtime of the iGMRES and GMRES, both with the deflation. The results are shown in Table V.

From Table V, we can see that a non-incremental approach needs to construct its preconditioner whenever there is an update of variations, which is very time consuming. Our proposed *iGMRES* can reduce CPU time greatly during the construction of the preconditioner by only updating the nominal spectral preconditioner incrementally with (25). The result of iGMRES shows a speedup up to $15X$ over non-incremental algorithms and only iGMRES can finish all large-scale examples up to 14760 panels.

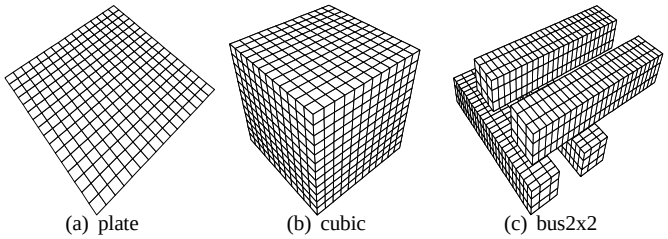


Fig. 4. Test structures:(a)plate;(b)cubic;(c)cross-over2x2

TABLE V
TOTAL RUNTIME (SECONDS) COMPARISON FOR 2-LAYER 20-CONDUCTOR BY DIFFERENT METHODS

discretization w×t×l	#panel	#variable	Total Runtime(s)	
			non-incremental	incremental
$3 \times 3 \times 7$	2040	6120	419.438	81.375
$3 \times 3 \times 15$	3960	11880	3375.205	208.266
$3 \times 3 \times 24$	6120	18360	-	504.202
$3 \times 3 \times 60$	14760	44280	-	7584.674

VII. CONCLUSIONS

In this paper, we have proposed the use of geometrical moments to capture local random variations for full-chip capacitance extraction. Based on geometrical moments, the stochastic capacitance can be thereby calculated via stochastic orthogonal polynomials (SoPs) by fast multi-pole method (FMM) in a parallel fashion. As such, the complexity of the matrix-vector product (MVP) can be largely reduced to evaluate both nominal and stochastic values. Moreover, one incrementally preconditioned GMRES is developed to consider different types of update of variations with an improved convergence by spectrum deflation.

A number of experiments show that our approach is $\sim 347X$ faster than the Monte-Carlo based evaluation of variation with a similar accuracy, up to $3X$ faster than the serial method in MVP, and up to $15X$ faster than non-incremental GMRES methods. In detail, the observed speedup of our approach is analyzed from twofold: the first is from the efficient parallel FMM, and the other is from the non-Monte-Carlo evaluation by SoPs. The future work is planned to extend our approach to deal with the general capacitance extraction with a non-square-panel geometry.

REFERENCES

[1] Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas, "Impact of interconnect variations on the clock skew of a gigahertz microprocessor," in *Proceedings of the 37th Annual Design Automation Conference*, Los Angeles, California, United States, pp. 168–171, 2000.

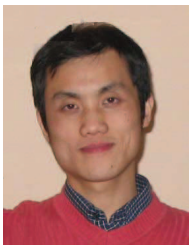
[2] R. Chang, Y. Cao, and C. Spanos, "Modeling the electrical effects of metal dishing due to CMP for on-chip interconnect optimization," *Electron Devices, IEEE Transactions on*, vol. 51, no. 10, pp. 1577–1583, 2004.

[3] Z. Zhu and J. White, "Fastsies: A fast stochastic integral equation solver for modeling the rough surface effect," in *Proc. IEEE/ACM Int. Conf. Computer-aided-design (ICCAD)*, Los Alamitos, CA, USA, pp. 675–682, 2005.

[4] H. Zhu, X. Zeng, W. Cai, J. Xue, and D. Zhou, "A sparse grid based spectral stochastic collocation method for variations-aware capacitance extraction of interconnects under nanometer process technology," in *Proc. IEEE/ACM Design, Automation, and Test in Europe (DATE)*, Nice, France, pp. 1514–1519, 2007.
[5] J. Cui, G. Chen, R. Shen, S. Tan, W. Yu, and J. Tong, "Variational capacitance modeling using orthogonal polynomial method," in *Proceedings of the 18th ACM Great Lakes symposium on VLSI*, Orlando, Florida, USA, pp. 23–28, 2008.
[6] H. Yu, C. Chu, Y. Shi, D. Smart, L. He, and S. Tan, "Fast analysis of large scale inductive interconnect by block structure preserved macromodeling," *IEEE Tran. on Very Large Scale Integration Systems (TVLSI)*, vol. 18, no. 10, pp. 1399–1411, 2010.
[7] H. Yu and L. He, "A provably passive and cost efficient model for inductive interconnects," *IEEE Tran. on Computer-aided-design (TCAD)*, vol. 24, no. 8, pp. 1283–1294, 2005.
[8] K. Nabors and J. White, "FastCap: A multipole accelerated 3D capacitance extraction program," *IEEE Tran. on Computer-aided-design (TCAD)*, vol. 10, no. 11, pp. 1447–1459, 1991.
[9] S.Kapur and D.Long, "IES3: a fast integral equation solver for efficient 3-dimensional extraction," in *Proc. IEEE/ACM Int. Conf. Computer-aided-design (ICCAD)*, San Jose, CA , USA, pp. 448 –455, 1997.
[10] W. Shi, J. Liu, N. Kakani, and T. Yu, "A fast hierarchical algorithm for 3-D capacitance extraction," in *Proc. ACM/IEEE Design Automation Conf. (DAC)*, San Francisco, California, United States, pp. 212–217, 1998.
[11] Y. Yang, P. Li, V. Sarin, and W. P. Shi, "Impedance extraction for 3-D structures with multiple dielectrics using preconditioned boundary element method," in *Proc. IEEE/ACM Int. Conf. Computer-aided-design (ICCAD)*, San Jose, CA, USA, pp. 7–10, 2007.
[12] D. Xiu and G. E. Karniadakis, "The Wiener-Askey polynomial chaos expansion for stochastic differential equations," *SIAM J. Scientific Computing*, vol. 24, pp. 619–644, 2002.
[13] S. Vrudhula, J. M. Wang, and P. Ghanta, "Hermite polynomial based interconnect analysis in the presence of process variations," *IEEE Tran. on Computer-aided-design (TCAD)*, vol. 25, no. 10, pp. 2001–2011, 2006.
[14] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *IEEE J. of Solid State Circuits*, vol. 305, no. 3, pp. 1433–1439, 1989.
[15] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.
[16] Y. Saad, *Iterative methods for sparse linear systems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2nd ed., 2003.
[17] M. S. Warren and J. K. Salmon, "A parallel hashed oct-tree n-body algorithm," in *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, Supercomputing '93, Portland, Oregon, United States, pp. 12–21, 1993.
[18] W. T. Rankin, III, *Efficient parallel implementations of multipole based n-body algorithms*. PhD thesis, Durham, NC, USA, 1999.
[19] J. D. Jackson, *Classical Electrodynamics*. John Wiley and Sons, 1975.
[20] C. Brau, *Modern Problems In Classical Electrodynamics*. Oxford Univ. Press, 2004.
[21] L. Ying, G. Biros, D. Zorin, , and H. Langston, "A new parallel kernel-independent fast multi-pole method," in *Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, Phoenix, AZ, USA, pp. 14–23, 2003.
[22] L. Giraud, S. Gratton, and E. Martin, "Incremental spectral preconditioners for sequences of linear systems," *Appl. Num. Math.*, vol. 57, pp. 1164–1180, 2007.
[23] V. Simoncini and D. B. Szyld, "Recent computational developments in Krylov subspace methods for linear systems," *Numerical Linear Algebra with Application*, vol. 14, pp. 1–59, 2007.
[24] G. W. Stewart, *Matrix algorithms (Volume II): Eigensystems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001.

Fang Gong (S'08) received his B.S. degree from Computer Science Department at Beijing University of Aeronautics and Astronautics in 2005. Also, he graduated from Computer Science Department at Tsinghua University with M.S. degree in 2008. After that, he continue to work toward his Ph.D. degree in the Electrical Engineering Department at University of California, Los Angeles. His research interests mainly focus on numerical computing and stochastic techniques for CAD, including fast circuit simulation, yield estimation and optimization. He also works on numerics parallel and distributed computing.

Hao Yu (S'02–M'06) obtained his B.S. degree from Fudan University (Shanghai China) and M.S/Ph. D degrees both from electrical engineering department at UCLA, with major of the integrated circuit and embedded computing. He was a senior research staff at Berkeley Design Automation (BDA), one of top-100 start-ups selected by Red-herrings at Silicon Valley. Since October 2009, he is an assistant professor at circuits and systems division of electrical and electronic engineering school, Nanyang Technological University (NTU), Singapore. Dr. Yu has 43 referred international publications, 5 book/chapters, 1 best paper award in ACM Transactions on Design Automation of Electronic Systems (TODAES), 2 best paper award nominations in design automation conference (DAC) and international conference of computer-aided-design (ICCAD), 1 inventor award from Semiconductor research cooperation (SRC), and 4 patent applications in pending. He is the associate editor of Journal of low power electronics, reviewer of IEEE TCAD, TCAS-I/II, TVLSI, ACM-TODAEs, VLSI Integration, technical program committee member and session chair of several conferences. The industry work at BDA is also recognized with an EDN magazine innovation award and multi-million venture capital funding.



Lingli Wang (M'99) received the Master degree from Zhejiang University in 1997 and the Ph.D degree from Napier University, UK in 2001. Both are in the area of electrical engineering. He has worked in Altera European Technology Center for 4 years. From 2005, he joined State-Key-Lab of ASIC and System, Fudan University, China. Currently he is a full professor with State Key Lab of ASIC and System, School of Microelectronics, Fudan University, Shanghai. His research interests include logic synthesis, reconfigurable computing, and quantum computing.

Lei He (IEEE M'99–SM'08) is a professor at electrical engineering department, University of California, Los Angeles (UCLA) and was a faculty member at University of Wisconsin, Madison between 1999 and 2002. He also held visiting or consulting positions with Cadence, Empyrean Soft, Hewlett-Package, Intel, and Synopsys, and was technical advisory board member for Apache Design Solutions and Rio Design Automation. Dr. He obtained Ph.D. degree in computer science from UCLA in 1999. His research interests include modeling and simulation, VLSI circuits and systems, and cyber physical systems. He has published one book and over 200 technical papers with 12 best paper nominations mainly from Design Automation Conference and International Conference on Computer-Aided Design and five best paper or best contribution awards including the ACM Transactions on Electronic System Design Automation 2010 Best Paper Award.