SIAM J. IMAGING SCIENCES
Vol. 7, No. 4, pp. 2657–2674

A Stochastic Algorithm Based on Fast Marching for Automatic Capacitance Extraction in Non-Manhattan Geometries*

Francisco Bernal†, Juan A. Acebrón‡, and Immanuel Anjam§

Abstract. We present an algorithm for two- and three-dimensional capacitance analysis on multidielectric integrated circuits of arbitrary geometry. Our algorithm is stochastic in nature and as such fully parallelizable. It is intended to extract capacitance entries directly from a pixelized representation of the integrated circuit (IC), which can be produced from a scanning electron microscopy image. Preprocessing and monitoring of the capacitance calculation are kept to a minimum, thanks to the use of distance maps automatically generated with a fast marching technique. Numerical validation of the algorithm shows that the systematic error of the algorithm decreases with better resolution of the input image. Those features render the presented algorithm well suited for fast prototyping while using the most realistic IC geometry data.

Key words. capacitance extraction, fast marching, floating random walk, non-Manhattan IC, SEM image segmentation

AMS subject classifications. 65C05, 68U20, 68U10

DOI. 10.1137/140961328

1. Introduction.

As an ever larger number of electronic components are packed closer together in integrated circuits (ICs), there is an interest in tools for detecting faulty designs prior to manufacture. Typically, these flaws are due to the effect of parasitic resistances, capacitances, and inductances in IC interconnects. For more background on parasitic extraction, see [11]. In this paper, we put forward a new method, stochastic in nature, for capacitance extraction in ICs. Inside an IC, $M \gg 1$ electrodes (the electronic components and interconnects) are embedded, each of which stands at a fixed voltage $V_1, \ldots, V_M$, and which are separated from one another by means of dielectric insulators. Assuming the IC to be isolated, the capacitance matrix of the system is defined as the set of coefficients C_{ij} such that

$$Q_i = \sum_{j=1}^{M} C_{ij} V_j, \qquad i = 1, \ldots, M, \tag{1}$$

where $Q_1, \ldots, Q_M$ are the electrodes' charges. Regardless of the dielectric environment, the capacitance matrix has a number of symmetries which reflect the underlying properties of the

*Received by the editors March 18, 2014; accepted for publication (in revised form) September 2, 2014; published electronically December 16, 2014. This research was supported by the Portuguese FCT under grant PTDC/EIA-CCO/098910/2008, FCT grant SFRH/BPD/79986/2011 (FB), and grant PEst-OE/EEI/LA0021/2013 (JAA).
http://www.siam.org/journals/siims/7-4/96132.html

†Center for Mathematics and Its Applications, Instituto Superior Técnico, 1049-001 Lisbon, Portugal (francisco.bernal@ist.utl.pt).

‡ISCTE-Instituto Universitário de Lisboa Departamento de Ciências e Tecnologias de Informação, 1649-026 Lisbon, Portugal, and INESC-ID/IST, TU Lisbon, 1000-029 Lisbon, Portugal (juan.acebron@ist.utl.pt).

§Department of Mathematical Information Technology, University of Jyväskylä, FI-40014 Jyväskylä, Finland (immanuel.anjam@jyu.fi).

electrostatic field [4]. We list them here for future reference:

- $C_{ij} = C_{ji}$ because the electrostatic field is irrotational.
- $\sum_{j=1}^{M} C_{ij} = 0$ $(i = 1, \ldots, M)$ due to the global conservation of charge: the total charge of the isolated system is zero.
- $\sum_{i=1}^{M} C_{ij} = 0$ $(j = 1, \ldots, M)$ as a consequence of the previous properties.

A straightforward way to extract the capacitance matrix is to solve the electric potential u inside the IC, compute the electric field $\mathbf{E} = -\epsilon \nabla u$ ($\epsilon > 0$ is the permittivity), and calculate the charge on each electrode according to Gauss's theorem:

$$Q = \oint_{\omega} (\mathbf{E} \cdot \mathbf{N}_{\omega}) \, \mathrm{d}\mathbf{x}^{d-1}, \tag{2}$$

where ω is any closed surface in $\mathbb{R}^d$ $(d = 2, 3)$ separating the considered electrode from the rest, and $\mathbf{N}_{\omega}$ is the outward unit normal. The entries of the capacitance matrix can then be approximated by numerical differentiation,

$$C_{ij} = \partial Q_i / \partial V_j, \tag{3}$$

which involves $M(M+1)/2$ solutions of the electrostatic potential with a field solver such as those based on finite elements (FEM) or boundary elements (BEM). (This assumes that all the symmetries of the capacitance matrix have been taken into account.) For realistic problems, it takes a parallel computer to handle the sheer size of the discretization of the IC. Unfortunately, deterministic methods such as FEM and BEM are ill-suited to parallelization. The reason is that with deterministic methods, the IC discretization is partitioned among the many processors in the parallel computer. Since the solution is global (not pointwise such as in stochastic algorithms, as will be explained below), there is unavoidable interprocessor communication, which brings about an inherent lack of scalability due to Amdahl's law [5]. This turns out to be particularly dramatic in three-dimensional (3D) problems, when accuracy in describing the fringe fields calls for geometrical detail and thus very fine meshes. This is why stochastic methods for capacitance analysis are an often-preferred alternative. Let us highlight the main advantages:

- These methods are matrix-free. Thanks to this, the computational cost grows roughly linearly with the problem size (rather than with an exponent between 2 and 3 as with FEM).
- Since they rely on averages over independent realizations, they are intrinsically parallelizable, thus allowing an optimal use of a parallel computer.
- They can be efficiently used to extract the capacitance of a large number of topologically different, but geometrically equivalent, configurations [6].

On the other hand, it is well known that stochastic methods have relatively poor accuracy compared to deterministic methods. Nonetheless, in capacitance extraction one is typically satisfied (even with deterministic methods) with "10% accuracy as long as the simulation can run overnight" [21], which is within the capabilities of stochastic methods. The seminal stochastic algorithm for capacitance extraction is the floating random walk (FRW) put forward by Le Coz and Iverson [14]. It is based on the Green's function in a square for the gradient of a harmonic function. An IC is said to have a Manhattan layout if its electrodes and dielectric

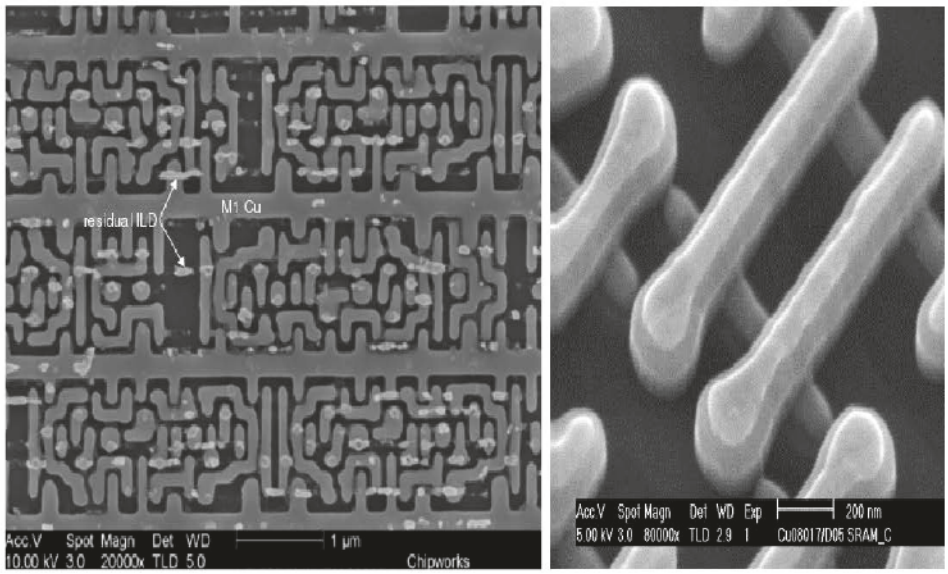

Figure 1. *SEM images of commercial SRAM memories. Non-Manhattan geometries are evident at the microscale.*

interfaces are either parallel or perpendicular to one another (i.e., form only straight angles). As long as the IC is Manhattan, the FRW directly yields the pointwise component of the electric field needed in (2). In a later paper [13], a comparison is made between the FRW and mesh-based methods, and the accuracy and complexity of the former were found to be satisfactory.

The FRW has been much improved since [10, 22], but the efficient handling of non-Manhattan designs remains inadequate. It must be stressed that Manhattan geometries are just a useful idealization of the IC, for there may be individual components which are not rectangular. Moreover, photolithography and etching bring about small irregularities in the shape of the manufactured electrodes or on the permittivity interfaces. Such departures from the Manhattan model cannot be ignored as the typical geometrical length shrinks down to the microscale, as shown in the scanning electron microscopy (SEM) images in Figure 1. To illustrate this point further, we give a toy numerical example. The areas below the curves in Figure 2 are proportional to the capacitance of two systems of two electrodes each: the rounded set's capacitance is significantly smaller than that of the Manhattan set. Physically, this is due to the larger overlap of the electrodes in the second set and to the spikes in the electric field close to sharp corners. The same effect has been reported in more complicated, 3D simulations [23]. In general, the capacitances for submicron structures are highly dependent on process geometry especially for multidielectric technologies. An extension of the FRW to general geometries has been presented [2], albeit restricted to single-permittivity designs, and apparently it has not been further developed since.

Another important aspect which is not explicitly included in many FRW-type algorithms is the preprocessing overhead claimed by the mathematical representation of the IC. The

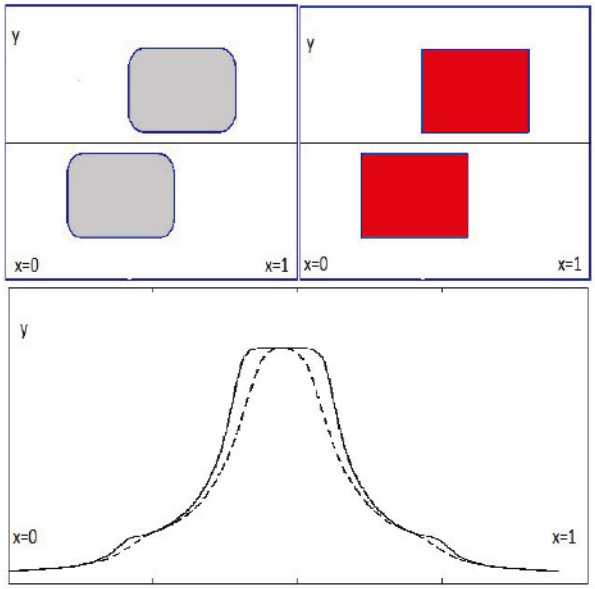

Figure 2. *Influence of sharp corners on the capacitance. Both systems of two electrodes each are isolated so that the capacitance is proportional to the integral of the flux of the electric field along the midline (shown). The area below the solid curve (corresponding to the sharp-cornered set) is about 1/6 larger than that under the dashed curve (blunt corners).*

FRW requires the largest rectangle tangent to the electrodes at any dielectric point inside the IC (see [1] for appropriate algorithms). The situation is more complicated in the presence of dielectric interfaces, and the performance of the FRW tends to degrade [13, 24]. Non-Manhattan geometries require a level set (contour map of constant values) distance map, which is the solution of several eikonal equations inside the IC. Moreover, in the submicron scale, one is naturally interested in incorporating information gained from SEM data, such as that in Figure 1, into the IC representation.

In this paper, we present an algorithm which aims to extract capacitances (almost) directly from image data of the IC. Edges on the input images must be sufficiently well resolved so that image processing software can segment them into conductor and dielectric regions. See Figure 3 for a simple illustration and [15] for more in-depth background. Based on this pixelization of the IC, our algorithm then proceeds to efficiently compute a set of distance maps inside the IC by means of the fast marching method [20]. This enables the automatic (i.e., unsupervised) construction of the Gaussian surfaces in (2) by offsetting the electrode shape into the dielectric regions. Finally, the fact that so-constructed Gaussian surfaces are a level set of the distance map allows one to compute the electric field with a single-point evaluation of the potential (details will be given in section 2), meaning that our algorithm's cost has the same order of magnitude as the FRW or Brambilla's method [2]. Pointwise evaluation of the potential is done quickly and efficiently by means of the walk on spheres method [19], which can handle completely arbitrary shapes of electrodes and dielectric interfaces. With a 3D digital input (based on design data, SEM slicing [9], or reconstruction of the manufactured IC [17]), our algorithm could be used for capacitance extraction in 3D IC models as well.

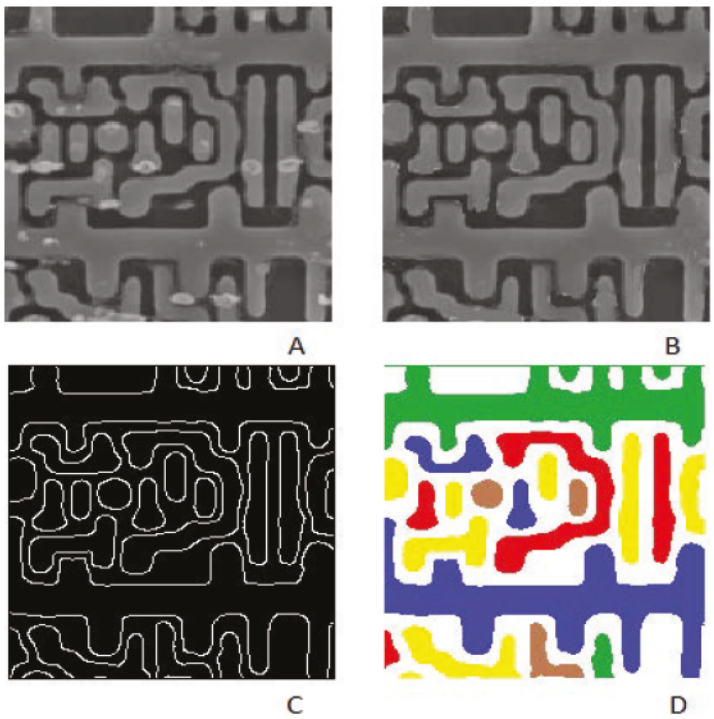

Figure 3. *Illustration of the preprocessing of a SEM image into an IC pixelization which can be used as input:* (A) *Original image (top left section of Figure 1).* (B) *"Polished" image after cleaning it (here, by hand).* (C) *Edge detection using Canny's filter.* (D) *Final segmentation into electrodes. The side of the square is about* $1\ \mu m$.

The remainder of this paper is organized as follows. In section 2, we present our stochastic method to estimate the entries of the capacitance matrix. Section 3 presents an efficient way to produce and handle the critical distance maps from IC pixelizations, as well as a summary of the numerical errors incurred and some programming notes. The algorithm is validated numerically in section 4. Preliminary conclusions are given in section 5.

2. Rowwise extraction of the capacitance matrix.

Electrostatic potential. Mathematically, the problem is to solve the electrostatic potential $u : \Omega \subset \mathbb{R}^d \mapsto \mathbb{R}$ in $d = 2, 3$, where Ω (the dielectric part of the IC) is a connected domain with boundary $\partial\Omega = \cup_{i=0}^{M}\Gamma_0$ such that the Γ_i do not intersect each other. The exterior boundary is Γ_0, on which a homogeneous Neumann boundary condition (BC) applies [14], meaning that the system is isolated. The M "holes" $\Gamma_{i\neq 0}$ represent the electrode shapes and act as piecewise constant Dirichlet BCs [14] since on each of them a voltage V_i, $i = 1, \ldots, M$, is set. We also consider a piecewise constant (but not necessarily layered) permittivity, so that

$$\Omega = \cup_{i=1}^{p}\Omega_i \text{ such that } \epsilon(\mathbf{x} \in \Omega_i) = \epsilon_i > 0,\ i = 1, \ldots, p, \tag{4}$$

where the Ω_i represent the p dielectric regions of homogeneous permittivity. Let us also call Λ_{ij} the permittivity interface between regions i and j, whose outward unit normal $\mathbf{N}_{\Lambda_{ij}}$ points

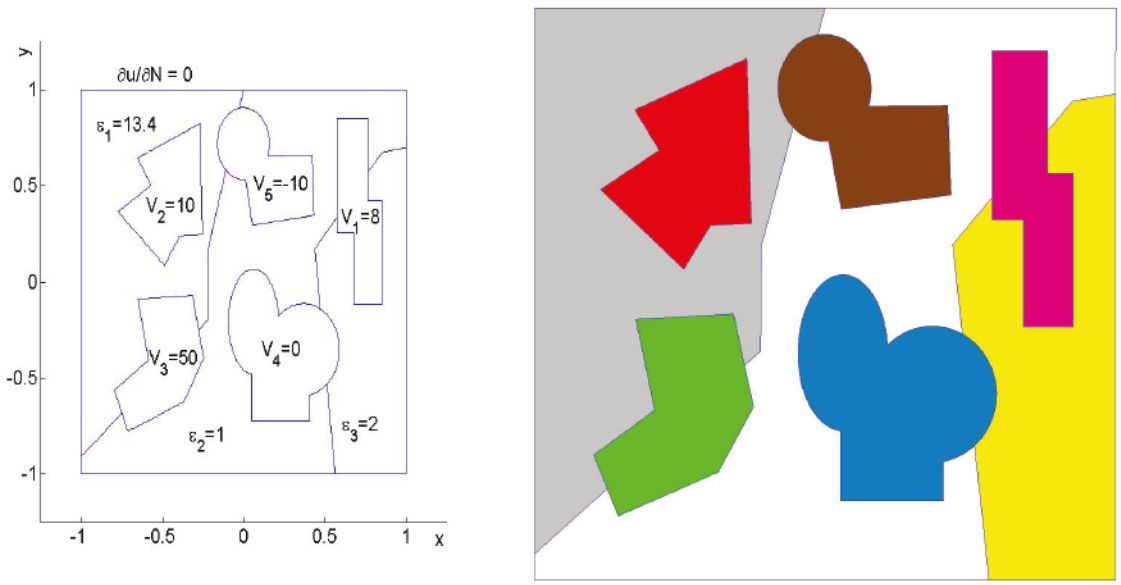

Figure 4. Left: *Synthetic two-dimensional (2D) non-Manhattan model of an IC with $M = 5$ electrodes and $p = 3$ nonlayered permittivity regions, showing some labels for illustration.* Right: *The 1268×1268 bitmap image serving as input file. Each electrode and disjoint permittivity region must have a different color. From this, the distance maps on Figure 5 are generated.*

into region j. See Figure 4 (left) for clarification. For notational convenience we allow the ambiguity $\Lambda_{ij} = \Lambda_{ji}$ but note that $\mathbf{N}_{\Lambda_{ij}} = -\mathbf{N}_{\Lambda_{ji}}$. If regions i and j are not in contact, then $\Lambda_{ij} = \emptyset$. Every region Ω_k is in contact with a portion of at least one permittivity interface. The potential can then be expressed as $u = u_1 + u_2 + \cdots + u_p$, which are the solution of the following system of coupled Laplace's equations for $k = 1, \ldots, p$:

$$
(5) \qquad \begin{cases} u_k = 0 & \text{if } \mathbf{x} \notin \Omega_k, \\ \nabla^2 u_k = 0 & \text{if } \mathbf{x} \in \Omega_k, \\ u_k = V_j & \text{if } \mathbf{x} \in \partial\Omega_k \cap \Gamma_j, \quad j = 1, \ldots, M, \\ \nabla u_k \cdot \mathbf{N}_{\Gamma_0} = 0 & \text{if } \mathbf{x} \in \partial\Omega_k \cap \Gamma_0, \\ \epsilon_k \nabla u_k \cdot \mathbf{N}_{\Lambda_{ki}} = \epsilon_i \nabla u_i \cdot \mathbf{N}_{\Lambda_{ki}} & \text{if } \mathbf{x} \in \Lambda_{ki}, \qquad i = 1, \ldots, p. \end{cases}
$$

The last BC represents the continuity of the normal component of the electric field across the permittivity interfaces: $\mathbf{E}|^+_{\Lambda_{ij}} = \mathbf{E}|^-_{\Lambda_{ij}}$. Notice that it is this BC which couples the system together.

Walk on spheres. It is well known [18] that the equation $\nabla^2 u = 0$ with Dirichlet BCs $u(\mathbf{x} \in \partial\Omega) = g(\mathbf{x})$ admits the probabilistic representation

$$
(6) \qquad u(\mathbf{x}) = \mathbb{E} g(\mathbf{X}_\tau) \approx (1/N) \sum_{j=1}^{N} g(\mathbf{X}_\tau^j),
$$

where $\mathbf{X}^j(t)$ is the trajectory of a particle starting at $\mathbf{x}$ and diffusing isotropically inside Ω until hitting $\partial\Omega$ for the first time at $t = \tau^j$ on $\mathbf{X}_\tau^j$. Numerically, the optimal method for

approximating the expected value above is the so-called walk on spheres (WoS) [19]. The mechanics of the WoS can be visualized as the particle jumping from its current position at $\mathbf{X}_k$ (we omit the superindex j) to some random point $\mathbf{X}_{k+1}$ on the surface of the largest ball contained in the domain and centered at $\mathbf{X}_k$, hence the name of the algorithm. In order for the number of such jumps to be finite, an absorption band of width $\delta > 0$ must be set, so that when the particle enters it, its projection on $\partial\Omega$, defined as $\mathbf{X}^B$, is an approximation to the exit point $\mathbf{X}_\tau$. For Dirichlet BCs and under technical assumptions (see [19]), particle trajectories take $O(1/\delta)$ steps on average to hit the boundary, and the WoS has an error:

$$\varepsilon_{WoS}(\mathbf{x}_0, N, \delta) \leq q\sqrt{\frac{var[v(\mathbf{X}_\tau)]}{N}} + K\delta, \tag{7}$$

where $q = 1, 2, 3, \ldots$, $var[v(\mathbf{X}^B)]$ is the variance of the scores, and $K > 0$ is a constant independent of N and δ. The first summand on the right-hand side of (7) is the statistical error (with probability $P_q = 68.3\%, 95.5\%, 99.7\%, \ldots$), which is due to the replacement of the expected value by a mean. The second is the bias, due to the fact that the boundary is shrunk by an amount δ. Thus there is a tradeoff between computational cost and bias. The WoS can be adapted to provide a stochastic solution to (5). The idea is as follows: while the particle is farther away than the absorption threshold δ from all boundaries and permittivity interfaces, it evolves according to the standard WoS. If the particle hits the zero-flux external boundary Γ_0 or any permittivity interface, it takes special steps, which will be defined later. Finally, if the particle hits an electrode surface Γ_i, $i = 1, \ldots, M$, it is stopped and contributes V_i (i.e., the value of the potential on Γ_i) to the mean in (6). Because in this particular problem the Dirichlet BCs are piecewise constant, it is just the electrode eventually hit by the particle which is required, rather than the approximation $\mathbf{X}^B$ to the exit point $\mathbf{X}_\tau$, thus removing a source of error. The modified WoS algorithm for (5) (see Algorithm 1) is similar to that used in [16] to solve a biochemical problem.

Homogeneous Neumann BCs. On Γ_0, the homogeneous Neumann BC there can be handled as follows:

$$\mathbf{X}_{k+1} = \mathbf{X}_k^B - \gamma \mathbf{N}_k. \tag{8}$$

The bouncing of the trajectory off Γ_0 can be justified by "randomizing" a finite difference approximation to the BC: $\partial u/\partial N|_{\Gamma_0} \approx \big(u(\mathbf{X}_k^B) - u(\mathbf{X}_k^B - \gamma \mathbf{N}_k)\big)/\gamma = 0$, so that the sought-for score is $u(\mathbf{X}_k^B) = u(\mathbf{X}_{k+1})$ as in (8). The bouncing length $\gamma > \delta$ is a parameter. This procedure has a bias $O(\gamma)$ [16].

Permittivity interfaces. This BC can analogously be tackled by randomization of finite differences. Upon hitting an interface Λ_{ij} from the permittivity region i, the trajectory takes a step of fixed length γ (the same as in (8), for simplicity) along the direction of $\mathbf{N}_{\Lambda_{ij}}(\mathbf{X}_k^{\Lambda_{ij}})$ into either Ω_i or Ω_j according to the rule

$$\mathbf{X}_{k+1} = \begin{cases} \mathbf{X}_k + \gamma \mathbf{N}_{\Lambda_{ij}}(\mathbf{X}_k^{\Lambda_{ij}}) & \text{with probability } \frac{\epsilon_i}{\epsilon_i + \epsilon_j}, \\ \mathbf{X}_k - \gamma \mathbf{N}_{\Lambda_{ij}}(\mathbf{X}_k^{\Lambda_{ij}}) & \text{with probability } \frac{\epsilon_j}{\epsilon_i + \epsilon_j}, \end{cases} \tag{9}$$

where $\mathbf{X}_k^{\Lambda_{ij}}$ is the projection of $\mathbf{X}_k$ on Λ_{ij}. This rule is justified by the observation that, for $\mathbf{x} \in \Lambda_{ij}$, Taylor expansions in the normal direction are

$$\begin{bmatrix} \epsilon_i \\ \epsilon_j \end{bmatrix} u(\mathbf{x} \pm \gamma \mathbf{N}_{\Lambda_{ij}}(\mathbf{x})) = \begin{bmatrix} \epsilon_i \\ \epsilon_j \end{bmatrix} u(\mathbf{x}) \pm \gamma \begin{bmatrix} \epsilon_i \\ \epsilon_j \end{bmatrix} [\nabla u(\mathbf{x})]^{\pm} + O(\gamma^2), \tag{10}$$

which, together with the continuity of the normal component of the electric field, yields to first order in γ

$$u(\mathbf{x}) = \frac{\epsilon_i}{\epsilon_i + \epsilon_j} u\big(\mathbf{x} + \gamma \mathbf{N}_{\Lambda_{ij}}(\mathbf{x})\big) + \frac{\epsilon_j}{\epsilon_i + \epsilon_j} u\big(\mathbf{x} - \gamma \mathbf{N}_{\Lambda_{ij}}(\mathbf{x})\big), \tag{11}$$

which suggests splitting (in a statistical sense) the trajectory of the particle as in (9) [16].

Algorithm 1. Pointwise potential in IC by modified WoS.

Require: $\mathbf{x}_0 \in \Omega \subset \mathbb{R}^d$, $N \gg 1$, $\gamma > \delta > 0$
Require: ν_d is a homogeneous distribution on a ball in $\mathbb{R}^d$.
  for $n = 1, \dots, N$ **do**
    $k = 0$, $\mathbf{X}_k^n = \mathbf{x}_0$
    while $r_0 = \text{distance}(\mathbf{X}_k^n, \partial\Omega \backslash \Gamma_0) > \delta$ **do**
      if $r_1 = \text{distance}(\mathbf{X}_k^n, \Gamma_0) < \delta$ **then**
        compute $\mathbf{X}_{k+1}^n$ according to γ and (8)
      else if $r_2 = \min_{1 \le i \le p} \text{distance}(\mathbf{X}_k^n, \Lambda_{ij}) < \delta$ **then**
        compute $\mathbf{X}_{k+1}^n$ according to γ and (9)
      else
        $\mathbf{X}_{k+1}^n = \mathbf{X}_k^n + \min(r_0, r_1, r_2)\nu_d$
        $k = k + 1$
      end if
    end while
    $I = \arg\min_{1 \le i \le M} \text{distance}(\mathbf{X}_k^n, \Gamma_i)$
    $[score]_n = V_I$
  end for
  $u(\mathbf{x}_0) \approx \frac{1}{N} \sum_{n=1}^{N} [score]_n$

Electrostatic field and electrodes' charge. Once again we can exploit the fact that the electrodes are isosurfaces (isolines in two dimensions) of the potential to approximate the directional derivative in (2) by means of central finite differences with a single computation of the potential (and less statistical error):

$$\nabla u\left(\mathbf{y} - \frac{h\mathbf{N}_{\Gamma_k}(\mathbf{y})}{2}\right) \cdot \mathbf{N}_{\Gamma_k}(\mathbf{y}) \approx \frac{u(\mathbf{y}) - V_k}{h} + O(h^2). \tag{12}$$

Therefore, if the Gaussian integration surface ω_k in (2) is obtained by offsetting the electrode's shape Γ_k a small distance $h > 0$ toward the interior of the IC (see Figure 6), then

$$Q_k \approx \frac{1}{h} \oint_{\omega_k} \epsilon\left(\mathbf{x} - \frac{h\mathbf{N}_{\omega_k}}{2}\right) [V_k - u(\mathbf{x})] \, \mathrm{d}\mathbf{x}^{d-1} \qquad \text{if} \quad \text{distance}(\omega_k, \Gamma_k) = h. \tag{13}$$

In the integral (13), dx^{d-1} is the Jacobian of a parametrization $\chi : \mathbb{R}^{d-1} \mapsto \omega_k \subset \mathbb{R}^d$ (so that $dS := \mathrm{dx}^2$ and $dl := \mathrm{dx}^1$ are the differential surface element and arc length, respectively). Note that the permittivity is not evaluated on ω_k, but halfway between ω_k and Γ_k. For practical purposes, (13) can be substituted with a quadrature based on n_k quadrature nodes:

$$Q_k \approx \frac{\oint_{\omega_k} \mathrm{dx}^{d-1}}{h} \sum_{i=1}^{n_k} w_i \epsilon \left(\mathbf{x}_i - \frac{h \mathbf{N}_{\omega_k}(\mathbf{x}_i)}{2} \right) [V_k - u(\mathbf{x}_i)], \tag{14}$$

where $\mathbf{x}_i \in \omega_k$ and w_i is the weight of node i in the quadrature scheme. In two dimensions, (14) becomes a closed line integral, and $L_k := \oint_{\omega_k} dl$ is the length of the integration curve ω_k. Considering trapezoidal quadrature, if there are n_k equispaced subintervals between the quadrature nodes $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_k}, \mathbf{x}_1$ (notice that the first and last quadrature nodes are one and the same),

$$Q_k^{(2D)} \approx \frac{L_k}{n_k h} \sum_{i=1}^{n_k} \epsilon \left(\mathbf{x}_i - \frac{h \mathbf{N}_{\omega_k}(\mathbf{x}_i)}{2} \right) [V_k - u(\mathbf{x}_i)]. \tag{15}$$

The error involved in the quadrature (15) is of order $(L_k/n_k)^2$.

Capacitance matrix. Let us define $\epsilon_i := \epsilon(\mathbf{x}_i - h\mathbf{N}_{\omega_k}(\mathbf{x}_i)/2)$ with $i = 1, \ldots, n_k$. Inserting (6) into (14),

$$Q_k \approx \frac{\oint_{\omega_k} \mathrm{dx}^{d-1}}{h} \sum_{i=1}^{n_k} w_i \epsilon_i \left(V_k - \frac{1}{N_i} \sum_{j=1}^{N_i} V_i^j \right), \tag{16}$$

where V_i^j, $j = 1, \ldots, N_i$, labels the score of the individual trajectories. Denoting by μ_{im} the number of trajectories which originate at the quadrature node $\mathbf{x}_i$ of ω_k and hit electrode m, and applying (3) to (16),

$$C_{km} \approx \frac{\oint_{\omega_k} \mathrm{dx}^{d-1}}{h} \sum_{i=1}^{n_k} w_i \epsilon_i \left(\delta_{km} - \frac{\mu_{im}}{N_i} \right), \qquad \delta_{km} := \begin{cases} 1 & \text{if } k = m, \\ 0 & \text{if } k \neq m; \end{cases} \tag{17}$$

i.e., all the entries of the kth row of the capacitance matrix are obtained simultaneously. If ϵ is constant, $\sum_{j=1}^{M} C_{ij} = 0$ (but this need not be the case otherwise, due to the displacement charge inside ω_k). One further advantage of this algorithm is that, if the quantities $\sum_{i=1}^{n_k} (w_i \epsilon_i \mu_{im}/N_i)$ and $\sum_{i=1}^{n_k} w_i \epsilon_i$ are stored for each electrode, the new capacitance entries can be obtained for a different set of voltages without rerunning the complete simulation. In two dimensions and assuming trapezoidal quadrature as before, (17) becomes

$$C_{km}^{(2D)} \approx \frac{L_k}{n_k h} \sum_{i=1}^{n_k} \epsilon_i \left(\delta_{km} - \frac{\mu_{im}}{N_i} \right). \tag{18}$$

3. Generation and manipulation of distance maps.

Representation of the IC geometry and input file. The IC geometry Ω is represented as a d-dimensional array of integers, which we call $[INPUT]$, whose entries label the electrode/permittivity region at the entry's geometrical location in the IC. The array $[INPUT]$ must be provided as input, along with a separate table $T(l)$ which maps the entry labels $\{l\}$ to specific electrode voltages and permittivities. In two dimensions, the input file (a matrix) is a bitmap image file, with as many entries as pixels (see Figures 3 and 4). There, color 249 (red) corresponds to $V = T(249) = 10$, color 7 (gray) stands for $\epsilon = T(7) = 13.4$, and so on. Notice that two electrodes with the same voltage (as two permittivity regions with the same permittivity) will still have different labels (and therefore colors). In order for subpixel resolution (needed in Algorithm 1) to preserve this information, the interpolated value is taken as that of the closest input file entry (using closest neighbor interpolation).

Signed distance map. A signed distance map of a d-dimensional domain θ is a function $\Psi_{\partial\theta} : \theta \subset \mathbb{R}^d \mapsto \mathbb{R}$ such that

$$\begin{aligned} \Psi_{\partial\theta}(\mathbf{x}) &= -\min_{\mathbf{y}\in\partial\theta} ||\mathbf{x}-\mathbf{y}||_2 \quad \text{if } \mathbf{x} \in \theta, \\ \Psi_{\partial\theta}(\mathbf{x}) &= +\min_{\mathbf{y}\in\partial\theta} ||\mathbf{x}-\mathbf{y}||_2 \quad \text{if } \mathbf{x} \notin \theta. \end{aligned} \tag{19}$$

For the sake of convenience, the distance inside θ is taken as negative, so that $\nabla\Psi_\theta \cdot \mathbf{N}_\theta = 1$, where $\mathbf{N}_\theta$ is the outward normal to $\partial\theta$. The signed distance map can be constructed as $\Psi_{\partial\theta} = \varphi_{\mathbb{R}^d\setminus\theta} - \varphi_\theta$, where φ_D is the solution of the eikonal equation in the domain $D \subset \mathbb{R}^d$,

$$|\nabla\varphi_D| = 1, \qquad \varphi(\mathbf{x} \in \partial D) = 0, \tag{20}$$

extended in such a way that $\varphi_{\mathbb{R}^d\setminus D} = 0$ (notice that (20) can be an interior or exterior boundary value problem).

Fast marching. The eikonal equation (20) is hyperbolic and nonlinear, and in general admits only viscosity solutions. Numerical approximations to it are computed on a d-dimensional grid of step Δx by replacing partial derivatives with upwind finite differences. Sethian's fast marching method [20] allows for a numerical solution in just $O(m \log m)$ operations, where m is the number of grid nodes. The original fast marching had an error $O(\Delta x)$, but it subsequently improved to $O(\Delta x^2)$ [8]. Moreover, it has a reputation for being very robust, which is convenient in our case when the input has a low resolution.

Distance maps on the IC. Recall that the stochastic algorithm for capacitance extraction presented in section 2 requires signed distance maps of the IC domain, Ω, for two tasks:

1. defining the surfaces (in $\mathbb{R}^3$)/curves (in $\mathbb{R}^2$) ω_k needed to compute the charge on electrode k in the IC,
2. controlling the individual trajectories in Algorithm 1.

Let us define the electrode distance map $\psi_E(\mathbf{x})$ as the one obtained by ignoring the permittivity interfaces, while the global distance map ψ takes them into account:

$$\psi_E := \Psi_{\partial\Omega}, \qquad \psi := \min_{1\le i\le p} \Psi_{\partial\Omega_i}. \tag{21}$$

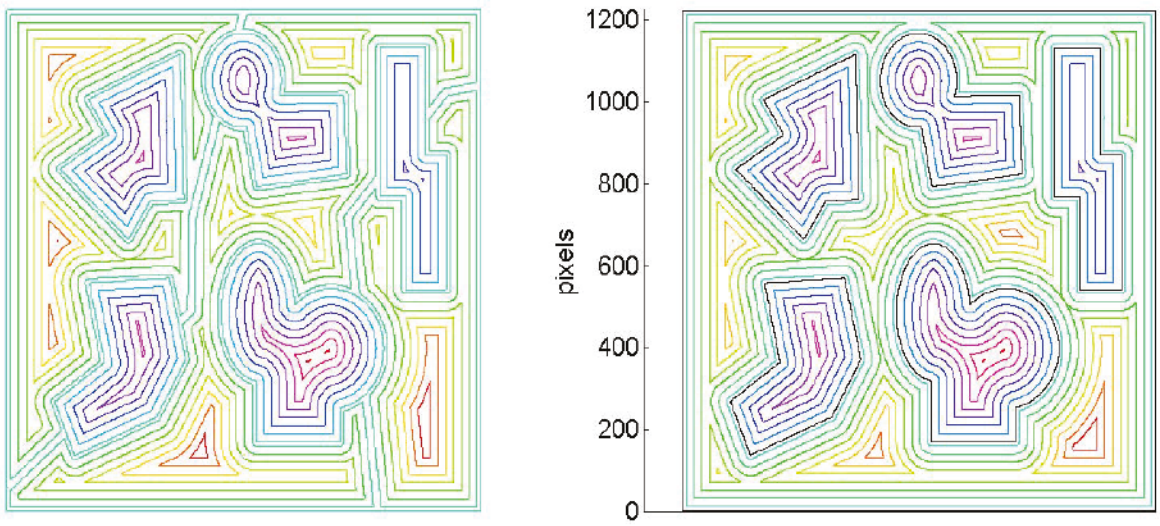

Figure 5. *Distance maps for Figure 4.* Right: ψ_E *(here, overlaid with* Γ_0 *and the electrodes' surfaces) ignores permittivity interfaces (if any) and is used to extract the Gaussian integration curves in Figure 6.* Left: ψ *(with no overlay) takes all interfaces into account and is used to control the particles' trajectories.*

Figure 5 shows both distance maps for the IC in Figure 4. Notice that both distance maps would be the same if there were no permittivity interfaces.

Before running Algorithm 1 there are two preprocessing steps. First, the distance maps ψ_E and ψ are computed with fast marching on a d-dimensional grid of spatial step $\triangle x$, which contains the IC domain Ω, and stored as a look-up table (a d-dimensional matrix) for later reference. The gradients $\nabla\psi_E$ and $\nabla\psi$ are obtained from the distance maps by numerical differentiation and also stored componentwise. The value of either distance map or its gradient can then be approximated anywhere inside Ω by interpolation from the corresponding look-up table(s). For instance, Algorithm 1 uses not only $\psi(\mathbf{X}_k)$ but also $\mathbf{X}_k^B$ and $\mathbf{N}(\mathbf{X}_k^B)$. By the properties of the eikonal equation, $\mathbf{N}(\mathbf{X}_k^B) = \nabla\psi(\mathbf{X}_k)$, and $\mathbf{X}_k^B = -\psi(\mathbf{X}_k)\nabla\psi(\mathbf{X}_k)$.

The second preprocessing step is the determination of the quadrature nodes for each of the $k = 1, \ldots, M$ Gaussian integration surfaces/curves ω_k. In order to do so, a parametric representation $\chi_k : \mathbb{R}^d \mapsto \omega_k \subset \mathbb{R}^{d-1}$ is required. χ_k is approximated with a closed d-dimensional spline. That spline (henceforth denoted as χ_k as well) can be produced from a cloud of ordered points, distributed (not necessarily evenly) around Γ_k at a distance h (the precise meaning of "ordered" will be discussed later). Let us define a footpoint $\mathbf{z}$ as a point which belongs to the spline, and let us consider the question of "extracting" footpoints on ω_k. Starting from a footpoint guess $\mathbf{z}^0$ (close to Γ_k and such that $\psi_E(\mathbf{z}^0) \approx -h$), the sequence $\{\mathbf{z}^m\}$ converges to $\mathbf{z}$ such that $\psi_E(\mathbf{z}) = -h$, where

$$\mathbf{z}^{m+1} = \mathbf{z}^m - [\nabla\psi_E(\mathbf{z}^m) + h]\frac{\nabla\psi_E(\mathbf{z}^m)}{|\nabla\psi_E(\mathbf{z}^m)|}. \tag{22}$$

Notice that $\omega_k \in \Omega$ and that we have used the distance sign criterion (19). A footpoint guess can be obtained as

$$\mathbf{z}^0 = \mathbf{p}_{\Gamma_k}^{EDGE} - h\nabla\psi_E(\mathbf{p}_{\Gamma_k}^{EDGE}), \tag{23}$$

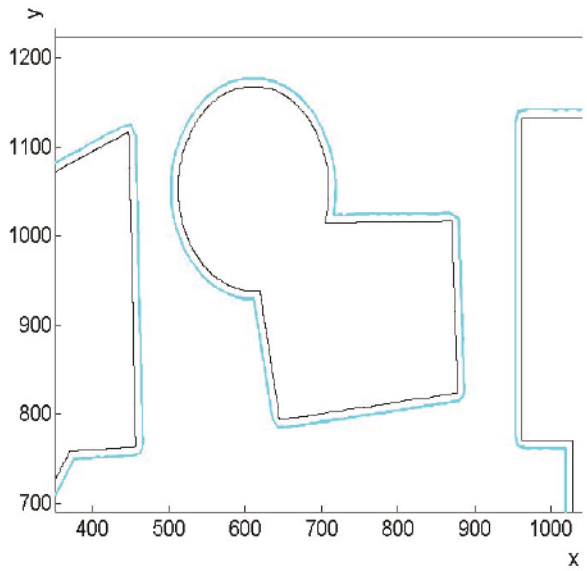

Figure 6. *Detail of the Gaussian integration curves (outer curves, parametrized as periodic splines) automatically constructed from ψ_E in Figure 5. Coordinates are in pixels.*

where $\mathbf{p}_{\Gamma_k}^{EDGE}$ is the location of a pixel center which belongs to the edge of the array $[A_k]$ defined by

$$[A_k](i,j) = \begin{cases} 0 & \text{if } [INPUT](i,j) = x \text{ such that } x \text{ is the label of electrode } k, \\ \kappa & \text{otherwise,} \end{cases} \tag{24}$$

where $\kappa > 0$ is arbitrary. In (24), (i,j) implies that $\Omega \subset \mathbb{R}^2$, the 3D case, is straightforward. The edge of $[A_k]$ can be extracted by any edge-detection routine such as Canny's algorithm [3] (the larger κ is, the more distinct the edge will be).

Full algorithm and sources of error. Algorithm 2 is the complete pseudocode. The new variable ρ is a superficial/linear density of quadrature nodes, N_0 is a batch size, and s is the largest admissible statistical error (with probability P_q) of the potential on a quadrature node. The numerical sources of error of Algorithm 2 are

1. the statistical error in the Monte Carlo approximation (6) (see section 2);
2. the bias of Algorithm 1, due to the discrete approximation of the various BCs involved (section 2);
3. the truncation error in the computation of the electric field (12);
4. the error in the quadrature (14);
5. the errors in the distance maps, due to fast marching (discussed above), and interpolation from the look-up tables; and
6. the error involved in the approximate representation of the IC domain Ω as a d-dimensional array $[INPUT]$. This level of discretization is characterized by the length of a pixel, and we will refer to it as the input resolution, r.

The above errors propagate to the entries of the capacitance matrix. Given the numerous sources, we have not carried out a rigorous analysis of the error of Algorithm 2. Instead, we investigate a numerical example in section 4. Specific to the problem presented in this paper is the way that the input resolution affects the accuracy of the calculation; let us discuss it in some more detail, focusing on the 2D case. When a particle gets close to a boundary pixel (one which represents an electrode's surface), it has to decide whether or not the electrode has been entered by interpolating from the table containing the distance map. Since the size of that table is the same as that of the image, the interpolated distance will better approximate the actual distance to the electrode when the input has high resolution. Conversely, for low-resolution inputs more particles will go astray and contribute a wrong voltage. In summary, the better the resolution is, the better the BCs resemble the BCs of the actual IC. In any case, we expect the error of Algorithm 2 to be made up of two components: a statistical error and a deterministic bias induced by all of the levels of spatial discretization. Note that the statistical error depends not only on N but also on the spatial discretization, via the variance.

Algorithm 2. Full stochastic algorithm for capacitance extraction.

Require: $[INPUT]$, T; $\{V_i\}_{i=1}^{M}, \{\epsilon_k\}_{k=1}^{p}$; $h, \delta > \gamma > 0$; $\rho, s, N_0 > 0$, $q = 1, 2, \ldots$
  compute the distance maps ψ_E and ψ with fast marching
  compute $\nabla\psi_E$ and $\nabla\psi$
  for $k = 1, \ldots, M$ **do**
    construct the spline χ_k around Γ_k
    extract $n_k = \rho L_k$ evenly distributed quadrature nodes $\{\mathbf{x}_i\}$ on χ_k
    for $i = 1, \ldots, n_k$ **do**
      let $N_i = 0$.
      while statistical error in (7) $> s$ **do**
        run Algorithm 1 with N_0 particles
        $N_i = N_i + N_0$
      end while
    end for
    compute $C_{k1}, \ldots, C_{kM}$
  end for

Programming details. Algorithm 2 was coded in MATLAB (Release 7.5) in vectorized style and run on a single computer. The fast marching toolbox [12] was used and achieved excellent performance. 2D interpolation was carried out by means of *interp2*. In the case of distance maps (and their gradients), bicubic interpolation was used, while for subpixel determination of discontinuous voltages and permittivities, first neighbor interpolation is needed, as explained in section 3.

The generation and manipulation of the splines χ_k which parametrize the ω_k Gaussian integration surfaces rely heavily on the MATLAB spline toolbox. In the remainder of this section, we comment on the 2D case, where the array $[INPUT]$ is a matrix, whose entries are the pixels of a bitmap image. First, $j = 1, \ldots, n_P$ edge pixels $\mathbf{p}_j := \mathbf{p}_{\Gamma_k}^{EDGE}|_j$ are picked on the edge of Γ_k according to (24); the command *edge* can be used for this task. They lead to the footpoint guesses $\{\mathbf{z}_j^0\}$ according to (23) and, by (22), to the footpoints $\{\mathbf{z}_j\}$. Now,

χ_k can be constructed on them if the footpoints are ordered sequentially (but not necessarily equispaced) along the level set $\psi_E = -h$ around Γ_k. Notice that the set $\{\mathbf{z}_j\}_{j=1}^{n_P}$ is ordered if the set $\{\mathbf{p}\}_{j=1}^{n_P}$ also is *and* the vectors $\mathbf{z}_j - \mathbf{p}_j$ do not intersect each other. This can be achieved by not taking many points $\mathbf{p}_j$ close to the inward corners of Γ_k. Therefore, n_P must be chosen (heuristically, for the time being) large enough for the spline to closely fit the level set $\psi_E = -h$, but in such a way that the inward corners are not oversampled. Finally, the set $\{\mathbf{p}\}_{j=1}^{n_P}$ can be ordered as follows. An arbitrary point in that set is taken and labeled as 1, the closest one to it as 2, the closest to 2 (other than 1) as 3, and so on. Figure 6 shows the spline which parametrizes ω_k around one of the electrodes (outer curve) in Figure 4. Once χ_k is constructed, it is straightforward to compute its length L_k and to extract quadrature nodes at constant intervals from it. In three dimensions, everything is similar up to the footpoint set $\{\mathbf{z}_j\}_{j=1}^{n_P}$, but there may be better alternatives to the parametrization of the surface ω_k than splines (see [7]).

Remark. The computational issues addressed in this section are inherent to irregular geometries and absent in Manhattan designs. If a non-Manhattan geometry were to be tackled by a field solver capable of extracting the capacitance matrix (i.e., not the FRW), a discretization of the IC should be produced first. For mesh-based methods, this usually involves processing the distance map via an algorithm such as marching cubes. Algorithm 2, on the other hand, can extract the capacitance values directly from the distance map.

4. Numerical experiment.

Three permittivities. We consider the 2D model of an IC in Figure 4 and two input bitmap files of sizes 634×634 and 1268×1268 pixels (respectively labeled LD and HD). The three permittivities are illustrated in the left diagram of Figure 4. The corresponding resolutions are thus $r_{LD} = 2/634$ (since Ω is a square of side 2) and $r_{HD} = r_{LD}/2$. All the look-up tables are matrices of the same size as the input file. Other parameters are $q = 2$, $N_0 = 100$, $n_k = 1000$, $h = 0.015$, $\gamma = 0.003$ (all in length units), and $\delta = 1.0$ pixels. In addition to the resolution, we also monitor the effect of the "noise" present in the line integral (15), which is brought about by the statistical error in (7) and controlled by parameter s in Algorithm 2. Specifically, we use the thresholds $s = 1$ and $s = 1/4$.

Remark. The reason for picking a synthetic pixelization such as this (instead of a SEM image such as that of Figure 3) is to be able to estimate the errors incurred by comparison with a reference solution. In this case it was an FEM solution obtained with the MATLAB *pdetool*. Note that this problem *cannot* be solved with the FRW.

Additional test with constant permittivity. In order to isolate the effect of the permittivity interfaces on the performance of the stochastic algorithm, we have also run the model in Figure 4 with a single permittivity $\epsilon = 1$.

Recall that $C_{ij} = \partial Q_i / \partial V_j$. The entries of the capacitance matrix were approximated as $C_{ij}^{FEM} \approx \Delta Q_i^{REF} / \Delta V_j$, with $\Delta V_j = \pm 1$. Finally, they were symmetrized to yield the reference matrix $C^{REF} := [C^{FEM} + (C^{FEM})^T]/2$. The ω_k curves, at a distance h from the electrodes, were also used in this case, so that the FEM solution is also slightly contaminated by the effect of the displacement charge when $p > 1$. Although they are not explicitly needed for the capacitance entries, we first consider the electrodes' charges for illustration purposes; see Table 1.

Table 1
Computed electrodes' charges.

Constant permittivity						
FEM	93.43	−1.13	267.96	−180.81	−179.31	0.14
LD, $s = 1$	87.50	1.20	248.94	−163.02	−171.39	3.23
LD, $s = 1/4$	86.46	0.42	251.64	−169.57	−170.72	−1.77
HD, $s = 1$	90.27	−0.77	256.30	−166.38	−172.23	7.18
HD, $s = 1/4$	90.10	−3.26	256.37	−172.80	−176.50	−6.10
Three permittivities						
FEM	113.95	−380.49	1532.40	−232.13	−1037.50	−3.77
LD, $s = 1$	108.64	−249.53	1409.07	−212.27	−976.83	79.09
LD, $s = 1/4$	106.62	−316.48	1403.87	−215.01	−998.64	−19.64
HD, $s = 1$	114.25	−334.38	1458.32	−219.75	−988.49	29.96
HD, $s = 1/4$	110.27	−383.18	1473.82	−222.48	−1023.48	−45.05
	Q_1	Q_2	Q_3	Q_4	Q_5	$\sum_{j=1}^{5} Q_j$

Table 1 indicates that the electrodes' charges calculated with the stochastic algorithm converge toward the correct values as the input file resolution increases and the statistical error drops. However, we are seeking the capacitances rather than the charges. The reference capacitance matrices are (C_1^{REF} and C_3^{REF} stand for the constant-permittivity and three-permittivities cases, respectively)

$$C_1^{REF} = \begin{pmatrix} 6.65 & -0.004 & -0.003 & -2.46 & -4.21 \\ -0.004 & 7.48 & -2.41 & -0.60 & -4.45 \\ -0.003 & -2.41 & 5.88 & -3.45 & -0.06 \\ -2.46 & -0.60 & -3.45 & 7.99 & -1.48 \\ -4.21 & -4.45 & -0.06 & -1.48 & 10.24 \end{pmatrix}, \tag{25}$$

$$C_3^{REF} = \begin{pmatrix} 8.87 & -0.008 & -0.004 & -4.54 & -4.33 \\ -0.008 & 80.71 & -33.13 & -0.83 & -46.93 \\ -0.004 & -33.13 & 37.27 & -4.01 & -0.08 \\ -4.54 & -0.83 & -4.01 & 10.67 & -1.30 \\ -4.33 & -46.98 & -0.08 & -1.30 & 52.79 \end{pmatrix}. \tag{26}$$

As an illustration, we show below two realizations of the raw capacitance matrix (i.e., before symmetrization) for the three-permittivities case, calculated via (18):

$$C_{s=1}^{LD} = \begin{pmatrix} 8.61 & -0.012 & -0.010 & -4.55 & -4.04 \\ 0.000 & 75.28 & -29.14 & -0.70 & -45.44 \\ -0.005 & -30.39 & 34.24 & -3.75 & -0.10 \\ -4.36 & -0.74 & -3.65 & 10.00 & -1.25 \\ -4.17 & -44.31 & -0.06 & -1.25 & 49.78 \end{pmatrix}, \tag{27}$$

$$C^{HD}_{s=1/4} = \begin{pmatrix} 8.54 & -0.009 & -0.002 & -4.31 & -4.22 \\ -0.011 & 77.69 & -32.15 & -0.81 & -44.73 \\ -0.004 & -31.56 & 35.77 & -4.14 & -0.077 \\ -4.31 & -0.80 & -3.85 & 10.24 & -1.27 \\ -4.22 & -46.50 & -0.08 & -1.30 & 52.09 \end{pmatrix}. \tag{28}$$

The relative absolute errors of $[C^{LD}_{s=1} + (C^{LD}_{s=1})^T]/2$ in (27) and of $[C^{HD}_{s=1/4} + (C^{HD}_{s=1/4})^T]/2$ in (28) with respect to (26) are, respectively,

$$\varepsilon^{LD}_{s=1} = \begin{pmatrix} .030 & .268 & .702 & .018 & .052 \\ .268 & .067 & .102 & .126 & .044 \\ .702 & .102 & .081 & .079 & .064 \\ .018 & .126 & .079 & .063 & .034 \\ .052 & .044 & .064 & .034 & .057 \end{pmatrix}, \qquad \varepsilon^{HD}_{s=1/4} = \begin{pmatrix} .038 & .238 & .364 & .049 & .026 \\ .238 & .037 & .039 & .027 & .028 \\ .364 & .039 & .040 & .005 & .051 \\ .049 & .027 & .005 & .041 & .012 \\ .026 & .028 & .051 & .012 & .013 \end{pmatrix}.$$

Since the entries C_{12}, C_{13} (and their symmetric counterparts) are much smaller than the rest, the relative errors are distinctly larger on those four entries. In order to quantify the effect of r and s on the accuracy, we have performed simple statistics on the capacitance matrix entries. Let us define $\langle \varepsilon^* \rangle$ as the mean of the entrywise relative errors, and $var(\varepsilon^*)$ as the variance, in both cases after removing entries $C_{12}, C_{13}, C_{21}, C_{31}$ from the sample. (The rationale for doing this is that they involve too few trajectories for their sample estimates to be reliable.) See Table 2. By comparing both sets of results, we can conclude that nonaligned permittivity interfaces do not significantly alter the bias or the variance.

Table 2
Errors on one realization of the stochastic capacitance matrix.

	Constant permittivity		Three permittivities	
LD, s=1	.075	.060	.063	.030
LD, s=1/4	.062	.018	.074	.027
HD, s=1	.048	.050	.055	.049
HD, s=1/4	.027	.013	.031	.014
	$\langle \varepsilon^* \rangle$	$\sqrt{\langle var\ \varepsilon^* \rangle}$	$\langle \varepsilon^* \rangle$	$\sqrt{\langle var\ \varepsilon^* \rangle}$

5. Conclusions. We have presented a new stochastic algorithm for capacitance extraction. It is specifically designed to exploit microscopic image data ICs in the submicron scale with a minimal amount of human supervision. At such scales, which are increasingly relevant in practice, departure of the IC from the ideal Manhattan layout is significant. Contrary to the FRW, the presented algorithm can handle irregular geometries. Thanks to a specific selection of Gaussian surfaces, the correct component of the electric field is approximated directly, as in the FRW and in the method in [2]. We have listed the various sources of error affecting the algorithm and investigated numerically the aggregate error of the complete algorithm, including preprocessing. Preliminary results show that the capacitance matrix can indeed be

extracted from an image of the IC with competitive accuracy. We are currently working on 3D and nonsynthetic examples.

REFERENCES

[1] A. Aggarwal and S. Suri, *Fast algorithms for computing the largest empty rectangle*, in Proceedings of the Third Annual Symposium on Computational Geometry Waterloo, ON, ACM, New York, 1987, pp. 278–290.

[2] A. Brambilla and P. Maffezzoni, *A statistical algorithm for 3-D capacitance extraction*, IEEE Microw. Guided Wave Lett., 10 (2000), pp. 304–306.

[3] J. Canny, *A computational approach to edge detection*, IEEE Trans. Pattern Anal. Mach. Intell., 8 (1986), pp. 679–698.

[4] J. Denker, *Capacitance*, http://www.av8n.com/physics/capacitance.htm (2008).

[5] J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, and A. White, eds., *Sourcebook of Parallel Computing*, Morgan Kaufmann, San Francisco, 2003.

[6] T. El-Moselhy, I. M. Elfadel, and L. Daniel, *A Markov chain based hierarchical algorithm for fabric-aware capacitance extraction*, IEEE Trans. Adv. Packag., 33 (2010), pp. 818–827.

[7] E. Hartmann, *Numerical parameterization of curves and surfaces*, Comput. Aided Geom. Design, 17 (2000), pp. 251–266.

[8] M. S. Hassouna and A. A. Farag, *Multistencils fast marching methods: A highly accurate solution to the Eikonal equation on Cartesian domains*, IEEE Trans. Pattern Anal. Mach. Intell., 29 (2007), pp. 1563–1574.

[9] L. Holzer and M. Cantoni, *Review of FIB-tomography*, in Nanofabrication Using Focused Ion and Electron Beams, I. Utke, S. Moshkalev, and P. Russell, eds., Oxford University Press, New York, 2012, pp. 410–435.

[10] R. B. Iverson and Y. L. Le Coz, *A floating random-walk algorithm for extracting electrical capacitance*, Math. Comput. Simulation, 55 (2001), pp. 59–66.

[11] W. H. Kao, C.-Y. Lo, M. Basel, and R. Singh, *Parasitic extraction: Current state of the art and future trends*, Proc. IEEE, 89 (2001), pp. 729–739.

[12] D.-J. Kroon, *Accurate Fast Marching* (MATLAB toolbox 2009–2011), http://www.mathworks.com/matlabcentral/fileexchange/24531-accurate-fast-marching.

[13] Y. L. Le Coz, H. J. Greub, and R. B. Iverson, *Performance of random walk capacitance extractors for IC interconnects: A numerical study*, Solid-State Electronics, 42 (1998), pp. 581–588.

[14] Y. L. Le Coz and R. B. Iverson, *A stochastic algorithm for high speed capacitance extraction in integrated circuits*, Solid-State Electronics, 35 (1992), pp. 1005–1012.

[15] J. H. Lee and S. I. Yoo, *An effective image segmentation technique for the SEM image*, in Proceedings of the IEEE International Conference on Industrial Technology (ICIT 2008), 2008, pp. 1-5.

[16] M. Mascagni and N. A. Simonov, *Monte Carlo methods for calculating some physical properties of large molecules*, SIAM J. Sci. Comput., 26 (2004), pp. 339–357.

[17] E. Ramsay, D. T. Reid, and K. Wilsher, *Three-dimensional imaging of a silicon flip chip using the two-photon optical-beam induced current effect*, Appl. Phys. Lett., 81 (2002), pp. 7–9.

[18] G. M. Royer, *A Monte Carlo procedure for potential theory problems*, IEEE Trans. Microw. Theory Tech., 19 (1971), pp. 813–818.

[19] K. K. Sabelfeld, *Monte Carlo Methods in Boundary Value Problems*, Springer Ser. Comput. Phys., Springer-Verlag, Berlin, 1991.

[20] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, UK, 1999.

[21] W. Yu and Z. Wang, *Capacitance extraction*, in Encyclopedia of RF and Microwave Engineering, K. Chang, ed., John Wiley & Sons, Hoboken, NJ, 2005, pp. 565–576.

[22] W. Yu, H. Zhuang, C. Zhang, G. Hu, and Z. Liu, *RWCap: A floating random walk solver for 3-D capacitance extraction of very-large-scale integration interconnects*, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 32 (2013), pp. 353–366.

[23] Y. Zhou, Z. Li, Y. Tian, W. Shi, and F. Liu, *A new methodology for interconnect parasitics extraction considering photo-lithography effects*, in Proceedings of the Asia South Pacific Design Automation Conference (ASP-DAC), IEEE, Piscataway, NJ, 2007, pp. 450–455.

[24] H. Zhuang, W. Yu, G. Hu, Z. Liu, and Z. Ye, *Fast floating random walk algorithm for multi-dielectric capacitance extraction with numerical characterization of Green's functions*, in Proceedings of the 17th Asia and South Pacific Design Automation Conference (ASP-DAC), Sydney, Australia, IEEE, Piscataway, NJ, 2012, pp. 377–382.