PiCAP: A Parallel and Incremental Capacitance Extraction Considering Stochastic Process Variation

Fang Gong
UCLA EE Department
Los Angeles, CA 90095
gongfang@ucla.edu

Hao Yu
Berkeley Design Automation
Santa Clara, CA 95054
hao.yu@berkeley-da.com

Lei He
UCLA EE Department
Los Angeles, CA 90095
lhe@ee.ucla.edu

ABSTRACT

It is unknown how to include stochastic process variation into fast-multipole-method (FMM) for a full chip capacitance extraction. This paper presents a parallel FMM extraction using stochastic polynomial expanded geometrical moments. It utilizes multiprocessors to evaluate in parallel for the stochastic potential interaction and its matrix-vector product (MVP) with charge. Moreover, a generalized minimal residual (GMRES) method with deflation is modified to incrementally consider the nominal value and the variance. The overall extraction flow is called piCAP. Experiments show that the parallel MVP in piCAP is up to 3X faster than the serial MVP, and the incremental GMRES in piCAP is up to 15X faster than non-incremental GMRES methods.

Categories and Subject Descriptors: B.7.2[Hardware]: Integrated circuits – Design aids

General Terms: Algorithms, design

Keywords: Capacitance extraction, Stochastic geometrical moments, Parallel fast-multipole method, Incremental precondition

1. INTRODUCTION

Capacitance extraction, which considers process variations, has recently regained attention. As IC designs are approaching processes below 45nm, the fabricated interconnect and dielectric can show significant difference from the nominal shape due to large uncertainties from chemical mechanical polishing (CMP), etching and lithography [1, 2, 3, 4, 5, 6]. As a result, the value of extracted capacitance can differ from the nominal value by a large margin, which may further lead to a significant variability for the timing analysis. This leads to a need to accurately extract the capacitance with stochastic process variation.

An accurate extraction usually leads to higher computational complexity. To avoid discretizing the entire space, the boundary element method (BEM) is one approach to evaluate capacitance by discretizing the surface into panels on the boundary of the conductor and the dielectric. Though it results in a discretized system with a small dimension, the discretized system under BEM is dense. FastCap [7] employs an iterative method to solve the dense system by a generalized minimal residual (GMRES) method. Instead of performing the expensive LU decomposition, the GMRES iteratively reaches the solution with use of the matrix-vector multiplication. The computational cost of the matrix-vector-product (MVP) can be reduced by a fast-multipole-method (FMM) [7], a low-rank approximation [8], or a hierarchical-tree decomposition [9].

There are a few recent works [3, 4, 5, 6] on the interconnect extraction considering the process variation. [3] calculates the variational capacitance through a stochastic integral, where the computational cost is alleviated by the low-rank approximation. The works in [4, 5, 6] represent the variation by the stochastic orthogonal polynomial [10] when calculating a variational capacitance. [4] employs a spectral collocation, [5] applies a perturbation analysis, and [6] is in the framework of the stochastic integral similar to [3]. 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, 6]. 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. Including the stochastic integral or collocation in the full chip capacitance extraction would be computationally expensive. Instead, similar to deal with the stochastic analog mismatch for transistors [11], a full chip extraction needs to explore an explicit relation between the stochastic variation and the geometrical parameter. As such, the electrical property has an explicit dependence on geometrical parameters, and it can lead to a scalable extraction algorithm similar to [7, 8, 9].

The complexity of fast full chip extractions in [7, 8, 9] generally comes from two parts: the evaluation of MVP and the preconditioned GMRES iteration. Note that the FMM algorithm in [7] can be parallelized on the modern multi-processor system. However, it is unknown how to incorporate the stochastic process variation in the FMM. Moreover, the GMRES needs to be designed in an incremental fashion to consider the update from the process vacation.

Parallel and incremental analysis are the two fundamental concepts in reducing computational cost. In this paper, we have developed a parallel and incremental full chip capacitance extraction considering stochastic variation, namely *piCAP*. Our contributions are two-fold. First, we reveal that the potential interaction can be 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. Therefore, the variation can be incorporated into a modified FMM algorithm to evaluate the MVP in parallel. Next, we develop a preconditioned GMRES method that can incrementally update the preconditioner with different variations. Experiments show that our method with the 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 in Section 2. In Section 3, we present the parallel FMM based on the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC'09, July 26-31, 2009, San Francisco, California, USA.
Copyright 2009 ACM 978-1-60558-497-3/09/07 ...$10.00.

stochastic geometrical moments. In Section 4, we introduce a new incremental GMRES method and its application for solving large-augmented stochastic systems. We present experiment results in Section 5 and conclude the paper in Section 6.

2. BACKGROUND

The boundary element method (BEM) starts with an integral equation

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

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 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* [7]:

$$Pq = b, \quad (2)$$

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 potential. By probing v 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:

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

Since panels are assumed as well-separated, P_{ij} can be well approximated by $\frac{1}{4\pi\epsilon_0 |r_i - r_j|}$ [7, 5].

The resulting potential coefficient matrix P is usually dense in the BEM method. As such, directly solving (2) would be computationally expensive. FastCap [7] applies the iterative GMRES method to solve (2). Instead of performing an expensive LU decomposition of the dense P, GMRES first forms a preconditioned $W \cdot P$ to approximate P and then finds an initial solution q_0 accordingly by

$$(W \cdot P) q_0 = b.$$

Then, using either the multipole-expansion [7], low-rank approximation [8] or hierarchical-tree method [9] to efficiently evaluate the matrix-vector-product (MVP) for Pq_i, the GMRES method minimizes the residue error

$$min: \ ||b - Pq_i||$$

iteratively till converged.

Therefore, the use of GMRES requires a well-designed preconditioner and a fast matrix-vector-product. To this end, our stochastic extraction has a parallel evaluation Pq with variations presented in Section 3, and has an incremental preconditioner presented in Section 4.

3. PARALLEL FMM BY STOCHASTIC GEOMETRICAL MOMENT

A 3D 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. As such, the MVP of Pq can be provided efficiently during the GMRES iteration. Since such a spatial decomposition is geometrically dependent, it is helpful to express P by 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 property to preserve in presence of the stochastic variation. In this section, we first introduce a stochastic geometrical moment to calculate the potential interaction with variations. Using stochastic geometrical moments, we then illustrate how to develop a parallel FMM considering variations.

3.1 Stochastic Geometrical Moment

The process variation includes global systematic variation and local random variation. This paper focuses on the local random variation, or stochastic variations. Due to the local random variation, the width of discretized panel may show deviations from nominal values, and so does the distance between panels. Therefore, our paper considers two geometrical parameters under the stochastic vacation: panel-distance (d) and panel-width (h). With expansions in Cartesian coordinates, we can connect the potential interaction with the geometry parameter through *geometrical moments* that can be extended to consider stochastic variations.

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

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

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

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

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

In Cartesian coordinates ($x-y-z$), when the observer is outside the source region ($d > r$), a *multipole expansion* (ME) [12, 13] can be defined as

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

by expanding r around r_c, where

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

Note that ∇ 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, changing 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)

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

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)

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

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_i, L_i and their spatial shifts show an explicit dependence on the panel-width h and panel-distance d. As such, we call M_i and L_i *geometrical moments*, and we also express P

as a geometrical-dependence function $P(h,d)$ via geometrical moments. Usually, a second-order expansion can be accurate enough to calculate P_{ij}.

Moreover, assuming that the 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

$$\hat{M}_p(\xi_h,\xi_d) = M_p(h_0+h_1\xi_h, d_0+d_1\xi_d)$$
$$\hat{L}_p(\xi_h,\xi_d) = L_p(h_0+h_1\xi_h, d_0+d_1\xi_d) \quad (8)$$

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

3.2 Parallel FMM

Based on the multipole expansion (ME) (4), the local expansion (LE) (7), and the stochastic geometrical moments (8) of both, we can apply the $O(n)$ M2M, M2L and L2L operations and form a fast-multipole method (FMM) as shown in Fig. 1. We assume that there are N panels at the finest (or bottom) level. Providing depth H, we build an oct-tree with $H = \lceil log_8 \frac{N}{n} \rceil$ by assigning n panels in one cube. I.e., there are 8^h cubes at the bottom level. A parallel FMM further distributes a number of cubes into different processors to evaluate P. The decomposition of the tasks needs to minimize the communication cost and balance the workload.

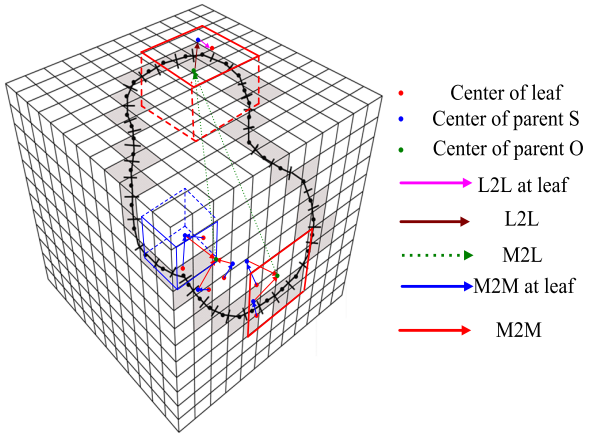


Figure 1: The 3D flow of FMM including M2M, M2L and L2L operations.

3.2.1 Upward Pass

The upward-pass manages the computation during the source expansion. It accumulates the multipole-expanded near-field interaction starting from the bottom level ($l = 0$). For each child cube (leaf) at the bottom level, it first evaluates the stochastic geometrical moment (8) for all panels in that cube. Then, 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 an M2M operation. Based on (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.

3.2.2 Downward Pass

The potential evaluation at the observer is managed during a downward pass. At lth 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* to 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 $l = H-1$, the interactions from cubes in the interaction list to one cube are calculated by an M2L operation at one level. Assuming that a source-parent center r_c is changed to a 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 to prevent a parallel evaluation.

After the M2L operation, interactions are further distributed down to the children from their parents by a L2L operation. 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), an L2L operation updates $\mathbf{r}$ by $\mathbf{r'} = \mathbf{r} + \mathbf{h}$ for all children's $\hat{L}_k$s in a parallel manner.

Finally, the FMM sums the L2L results for all leaves at the bottom level ($l = 0$) and returns $\hat{P}$ and its product with q_i for the next GMRES iteration.

3.2.3 Data Sharing and Communication

The total runtime complexity for the parallel FMM with 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, this part discusses how to minimize the overhead of data sharing and communication during a parallel evaluation. In our parallel FMM implementations,. the message-passing-interface (MPI) is used for the data communication and synchronization between multiple processors. 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 the computation and communication, our implementation uses a complement interaction list. For a given cube, its complement interaction list records cubes that require ME moments. As such, the implemented communication protocol first anticipates which ME moments will be needed by other processes, and then distributes the ME moments prior to the computation where they will be required. Therefore, the communication overhead can be significantly reduced.

4. INCREMENTAL GMRES

The parallel FMM presented in Section 3 provides a fast matrix-vector-product for the fast GMRES iteration. As discussed in Section 2, 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 GMRES method with the use of a spectral precondition. Then, we discuss how to incrementally update the spectral preconditioner for the deflated GMRES, and apply such an incremental GMRES to efficiently consider the perturbations for an augmented system equation.

4.1 Deflated GMRES

The convergence of GMRES can be slow in the presence of degenerated small eigen-values of the potential matrix P. This is the case for most extraction problems with fine meshes. Constructing a preconditioner W to shift the eigen-value distribution (spectrum) of a preconditioned matrix $W \cdot P$ can significantly improve the convergence [14]. This is one of the so called *deflated GMRES* methods [15].

To avoid fully decomposing 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

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

by the first K eigen-vector $V_K = [v_1, ..., v_K]$, where

$$PV_K = V_K D_K, \quad V_K^T V_K = I. \tag{9}$$

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

$$D_K = V_K^T A V_K = diag[\lambda_1, ..., \lambda_K]. \tag{10}$$

Then, an according spectrum preconditioner is

$$W = I + \sigma(V_K D_K^{-1} V_K^T), \tag{11}$$

which leads to a shifted eigen-spectrum by

$$WPv_i = (\sigma + \lambda_i)v_i \ i = 1, ..., K. \tag{12}$$

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

4.2 Incremental Precondition

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

If there is a perturbation δP ($P^{(1)}$) in P, the perturbation δv_i of ith eigen-vector v_i ($k = 1, ..., K$) can be given by [16]

$$\delta v_i = V_i B_i^{-1} V_i^T \delta P v_i. \tag{13}$$

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'

$$\begin{aligned} W' &= (I + \sigma V'_K (D'_K)^{-1} (V'_K)^T) \\ &= W + \delta W \end{aligned} \tag{14}$$

where

$$V'_K = V_K + \delta V_K, \quad D'_K = (V'_K)^T P V'_K. \tag{15}$$

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

$$\delta W = \sigma(E_K - V_K D_K^{-1} F_K D_K^{-1} V_K) \tag{16}$$

where

$$E_K = \delta V_K D_K^{-1} V_K^T + (\delta V_K D_K^{-1} V_K^T)^T, \tag{17}$$

and

$$F_K = \delta V_K^T V_K D_K + (\delta V_K^T V_K D_K)^T. \tag{18}$$

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

As there is only one Arnoldi iteration to construct a nominal spectral preconditioner W, it can be efficiently updated when $P^{(1)}$ changes. For example, $P^{(1)}$ 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.*

4.3 iGMRES for Stochastic Extraction

We further discuss how to apply iGMRES for our stochastic capacitance extraction in this part. For a full-chip extraction, simultaneously considering variations from all kinds of geometrical parameters would significantly increase model complexity, if not impossible. In this paper, we study the stochastic variation contributed by each parameter individually in an incremental fashion. Together with the incremental GMRES discussed above, the computational cost can be dramatically reduced for a large-scale extraction.

For example, to study the variation caused by the panel-distance d, one can assume that the potential coefficient matrix by

$$\hat{P}(\xi_d) = P^{(0)}(d_0) + P^{(1)}(d_1)\xi_d. \tag{19}$$

Note that as each panel center can experience a variation, $P^{(1)}$ is also a dense matrix with the same dimension as $P^{(0)}$.

Then, instead of performing an expensive Monte-Carlo simulation on ξ_d, one can span the distribution of ξ_d by an orthogonal polynomial [10, 4, 5, 6]. Based on the *Askey scheme* [10], a Gaussian distribution of ξ_d can be spanned by Hermite polynomials

$$\Phi(\xi_d) = [1, \ \xi_d, \ \ \xi_d^2 - 1, ...,]^T. \tag{20}$$

Accordingly, the charge-density vector q becomes

$$q(\xi_d) = \sum_{i=0} \alpha_j \Phi_j(\xi_d). \tag{21}$$

Usually, the expansion order is 2 to calculate the mean and the variance. As such, by further applying an inter-product with Φ_j $(i, j = 0, 1, 2)$ to minimize the residue,

$$\langle \hat{P}(\xi_d) q(\xi_h) - b, \ \Phi_j \rangle = 0 \tag{22}$$

we have

$$\begin{bmatrix} P^{(0)} & P^{(1)} & 0 \\ P^{(1)} & P^{(0)} & 2P^{(1)} \\ 0 & 2P^{(1)} & P^{(0)} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix} \tag{23}$$

due to the orthogonality of the Hermite polynomials.

By solving α_0, α_1 and α_2, the mean and the variance can be obtained from

$$\begin{aligned} &E(q(\xi_d)) = \alpha_0 \\ &Var(q(\xi_d)) = \alpha_1^2 Var(\xi_d) + \alpha_2^2 Var(\xi_d^2 - 1) = \alpha_1^2 + 2\alpha_2^2, \end{aligned}$$

where the parallel FMM discussed in Section 3 is applied to calculate the MVPs among $P^{(0)}$, $P^{(1)}$ and α_0, α_1, α_2 with the use of the spectrum preconditioner.

The above procedure can be similarly applied to calculate the variance and the mean for the geometrical parameter h. Clearly, the stochastic orthogonal expansion leads to an augmented system with perturbed blocks in the off-diagonal. It increases the computational cost for any GMRES method, and remains an unresolved issue in the previous applications of the stochastic orthogonal polynomial [10, 4, 5, 6]. In addition, when variation changes, a different $P^{(1)}$ results. Forming a new preconditioner to consider the augmented (23) would therefore be expensive.

To efficiently solve the augmented (23) under different $\delta P(h)$ or $\delta P(d)$, we first analyze an augmented nominal system with

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

which are all block diagonal. Hence there is only one preconditioning cost from the nominal block $P^{(0)}$.

The variation contributes to the perturbation matrix by

$$\delta\mathcal{P} = \begin{bmatrix} 0 & P^{(1)} & 0 \\ P^{(1)} & 0 & 2P^{(1)} \\ 0 & 2P^{(1)} & 0 \end{bmatrix}. \tag{24}$$

Table 1: 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)	10.786037	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)	10.7763	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)	11.17167	0.008684

Based on (16) we can do an incremental update of the preconditioner W to consider a new variation $P^{(1)}$ when changing the perturbation range of h_1 or d_1. Moreover, we can also make an incremental update of W when changing the variation type from $P^{(1)}(h)$ to $P^{(1)}(d)$. This can dramatically reduce the cost when applying the deflated GMRES during the variational capacitance extraction. The same procedure can be easily extended for the high-order expansion with stochastic orthogonal polynomials.

5. EXPERIMENT RESULTS

We have implemented our *piCap* algorithm in c++. The experiments were carried out on Linux network servers with Xeon processors (2.4GHz CPU and 2GB memory). We first show the accuracy of using stochastic geometrical moments when compared to the Monte-Carlo integral. We then study the parallel runtime scalability when building the potential interaction and its MVP with charge. Finally, we show the impact of using the incremental GMRES preconditioner.

5.1 Accuracy Validation

The first example is two square panels with introduced random variation for their distance d and width h. In the experiment, we use a set of different perturbation ranges for d and h. First, Monte-Carlo method is used to calculate their C_{ij}s 3000 times, and each time the variation is added to d randomly with a normal distribution. As such, we can evaluate the mean value μ and standard deviation σ. Then we introduce the same random variation to geometric moments in (8) with stochastic polynomial expansion. Because of an explicit dependence on geometrical parameters according to (19), we can efficiently calculate $\hat{C}_{ij}$s from (23). Table I shows the C_{ij} value and runtime using the aforementioned two approaches. The comparison in Table I shows that stochastic geometric moments can not only keep high accuracy, which yields an average error of 1.8%, but also be up to $\sim 1000X$ faster than the Monte-Carlo method.

5.2 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 precondition.

5.2.1 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×thick×len), and *piCap* employs a uniform $3\times3\times50$ discretization. Fig.2 shows its structure and surface discretization.

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 10% perturbation range. As shown in Table 2, 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 vs. the number of processors. For all examples, the runtime with four processors is about $3X$ faster on average when compared to the runtime of the single processor.

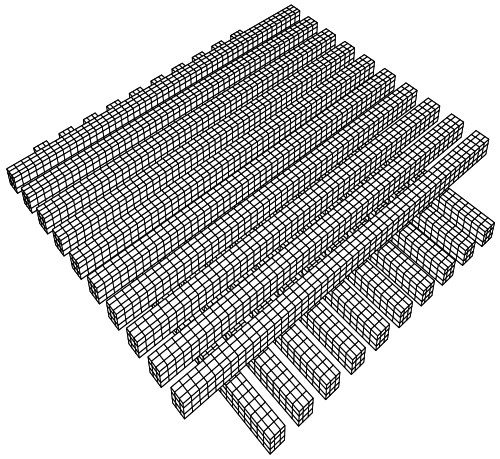


Figure 2: The structure and discretization of two-layer example with 20 conductors.

Table 2: 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

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 essential impact on the total runtime of the solution, especially when the problem size increases rapidly.

5.2.2 Deflated GMRES

piCap has been used to perform analysis for three different structures as shown in Fig.(3). The first is a plate with size $32\mu m \times 32\mu m$ and be 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 (23) by considering variations separately from width h of each panel and from the centric distance d between two panels, both with 20% perturbation range from their nominal values.

To demonstrate the effectiveness of the deflated GMRES with a spectral preconditioner, two different algorithms are compared in Table 3. 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 3, the deflated GMRES consistently reduces the number of iterations by 3X on average.

5.2.3 Incremental Preconditioner

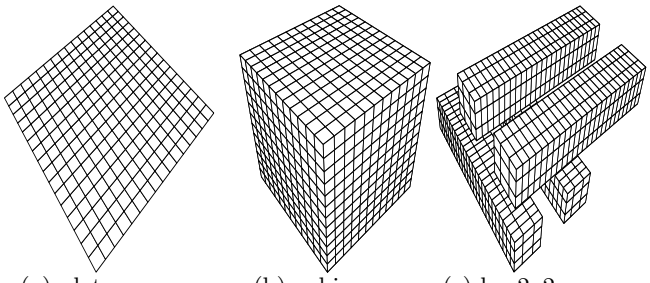
With the spectral preconditioner, an incremental GMRES can be designed easily to update the preconditioner when considering different stochastic variations. It quite often happens that a change occurs in the perturbation range of one geometry param-

Table 3: 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

eter, or in the variation type from one geometry parameter to the other. As the system equation in (23) 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 4, iGMRES only needs to perform the precondition 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. 2, and then compare the total runtime of the iGMRES and GMRES, both with the deflation. The results are shown in Table 4.



(a) plate (b) cubic (c) bus2x2

Figure 3: Test structures:(a)plate;(b)cubic;(c)cross-over2x2

Table 4: 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
$3 \times 3 \times 50$	12360	37080	-	3637.391

From Table 4, we can see that a non-incremental approach needs to constructs its preconditioner whenever there is an update of variations, which is very time consuming. For our proposed *iGMRES*, it can reduce much CPU time during the construction of the preconditioner by only updating the nominal spectral preconditioner incrementally with (16). 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 12360 panels.

6. CONCLUSIONS

To consider local random variations for a full chip capacitance extraction, this paper presents a fast parallel FMM method with an incremental precondition, namely piCAP. With the use of stochastic-polynomial expanded geometrical moments, the parallel FMM performs a fast evaluation of both potential interaction and its variation, and hence dramatically reduces the computational cost of the matrix-vector product (MVP). Moreover, the incrementally preconditioned GMRES is developed to consider the different update of variations with an improved convergence by deflation. Experiments on a few different large examples show that piCAP is hundreds of times faster than the Monte-Carlo based evaluation of variation with a similar accuracy, is up to 3X faster than the serial method in MVP, and is also up to 15X faster than non-incremental GMRES methods.

Future studies are planned to show the impact to the distribution of the eigenvalues when adding variations, and to identify the important parameters that lead to significant variations.

7. REFERENCES

[1] Y. Liu, S. Nassif, L. Pileggi, and A. Strojwas, "Impact of interconnect variations on the clock skew of a gigahertz microprocessor," in *Proc. ACM/IEEE Design Automation Conf. (DAC)*, 2000.

[2] R. Chang, Y. Cao, and C. Spanos, "Modeling the electrical effects of metal dishing due to CMP for on-chip interconnect optimization," *IEEE Trans. on Semi. Manufacturing*, 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)*, 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)*, 2007.

[5] J. Cui, G. Chen, R. Shen, S. X.-D. Tan, W. Yu, and J. Tong, "Variational capacitance modeling using orthogonal polynomial method," in *Proc. ACM/IEEE Great Lake VLSI*, 2008.

[6] T. El-Moselhy and L. Daniel, "Stochastic integral equation solver for efficient variation-aware interconnect extraction," in *Proc. ACM/IEEE Design Automation Conf. (DAC)*, 2008.

[7] K. Narbos and J. White, "FastCap: A multipole accelerated 3D capacitance extraction program," *IEEE Tran. on Computer-aided-design (TCAD)*, pp. 1447–1459, 1991.

[8] S.Kapur and D.Long, "Ies3: a fast integral equation solver for efficient3-dimensional extraction," in *Proc. IEEE/ACM Int. Conf. Computer-aided-design (ICCAD)*, 1997.

[9] 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)*, 1998.

[10] 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)*, pp. 2001–2011, 2006.

[11] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *IEEE J. of Solid State Circuits*, pp. 1433–1439, 1989.

[12] J. D. Jackson, *Classical Electrodynamics*. John Wiley and Sons, 1975.

[13] C. Brau, *Modern Problems In Classical Electrodynamics*. Oxford Univ. Press, 2004.

[14] L. Giraud, S. Gratton, and E. Martin, "Incremental spectral preconditioners for sequences of linear systems," *Appl. Num. Math.*, pp. 1164–1180, 2007.

[15] V. Simoncini and D. Szyld, "Recent computational developments in Krylov subspace methods for linear systems," *Num. Lin. Alg. with Appl.*, pp. 1–59, 2007.

[16] G. W. Stewart, *Matrix algorithms (Volume II): Eigensystems*. SIAM, 2001.