A Capacitance Solver for Incremental Variation-Aware Extraction

Tarek A. El-Moselhy
Research Lab in Electronics
Massachusetts Institute of Technology
tmoselhy@mit.edu

Ibrahim M. Elfadel
Systems & Technology Group
IBM Corporation
elfadel@us.ibm.com

Luca Daniel
Research Lab in Electronics
Massachusetts Institute of Technology
luca@mit.edu

***Abstract*—Lithographic limitations and manufacturing uncertainties are resulting in fabricated shapes on wafer that are topologically equivalent, but geometrically different from the corresponding drawn shapes. While first-order sensitivity information can measure the change in pattern parasitics when the shape variations are small, there is still a need for a high-order algorithm that can extract parasitic variations incrementally in the presence of a large number of simultaneous shape variations. This paper proposes such an algorithm based on the well-known method of floating random walk (FRW). Specifically, we formalize the notion of random path sharing between several conductors undergoing shape perturbations and use it as a basis of a fast capacitance sensitivity extraction algorithm and a fast incremental variational capacitance extraction algorithm. The efficiency of these algorithms is further improved with a novel FRW method for dealing with layered media. Our numerical examples show a 10X speed up with respect to the boundary-element method adjoint or finite-difference sensitivity extraction, and more than 560X speed up with respect to a non-incremental FRW method for a high-order variational extraction.**

I. Introduction

Fast and efficient capacitance extraction is the cornerstone of integrated circuit electrical extraction. The many available techniques can be divided into deterministic techniques, such as the boundary element method or the finite difference method, and stochastic techniques such as the floating random walk. In general, deterministic techniques require a linear system solve. For large geometries the time of such solve dominates the computational complexity and therefore different acceleration techniques ("fast-solvers"), such as pre-corrected FFT [1], multipole expansion [2], and hierarchical techniques [3], have been proposed to speed up the system solves. However, the criteria predominantly used to compare and characterize the speed of such techniques is the time required per *one single* solve. Such criteria might not seem optimal if we consider using the solver to extract the capacitance of a large number of similar configurations, i.e. configurations that are topologically equivalent but geometrically different. In such a case the objective is to minimize the total simulation time required to solve *all configurations*. Indeed, such usage of the solver is very common. In particular, most variation-aware extraction flows exhibit such usage. This problem of efficiently solving a large number of similar configurations is encountered in the extraction of lithographic- and variation-aware layouts of the type shown in Fig. 1. This figure illustrates the wafer contours of the active areas of an SRAM cell fabricated in a 45*nm* technology. It is interesting that the highly irregular contours are present despite the use of lithographic improvement techniques such as optical pre-correction and resolution enhancement. While contour-aware extraction has already been proposed [4] in order to improve the accuracy of layout parasitics in the presence of lithographic irregularities, the highly irregular nature of these contours make such an approach very CPU-intensive. One possible alternative is to use edge bounding boxes bracketing the irregularities of each edge. The resulting shapes will be topologically similar yet geometrically each edge will have two states, and the number of resulting configurations will be exponential in the number of edges. Furthermore, lithographic wavelength and reduced feature sizes are conspiring to increase the radius of shape interactions thus increasing the size of patterns that have to be taken into account in the parasitic extraction flow.

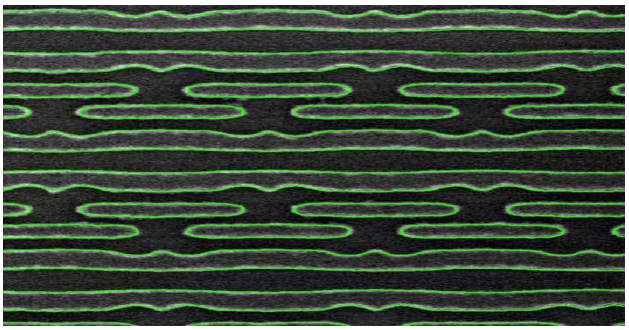

Fig. 1. Top view of the active areas of an SRAM cell fabricated in a 45nm technology

While lithography mainly impacts the layout shapes in the mask planes, chemical-mechanical polishing (CMP) contributes to the uncertainties in interconnect heights perpendicular to the masks, thus increasing even further the number of configurations that need to be considered in variation-aware capacitance extraction.

Other applications that require the solution of a very large number of "similar" configurations are for instance: the generation of capacitance tables used in macro and full-chip parasitic layout extraction [5], [6]; the extraction of capacitance distributions using sampling based techniques as

978-1-4244-2820-5/08/$25.00 ©2008 IEEE

is found in a number of stochastic extraction techniques such as the Monte Carlo or stochastic collocation algorithm [7]; and the generation of parameterized reduced-order models used in timing and noise analysis.

From the above many examples it is apparent that there is a need to develop solvers that utilize information from previous similar solves in a smart way such that the solution time for subsequent configurations is reduced. Consequently, we propose to use, as a measure of solver efficiency, the average time required to solve a large number of similar simulations rather than the solution time for a single configuration.

Little research has been directed toward such an objective. However, careful investigation of the few existing ideas, such as subspace recycling [8], [9], reveals that the simulation time is asymptotically constrained by at least one matrix vector product. This means that for the existing approaches the computational complexity of solving K configurations is at least K times the complexity required for one single configuration. In this paper we argue that the floating random walk (FRW) algorithm [10] can be used to efficiently solve a very large number K of similar configurations in a time almost completely independent of the number of configurations K.

Before proceeding with the discussion, we mention a few of the additional advantages of the FRW algorithm [10]. Foremost, the complexity of the algorithm is independent of the number of conductors thus enabling the handling of very large and complex conductor systems quite efficiently. In addition, the algorithm is extremely efficient in terms of memory utilization since it does not involve matrix assembly or system solves. Furthermore, FRW offers the ability to report intermediate results with error bounds, which enables the design of stopping criteria that are in line with the required accuracy of the extraction case at hand. Finally, FRW is an "embarrassingly" parallel algorithm and can therefore utilize current advances in multi-threaded, multi-core computer architectures.

In this paper we report on three novel contributions to the FRW algorithm. In section III we reformulate the algorithm to efficiently handle multi-layered dielectric configurations. In section IV we discuss how the FRW algorithm can be modified to efficiently compute capacitance sensitivities to small parameter variations. In section V we propose a new path recycling FRW algorithm that computes incrementally capacitance variations due to large changes in the geometry without the need to recompute the entire structure. Finally, in section VI we demonstrate a variety of examples to validate our analysis.

II. Background

A. Floating Random Walk

The FRW algorithm [10] is based on expressing the capacitance C_{ij} between conductors i and j as a multidimensional (possibly infinite dimensional) integral of known potentials on the conductor surfaces. This is achieved by writing the unknown potential at any arbitrary intermediate point $\eta^{(i)}$ as a function of the potential at the boundary of the largest square S_{i+1} constructed such that it is centered around point $\eta^{(i)}$ and extends to the nearest conductor without including any metal structures (Fig. 2). By construction, part of the boundary of S_{i+1} is touching part of some conductor boundary and therefore has a known potential. Consequently, the boundary of S_{i+1} is partitioned into two mutually exclusive segments $S_{i+1} = K_{i+1} \cup U_{i+1}$ (see Fig. 2), where K_{i+1} and U_{i+1} are the parts of the boundary with known and unknown potentials, respectively. The final capacitance formula is given by:

$$\begin{aligned} C_{ij} &= -\int_{S_0} d\eta^{(0)} \hat{n} \cdot \nabla \int_{K_1} d\eta^{(1)} G(\eta^{(0)}, \eta^{(1)}) \phi(\eta^{(1)}) + \\ &+ \int_{U_1} d\eta^{(1)} G(\eta^{(0)}, \eta^{(1)}) \int_{K_2} d\eta^{(2)} G(\eta^{(1)}, \eta^{(2)}) \phi(\eta^{(2)}) + \dots \\ &+ \int_{U_1} d\eta^{(1)} G(\eta^{(0)}, \eta^{(1)}) \int_{U_2} d\eta^{(2)} G(\eta^{(1)}, \eta^{(2)}) \times \\ &\quad \times \dots \int_{K_m} d\eta^{(m)} G(\eta^{(m-1)}, \eta^{(m)}) \phi(\eta^{(m)}), \end{aligned} \tag{1}$$

where S_0 is a Gaussian surface surrounding conductor i, $\hat{n}$ is the corresponding normal, ϕ is the electrostatic potential, $G(\eta^{(i)}, \eta^{(i+1)})$ is the Green's function used to express the potential at $\eta^{(i)}$ as a function of the potential at the boundary of S_{i+1}.

Important Note: Within the floating random walk algorithm, the Green's function $G(\eta^{(i)}, \eta^{(i+1)})$ associated with the square S_{i+1} can be given a probabilistic interpretation, namely, it identifies a transition probability that measures the likelihood of a point $\eta^{(i)}$ inside the square S_{i+1} to be connected with a point $\eta^{(i+1)}$ on the boundary of the square. Therefore, in the sequel, we will use the terminology of a transition square and a transition probability, the latter interchangeably with a Green's function.

The multidimensional integral (1) is computed using standard Monte Carlo integration, where only one quadrature point is select for each integral over a transition square boundary. The sequence of quadrature points on the boundary of different transition squares can be interpreted as a random walk from a transition square to another. The stopping criterion of this walk is achieved when the random step falls within a distance ε from a conductor boundary where the potential is known and does not need to be calculated further.

B. Handling Multi-layered Media

The standard floating random walk algorithm can handle arbitrary layered media. This is achieved by treating the boundaries between the dielectric layers as constraints on the step size and consequently as acceptable stopping points [10]. The difference between a conductor boundary and a dielectric interface is that the potential of the former is known while that of the latter is unknown. Consequently, the random walk is restarted if it terminates at a dielectric interface. The restarts are repeated until the walk terminates at a conductor boundary. However, if we consider current technologies with complex layered configurations and small thicknesses such

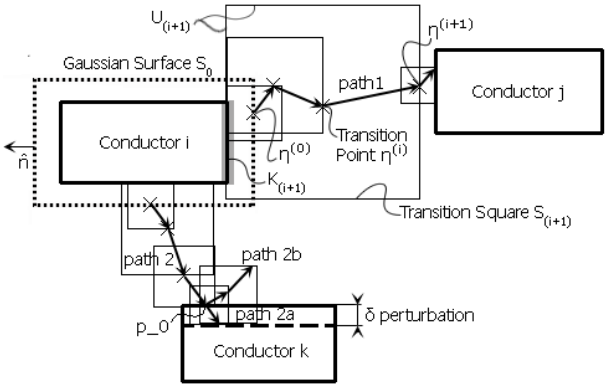

Fig. 2. Typical random walk path (path 1) from conductor i to conductor j. Another path (path 2) ending at conductor k is shown. A step of the sensitivity continuation algorithm is also shown (path 2a, path 2b).

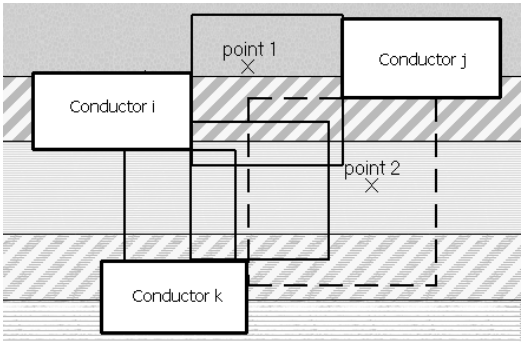

Fig. 3. Typical non-homogeneous dielectric media. Transition point 1 lies within one of the transition squares (solid boundaries) for which the transition probabilities have been precomputed. Transition point 2 will instead cause the generation of a new transition square (dashed boundary) and the computation of all its transition probabilities.

a random walk with restarts becomes time consuming. A more efficient alternative approach to restarting the random walk was derived for simple dielectric configurations [11]. It relies on pre-computing the Green's functions offline using a stochastic algorithm. Such Green's functions are tabulated and used within the random walk to compute the transition probabilities $G(\eta^{(i)},\eta^{(i+1)})$ at step $i+1$. Unfortunately, this approach is limited to a small number of dielectrics and is hard to generalize since it requires the precomputation and tabulation of all possible Green's functions necessary to complete the random walk. Furthermore, it does not seem to exploit the possibility of computing the layered Green's function using a deterministic algorithm, nor does it benefit from the distinct advantages of computing the layered Green's function online rather than offline. In section III we present a novel algorithm that solves the above two problems.

III. Modified FRW for Non-Homogeneous Media

In this section we present an efficient algorithm for handling structures in non-homogeneous media (i.e. with arbitrary dielectric profile). In our algorithm, transition squares are constrained only by surrounding metals (Fig. 3), and NOT by dielectric interfaces as in all other available FRW algorithms which account for dielectrics . Although the Green's function inside such resulting non-homogeneous transition squares is not available in closed form, we can easily calculate it using a standard finite difference method [5]:

$$\begin{pmatrix} M_{11} & M_{12} \\ 0 & I \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \phi_B \end{pmatrix}, \tag{2}$$

where M_{11} and M_{12} are the standard extremely sparse finite difference matrices discretizing the Laplace operator and relating ϕ_1, the grid point potentials in the interior of the transition square, to ϕ_B the grid point potentials on the boundary of the transition square. This system of equations can be reduced by eliminating the intermediate variable ϕ_2 to obtain

$$\phi_1 = \left[-M_{11}^{-1}M_{12}\right]\ \phi_B \tag{3}$$

Since this equation allows to express potentials in the interior of a transition square as a function of potentials on its boundary, it is straight forward to recognize that the γ^{th} row of the matrix $-M_{11}^{-1}M_{12}$ represents the elements of the desired discretized Green's function

$$\left[G(\eta^{(i)},\eta_1^{(i+1)})\Delta_1, G(\eta^{(i)},\eta_2^{(i+1)})\Delta_2, \cdots, G(\eta^{(i)},\eta_{N_B}^{(i+1)})\Delta_{N_B}\right], \tag{4}$$

where $\eta^{(i)}$, γ are the location and index of a point inside of the transition square, respectively, $\eta_k^{(i+1)}$ and Δ_k are the center point and the length of the k^{th} discretization of the transition square boundary, respectively.

Theorem 3.1: The Green's function obtained from (4) is a discrete probability density function, i.e. it is positive, and it integrates to one. [The proof is is omitted due to lack of space].

The Green's function obtained from (4) is therefore precisely the desired step transition probability associated with the non-homogeneous transition square for our modified FRW algorithm.

In order to further optimize our approach we note that each transition point does not necessarily need to be in the center of the transition square as prescribed by the standard FRW algorithms. In fact ANY point within the square can be written in terms of the potential of the boundary of the transition square. Since the transition squares are determined only by the geometry, which is fixed at every step for every walk, and since the computational domain can be fully covered by a small number of unique transition squares, the transition probabilities (i.e. Green's functions in (4)) need to be calculated only for such a small number of squares. It should be further emphasized that squares and transition probabilities calculated in the first random walk sequence can be re-used in all the subsequent walks. Moreover, the required memory to store the transition probabilities is insignificant.

Our complete approach is finally summarized in Algorithm 1 where the transition probabilities are computed incrementally only for the needed transition squares and then stored for efficient re-use.

Algorithm 1 Modified FRW for Non-Homogeneous Media

for each given transition point **do**
  Search for a precomputed transition square fully enclosing given point
  if precomputed square found (e.g. point 1 in Fig. 3) **then**
    Use precomputed transition probability and proceed to next transition point
  else {precomputed square not found (point 2 Fig. 3)}
    Generate new transition square such that it extends to all neighboring conductors, fully enclosing new point. (Note: new point will not be generally in center of new square)
    Call finite difference routine and compute $-M_{11}^{-1}M_{12}$, the full transition probability matrix for every point within interior of square
    Update database with new transition square and corresponding transition probabilities
  end if
end for

IV. Sensitivity Analysis

Sensitivity analysis within a deterministic capacitance extractor is generally implemented using the adjoint method [5], [12]. Such method facilitates computing the sensitivity of the capacitance with respect to a large number of independent parameters with a computational complexity independent of the number of parameters. However, the complexity of the adjoint method depends linearly on the number of output capacitances. Another approach to sensitivity analysis is the finite difference method (direct method). This method relies on perturbing each parameter (one at a time), computing the capacitance corresponding to the perturbed system, and then using a finite difference approximation of the gradient to compute the sensitivities of the capacitances to the perturbed parameter. The complexity of such approach is independent of the number of output capacitances and depends linearly on the number of parameters. In this section we demonstrate that the FRW algorithm facilitates an efficient finite-difference-based approach to computing the sensitivity of the capacitance matrix with respect to a large number of independent variations.

A. New FRW-Based Sensitivity Analysis Algorithm

As long as the perturbation is small, the finite difference sensitivity for a positive parameter perturbation is the same as for a negative perturbation. Consequently, we are free to choose the most convenient sign. One of the key ideas in this section is to always choose geometrical perturbations that "reduce" the size of the conductors as opposed to "increasing" them (as demonstrated by the δ shift in lower conductor k in Fig. 2). In this case the computational domain occupied by the transition squares during the computation of the nominal capacitance is simply extended and therefore all transition squares and random walk sequences are valid and could be potentially reused also in the perturbed configuration.

The transition squares that were previously touching the perturbed edge of a conductor are now obviously not touching it in the new configuration, and therefore have zero stopping probability in such configuration. This indicates that a capacitance difference between the nominal and perturbed geometries occurs if and only if at least one path in the nominal geometry terminates at one of the conductor boundaries belonging to the subset of conductor edges that will undergo perturbation (e.g. termination point p_0 of path 2 in Fig. 2). When such a potential difference is detected, it can be extremely efficiently calculated by simply continuing all those now unterminated random walk paths.

Instead of performing first a FRW on the nominal geometry and then continuing the unterminated paths for the sensitivity analysis a further memory usage optimization can be achieved by interleaving nominal and sensitivity calculations and hence avoiding altogether the need to store any of the FRW paths. Algorithm 2 summarizes our final proposed procedure computing the sensitivity matrix $C.S.(x,k) = \frac{\partial C_{ix}}{\partial P_k}$ of the capacitance vector representing the capacitances between a specific conductor i (target) and all other N conductors in the system C_{ix}, $x \in \{1,2,\ldots,N\}$ with respect to the set of parameters $\{P_k\}$, $k \in \{1,2,\ldots,K\}$. In such algorithm we compute incrementally and simultaneously all the capacitances C_{ix}^l associated with all possible perturbations $l \in \{0,1,\cdots,K\}$, where $l = 0$ corresponds to the nominal unperturbed configuration.

Algorithm 2 Combined Nominal plus Sensitivity FRW Solver

Generate K perturbed geometries by changing parameter P_k by δ_k (choose parameter perturbation such that each conductor dimensions are "reduced" as opposed to "increased")
repeat
  Compute a path of FRW for the nominal geometry starting from conductor i
  if walk terminates at any unperturbed boundary of conductor x **then**
    Add value of path to ALL capacitances C_{ix}^l
  else {walk terminates at some boundary of conductor x associated with perturbation parameter P_k}
    Add the value of path to all C_{ix}^l EXCEPT FOR $l = k$
    Continue path for perturbed geometry (e.g. path 2 in Fig. 2)
    if continued path terminates at conductor m **then**
      Add value of continued path to C_{im}^k
    end if
  end if
until desired accuracy achieved
$C.S.(x,k) = \frac{C_{ix}^k - C_{ix}^0}{\delta_k}$

B. Complexity Estimation

Step 7 can be interpreted as a random walk starting at a distance δ_k away from the surface of conductor k, and continuing until it terminates on any conductor including conductor k itself (Fig. 2). The change in the capacitance value is intimately related to the number of paths that terminate at a conductor different from k (path 2b in Fig. 2). Since the the difference between the capacitance of the nominal and perturbed systems is typically very small, it is very reasonable to expect (and we have actually observed experimentally) that the majority of such paths terminate back at conductor k in one iteration (path 2a in Fig. 2) or few more iterations in rare cases. Hence the extra cost of this combined nominal plus sensitivity algorithm over the nominal algorithm is quite small.

More specifically let's compare the complexity of the proposed algorithm with that of other commonly applied methods. Assume the worst case scenario where all conductors in the geometry are perturbed as described in subsection IV-A. This means that all the random walk paths will undergo step 7 of the algorithm. Consequently, the incremental cost of computing the capacitances of the perturbed geometry will be proportional to the length of the continuation paths. The average length of the continuation paths is smaller than the average length of the nominal random walk paths. Therefore the total cost of computing the finite-difference-based sensitivity matrix is not more than $2\times$ the cost of solving just the nominal capacitance. This very conservative estimate indicates that our method, despite being finite difference based, has a computational complexity that is independent of both the number of parameters (unlike standard finite-difference method) and the number of output capacitances (unlike standard adjoint method). Consequently, our proposed algorithm is computationally superior to both the adjoint method and the standard finite-difference method.

V. Incremental Variational Analysis

In this section we demonstrate how the FRW can be used to compute capacitances of "similar" geometries resulting from multiple **large** variations, not captured efficiently via sensitivity analysis. The number of possible "similar" geometries depends exponentially on configuration order, i.e. the total number of geometrical parameters that are allowed to change simultaneously, thus restricting all existing variational algorithms to a very small number of parameters.

A. General Observations

The key difference from the sensitivity analysis is that here perturbations are not small, and hence we are not free to choose if they reduce or increase the size of conductors based on what is more convenient for our solver. Obviously when conductors are decreased in size the same exact method used for the sensitivity would work just fine. However when instead a given conductor is increased in size, any path that in the nominal configuration depended on such conductor will become at least partially useless for the perturbed configuration. When we state in this section that "a path depends on a given conductor" we mean that some transition squares used to generate the transition probabilities of some points in the path are constrained by the given conductor. It is fortunately true that each single walk has generally a very sparse dependence on the overall set of all conductors, i.e. the number of conductors constraining the transitions squares of a single path is very small compared to the total number of conductors. Such a property is even more emphasized in structures with a large density of conductors, since the probability of path termination is large in such structures. Consequently, for any given new configuration, almost all the paths of the nominal configuration can still be completely re-utilized to compute the capacitance of the perturbed geometry, provided such paths are not dependent on any perturbed enlarged conductor.

If a path depends on a perturbed enlarged conductor, then such path must be partially re-simulated starting from the first non-reusable transition square affected by the change in geometry. Since the number of paths needing updating is very small compared to the total number of paths, the FRW algorithm still obtains almost instantly the solution of the new configuration. This is formalized in the theorem below.

Assume that the total parameter set describing the geometrical variations is composed of K parameters. Define a set of configurations $C^K_{\mathbf{j}}$ as the set of geometries constructed by altering the parameters indexed by the j-tuple $\mathbf{j} = (i_1, i_2, \ldots, i_j)$ such that $i_l \in \{1, \cdots, K\}$. Assume C^K_0 is the nominal configuration. C^K_1 would contain configurations of order one where only one parameter at the time is allowed to change. As an example C^K_4 would instead contain all configurations of order four where only four parameters at the time are allowed to change.

Theorem 5.1: Assume that the FRW simulation of the configurations in $C^K_{\mathbf{j}}$ has been completed. Let the resulting paths be indicated by $\wp_0$. Partition the paths into groups $\wp_{i_l}$ such that each group includes all the paths that depends on parameter i_l. Note that these groups are not mutually exclusive. Let the number of paths in group $\wp_{i_l}$ be indicated by N_{i_l}. The set containing the paths required to be re-simulated to compute the capacitance of configuration $C^K_{\mathbf{j}}$ is given by:

$$\Delta\wp^K_{\mathbf{j}} = \bigcap_{l=1}^{j} \wp_{i_l} \tag{5}$$

The number of such paths is given by the cardinality of the set and is less than $\min_{l=1,\ldots,j} N_{i_l}$. Consequently, the number of re-simulated paths *decreases* as the number of varying parameters increases.

From Theorem 5.1 we observe that in order to simulate the configurations in $C^K_{\mathbf{j}}$ we only need to re-simulate those paths that depend on **all** the varying conductors. Note that in Theorem 5.1 we describe the subset of paths that needs to be re-simulated, however, the theorem does not make any statement about the rest of the paths or how those can be reused. This is the aim of the following theorem.

Theorem 5.2: Use the same assumption in Theorem 5.1. Further assume that the capacitances of all configurations $C^K_{\mathbf{m}}$,

constructed by altering the parameters indexed by the m-tuple $\mathbf{m} = (i_1, i_2, \ldots, i_m)$ such that $i_l \in \{1, \cdots, K\}$ and $m < j$ have been extracted. Consequently, the paths of all configurations $C^K_{\mathbf{m}}$ are available. To compute the capacitance of configuration $C^K_{\mathbf{j}}$ we need to reuse the paths in the set $\wp_0 \setminus \Delta\wp^K_{\mathbf{j}}$. By construction, a path $p_{\mathbf{m}} \in \wp_0 \setminus \Delta\wp^K_{\mathbf{j}}$, where $\mathbf{m}$ is the m-tuple containing the indexes of the parameters that constrain the path, was re-simulated in the configuration $C^K_{\mathbf{m}}$ and can therefore be used to populate the random walks of $C^K_{\mathbf{j}}$.

One main implication of Theorem 5.2 is that the configurations are simulated in a top-down fashion, i.e. starting from the nominal configuration and completing the configurations according to their order $C^K_{\mathbf{1}}$, $C^K_{\mathbf{2}}$, $\cdots$, $C^K_{\mathbf{K}}$.

To the best of the authors' knowledge, the presented algorithm is the only variation-aware extraction algorithm, for which the incremental computational effort required to solve high order perturbations is strictly non-increasing as a function of the perturbation order.

B. Memory Management

If the capacitances of the configurations $C^K_{\mathbf{1}}$, $C^K_{\mathbf{2}}$, $\cdots$, $C^K_{\mathbf{K}}$ were to be computed sequentially, we would need to keep track (store) the details of every simulated or re-simulated path (e.g. transition points, squares, and the conductors constraining the transition squares of every path). Such an algorithm would loose one of the main advantages of the original FRW algorithm, namely, the fact that it requires minimal memory usage. To overcome such a drawback, as we did in the sensitivity solver, we compute the capacitances of all configurations concurrently and incrementally as summarized in Algorithm 3 Consequently, there is no requirement for extra memory.

Algorithm 3 Incremental FRW Variational Analysis

1: **repeat**
2: Compute a FRW path for nominal geometry
3: Use it to update nominal capacitance
4: Reuse it according to Theorem 5.2 to update capacitances of NOT dependent configurations
5: **for** each dependent configuration (according to Theorem 5.1) **do**
6: Partially re-simulate it
7: Re-use it according to Theorem 5.2 to update capacitance of dependent configuration
8: **end for**
9: **until** desired accuracy achieved

VI. Results

A. Dielectric Validation

In this subsection we show implementation results of the numerically obtained Green's function, and the effect of using a non-centered Green's function on the speed of convergence of the floating random walk algorithm. The multilayered stack is composed of a substrate with dielectric constant $\varepsilon = 11.9$, and 10 successive layers of dielectric constants ranging from 2.2 to 4.4, and finally a half space of free space. The stack is simulated using both the standard technique and the modified finite-difference based technique proposed in section III. We have observed that using our new Green's function the average path length is reduced from 19 to 6 steps, consequently, the simulation time is reduced by a factor of 3. We further observed that the number of unique Green's function computations is on the order of 1000, which is very small compared to the total number of random walks ($\simeq 10^5$). This in turn explains why the average walk cost remains approximately the same. Finally, we show in Fig. 4 a sample Green's function computed for a portion of the dielectric stack.

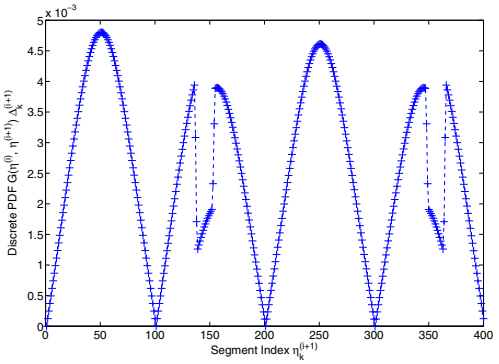

Fig. 4. Example of a numerically computed Green's function $G(\eta^{(i)}, \eta^{(i+1)})$, where $\eta^{(i+1)}$ is the variable parameterizing the contour of a transition square. The discontinuities around segment indexes 150 and 350 correspond to the thin central layer with lower dielectric constant.

B. Sensitivity Analysis

In this subsection we demonstrate the effectiveness of our sensitivity analysis by computing the sensitivities of a 20 conductors structure (Fig. 5) to variations in the conductor geometries. Conductor 14 in Fig. 5 is the target conductor for which we extract the capacitance vector. Configuration k is constructed by reducing the width and thickness of conductor k by 2%, while keeping all other conductors in their nominal size. The total number of configurations is 20. Consequently, we can compute the sensitivity of the geometry with respect to shape variations (shrinking, expansion). In Fig. 6 (a) we compare the capacitances $C_{14,19}$ obtained from our FRW algorithm with those obtained from a standard boundary element method (BEM) for all different configurations. The confidence interval is set to 1%. Configuration 21 represents the nominal configuration. In Fig. 6(b) we compare the percentage relative variation in the capacitance $\frac{C^{(i)}_{14,19} - C^{(21)}_{14,19}}{C^{(21)}_{14,19}} \times 100$, where i is the configuration index and C_{nom} is the nominal capacitance. We observe that the accuracy of the absolute variation $C^{(i)}_{14,19} - C^{(21)}_{14,19}$ is about 1%, due to the error cancellation resulting from the correlation (path sharing) between the perturbed and nominal configurations. The sample correlation coefficient is approximately 0.8. The total time required to complete the sensitivity analysis using

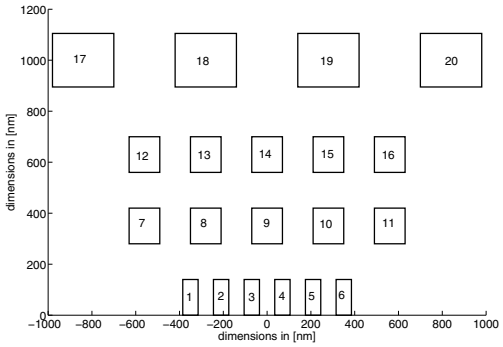

Fig. 5. Two-dimensional cross-sectional-view of 20 conductors geometry.

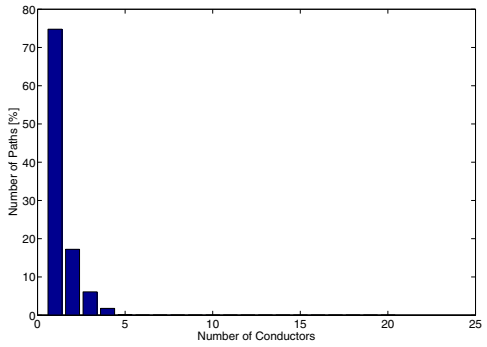

Fig. 7. Percentage of paths dependent on a particular number of conductors.

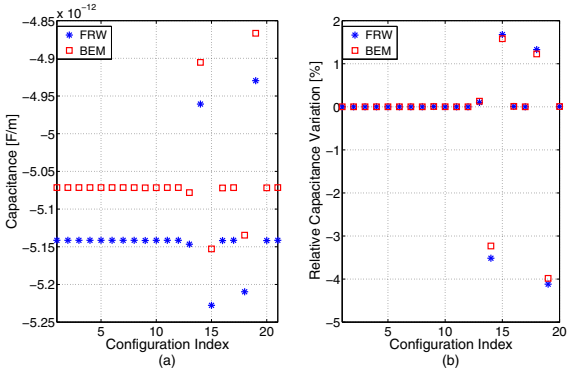

Fig. 6. (a) Capacitance $C^{(i)}_{14,19}$ obtained from both FRW and BEM algorithms. (b) Relative capacitance variation $\frac{C^{(i)}_{14,19}-C^{(21)}_{14,19}}{C^{(21)}_{14,19}} \times 100$ demonstrating error cancellation due to random walk correlation.

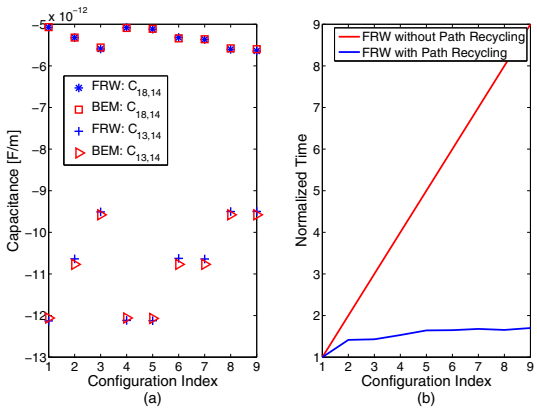

Fig. 8. (a) Validation of the accuracy of the random walk algorithm variational analysis compared to a robust boundary element method code. (b) Cumulative simulation time for handling additional configurations using FRW as compared to the estimated standard time.

FRW is only a factor 1.39 larger than (i.e. $1.39\times$) the nominal simulation time, as opposed to $20\times$ obtained from the standard finite difference technique or the $20\times$ obtained from the standard adjoint method. Furthermore, our floating random walk sensitivity analysis is about $10\times$ faster than the BEM adjoint sensitivity analysis and about $10\times$ faster than the BEM finite difference sensitivity analysis.

C. Variational Analysis

In this subsection we demonstrate how the floating random walk algorithm is used to efficiently compute capacitances of perturbed geometries. The focus of this subsection is on large perturbations not captured via sensitivity analysis.

First, in Fig. 7 we demonstrate the sparse dependence of the nominal random walk paths on the overall set of conductors. We observe that more than 73% of all paths are responsible for the self-capacitance and therefore end at the target conductor without touching any other conductor. Another 18% of the paths depend only 2 conductors. We further observe that almost all the rest of the paths depend on no more than 3 or 4 conductors. Consequently, any perturbation affecting more than 5 conductors can be simulated with almost no additional effort. Such sparse dependence constitutes the fundamental strength of the FRW.

The accuracy of the variational analysis presented in section V is demonstrated in Fig. 8(a) by comparing the capacitances $C_{18,14}$ and $C_{13,14}$ obtained from our algorithm with those obtained from the standard boundary element method. The comparison is demonstrated for 9 different configurations. These correspond to shrinking conductors 13 and 15 (i.e. the right and left conductors surrounding conductor 14) by factors of $(0,0)$, $(12.5\%,0)$, $(25\%,0)$, $(0,12.5\%)$, $(0,25\%)$, $(12.5\%,12.5\%)$, $(12.5\%,25\%)$, $(25\%,12.5\%)$ and $(25\%,25\%)$, respectively. The accuracy is better than 5% for all configurations. Furthermore, in Fig. 8(b) we show the simulation time required to compute the capacitance of 9 different configurations using the FRW algorithm as compared to the linear increase in time typical of the standard method without path recycling. The sublinear complexity of our algorithm is clearly demonstrated.

Finally, we validate this observation by computing the time required to generate all sparse grid points in a 20-dimensional space required for expansions of polynomial exactness 3, 5, 7, and 9 [13]. The total number of grid points of such construc-

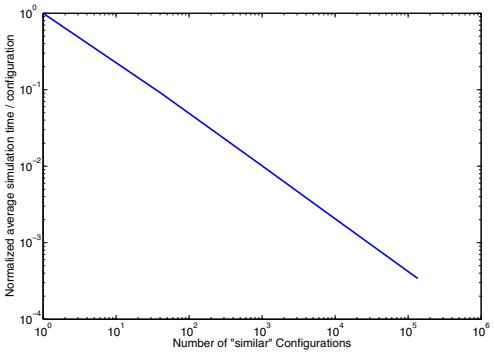

Fig. 9. Log-log plot demonstrating the reduction in the average simulation time with the increase number of configurations. Configurations constructed based on the 5^{th} order sparse grid.

tions is 41, 861, 12341, and 135751, respectively. The relation between the number of solves and the average simulation time per configuration is shown in Fig. 9. We observe that the average simulation time per configuration is reduced when the number of similar configurations are increased. Practically speaking, the total simulation time required for solving all the 12,341 configurations is less than the time required by our algorithm to solve 22 independent configurations (i.e. less than 0.2% of the time required to solve all configurations independently, corresponding to a speedup of 561 times). Moreover, the time required to solve a total of 130,000 "similar" configurations is the same time required for solving less than 50 independent configurations, or equivalently the average simulation time per one solve is reduced by three orders of magnitude.

VII. Conclusion

In this paper we have presented three modifications of the FRW algorithm that are of immediate relevance to variation-aware and lithography-driven layout parasitic extraction flows.

First, we have presented a new algorithm to efficiently compute the capacitance of conductors in non-homogeneous media using a finite-difference technique to compute the specialized Green's function of the FRW algorithm. The new algorithm results in an average decrease in the simulation time by a factor of $3\times$ when compared to the standard FRW algorithm. This new algorithm is general enough to handle any dielectric configuration.

Second, we have presented a new finite-difference-based sensitivity analysis within the improved FRW algorithm to efficiently compute capacitance sensitivities with respect to a large number of **small** parameter variations. The new algorithm is $10\times$ faster than both the BEM finite difference sensitivity analysis and the BEM adjoint sensitivity analysis. Furthermore, we have demonstrated that the cost for computing nominal capacitances and all the sensitivities can be conservatively estimated to be less than $2\times$ the cost of computing only the nominal capacitance, regardless of the number of parameters. In our examples the cost for nominal plus sensitivity analysis was observed to be $1.4\times$ the cost of nominal analysis alone.

Third, we have presented a new FRW algorithm to compute the capacitances of "similar" configurations resulting from simultaneous **large** perturbations of the geometrical parameters of the original geometry. The new algorithm satisfies the objective of this paper, namely, that the *average* time required to solve a single configuration within a set of similar configurations is reduced as the cardinality of the set is increased. We have observed that the average simulation time per configuration for a set of 10^5 similar configurations is three orders of magnitude smaller than the simulation time required by the solution of one independent configuration. Consequently, we were able to solve more than $130,000$ similar configurations in the time required to solve just 50 independent configurations. We believe that the latter result will naturally fit in a litho- and CMP-aware extraction flow.

Acknowledgment

The authors would like to acknowledge the support of IBM, Cadence and Interconnect Focus Center (a Semiconductor Research Corporation and DARPA program). In particular, the authors would like to thank Dario Gil (IBM Research) for providing Fig. 1, and Fook-Luen Heng, David Kung (IBM Research) and Patrick Williams (IBM Systems and Technology Group) for their management support.

References

[1] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3D structures" *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1059–1072, 1997.

[2] K. Nabors and J. K. White, "FASTCAP A multipole-accelerated 3D capacitance extraction program" *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1447–1459, 1991.

[3] W. Hackbusch, "A sparse matrix arithmetic based on H-matrices. I. Introduction to H-matrices" *Computing*, pp. 89–108 ,1999.

[4] Y. Zhou, Z. Li, Y. Tian, W. Shi, F. Liu, "A New Methodology for Interconnect Parasitic Extraction Considering Photo-Lithography Effects" *ASP-DAC'07*, pp. 450 – 455, Yokohama, Japan, 2007.

[5] T. El-Moselhy, I. Elfadel and D. Widiger, "Efficient Algorithm for the Computation of On-Chip Capacitance Sensitivities with respect to a Large Number of Parameters" *Design Automation Conference*, DAC 2008.

[6] N. Chang *et. al.*, "Fast Generation of Statistically-based Worst-Case Modeling of On-Chip Interconnect" *Proc. ICCD*, pp. 720 – 725, 1997.

[7] D. Xiu and J. Hesthaven, "High-Order Collocation Method for Differential Equations with Random Inputs" *SIAM J. Sci. Comput.*, 2005.

[8] D. Chan and M. K. NG, "Galerkin Projection Methods for Solving Multiple Linear Systems" *SIAM Journal on Scientific Computing*, pp. 836 – 850, Vol. 21, Issue 3, 1999.

[9] M. Parks, E. Sturler, G. Mackey, D. Johnson and S. Maiti, "Recycling Krylov Subspaces for Sequences of Linear Systems" *SIAM Journal on Scientific Computing*, pp. 1651 – 1674, Vol. 28, Issue 5, 2006.

[10] R. B. Iverson and Y. L. Le Coz, "A Stochastic Algorithm for High Speed capacitance Extraction in Integrated Circuits," *Solid-State Electronics*, Vol. 35, No. 7, pp. 1005–1012, 1992.

[11] J. N. Jere and Y. L. Le Coz, "An Improved Floating-Random Walk Algorithm for Solving Multi-Dielectric Dirichlet Problem" *IEEE Transaction on Microwave Theory and Techniques*, Vol. 41, No. 2, pp. 325 – 329, 1993.

[12] J. Wang and J. White., "Fast Algorithms for Computing Electrostatic Geometric Sensitivities" *International Conference on Simulation of Semiconductor Processes and Devices*, pp. 121-123, 1997.

[13] T. Gerstner and M. Griebel, "Numerical Integration using Sparse Grids" *Numerical Algorithms*, 1998.