Journal of Computational Physics **152,** 281–304 (1999)
Article ID jcph.1999.6249, available online at http://www.idealibrary.com on IDEAL®

A Numerical Study of Electro-migration Voiding by Evolving Level Set Functions on a Fixed Cartesian Grid

Zhilin Li,*,[1] Hongkai Zhao,†,[2] and Huajian Gao‡

*Center For Research in Scientific Computation and Department of Mathematics, North Carolina State University, Raleigh, North Carolina 27695-8205; †Department of Mathematics, Stanford University, Stanford, California 94305; and ‡Department of Mechanical Engineering, Stanford University, Stanford, California 94305-4040

E-mail: *zhilin@math.ncsu.edu, †zhao@cartan.Stanford.EDU, and ‡gao@amsun2.Stanford.EDU

Received October 23, 1998; revised February 24, 1999

A numerical method for studying migration of voids driven by surface diffusion and electric current in a metal conducting line is developed. The mathematical model involves moving boundaries governed by a fourth order nonlinear partial differential equation which contains a nonlocal term corresponding to the electrical field and a nonlinear term corresponding to the curvature. Numerical challenges include efficient computation of the electrical field with sufficient accuracy to afford fourth order differentiation along the void boundary and to capture singularities arising in topological changes. We use the modified immersed interface method with a fixed Cartesian grid to solve for the electrical field, and the fast local level set method to update the position of moving voids. Numerical examples are performed to demonstrate the physical mechanisms by which voids interact under electromigration.

1. INTRODUCTION

In many physical problems, mass transport along interfaces such as surface diffusion and grain boundary diffusion becomes increasingly important as the characteristic length scale is reduced. The diffusional mass transport is governed by a relevant chemical potential along the interface. Converging or diverging atomic fluxes cause motion or relocation of boundaries. The dynamics of these processes is of great interest to material scientists and

[1] The author is partially supported by the NSF under Grant DMS-96-26703 and North Carolina State FR&PD Fund.

[2] The author is partially supported by the NSF under Grant DMS-97-06566.

0021-9991/99 $30.00

biologists. There is a large literature on this topic. We refer the readers to two recent survey articles by Mullins [14] and Cahn and Taylor [5] and the references therein. The problem considered in this paper involves the evolution of voids under electro-migration in a conducting metal line where the driving forces for diffusion are the gradient of the curvature and the electric potential along the void boundary. The normal velocity of the void surface is given by the partial differential equation (PDE)

$$U_n = \Delta_s(C_1\phi + C_2\kappa), \tag{1.1}$$

where Δ_s is the surface Laplacian, ϕ is the potential function associated with an applied electric field, and κ is the mean curvature along the boundary; for a circle, the curvature is a positive constant. The coefficients C_1, C_2 are related to the physical constants

$$C_1 = \frac{eD_s\Omega^{1/3}Z^*}{k_BT_k}, \qquad C_2 = \frac{D_s\Omega^{4/3}\gamma_s}{k_BT_k}, \tag{1.2}$$

where e is the charge of an electron, Ω is the atomic volume, Z^* is a phenomenological constant related to the effective valence of an atom, k_B is Boltzmann's constant, T_k is the temperature, γ_s is the surface energy. The constant D_s is defined as

$$D_s = \frac{D_s^*\delta_s}{k_BT_k}e^{-Q_s/k_BT_k}, \tag{1.3}$$

where δ_s is the thickness of the diffusion layer, $D_s^*e^{-Q_s/k_BT_k}$ is the surface diffusion coefficient, and Q_s is the activation energy for surface diffusion. The electric potential ϕ satisfies the Laplace equation $\Delta\phi = 0$, with no-flux boundary condition $\frac{\partial\phi}{\partial n} = 0$ on the void boundary as well as other appropriate boundary conditions on the computational boundary.

For a void bounded by a closed surface, it can be shown from the divergence theorem that the void conserves its volume (or area) during surface diffusion. In Eq. (1.1), the first term, $\Delta_s\phi$, is a nonlocal driving force which tends to drift the void along with the electric current. The second term, $\Delta_s\kappa$, is a fourth order nonlinear term which only depends on the local geometry. The boundary evolution governed by the surface Laplacian of the mean curvature, as studied by numerous authors [3–5, 14], can be regarded as a gradient flow with H^{-1} inner product for the surface area given by the equation

$$\frac{dA}{dt} = \int_S U_n\kappa\, ds = \int_S \kappa\Delta_s\kappa\, ds = -\int_S |\nabla_s\kappa|^2\, ds. \tag{1.4}$$

Such a process tends to minimize the surface area while conserving the volume. Anisotropy can be included in both the free energy and the above inner product in a variational form [5]. Another type of gradient flow which minimizes the surface area while conserving the volume with L^2 inner product is $\kappa - \bar{\kappa}$, where $\bar{\kappa}$ is the average of the mean curvature along the interface. These two types of gradient flows correspond to two limiting cases of combined surface diffusion and growth process. The connection between sharp and diffuse interface motion laws via a gradient flow can be found in [19]. In general, surface diffusion problems admit few analytical results. Due to the intrinsic nonlinearity and lack of a maximum principle, smooth solutions only exist locally in time while topological singularities occur in finite time. Surfaces (curves) can merge or pinch-off in both two and three dimensions. A linear stability analysis [15, 18] indicates Rayleigh instability at

long wavelength perturbations. It has been shown using a perturbation analysis [3] that the only stable equilibria are surfaces with constant mean curvature which locally minimize the surface area. The dynamics and stability of self-similar pinch-off were also studied in [3]. For a closed plane curve, suppose $(x(\alpha, t), y(\alpha, t))$ is some parametrization of the curve. Define the metric $g(\alpha, t) = \sqrt{x_\alpha^2 + y_\alpha^2}$ and angle $\theta = \tan^{-1}(y_\alpha / x_\alpha)$. Following the derivation given in [16], we have

$$g_t(\alpha, t) = \theta_\alpha U_n, \qquad \theta_t(\alpha, t) = -\frac{1}{g}\frac{\partial U_n}{\partial \alpha},$$

where U_n is the normal velocity of the curve. Using the fact $\kappa = \theta_s = \theta_\alpha / g$ and $\frac{\partial}{\partial s} = \frac{1}{g}\frac{\partial}{\partial \alpha}$, where s is the arc length, we get the time evolution equation below for the curvature κ with respect to the arc length s,

$$\kappa_t = -\frac{\partial^2 U_n}{\partial s^2} - \kappa^2 U_n.$$

If $U_n = \kappa_{ss}$, then

$$\kappa_t = -\kappa^2 \kappa_{ss} - \kappa_{ssss}. \tag{1.5}$$

We can see from the evolution equation that the first term, $-\kappa^2\kappa_{ss}$, is a second order and nonlinear term corresponding to a backward heat equation which causes instability, while the second term, $-\kappa_{ssss}$, is a linear fourth order stabilizing term. Through simple linearized analysis we can see the long wave instabilities. If we multiply the equation by κ and integrate it with respect to s, after integration by parts we have the energy equation

$$\frac{d}{dt}\int_0^L \frac{1}{2}\kappa^2\, ds = 3\int_0^L \kappa^2\kappa_s^2\, ds - \int_0^L \kappa_{ss}^2\, ds,$$

where L is the total length of the curve. A similar type of equation, discussed in [7], shows that a closed curve will converge to a circle only if its initial shape does not deviate too much from a circle; otherwise singularities may occur in finite time. This phenomenon is also discussed through a modified Kuramoto–Sivashinsky equation for nearly planar interface motion for phase transitions [2].

In this paper we shall address some difficulties in the numerical implementation of moving boundary problems arising in electro-migration voiding. These include (a) constructing an efficient and accurate Poisson solver for the equations defined on arbitrary domains, (b) tracking topological changes (breaking, merging) along a moving interface, (c) evaluating the surface Laplacian operator along the interface. We wish to explore the possibility of simulating moving interfaces on a fixed Cartesian grid without having to re-mesh as the interfaces migrate. For this purpose, we introduce the modified immersed-interface-method (IIM) developed in [10, 11] to solve the Poisson equation for the electric potential. We show that the system can be preconditioned so that the convergence is almost independent of the mesh size. We use the local level set method to update the interface according to Eq. (1.1). We will derive a general formula for the surface Laplacian operator in Cartesian coordinates along the interface represented implicitly by a level set function. As discussed in [5], the level set function for a geometric PDE cannot be arbitrarily chosen. Since each contour of the level set function moves according to its own curvature variation, and because of the

lack of a maximum principle, different contours may cross each other. We adopt a modified level set function method [22] which allows us to get around this problem by (i) using the local level set method so that only those level sets that are very close to the interface are involved; (ii) sticking to the signed distance function as our unique choice of level set function; and (iii) extending the quantities at the interface, if necessary, to other level sets so that they are normal to the interface. It has been shown [22] that this extension of the normal velocity maintains the normal distance between different contours. After these modifications, the level set method can be used efficiently to capture moving interfaces for the eletro-migration problems discussed here. An alternative approach was proposed in [9, 20] using the finite-element formulation with adaptive re-meshing. The numerical algorithm that we propose in this paper has the advantage of simulating a moving interface on a fixed Cartesian grid, which demands a relatively small amount of computational resources and may be desirable as a practical analysis tool for many applications.

2. OUTLINE OF THE NUMERICAL ALGORITHM

We consider a moving boundary with the normal velocity given by surface diffusion under a linear combination of electrical potential and surface tension,

$$U_n = \Delta_s = \frac{\partial^2}{\partial s^2}(C_1\phi + C_2\kappa), \tag{2.6}$$

where ϕ is the electrical potential which is a solution of the Laplace equation on the domain outside of voids (see Fig. 1), κ is the curvature along the boundary of voids which are immersed in a fixed Cartesian grid over the computational domain.

Below is an outline of our numerical algorithm along with reference to the specific sections where more details are given.

Outline of the Method

- From the level set function at a time level t^n, find necessary interface information such as the normal and tangential directions, the projections of irregular grid points, the curvatures, etc.
- Use the IIM to solve the electric potential function; see Section 3.
- Find the component of the normal velocity of the surface due to the electric potential; see Section 4.1.
- Find the component of the normal velocity of the surface due to the curvature variation; see Section 4.2.
- Find the normal velocity of the level set function.
- Update the level set function.
- Reinitialize the level set function.
- Go to the next time level.

3. A FAST ALGORITHM FOR COMPUTING THE ELECTRICAL POTENTIAL ON IRREGULAR DOMAINS

We need to solve the Poisson equation

$$\Delta\phi = f(x, y), \qquad (x, y) \in \Omega_1, \tag{3.7}$$

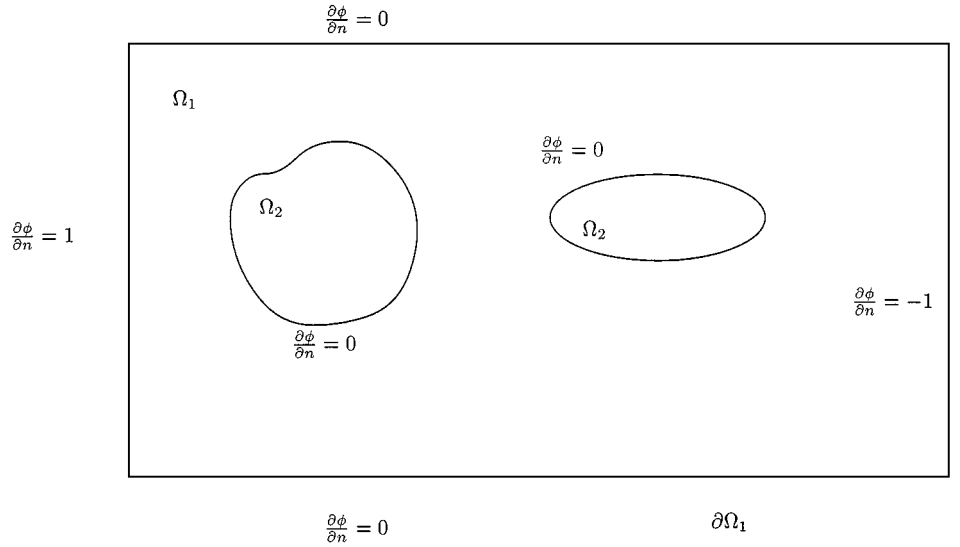

FIG. 1. A sketch of a computational domain. The regions of Ω_2 are voids. We want to track the motion of their boundaries.

$$\left.\frac{\partial \phi}{\partial n}\right|_{\partial\Omega_2} = v(s), \tag{3.8}$$

where s is the arc length of the interface $\partial\Omega_2$. The solution is defined on a multi-connected domain with given boundary conditions on $\partial\Omega_1$ (see Fig. 1). There is also a compatibility condition if the boundary conditions are of Neumann type. In our electro-migration problem, $v(s) \equiv 0$.

In recent years, the level set method, first proposed in [16], has become a powerful tool for computing moving boundary/interface problems, especially for problems that involve topological changes and/or in three dimensions. We intend to present a simple, second order discretization method for solving (3.7) with the interface described by a level set function. There are a variety of methods based on finite difference for this kind of problems, for example, the capacitance matrix method and fast methods based on integral equations. However, many of these methods require explicit information about the boundary of exclusions, such as spline interpolations or Fourier expansions for closed regions. It is not clear how to implement these methods if the boundary is described by a two dimensional level set function because the information is given only at grid points. Our discretization method is based on the immersed interface method [12, 21], and is designed for treating Neumann boundary conditions under the level set formulation with an efficient preconditioning technique.

To take advantage of a fast Poisson solver, we extend the equation to the entire rectangular region

$$\Delta\phi = f(x, y), \qquad (x, y) \in \Omega_1 \cup \Omega_2, \tag{3.9}$$

$$\left.\frac{\partial \phi^+}{\partial n}\right|_{\partial\Omega_2} = v(s), \tag{3.10}$$

where $f(x, y)$ is defined as zero in Ω_2. We use a $+$ sign to indicate the limit of a function when approaching from the exterior of Ω_2. The solution ϕ inside Ω_2 depends on how the

jump conditions along the interface are specified. If a single layer is introduced, there is a jump in the normal derivative. If a double layer is introduced, the function is discontinuous across the interface. Usually, we cannot require the function and its normal derivative to be continuous across the interface at the same time. Following the idea introduced in [12, 21], we either introduce an unknown jump in the solution and require the normal derivative to be continuous or vice versa. However, to force the solution to be continuous may lead to rapid changes in the function, and an ill-conditioned system for the unknown jump in the derivative.

Now consider the functional $\phi(g)$,

$$\begin{aligned} \Delta\phi &= f(x, y), \qquad (x, y) \in \Omega_1 \cup \Omega_2, \\ [\phi]|_{\partial\Omega_2}(s) &= g(s), \qquad [\phi_n]|_{\partial\Omega_2} = 0, \end{aligned} \tag{3.11}$$

where the jump across $\partial\Omega_2$ is the difference between the limiting value from the exterior of the voids and that from the interior. Since the jump conditions always refer to the interface $\partial\Omega_2$, we will omit the explicit reference to $\partial\Omega_2$ from now on. The solution $\phi(g)$ of the system above is a functional of the jump $g(s)$ with $[u] = g(s)$. We are interested in the particular $g(s)$ such that $\partial\phi(g)/\partial n = v(s)$. This corresponds to a double layer in the potential theory.

3.1. *Discretization of the Interface*

In order to construct a fast and convenient numerical algorithm for moving interface problems, we adopt a uniform Cartesian grid with a fast Poisson solver and use the level set method to update the interface at each time step.

The grid points are divided into two categories: *regular* grid points are those located away from the interface and *irregular* ones are those located where the interface cuts through the standard five point stencil. Our attention is focused on the irregular grid points.

A two dimensional level set function $\varphi(x, y)$, where $\varphi(x, y) = 0$ describes the location of the interface, is introduced. In the level set formulation, the interface information is only given at grid points. To solve for the unknown jump function $g(s)$ so that the the solution satisfies the Neumann boundary condition, we need to discretize $g(s)$ as well. In other words, we need to find values of $g(s)$ at discrete points on the interface. Too many points often lead to a large and ill-conditioned system which also means more storage. Too few points often lead to loss of accuracy. Our strategy is to find the projections of irregular grid points on the interface. Let $\mathbf{X} = (x_i, y_j)$ be an irregular grid point. Its projections $\mathbf{X}^* = (x^*, y^*)$ can be found using the following procedure (see also Fig. 2):

1. Find the unit steepest ascent direction $\mathbf{p}$ at $\mathbf{X}$:

$$\mathbf{p} = \frac{\nabla\varphi}{\|\nabla\varphi\|}. \tag{3.12}$$

2. Locate the projection of $\mathbf{X}$ on the interface along the direction p,

$$\mathbf{X}^* = \mathbf{X} + \alpha\mathbf{p}, \tag{3.13}$$

where α is determined from the quadratic equation

$$\varphi(\mathbf{X}) + \|\nabla\varphi\|\alpha + \tfrac{1}{2}(\mathbf{p}^T \operatorname{He}(\varphi)\mathbf{p})\alpha^2 = 0 \tag{3.14}$$

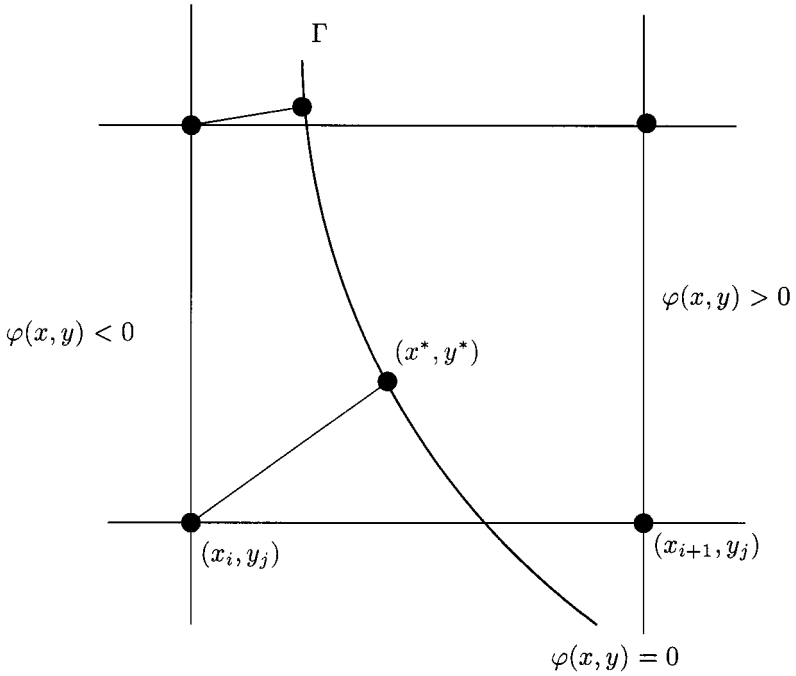

FIG. 2. Finding the control point $\mathbf{X}^*$ from an irregular grid point (x_i, y_j), $\varphi(x_i, y_j) \le 0$. It can be chosen as the projection of the grid point on the interface.

and $\mathrm{He}(\varphi)$ is the Hessian matrix of φ,

$$\mathrm{He}(\varphi) = \begin{bmatrix} \varphi_{xx} & \varphi_{xy} \\ \varphi_{yx} & \varphi_{yy} \end{bmatrix}, \tag{3.15}$$

evaluated at $\mathbf{X}$.

We only define $g(s)$ on those projections from a particular side, for example, $\varphi(x, y) \ge 0$. We will call them $\mathbf{Set_I}$. The projections from the other side of the interface are called $\mathbf{Set_{II}}$. The values of $g(s)$ on those projections in $\mathbf{Set_{II}}$ can be defined via interpolation using the values defined at those in $\mathbf{Set_I}$. The interpolation scheme will be explained later. Note that we do not need to *order* the projections where the unknown jumps are defined, a very important feature of the level set formulation compared with the Lagrangian formulation.

3.2. *Discretization of the Poisson Equation with Jumps*

Given jump conditions across the interface $\partial\Omega_2$,

$$[\phi] = g(s), \qquad [\phi_n] = 0, \qquad [f] = f^+(s) \tag{3.16}$$

at all projections, we use the immersed interface method to solve the Poisson equation. The essence of this method [10–12] consists of a finite difference scheme with the standard discrete Laplacian plus a correction term at the right hand side associated with irregular grid points, leading to a discrete system which can be solved by a fast Poisson solver. The details can be found in [10–12]. The interface information such as the tangential and

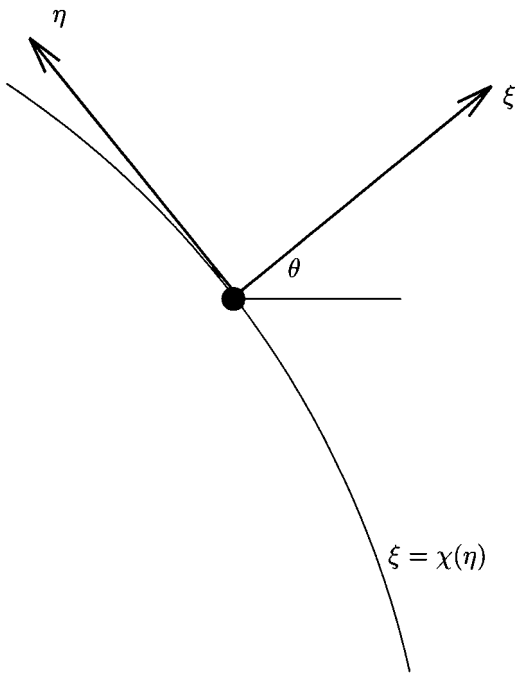

FIG. 3. A diagram of the local coordinates. The interface can be expressed as $\xi(\eta)$ with $\xi(0) = 0$ and $\xi'(0) = 0$.

normal derivatives and the curvature at projections is obtained from the values of the level set function at grid points plus a bilinear interpolation; see [12] for details. A key part is evaluating the first and second derivatives of the jump $[\phi]$ along the interface.

Evaluation of the derivatives of the jump function $[\phi]$. The idea is to use the weighted least squares interpolation [12] to approximate the derivatives of the jump along the interface at a given projection $\mathbf{X}^* = (x^*, y^*)$. We use the local coordinates at (x^*, y^*) in the tangential and normal directions (see Fig. 3),

$$\begin{aligned} \xi &= (x - x^*)\cos\theta + (y - y^*)\sin\theta, \\ \eta &= -(x - x^*)\sin\theta + (y - y^*)\cos\theta, \end{aligned} \tag{3.17}$$

where θ is the angle between the x-axis and the normal direction, pointing to the region of Ω_1.

In order to use the weighted least squares interpolation, we take a small circle $\odot$ centered at (x^*, y^*) with a radius α; usually α is between $1.5h$ and $4h$ so that at least three projections from $\mathbf{Set_I}$ are enclosed. If we can find the signed arc length starting from (x^*, y^*), we can use an appropriate interpolation scheme to approximate the derivatives of the jump function with respect to the arc length. Near (x^*, y^*), the interface has the form

$$\xi(\eta) = C\eta^2 + D\eta^3 + O(\eta^4). \tag{3.18}$$

Let $\mathbf{X}_1^* = (x_1^*, y_1^*)$ be a projection in $\mathbf{Set_I}$ but different from (x^*, y^*). We can determine the constants C and D using the interface information. Denote (n_ξ, n_η) as the unit normal

direction of the interface; we have

$$\begin{aligned}\frac{2C\eta_1 + 3D\eta_1^2}{\sqrt{1 + \left(2C\eta_1 + 3D\eta_1^2\right)^2}} &= -n_\eta,\\ \frac{1}{\sqrt{1 + \left(2C\eta_1 + 3D\eta_1^2\right)^2}} &= n_\xi,\end{aligned} \tag{3.19}$$

where (ξ_1, η_1) are the coordinates under the transformation (3.17). We arrive at the following linear system of equations for C and D:

$$\begin{aligned}C\eta_1^2 + D\eta_1^3 &= \xi_1,\\ 2C\eta_1 + 3D\eta_1^2 &= -\frac{n_\eta}{n_\xi}.\end{aligned} \tag{3.20}$$

In other words, the curve is approximated by a Hermite spline interpolation between $\mathbf{X}^*$ and $\mathbf{X}_1^*$. Once we have solved for C and D, we have an analytic expression for approximating the interface. The arc length between $\mathbf{X}^*$ and $\mathbf{X}_1^*$ is determined from

$$|s_1| = \int_0^{\eta_1} \sqrt{1 + (2C\eta + 3D\eta^2)^2}\, d\eta. \tag{3.21}$$

This definite integral can be approximated by the Simpson rule or a Gaussian quadrature formula using the approximate analytic expression of the interface. In this way, the arc length is evaluated to third order accuracy, which is necessary for the second order scheme for the Poisson equation. Note that the distance between two points on the interface is only a second order approximation to the arc length and is not accurate enough for our requirement. Finally, we need to determine the sign of the arc length according to the relative position between $\mathbf{X}^*$ and $\mathbf{X}_1^*$,

$$s_1 = \begin{cases} |s_1| & \text{if } (\mathbf{X}_1^* - \mathbf{X}^*) \cdot \mathbf{t}^* > 0\\ -|s_1| & \text{otherwise,}\end{cases} \tag{3.22}$$

where $\mathbf{t}^*$ is the tangential vector at $\mathbf{X}^*$.

We use uppercase letters to express discrete quantities. For example, $G(s)$ is the discrete form of $g(s)$. Once we have the signed arc length between $\mathbf{X}^*$ and other projections from $\mathbf{Set_I}$, it is easy to interpolate $G(s)$ from $\mathbf{Set_I}$ to obtain $G'(s^*)$ and $G''(s^*)$ at any projection on the interface, either in $\mathbf{Set_I}$ or in $\mathbf{Set_{II}}$, using the following weighted least squares interpolations,

$$G(s^*) = \sum_{\mathbf{X}_k^* \in \mathbf{Set_I} \cap \odot} \alpha_k G_k(\mathbf{X}_k^*), \tag{3.23}$$

$$G'(s^*) = \sum_{\mathbf{X}_k^* \in \mathbf{Set_I} \cap \odot} \beta_k G_k(\mathbf{X}_k^*), \tag{3.24}$$

$$G''(s^*) = \sum_{\mathbf{X}_k^* \in \mathbf{Set_I} \cap \odot} \gamma_k G_k(\mathbf{X}_k^*), \tag{3.25}$$

where α_k, β_k, and γ_k are the coefficients of the interpolation which is determined from the system

$$\begin{bmatrix} \sum_k d_k \\ \sum_k d_k s_k \\ \sum_k d_k \frac{s_k^2}{2} \end{bmatrix} [\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{3.26}$$

where d_k is the distance between $\mathbf{X}_k^*$ and $\mathbf{X}^*$ and $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ are the vectors whose components are the coefficients α_k, β_k, and γ_k. These are under-determined systems with the same coefficient matrix of full row-rank. We use the least squares solution from the pseudo-inverse. The solution minimizes the Euclidean norm among all possible solutions. With the weighted least squares interpolation, the errors in the interpolation vary smoothly and those points closer to the center are given more weight than the other points.

With the scheme discussed above for evaluating the jump $[\phi]$ and its first and second derivatives, the rest of the implementation of the IIM method is straightforward. The process of the discretization can be written in a matrix–vector form:

$$A\Phi + BG = F. \tag{3.27}$$

Here A is the discrete Laplacian using the standard five point stencil, B is the deviation of the difference scheme due to the jump in the solution, and F is the source term plus the correction terms at irregular grid points.

3.3. *Discretization of the Residual*

We want to find $g(s)$ such that $\phi(g)$ satisfies the Neumann boundary condition

$$\phi_n^+ = v(s). \tag{3.28}$$

For an arbitrary $[\phi]$, the solution $\phi(x, y)$ usually does not satisfy the equation above. The discrete difference of the two sides is the residual. An iterative scheme is needed until (3.28) is satisfied to a given accuracy. Taking into account all of Φ_{ij} enclosed in a circle surrounding a projection, the residual, once again, can be computed using the weighted least squares interpolation,

$$\Phi_n^+(G(s)) = \sum_{ij} \alpha_{ij} \Phi_{ij} + C(s), \tag{3.29}$$

where $C(s)$ is a vector of the correction terms at the projections. We refer the reader to [12] for the details.

The matrix–vector form of such discretization and the interpolation can be written in terms of Φ and G as

$$\begin{bmatrix} A & B \\ E & D \end{bmatrix} \begin{bmatrix} \Phi \\ G \end{bmatrix} = \begin{bmatrix} F \\ V \end{bmatrix}. \tag{3.30}$$

The Schur complement for G then is

$$\begin{aligned} (D - EA^{-1}B)G &= V - EA^{-1}F \\ &\stackrel{\text{def}}{=} \bar{F}. \end{aligned} \tag{3.31}$$

We use the GMRES method to solve the much smaller Schur complement system. Each iteration involves one fast Poisson solver and an interpolation scheme to evaluate the residual. A preconditioning strategy is used to accelerate the convergence. The initial guess has little effect on the number of iterations. So we simply take the initial guess as zero vector. Note that if the interface is a circle, and the domain is infinite, then the solution is indeed zero (see the discussions in the following section).

3.4. *A Preconditioning Strategy*

Since the homogeneous Neumann condition is prescribed on the boundary $\partial\Omega_2$, the solution in the interior of the voids is defined only up to an arbitrary constant. However, the conditition number of the Schur complement depends on the constant solution in Ω_2. This situation is remedied by setting the solution in Ω_2 to be the average value of the solution at all irregular grid points from the other side of the interface at each iteration.

The preconditioning techniques discussed above can be justified using the theory of integral equations. The solution of the Poisson equation can be written as the distribution of sources and dipoles along the boundaries,

$$\phi(\mathbf{x}) = \frac{1}{2\pi}\int_{\partial\Omega_1} \mu_1(s)\log|\mathbf{x}-\mathbf{X}(s)|\,ds + \frac{1}{2\pi}\int_{\partial\Omega_2} \mu_2(s)\frac{\partial}{\partial\mathbf{n}}\log|\mathbf{x}-\mathbf{X}(s)|\,ds. \quad (3.32)$$

The source strengths $\mu_1(s)$ and $\mu_2(s)$ are determined from the boundary conditions on $\partial\Omega_1$ and $\partial\Omega_2$. Several fast Poisson solvers for problems on irregular domains are based on solving the integral equations. However, for a bounded domain, the number of unknowns usually is more than double that in the approach described here because of an additional unknown source strength along $\partial\Omega_1$. Note that the source strength $\mu_2(s)$ corresponds to the jump of the solution across the interface $\partial\Omega_2$. Since the solution inside the voids can be any arbitrary constant, we need to have an additional condition to uniquely determine the solution. A convenient condition is to define the constant as the average of the limit solution from Ω_1 side approaching to $\partial\Omega_2$, that is,

$$C = \int_{\partial\Omega_2} u^+(\mathbf{X})\,ds. \quad (3.33)$$

This condition minimizes the jump in the solution, which is equivalent to minimizing the condition number for the Schur complement. If $\partial\Omega_2$ is a single circle and the overall rectangular domain is very large, then the solution is close to being axi-symmetric, and we will obtain the convergence in a couple of iterations.

Our numerical examples indicate that the number of iterations for solving the Schur complement is small and is independent of the grid size. It turns out this is also true for the cases where V is not zero; see the example in Section 3.5.

In practice, the matrices and vectors in (3.30) and (3.31) are never formed explicitly. The algorithm is outlined below:

Outline of the algorithm to evaluate the electrical potential

1. Initialization: Setting up the grid and the boundary conditions on $\partial\Omega_1$; indexing the grid points as regular or irregular; obtaining projections of irregular grid points; evaluating the normal direction and the curvature information of the interface $\partial\Omega_2$ at the projections of irregular grid points. It is affordable to store these data since the number of irregular grid points is roughly the square root of the total number of grid points.

2. Finding the right hand side of the Schur complement system (3.31). This can be done by setting $G = 0$ in (3.27) to obtain the solution of the Poisson equation and calculating the residual of (3.31).

3. Applying the GMRES iteration with an initial guess of G. The main process of the GMRES method requires only matrix–vector multiplications. This involves essentially two steps in our algorithm. The first step is to solve the Poisson equation, which corresponds to the matrix–vector multiplication in the GMRES method. The main cost of this stage is obtaining the correction term at each irregular grid point due to the jump in the solution. The fast Poisson solver that we employed is the modified HWSCRT routine from FISH Package. The second step is to interpolate the solution of the Poisson equation to obtain the residual.

3.5. *Numerical Examples for the Evaluation of the Poisson Equation*

We provide one example with different boundary conditions to show the efficiency of the algorithm proposed above. This is the most costly part in the simulation of electro-migration voids. We want to verify second order accuracy of the solution procedure, and more important, also to verify the assumption that the number of iterations is nearly independent of the mesh size except for a factor of $\log h$.

We construct an exact solution:

$$\begin{aligned} &\phi(x, y) = -\tfrac{1}{2}\log r + r^2, \qquad r = \sqrt{x^2 + y^2}, \\ &f(x, y) = 4. \end{aligned} \tag{3.34}$$

The boundary $\partial\Omega_1$ is the unit rectangle: $-1 \le x, y \le 1$. The interior boundary $\partial\Omega_2$ is an ellipse

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

Case 1. The Dirichlet boundary condition on $\partial\Omega_1$ and the normal derivative boundary condition on $\partial\Omega_2$ are given using the exact solution. The first part of Table I shows the grid refinement analysis and other information with $a = 0.5$ and $b = 0.4$.

In Table I, n is the number of grid lines in the x- and y-directions; e is the error of the computed solution in the maximum norm; n_1 is the total number of irregular grid points; n_2 is the number of irregular grid points on one side, which is also the dimension of the Schur complement; k is the number of iterations of the GMRES method, which is also the number of the fast Poisson solver called. We observe a second order rate of convergence in the maximum norm. The number of iterations for the GMRES iteration is proportional to $\log h$. Note that the error may not be reduced exactly by a factor of four but rather may fluctuate, as explained in [12].

Case 2. The normal derivative $\partial\phi/\partial n$ is prescribed on $\partial\Omega_1$ using the exact solution. In this case, the solution is not unique and the compatibility condition must be imposed in order to obtain a reasonably accurate solution. To obtain a unique solution, we specify the solution at one corner using the exact solution. In this way, we can measure the error of the computed solution. Table II shows the results of the grid refinement analysis. We have results similar to those analyzed above.

TABLE I

The Grid Refinement Analysis with $a = 0.5$, $b = 0.4$, and $b = 0.15$

n	a	b	e	r	n_1	n_2	k
40	0.5	0.4	5.7116×10^{-4}		100	52	16
80	0.5	0.4	1.4591×10^{-4}	3.9146	204	104	17
160	0.5	0.4	3.5236×10^{-5}	4.1408	412	208	19
320	0.5	0.4	8.1638×10^{-6}	4.3161	820	412	21
n	a	b	e	r	n_1	n_2	k
40	0.5	0.15	4.4820×10^{-3}		68	36	13
80	0.5	0.15	1.1466×10^{-3}	3.9089	132	68	15
160	0.5	0.15	2.6159×10^{-4}	4.3832	68	136	17
320	0.5	0.15	6.7733×10^{-5}	3.8621	68	268	20

Note. Dirichlet boundary conditions are prescribed on $\partial\Omega_1$, and Neumann boundary conditions are prescribed on $\partial\Omega_2$. Second order convergence is observed. The number of iterations is almost independent of the mesh size.

TABLE II

The Grid Refinement Analysis for Pure Neumann Type Boundary Conditions

n	a	b	e	r	n_1	n_2	k
40	0.5	0.15	4.5064×10^{-3}		84	44	14
80	0.5	0.15	1.2529×10^{-3}	3.5967	164	84	17
160	0.5	0.15	3.3597×10^{-4}	3.7292	332	168	19
320	0.5	0.15	7.9409×10^{-5}	4.2309	668	336	21

4. EVALUATION OF THE NORMAL VELOCITY OF THE SURFACE

Calculating the surface Laplacian operator along a curve in two dimensions is fairly straightforward. In higher dimensions, however, a nice parameterization of the hypersurface can be very difficult to maintain in a Lagrangian formulation and hence the surface Laplacian operator can be problematic in numerical computation. Since we are using a fixed Cartesian grid we will use a formula for the surface Laplacian operator in Cartesian coordinates. We are using the level set method to capture the moving interface and the geometry is embedded in the level set function. Our surface Laplacian operator thus must also be expressed in terms of this level set function. We start with the two dimensional case and then extend it to any number of dimensions. Let $\boldsymbol{\tau}$ be the tangential direction of the curve and $\mathbf{n}$ be its normal direction. Assume s is the arc length parameter. If f is a function that is defined in a neighborhood of the curve, then the surface Laplacian of f is

$$\Delta_s f = \frac{d^2 f}{ds^2} = \frac{d}{ds}(\boldsymbol{\tau} \cdot \nabla f) = \kappa \mathbf{n} \cdot \nabla f + \boldsymbol{\tau}^T \operatorname{He}(f)\boldsymbol{\tau} = \kappa \frac{\partial f}{\partial n} + \Delta f - \frac{\partial^2 f}{\partial n^2},$$

where κ is curvature and $\operatorname{He}(f)$ is the Hessian of f. Therefore we have

$$\Delta f = \boldsymbol{\tau}^T \operatorname{He}(f)\boldsymbol{\tau} + \mathbf{n}^T \operatorname{He}(f)\mathbf{n} = \frac{\partial^2 f}{\partial \tau^2} + \frac{\partial^2 f}{\partial n^2}.$$

In general, let Γ be an $n-1$ dimensional hyper-surface in R^n and f be a function defined in a neighborhood of Γ. Let $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ be a local coordinate frame of R^n, where $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{n-1}$ are in the tangent plane of Γ and $\mathbf{e}_n$ is the normal of Γ. The surface Laplacian of f on Γ is

$$\Delta_s f = \kappa \frac{\partial f}{\partial e_n} + \sum_{i=1,n-1} \frac{\partial^2 f}{\partial e_i^2} = \kappa \frac{\partial f}{\partial e_n} + \Delta f - \frac{\partial^2 f}{\partial e_n^2}, \tag{4.35}$$

where κ is the mean curvature (sum of the principal curvatures) of the surface. In the level set formulation, with the surface defined by the zero level set of φ,

$$\kappa = \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|}, \qquad \mathbf{e}_n = \frac{\nabla \varphi}{|\nabla \varphi|}, \qquad \frac{\partial f}{\partial e_n} = \frac{\nabla \varphi}{|\nabla \varphi|} \cdot \nabla f, \qquad \frac{\partial^2 f}{\partial e_n^2} = \frac{\nabla \varphi^T}{|\nabla \varphi|} \mathrm{He}(f) \frac{\nabla \varphi}{|\nabla \varphi|}.$$

If $f = \kappa = \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|}$ is the mean curvature of the level set function we obtain a fourth order nonlinear PDE for the level set formulation of surface diffusion of the mean curvature flow. There is very little analysis for this PDE as far as we know. If the level sets of φ are parallel, i.e., $|\nabla \varphi| = \text{constant}$, then

$$\frac{\partial^2 f}{\partial e_n^2} = \mathbf{e}_n \cdot \nabla (\mathbf{e}_n \cdot \nabla f) = \frac{\nabla \varphi}{|\nabla \varphi|} \cdot \nabla \left(\frac{\nabla \varphi}{|\nabla \varphi|} \cdot \nabla f \right).$$

If f is constant in the normal direction, i.e., $\nabla f \cdot \mathbf{e}_n = 0$, we have $\partial f / \partial e_n = 0$. Numerically if we can make φ a distance function (through the re-initialization process) and extend f such that $\nabla f \cdot \nabla \varphi = 0$ (see [17]), then we have

$$\Delta_s f = \Delta f,$$

which is very easy to compute on a Cartesian grid. For accuracy we prefer to use (4.35) in our numerical computation.

4.1. *Evaluation of the Surface Velocity due to the Electrical Potential*

The electric potential function is only defined outside of the voids. However, for the level set formulation, we need the velocity field in a thin tube surrounding the interface. The difficulty is that we do not have any information inside the voids and the computed potential is only second order accurate near or on the interface where the error of the potential is not smooth. As a result, oscillations in the velocity will be present after second order differentiation. Such oscillations will also affect the choice of the time step and area conservation (see Section 4.3 as well). Our strategy is to use a second order extrapolation to extend the information outside of the voids to the irregular grid points inside, and then to compute the derivative along the tangential direction to obtain the normal velocity of the surface due to the variation in the electric potential. There are a variety of choices as to when and how to do the extrapolation along the normal direction and differentiating along the tangential direction. The scheme used in our simulation is the following:

- Use the standard central scheme to compute

$$\frac{\partial \phi}{\partial s} = -\frac{\partial \phi}{\partial x} n_y + \frac{\partial \phi}{\partial y} n_x \tag{4.36}$$

at *regular* grid points which lie *inside* the computational tube but *outside* the voids.

• Use the weighted least squares interpolation to obtain $\frac{\partial \phi}{\partial s}$ at all *irregular* grid points, both inside and outside the voids by extrapolation and interpolation, respectively.

• Extend $\frac{\partial \phi}{\partial s}$ from the interface along the normal direction $\mathbf{n}$ to all grid points as well as to those points which are located inside the void within the computational tube using the formula

$$\left(\frac{\partial \phi}{\partial s}\right)_t - \left(\nabla \frac{\partial \phi}{\partial s}\right) \cdot \frac{\nabla \varphi}{|\nabla \varphi|} \operatorname{sign}(\varphi) = 0. \tag{4.37}$$

Here $\operatorname{sign}(\varphi)$ is the sign function of φ. It switches the direction of information propagation along characteristics at the interface. For more information on velocity extension, see also [6, 13, 17, 22].

• Finally, use the standard central scheme to compute the part of the surface velocity due to the electric potential

$$\frac{\partial}{\partial s}\left(\frac{\partial \phi}{\partial s}\right) = -\frac{\partial}{\partial x}\left(\frac{\partial \phi}{\partial s}\right) n_y + \frac{\partial}{\partial y}\left(\frac{\partial \phi}{\partial s}\right) n_x. \tag{4.38}$$

For two dimensional surface diffusion problems, the total internal area of voids should remain unchanged, by the consideration

$$\int\int \nabla \cdot \mathbf{u}\, dx\, dy = \oint V_n\, ds = \oint \frac{\partial^2}{\partial s^2}(C_1 \phi + C_2 \kappa)\, ds \equiv 0.$$

In our method, the surface velocity is proportional to the tangential derivative of a function. Theoretically and numerically this approach preserves the area better than the approach that extends the velocity after it is computed at or near the interface.

4.2. *Evaluation of the Surface Velocity due to the Curvature*

In two dimensions, it is relatively straightforward to evaluate the surface velocity due to the variations in curvature since the curvature is defined at all grid points. It can be done by finding the second directional derivative of the curvature; see (4.36) and (4.38). Since the computational process involves fourth order derivatives of the level set, one may be tempted to use a fourth order scheme to compute the first order derivatives, a third order scheme to compute the second order derivatives, and so forth. While this approach is sound in principle, it results in larger errors compared with the standard central difference scheme and requires a larger width for the computational tube used in the local level set method. The time step size also needs to be cut to keep stability. Furthermore, with wider support and fourth order derivatives, special care must be taken to handle the boundary conditions of the voids. Numerically, a high order method does not give better results in our case because errors can be amplified and oscillations can occur at or near the interface. We find that the standard central difference scheme, also used in [1] for etching and deposition problems, works best in our numerical experiment. In fact, it can be shown that, if a function is sufficiently smooth, the computed fourth order derivatives using the standard central difference scheme are second order accurate. In [1], a two dimensional model involving only surface diffusion terms is considered. There are no topological changes in the geometry.

After we computed the normal velocity of the boundaries of the voids, we used the weighted ENO (essentially non-oscillatory) scheme to update the level set function and

perform the re-initialization process. The WENO scheme is more stable and accurate and the errors are smoother than those of the ENO scheme (see [8]).

4.3. *Choosing Adaptive Time Step Steps*

The CFL condition for the surface diffusion is

$$\Delta t_1 \le h^4/C_2. \tag{4.39}$$

The CFL condition for the Hamilton–Jacobian equation in updating the velocity is

$$\Delta t_2 \le h/U_{\max}, \tag{4.40}$$

where $U_{\max}$ is the largest magnitude of the normal velocity in the computational tube. Since the electrical potential is the inversion of the Laplacian operator, the CFL condition for its surface diffusion is

$$\Delta t_3 \le h^2/C_1. \tag{4.41}$$

The adaptive time step is chosen from the smallest of the three for the explicit method. Usually C_2 is relatively small compared with C_1, and the time step is tolerable for the simulations on most workstations. However, it is known that the desired level set function is the signed distance function. Such a level set function has kink at some points or along a curve where $|\nabla\phi|$ is not well defined (see Fig. 4). When the computational tube contains those points, the calculations are not correct at those points since the derivatives do not exist. The normal velocity can have very large magnitudes at these points, leading to very small time step sizes and sometimes instabilities. Another mathematical concern is, as we pointed out in the Introduction, that if each level set function moves according to its own geometry and motion law independently the sets will run into each other due to the lack of a maximum principle in this problem. Our numerical solution to this problem is to use a cut-off function $\omega(x)$,

$$\omega(x) = \begin{cases} 1 & \text{if } x \le 0 \\ (x-1)^4(20x^3 + 10x^2 + 4x + 1) & \text{if } 0 < x \le 1 \\ 0 & \text{if } 1 < x. \end{cases} \tag{4.42}$$

The new velocity is defined as

$$U_n^{\text{new}} = \omega\left(\frac{|\varphi| - \varphi_0}{\delta}\right) U_n. \tag{4.43}$$

In a very thin tube containing the interface, $|\varphi| \le \varphi_0$, the normal velocity is unchanged. It is gradually lowered to zero to take care of the troublesome points without a significant effect on accuracy. We choose $\delta = 3h$ and φ_0 is between $2h$ and $4h$ in our numerical tests. Notice that we used a cut-off function with up to third order continuous derivatives, in an effort to reduce the oscillations introduced by the cut-off near the boundary. Numerically speaking, the level set function provides body fitted parallel frames (level sets) near the interface which can capture both the geometry and the motion of the interface on a Euclidean framework.

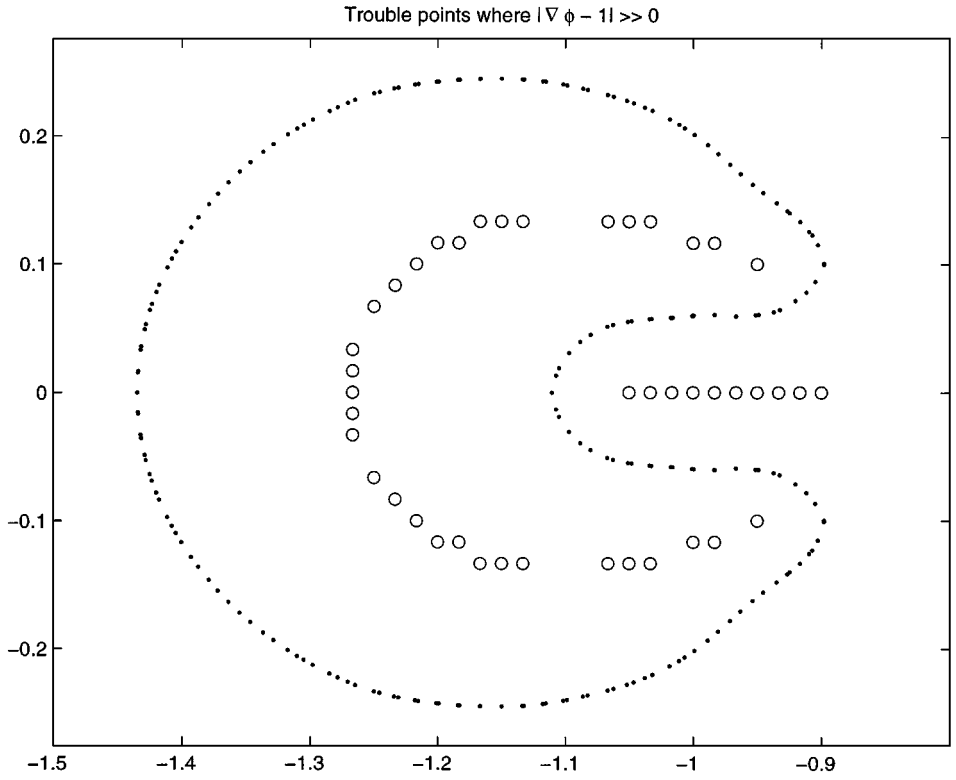

FIG. 4. Grid points shown as open circles indicate points that are either on or close to the kink of the signed distance function; the solid dots are projections of irregular grid points on the interface.

Physically and mathematically, what really matters is just the motion of level sets very close to the interface. Due to a numerical CFL condition and correct scaling of the level set function (distance function) the level set formulation is guaranteed to be trouble free. The reason why we use a cut-off function instead of extending the normal velocity is to enforce area conservation, as we mentioned before. Our approach effectively eliminates the possibility of having overlarge normal velocities near the kink of the level set function near the interface.

Remark 4.1. When topological changes take place, we can no longer cut off the kink. The calculation may lose some accuracy at those points. On the other hand, we need some mathematical insights before we can fix the problem numerically. Some different approaches combining perturbation techniques, the particle method, and/or the boundary integral method with the level set formulation are under investigation.

5. NUMERICAL EXPERIMENTS

We have performed a series of numerical experiments on the interaction of voids during the electro-migration process in an attempt to draw some physical insights from these experiments. For all the tests shown here, we used Neumann boundary conditions (flux) at the left and right edges of the rectangles, and homogeneous Neumann boundary conditions at the top and bottom of the rectangles.

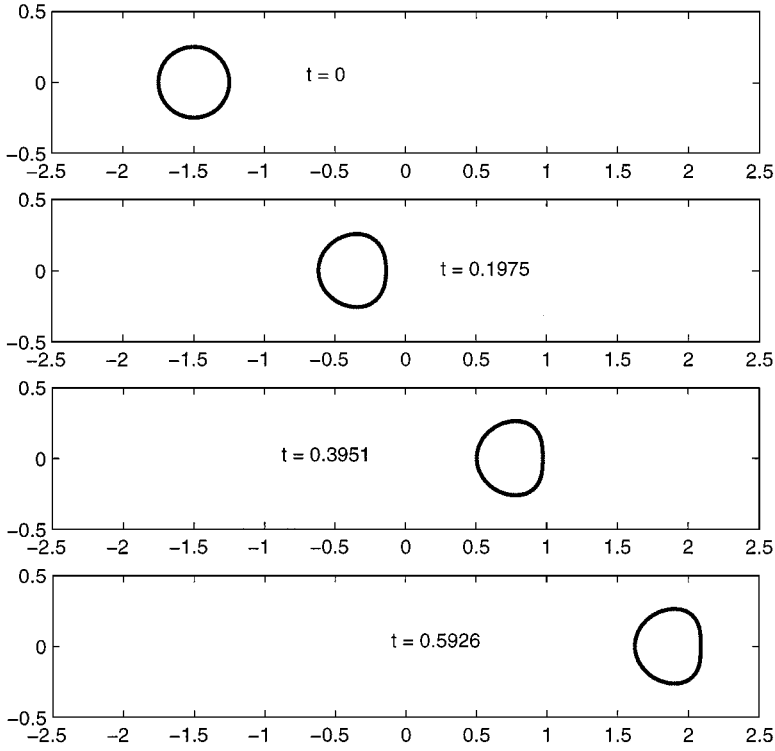

FIG. 5. A void evolution with large chemical potential. The motion is relatively stable.

In the first two experiments, we try to reproduce the results from [20]. The computational domain is the rectangle $-2.5 \le x \le 2.5$, $-0.5 \le y \le 0.5$. The initial void is a circle centered at $(-1.5, 0)$ with radius $r = 0.25$. In the first case, the coefficients of the potentials are taken to be $C_1 = 0.625$ and $C_2 = 0.625 \times 6.25 \times 10^{-3}$, corresponding to the parameters chosen in [20]. The computational grid is 300×60, or $h = 1/60$. Figure 5 shows the evolution of the void with time. For this test problem, the surface tension is relatively large and the configuration is stable. The relative area change for this case is less than 0.0619%.

In the second case see (Fig. 6), the coefficient of the chemical potential, $C_2 = 1.875 \times 2.08333 \times 10^{-4}$, is small compared to that of the electrical potential, $C_1 = 1.875$, and the motion is less stable than that in the first case. The grid spacing is $h = 1/100$. The area loss is about 6% just before the void breaks up. Our simulation has gone beyond the results obtained in [20]. The evolution process seems to agree with the results in that paper up to some time by comparing the graphs there. Note that we used Neumann boundary conditions at the left and right edges of the rectangle while Dirichlet boundary conditions were used in [20]. Usually it is harder to solve Poisson equations with Neumann boundary conditions than with those of Dirichlet type.

As discussed in [20], the stability of the void shape depends on the ratio between the driving force associated with the surface energy and that associated with the electro-migration. For fixed line geometry and void size, this ratio is proportional to C_1/C_2. Our first two test cases, which essentially reproduce the corresponding examples given in [20], demonstrate that a nearly circular void shape is stable when $C_2/C_1 = 6.25 \times 10^{-3}$ and unstable at a smaller ratio when $C_2/C_1 = 2.08333 \times 10^{-4}$. In the first test case, and for all cases with larger C_2/C_1, the nearly circular void shape is stable because the dominating driving force

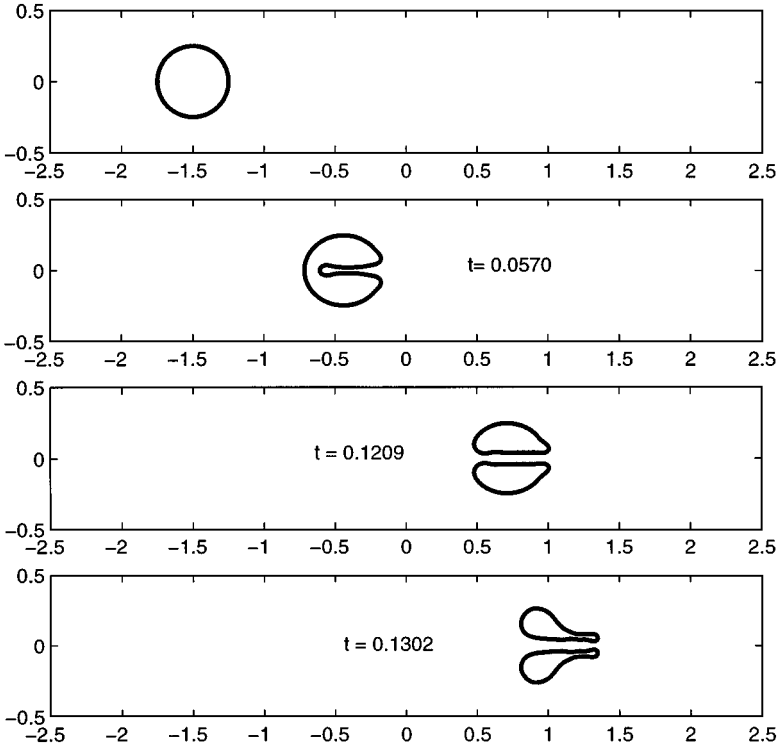

FIG. 6. A void evolution with small chemical potential. The motion is less stable.

is the surface energy, which tends to resist deviation from the circular shape. In the second case, and for all cases with smaller C_2/C_1, electro-migration forces become dominant and cause drastic deviations from the circular void shape. Figure 7 shows an evolution process with even smaller C_2/C_1. The calculation was done with a coarse grid, $h = 1/50$. While we see more loss in the area, we see a similar pattern in the evolution process. The loss in the area is due to the poor resolution and the large curvatures that appear immediately following topological change.

The algorithm developed here can be used to study the interactions of multiple voids in an electro-migration line. Figure 8 shows the evolution process of two voids. The first one is a circle centered at $(-2.1, 0)$ with radius 0.15. The second circle is centered at $(-1.55, 0)$ with radius 0.3. The coefficient of the chemical potential is chosen as $C_2 = 3.9062 \times 10^{-5}$ and the coefficient of the electrical potential 1.875, with the ratio $C_2/C_1 = 2.08 \times 10^{-5}$ in the range of instability. The electrical potential is dominant and the motion is toward a more unstable pattern with the creation of a number of small voids from the larger voids through shape instabilities. In this case, $h = 0.01$ and the area change is less than 2.7%. Most of the loss in the area occurs after small voids are produced. The large curvature of the small voids compensates for the small chemical potential coefficient C_2 relative to the electro-migration coefficient C_1. This is not unexpected because a small C_2/C_1 indicates shape instability of large voids, which tend to break up to into smaller voids in order to seek a balance between the driving force associated with the surface energy and that associated with the electro-migration. The same argument would indicate that conditions of small C_2/C_1 tend to prevent merger of voids into larger ones, which is consistant with our observations. In this simulation, we see also that small voids move faster than large ones.

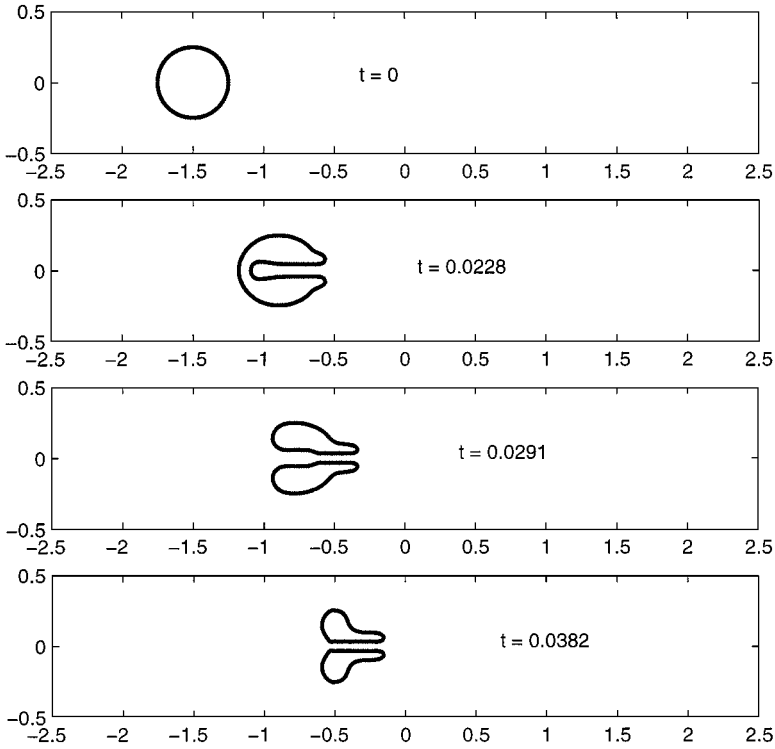

FIG. 7. A void evolution with very small chemical potential on a coarse grid.

It appears that the interaction of voids evolves toward an energetic balance between the surface tension and electro-migration. We have already seen that larger voids tend to break up into smaller ones. If the surface tension force is sufficiently large, two or more smaller voids may merge to form larger voids, as is demonstrated in Fig. 9, where we take $C_2 = 0.0391$ and $C_1 = 1.875$. In the simulation, $h = 1/60$. Initially the voids are two ellipses: The first one is centered at (0.35, 0.5) with $a = 0.2$ and $b = 0.14$, where a is the major and b is the

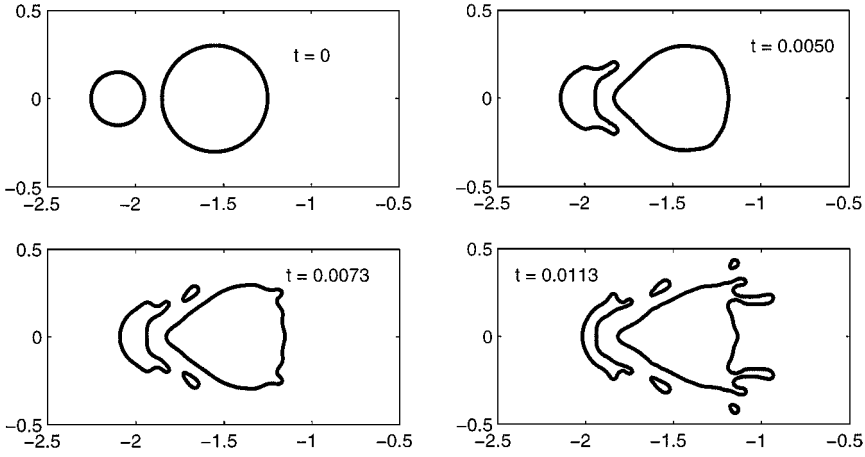

FIG. 8. An evolution process of two voids with small chemical potentials.

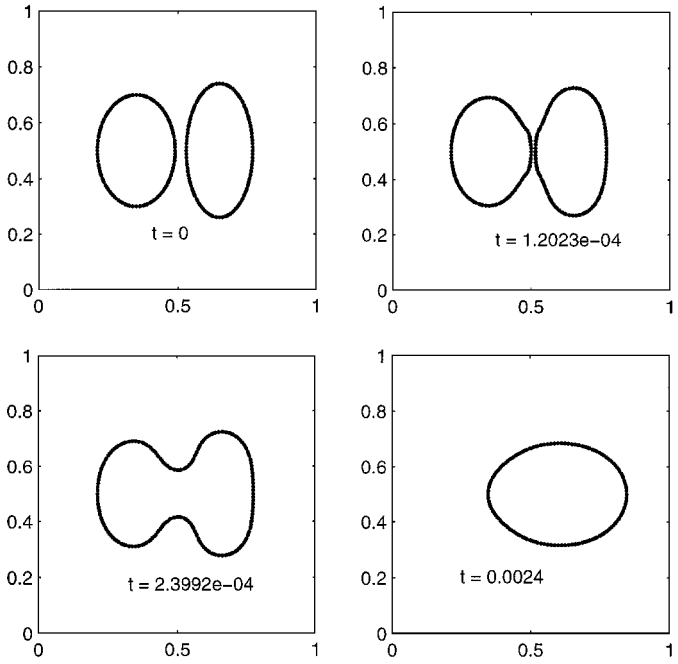

FIG. 9. An evolution process of two voids with large chemical potentials.

minor axis. The ellipse to the right is centered at (0.65, 0.5) with $a = 0.24$ and $b = 0.12$. The area change is about 10%, which is much worse than the values in our previous cases. We believe there are two factors which affect the loss of the area: The major factor is the large h, which, of course, means lack of resolution in the simulation. For problems involving fourth order derivatives, we find that reasonable resolution is necessary to preserve the area. The second factor is the singularity before or after the topological changes, i.e., the merging, where some parts of the level set have very large curvatures.

Figure 10 shows the interaction of two voids under a relatively large C_2/C_1 ratio. The first one is an ellipse centered at $(-2.1, 0)$ with the major axis 0.3 and the minor axis 0.2. The force coefficients are $C_1 = 1.875$ and $C_2 = 0.007324$. The void to the right is a circle centered at $(-1.5, 0)$ with radius 0.2. The electric field becomes concentrated between the boundary of the conducting line and the larger void. Due to this concentration, the upper and lower tips of the larger void are driven faster than the medium portion and the smaller void is shielded from the electric field by the presence of the large void. As a result, the large void wraps around the small void without breaking up into smaller voids. The case demonstrates the strong shielding effect of larger voids when they are relatively stable.

As a final example, we investigate the effect of initial shape of a void. Figure 11 shows the evolution process of an ellipse centered at $(-1.8, 0)$ with the major axis 0.4 and the minor axis 0.2. The coefficients are $C_1 = 1.875$ and $C_2 = 0.3.90625 \times 10^{-3}$. The surface tension force tends to smooth the void shape into a circle. The effect of electrical potential, on the other hand, causes a protrusion of the void with a relatively large curvature at the frontal tip of the void. This sharp tip moves rapidly under the electro-migration forces and drags the rest of the void along. Such a worm-like motion of void demonstrates another delicate balance between surface tension and electro-migration forces.

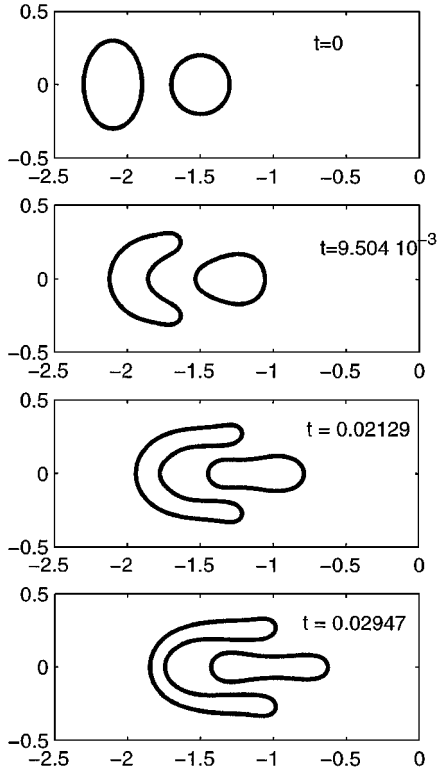

FIG. 10. An evolution process of two voids with large chemical potentials.

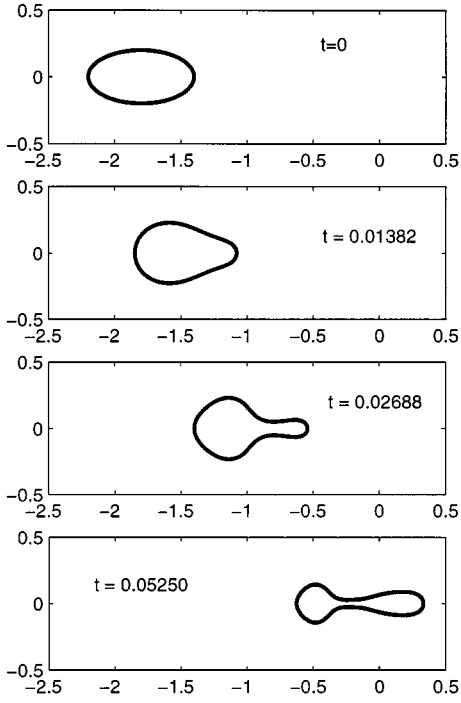

FIG. 11. An evolution process of two voids with large chemical potentials.

6. CONCLUSION

In this paper, we have described a numerical method for the simulation of moving interface problems which arise in electro-migration voiding in integrated circuits. In comparison with a conventional finite element- or a boundary element-based numerical method with adaptive meshing, our method has the advantage of employing a fixed Cartesian grid without the need of re-meshing at each time step. A fast iterative method using the GMRES iteration has been adopted to solve the Laplace equations associated with electric potential on the exterior of voids. The evolution of void boundaries is described by a two dimensional level set function which is updated at each time step according to surface diffusion equations governed by combined driving forces associated with surface tension and electric potential. The level set method is used to efficiently update the motion of voids and is capable of capturing complex topological changes of voids such as void merging and void break-up. Our numerical examples have reproduced and gone beyond some test cases in the literature using a finite element-based simulation method. We have also performed a series of numerical experiments to elucidate the mechanisms by which electro-migration voids interact. It appears that the interaction of multiple voids tends to evolve the shape and size of the voids into a state in which the surface tension is better balanced with the electro-migration force. This is demonstrated in examples where small voids dominated by surface tension tend to merge into larger voids while large voids dominated by electromigration forces tend to undergo an instability and break up into smaller voids. Also, the motion of small voids could be significantly retarded by large voids in their vicinity as the electric field at small voids is shielded by the large voids, especially when the sizes of the voids are on the order of the conducting line width. The method that we propose does not require a sophisticated meshing technique and makes a relatively low demand on computational resources. All the simulations involve interaction of multiple voids and can be done rapidly on common workstations. It is also relatively straightforward to generalize the method to three dimensional simulations where the computational savings over re-meshing is expected to be even more significant.

It is true that for fourth order differential equations, the time step limitation is still a concern for explicit algorithms. An implicit or semi-implicit level set method is under investigation. Adaptive Cartesian grid refinement techniques will be considered also in further development of the method discussed in this paper.

ACKNOWLEDGMENT

The authors thank the referees for their useful comments and suggestions.

REFERENCES

1. D. Adalsteinsson and J. Sethian, A level set approach to a unified model for etching, deposition, and lithography. III. Redeposition, reemission, surface diffusion, and complex simulations, *J. Comput. Phys.* **138** (1997).
2. A. J. Bernoff and A. L. Bertozzi, Singularities in a modified Kuramoto-Sivashinsky equations describing interface motion for phase transition. *Physica D* **85**, 375 (1995).
3. A. J. Bernoff, A. L. Bertozzi, and T. P. Witelski, Axisymmetric surface diffusion: Dynamics and stability of self-similar pinch-off, preprint, 1998.
4. J. W. Cahn, C. M. Elliott, and A. Novick-Cohen, The Cahn–Hilliard equation with a concentration dependent mobility: Motion by minus the Laplacian of the mean curvature, *Eur. J. Appl. Math.* **7**, 287 (1996).

5. J. W. Cahn and J. E. Taylor, Surface motion by surface diffusion, *Acta Metall. Mater.* **42**, 1045 (1994).
6. S. Chen, B. Merriman, P. Smereka, and S. Osher, A fast level set based algorithm for Stefan problems, *J. Comput. Phys.* **135**, 8 (1997).
7. C. M. Elliott and S. Zheng, The Cahn–Hilliard equation, *Arch. Rational Mech. Anal.* **96**, 339 (1996).
8. G. Jiang and D. Peng, *Weighted ENO Schemes for Hamilton–Jacobi Equations*, UCLA CAM Report 97-29, 1998.
9. O. Kraft and E. Arzt, Electromigration mechanisms in conductor lines: Void shape changes and slit-like failure, *Acta Mater.* **45**, 1599 (1997).
10. R. J. LeVeque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* **31**, 1019 (1994).
11. Z. Li, *The Immersed Interface Method—A Numerical Approach for Partial Differential Equations with Interfaces*, Ph.D. thesis, University of Washington, 1994.
12. Z. Li, A fast iterative algorithm for elliptic interface problems, *SIAM J. Numer. Anal.* **35**, 230 (1998).
13. Z. Li and B. Soni, Fast and accurate numerical approaches for Stefan problems and crystal growth, *Numer. Heat Transfer, Part B*, in press.
14. W. W. Mullins, Mass transport at interface in single component systems, *Metall. Trans. A* **26**, 1917 (1995).
15. F. A. Nichols and W. W. Mullins, Surface (interface) and volume-diffusion contributions to morphological changes driven by capallarity, *Trans. Metall. Soc. AIME* **233**, 1840 (1965).
16. S. Osher and J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).
17. D. Peng, S. Osher, B. Merriman, H. Zhao, and M. Kang, *A PDE Based Fast Local Level Set Method*, UCLA CAM Report 98-25, 1998.
18. R. F. Sekerka and T. F. Marinas, Dynamics of morphological change during solid–solid transformations, in *Solid–Solid Phase Transformations*, edited by H. I. Aaronson *et al.* (Metallurgical Society of AIME, 1982), pp. 67–86.
19. J. E. Taylor and J. W. Cahn, Linking anisotropic sharp and diffuse surface motion laws via gradient flows, *J. Statist. Phys.* **77**, 183 (1994).
20. L. Xia, A. F. Bower, Z. Suo, and C. Shih, A finite element analysis of the motion and evolution of voids due to strain and electromigration induced surface diffusion, *J. Mech. Phys. Solids* **45**, 1473 (1997).
21. Z. Yang, *A Cartesian Grid Method for Elliptic Boundary Value Problems in Irregular Regions*, Ph.D. thesis, University of Washington, 1996.
22. H. Zhao, T. Chan, B. Merriman, and S. Osher, A variational level set approach to multiphase motion, *J. Comput. Phys.* **127**, 179 (1996).