

UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

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

Zhilin Li
Hongkai Zhao
Huajian Gao

November 1998
CAM Report 98-45

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

A NUMERICAL STUDY OF ELECTRO-MIGRATION VOIDING BY EVOLVING LEVEL SET FUNCTIONS ON A FIXED CARTESIAN GRID

ZHILIN LI*, HONGKAI ZHAO†, AND HUAJIAN GAO‡

Abstract. A numerical method is developed to study migration of voids driven by surface diffusion and electric current in a metal conducting line. 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 an sufficient accuracy to afford fourth order differentiation along the void boundary and to capture singularities arising in topological changes. We use a modified immersed interface method with a fixed Cartesian grid to solve 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 electro-migration.

Key words. electro-migration, moving interface/boundary problems, immersed interface method, level set method, surface diffusion.

AMS subject classifications. 65M06, 65C20, 35R35

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 causes motion or relocation of boundaries. The dynamics of these processes has been of great interests to material scientists and biologists. There is a large literature on this topic. We refer to the readers to two recent survey articles by Mullins[13], and Cahn and Taylor[5] and the references therein. The problem considered in this paper involves interaction of voids under electro-migration in a conducting metal line where the driving forces for diffusion are the gradient of curvature and electric potential along the void boundary. The normal velocity of the void surface is given by the partial differential equation (PDE)

$$(1.1) \quad U_n = \Delta_s (C_1 \phi + C_2 \kappa),$$

where Δ_s is 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 physical constants as

$$(1.2) \quad C_1 = \frac{eD_s\Omega^{1/3}Z^*}{k_B T_k}, \quad C_2 = \frac{D_s\Omega^{4/3}\gamma_s}{k_B T_k},$$

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

$$(1.3) \quad D_s = \frac{D_s^* \delta_s}{k_B T_k} e^{-Q_s/k_B T_k},$$

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

*Center For Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205. The author is partially supported by an NSF grant DMS-96-26703 and North Carolina State FR&PD Fund.

†Department of Mathematics, Stanford University, Stanford, CA 94305.

‡Department of Mechanical Engineering, Stanford University, Stanford, CA 94305-4040.

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 equation (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. Boundary evolution governed by surface Laplacian of mean curvature, as studied by numerous authors [3, 4, 5, 13], can be regarded as a gradient flow with H^{-1} inner product for the surface area given by the following equation

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

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 a combined surface diffusion and growth process. The connection between sharp and diffuse interface motion laws via gradient flow can be found in [18]. 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 [14, 17] indicates Rayleigh instability at long wavelength perturbations. It has been shown with 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 has also been 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 [15], we have

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

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 arclength, we get the time evolution equation for 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

$$(1.5) \quad \kappa_t = -\kappa^2 \kappa_{ss} - \kappa_{ssss}.$$

We can see from the evolution equation that the first term $-\kappa^2 \kappa_{ss}$ is second order and nonlinear which causes the 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 in s , after integration by part we have the following 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 is discussed in [6] which shows that the close 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 transition [2].

In this paper we are going to address some challenges in numerical implementation of moving boundary problems arising in electro-migration voiding. The main challenges for the present problem include (a) constructing an efficient and accurate Poisson solver in arbitrary domain, (b) tracking topological changes

(breaking, merging) along a moving interface, (c) evaluating surface Laplacian operator along the interface. We wish to explore the possibility of simulating a moving interface on a fixed Cartesian grid without having to re-mesh as the interface migrates. For this purpose, we introduce the modified immersed-interface-method developed (IIM) in [9, 10] 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 equation (1.1). We will derive a general formula for 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 can not be arbitrarily chosen. Since each contour of the level set function moves according to its own curvature variation and, due to the lack of a maximum principle, different contours may cross each other. We adopt a modified level set function method [21] which allows us to get around this problem by, (i) using 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 quantities at the interface, if necessary, to other level sets so that they are constant normal to the interface. It has been shown [21] that the extension of normal velocity moves all contours of the level set function according to the zero-level contour and maintain the normal distance between different contours. After these modifications plus CFL constraint we will use level set formulation to capture moving interfaces in this case. An alternative approach to a moving interface problem is to use the finite element formulation with adaptive re-meshing [8, 19]. 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 resource and may be desirable as a practical analysis tool for many applications.

2. Numerical Implementation. We consider a moving boundary with the normal velocity given by surface diffusion under a linear combination of electrical potential and surface tension:

$$(2.1) \quad U_n = \Delta_s = \frac{\partial^2}{\partial s^2} (C_1 \phi + C_2 \kappa)$$

where ϕ is the electrical potential which is a solution of the Laplace equation on the domain outside of voids, see Sec. 3.1; 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 algorithms 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 normal and tangential directions, projections of irregular grid points, curvatures, etc.
- Use IIM to solve the electric potential function, see Sec. 3.
- Find the component of the normal velocity of the surface due to the electric potential, see Sec. 4.1.
- Find the component of the normal velocity of the surface due to the curvature variation, see Sec. 4.2.
- Find the normal velocity for the level set function.
- Update the level set function.
- Reinitialize the level set function.
- Go to the next time level.

3. A fast algorithm to compute the electrical potential on irregular domains. We need to solve the Poisson equation

$$(3.1) \quad \Delta \phi = f(x, y), \quad (x, y) \in \Omega_1,$$

$$(3.2) \quad \left. \frac{\partial \phi}{\partial n} \right|_{\partial \Omega_2} = v(s),$$

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. 3.1. There is also a compatibility condition if the boundary conditions are of Neumann type. In our electro-migration problem, $v(s) \equiv 0$.

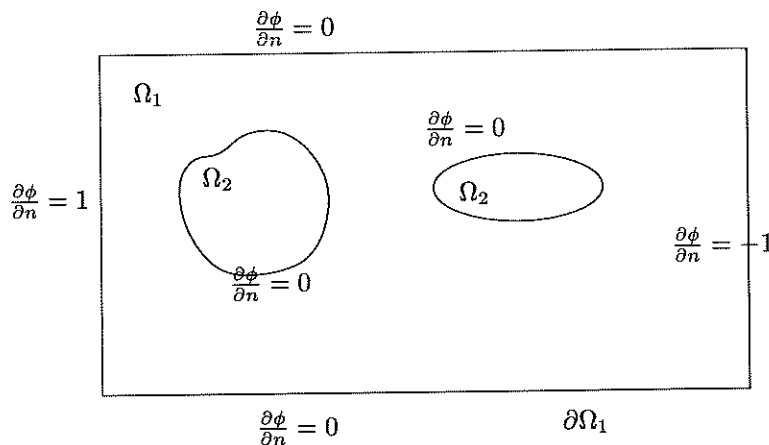


FIG. 3.1. A sketch of a computational domain. The regions of Ω_2 are voids. We want to track the motion of their boundaries.

In recent years, the level set method, first proposed in [15], has become a powerful tool for computing moving boundary/interface problems, especially the problems involve topological changes or in three dimensions. We intend to present a simple, second order discretization method for solving (3.1) with the interface described by a level set function. There are variety of methods based on finite difference for this kind of problems, for example, the capacitance matrix method, and fast method based on the integral equations. However, many of these methods require explicit information of 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 [11, 20], and 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

$$(3.3) \quad \Delta\phi = f(x, y), \quad (x, y) \in \Omega_1 \cup \Omega_2,$$

$$(3.4) \quad \left. \frac{\partial\phi^+}{\partial n} \right|_{\partial\Omega_2} = v(s),$$

where $f(x, y)$ is defined as zero in Ω_2 . We use $+$ 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 is specified. If a single layer is introduced, then there is a jump in the normal derivative. If a double layer is introduced, the function is discontinuous across the interface. Usually, we can not require both the function and its normal derivative be continuous across the interface at the same time. Following the idea introduced in [11, 20], 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 change in the gradient of the function, and an ill-conditioned system for the unknown jump in the derivative.

Now consider the following functional $\phi(g)$:

$$(3.5) \quad \begin{aligned} \Delta\phi &= f(x, y), & (x, y) \in \Omega_1 \cup \Omega_2, \\ [\phi]|_{\partial\Omega_2}(s) &= g(s), & [\phi_n]|_{\partial\Omega_2} = 0. \end{aligned}$$

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 located away from the interface and *irregular* ones are those where the interface cuts through the standard five point stencil. The attention is focused on the irregular grid points.

A two dimensional level set function $\varphi(x, y)$ is introduced where $\varphi(x, y) = 0$ describes the location of the interface. 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 $\vec{X} = (x_i, y_j)$ is an irregular grid point, its projections $\vec{X}^* = (x^*, y^*)$ can be found using the following procedure:

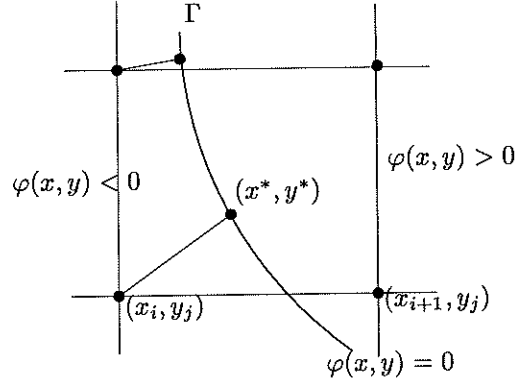


FIG. 3.2. Finding the control point \vec{X}^* from an irregular grid point (x_i, y_j) , $\varphi(x_i, y_j) \leq 0$. It can be chosen as the projection of the grid point on the interface.

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

$$(3.6) \quad \vec{p} = \frac{\nabla \cdot \varphi}{\|\nabla \cdot \varphi\|}.$$

2. Locate the projection of \vec{X} on the interface along the direction \vec{p} :

$$(3.7) \quad \vec{X}^* = \vec{X} + \alpha \vec{p},$$

where α is determined from the following quadratic equation:

$$(3.8) \quad \varphi(\vec{X}) + \|\nabla \cdot \varphi\| \alpha + \frac{1}{2} (\vec{p}^T \text{H}e(\varphi) \vec{p}) \alpha^2 = 0,$$

and $He(\varphi)$ is the Hessian matrix of φ

$$(3.9) \quad He(\varphi) = \begin{bmatrix} \varphi_{xx} & \varphi_{xy} \\ \varphi_{yx} & \varphi_{yy} \end{bmatrix}$$

evaluated at \vec{X} .

We only define $g(s)$ on those projections from a particular side, for example, $\varphi(x, y) \geq 0$. We will call them as \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 *order* the projections where the unknown jumps are defined. In other words, the quantities remain local, an very important feature of the level set formulation compared with the Lagrangian formulation.

3.2. Discretization of the Poisson equation with a jump function. Given jump conditions across the interface $\partial\Omega_2$:

$$(3.10) \quad [\phi] = g(s), \quad [\phi_n] = 0, \quad [f] = f^+(s)$$

at all projections, we use the immersed interface method to solve the Poisson equation. The essence of this method [9, 10, 11] 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 [9, 10, 11]. The interface information such as the tangential and normal derivatives, and the curvature at projections are obtained from the values of the level set function at grid points plus a bi-linear interpolation, see [11] for details. A key part is in evaluating the first and second derivatives of the jump $[\phi]$ along the interface.

Evaluate the derivatives of the jump function $[\phi]$. The idea is to use the weighted least squares interpolation [11] to approximate the derivatives of the jump along the interface at a given projection at $\vec{X}^* = (x^*, y^*)$. We use the local coordinates at (x^*, y^*) in the tangential and normal directions:

$$(3.11) \quad \begin{aligned} \xi &= (x - x^*) \cos \theta + (y - y^*) \sin \theta, \\ \eta &= -(x - x^*) \sin \theta + (y - y^*) \cos \theta, \end{aligned}$$

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 $1.5h \sim 4h$ so that at least three projections from the \mathbf{Set}_I are enclosed. If we can find the signed arc-length starting from (x^*, y^*) , then 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

$$(3.12) \quad \xi(\eta) = C\eta^2 + D\eta^3 + O(\eta^4).$$

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

$$(3.13) \quad \begin{aligned} \frac{2C\eta_1 + 3D\eta_1^2}{\sqrt{1 + (2C\eta_1 + 3D\eta_1^2)^2}} &= -n_\eta, \\ \frac{1}{\sqrt{1 + (2C\eta_1 + 3D\eta_1^2)^2}} &= n_\xi, \end{aligned}$$

where (ξ_1, η_1) is the coordinates under the transformation (3.11). We arrive at the following linear system of equation for C and D :

$$(3.14) \quad \begin{aligned} C\eta_1^2 + D\eta_1^3 &= \xi_1, \\ 2C\eta_1 + 3D\eta_1^2 &= -\frac{n_\eta}{n_\xi}. \end{aligned}$$

In other words, the curve is approximated by a Hermite spline interpolation between \vec{X}^* and \vec{X}_1^* . The arc-length between \vec{X}^* and \vec{X}_1^* is determined from

$$(3.15) \quad |s_1| = \int_0^{\eta_1} \sqrt{1 + (2C\eta + 3D\eta^2)^2} d\eta.$$

This definite integral can be approximated by the Simpson rule or Gaussian quadrature formula. 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 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 \vec{X}^* and \vec{X}_1^* :

$$(3.16) \quad s_1 = \begin{cases} |s_1| & \text{if } (\vec{X}_1^* - \vec{X}^*) \cdot \vec{t}^* > 0 \\ -|s_1| & \text{otherwise,} \end{cases}$$

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

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

$$(3.17) \quad G(s^*) = \sum_{\vec{X}_k^* \in \text{Set}_I \cup \odot} \alpha_k G_k(\vec{X}_k^*),$$

$$(3.18) \quad G'(s^*) = \sum_{\vec{X}_k^* \in \text{Set}_I \cup \odot} \beta_k G_k(\vec{X}_k^*),$$

$$(3.19) \quad G''(s^*) = \sum_{\vec{X}_k^* \in \text{Set}_I \cup \odot} \gamma_k G_k(\vec{X}_k^*),$$

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

$$(3.20) \quad \begin{bmatrix} \sum_k d_k \\ \sum_k d_k s_k \\ \sum_k d_k \frac{s_k^2}{2} \end{bmatrix} X = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \text{or,} \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{or,} \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

where d_k is the distance between \vec{X}_k^* and \vec{X}^* . 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 center weight more than 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 the matrix-vector form:

$$(3.21) \quad A\Phi + BG = F.$$

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 term 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

$$(3.22) \quad \phi_n^+ = v(s).$$

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 (3.22) is satisfied to certain 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:

$$(3.23) \quad \Phi_n^+(G(s)) = \sum_{ij} \alpha_{ij} \Phi_{ij} + C(s),$$

where $C(s)$ is a correction term at the projection. We refer the reader to the [11] for the details.

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

$$(3.24) \quad \begin{bmatrix} A & B \\ E & D \end{bmatrix} \begin{bmatrix} \Phi \\ G \end{bmatrix} = \begin{bmatrix} F \\ V \end{bmatrix}.$$

The Schur complement for G then is:

$$(3.25) \quad \begin{aligned} (D - EA^{-1}B)G &= V - EA^{-1}F \\ &\stackrel{\text{def}}{=} \bar{F}. \end{aligned}$$

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.

3.4. A Pre-conditioning Strategy. Since homogeneous Neumann condition is prescribed on the boundary $\partial\Omega_2$, the solution in the interior of the voids can differ by an arbitrary constant. If we do not impose any restriction on the constant, the solution in Ω_2 has an arbitrary constant which affects the condition number of the Schur complement. 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.

$$(3.26) \quad \phi(\vec{x}) = \frac{1}{2\pi} \int_{\partial\Omega_1} \mu_1(s) \log |\vec{x} - \vec{X}(s)| ds + \frac{1}{2\pi} \int_{\partial\Omega_2} \mu_2(s) \frac{\partial}{\partial \vec{n}} \log |\vec{x} - \vec{X}(s)| ds$$

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 are more than doubled compared with the approach described here because of 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

$$(3.27) \quad C = \int_{\partial\Omega_2} u^+(\vec{X}) ds.$$

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 be axi-symmetric, we will get the convergence in a couple of iterations.

Our numerical examples indicate that the number of iterations for solving the Schur complement is small and it 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 (3.24) and (3.25) are never formed explicitly. The algorithms is outlined below:

Outline of the algorithm to evaluate the electrical potential:

1. Initialization: Including setting up the grid and the boundary conditions on $\partial\Omega_1$; indexing the grid points as regular or irregular; getting 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. These data are affordable to store since the number of irregular grid points is usually one-dimensional lower than the total number of grid grid points.
2. Finding the right hand side of the Schur complement system. This can be done by setting $G = 0$ to get the solution of the Poisson equations. The right hand side is the residual of (3.25) with a negative sign.
3. Applying the GMRES iteration with an initial guess of G . The main process of GMRES method requires only matrix-vector multiplication. 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 to get the correction terms at each irregular grid point due to the jump in the solution. The fast Poisson solver that we employ is the modified HWSORT routine from FISH Package. The second step is to interpolate the solution of the Poisson equation to get the residual.

3.5. Numerical examples for the evaluation of Poisson equations. We provide one example with different boundary conditions to show the efficiency of the algorithm proposed above. This is the main part in simulation of electro-migration voids. We want to verify second order accuracy of the solution procedure, and more importantly, that the number of iterations is nearly independent of mesh size except a factor of $\log h$.

We construct an exact solution:

$$(3.28) \quad \begin{aligned} \phi(x, y) &= -\frac{1}{2} \log r + r^2, \quad r = \sqrt{x^2 + y^2}, \\ f(x, y) &= 4. \end{aligned}$$

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

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

Case 1: 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 3.1 shows the grid refinement analysis and other information with $a = 0.5$ and $b = 0.4$.

In Table 3.1, n is the number of grid lines in x - and y - directions; e is the error of the computed solution in the infinity norm; n_1 is the total number of irregular grid points; n_2 is the number of irregular

TABLE 3.1

The grid refinement analysis with $a = 0.5$ $b = 0.4$ and $b = 0.15$. 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.

n	a	b	e	r	n_1	n_2	k
40	0.5	0.4	$5.7116 \cdot 10^{-4}$		100	52	16
80	0.5	0.4	$1.4591 \cdot 10^{-4}$	3.9146	204	104	17
160	0.5	0.4	$3.5236 \cdot 10^{-5}$	4.1408	412	208	19
320	0.5	0.4	$8.1638 \cdot 10^{-6}$	4.3161	820	412	21

n	a	b	e	r	n_1	n_2	k
40	0.5	0.15	$4.4820 \cdot 10^{-3}$		68	36	13
80	0.5	0.15	$1.1466 \cdot 10^{-3}$	3.9089	132	68	15
160	0.5	0.15	$2.6159 \cdot 10^{-4}$	4.3832	68	136	17
320	0.5	0.15	$6.7733 \cdot 10^{-5}$	3.8621	68	268	20

grid points on one side, which is also the dimension of the Schur complement; k is the number of iteration of the GMRES method which is also the number of the fast Poisson solver called. We can observe second order convergence in the infinity 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 be fluctuated as explained in [11].

Case 2: Normal derivative condition $\partial\phi/\partial n$ is prescribed on $\partial\Omega_1$ using the exact solution. In this case, the solution is not unique and the compatibility condition has to be imposed in order to get a reasonable accurate solution. To get 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 3.2 is the result of the grid refinement analysis. We have similar results as we analyzed above.

TABLE 3.2

The grid refinement analysis for pure Neumann type boundary condition.

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

4. Evaluation of normal velocity of the surface: surface Laplacian of electrical potential and curvature. Surface Laplacian operator along a curve in two dimensions is pretty straight forward. In higher dimensions however, a nice parameterization of the hyper-surface can be very difficult to maintain in Lagrangian formulation and hence surface Laplacian operator can be problematic in numerical computation. Since we are using a framework of fixed Cartesian grid we will use a formula for surface Laplacian operator in a Cartesian coordinates. To be more specific, we are using the level set method to capture the moving interface. The interface location and geometry are embedded in the level set function. So our surface Laplacian operator has also to be expressed in terms of level set function. Let us start with a two dimensional case and then extend it to any dimensions. We denote $\vec{\tau}$ to be the tangential direction of the curve and \vec{n} to be the normal direction, and assume s is the arc length parameter. If f is a function that

is defined in a neighborhood of the curve the surface Laplacian of f is

$$\Delta_s f = \frac{d^2 f}{ds^2} = \frac{d}{ds}(\vec{\tau} \cdot \nabla f) = \kappa \vec{n} \cdot \nabla f + D^2 f(\vec{\tau}, \vec{\tau}) = \kappa \frac{\partial f}{\partial n} + \Delta f - \frac{\partial^2 f}{\partial n^2},$$

where κ is the curvature and $D^2 f$ is the Hessian of f . Therefore we have

$$\Delta f = D^2 f(\vec{\tau}, \vec{\tau}) + D^2 f(\vec{n}, \vec{n}) = \frac{\partial^2 f}{\partial \tau^2} + \frac{\partial^2 f}{\partial n^2}.$$

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

$$(4.1) \quad \Delta_s f = \kappa \frac{\partial f}{\partial e_n} + \sum_{i=1, n-1} \frac{\partial^2 f}{\partial e_i^2} = \kappa \frac{\partial f}{\partial e_n} + \Delta f - \frac{\partial^2 f}{\partial e_n^2},$$

where κ is mean curvature (sum of the principle curvatures) of the surface. In the level set formulation, say the surface is the zero level set of φ , then

$$\kappa = \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|}, \quad \vec{e}_n = \frac{\nabla \varphi}{|\nabla \varphi|}, \quad \frac{\partial f}{\partial e_n} = \frac{\nabla \varphi}{|\nabla \varphi|} \cdot \nabla f, \quad \frac{\partial^2 f}{\partial e_n^2} = D^2 f \left(\frac{\nabla \varphi}{|\nabla \varphi|}, \frac{\nabla \varphi}{|\nabla \varphi|} \right)$$

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

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

In the special case where f is constant in the normal direction, i.e. $\nabla f \cdot \vec{e}_n = 0$, we have $\frac{\partial f}{\partial e_n} = 0$. Numerically if we can make φ a distance function (through re-initialization) and extend f such that $\nabla f \cdot \nabla \varphi = 0$ (see [16]) we have

$$\Delta_s f = \Delta f$$

which is very easy to compute in Cartesian grid. In terms of accuracy we would prefer (4.1) in our numerical computation.

4.1. Evaluation of 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, oscillation will be present after second order differentiation. Such oscillation will also affect the choice of time increment 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 compute the derivative along the tangential direction to get the normal velocity of the surface due to the variation in the electric potential. There are a variety of choices as 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

$$(4.2) \quad \frac{\partial \phi}{\partial s} = -\frac{\partial \phi}{\partial x} n_y + \frac{\partial \phi}{\partial y} n_x$$

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

- Use the weighted least squares interpolation to get $\frac{\partial \phi}{\partial s}$ 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 those points which are located inside the void within the computational tube using the following formula:

$$(4.3) \quad \left(\frac{\partial \phi}{\partial s} \right)_t - \left(\nabla \frac{\partial \phi}{\partial s} \right) \cdot \frac{\nabla \varphi}{|\nabla \varphi|} \text{sign}(\varphi) = 0.$$

Here $\text{sign}(\varphi)$ 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 [12, 16].

- Finally we use the standard central scheme to compute the part of the surface velocity due to the electric potential

$$(4.4) \quad \frac{\partial}{\partial s} \left(\frac{\partial \phi}{\partial s} \right) = -\frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial s} \right) n_y + \frac{\partial}{\partial y} \left(\frac{\partial \phi}{\partial s} \right) n_x.$$

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

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

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

4.2. Evaluation of surface velocity due to the curvature. In two dimensions it is relatively straightforward to evaluate the surface velocity due to variation 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 (4.2) and (4.4). Since the computation process involves fourth 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 larger width for the computational tube; the time step size also needs to be cut to keep stability. Furthermore, with wider support and fourth order derivatives, special care has to be exerted in accounting for the boundary conditions of the voids. Numerically, 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 well in our numerical experiment. In fact, it can be shown that, if a function is sufficiently smooth, the computed fourth derivatives using the standard central difference scheme is second order accurate. In [1], a two dimensional model involves only surface diffusion term is considered. There is no topological changes in the geometry.

After we computed the normal velocity of the boundaries of the voids, we use the weighted ENO (essentially non-oscillatory) scheme to update the level set function and do the re-initialization process. WENO scheme is more stable and accurate and the errors are smoother than ENO scheme. (see [7]).

4.3. Choosing adaptive time step steps. The CFL condition for the surface diffusion is

$$(4.5) \quad \Delta t_1 \leq h^4 / C_2.$$

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

$$(4.6) \quad \Delta t_2 \leq h / U_{max},$$

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

$$(4.7) \quad \Delta t_3 \leq h^2 / C_1.$$

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 4.1. 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 magnitude at these points, leading to

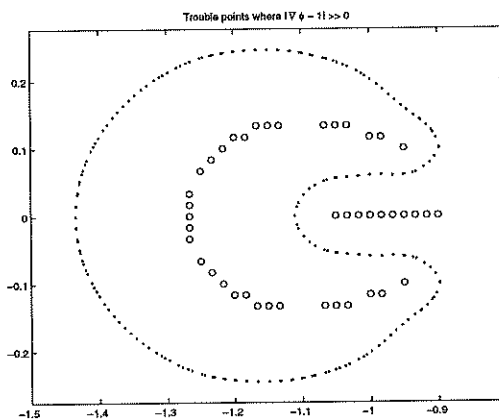


FIG. 4.1. Grid points marked as little 'o's, indicate those points that are either on or close to the kink of the signed distance function, the solid dots are projections of irregular grid points on the interface.

very small time step size and sometimes instabilities. Another mathematical concern is, as we pointed out in the introduction section, that if each level set of the level set function moves according its own geometry and motion law independently they will run into each other due to the lack of maximum principle in this problem. Our numerical solution to this problem is to use a cut-off function $\omega(x)$,

$$(4.8) \quad \omega(x) = \begin{cases} 1 & \text{if } x \leq 0 \\ (x-1)^4 (20x^3 + 10x^2 + 4x + 1) & \text{if } 0 < x \leq 1 \\ 0 & \text{if } 1 < x. \end{cases}$$

The new velocity is defined as:

$$(4.9) \quad U_n^{new} = \omega\left(\frac{|\varphi| - \varphi_0}{\delta}\right) U_n.$$

In a very thin tube containing the interface, $|\varphi| \leq \varphi_0$, the normal velocity is unchanged. It is gradually leveled down to zero to take care of those troubled points without significant effect on the accuracy. We choose $\delta = 3h$ and φ_0 is between $2h \sim 4h$ in our numerical tests. Notice that the cut-off function used here has up to third continuous derivatives, in an effort to reduce the oscillations introduced by the cut-off near the boundary. Numerically speaking, level set function provides a body fitted parallel frames (level sets) near the interface which can capture both the geometry and motion of the interface on a Eulerian framework. Physically and mathematically what really matters is just the motion of level sets very close to

the interface. Due to numerical CFL condition and right scaling of level set function (distance function) we are guaranteed to be trouble free. The reason why we use cut-off function instead of extending the normal velocity is area conservation as we mentioned before. Our approach effectively eliminates the possibility of having too large normal velocities near or on kink.

REMARK 4.1. *When topological changes take place, we can not cut off the kink anymore. The calculation may loose 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 Experiment. We have performed a series of numerical experiments on the interaction of voids during 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.

In the first two experiments, we try to reproduce the results from [19]. The computational domain is the rectangle $-2.5 \leq x \leq 2.5$, $-0.5 \leq y \leq 0.5$. 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 [19]. The computational grid is 300×60 , or $h = 1/60$. Fig 5.1 show 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%.

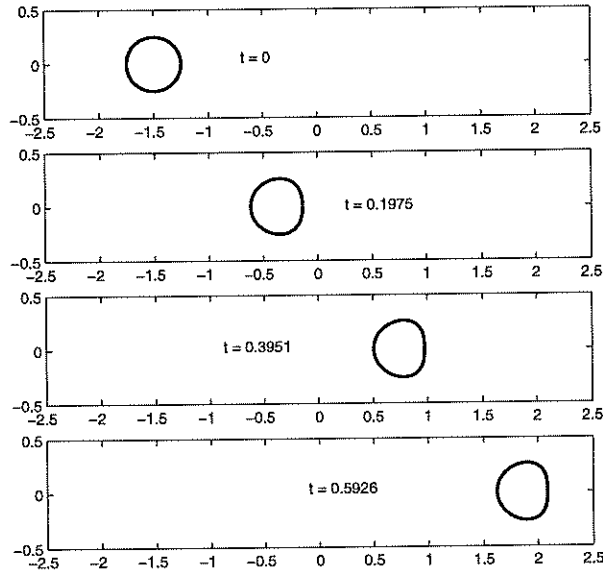


FIG. 5.1. *A void evolution with large chemical potential. the motion is relatively stable*

In the second case, the coefficient of the chemical potential, $C_2 = 1.875 \times 2.08333 \times 10^{-4}$, is small compared to that of electrical potential, $C_1 = 1.875$, and the motion is less stable compared to 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 that was obtained in [19]. The evolution process seems to agree the result in that paper up to some time. Note that, we used Neumann boundary conditions at the left and right edges of the rectangle while they used Dirichlet boundary conditions. Usually it is harder to solve Poisson equations with Neumann boundary conditions than that of Dirichlet type.

As discussed in [19], 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 reproduces the corresponding examples given in [19], 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 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 becomes dominating and causes drastic deviation from the circular void shape. Fig. 5.3 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 similar pattern in the evolution process. The loss in the area is due to the poor resolution and the large curvatures right after topological change took place.

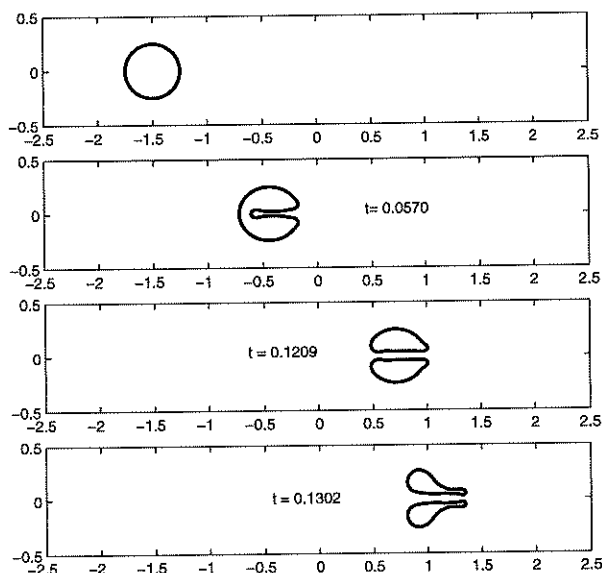


FIG. 5.2. A void evolution with small chemical potential. The motion is less stable.

The algorithm developed here can be used to study interaction of multiple voids in an electro-migration line. Fig. 5.4 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 towards more unstable pattern with the creation of a number of small voids from the bigger 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 happens after small voids are produced. The large curvature of the small voids compensates the small chemical potential coefficient C_2 relative to the electro-migration coefficient C_1 . This is not unexpected because small C_2/C_1 indicates shape instability of large voids which tend to break up into smaller voids in order to seek a balance between driving force associated with the surface energy and that associate with the electro-migration. The same argument would indicate that conditions of small C_2/C_1 tend to prevent merge of voids into larger ones, in consistency with our observations. In this simulation, we see also that small voids move faster than large ones.

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

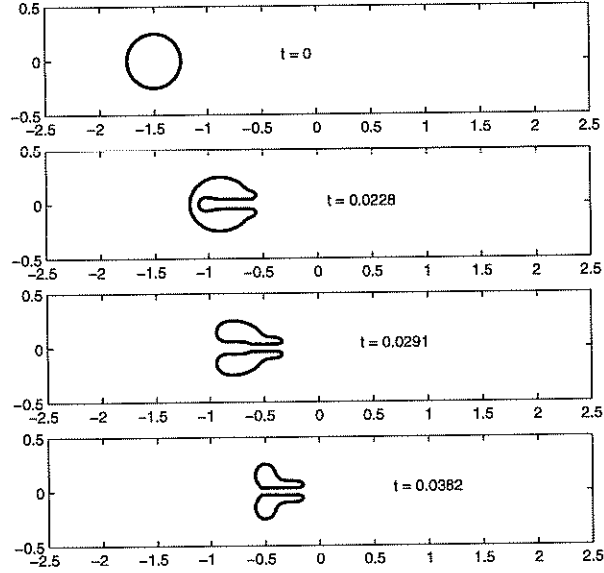


FIG. 5.3. A void evolution with very small chemical potential on a coarse grid.

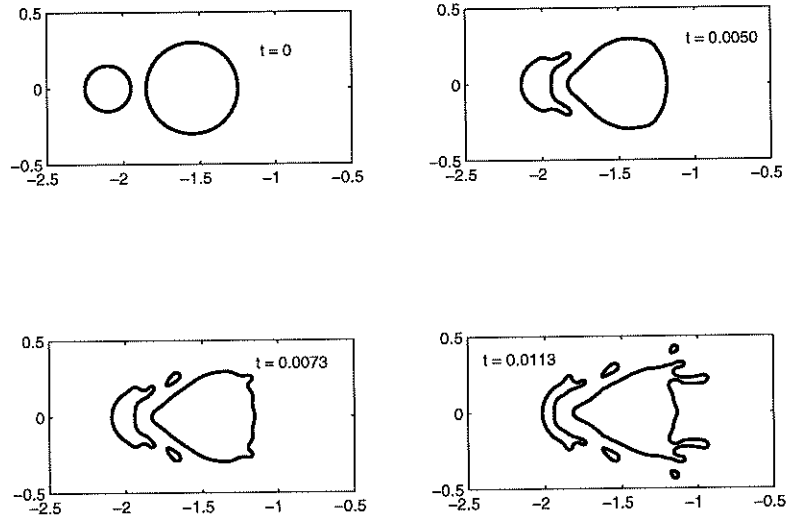


FIG. 5.4. An evolution process of two voids with small chemical potentials.

surface tension force is sufficiently large, two or more smaller voids may merge to form larger voids, as is demonstrated in Fig. 5.5 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 axis and b is minor. 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%. It is much worse compared with previous cases. We believe there are two factors which affect the loss of the area: The major cause is the large h and lack of resolution in the simulation. For the problems involving fourth order derivatives, we find that a reasonable revolution is necessary to preserve the area. The second fact is the singularity before or after the topological changes,

the merging, where some parts of level set have very large curvatures.

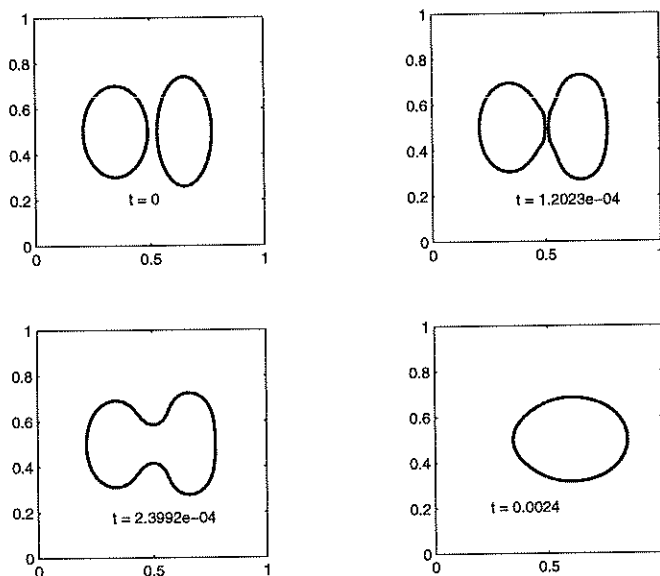


FIG. 5.5. An evolution process of two voids with large chemical potentials.

Fig. 5.6 shows the interaction of two voids under relatively large C_2/C_1 ratio. The first one is an ellipse centered at $(-2.1, 0)$ with major axis 0.3 and 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. Fig. 5.7 shows the evolution process of an ellipse centered at $(-1.8, 0)$ with major axes 0.4 and minor axes 0.2. The coefficients are $C_1 = 1.875$ and $C_2 = 0.3.90625 \cdot 10^{-3}$. 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 void with a relatively large curvature at the frontal tip of the void. This sharp tip moves rapidly under electromigration forces and drags the rest of the void along. Such a worm-like motion of void is another delicate balance between surface tension and electromigration forces.

5.1. Conclusion. In this paper, we describe a numerical method for simulation of moving interface problems which arise in electro-migration voiding in integrated circuits. In comparison with a conventional finite element or boundary element based numerical methods with adaptive meshing, our method has the advantage of employing a fixed cartesian grid without the need of adaptive remeshing at each time step. A fast iterative method using GMRES iteration has been adopted to solve Laplace equations associated with electric potential on the exterior of voids. The evolution of void boundaries are described by a two dimensional level set function which are 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 some test cases in the literature using finite element based simulation method. We have also performed a series of numerical experiments to elucidate the mechanisms by which electromigration voids interact. It appears

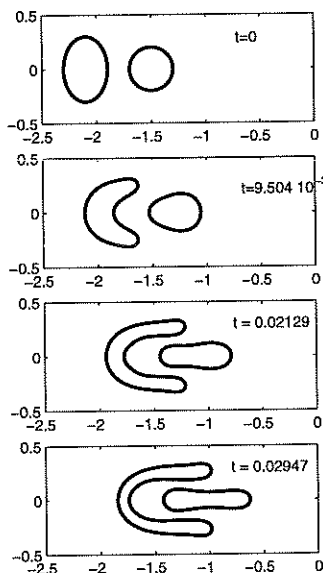


FIG. 5.6. An evolution process of two voids with large chemical potentials.

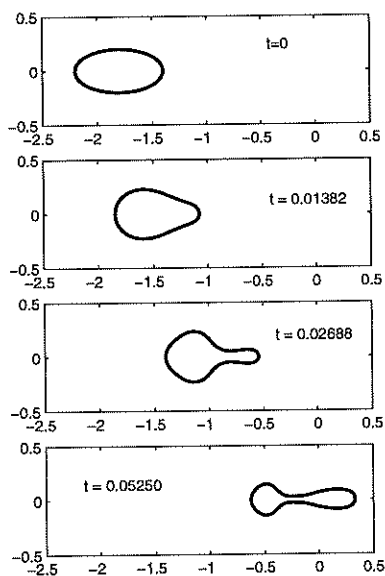


FIG. 5.7. An evolution process of two voids with large chemical potentials.

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 electromigration 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 size of the voids are on the order of the conducting line width. The method we propose does not require sophisticated meshing technique and has 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 saving on meshing is expected to be even more significant.

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*, 85:375–404, 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. *European J. Appl. Math.*, 7:287–301, 1996.
- [5] J. W. Cahn and J. E. Taylor. Surface motion by surface diffusion. *Acta Metallica Materiala*, 42:1045–1063, 1994.
- [6] C. M. Elliott and S. Zheng. The Cahn-Hilliard equation. *Arch. Rat. Mech. Anal.*, 96:339–357, 1996.
- [7] G. Jiang and D. Peng. Weighted ENO schemes for hamilton-jacobi equations. UCLA CAM Report 97-29, 1998.
- [8] O. Kraft and E. Arzt. Electromigration mechanisms in conductor lines: void shape changes and slit-like failure. *Acta Mater*, 45:1599–1611, 1997.
- [9] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31:1019–1044, 1994.
- [10] Z. Li. *The Immersed Interface Method — A Numerical Approach for Partial Differential Equations with Interfaces*. PhD thesis, University of Washington, 1994.
- [11] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, 35:230–254, 1998.
- [12] Z. Li and B. Soni. Fast and accurate numerical approaches for stefan problems and crystal growth. preprint, 1998.
- [13] W. W. Mullins. Mass transport at interface in single component systems. *Metall. Trans. A*, 26:1917–1929, 1995.
- [14] F. A. Nichols and W. W. Mullins. Surface (interface) and volume-diffusion contributions to morphological changes driven by capallarity. *Trans. Metall. Soc. AIME*, 233:1840–1847, 1965.
- [15] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.
- [16] 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.
- [17] R. F. Sekerka and T. F. Marinas. Dynamics of morphological change during solid-solid transformations. In H.I. Aaronson et al., editor, *Solid-solid Phase Transformations*, pages 67–86. The Metallurgical Society of AIME, 1982.
- [18] J. E. Taylor and J. W. Cahn. Linking anisotropic sharp and diffuse surface motion laws via gradient flows. *J. of Stat. Phys.*, 77:183–197, 1994.
- [19] 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*, 45:1473–1493, 1997.
- [20] Z. Yang. *A Cartesian grid method for elliptic boundary value problems in irregular regions*. PhD thesis, University of Washington, 1996.
- [21] H. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127:179–195, 1996.