An Efficient Resistance Sensitivity Extraction Algorithm for Conductors of Arbitrary Shapes

An Efficient Resistance Sensitivity Extraction Algorithm for Conductors of Arbitrary Shapes

Tarek El-Moselhy
MIT
Cambridge, MA 02139
tmoselhy@mit.edu

I. M. Elfadel
IBM
Yorktown Heights, NY 10598
elfadel@us.ibm.com

Bill Dewey
IBM
East Fishkill, NY 12533
bdewey@us.ibm.com

ABSTRACT

Due to technology scaling, integrated circuit manufacturing techniques are producing structures with large variabilities in their dimensions. To guarantee high yield, the manufactured structures must have the proper electrical characteristics despite such geometrical variations. For a designer, this means extracting the electrical characteristics of a whole family of structure realizations in order to guarantee that they all satisfy the required electrical characteristics. Sensitivity extraction provides an efficient algorithm to extract all realizations concurrently. This paper presents a complete framework for efficient resistance sensitivity extraction. The framework is based on both the Finite Element Method (FEM) for resistance extraction and the adjoint method for sensitivity analysis. FEM enables the calculation of resistances of interconnects of arbitrary shapes, while the adjoint method enables sensitivity calculation in a computational complexity that is independent of the number of varying parameters. The accuracy and efficiency of the algorithm are demonstrated on a variety of complex examples.

Categories and Subject Descriptors. J.6 [**Computer Aided Engineering**]: Computer-aided design (CAD)

General Terms. Algorithms

Keywords. resistance extraction, finite-element method, sensitivity, adjoint method, shape variations

1. INTRODUCTION

Resistance calculation is a fundamental component of any standard VLSI layout extraction flow. Several algorithms have been proposed for resistance calculation, ranging from simple analytical formulae to complex field solvers [1, 2, 3, 4, 5, 6].

Among the latter, more emphasis has been placed recently on the finite element method (FEM) [4, 5, 6]. This emphasis is justified by the fact that rigorous and accurate resistance calculation is the result of solving a Laplace equation in a *closed* domain that may enclose a conducting medium with a non-homogeneous resistivity distribution. FEM enables solving such equation even if the boundaries of the closed domain are irregular or fall outside a pre-determined Manhattan grid. Common instances of such boundaries are the ones outlining the contours due to lithographic processing. Common instances of non-homogeneous resistivity distributions include low-resistivity on-chip copper interconnect and vias surrounded by high-resistivity liners. FEM is also very well adapted to the specific boundary conditions of the resistance calculation problem, namely, the mixed Neumann-Dirichlet boundary conditions. These features of resistance calculation have to be contrasted with those of capacitance calculation which is the result of solving a Laplace equation in an *open* domain with Dirichlet boundary conditions. Such calculation is better handled with a boundary element method. The reader is referred to [7] for more details about the fundamentals of FEM and to [8] for its application to resistance calculation.

Despite a fair amount of research directed towards resistance calculation and its incorporation in the VLSI layout extraction flow, the impact of manufacturing variability on the extracted resistance has been barely addressed. The several sources of variability, whether systematic and random, will combine to impact the boundaries of shapes printed on the wafer, thus leading to variations in their electrical properties. Lithography, etching, and chemical-mechanical polishing are such sources. With the decrease in feature sizes, the radius of mutual interaction between shapes is scaling up, thus complicating even further the correct prediction of the ultimate boundaries of a given shape. This uncertainty on the shape boundary is at the root of a pressing need to come up with efficient resistance calculation methods that can extract the resistance of a given shape for a whole family of boundary realizations.

Accounting for variability in extraction has focused on the capacitance calculation problem where the proposed solutions have ranged from very simple analytical correction formulas to very complex stochastic integral methods. Very little work has been done for resistance calculation [9], perhaps because of the latter's deceptive simplicity when it deals with wire-like patterns. Yet, non wire-like geometric patterns such as bends, jogs, corners, steps, pads, ports, fingers, diffusions, contacts, vias, and all their lithographic variations require very special attention when it comes to resistance extraction. Furthermore, in a variation-aware CAD context, even the resistance of wire-like patterns now require special attention for the simple reason that these patterns are no longer rectangular. The work described in this paper aims at addressing the resistance "gap" in the variability-aware extraction flow by proposing a rigorous resistance variation analysis based on sensitivity calculation. Among the range of variational methods, sensitivity-based ones strike a needed balance between accuracy and computational efficiency. Such methods have been extensively applied to capacitance calculation in its various formulations [10, 11, 12, 13] but are yet to be derived, implemented, and validated for resistance extraction. In a variation-aware VLSI extraction flow, one can make use of both the nominal resistance and the sensitivity of the resistance

DAC'09, July 26-31, 2009, San Francisco, California, USA

to the geometrical variations to predict the resistance of a slightly perturbed shape. More precisely, one may approximate the resistance function using a multivariate, first-order Taylor expansion

$$R = R_0 + \sum_m \frac{\partial R}{\partial p_m} \Delta p_m \tag{1}$$

where Δp_m is the perturbation around a nominal value of the p_m parameter and $\frac{\partial R}{\partial p_m}$ is the sensitivity (expressed as a partial derivative) of the resistance with respect to such perturbation. This paper is devoted to presenting an algorithm for computing such sensitivities for conductors of arbitrary shapes. In particular, we show how a pre-existing tool for FEM resistance calculation can be augmented with a sensitivity calculation capability using adjoint variational analysis.

It is important to note that in a VLSI layout extraction flow, FEM may be used in two different ways. The first way is in computing accurate look-up tables of specific wiring patterns for which the wire-like resistance formula will fail. Such computations are done off-line and do not impact the CPU cost of the resistance extraction step. The second way is on-line where FEM is applied very selectively using rules similar to those given in [1]. In this latter approach, caching and pattern recognition are extensively employed to reduce the number of times the FEM solver is actually called. Another important aspect is that in the full layout extraction flow, the overall performance is gated by the capacitance extraction phase rather than the resistance extraction phase. Our work on FEM-based resistance sensitivities should be placed within this overall VLSI layout extraction context.

This paper is organized as follows. Section 2 is a review of both FEM for resistance calculation and adjoint sensitivity analysis. Section 3 is the core of our theory and algorithmic contributions in this paper, namely how FEM and the adjoint analysis can be combined to produce an efficient algorithm for resistance sensitivity extraction. Section 4 illustrates such an algorithm on a set of numerical examples of industrial relevance. These results have been obtained by coding our sensitivity algorithm within an industrial FEM resistance calculator.

2. BACKGROUND

2.1 Resistance Calculation

The problem of resistance calculation is governed by the following partial differential equation

$$\begin{array}{cc} \nabla \cdot (\sigma(r)(-\nabla\phi(r))) = 0 & r \in D \\ \sigma(r)(-\nabla\phi(r)) \cdot \hat{n} = 0 & r \in \partial D_{nc} \\ \phi(r) = \phi_0 & r \in \partial D_c \end{array} \tag{2}$$

where $\phi(r)$ is the electric potential, $\sigma(r)$ is the electric conductivity of the material, D is the *closed* domain of the problem, ∂D_{nc} is the union of the boundary segments which are not assigned a particular potential (referred to as non-contact), ∂D_c is the union of the boundary segments which are assigned a specific potential (referred to as contact), and $\hat{n}$ is the normal to the boundary surface. The second equation is the Neumann boundary condition at the non-contact boundary of the problem and indicates that the current does not flow outside of the metal (the perpendicular current component vanishes). The last equation is the Dirichlet boundary condition at the contacts with prescribed potential ϕ_0. For a set of N_c contacts the resistance between contacts p and q is computed by assigning unit potential to contact p, i.e., $\phi_p = V_p (\equiv 1)$, and assigning zero potential to all other contacts. One then solves (2) to find $\phi(r)$ everywhere in D and subsequently computes the total current

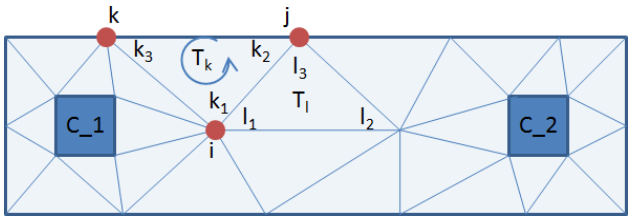

Figure 1: Triangular mesh of a 2-D structure. Triangular element local indexing is *anti* clockwise. Each node has a unique global index and multiple local indices.

entering contact q as

$$I_q = \int_{\partial D_{cq}} \sigma(r)(-\nabla\phi(r)) \cdot \hat{n} dS$$

where ∂D_{cq} is the boundary of port q.

The required resistance is then $R(p,q) = V_p / I_q$. The previous discussion makes it very clear that only one simulation is needed to compute an entire row of the resistance matrix $R(p,x) : 1 \le x \le N_c$.

2.2 Finite Element Method (FEM)

For the sake of simplicity, we will present the basics of FEM in 2D. The extension to 3D is straightforward. Since Problem (2) has mixed Dirichlet-Neumann boundary conditions, the continuous FEM formulation is derived from minimizing the following energy functional

$$E(\phi(x,y)) = \int_D \sigma(x,y) \nabla\phi(x,y) \cdot \nabla\phi(x,y) dx dy \tag{3}$$

To solve for $\phi(r)$, the geometry is first subdivided into smaller elements (Fig. 1). For 2D structures, the Delaunay triangulation is used for the discretization, since it tends to guarantee triangles of reasonable aspect ratios [7]. The resulting elements are described by the coordinates of their nodes. Each node has a unique global index to identify it in the mesh as well as a local index within each triangle it belongs to. Clearly, a given node may have more than one local index, since it may belong to more than one triangle. The local nodes of triangle T_k are labeled k_1, k_2 and k_3 (Fig. 1). The potential within each triangular element is then approximated using a basis of polynomial functions. For simplicity, this basis is taken to be that of first-order polynomials, so that

$$\phi(x,y) = \beta_x x + \beta_y y + \beta_0 \tag{4}$$

Consequently, the potential of every element T_k is described by the three unknown potentials of its nodes $\phi(k_1)$, $\phi(k_2)$ and $\phi(k_3)$ and three coefficients β_x, β_y, and β_0 of (4). The gradient of the potential in (4) can be rewritten in terms of the nodal potential as

$$\begin{aligned} \nabla\phi &= \sum_{i=1}^{3} \frac{1}{\alpha_{T_k}} A(k_i)\phi(k_i) \\ A(k_i) &= \left(y_{k_{i+1}} - y_{k_{i-1}}\right)\hat{x} + \left(x_{k_{i-1}} - x_{k_{i+1}}\right)\hat{y} \end{aligned}$$

where the coordinates of the node k_i are (x_{k_i}, y_{k_i}), α_{T_k} is the area of the triangle T_k, and $\hat{x}$ and $\hat{y}$ are the unit vectors of the x and y axis. Substituting in (3) we obtain the discretized quadratic form

$$E(\phi) = \phi_n^T A \phi_n \tag{5}$$

where ϕ_n is the vector of all node potentials in the mesh and A is the system matrix. The elements of A are written in a compact form

$$A(i,j) = \sum_{k:(i,j)\in T_k} \frac{\sigma(k)}{\alpha_{T_k}} A(k_i) \cdot A(k_j) \tag{6}$$

where i and j are the global indices of the nodes, k is the index of the triangle, $(i,j) \in T_k$ means that i and j belong to a triangle T_k, and $\sigma(k)$ is the conductivity of the region bounded by triangle k. It is further assumed that the local indices of i and j in T_k are k_i and k_j, respectively. Equation (5) is then rewritten as

$$\phi_n^T \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \phi_n = \phi_1^T A_{11}\phi_1 + \phi_2^T A_{21}\phi_1 + \phi_1^T A_{12}\phi_2 + \phi_2^T A_{22}\phi_2 \tag{7}$$

where ϕ_1 is the vector of the unknown potential (potential of all N non-contact nodes), ϕ_2 is the vector of the known fixed potential (potential of N_f nodes on the contacts), A_{11} represents the self interaction of the non-contact nodes, $A_{12} = A_{21}^T$ mutual interaction of contact and non-contact nodes, and A_{22} self interactions of the contact nodes. Equation (7) is then minimized with respect to the unknown potential vector ϕ_1 to obtain

$$A_{11}\phi_1 = -A_{12}\phi_2 \tag{8}$$

Equation (8) is cast in the standard compact form $M\phi = b$, where $M = A_{11}$, $\phi = \phi_1$ and $b = -A_{12}\phi_2$. This linear system is then solved for ϕ to obtain the potential everywhere inside the domain D. Because of our interest in sensitivity calculation, the dependence of M, ϕ, and b on the problem parameters has to be made explicit, so the above linear system is written as

$$M(\mathbf{P})\phi(\mathbf{P}) = b(\mathbf{P}) \tag{9}$$

where $\mathbf{P}$ is a vector of geometrical parameters, such as the dimensions of the structure and the relative position of the contacts within it. The goal is to efficiently evaluate the impact of the variation of such parameters on the computed resistances. This we undertake in the next subsection.

2.3 Adjoint Sensitivity Analysis

The adjoint sensitivity computation is a very efficient algorithm for finding the sensitivity of a given vector $f(\mathbf{P},\phi(\mathbf{P}))$ of length n_o with respect to a parameter vector $\mathbf{P}$ of length n_p. In this subsection, we summarize the simple derivation of the adjoint method as given in [14].

Taking the total derivative of $f(\mathbf{P},\phi(\mathbf{P}))$ with respect to $\mathbf{P}$, we get

$$\frac{df(\mathbf{P},\phi(\mathbf{P}))}{d\mathbf{P}} = \frac{\partial f(\mathbf{P},\phi(\mathbf{P}))}{\partial \mathbf{P}} + \frac{\partial f(\mathbf{P},\phi(\mathbf{P}))}{\partial \phi(\mathbf{P})}\frac{d\phi(\mathbf{P})}{d\mathbf{P}} \tag{10}$$

where $\frac{df(\mathbf{P},\phi(\mathbf{P}))}{d\mathbf{P}}$, $\frac{\partial f(\mathbf{P},\phi(\mathbf{P}))}{\partial \mathbf{P}}$ are matrices of size $n_o \times n_p$, $\frac{\partial f(\mathbf{P},\phi(\mathbf{P}))}{\partial \phi(\mathbf{P})}$ is a matrix of size $n_o \times N$ and $\frac{d\phi(\mathbf{P})}{d\mathbf{P}}$ is a matrix of size $N \times n_p$.

Direct sensitivity methods are based on computing $\frac{d\phi(\mathbf{P})}{d\mathbf{P}}$ using a finite difference (FD) perturbation for each component of $\mathbf{P}$, which is computationally very expensive since it requires n_p independent system solves. However, taking the derivative of linear system (9) with respect to $\mathbf{P}$, we get

$$\frac{\partial M(\mathbf{P})\phi(\mathbf{P})}{\partial \mathbf{P}} + M(\mathbf{P})\frac{d\phi(\mathbf{P})}{d\mathbf{P}} = \frac{db(\mathbf{P})}{d\mathbf{P}} \tag{11}$$

Solving for $\frac{d\phi(\mathbf{P})}{d\mathbf{P}}$ and substituting into (10), we get

$$\frac{df(\mathbf{P},\phi(\mathbf{P}))}{d\mathbf{P}} = \frac{\partial f(\mathbf{P},\phi(\mathbf{P}))}{\partial \mathbf{P}} + \frac{\partial f(\mathbf{P},\phi(\mathbf{P}))}{\partial \phi} M(\mathbf{P})^{-1} \left(\frac{db(\mathbf{P})}{d\mathbf{P}} - \frac{\partial M(\mathbf{P})\phi(\mathbf{P})}{\partial \mathbf{P}}\right) \tag{12}$$

Defining now the *adjoint* vector Λ as the solution of the *adjoint* linear system

$$M(\mathbf{P})^T \Lambda = \left(\frac{\partial f(\mathbf{P},\phi(\mathbf{P}))}{\partial \phi(\mathbf{P})}\right)^T \tag{13}$$

we get the governing equation of the adjoint sensitivity method

$$\frac{df(\mathbf{P},\phi(\mathbf{P}))}{d\mathbf{P}} = \frac{\partial f(\mathbf{P},\phi(\mathbf{P}))}{\partial \mathbf{P}} + \Lambda^T \left(\frac{db(\mathbf{P})}{d\mathbf{P}} - \frac{\partial M(\mathbf{P})\phi(\mathbf{P})}{\partial \mathbf{P}}\right) \tag{14}$$

The main advantage of the adjoint method is that with only two system solves, one for the nominal system (9) and one for the adjoint system (13), we can obtain the sensitivity of $f(\mathbf{P},\phi(\mathbf{P}))$ with respect to an arbitrary number of parameters n_p. In other words, when compared with the standard direct sensitivity method the time complexity of the adjoint method is independent of the number of parameters. Furthermore, its accuracy is independent of numerical differentiation.

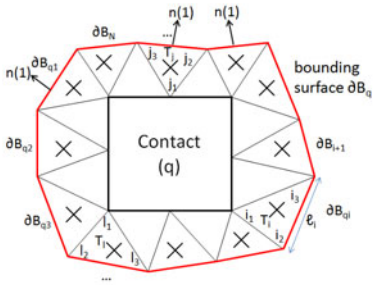

Figure 2: Hypothetical boundary surrounding the contact. Such boundary is useful for current computation.

3. RESISTANCE SENSITIVITY EXTRACTION

3.1 Defining the Output Function

Recall that the resistance $R_{pq} = \frac{V_p}{I_q}$ is a function of the total current I_q at a particular port q, which in turn is a linear function of the potential $\phi(r)$. The total derivative of the resistance with respect to the parameter vector $\mathbf{P}$ is given by

$$\frac{dR_{pq}}{d\mathbf{P}} = -\frac{V_p}{I_{pq}^2}\frac{dI_q}{d\mathbf{P}} \tag{15}$$

Consequently, the quantity of interest to extract R_{pq} is the total current at port q due to a unit potential $V_p = 1$ excitation at port p, i.e., the q-th component of the output vector is given by $f(\mathbf{P},\phi(\mathbf{P}))(q) = I_q$. Let ∂B_q be a hypothetical closed boundary surrounding port q (Fig. 2). By current continuity we know that the net current flowing through port q is the same as the net current flowing through ∂B_q. The latter is computed from the relation between the current density and the potential

$$I_q = -\int_{\partial B_q} \sigma(r)\nabla\phi(r)\cdot\hat{n}d\ell$$

where ∂B_q is constructed as the union of the sides of the triangles touching the contact q at a single point (marked with an "X" in Fig. 2). Without loss of generality, the local numbering of these triangles is made such that the point touching the contact is always numbered l_1, where l is the index of the triangle. Let the side of triangle T_l belonging to ∂B_q be referred to as ∂B_{ql}. Note that the local normal to the boundary is the unit vector in direction of $A(l_1)$,

i.e., $\hat{n} = \hat{n}(l_1) = \hat{A}(l_1)$. The I_q integral is discretized as

$$\begin{aligned} I_q &= \sum_{T_l} \int_{\partial B_{ql}} \sum_{k=1}^{3} \frac{\sigma(l)}{\alpha_{T_l}} A(l_k)\phi(l_k) \cdot \hat{A}(l_1) d\ell \\ &= \sum_{T_l} \sum_{k=1}^{3} \frac{\sigma(l)}{\alpha_{T_l}} A(l_k)\phi(l_k) \cdot \hat{A}(l_1) \int_{\partial B_{ql}} d\ell \\ &= \sum_{T_l} \sum_{k=1}^{3} \frac{\sigma(l)}{\alpha_{T_l}} A(l_k) \cdot A(l_1)\phi(l_k) \end{aligned}$$

where we have used the relation $A(l_1) = \hat{A}(l_1) \int_{\partial B_{ql}} d\ell$.

Note that due to the assumed local indexing of triangle T_l, node l_1 is on the boundary of the contact and $\phi(l_1) = 0$. This leads to

$$I_q = \sum_{T_l} \sum_{k=2}^{3} \frac{\sigma(l)}{\alpha_{T_l}} A(l_k) \cdot A(l_1)\phi(l_k)$$

which is simply the addition of the contributions of all the points on the boundary ∂B_q that are connected to points on the contact boundary. A careful investigation of this formula reveals that with the aid of (6) it can be cast in the following compact form

$$I_q = S_q \left(A_{21}\phi_1 + A_{22}\phi_2\right) \tag{16}$$

where S_q is a row vector of zeros and ones and the rest of the notation is as in (7). S_q has ones at columns corresponding to the global indices of the nodes representing port q. Equation (16) indicates that the total current depends linearly on the potential of *any* point connected to a boundary point through a common triangle. More importantly, (16) indicates that the entries of the output matrices $S_q A_{21}$ and $S_q A_{22}$, along with those of both the system matrix M and the RHS vector b all share the same formulas, i.e., they all rely on elements of the form (6). This will become very useful later on when we compute derivatives of such elements with respect to geometrical variations. Finally, (16) can be cast in a more compact linear relation between the current and potential

$$I(\phi(\mathbf{P}), \mathbf{P}) = C_1^T(\mathbf{P})\phi(\mathbf{P}) + C_2^T(\mathbf{P})\phi_2 \tag{17}$$

where $\phi(\mathbf{P})$ is the unknown potential of the non-contact nodes, and ϕ_2 is the vector of fixed potentials of the contact nodes and is of length N_f, while $C_1(\mathbf{P})$ and $C_2(\mathbf{P})$ are known parameter-dependent matrices of size $N \times n_o$ and $N_f \times n_o$, respectively. Note that the above derivation is valid *only* for ports that are assigned zero potential, i.e., $q \neq p$, where p is the index of the excited port. However this is not a limitation since only one port is assigned a nonzero potential and the self-resistance of such port is given by the sum of all the mutual resistances of the port $R_{pp} = \sum_{q=1, q\neq p}^{N_c} R_{pq}$.

Finally, the derivatives of the current function required for the adjoint equations (13) and (14) are given by

$$\begin{aligned} \frac{\partial I(\phi(\mathbf{P}), \mathbf{P})}{\partial p_i} &= \frac{dC_1^T(\mathbf{P})}{dp_i}\phi(\mathbf{P}) + \frac{dC_2^T(\mathbf{P})}{dp_i}\phi_2 \\ \frac{\partial I(\phi(\mathbf{P}), \mathbf{P})}{\partial \phi} &= C_1^T(\mathbf{P}) \end{aligned}$$

where p_m is the m-th component of $\mathbf{P}$.

3.2 Computing Derivatives w.r.t. P

In this subsection we are interested in computing the derivatives of the different matrix and vector entries with respect to the parameter vector $\mathbf{P}$. To do so we first recall that the matrix elements of the system matrix M, the right hand side vector b and the output matrices C_1 and C_2 are all computed from (6) and therefore computing

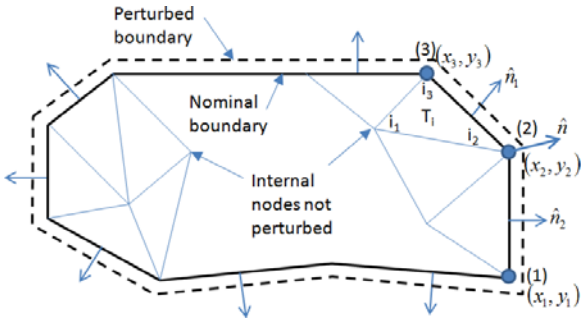

Figure 3: Typical variations defined by boundary node perturbations.

the derivative of (6) with respect to the geometrical parameters covers all the required derivatives. Second we make the observation that since all the entries of the matrices and vectors depend solely on the coordinates of the nodes, one can define all geometrical perturbations by their effect on such nodal coordinates. Using (6) the derivative of any matrix element with respect to the geometrical parameter p_m is given by

$$\begin{aligned} \frac{dA(i,j)}{dp_m} &= \sum_{k:(i,j)\in T_k} \sum_{z_\ell \in T_k} \frac{\partial a(k_i,k_j)}{\partial z_\ell} \frac{dz_\ell}{dp_m} \\ a(k_i,k_j) &= \frac{\sigma(k)}{\alpha_{T_k}} A(k_i) \cdot A(k_j) \end{aligned} \tag{18}$$

where z_ℓ is one of the six coordinates (x and y coordinates of the three nodes of the triangle) on which the term $A(i,j)$ depends. The global indices of the nodes of triangle T_k are (i,j,l) and the corresponding local indices are (k_i,k_j,k_l). The computation of $\frac{\partial a(k_i,k_j)}{\partial z_\ell}$ is illustrated by the following example. The goal is to compute $\frac{\partial a(k_i,k_j)}{\partial z_\ell}$ where $z_\ell = x_l$, the x coordinate of the l-th global node in triangle T_k. This is achieved through the following algebra steps

$$\begin{aligned} a(k_i,k_j) &= \frac{\sigma(k)}{\alpha_{T_k}} \left(\left(y_{k_j} - y_{k_l}\right)\left(y_{k_l} - y_{k_i}\right) + \left(x_{k_l} - x_{k_j}\right)\left(x_{k_i} - x_{k_l}\right) \right) \\ \frac{\partial a(k_i,k_j)}{\partial x_l} &= \frac{\sigma(k)}{\alpha_{T_k}} \left(-2x_{k_l} + x_{k_j} + x_{k_i}\right) + \frac{-\sigma(k)}{\alpha_{T_k}} \frac{\partial \alpha_{T_k}}{\partial x_{k_l}} a(k_i,k_j) \\ \alpha_{T_k} &= 0.5 \left(\left(x_{k_j} - x_{k_i}\right)\left(y_{k_l} - y_{k_i}\right) - \left(x_{k_l} - x_{k_i}\right)\left(y_{k_j} - y_{k_i}\right) \right) \\ \frac{\partial \alpha_{T_k}}{\partial x_{k_l}} &= -0.5 \left(y_{k_j} - y_{k_i}\right) \end{aligned} \tag{19}$$

Next we illustrate how to compute the chain rule factor $\frac{dz_\ell}{dp_m}$ in (18). We do so using an instance of a generic perturbation that implements uniform shape changes such as expansion or shrinking, as shown in (Fig. 3). Since this type of perturbations affects only the boundary of the structure, all the internal nodes will remain unchanged, i.e., $\frac{dz_\ell}{dp_m} = 0$ for any z_ℓ coordinate of an internal node. Only nodes defining the outer boundary will change. The direction of the boundary node perturbation is in the average direction of the normals to both boundary segments connected through the node. This is direction $\hat{n}$ in Fig. 3. As an example, we will compute $\hat{n}$, the direction of perturbation of (x_2, y_2)

$$\hat{n} = \frac{1}{||\hat{n}_1 + \hat{n}_2||} \left(\hat{n}_1 + \hat{n}_2\right)$$

where $||v||$ is the length of vector v and

$$\hat{n}_1 = \frac{1}{\sqrt{(y_3 - y_2)^2 + (x_2 - x_3)^2}} \left((y_3 - y_2)\hat{x} + (x_2 - x_3)\hat{y}\right)$$

$$\hat{n}_2 = \frac{1}{\sqrt{(y_2-y_1)^2+(x_1-x_2)^2}}\left((y_2-y_1)\hat{x}+(x_1-x_2)\hat{y}\right)$$

Consequently, the sensitivities of coordinates (x_2, y_2) to a small node perturbation p_i along the normal $\hat{n}$ are given by

$$\frac{dx_2}{dp_i} = \hat{n}\cdot\hat{x} \qquad \frac{dy_2}{dp_i} = \hat{n}\cdot\hat{y} \tag{20}$$

The mechanism suggested above for defining a perturbation is in fact general and can be used to model any geometric perturbation of either the domain boundaries or the contact locations. All that is required is to determine the set of nodes defining the perturbation, determine the changes in the coordinates of these nodes in response to a unit variation, and finally determine the partial derivatives. This process is summarized in Algorithm 1.

Algorithm 1 Efficient Assembly of Derivative Terms

1: Determine the set of nodes S_1 defining the perturbation
2: **for all** nodes in set S_1 **do**
3: create a list of the parameters on which the node depends
4: determine the partial derivatives of the node coordinates with respect to every parameter on which it depends $\frac{dz_k}{dp_i}$ (similar to (20))
5: **end for**
6: **for all** parameter p_i in parameter set **P do**
7: create a set of nodes S_2 (by global index) that depend on the parameters
8: **end for**
9: When filling matrices M, b, C_1 and C_2, assemble $\frac{dM}{dp_i}$, $\frac{db}{dp_i}$, $\frac{dC_1}{dp_i}$ and $\frac{dC_2}{dp_i}$

$$\frac{d(M,b,C_1,C_2)}{dp_i} = \sum_{z_k\in S_2} \frac{d(M,b,C_1,C_2)}{dz_k}\frac{dz_k}{dp_i} \tag{21}$$

3.3 Complexity Analysis of Sensitivity Extraction

It is well known that the FEM system matrix M is symmetric and very sparse. Moreover, by proper numbering of the nodes in the FEM mesh one can generate a banded system matrix M [7]. The maximum bandwidth B of the matrix is the maximum difference between the global indices of any interacting non-contact nodes (i.e., nodes that share a common triangle). In the remainder of this subsection, B is assumed a constant much smaller than N but in the order of both n_p and n_o. The most important observation is that the adjoint system matrix in (13) is the transpose of the symmetric matrix M. Consequently, both the linear system and the adjoint system share the same system matrix. Following all the previous observations the complete set of equations can be summarized as

$$\begin{aligned} M(\mathbf{P})\left[\begin{array}{cc}\phi(\mathbf{P}) & \Lambda\end{array}\right] &= \left[\begin{array}{cc} b(\mathbf{P}) & C_1(\mathbf{P})\end{array}\right] \\ \frac{df(\mathbf{P},\phi(\mathbf{P}))}{d\mathbf{P}} &= \frac{dC_1(\mathbf{P})^T\phi(\mathbf{P})}{d\mathbf{P}} + \frac{dC_2(\mathbf{P})^T\phi_2}{d\mathbf{P}} \\ &+ \Lambda^T\left(\frac{db(\mathbf{P})}{d\mathbf{P}} - \frac{\partial M(\mathbf{P})\phi(\mathbf{P})}{\partial\mathbf{P}}\right)\end{aligned}$$

The complexity of solving the first equation is that of solving the same nominal sparse linear system with multiple right hand sides. The number of right hand sides is equal to $1+n_o$. Therefore, the complexity of solving all systems concurrently using Gaussian elimination is $O(B^2N)$. In other words, the complexity of our method is independent of the number of outputs, and indeed, we have avoided one of the main pitfalls of the adjoint sensitivity method, namely, the linear growth of the complexity as a function of the number of outputs. The Gaussian elimination complexity is inherited from the solution of the nominal system and therefore the incremental complexity of solving both systems as compared to solving only the original system is insignificant. The added complexity of forming $\frac{dC_{1,2}(\mathbf{P})^T\phi(\mathbf{P})}{d\mathbf{P}}$, which is n_o matrix-vector products, is $O(n_o n_p)$ due to the sparsity of matrices C_1 and C_2. Finally, the complexity of forming the term $\frac{\partial M(\mathbf{P})\phi(\mathbf{P})}{\partial\mathbf{P}}$, which is n_p sparse matrix -vector products, is $O(Nn_p)$. The total complexity is $O(B^2N+n_on_p+Nn_p)$, which is $O(B^2N)$, i.e., it is the exact same complexity as solving only the nominal system.

The memory complexity can also be shown to be $O(B^2N)$, i.e., exactly the same memory complexity required

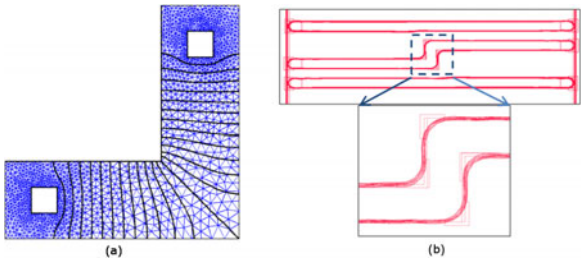

Figure 4: a) A simple 2-port rectangular corner. Equipotential contours clearly visible. b) 2-port jog. Different litho generated contours clearly visible.

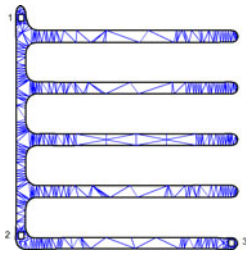

Figure 5: Three-port comb structure.

4. RESULTS

All results are obtained from a C implementation of the algorithm on a PowerPC workstation running at slightly more than 1GHz with 16GB of RAM.

4.1 Accuracy Validation: Rectangular Corner

The first example is that of a rectangular corner Fig. 4.a. Each side of the corner is made of 9×3 squares, where the side of the square is equal to the minimum width per the design rules. The port size is 1 square and is centered at a point 1.5 square away from the corners. In this calculation the objective is to illustrate the accuracy of our FEM solver. The structure is discretized with 5000 triangles. The calculated resistance between the two ports of the corner is 0.807Ω. Such values are correlated with hardware using delay measurements on ring oscillator structures. Fig. 4.a shows the structure, the discretization and some of the equipotential surfaces.

4.2 Jog and Multiport Comb Structures

The second and third examples are that of a 2-port jog (Fig. 4.b) and a 3-port 5-finger comb structure (Fig. 5), respectively. Note that the structure boundaries do not follow a Manhattan pattern. They are rather described by piecewise linear approximations. The jog is

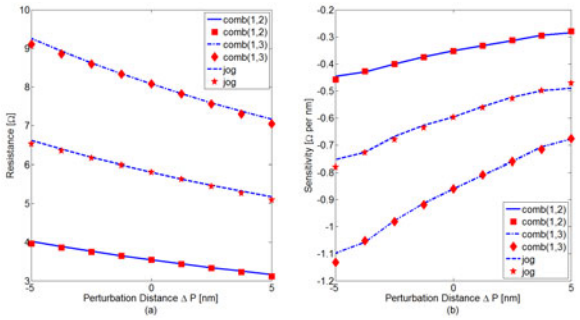

Figure 6: a) Resistance of the comb and jog structures as a function of the perturbation distance. The lines represent reference FEM results, the dots represent resistance calculated with extracted FEM sensitivities. b) Sensitivity of the resistances of the comb and jog structures w.r.t. structural perturbations. The lines represent sensitivities calculated using direct FD method, the dots represent extracted FEM sensitivities.

discretized using 5000 triangles, while the comb is discretized using 15,000 triangles. We extract the resistance between the ports of the jog and the resistance from port 1 to the other two ports (2 and 3) of the comb structure. We also extract the sensitivity of such resistances with respect to orthogonal uniform shape variations, i.e., variations which result in the boundary contours being uniformly offset from their nominal location along the normal to the boundary by distances ranging from -5nm to +5nm. This corresponds to 13% relative variation in the minimum dimension. This type of variation is illustrated in Fig. 3.b and was described mathematically by (20). The total extraction time (including resistances and sensitivities) for the jog and the comb drive is 0.8sec and 2.9sec, respectively. We further extract the resistances and sensitivities of both structures for 9 different boundary contour realizations corresponding to -5, -3.75, -2.5, -1.25, 0, 1.25, 2.5, 3.75, 5 nm perturbation from the nominal geometry. [1] The resistances of such different realizations are used to approximate the derivatives via finite difference approximations.

Fig. 6.a shows the excellent agreement between the resistances computed with the first order gradient model (1) and the FEM extracted sensitivity on the one hand (*red dots*) and on the other hand the resistances extracted directly from the finite element program (*blue lines*) for the structures under test. The maximum relative discrepancy was about 2 % and attained as expected at the end points of the perturbation interval. As apparent from Fig. 6.a, the resistance function can be accurately approximated using a first-order gradient model even for relative variations as high as 13%. This result is a nice illustration of the value of sensitivity extraction.

Fig. 6.b also shows the excellent agreement between the FEM-based sensitivities (*red dots*) and the sensitivities computed with an FD formula using the 9 sample points mentioned above (*blue lines*). Finally, it is worth noting that the cost of the first order gradient model is 1.5× the nominal extraction, while that of the full FD alone is 9× the nominal extraction.

5. CONCLUSIONS

In this paper we have presented a complete framework for extracting the resistance sensitivities to geometrical variations of conductors of arbitrary shapes. The sensitivity algorithm is based on the adjoint method and has been fully integrated into an industrial FEM resistance solver. Due to the sparsity and symmetry of the FEM formulation, the complexity of the sensitivity extraction is overshadowed by that of the resistance calculation. The overall CPU complexity remains therefore unchanged. Furthermore, because of the sparsity of the FEM method and the ease of defining perturbations through the exclusive usage of boundary elements, there is negligible memory utilization associated with our algorithm. We have demonstrated the validity and efficiency of our algorithm on a variety of examples of industrial relevance. Future work will address how such resistance sensitivities can be used in a litho-aware VLSI extraction flow.

[1] This way of perturbing the structure is physically meaningful as it mimics the way lithographic process variations impact the shape of nominal boundaries of the conductors.

Acknowledgments

The authors would like to acknowledge very helpful discussions with Aditya Bansal, Koushik Das, Fook-Luen Heng, Mark Lavin, Rama Singh, and Amith Singhee from IBM Research. T. El-Moselhy would like to acknowledge IBM financial support in the form of a PhD Fellowship from IBM Research.

6. REFERENCES

[1] M. Horowitz, and R. W. Dutton, "Resistance Extraction from Mask Layout Data", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 2, Issue 3, Jul. 1983, pp. 145–150.

[2] A. Vithayathil, X. Hu, and J. White, "Substrate Resistance Extraction using a Multi-Domain Surface Integral Formulation" *International Conference on Simulation of Semiconductor Processes and Devices, SISPAD 2003*, pp. 323-326.

[3] X. Wang, W. Yu, and Z. Wang, " Efficient Direct Boundary Element Method for Resistance Extraction of Substrate With Arbitrary Doping Profile" *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, Issue 12, Dec. 2006, pp. 3035-3042.

[4] B. Yang, and H. Murata, "A Finite Element-Domain Decomposition Coupled Resistance Extraction Method with Virtual Terminal Insertion" *Midwest Symposium on Circuits and Systems, MWSCAS 2007*, pp. 1425-1428.

[5] S. Rajagopalan, and S. Batterywala, "A 3-Dimensional FEM Based Resistance Extraction" 20th *International Conference on VLSI Design, 2007*, pp. 565-570.

[6] M. Oppeneer, P. Sumant, and A. Cangellaris, "Robust Iterative Finite Element Solver for Multi-Terminal Power Distribution Network Resistance Extraction" *Electrical Performance of Electronic Packaging IEEE-EPEP, 2008*, pp. 181-184.

[7] M. Sadiku, *Numerical Techniques in Electromagnetics*, Boca Raton : CRC Press, 2000.

[8] C. Sakkas, "Potential Distribution and Multi-Terminal DC Resistance Computations for LSI Technology" *IBM Journal of Research and Development*, Vol. 23, No. 6, Nov. 1979, pp. 640-651.

[9] Q. Chen, and N. Wong, "A Stochastic Integral Equation Method for Resistance Extraction of Conductors with Random Rough Surfaces" *ISPACS '06*, pp. 411-414.

[10] I. Park, J. Coulomb, and S. Hahn, "Implementation of Continuum Sensitivity Analysis with Existing Finite Element Code" *IEEE Trans. on Magnetics*, Vol. 29, No.2, pp. 1787-1790, 1993.

[11] H. Qu, L. Kong, Y. Xu, X. Xu and Z. Ren, "Finite-Element Computation of Sensitivities of Interconnect Parasitic Capacitance to the Process Variation in VLSI" *IEEE Trans. on Magnetics*, Vol. 44, No.6, pp. 1386-1389, 2008.

[12] T. El-Moselhy, I. Elfadel and D. Widiger, "Efficient Algorithm for the Computation of On-Chip Capacitance Sensitivities with respect to a Large Set of Parameters" *ACM/IEEE Design Automation Conference, DAC 2008*, pp. 906-911.

[13] Y. Bi, N. van der Meijs and D. Ioan, "Capacitance Sensitivity Calculation for Interconnects by Adjoint Field Technique" 12th *IEEE Workshop on Signal Propagation on Interconnects, 2008*, pp. 1-4.

[14] F. H. Branin, Jr., "Network Sensitivity and Noise Analysis Simplified" *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 285-288, 1973.