

Variation-Aware Stochastic Extraction with Large Parameter Dimensionality: Review and Comparison of State of the Art Intrusive and Non-intrusive Techniques

Tarek El-Moselhy
Massachusetts Institute of Technology
tmoselhy@mit.edu

Luca Daniel
Massachusetts Institute of Technology
luca@mit.edu

ABSTRACT

In this paper we review some of the state of the art techniques for parasitic interconnect extraction in the presence of random geometrical variations due to uncertainties in the manufacturing processes. We summarize some of the most recent development in both sampling based (non-intrusive) and expansion based (intrusive) algorithms for the extraction of both general impedance and capacitance in the presence of random geometrical variations. In particular, for non-intrusive algorithms we discuss both the stochastic model reduction algorithm for general impedance extraction under uncertainty and the path recycling floating random walk algorithm for capacitance extraction under uncertainty. For intrusive algorithms we summarize the Neumann expansion, the standard stochastic Galerkin method, the combined Neumann Hermite expansion and the stochastic dominant singular vectors method. Finally, we end the paper by comparing the presented algorithms on four very large and complex impedance and capacitance extraction examples.

1. INTRODUCTION

Field solvers are very important in order to determine the performance of the electrical circuits in the presence of high frequency effects, proximity effects and electromagnetic and substrate coupling. Over the last decades a lot of research has been dedicated to the development of very efficient field solvers, which can handle very large structures. However, such solvers are optimized to handle a single deterministic structure, and do not account for possible geometrical variations produced by uncertainties in the manufacturing process. Due to technology scaling, such geometrical uncertainties are beginning to strongly affect the performance of both transistors and interconnect structures. In particular, the performance of the transistors is affected by both line edge roughness and the variations of the doping profile. The performance of on-chip interconnect structures is affected by width and thickness variations. Finally, the performance of off-chip interconnect structures is affected by surface roughness.

A large variety of CAD tools are required in order to fully analyze the effect of manufacturing processes on the performance of

the final electrical circuits. Among such tools are process simulation tools which transform uncertainties of the manufacturing process parameters into geometrical uncertainties. Then tools to transform geometrical uncertainties into uncertainties of the electrical parameters of both the transistors and the interconnects. And finally tools which transform the uncertainties of the electrical parameters of transistors and interconnects into uncertainties of the performance parameters of complex electrical circuits.

In this paper we focus on stochastic variation-aware solvers for interconnect structures, which analyze the effect of geometrical uncertainties on the electrical performance of the structures and aid in analyzing the reliability of the designed circuits. We will summarize the state of the art techniques which can be used to analyze the different types of geometrical variations (such as surface roughness and width/thickness variations) on the electrical characteristics (such as capacitance, resistance or full impedance) of interconnect structures. It should be noted that the on-chip width variations are typically produced by uncertainties of the lithography and etching processes and that the thickness variations are typically produced by the uncertainties in the chemical mechanical polishing process (CMP). In addition, off-chip surface roughness are typically produced by uncertainties in the electroplating procedure. It should also be noted that the geometrical uncertainties are treated in this paper as random variations. This is primarily due to two factors. First, despite the ability of process simulation tools to predict the geometrical description of a structure for a given set of process parameters, such process parameters are themselves typically assumed random and therefore the geometrical parameters predicted by the process simulation tool are random. Second, accurate process simulation tools require knowing the exact location of the interconnects in the full chip and details of their surrounding. Since such information is typically not available at the early design phases, this lack of information makes accurate process simulations impossible. This lack of knowledge is in turn reflected by assuming that the geometrical parameters of the interconnects are random.

This paper is organized as follows. Section 2 summarizes the basic mathematical setup of the problem and some of the tools which are necessary for analyzing the different methods presented in the rest of the paper. Then in section 3 sampling based methods are addressed. In particular details are given for both the stochastic model order reduction and the path recycling floating random walk methods. In section 4 the state of the art intrusive algorithms are summarized. Finally, in section 5 a comparison between the different solution methods is performed on a variety of large scale examples.

2. PROBLEM SETUP

The general impedance extraction problem is described by Maxwell's

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.

equations. Different formulations ranging from differential to integral formulations have been proposed in literature to solve the impedance extraction problem [1, 2, 3, 4, 5]. All discretization based methods, whether differential or integral, rely on first discretizing the set of unknowns using appropriate basis functions. The discretized equations are then transformed into a linear system of equations using standard testing procedure $\mathbf{Ax} = \mathbf{b}$.

For the case of stochastic variation-aware extraction the boundaries of the computational domain are described by random geometrical parameters. Discretization in the presence of random variations is a difficult task. Indeed, the only way to discretize a structure using a constant number of basis, while still allowing the structure to vary (see Figures 1 and 2), is to use stochastic basis functions, i.e. basis that have a stochastic support. Another way to understand the stochastic nature of the basis function, is to realize that in order to keep the total number of volume or surface discretization elements constant while still allowing the geometry to vary, is to allow the discretization elements themselves to vary. Galerkin or collocation testing is then used to transform the Maxwell's equations into a stochastic linear systems of equations. Notice that in case of Galerkin testing the test functions are themselves random. Following the testing procedure, the resulting stochastic linear systems of equations is of the form:

$$\mathbf{A}(\mathbf{p}) \mathbf{x}(\mathbf{p}) = \mathbf{b} \quad (1)$$

where \mathbf{p} is a vector of length N_p of the random geometrical parameters, $\mathbf{A}(\mathbf{p})$ is a known stochastic matrix of size $N \times N$, \mathbf{b} is the known RHS, and the objective is to obtain the complete distribution of the $N \times 1$ unknown vector $\mathbf{x}(\mathbf{p})$.

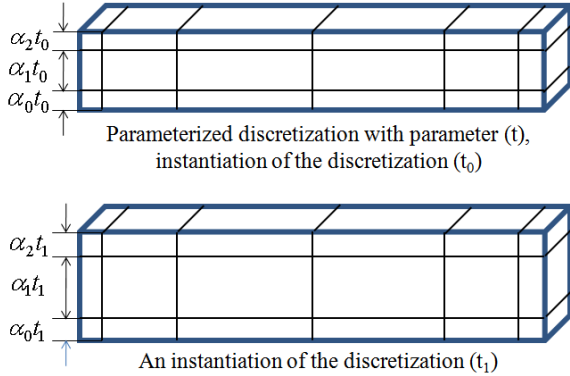


Figure 1: Discretized conductor surface using parameterized basis functions.

In electrical extraction applications, the output quantity of interest $y(\mathbf{p})$ (e.g. capacitance or admittance) is written as a linear function of the unknown vector $\mathbf{x}(\mathbf{p})$

$$y(\mathbf{p}) = \mathbf{c}^T \mathbf{x}(\mathbf{p}) \quad (2)$$

where \mathbf{c} is a column vector of size $N \times 1$. In certain applications, such as capacitance extraction, we might be interested in more than a single quantity (e.g. all coupling capacitances between a target conductor and all surrounding conductors). In such cases $y(\mathbf{p})$ is a vector rather than a scalar, and the vector \mathbf{c} becomes a matrix \mathbf{C} .

It should be noted that in the very special cases of capacitances or resistance extraction (which require only solving the Laplace equation) the extraction can be implemented most efficiently using random walk methods. Such methods are discretization-free and do not rely on assembling or solving any linear systems of equa-

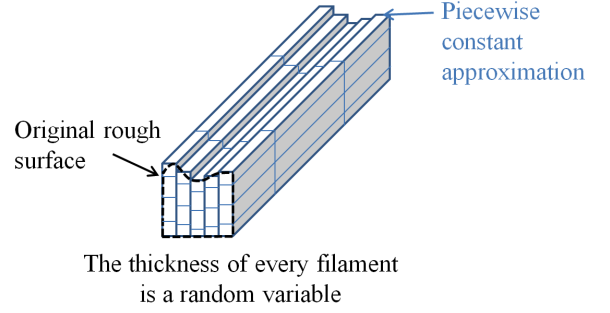


Figure 2: Discretized conductor volume. Conductor upper surface is rough. The thickness of every filament in a random variable.

tions [6, 7, 8, 9]. One of these methods, namely the floating random walk (FRW), is addressed in section 3.2

2.1 Random Variability Modeling

Most techniques presented in this paper work with general probability density functions, but for simplicity of exposition we assume that the stochastic vector \mathbf{p} is described by a multivariate Gaussian probability density function. We will further assume that in general such random variables are spatially correlated. Based on the spatial correlation function, we divide the variations into strongly spatially correlated and weakly spatially correlated variations. As an example of strongly spatially correlated variations, we consider surface roughness in “off-chip” interconnects. We will assume that the correlation function is Gaussian in order to guarantee the smoothness of the surface. However, other correlation functions, such as exponential, have been also applied in the literature [10]. As an example of weakly spatially correlated variations, we will consider the case of “on-chip” interconnect width and height variations. In such case we will in general assume that the random parameters in \mathbf{p} represent uncorrelated Gaussian variations in the geometrical dimensions (e.g. width and height) of the interconnects.

If the vector \mathbf{p} is composed of correlated random variables then a Karhunen Loeve expansion (or in other words a truncated Singular Value Decomposition SVD) is used to expand the random process as a summation of independent random variables $\vec{\eta}$ [11],

2.2 Polynomial Chaos Expansion (PCE)

For the general extraction problem the elements of the system matrix depend nonlinearly on the uncorrelated random vector $\vec{\eta}$. To simplify the computations and the representation of the nonlinear dependence, the matrix elements are often expanded in terms of orthogonal basis polynomials $\Psi_k(\vec{\eta})$ in the variable $\vec{\eta}$. Such expansion is referred to as polynomial chaos expansion [12].

$$\mathbf{A}(\mathbf{p}) = \sum_{i=0}^K \mathbf{A}_i \Psi_i(\vec{\eta}) \quad (3)$$

where K is a function of both N_O , the order of the orthogonal polynomial expansion, and N_p , the number of independent random variables [13]:

$$K = \binom{N_O + N_p}{N_p}. \quad (4)$$

A complete Askey scheme has been developed [14] to choose the set of orthogonal polynomials such that its weighting function is the probability density function of the set of independent random variables $\vec{\eta}$. For the particular case of Gaussian random variables,

the set of probabilistic Hermite polynomials $\Psi_i(\vec{\eta})$ is the standard choice, resulting into:

$$\begin{aligned} \mathbf{A}_i &= \langle \mathbf{A}(\mathbf{p}), \Psi_i(\vec{\eta}) \rangle \\ &= \int_{\eta_1} \cdots \int_{\eta_{N_p}} \mathbf{A}(\vec{\eta}) \Psi_i(\vec{\eta}) \frac{\exp(-\frac{\vec{\eta}^T \vec{\eta}}{2})}{(2\pi)^{\frac{N_p}{2}}} d\eta_1 \cdots d\eta_{N_p}, \\ &= \mathbb{E}[\mathbf{A}(\vec{\eta}) \Psi_i(\vec{\eta})] \end{aligned} \quad (5)$$

Despite the fact that the exponential function in (5) is separable, the fact that $\mathbf{A}(\mathbf{p})$ is dependent on all the independent random variables in vector $\vec{\eta}$ used to expand the random process, results in an integral (5) of very large dimension N_p independent of N_O . Such integrals are computationally impractical to evaluate. Several techniques have been proposed to avoid this inherent complexity, such as Monte Carlo integration (not to be confused with Monte Carlo simulation), and sparse grid integrations [15]. Nevertheless, the problem of calculating a large number of such integrals in a computationally efficient framework remains one of the bottlenecks of any stochastic algorithm that relies on the polynomial chaos expansion. In [16] a new method, relying on the inner product interpretation of the expectation, is proposed to overcome this inherent difficulty.

DEFINITION 2.1. Let the modified inner product $\langle \cdot, \cdot \rangle_{\mathcal{W}}$ be defined by

$$\langle f(\mathbf{p}), \Psi_i(\vec{\eta}) \rangle_{\mathcal{W}} = \int_{\mathbf{p}} \int_{\vec{\eta}} f(\mathbf{p}) \Psi_i(\vec{\eta}) \mathcal{W}(\mathbf{p}, \vec{\eta}) d\mathbf{p} d\vec{\eta}, \quad (7)$$

where

$$\mathcal{W}(\mathbf{p}, \vec{\eta}) = \frac{\exp(-0.5\mathbf{v}^T \mathbf{C}_V^{-1} \mathbf{v})}{(2\pi)^{\frac{D+N_O}{2}} \sqrt{|\mathbf{C}_V|}},$$

and \mathbf{C}_V is the correlation matrix between \mathbf{p} and $\vec{\eta}$ and

$$\mathbf{v} = \begin{pmatrix} \mathbf{p} \\ \vec{\eta} \end{pmatrix}, \quad \mathbf{C}_V = \mathbb{E}[\mathbf{v}\mathbf{v}^T] = \begin{pmatrix} \mathbb{E}[\mathbf{p}\mathbf{p}^T] & \mathbb{E}[\mathbf{p}\vec{\eta}^T] \\ \mathbb{E}[\vec{\eta}\mathbf{p}^T] & \mathbb{E}[\vec{\eta}\vec{\eta}^T] \end{pmatrix}$$

where \mathbb{E} is the expectation operator, $\vec{\eta}$ is an at most N_O element subvector of $\vec{\eta}$ upon which the N_O^{th} order Hermite polynomial $\Psi_{N_O}(\vec{\eta}) = \Psi_{N_O}(\vec{\eta})$ depends, and $|\mathbf{C}_V|$ is the determinant of the correlation matrix \mathbf{C}_V .

THEOREM 2.1. The Hermite polynomials $\Psi(\vec{\eta})$ are orthogonal with respect to the modified inner product (7).

THEOREM 2.2. The coefficients f_i of the standard Hermite expansion of the function $f(\mathbf{p})$ can be computed using the modified inner product:

$$f_i = \langle f(\mathbf{p}), \Psi_i(\vec{\eta}) \rangle = \langle f(\mathbf{p}), \Psi_i(\vec{\eta}) \rangle_{\mathcal{W}} \quad (8)$$

THEOREM 2.3. The maximum dimension of the integrals in (8) required to obtain the Hermite expansion using the modified inner product is $D + N_O$, where D is the length of vector \mathbf{p} and N_O is the order of the Hermite polynomial.

In other words, the dimension of the integral in (8) is independent of the number of orthogonal random variables N_p used for expanding the random process.

Using the modified inner product, one can then easily and very efficiently calculate the expansion coefficient \mathbf{A}_i in (3):

$$\mathbf{A}_i = \langle \mathbf{A}(\mathbf{p}), \Psi_i(\vec{\eta}) \rangle_{\mathcal{W}} = \int_{\mathbf{p}} \int_{\vec{\eta}} \mathbf{A}(\mathbf{p}) \Psi_i(\vec{\eta}) \mathcal{W}(\mathbf{p}, \vec{\eta}) d\mathbf{p} d\vec{\eta} \quad (9)$$

3. SAMPLING-BASED METHODS

In non-intrusive (i.e. sampling based) methods, the system (1) is solved N_S times, for different realizations of the random process \mathbf{p} , obtaining the set of solutions $\{\mathbf{x}(\mathbf{p}_1), \mathbf{x}(\mathbf{p}_2), \dots, \mathbf{x}(\mathbf{p}_{N_S})\}$. The solution at the different sample points is then used to either directly compute the statistics of the output quantity or to construct a functional representation of the output by using standard multivariate Lagrange interpolation or polynomial chaos expansions. Examples of sampling based methods include the Monte Carlo method [17], and the stochastic collocation method (SCM) [18, 19].

The main advantage of sampling based methods is that they rely on using a standard deterministic solver in order to compute the solution at the different sample points. Consequently, such methods can exploit all the advanced field solver technologies developed over the last decades [20, 21, 22, 2, 23, 24, 25, 26, 27, 28, 7]. Since such sampling-based methods do not require the development of new field solver technologies, they are referred to as non-intrusive. Another advantage of sampling-based methods is that they are highly parallelizable and therefore they can exploit the great advances in parallel computing architectures. On the other hand, the main disadvantage of sampling-based methods is that a very large number of sample points is needed in order to accurately represent the solution.

There are two important research challenges associated with sampling based methods. The first challenge is associated with the development of efficient algorithms for sampling large dimensional stochastic spaces (i.e. large number of statistical parameters). A variety of methods have been proposed in literature to address this problem: quasi-Monte Carlo [29], quadrature using sparse grids [30, 31], quadrature using adaptive sparse grids [32] and sampling using greedy algorithms [33, 34].

The second challenge is associated with the development of efficient algorithms to simultaneously solve a large number of "similar" linear systems of equations, which result from sampling the stochastic linear system of equations. Different strategies have been proposed in the literature to address such challenge, such as parallelization, Krylov subspace recycling [35, 36], stochastic model order reduction [37, 38, 33, 39] and recycling of FRW [40] (whenever FRW applies). In the following subsections we will elaborate more on both the stochastic model order reduction based methods and the recycling of FRW.

3.1 Statistical Moment Preserving Model Order Reduction (SMOR)

In SMOR the different solutions obtained by solving the linear system at the different sample points in the stochastic space are used in order to build a projection matrix, which spans the subspace in which the solution lives.

$$\mathbf{x}(\vec{\eta}) = \mathbf{U}\mathbf{x}_r(\vec{\eta}) \quad (10)$$

The reduced order solution is then computed as $y_r(\vec{\eta}) = \mathbf{c}^T \mathbf{U}\mathbf{x}_r(\vec{\eta})$. In [39], the projection matrix \mathbf{U} is chosen such that the statistical moments of $y_r(\vec{\eta})$ match those of $y(\vec{\eta})$

$$\begin{aligned} \mathbb{E}[y(\vec{\eta})^k] &= \int y(\vec{\eta})^k \mathcal{P}(\vec{\eta}) d\vec{\eta} = \int (\mathbf{c}^T \mathbf{A}(\vec{\eta})^{-1} \mathbf{b})^k \mathcal{P}(\vec{\eta}) d\vec{\eta} \\ &\simeq \sum_{q=1}^{N_q} \alpha_q (\mathbf{c}^T \mathbf{A}(\vec{\eta}_q)^{-1} \mathbf{b})^k \mathcal{P}(\vec{\eta}_q) \end{aligned} \quad (11)$$

and/or the coefficients β_i of the PCE projection of the reduced out-

put $y_r(\bar{\eta})$ match those of $y(\bar{\eta})$

$$\begin{aligned} \beta_i &= \langle y(\bar{\eta}), \Psi_i(\bar{\eta}) \rangle = \int y(\bar{\eta}) \Psi_i(\bar{\eta}) \mathcal{P}(\bar{\eta}) d\bar{\eta} \\ &\simeq \sum_{q=1}^{N_q} \alpha_q \mathbf{c}^T \mathbf{A}(\bar{\eta}_q)^{-1} \mathbf{b} \Psi_i(\bar{\eta}_q) \mathcal{P}(\bar{\eta}_q) \end{aligned} \quad (12)$$

Notice that α_q are the weights associated with the quadrature points $\bar{\eta}_q$ and N_q is the total number of quadrature points required by the numerical quadrature scheme used to calculate the integral (N_q increases as k increases).

Theorem 3.1 below provides a way to achieve both objectives, i.e. matching of the statistical moments, and matching of the coefficients for a multivariate Hermite representation of the output.

THEOREM 3.1. *IF \mathbf{U} is an orthonormal projection matrix and*

$$\text{colspan}\{\mathbf{U}\} \supset \text{span}\left\{\mathbf{b}, \mathbf{A}(\bar{\eta}_1)^{-1}\mathbf{b}, \mathbf{A}(\bar{\eta}_2)^{-1}\mathbf{b}, \dots, \mathbf{A}(\bar{\eta}_{N_q})^{-1}\mathbf{b}\right\}$$

where $\{\bar{\eta}_q : q = 1, \dots, N_q\}$ is a set of quadrature points such that (11) and (12) are exactly integrable for some choice of orthogonal polynomials $\Psi_i(\bar{\eta})$ THEN

$$\mathbb{E}\left[y(\bar{\eta})^k\right] = \mathbb{E}\left[y_r(\bar{\eta})^k\right] \quad (13)$$

and

$$\langle y(\bar{\eta}), \Psi_i(\bar{\eta}) \rangle = \langle y_r(\bar{\eta}), \Psi_i(\bar{\eta}) \rangle \quad (14)$$

where the reduced output $y_r(\bar{\eta})$ is given by

$$y_r(\bar{\eta}) = \mathbf{c}^T \mathbf{U}(\mathbf{U}^H \mathbf{A}(\bar{\eta}) \mathbf{U})^{-1} \mathbf{U}^H \mathbf{b} \quad (15)$$

In terms of actual implementation of the algorithm, stacking the solutions at all quadrature points in one projection matrix \mathbf{U} would result in three different issues, namely, it would require a very large number of linear system solves and would result in a large-dimensional and ill-conditioned projection matrix. The standard solution for the second and third issues (which does not address the first one) is to compress the projection subspace using a singular value decomposition (SVD). Instead in SMOR an alternative solution is presented that addresses *all* three issues at the same time. Specifically, the projection matrix is computed sequentially by introducing the solutions at the different quadrature points one at a time and updating the projection matrix only whenever necessary. In more detail, before solving a particular large system at a new quadrature point $\bar{\eta}_q$, the algorithm tests if the current projection matrix accurately represents the solution by computing the following residual

$$\mathbf{r}(\bar{\eta}_q) = \mathbf{b} - \mathbf{A}(\bar{\eta}_q) \mathbf{x}_r(\bar{\eta}_q) = \mathbf{b} - \mathbf{A}(\bar{\eta}_q) \mathbf{U} \left(\mathbf{U}^H \mathbf{A}(\bar{\eta}_q) \mathbf{U} \right)^{-1} \mathbf{U}^H \mathbf{b}$$

If the norm of the residual is large, then the current subspace does not faithfully represent the solution and therefore needs to be expanded with the solution of the new system orthogonalized with respect to the current subspace. If, on the other hand, the norm is small then the solution at the new sampling point is accurately represented using the currently explored subspace and does not need to be added to the basis. The advantage of using such a proximity measure is that only *one* matrix-vector product is done in the original space $O(N^2)$ and all the rest of the computation, i.e. computing $\mathbf{x}_r(\bar{\eta})$, is done in the reduced space, and is therefore very efficient. The complete SMOR is summarized in Algorithm 1. Notice that a similar residual measure was used in [34] to determine the points at which the parameter space should be sampled.

Algorithm 1 Stochastic Model Reduction Method for Solving Linear Systems with Random Matrices.

```

1:  $\mathbf{U} \leftarrow \mathbf{b}$ 
2:  $q \leftarrow 0$ 
3: for each quadrature point  $\bar{\eta}_q$  do
4:    $q \leftarrow q + 1$ 
5:   generate linear system  $\mathbf{A}(\bar{\eta}_q)$ .
6:   compute  $\mathbf{x}_r(\bar{\eta}_q) \leftarrow \mathbf{U} \left( \mathbf{U}^H \mathbf{A}(\bar{\eta}_q) \mathbf{U} \right)^{-1} \mathbf{U}^H \mathbf{b}$ 
7:   compute residual  $\mathbf{r}(\bar{\eta}_q) \leftarrow \mathbf{b} - \mathbf{A}(\bar{\eta}_q) \mathbf{x}_r(\bar{\eta}_q)$ .
8:   if  $\frac{\|\mathbf{r}(\bar{\eta}_q)\|}{\|\mathbf{b}\|} > \text{threshold}$  then
9:     solve for  $\mathbf{x}(\bar{\eta}_q)$ ,  $\mathbf{A}(\bar{\eta}_q) \mathbf{x}(\bar{\eta}_q) = \mathbf{b}$ 
10:    extend the basis  $\mathbf{U}$  with  $\mathbf{x}(\bar{\eta}_q) - \mathbf{U} \mathbf{U}^H \mathbf{x}(\bar{\eta}_q)$ 
11:  end if
12: end for

```

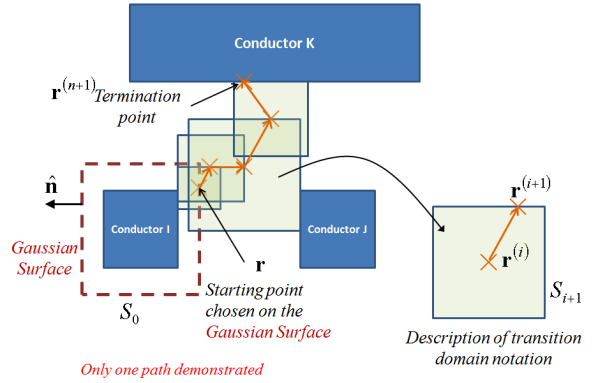


Figure 3: Typical random walk path from conductor I to conductor K.

3.2 Recycling of FRW

The floating random walk algorithm is a discretization-free algorithm for solving the Laplace equation. Consequently, such an algorithm is only suitable for capacitance and resistance extraction. In such methods the desired output is computed from the average contribution of a large number random walks which all start from the target conductor and end at the source conductor. Each walk is associated with a random variable, which is derived and computed from the governing PDEs. The walks are constructed from steps of nonuniform size and hence the name floating. Each step is taken by constructing a transition domain, which is centered at the most recent spatial location reached by the path and which extends until touching the nearest conductor (the domain is not allowed to enclose any conductors). The probability density function necessary for making a transition from the center point of the domain to any point on the boundary of the domain is computed from the Cube's Green's function. Each walk continues until terminating at a conductor surface (see figure 3). For more details on the FRW method and the most recent advances of the method, the reader is referred to [7, 41, 42, 43, 40, 44].

One of the main advantages of the floating random walk algorithm is the fact that the algorithm is inherently local, since each floating random walk path typically explores only a small part of the geometry. Such locality implies that the number of conductors which constrain a particular path is very small compared to the total number of conductors in the structure. This property is even more emphasized in typical VLSI structures with a large density of conductors, where the paths are typically very short and confined.

The proposed path recycling algorithm relies on solving the stochastic PDE using sampling-based methods. As will be clear in the following discussion, the proposed algorithm works best for sparse grid based sampling schemes, in which each sample point in the stochastic space corresponds to perturbing only a small subset of the parameters. In the path recycling algorithm, the capacitance of a nominal configuration is first computed. All the paths generated while solving such nominal configuration are stored. For any other configuration, which is generated by perturbing some of the geometrical parameters of the nominal configuration, the locality implies that many of the paths generated during the solution of the nominal configuration do not depend on the perturbed conductors. Consequently, such paths can be completely reused to compute the capacitance of the new configuration.

Even the paths that depend on a perturbed conductor can be partially reused in the new configuration. Specifically one can truncate them and complete them by resuming the walk from the first non-reusable transition domain affected by the change in configuration. Since the number of paths that require updating is very small compared to the total number of paths, the path recycling FRW algorithm will obtain almost instantly the solution for the new configuration. The complete procedure is formalized in Algorithm 2.

Notice that all the paths associated with the capacitance extraction of a given configuration need to be statistically independent. However, different configurations do not require to have statistically independent paths. Moreover, in some special cases, such as sensitivity analysis, where the output depends on the difference between the capacitances of two different configurations, sharing the same paths is not only an advantage but even a necessity to ensure numerical accuracy ([40]).

Algorithm 2 Path Recycling

```

1: while not all configurations tagged do
2:   Tag one of the untagged configurations
3:   repeat
4:     Generate a path for the last tagged configuration
5:     for each untagged configuration  $k$  do
6:       Map current path to configuration  $k$ 
7:       if not valid in configuration  $k$  then
8:         truncate and complete path
9:       end if
10:      Use path to update capacitance matrix of configuration  $k$ 
11:    end for
12:  until convergence of the last tagged configuration
13: end while

```

4. INTRUSIVE EXPANSION-BASED METHODS

As implied by the name, in expansion based methods one typically seeks to compute a stochastic expansion of the unknown. Examples of such expansions are the Neumann expansion and the polynomial chaos expansion. Specialized solvers are needed in order to either compute desired statistical moments from such expansions or the coefficients of such expansions. Hence such methods are sometimes referred to as intrusive methods. In this section we review four different intrusive methods. The first of them is based on the Neumann expansion. The last three methods are all based on the polynomial chaos expansion (PCE). The three methods are organized as follows. We first start by the standard method to compute the PCE coefficients which is the stochastic Galerkin

method. We then present a more efficient method to compute the coefficients in cases which are characterized by small variations. We finally present the stochastic dominant singular vectors method to efficiently compute the coefficients of the PCE for any general problem and which is very efficient for cases of large variations.

4.1 Neumann Expansion

The stochastic system matrix $\mathbf{A}(\mathbf{p})$ in (1) is first written as $\mathbf{A}(\mathbf{p}) = \mathbf{A}_0 + \Delta_{\mathbf{A}}(\mathbf{p})$, a sum of a deterministic expectation matrix $\mathbf{A}_0 = \mathbb{E}[\mathbf{A}(\mathbf{p})]$ and a stochastic matrix $\Delta_{\mathbf{A}}(\mathbf{p})$. Second, $\mathbf{A}(\mathbf{p})$ is substituted in the linear system (1), and the inverse is expanded using the Neumann expansion to obtain:

$$\mathbf{x}(\mathbf{p}) = \mathbf{x}_0 - \mathbf{A}_0^{-1} \Delta_{\mathbf{A}}(\mathbf{p}) \mathbf{x}_0 + \mathbf{A}_0^{-1} \Delta_{\mathbf{A}}(\mathbf{p}) \mathbf{A}_0^{-1} \Delta_{\mathbf{A}}(\mathbf{p}) \mathbf{x}_0 + \dots \quad (16)$$

where $\mathbf{x}_0 = \mathbf{A}_0^{-1} \mathbf{b}$. The above series converges provided that

$$\max |\lambda(\mathbf{A}_0^{-1} \Delta_{\mathbf{A}}(\mathbf{p}))| < 1. \quad (17)$$

The required statistics of the unknown current vector $\mathbf{x}(\mathbf{p})$ can then be obtained from (16), however, such computation (even just the average) is computationally very expensive [45, 46]. The main complexity is associated with the term $\mathbb{E}[\Delta_{\mathbf{A}}(\mathbf{p}) \mathbf{A}_0^{-1} \Delta_{\mathbf{A}}(\mathbf{p})]$, which involves computing the expectation of the product of two stochastic matrices separated by a deterministic matrix. This term can be rewritten using the Kronecker product \otimes and the vector operator $\text{vec}(\cdot)$ as ¹:

$$\mathbb{E}[\Delta_{\mathbf{A}}(\mathbf{p}) \mathbf{A}_0^{-1} \Delta_{\mathbf{A}}(\mathbf{p})] = \text{vec}^{-1} \left(\mathbb{E}[\Delta_{\mathbf{A}}(\mathbf{p})^T \otimes \Delta_{\mathbf{A}}(\mathbf{p})] \text{vec}(\mathbf{A}_0^{-1}) \right) \quad (18)$$

The complexity of storing and evaluating (18) is $O(N^4)$, which is computationally prohibitive. The complexity of computing the statistics increases even more dramatically if the order of the term in the Neumann expansion is increased and/or if we compute high order statistical moments.

A couple of techniques have been proposed to reduce the complexity of computing (18) [47, 48]. In particular, the variation-aware capacitance extraction algorithm in [47] relies on using the H-matrix [23] to sparsify both \mathbf{A}_0^{-1} and $\Delta_{\mathbf{A}}(\mathbf{p})^T \otimes \Delta_{\mathbf{A}}(\mathbf{p})$. The final complexity of the algorithm is $O(N \log^2 N)$. On the other hand, the variation-aware resistance/inductance extraction algorithm in [48] relies on using “stochastic” high order basis functions in order to reduce N (without reducing the complexity of the problem $O(N^4)$). Both [47, 48] are specialized for very particular applications. Moreover, they are useful only for computing the average and possibly a low order approximation of the variance. It is very hard to generalize such algorithms to general or more complex applications. Furthermore, computing high order statistics using such a method is for all practical purposes computationally impossible.

4.2 Stochastic Galerkin Method

The stochastic Galerkin method (SGM) has been developed by Ghanem [45]. It has been traditionally called the stochastic finite element method. In SGM the objective is to compute the coefficients of the PCE of the solution. This is achieved by first expanding the system matrix $\mathbf{A}(\mathbf{p})$ using polynomial chaos (5). The unknown $\mathbf{x}(\vec{\eta})$ is then written as an expansion of the same orthogonal polynomials $\mathbf{x}(\vec{\eta}) = \sum_{j=0}^K \mathbf{x}_j \Psi_j(\vec{\eta})$, where \mathbf{x}_j are unknown vectors.

¹When the vector operator is applied to a matrix \mathbf{A} of dimension $N \times N$ the result is a vector of size $N^2 \times 1$ obtained by stacking the columns of the matrix into a large vector. Using Matlab’s notation $\text{vec}(\mathbf{A}) = \mathbf{A}(:)$

Both expansions are then substituted in (1)

$$\sum_{i=0}^K \mathbf{A}_i \Psi_i(\bar{\boldsymbol{\eta}}) \sum_{j=0}^K \mathbf{x}_j \Psi_j(\bar{\boldsymbol{\eta}}) = \mathbf{b}. \quad (19)$$

A Galerkin testing, i.e. projection on the space of the same set of orthogonal polynomials, is applied to obtain a linear system of equations.

$$\sum_{i=0}^K \sum_{j=0}^K \langle \Psi_i(\bar{\boldsymbol{\eta}}) \Psi_j(\bar{\boldsymbol{\eta}}), \Psi_\ell(\bar{\boldsymbol{\eta}}) \rangle \mathbf{A}_i \mathbf{x}_j = \langle \mathbf{b}, \Psi_\ell(\bar{\boldsymbol{\eta}}) \rangle \quad \forall \ell \in \{1, \dots, K\} \quad (20)$$

Equation (20) is equivalent to a linear system of the form

$$\tilde{\mathbf{A}} \tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \quad (21)$$

where

$$\tilde{\mathbf{A}} = \sum_{i=0}^K \left(\begin{pmatrix} \gamma_{i00} & \gamma_{i10} & \cdots & \gamma_{iK0} \\ \gamma_{i01} & \gamma_{i11} & \cdots & \gamma_{iK1} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{i0K} & \gamma_{i1K} & \cdots & \gamma_{iKK} \end{pmatrix} \otimes \mathbf{A}_i \right)$$

$$\tilde{\mathbf{x}} = (\mathbf{x}_0^T \quad \mathbf{x}_1^T \quad \cdots \quad \mathbf{x}_K^T)^T, \quad \tilde{\mathbf{b}} = (\mathbf{b}^T \quad 0 \quad \cdots \quad 0)^T,$$

$\gamma_{ij\ell} = \langle \Psi_i(\bar{\boldsymbol{\eta}}) \Psi_j(\bar{\boldsymbol{\eta}}), \Psi_\ell(\bar{\boldsymbol{\eta}}) \rangle$ and \otimes is the Kronecker product. The size of this linear system is $O(NK)$, i.e. the original system size N , times K the total number of multivariate orthogonal polynomials used to expand the random function. Notice that for a moderate size 100-dimensional parameter space, K is 5,000. Solving the corresponding linear system would require $O(K^2 N^2)$ time complexity when using an iterative solver, and $O(K^3 N^3)$ time complexity when using a direct solver. Such complexity would result in an extremely inefficient electrical extraction algorithm, probably much worse than Monte Carlo simulation.

4.3 Combined Neumann Hermite Expansion (CNHE)

In the CNHE the coefficients of the PCE of the solution are computed very efficiently by starting from the Neumann expansion (instead of by solving the very large and complex equation 21). The approach in this section is ideal for small variations which satisfy the convergence properties of the Neumann expansion. An example of problems for which the CNHE is very efficient is impedance extraction in the presence of moderate surface roughness.

In the CNHE the Neumann expansion (16) is used to avoid the complexity of solving a large linear system, and the modified polynomial chaos expansion is used to simplify the calculation of the statistics of the unknown vector. This combined approach is implemented by first expanding $\mathbf{A}(\mathbf{p})$ in terms of Hermite polynomials using the modified inner product (7)

$$\mathbf{A}(\mathbf{p}) = \mathbf{A}_0 + \sum_{i=1}^K \mathbf{A}_i \Psi_i(\bar{\boldsymbol{\eta}}) \quad (22)$$

and then substituting this expansion in the Neumann expansion (16):

$$\mathbf{x}(\bar{\boldsymbol{\eta}}) = \mathbf{A}_0^{-1} \mathbf{b} - \mathbf{A}_0^{-1} \left(\sum_{i=1}^K \mathbf{A}_i \Psi_i(\bar{\boldsymbol{\eta}}) \right) \mathbf{A}_0^{-1} \mathbf{b} + \quad (23)$$

$$+ \mathbf{A}_0^{-1} \left(\sum_{i=1}^K \mathbf{A}_i \Psi_i(\bar{\boldsymbol{\eta}}) \right) \mathbf{A}_0^{-1} \left(\sum_{i=1}^K \mathbf{A}_i \Psi_i(\bar{\boldsymbol{\eta}}) \right) \mathbf{A}_0^{-1} \mathbf{b} + \quad (24)$$

From (24) it is observed that the problem has been transformed from solving a huge linear system into doing instead a large number of small matrix-vector multiplications, or small linear system solves. Using the output equation, equation (24) is simplified to:

$$y(\bar{\boldsymbol{\eta}}) \simeq y_0 - \sum_{i=1}^K \mathbf{z}_0^T \mathbf{u}_i \Psi_i(\bar{\boldsymbol{\eta}}) + \sum_{i=1}^K \sum_{j=1}^K \mathbf{v}_i^T \mathbf{t}_j \Psi_i(\bar{\boldsymbol{\eta}}) \Psi_j(\bar{\boldsymbol{\eta}}), \quad (25)$$

where $y_0 = \mathbf{c}^T \mathbf{x}_0$, $\mathbf{x}_0 = \mathbf{A}_0^{-1} \mathbf{b}$, $\mathbf{z}_0 = \mathbf{A}_0^{-T} \mathbf{c}$, $\mathbf{u}_i = \mathbf{A}_i \mathbf{x}_0$, $\mathbf{v}_i = \mathbf{A}_i^T \mathbf{z}_0$, and $\mathbf{t}_j = \mathbf{A}_0^{-1} \mathbf{u}_j$. Notice that the complexity of evaluating (25) is significantly less than that of evaluating (24) due to the use of the adjoint equations to compute \mathbf{z}_0 and \mathbf{v}_i .

Algorithm 3 Combined Neumann Hermite Expansion (CNHE)

- 1: Compute the coefficient matrices \mathbf{A}_i using (9) as in Theorem 2.2.
 - 2: Solve the nominal $\mathbf{A}_0 \mathbf{x}_0 = \mathbf{b}$ and adjoint problems $\mathbf{A}_0^T \mathbf{z}_0 = \mathbf{c}$ for \mathbf{x}_0 and \mathbf{z}_0 , respectively.
 - 3: **for** each coefficient matrix \mathbf{A}_i **do**
 - 4: $\mathbf{u}_i \leftarrow \mathbf{A}_i \mathbf{x}_0$
 - 5: $\mathbf{v}_i \leftarrow \mathbf{A}_i^T \mathbf{z}_0$
 - 6: Solve $\mathbf{A}_0 \mathbf{t}_j = \mathbf{u}_j$ for \mathbf{t}_j
 - 7: **end for**
 - 8: Use (25) to assemble the second order expansion for the required output $y(\boldsymbol{\eta})$ (for instance the input current).
-

4.4 Stochastic Dominant Singular Vectors Method (SDSV)

For problems which are described by large dimensional high-order PCE, computing the coefficients of the PCE either using the standard SGM or even by using the CNHE is a computationally very expensive task. The fundamental problem with the PCE is that it assumes that all the coefficients of the expansion are equally important and that all of them need to be computed. However, in many practical cases the PCE expansion can be significantly compressed by using problem-specific polynomial basis $d_i(\bar{\boldsymbol{\eta}})$ which are typically computed as linear combinations of the standard polynomial basis.

$$\mathbf{x}(\bar{\boldsymbol{\eta}}) = \sum_{i=0}^K \mathbf{x}_i \Psi_i(\bar{\boldsymbol{\eta}}) = \sum_{i=0}^K \mathbf{x}_i d_i(\bar{\boldsymbol{\eta}}) \quad (26)$$

Notice that the suggested representation is *nonlinear* of $\mathbf{x}(\bar{\boldsymbol{\eta}})$, since both the deterministic component as well as the stochastic component are unknown. To better understand the idea behind our work let us consider expressing $\mathbf{x}(\bar{\boldsymbol{\eta}})$ in terms of its dominant basis:

$$\mathbf{x}(\bar{\boldsymbol{\eta}}) = \sum_{i=0}^K \mathbf{x}_i \Psi_i(\bar{\boldsymbol{\eta}}) = \mathbf{X} \mathbf{h}(\bar{\boldsymbol{\eta}}) = \sum_{i=0}^r \sigma_i \mathbf{u}_i \tilde{\mathbf{v}}_i^T \mathbf{h}(\bar{\boldsymbol{\eta}}) = \sum_{i=0}^r \mathbf{u}_i \mathbf{v}_i^T \mathbf{h}(\bar{\boldsymbol{\eta}}). \quad (27)$$

where $\sum_{i=0}^r \sigma_i \mathbf{u}_i \tilde{\mathbf{v}}_i^T$ is the SVD of \mathbf{X} , $\sigma_i \tilde{\mathbf{v}}_i = \mathbf{v}_i$ and r is the total number of dominant basis and $\mathbf{h}(\bar{\boldsymbol{\eta}})$ is a vector of the Hermite orthogonal polynomials:

$$\mathbf{h}(\bar{\boldsymbol{\eta}}) = (\Psi_0(\bar{\boldsymbol{\eta}}) \quad \Psi_1(\bar{\boldsymbol{\eta}}) \quad \cdots \quad \Psi_K(\bar{\boldsymbol{\eta}}))^T. \quad (28)$$

Note that $\mathbf{v}_i^T \mathbf{h}(\bar{\boldsymbol{\eta}})$ is a scalar polynomial and that (27) can therefore be expressed as

$$\mathbf{x}(\bar{\boldsymbol{\eta}}) = \sum_{i=0}^r \mathbf{v}_i^T \mathbf{h}(\bar{\boldsymbol{\eta}}) \mathbf{u}_i \quad (29)$$

where \mathbf{u}_i is an unknown vector of length N and \mathbf{v}_i is an unknown vector of length K representing a direction in the stochastic space.

The Key Idea. The solution vector $\mathbf{x}(\bar{\eta})$ is decomposed in the form of the summation (29) and its components are sequentially computed, i.e. at every iteration n of the algorithm one solves only for \mathbf{u}_n and \mathbf{v}_n . These vectors are computed such that they minimize the norm of the residual at iteration n .

This is achieved by first substituting (29) in (1) to obtain

$$\mathbf{A}(\bar{\eta}) \sum_{i=0}^r \mathbf{v}_i^T \mathbf{h}(\bar{\eta}) \mathbf{u}_i = \mathbf{b}(\bar{\eta}) \quad (30)$$

Assume that at step n of the algorithm, all vectors $\mathbf{u}_i, \mathbf{v}_i : i = 0, \dots, n-1$ are known and define

$$\mathbf{x}_n(\bar{\eta}) = \sum_{i=0}^n \mathbf{v}_i^T \mathbf{h}(\bar{\eta}) \mathbf{u}_i = \mathbf{x}_{n-1}(\bar{\eta}) + \mathbf{v}_n^T \mathbf{h}(\bar{\eta}) \mathbf{u}_n \quad (31)$$

and

$$\mathbf{r}_n(\bar{\eta}) = \mathbf{b}(\bar{\eta}) - \mathbf{A}(\bar{\eta}) \mathbf{x}_n(\bar{\eta}) \quad (32)$$

Equation (32) can be put in a recursive form by using (31)

$$\begin{aligned} \mathbf{r}_n(\bar{\eta}) &= \mathbf{b}(\bar{\eta}) - \mathbf{A}(\bar{\eta}) \mathbf{x}_{n-1}(\bar{\eta}) - \mathbf{A}(\bar{\eta}) \mathbf{v}_n^T \mathbf{h}(\bar{\eta}) \mathbf{u}_n \\ &= \mathbf{r}_{n-1}(\bar{\eta}) - \mathbf{A}(\bar{\eta}) \mathbf{v}_n^T \mathbf{h}(\bar{\eta}) \mathbf{u}_n \end{aligned} \quad (33)$$

where $\mathbf{x}_0 = \mathbf{0}$ and $\mathbf{r}_0 = \mathbf{b}$.

As mentioned above, at step n of the algorithm, \mathbf{u}_n and \mathbf{v}_n are computed such that the stochastic norm of the residual $\mathbf{r}_n(\bar{\eta})$ is minimized

$$\min_{\mathbf{u}_n, \mathbf{v}_n} \mathbb{E} \left[\mathbf{r}_n(\bar{\eta}) \mathbf{r}_n(\bar{\eta})^T \right] \quad (34)$$

One approach to minimizing (34) is to set the gradient \mathbf{f}' of the objective function to zero using Newton's method. The implementation details are omitted, however, for more details the reader is referred to [49]. Notice that following the computation of the n th pair of dominant vectors, the norm of the residual is updated and is used as an indicator of the accuracy of the solution. In other words, the search for pairs of dominant vectors is repeated sequentially until the norm of the residual becomes smaller than a given threshold.

The main advantage of the SDSV is that the number of the free optimization parameters (length of both \mathbf{u}_n and \mathbf{v}_n) is $O(N+K)$, which means that at every iteration one only needs to solve a linear system of the same size for a complexity $O(N+K)^2$. Such minimization is performed only r times, where r is the number of dominant basis of $\mathbf{x}(\bar{\eta})$. Note, typically $r \ll K \simeq N$. Consequently, the complexity of SDSV scales with just $O(N^2)$, practically independent of the number of parameters.

Algorithm 4 presents a complete summary of SDSV. Note that both unknowns \mathbf{u}_n and \mathbf{v}_n are combined into a single vector \mathbf{z} .

5. RESULTS

In this section we compare the performance of the following state-of-the-art stochastic simulations methods: Monte Carlo method (MC), stochastic collocation method (SCM), stochastic model order reduction method (SMOR), path recycling via FRW, standard Neumann expansion (NE), stochastic Galerkin method (SGM), combined Neumann Hermite expansion (CNHE), and the stochastic dominant singular vector (SDSV). All algorithms are tested on the following four examples:

1. Large 3D On-Chip capacitance extraction example in the presence of width and thickness variations. This example is composed of 6 conductors arranged on top of 289 small metal

Algorithm 4 The Stochastic Dominant Singular Vectors Method (SDSV)

```

1:  $\mathbf{x}(\bar{\eta}) \leftarrow \mathbf{0}, \mathbf{r}(\bar{\eta}) \leftarrow \mathbf{b}(\bar{\eta})$ 
2:  $\mathbf{u}_n \leftarrow \mathbf{x}_0$ , the solution of the nominal problem
3:  $\mathbf{v}_n \leftarrow \mathbf{e}_1$ 
4:  $\mathbf{z} \leftarrow (\mathbf{u}_n^T \mathbf{v}_n^T)^T$ 
5: while  $\|\mathbf{r}(\bar{\eta})\|_S^2 > \text{Threshold}$  do
6:   repeat
7:     form first derivative  $\mathbf{f}'$ 
8:     form Hessian  $\mathbf{f}''$ 
9:     solve linear system  $\mathbf{f}'' \Delta \mathbf{z} = -\mathbf{f}'$ 
10:     $\mathbf{z} \leftarrow \mathbf{z} + \Delta \mathbf{z}$ 
11:     $\mathbf{u}_n \leftarrow \mathbf{z}(1:N, 1), \mathbf{v}_n \leftarrow \mathbf{z}(N+1:N+K, 1)$ 
12:    until  $\|\mathbf{f}'\| < \text{Threshold}$ 
13:     $\mathbf{x}(\bar{\eta}) \leftarrow \mathbf{x}(\bar{\eta}) + \mathbf{u}_n \mathbf{v}_n^T \mathbf{h}(\bar{\eta})$ .
14:     $\mathbf{r}(\bar{\eta}) \leftarrow \mathbf{r}(\bar{\eta}) - \mathbf{A}(\bar{\eta}) \mathbf{u}_n \mathbf{v}_n^T \mathbf{h}(\bar{\eta})$ .
15:     $\mathbf{u}_n \leftarrow \mathbf{x}_0$ 
16:     $\mathbf{v}_n \leftarrow \mathbf{e}_1$ 
17:     $\mathbf{z} \leftarrow (\mathbf{u}_n^T \mathbf{v}_n^T)^T$ 
18: end while

```

fill as shown in Figure 4. The red conductor (in the middle of the second layer) is the target conductor for which we want to compute the capacitance vector. The total number of discretization elements is $N=25,788$ and the total number of parameters is $N_P=295$.

2. Large 3D parallel-plate capacitance extraction example in the presence of surface roughness. This example is composed of 2 square plate conductors of size 20 separated by a distance of 1 unit length (see Figure 5). The upper plate is very rough ($\sigma = 0.2$ and $L_c = 1$). The total number of discretization elements is $N=21,000$ and the total number of parameters is $N_P=323$.
3. Large 3D impedance extraction in the presence of random geometrical variations. This example (shown in Figure 6) is composed of four different inductors. The dimensions of every wire (total of 48) is an independent random variable. For the discretization based methods, the total number of discretization elements is $N=10,358$ and the total number of parameters is $N_P = 48$.
4. Large 3D impedance extraction in the presence of random surface roughness. This example (shown in Figure 6) is composed of four different inductors. The upper surface of each conductor is assumed rough. For the discretization based methods, the total number of discretization elements is $N=10,358$ and the total number of parameters is $N_P = 400$.

All algorithms are optimized for computational efficiency. Whenever applicable, we exploit parallelization, computational reuse (e.g. table look-ups), numerical approximations, data compression and any other available implementation tricks (provided the accuracy threshold is met). Table 1 summarizes the comparison results. Notice that we are reporting two different times corresponding to 3% and 8% accuracy thresholds. Notice further that the entries in the table are rounded to simplify comparing the results and that all numbers in brackets are estimated. Finally notice that the SCM is of second order accuracy since the first order scheme produces error more than 8%.

From the table we can draw several conclusions related to the relative performance of the various methods:

Table 1: Comparison of the performance of the Monte Carlo Method (MC), stochastic collocation method (SCM), SMOR, FRW, Neumann expansion method (NE), SGM, CNHE, SDSV, for four different examples. N is the total number of discretization elements. P/K are respectively the total number of parameters and the total number of orthogonal polynomials (dimension of the basis) used to describe the variability. The percentage beside the Time (e.g. 3% and 8%) refers to the accuracy threshold. Reported times are all in hours. All numbers in brackets are estimated.

		MC	SCM	SMOR	FRW	NE	SGM	CNHE	SDSV
Ex.1: 3D-cap. param. var. N=25K, P/K=295/44K	Time 3%	(600)	(6,000)	(24)	1	(13,000)	(360,000)	(1200)	8
	Time 8%	60	(6,000)	12	0.25	(13,000)	(360,000)	12	6
	Memory	20GB	20GB	20GB	5MB	20GB	20GB	20GB	20GB
Ex.2: 3D-cap. rough surface N=21K, P/K=323/53K	Time 3%	(650)	(6,500)	(70)	6	(12,000)	(500,000)	(260)	12
	Time 8%	65	(6,500)	14	2	(12,000)	(500,000)	13	7
	Memory	20GB	20GB	20GB	20MB	20GB	20GB	20GB	20GB
Ex.3: 3D-imp. param. var. N=10K, P/K=48/1K	Time 3%	10	10	7	–	(100)	(50)	6	8
	Time 8%	10	10	7	–	(100)	(50)	6	8
	Memory	10GB	10GB	10GB	–	10GB	10GB	10GB	10GB
Ex.4: 3D-imp. rough surface N=10K, P/K=400/81K	Time 3%	(1400)	(21,000)	(56)	–	(100)	(2,000,000)	(720)	12
	Time 8%	(140)	(21,000)	28	–	(100)	(2,000,000)	36	10
	Memory	16GB	16GB	16GB	–	16GB	16GB	16GB	16GB

1. The SCM, NE and SGM perform very poorly in all examples. Except for the performance of the NE in the fourth example, these methods are outperformed by all the other methods. This is primarily due to the fact that such methods are not adaptive and require very large computational effort before any meaningful results are obtained. For all examples, a second order scheme of such methods was both necessary to reduce the error below 8% and sufficient to reduce it below 3%. This explains why the performance of such methods does not change for the different error thresholds.
2. The FRW is the best method to extract the resistance or capacitance in the presence of variations in the metal (not in the material properties). Such methods scale extremely well with the size of the problem and with the dimension of the parameter space.
3. The SDSV is the best method to extract the impedance in the presence of large dimensional variations. Despite the need for a large setup time, such method scales extremely well with the dimension of the parameter space.
4. The CNHE is the best method to extract the impedance in the presence of small dimensional variations, provided that the output is accurately described using a low order multivariate expansion in the parameter space. Such method requires solving linear systems which only involve the average matrix. Furthermore, such method utilizes the adjoint concepts to significantly reduce the computational effort.
5. The SMOR is the best method to extract the impedance in the presence of small dimensional variations, provided that the output is a high order multivariate expansion in the parameter space. Such method fully exploits fast field solver techniques and therefore significantly reduces the solution time of the independent linear system solves. In addition, due to the non-intrusive nature of the SMOR it is the easiest presented method in terms of implementation cost and it can be very easily generalized to handle non-linear and dynamic problems.

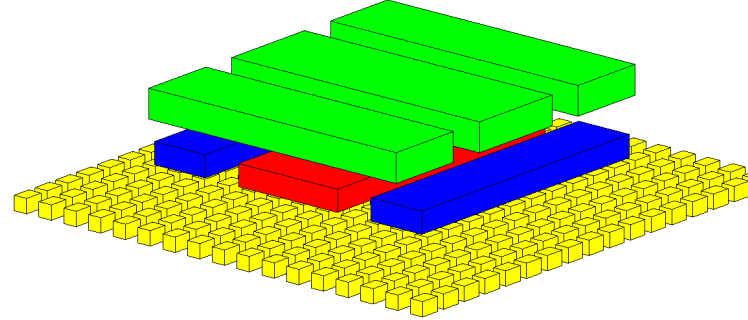


Figure 4: Large Capacitance Extraction Structure. Discretized using a total of 25788 different panels.

6. CONCLUSION

In this paper we have presented a large number of the state of the art algorithms which are suitable for the extraction of the electrical characteristics in the presence of manufacturing process variabilities. It has been observed that sampling based methods via FRW are best suited for stochastic capacitance extraction. It has also been observed that CNHE is best suited for general impedance extraction problems provided that the manufacturing variabilities are small, the SMOR is very best suitable for general stochastic impedance extraction provided that the stochastic space can be efficiently sampled. For complex problems characterized by large variations in large dimensional stochastic spaces the SDSV is the best technique. One of the important remaining research direction is the development of stochastic fast solvers such as stochastic fast precorrected-Fast Fourier transform or stochastic fast multipole method. Another important research direction is to complete the stochastic CAD flow by developing stochastic process simulation tools, stochastic device simulation tools and stochastic circuit simulators. To achieve such goal the stochastic algorithms need to be generalized to be both nonlinear and dynamic.

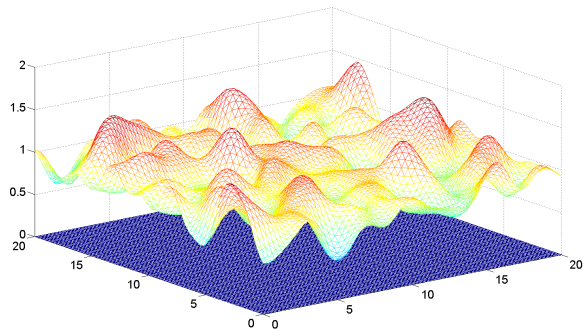


Figure 5: Large Parallel Plate Capacitance Extraction Structure. Discretized using a total of 21,000 different panels.

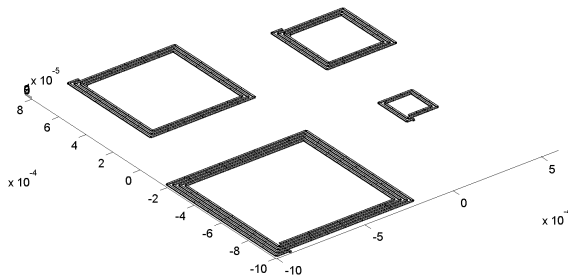


Figure 6: Array of 4 Inductors.

Acknowledgment

This work has been partially supported by Mentor Graphics, AMD, IBM, the Semiconductor Corporations and the Interconnect Focus Center, one of five research centers funded under the Focus Center Research Program, a DARPA and Semiconductor Research Corporation program.

7. REFERENCES

- [1] A. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 22, no. 3, pp. 216–221, Mar 1974.
- [2] M. Kamon, N. Marques, and J. White, "Fastpep: a fast parasitic extraction program for complex three-dimensional geometries," in *Computer-Aided Design, 1997. Digest of Technical Papers., 1997 IEEE/ACM International Conference on*, Nov 1997, pp. 456–460.
- [3] J. Gu, Z. Wang, and X. Hong, "Hierarchical computation of 3d interconnect capacitance using direct boundary element method," in *Design Automation Conference, 2000. Proceedings of the ASP-DAC 2000. Asia and South Pacific*, 2000, pp. 447–452.
- [4] L. J. Jiang, B. J. Rubin, J. D. Morsey, H. T. Hu, and A. Elfadel, "Novel capacitance extraction method using direct boundary integral equation method and hierarchical approach," in *Electrical Performance of Electronic Packaging, 2006 IEEE*, Oct. 2006, pp. 331–334.
- [5] J. Zhu and D. Jiao, "A unified finite-element solution from zero frequency to microwave frequencies for full-wave modeling of large-scale three-dimensional on-chip interconnect structures," *Advanced Packaging, IEEE Transactions on*, vol. 31, no. 4, pp. 873–881, Nov. 2008.
- [6] A. Haji-Sheikh and E. M. Sparrow, "The floating random walk and its application to monte carlo solutions of heat equations," *SIAM Journal on Applied Mathematics*, vol. 14, no. 2, pp. 370–389, 1966. [Online]. Available: <http://www.jstor.org/stable/2946271>
- [7] Y. L. Coz and R. Iverson, "A stochastic algorithm for high speed capacitance extraction in integrated circuits," *Solid-State Electronics*, vol. 35, no. 7, pp. 1005 – 1012, 1992. [Online]. Available: <http://www.sciencedirect.com/science/article/B6TY5-46VMC8J-MV/2/942a530ca48892f2bdc239be5ade5596>
- [8] S. S. K. Mandar K. Chati, Mircea D. Grigoriu and S. Mukherjee, "Random walk method for the two- and three-dimensional laplace, poisson and helmholtz's equations," *International Journal for Numerical Methods in Engineering*, vol. 51, no. 10, pp. 1133–1156, 2001.
- [9] K. Sabelfeld and N. Simonov, *Random Walks On Boundary For Solving Pdes.* VSP, 1994.
- [10] K. F. Warnick and W. C. Chew, "Numerical simulation methods for rough surface scattering," *Waves in Random Media*, vol. 11, 2001.
- [11] M. Loeve, *Probability theory II.* Springer, 1994.
- [12] N. Wiener, "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938. [Online]. Available: <http://www.jstor.org/stable/2371268>
- [13] D. W. Bolton, "The multinomial theorem," *The Mathematical Gazette*, vol. 52, no. 382, pp. 336–342, 1968. [Online]. Available: <http://www.jstor.org/stable/3611846>
- [14] D. Xiu and G. E. Karniadakis, "The wiener–askey polynomial chaos for stochastic differential equations," *SIAM J. Sci. Comput.*, vol. 24, no. 2, pp. 619–644, 2002.
- [15] T. Gerstner and M. Griebel, "Numerical integration using sparse grids," *NUMER. ALGORITHMS*, vol. 18, pp. 209–232, 1998.
- [16] T. El-Moselhy and L. Daniel, "Stochastic integral equation solver for efficient variation-aware interconnect extraction," in *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, June 2008, pp. 415–420.
- [17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953. [Online]. Available: <http://dx.doi.org/10.1063/1.1699114>
- [18] L. Mathelin and M. Y. Hussaini, "A stochastic collocation algorithm," *NASA*, pp. 2003–212 153, 2003.
- [19] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 1118–1139, 2005.
- [20] K. Nabors and J. White, "Fastcap: a multipole accelerated 3-d capacitance extraction program," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 10, no. 11, pp. 1447–1459, Nov 1991.
- [21] J. R. Phillips, J. K. White, and A. Member, "A precorrected-fft method for electrostatic analysis of complicated 3-d structures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, pp. 1059–1072, 1997.
- [22] M. Kamon, M. Tsuk, and J. White, "Fasthenry: a multipole-accelerated 3-d inductance extraction program," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 42, no. 9, pp. 1750–1758, Sept. 1994.

- [23] W. Hackbusch, "A sparse matrix arithmetic based on \mathcal{H} -matrices. part I: Introduction to \mathcal{H} -matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999. [Online]. Available: citeseer.ist.psu.edu/hackbusch99sparse.html
- [24] Z. Zhu, B. Song, and J. White, "Algorithms in fastimp: a fast and wideband impedance extraction program for complicated 3-d geometries," in *Design Automation Conference, 2003. Proceedings*, june 2003, pp. 712 – 717.
- [25] T. El-Moselhy, X. Hu, and L. Daniel, "pfft in fastmaxwell: A fast impedance extraction solver for 3d conductor structures over substrate," in *Design, Automation and Test in Europe Conference and Exhibition, 2007. DATE '07*, April 2007, pp. 1–6.
- [26] D. Jiao, J. Zhu, and S. Chakravarty, "A fast frequency-domain eigenvalue-based approach to full-wave modeling of large-scale three-dimensional on-chip interconnect structures," *Advanced Packaging, IEEE Transactions on*, vol. 31, no. 4, pp. 890–899, Nov. 2008.
- [27] S. Kapur and D. E. Long, "Ies3: Efficient electrostatic and electromagnetic simulation," *Computing in Science and Engineering*, vol. 5, no. 4, pp. 60–67, 1998.
- [28] T. Lu, Z. Wang, and W. Yu, "Hierarchical block boundary-element method (hbbem): a fast field solver for 3-d capacitance extraction," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 52, no. 1, pp. 10–19, Jan. 2004.
- [29] H. Niederreiter, "Quasi-monte carlo methods and pseudo-random numbers," *Bulletin of the American Mathematical Society*, vol. 84, no. 6, pp. 957–1041, 1978.
- [30] S. A. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Soviet Mathematics Doklady*, no. 4, pp. 240–243, 1963.
- [31] F. Heiss and V. Winschel, "Estimation with numerical integration on sparse grids," University of Munich, Department of Economics, Discussion Papers in Economics 916, Apr. 2006. [Online]. Available: <http://ideas.repec.org/p/lmu/muenec/916.html>
- [32] H.-J. Bungartz and S. Dirnstorfer, "Multivariate quadrature on adaptive sparse grids," *Computing*, vol. 71, pp. 89–114, 2003, 10.1007/s00607-003-0016-4. [Online]. Available: <http://dx.doi.org/10.1007/s00607-003-0016-4>
- [33] T. Bui-Thanh, K. Willcox, and O. Ghattas, "Model reduction for large-scale systems with high-dimensional parametric input space," *SIAM J. Sci. Comput.*, vol. 30, no. 6, pp. 3270–3288, 2008.
- [34] J. F. Villena and L. M. Silveira, "Arms - automatic residue-minimization based sampling for multi-point modeling techniques," in *DAC '09: Proceedings of the 46th Annual Design Automation Conference*. New York, NY, USA: ACM, 2009, pp. 951–956.
- [35] M. L. Parks, E. D. Sturler, G. Mackey, D. D. Johnson, and S. Maiti, "Recycling krylov subspaces for sequences of linear systems," *SIAM J. Sci. Comput*, Tech. Rep., 2004.
- [36] Z. Ye, Z. Zhu, and J. Phillips, "Generalized krylov recycling methods for solution of multiple related linear equation systems in electromagnetic analysis," in *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, June 2008, pp. 682–687.
- [37] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A. Patera, and G. Turinici, "Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bounds methods," *Journal of Fluids Engineering*, 2002.
- [38] J. Phillips, "Variational interconnect analysis via PMTBR," in *Proc. of IEEE/ACM International Conference on Computer Aided-Design*, Nov. 2004, pp. 872–9.
- [39] T. El-Moselhy and L. Daniel, "Variation-aware interconnect extraction using statistical moment preserving model order reduction," in *DATE '10: Proceedings of the conference on Design, automation and test in Europe*. San Jose, CA, USA: EDA Consortium, 2010.
- [40] T. El-Moselhy, I. Elfadel, and L. Daniel, "A capacitance solver for incremental variation-aware extraction," in *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, Nov. 2008, pp. 662–669.
- [41] J. Jere and Y. Le Coz, "An improved floating-random-walk algorithm for solving the multi-dielectric dirichlet problem," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 41, no. 2, pp. 325–329, Feb 1993.
- [42] P. Maffezzoni, "Analysis of substrate coupling by means of a stochastic method," *Electron Device Letters, IEEE*, vol. 23, no. 6, pp. 351–353, Jun 2002.
- [43] S. Batterywala, R. Ananthakrishna, Y. Luo, and A. Gyure, "A statistical method for fast and accurate capacitance extraction in the presence of floating dummy fills," *VLSI Design, International Conference on*, vol. 0, pp. 129–134, 2006.
- [44] T. El-Moselhy, I. Elfadel, and L. Daniel, "A hierarchical floating random walk algorithm for fabric-aware 3d capacitance extraction," in *Computer-Aided Design, 2009. ICCAD 2009. IEEE/ACM International Conference on*, Nov. 2009.
- [45] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*. Dover Publications, 2003.
- [46] Z. Zhu, J. White, and A. Demir, "A stochastic integral equation method for modeling the rough surface effect on interconnect capacitance," in *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 887–891.
- [47] Z. Zhu and J. White, "Fastsies: a fast stochastic integral equation solver for modeling the rough surface effect," in *Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference on*, Nov. 2005, pp. 675–682.
- [48] T. El-Moselhy and L. Daniel, "Stochastic high order basis functions for volume integral equation with surface roughness," in *Electrical Performance of Electronic Packaging, 2007 IEEE*, Oct. 2007, pp. 73–76.
- [49] —, "Stochastic dominant singular vectors method for variation-aware extraction," in *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, june 2010, pp. 667 –672.