Commun. Comput. Phys.
doi: 10.4208/cicp.230510.240910a

Vol. **10**, No. 1, pp. 32-56
July 2011

An Interface-Fitted Finite Element Level Set Method with Application to Solidification and Solvation

Bo Li[1,*] and John Shopple[2]

[1] *Department of Mathematics and the NSF Center for Theoretical Biological Physics, University of California, San Diego, 9500 Gilman Drive, Mail code: 0112. La Jolla, CA 92093-0112, USA.*
[2] *Department of Mathematics, University of California, San Diego, 9500 Gilman Drive, Mail code: 0112. La Jolla, CA 92093-0112, USA.*

Received 23 May 2009; Accepted (in revised version) 24 September 2009

Available online 7 February 2011

Abstract. A new finite element level set method is developed to simulate the interface motion. The normal velocity of the moving interface can depend on both the local geometry, such as the curvature, and the external force such as that due to the flux from both sides of the interface of a material whose concentration is governed by a diffusion equation. The key idea of the method is to use an interface-fitted finite element mesh. Such an approximation of the interface allows an accurate calculation of the solution to the diffusion equation. The interface-fitted mesh is constructed from a base mesh, a uniform finite element mesh, at each time step to explicitly locate the interface and separate regions defined by the interface. Several new level set techniques are developed in the framework of finite element methods. These include a simple finite element method for approximating the curvature, a new method for the extension of normal velocity, and a finite element least-squares method for the reinitialization of level set functions. Application of the method to the classical solidification problem captures the dendrites. The method is also applied to the molecular solvation to determine optimal solute-solvent interfaces of solvation systems.

AMS subject classifications: 65M

Key words: Level set method, finite element, interface-fitted mesh, curvature approximation, velocity extension, reinitialization, solidification, dendrites, molecular solvation, variational implicit-solvent models.

1 Introduction

Consider a moving interface $\Gamma = \Gamma(t)$ that separates two regions $\Omega_- = \Omega_-(t)$ and $\Omega_+ = \Omega_+(t)$, respectively, where t denotes time. The interface Γ can have multiple connected

*Corresponding author. *Email addresses:* bli@math.ucsd.edu (B. Li), jshopple@math.ucsd.edu (J. Shopple)

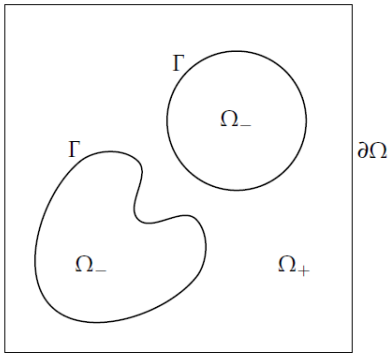

Figure 1: The geometry of interface motion.

components. We assume all Γ, Ω_-, and Ω_+ are inside a large computational domain Ω whose boundary is denoted by $\partial\Omega$, cf. Fig. 1.

The motion of the interface $\Gamma(t)$ is completely determined by its normal velocity, $v_n = v_n(x,t)$, at each point $x \in \Gamma(t)$ at each time t. For many physical systems, such a normal velocity consists of two parts. One is the local geometry, such as the curvature or mean curvature. The other is the external force such as flux of certain diffusive material. The concentration of such material is governed by a one-sided or two-sided diffusion equation, together with suitable boundary conditions on the moving interface $\Gamma(t)$. A typical system of a moving interface can thus be described by the following equations:

$$v_n = F_{geo}(H,K) + F_{ext}(u,\nabla u), \qquad \text{on } \Gamma(t), \tag{1.1a}$$
$$A_-(u) = f_-, \qquad \text{in } \Omega_-(t), \tag{1.1b}$$
$$A_+(u) = f_+, \qquad \text{in } \Omega_+(t), \tag{1.1c}$$
$$\text{boundary conditions for } u, \qquad \text{on } \Gamma(t). \tag{1.1d}$$

Here, $F_{geo}(H,K)$ is a function of the mean curvature H and Gaussian curvature K, $F_{ext}(u,\nabla u)$ is a function of a field variable u and its gradient ∇u, and $A_-(u)$ and $A_+(u)$ are partial differential operators of u, and f_- and f_+ are known functions on Ω_- and Ω_+, respectively. We shall call (1.1b) and (1.1c) the field equations for u. We also call (1.1d) the interface conditions.

There are many examples of moving interfaces that separate different media that are deforming or flowing. For instance, an ice-water boundary moves during the change of temperature, an interface separates two fluids with different viscosities and densities in a two-phase flow, phase boundaries in solid-solid phase transformations such as precipitates and martensite interfaces, domain boundaries that separate different parts of material such as grain boundaries in polycrystals, and the solute-solvent interface that separates a molecule from its surrounding solvent in a solvation system.

We use the level set method to numerically track the interface motion [12–14]. This method is perhaps one of the most popular methods due to its easy handling of topological changes, such as interface merging and breaking up. We denote by $\phi = \phi(x,t)$ a level

set function of $\Gamma(t)$, i.e., $\Gamma(t)=\{x:\phi(x,t)=0\}$. We assume that $\phi(x,t)<0$, for $x\in\Omega_-$, and $\phi(x,t)>0$, for $x\in\Omega_+$, for all t. Once a level set function $\phi(x,t)$ is determined, we can locate the interface $\Gamma(t)$ by finding the zero level set of $\phi(x,t)$. The level set function is determined by the following level set equation:

$$\frac{\partial\phi}{\partial t}+v_n|\nabla\phi|=0, \tag{1.2}$$

where v_n is the normal velocity given in (1.1a). The normal velocity needs to be extended away from the interface so that the level set equation (1.2) can be solved in the computational box Ω with some boundary conditions for $\phi(x,t)$ on the boundary $\partial\Omega$.

In this work, we develop a finite element level set method with interface-fitted meshes for tracking a moving interface that is described by (1.1a)-(1.1d) in the two-dimensional setting. We use a uniform mesh as our base mesh, cf. [10]. In each time step, we use the base mesh to construct an interface-fitted mesh. This means the restriction of the mesh onto Ω_- or Ω_+ is a finite element mesh covering Ω_- or Ω_+, respectively. This way, we can more accurately approximate the interface condition (1.1d) and hence the solution of the partial differential equations (1.1b) and (1.1c).

We also develop some new level set techniques. First, we extend the normal velocity by solving the Laplace equation on both sides of the equation, using the velocity on the interface as the Dirichlet boundary data. We will present some examples to show how this method works. Second, we reinitialize the level set function by solving a Poisson's equation. The reinitialization is necessary to prevent $|\nabla\phi(x,t)|$ from being too large or too small which can cause instabilities and large errors in locating the zero level set from $\phi(x,t)$. We keep the level set function smooth across the interface by imposing a gradient jump condition, then solve an overdetermined system by the least-squares method. This method produces a new level set function that is smooth over the domain and across the interface. Being smooth is important because it increases the accuracy of the curvature calculation. We will give some examples to demonstrate how this reinitialization works. Finally, we introduce a simple finite element method to approximate the curvature with an $\mathcal{O}(h^2)$ accuracy, where h is the mesh size of the base mesh.

Our algorithm consists of the following main steps:

1. Define the base mesh. Initialize the level set function. Input all parameters;
2. Locate the interface and construct an interface-fitted mesh;
3. Solve field equations;
4. Calculate normal velocity;
5. Extend normal velocity;
6. Unrefine the mesh;
7. Solve the level set equation on the base mesh and move the interface;
8. Go to Step 2.

We apply our method to solve the Stefan problem that models the dendritic solidification of a frozen seed placed into a supercooled liquid. The cases of zero, nonzero isotropic, and anisotropic surface tension are all considered. Our results recover those obtained in the previous study using finite difference schemes [2]. We also apply our method to determine the optimal molecular surfaces using the recently developed variational implicit-solvent model [3, 4, 7, 8]. The central quantity in this model is a free energy functional of all possible surfaces that enclose a set of points representing atoms of molecules (solutes) under consideration. The free energy includes the surface energy and the solute-solvent van der Waals type interaction energy. We apply our level set method to relax an initial solute-solvent interface. We present numerical examples to demonstrate that our method converges. These examples also show that the variational implicit-solvent model captures some key features of molecular solvation.

Finite element based level set methods have been applied to interface motion in fluid dynamics [6, 11, 15, 16]. In these earlier works, the authors developed special finite element methods for solving fluid flow equations that often have convection terms, and combined them with the level set method to numerically track moving interfaces. Special techniques of numerical integration are also designed in [15, 16]. In [6], a discontinuous Galerkin finite element method and a stabilized continuous finite element method are developed for fluid flow problems to prevent a significant loss of mass which occurs often in a simple level set implementation. However, the locally, interface fitting meshing does not seem to be developed systematically in these works. The finite element level set method proposed in [9] for dendritic growth of crystals is along the line of finite element methods for motion by mean curvature, cf. [5]. We notice that different techniques of interface-fitting meshing are proposed in [10, 17, 18].

The rest of this paper is organized as follows: in Section 2, we describe the construction of interface-fitted meshes; in Section 3 and Section 4, we present our new methods for the extension of normal velocity and the reinitialization of the level set function, respectively; in Sections 5, we describe our finite element method to approximate the curvature and prove that this approximation is second-order accurate; Section 6 and Section 7 are the application of our method to solidification and solvation, respectively; finally, in Section 8, we discuss our results and draw conclusions.

2 Interface-fitted finite element mesh

Let us fix our computational domain to be the rectangle $\Omega=[a_1,a_2]\times[b_1,b_2]$, in the x_1x_2 plane with $a_i,b_i\in\mathbb{R}$ and $a_i<b_i$ $(i=1,2)$. Let M_1 and M_2 be two positive integers and $h_i=(b_i-a_i)/M_i$ $(i=1,2)$. Let $x_{ij}=a_i+jh_i$ $(i=1,2;\ j=0,1,\cdots,M_i)$. Let $h=\max(h_1,h_2)$. We denote by $\mathcal{T}_h$ the uniform mesh of triangles with all the vertices (x_{1i},x_{2j}) $(i=0,\cdots,M_1;\ j=0,\cdots,M_2)$ and all edges parallel to the lines $x_1=0$, $x_2=0$, and $h_2x_1=h_1x_2$. This is our base mesh.

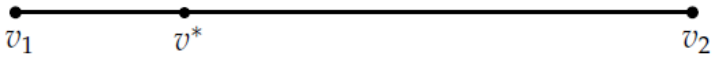

Figure 2: A refined edge.

We denote by S_h the corresponding linear finite element space:

$$S_h = \{\phi_h \in C(\overline{\Omega}) : \phi_h|_K \in P_1(K), \forall K \in \mathcal{T}_h\},$$

where $P_1(K)$ denotes the space of polynomials on K of degrees less than or equal to 1. We also denote by $N_{ij} \in S_h$ the linear finite element nodal basis at (x_{1i}, x_{2j}), i.e., $N_{ij} = 1$ at (x_{1i}, x_{2j}), and $N_{ij} = 0$ at all other vertices. Denote also $x = (x_1, x_2)$. Then any function $\phi_h \in S_h$ can be uniquely represented as

$$\phi_h(x) = \sum_{j=0}^{M_2} \sum_{i=0}^{M_1} \phi_h(x_{1i}, x_{2j}) N_{ij}(x), \quad \forall x \in \Omega.$$

We now construct an interface-fitted finite element mesh $\mathcal{T}_{h,\Gamma}$ using the level set function ϕ of the interface Γ, cf. Fig. 1. For convenience, we drop the time t for a moment. Fix an arbitrary edge with the two vertices v_1 and v_2 as in Fig. 2. We examine the value of the level set function ϕ at each vertex,

$$\phi_1 = \phi(v_1), \quad \text{and} \quad \phi_2 = \phi(v_2).$$

If the value is positive at one end and negative at the other, then the interface crosses that particular edge. In this case, we locate the point v^* along that edge where the level set function is zero and add a vertex there, splitting the edge into two edges. This point is given by the equation

$$v^* = \frac{\phi(v_2)v_1 - \phi(v_1)v_2}{\phi(v_2) - \phi(v_1)}.$$

We then call this edge *refined*.

Once all of these new vertices are added, we examine each triangle. For each triangle, there can be zero, one or two edges that were refined. If only one edge was refined, then the value of the level set function must be zero on the opposite vertex. In this case a new edge is added joining the new vertex with the opposite vertex, splitting the triangle into two triangles, cf. upper-left triangle in Fig. 3. The more common situation is when two edges of a triangle were refined. In this case, this splits the triangle into a triangle and a quadrilateral. We then split this quadrilateral into two triangles by adding an edge along one of the diagonals, cf. lower-right triangle in Fig. 3. This splits the original triangle into three triangles.

The mesh refinement process is illustrated in Fig. 4. We denote by $\mathcal{T}_{h,\Gamma}$ the interface-fitted mesh that we just constructed. We also denote by $S_{h,\Gamma}$ the corresponding linear finite element space.

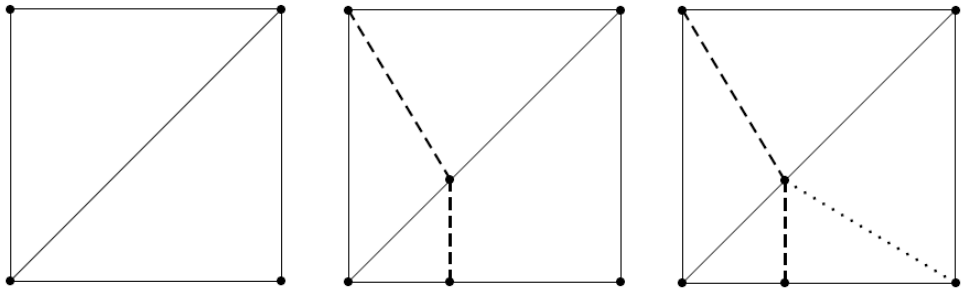

Figure 3: Example of two types of interface-fitted mesh refinement. (a) Original triangles in base mesh. (b) Edge (dashed) added along interface. (c) Edges (dotted) added to split lower triangle into three triangles.

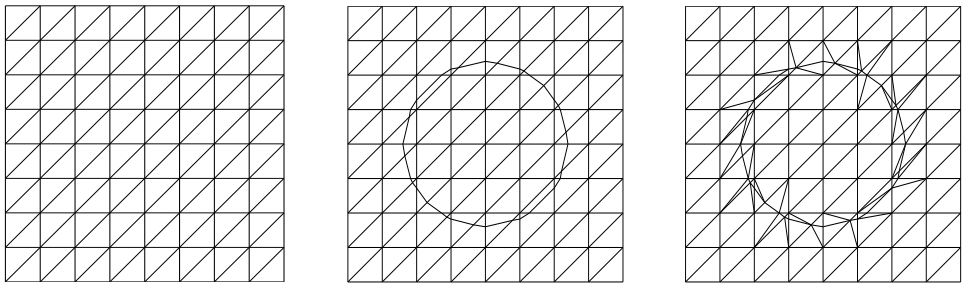

Figure 4: Left: The base mesh $\mathcal{T}_h$. Middle: The base mesh $\mathcal{T}_h$ with an interface Γ. Right: An interface-fitted mesh $\mathcal{T}_{h,\Gamma}$.

3 Extension of normal velocity

To have an efficient and smooth extension of the normal velocity, we solve the following system of equations

$$\begin{aligned}
&\Delta u = 0, && \text{in } \Omega_-,\\
&\Delta u = 0, && \text{in } \Omega_+,\\
&u = v_n, && \text{on } \Gamma,\\
&\frac{\partial u}{\partial n} = 0, && \text{on } \partial\Omega,
\end{aligned}$$

where v_n is the normal velocity on the interface Γ that needs to be extended and u is the extended velocity.

The extended normal velocity may not have a continuous derivative along Γ. Near the points on the interface with the highest velocity, the extension will have lower velocity values on either side. This is potentially a problem, since after unrefinement of the interface-fitted mesh, the explicit values of the normal velocity on the interface are lost, and the interpolated normal velocity values on the interface will be lower than they should be. See Fig. 5 for an illustration of this point in one dimension. When we unrefine

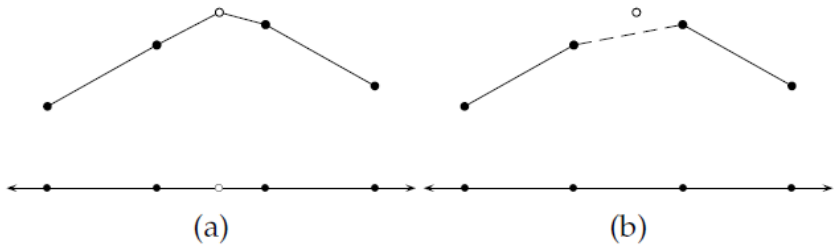

Figure 5: Illustration of how nonsmooth velocity extension can create errors. (a) Velocity value on the interface vertex (circle) is extended away to the base mesh vertices (solid dots) to create a continuous function. (b) When the interface vertex is removed and only base mesh vertices remain, the interpolated (dashed line) interface value is lower than it should be.

the interface-fitted mesh, we lose the values of the velocity at the nodes on the interface, so the extended normal velocity should capture the correct velocity of the interface implicitly. That is, the representation of the velocity on the unrefined mesh should be such that at the location of the interface it should still equal what we want the velocity to be on the interface. This can be accomplished by requiring the extension to be differentiable. Therefore, we extend the normal velocity by solving the following system:

$$\Delta u = 0, \qquad \text{in } \Omega_-, \tag{3.1a}$$

$$\Delta u = 0, \qquad \text{in } \Omega_+, \tag{3.1b}$$

$$u = v_n, \qquad \text{on } \Gamma, \tag{3.1c}$$

$$\left[\!\left[\frac{\partial u}{\partial n}\right]\!\right]_\Gamma = 0, \tag{3.1d}$$

$$\frac{\partial u}{\partial n} = 0, \qquad \text{on } \partial\Omega, \tag{3.1e}$$

where v_n is the normal velocity on the interface Γ that needs to be extended, u is the extended velocity, and $[\![z]\!]_\Gamma = z|_{\Omega_+} - z|_{\Omega_-}$ is the jump of a function z across the interface Γ. The system (3.1a)-(3.1e) is overdetermined. We shall explain at the end of this section how to find a least-squares solution of this system.

Unrefining the mesh is simply removing the vertices and edges that were added at the refinement step. In practice this just means reverting back to the original mesh. We do not need to explicitly remove each edge and vertex, so it does not require the computational time that refinement does. The quantities that were defined on the refined mesh retain the defining values at the vertices on the unrefined mesh. The values on the vertices that were on Γ are simply lost. This is not a problem for the extended normal velocity if we defined it to be smooth across Γ.

We now give details how we solve the overdetermined system (3.1a)-(3.1e) using a least-squares method. We express our finite element approximation $u_h \in S_h$ as

$$u_h(x) = \sum_{j=0}^{M_2}\sum_{i=0}^{M_1} u_{ij} N_{ij}(x), \quad \forall x \in \Omega, \tag{3.2}$$

where N_{ij} is the piecewise linear finite element basis function at the node (x_{1i},x_{2j}) and the coefficient u_{ij} approximates u at the corresponding node.

For convenience, we denote by I_- the set of (i,j) such that (x_{1i},x_{2j}) is a finite element node in Ω_- and by I_+ the set of (i,j) such that (x_{1i},x_{2j}) is a finite element node in $\Omega_+\cup\partial\Omega$. We also denote by I_Γ the set of (i,j) such that (x_{1i},x_{2j}) is a finite element node on Γ. We impose the condition (3.1c) by setting

$$u_{ij}=v_n(x_{1i},x_{2j}),\quad \text{if } (i,j)\in I_\Gamma.$$

Therefore only those values u_{ij} with $(i,j)\in I_-\cup I_+$ are to be determined.

The finite element approximation of (3.1a) and (3.1c) leads to the non-singular system of linear equations for the unknowns u_{ij} with $(i,j)\in I_-$

$$\sum_{(i,j)\in I_-} u_{ij}\int_{\Omega_-}\nabla N_{ij}\cdot\nabla N_{kl}\,dx=\sum_{(i,j)\in I_\Gamma} v_n(x_{1i},x_{2j})\int_{\Omega_-}\nabla N_{ij}\cdot\nabla N_{kl}\,dx,\quad \forall(k,l)\in I_-. \tag{3.3}$$

Similarly, we obtain by (3.1b), (3.1c), and (3.1e) the system of linear equations for u_{ij} with $(i,j)\in I_+$

$$\sum_{(i,j)\in I_+} u_{ij}\int_{\Omega_+}\nabla N_{ij}\cdot\nabla N_{kl}\,dx=\sum_{(i,j)\in I_\Gamma} v_n(x_{1i},x_{2j})\int_{\Omega_+}\nabla N_{ij}\cdot\nabla N_{kl}\,dx,\quad \forall(k,l)\in I_+. \tag{3.4}$$

To discretize the interface condition (3.1d), let us note that Γ consists of edges of finite elements. Denote by E_Γ the set of all such edges. Fix an edge $\gamma\in E_\Gamma$. This edge is shared by two triangle elements $T_-\subset\overline{\Omega_-}$ and $T_+\subset\overline{\Omega_+}$, respectively. Let $x_\gamma^-\in\Omega_-$ and $x_\gamma^+\in\Omega_+$ be the non-interface vertices of T_- and T_+, respectively. Denote by N_γ^- and N_γ^+ the local linear finite element shape functions on T_γ^- and T_γ^+ associated with x_γ^- and x_γ^+, respectively. The unit normal at γ is given by $\nabla N_\gamma^-/|\nabla N_\gamma^-|$ or $\nabla N_\gamma^+/|\nabla N_\gamma^+|$. Notice that these two unit vectors have opposite directions. The condition (3.1d) is then discretized to be

$$\nabla u_h|_{T_\gamma^-}\cdot\frac{\nabla N_\gamma^-}{|\nabla N_\gamma^-|}+\nabla u_h|_{T_\gamma^+}\cdot\frac{\nabla N_\gamma^+}{|\nabla N_\gamma^+|}=0,\quad \forall\gamma\in E_\Gamma. \tag{3.5}$$

Now all the linear equations in (3.3)-(3.5) form an overdetermined system of linear equations of the form $AU=F$, where A is a matrix with more rows than columns, U is the vector of all the unknowns u_{ij}, and F is a known vector. The least-squares solution of this system is the unique solution of the normal equation

$$A^TAU=A^TF,$$

and can be obtained by a direct method. The solution vector U and the expression (3.2) give us the desired approximate solution to the system (3.1a)-(3.1e).

4 Reinitialization

We propose to reinitialize the level set function by solving the following problem:

$$\begin{aligned}
&\Delta\phi = G, && \text{in } \Omega_-,\\
&\Delta\phi = 0, && \text{in } \Omega_+,\\
&\phi = 1, && \text{on } \partial\Omega,\\
&\phi = 0, && \text{on } \Gamma,
\end{aligned}$$

where $G>0$ is a constant and is often chosen to be in between 1 and 10. This problem consists in fact of two separate boundary-value problems on Ω_- and Ω_+, respectively. By the Maximum Principle, the reinitialized level set function ϕ, which is the solution to the above system, satisfies the designed property that

$$\phi<0, \quad \text{in } \Omega_- \quad \text{and} \quad \phi>0, \quad \text{in } \Omega_+.$$

Since we reinitialize on an interface-fitted mesh $\mathcal{T}_{h,\Gamma}$, the interface does not move. That is, we can set the values of ϕ on Γ to zero. But we need to consider the location of the interface (zero level set) after we unrefine the mesh. If the new level set function is not smooth over Γ, then unrefinement would implicitly move the zero level set, and consequently, the interface. To solve this problem, we can require that the jump in the gradient of ϕ across the interface is zero. Therefore, we solve the following system:

$$\begin{aligned}
&\Delta\phi = G, && \text{in } \Omega_-,\\
&\Delta\phi = 0, && \text{in } \Omega_+,\\
&\phi = 1, && \text{on } \partial\Omega,\\
&\phi = 0, && \text{on } \Gamma,\\
&\left[\!\left[\frac{\partial\phi}{\partial n}\right]\!\right]_\Gamma = 0,
\end{aligned}$$

where $G\geq 0$ can be set to 0. This is an overdetermined system and is solved by a least-squares method similar to that for the system (3.1a)-(3.1e).

Fig. 6 shows reinitialization without the gradient jump condition. Fig. 6, left, shows the level set function after reinitialization. It is shown from the side view to highlight the jump in the gradient. Fig. 6, right, shows the interface location before reinitialization and the location after reinitialization and unrefinement. After unrefining the mesh, the location of Γ moves since Γ is then interpolated from the values of ϕ on the base mesh. Fig. 7 shows reinitialization with the gradient jump condition. Fig. 7, left, shows the level set function, from the side view, after reinitialization. There is no jump in the gradient as there was in Fig. 6, left. Fig. 7, right, shows that the interface stays in the same location after mesh unrefinement.

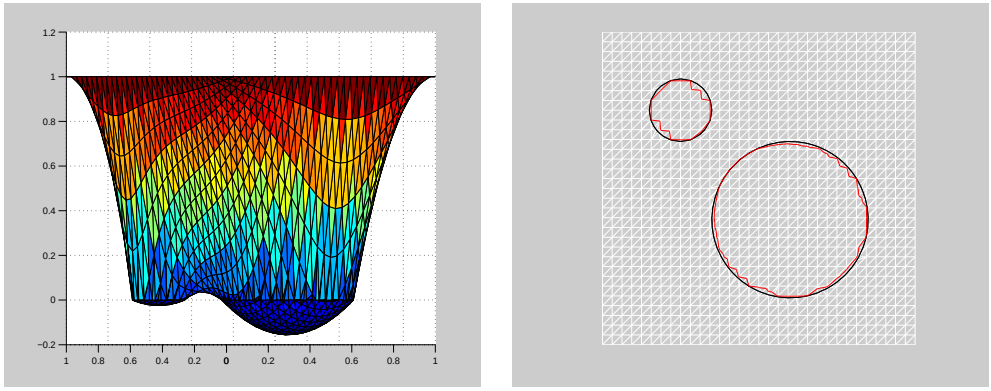

Figure 6: Example of reinitialization without gradient jump condition. Left: Level set function (side view) shown after reinitialization. Right: Interface movement during reinitialization. Interface is a circle before but becomes ragged after unrefinement.

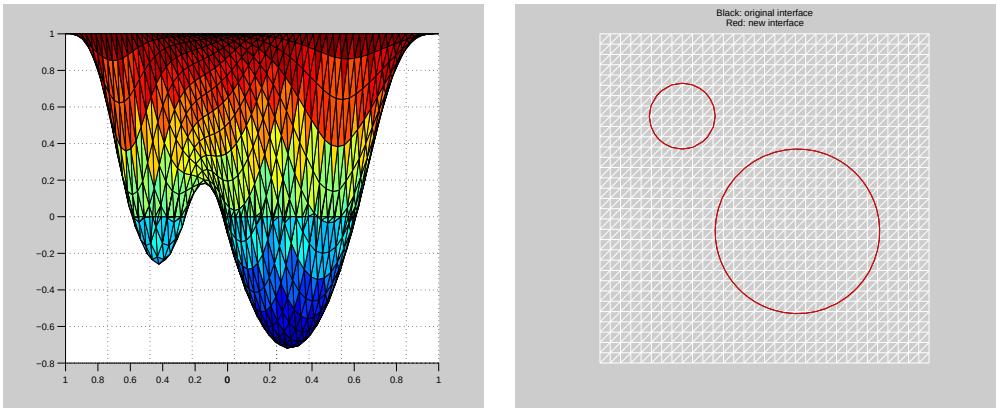

Figure 7: Example of reinitialization with gradient jump condition. Left: Level set function shown after reinitialization. Right: Interface movement during reinitialization with gradient jump condition. Interface remains a circle after unrefinement.

5 Curvature approximation

The (mean) curvature of any level set (not necessary the zero level set) of the level set function ϕ is given by

$$\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}, \tag{5.1}$$

if the level set function is smooth (e.g., $\phi \in C^2$) and the gradient $\nabla \phi$ is nonzero. To calculate this curvature, we use a local averaging method on the base mesh. We prove analytically and demonstrate numerically that this method results in $\mathcal{O}(h^2)$ accuracy for the practical case $h = h_1 = h_2$ which we assume in this section.

Let ω be an arbitrary patch of triangular elements in the base mesh that share the common vertex v_0, cf. Fig. 8. For each vertex v_i of the elements in the patch $(i = 0, \cdots, 6)$,

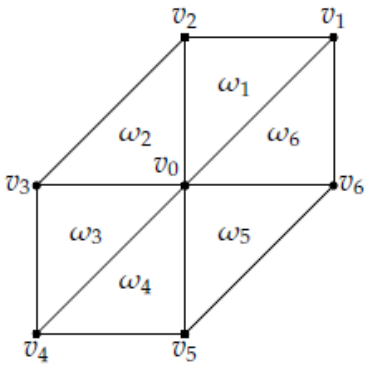

Figure 8: A patch of triangles around a vertex v_0 where $\omega=\cup_{i=1}^{6}\omega_i$.

let N_i be the corresponding piecewise linear finite element basis function: $N_i=1$ at vertex v_i and $N_i=0$ on other vertices. Since N_0 vanishes on the boundary of the patch ω, we have by (5.1) that

$$\begin{aligned}\int_\omega \kappa(v_0)N_0\,dx \approx \int_\omega \kappa N_0\,dx &= \int_\omega \Big(\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\Big)N_0\,dx\\ &= -\int_\omega \frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla N_0\,dx+\int_{\partial\omega}\Big(\frac{\nabla\phi}{|\nabla\phi|}\cdot n\Big)N_0\,dx\\ &= -\sum_{i=1}^{6}\int_{\omega_i}\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla N_0\,dx,\end{aligned}$$

where n is the exterior unit normal at the boundary $\partial\omega$. Thus,

$$\kappa(v_0)\approx -\frac{\int_\omega \frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla N_0\,dx}{\int_\omega N_0\,dx}=-\frac{1}{h^2}\int_\omega\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla N_0\,dx,$$

where we used the fact that the integral of N_0 over ω is h^2.

Consequently, we propose the following formula for computing $\kappa_h(v_0)$, the finite element approximation of the curvature $\kappa(v_0)$:

$$\kappa_h(v_0)=-\frac{1}{h^2}\int_\omega\frac{\nabla\phi_h}{|\nabla\phi_h|}\cdot\nabla N_0\,dx, \tag{5.2}$$

where ϕ_h is the finite element interpolation of the level set function ϕ. If we denote $\phi_i=\phi(v_i)$, for $i=0,\cdots,6$, then

$$\phi_h(x)=\sum_{i=0}^{6}\phi_i N_i(x)$$

for all $x\in\omega$.

By a series of calculations, we have

$$
\begin{aligned}
-\int_{\omega_1}\frac{\nabla\phi_h}{|\nabla\phi_h|}\cdot\nabla N_0\,dx &= \frac{(\phi_2-\phi_0)h}{2\sqrt{(\phi_1-\phi_2)^2+(\phi_2-\phi_0)^2}},\\
-\int_{\omega_2}\frac{\nabla\phi_h}{|\nabla\phi_h|}\cdot\nabla N_0\,dx &= \frac{(\phi_2+\phi_3-2\phi_0)h}{2\sqrt{(\phi_0-\phi_3)^2+(\phi_2-\phi_0)^2}},\\
-\int_{\omega_3}\frac{\nabla\phi_h}{|\nabla\phi_h|}\cdot\nabla N_0\,dx &= \frac{(\phi_3-\phi_0)h}{2\sqrt{(\phi_0-\phi_3)^2+(\phi_3-\phi_4)^2}},\\
-\int_{\omega_4}\frac{\nabla\phi_h}{|\nabla\phi_h|}\cdot\nabla N_0\,dx &= \frac{(\phi_5-\phi_0)h}{2\sqrt{(\phi_5-\phi_4)^2+(\phi_0-\phi_5)^2}},\\
-\int_{\omega_5}\frac{\nabla\phi_h}{|\nabla\phi_h|}\cdot\nabla N_0\,dx &= \frac{(\phi_5+\phi_6-2\phi_0)h}{2\sqrt{(\phi_6-\phi_0)^2+(\phi_0-\phi_5)^2}},\\
-\int_{\omega_6}\frac{\nabla\phi_h}{|\nabla\phi_h|}\cdot\nabla N_0\,dx &= \frac{(\phi_6-\phi_0)h}{2\sqrt{(\phi_6-\phi_0)^2+(\phi_1-\phi_6)^2}}.
\end{aligned}
$$

Assume now ϕ is smooth enough. Taylor expanding the right-hand side of each of the above equations and then summing over $i=1,\cdots,6$, we then obtain with the help of Mathematica that

$$
\kappa_h(v_0)=-\frac{1}{h^2}\int_{\omega}\frac{\nabla\phi_h}{|\nabla\phi_h|}\cdot\nabla N_0\,dx=\nabla\cdot\frac{\nabla\phi(v_0)}{|\nabla\phi(v_0)|}+\mathcal{O}(h^2)=\kappa(v_0)+\mathcal{O}(h^2).
$$

This proves that our local averaging scheme (5.2) for the finite element approximation of the curvature has the error $\mathcal{O}(h^2)$.

We numerically test our local averaging method by calculating the curvature of a circular interface using different grid sizes. The domain is a square with side length one. The level set function ϕ is a cone centered in the center of the domain, cf. Fig. 9. Thus, the level sets of ϕ are circles whose curvatures are $\kappa=1/\text{radius}$. The zero level set of ϕ (representing our interface) is a circle of radius $1/4$ with $\kappa=4$.

We first calculate the curvature on a uniform mesh. This gives a function whose value at a particular point is the approximate curvature of the level set of ϕ at that point. We then refine the mesh to an interface-fitted mesh. The value of the curvature on the interface nodes is interpolated from the pre-refinement curvature calculation. We repeat this procedure on several grids from coarse to fine.

Fig. 9 shows the level set function on a particular mesh. Fig. 10, left, shows the calculated curvature on the domain. Note that near the center of the domain the level sets become small circles whose curvatures become infinite, so we cannot expect high accuracy in this area. Fig. 10, right, shows the absolute error in the curvature calculation across the domain. Note the error is higher on the boundary of the domain as we expect because the approximation is not $\mathcal{O}(h^2)$ there due to the lack of symmetry. Also, the error is higher near the center where the curvature should become infinite. The interface is

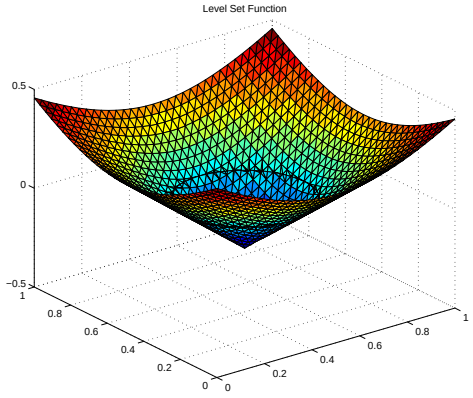

Figure 9: A cone shaped level set function used to test the curvature approximation. The interface is a circle of radius $1/4$.

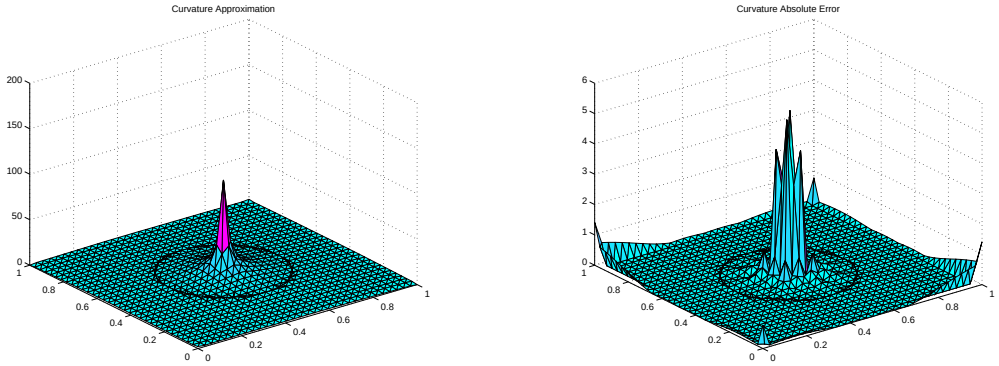

Figure 10: Left: Calculated curvature of the level set function. Right: Calculated curvature absolute error on the domain. Both with grid spacing $h=1/32$.

not located near these areas so the error is smaller on and near the interface. Fig. 11 is a log-log graph of the maximum relative error, defined by

$$\text{Relative Error} = \frac{|\kappa - \kappa_{calculated}|}{\kappa},$$

on the interface for several grid sizes. A simple linear regression on the data in the graph gives a best fit line with a slope of 2.065. This suggests that the error is $\mathcal{O}(h^2)$ as expected.

It is interesting to note from (5.2) that the largest curvature that can be calculated is inversely proportional to the grid spacing h, since by (5.2)

$$|\kappa_h(v_0)| = \left| -\frac{1}{h^2}\int_\omega \frac{\nabla\phi_h}{|\nabla\phi_h|}\cdot\nabla N_0\,dx \right| \le \frac{1}{h^2}\int_\omega |\nabla N_0|\,dx \le \frac{\sqrt{2}}{h^3}\int_\omega dx = \frac{3\sqrt{2}}{h}.$$

Fig. 12 illustrates this limit on various mesh sizes.

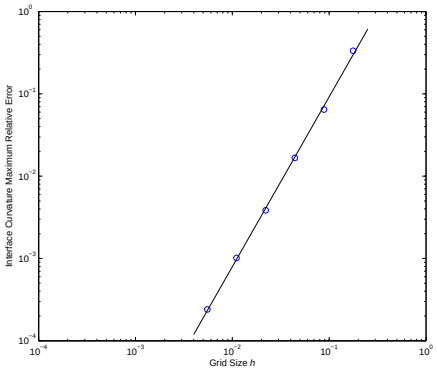

Figure 11: The maximum relative error of the calculated curvature on a circular interface versus grid spacing h. The simple linear regression line is plotted and it has slope 2.065.

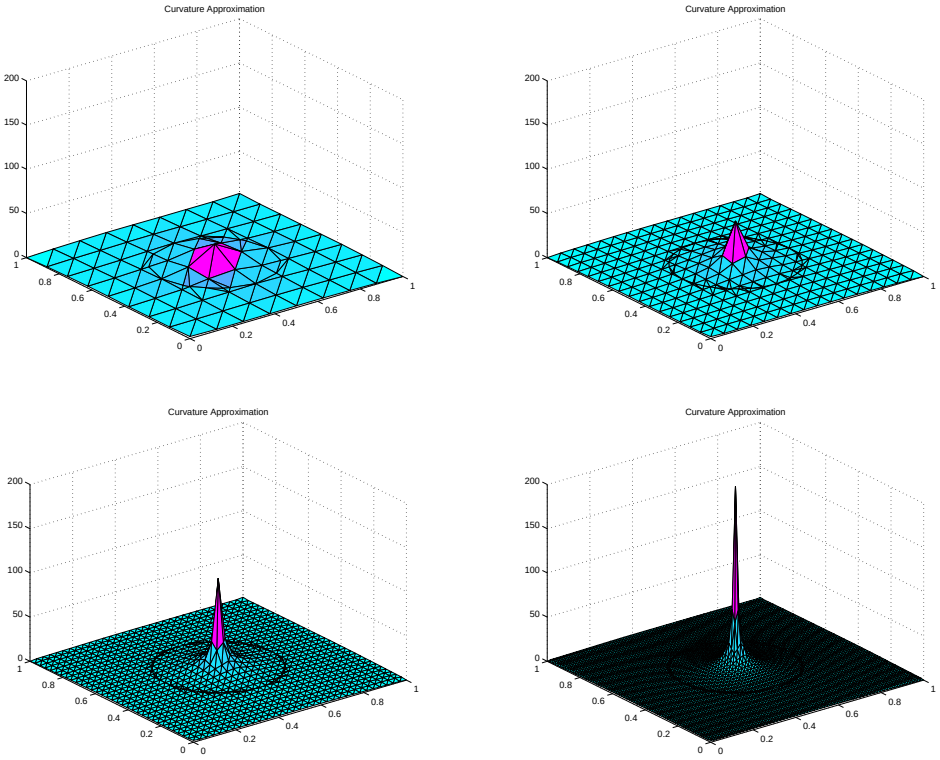

Figure 12: Calculated curvature of the level set function on various grid sizes: $h=1/8$, $1/16$, $1/32$, $1/64$.

6 Application to solidification

We apply our method to solve the Stefan problem that models dendritic solidification. We recover the previous results obtained by a finite difference level set method without interface-fitting [2].

Consider a frozen seed placed into a supercooled liquid. The frozen seed is represented by the area Ω_-, and the supercooled liquid is represented by the area Ω_+. The interface between these areas is Γ. As liquid solidifies around the frozen seed, Γ changes shape. In a rescaled form, the Stefan problem that describes the dynamics of this solidification process is as follows:

$$\frac{\partial T}{\partial t}=\Delta T, \qquad \text{in } \Omega_-(t)\cup\Omega_+(t), \tag{6.1a}$$

$$T(x,t)=-\varepsilon_C\kappa-\varepsilon_V v_n, \qquad \text{on } \Gamma(t), \tag{6.1b}$$

$$\frac{\partial T}{\partial n}=0, \qquad \text{on } \partial\Omega, \tag{6.1c}$$

$$v_n=-\left[\!\left[\frac{\partial T}{\partial n}\right]\!\right], \qquad \text{on } \Gamma(t), \tag{6.1d}$$

$$T(x,0)=0, \qquad \text{on } \Omega_-(t), \tag{6.1e}$$

$$T(x,0)=-\frac{1}{2}, \qquad \text{on } \Omega_+(t). \tag{6.1f}$$

Here, $T=T(x,t)$ is the temperature at point x and time t, ε_C and ε_V are the surface tension coefficient and the molecular kinetic coefficient, respectively, and κ and v_n are the curvature and normal velocity of the moving front Γ. Eq. (6.1b) is called the Gibbs-Thomson relation. Eq. (6.1d) is called the Stefan condition.

Our algorithm is as follows:

1. Choose a base mesh with spacing appropriate for the problem. Initialize the level set function. Input the parameters ε_C and ε_V. Input the initial temperature T;
2. Calculate the curvature κ as described in Section 5;
3. Interface-fit the mesh to Γ as detailed in Section 2. The calculated curvature κ is interpolated onto Γ, since it is used in the Gibbs-Thomson relation;
4. Solve the heat equation (6.1a) together with the interface and boundary conditions (6.1b) and (6.1c) for one time step;
5. Calculate the normal velocity using (6.1d);
6. Extend the normal velocity v_n onto the rest of the domain using the method explained in Section 3;
7. Unrefine the mesh back to the base mesh by simply throwing away the interface-fitted refinements;
8. Evolve ϕ on the base mesh by solving the level set equation (1.2) for one time step;
9. Reinitialize the level set function ϕ using the method detailed in Section 4, if necessary;
10. Go to Step 2.

Varying the surface tension coefficient ε_C. We vary our choice of ε_C and the initial shape of Γ to produce different results for comparison. The value ε_C can be zero, representing no surface tension; ε_C can be a constant, representing isotropic surface tension;

or ε_C can vary based on the direction of the normal to Γ, representing anisotropic surface tension. We let $\varepsilon_V = 0$, so the Gibbs-Thomson relation becomes simply $T(x,t) = -\varepsilon_C \kappa$. In the examples presented below, the initial frozen seed is a small pentagon. The computational domain is the square $\Omega = (-1.5,1.5) \times (-1.5,1.5)$. We use a 129×129 grid for the base mesh. The time step is $\Delta t = 0.00025$ and the final time is 0.4.

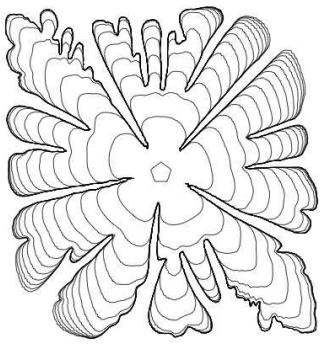

Figure 13: Unstable solidification: no surface tension $\varepsilon_C = 0$.

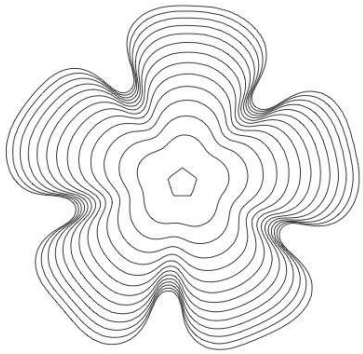

Figure 14: Solidification with isotropic surface tension $\varepsilon_C = 0.001$.

Fig. 13 shows the time progression of the case where $\varepsilon_C = 0$, representing no surface tension effects. As time progresses, the pentagon initially expands to form five dendritic arms, but these arms unstably split as the growth proceeds.

Fig. 14 shows the case where $\varepsilon_C = 0.001$, indicating a small isotropic surface tension effect. The initial frozen seed expands to form five dendritic arms, but unlike the previous case with no surface tension, the dendritic arms do not split. Instead, they expand outward with a smooth edge.

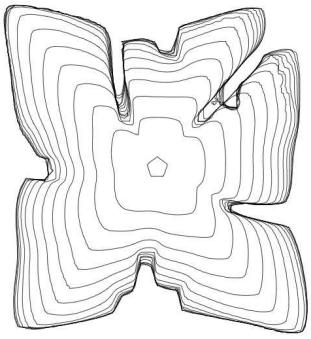

Figure 15: Solidification with anisotropic surface tension.

Fig. 15 shows the case where $\varepsilon_C = 0.001\left[(8/3)\sin^4(2(\theta - 45^\circ))\right]$, with $\overline{\varepsilon}_C = 0.001$, representing anisotropic surface tension effects. Growth is favored on the interface where the normal direction is diagonal while growth is inhibited where the interface normal direction is vertical or horizontal.

Varying the phase angle of an anisotropy. Here we vary the phase angle θ_0 of the anisotropic surface tension and anisotropic kinetic effects. Specifically, we consider the ε_C and ε_V in the Gibbs-Thomson relation to be

$$\varepsilon_C(n) = \overline{\varepsilon}_C\big[1-\cos(4(\theta+\theta_0))\big],$$
$$\varepsilon_V(n) = \overline{\varepsilon}_V\big[1-\cos(4(\theta+\theta_0))\big],$$

where θ is the angle formed by the outward normal n and the positive x-axis measured counterclockwise. These give a fourfold anisotropy. In our examples, we let $\overline{\varepsilon}_C = \overline{\varepsilon}_V = 0.001$. Our initial interface representing the frozen seed is an eight-sided star shape. Our computational domain is $\Omega = (-1,1)\times(-1,1)$. We use a 129×129 grid for the base mesh. The time step is $\Delta t = 0.0001$ and the final time is 0.4.

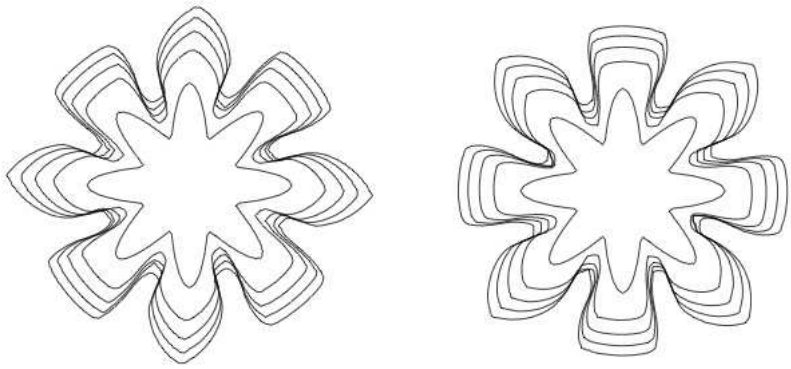

Figure 16: Solidification with a four-fold anisotropic surface tension. Left: The phase angle is 0°. Right: The phase angle is 45°. The shape is shown at $t = 0, 0.01, 0.02, 0.03$, and 0.04 in both cases.

Fig. 16 shows that the growth is favored in the vertical and horizontal directions when the phase angle $\theta_0 = 0^\circ$ and is favored in the diagonal directions when the phase angle $\theta_0 = 45^\circ$.

7 Application to solvation

We now apply our method to the numerical simulation of molecular solvation using the recently developed variational implicit-solvent (i.e., continuum-solvent) model [1, 3, 4, 7, 8]. In this model, an underlying system of molecules in a solution is divided geometrically into three parts: the solute region Ω_s, the solvent (e.g., water) region Ω_w, and the corresponding solute-solvent interface Γ that separates the solute region Ω_s and the solvent region Ω_w, cf. Fig. 17 where n is the unit normal along the solute-solvent interface Γ pointing from the solute region Ω_s to the solvent region Ω_w. Here, we assume that there is a sharp interface that separates the solvent and solutes, and we treat the solvent as a continuum. We also assume that there are N solute atoms in the system that are located at $x_1, \cdots, x_N$ inside Ω_s. These solute particles are fixed.

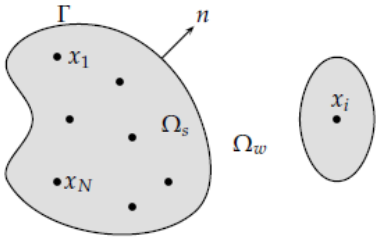

Figure 17: A schematic description of a solvation system with an implicit solvent.

The basic assumption in this model is that the experimentally observed equilibrium molecular structures minimize the following effective solvation free-energy functional:

$$G[\Gamma]=\int_\Gamma \gamma dS+\rho_w\sum_{i=1}^{N}\int_{\Omega_w}U_{sw}(|x-x_i|)dV. \tag{7.1}$$

Here, the first term is the surface energy, where γ is the surface tension. In a simple case, one can take γ to be a positive constant. The second term is the van der Waals type interaction energy between solute particles located at $x_1,\cdots,x_N$ and solvent molecules governed by a Lennard-Jones type potential (cf. Fig. 18)

$$U_{sw}(r)=4\varepsilon_{sw}\left[\left(\frac{\sigma_{sw}}{r}\right)^{12}-\left(\frac{\sigma_{sw}}{r}\right)^{6}\right]. \tag{7.2}$$

The parameter ρ_w is the solvent density which we take to be a constant. The parameters ε_{sw} and σ_{sw} can vary with different particles.

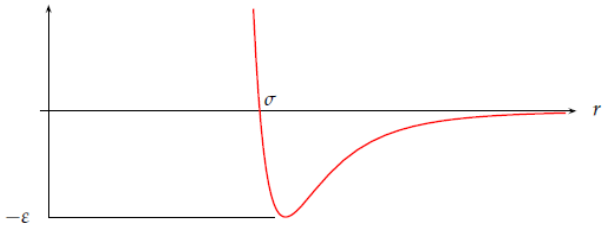

Figure 18: The Lennard-Jones Potential $U(r)=4\varepsilon\left[(\sigma/r)^{12}-(\sigma/r)^{6}\right]$.

To find a (local) minimizer of the functional (7.1), we use the level set method. We begin with an initial surface that surrounds all the solute particles located at $x_1,\cdots,x_N$. We then evolve the surface in the direction of steepest decent of the free energy. This means that the surface moves with the normal velocity same as the negative first variation of the free energy $G[\Gamma]$ with respect to the location change of the surface Γ [3]:

$$v_n=-\delta_\Gamma G[\Gamma]=-\gamma\kappa+\rho_w\sum_{i=1}^{N}U_{sw}(|x-x_i|), \tag{7.3}$$

where κ is the (mean) curvature.

Our algorithm is as follows:

1. Choose an initial base mesh with spacing appropriate for the problem. Input parameters and initialize the level set function. The input parameters are the surface tension constant γ, the Lennard-Jones energy parameters σ_{sw}, ε_{sw} (for each particle) and the solute particle locations $x_1,\cdots,x_N$;
2. Calculate the curvature κ as described in Section 5;
3. Interface-fit the grid to Γ;
4. Calculate the Lennard-Jones potential term $\rho_w\sum_{i=1}^{N}U_{sw}(|x-x_i|)$ on Γ (at the vertices on Γ);
5. Extend the Lennard-Jones potential term away from Γ using the method of normal velocity extension as explained in Section 3;
6. Solve the level set equation (1.2) with the extended normal velocity (7.3);
7. Reinitialize the level set function ϕ;
8. Go to Step 2.

Note that in Step 5 we extend the Lennard-Jones potential from the interface rather than using its default values away from the interface. This way we can avoid the rapid change of the Lennard-Jones potential near solute particles. Such rapid changes can cause numerical instabilities.

Since our functional (7.1) is in general nonconvex and our method is a relaxation method, different initial surfaces will lead to possibly different final, relaxed surfaces that represent equilibrium solute-solvent interfaces. It is therefore crucial to define different kinds of initial surfaces. Based on how close a surface is to some or all of the fixed solute particles, we classify initial surfaces into three types: loose wraps, tight wraps, and mixed wraps. A loose wrap is a large surface that loosely wraps all the particles. For instance, a large spherical surface that contains all the solute particles $x_1,\cdots,x_N$ is a loose wrap. A tight wrap is a small surface that tightly wraps all the particles. An initial level set function that represents a tight wrap can be generated by

$$\phi(x)=\min_{1\le i\le N}(|x-x_i|-\hat{\sigma}_i),$$

where each $\hat{\sigma}_i>0$ is small enough, e.g., $\hat{\sigma}_i<\sigma_i/2$. A mixed wrap is a surface with a locally loose wrap for some particles and locally tight wrap for other particles.

In what follows we test on three artificial systems to demonstrate how our model and method can capture quantitatively some of the physical properties in real molecular systems. In these tests, we choose our computational domain to be $\Omega=(0,1)\times(0,1)$. We use some artificial parameters that can be viewed as rescaled from real physical parameters. To avoid complication, we shall also use only a loose wrap or a tight wrap as an initial surface.

A three-particle system. We place two particles at $(0.3,0.35)$ and $(0.6,0.65)$. Both of them have $\varepsilon=0.005$ and $\sigma=0.15$. We place the third particle at $(0.5,0.2)$ with the corresponding values $\varepsilon=0.003$ and $\sigma=0.1$. We also set the parameters $\gamma=0.0001$ and

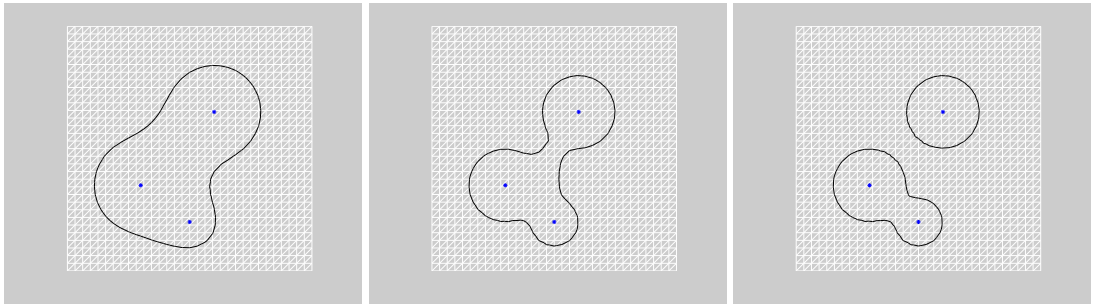

Figure 19: Level set relaxation of the solute-solvent interface for the three-particle system. Left: A loose wrap initial surface. Middle: The final, relaxed surface. Right: An intermediate surface during the relaxation.

$\rho = 0.5$ in the free energy functional (7.1). We generate a loose wrap by initializing the level set function ϕ to be at least 1.5σ from the ith particle for all i. Fig. 19 shows the progression of the surface Γ to the steady state that represents the optimal solute-solvent interface. We find our level set method captures the break up of the interface.

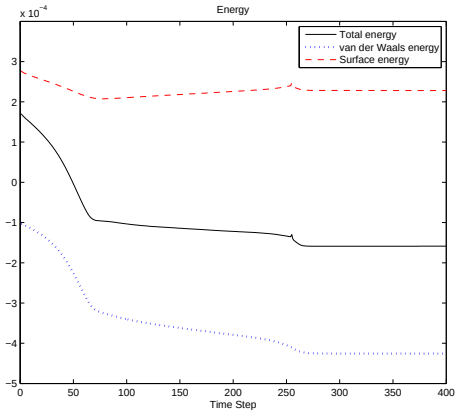

Figure 20: Free energy vs. step number for the three-particle system.

Fig. 20 shows the value of the surface energy, van der Waals energy, and the total free energy as the algorithm proceeds. The top line represents the surface energy. It increases from time step 75 to time step 250 as the interface becomes longer. As the surface energy increases, it is more than offset by the decrease of the van der Waals energy so that the total free energy (surface energy plus van der Waals energy) monotonically decreases to a steady state.

A ring system. We now consider an artificial ring molecule that consists of 8 solute particles evenly distributed on a circle. Fig. 21 shows a sequence of snapshots of our level set calculations starting with a large circle, a loose wrap. In the relaxed system, there is no solvent inside the ring region of solute molecule. This is the dewetting phenomena that is important in biomolecular solvation. Fig. 22 shows a different sequence of snapshots

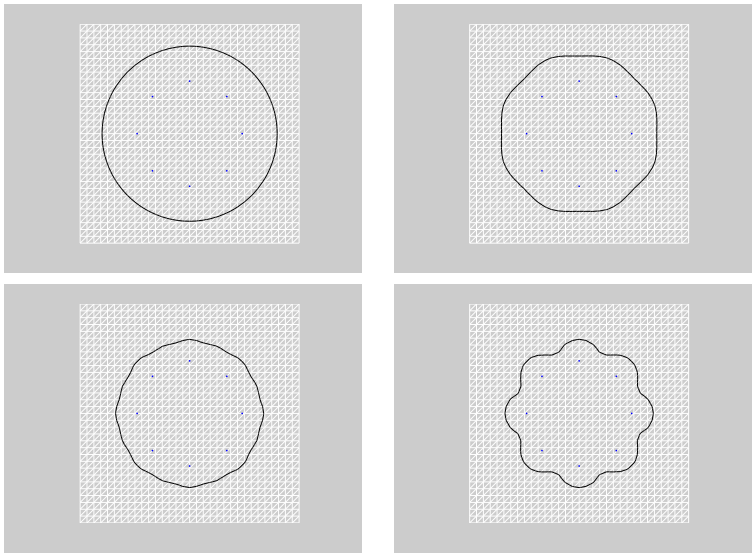

Figure 21: Snapshots of the level set relaxation of the solute-solvent interface for a ring molecule with a loose-wrap initial surface. Order: from left to right and from top to bottom.

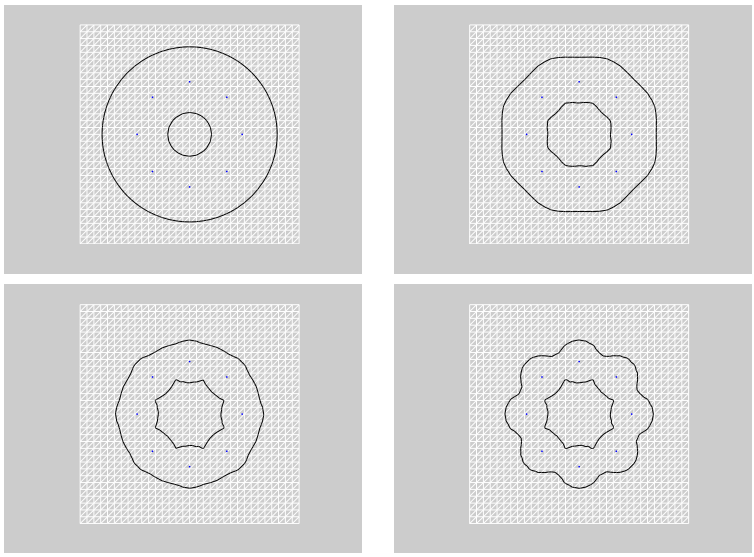

Figure 22: Snapshots of the level set relaxation of the solute-solvent interface for a ring molecule with a tight-wrap initial surface. Order: from left to right and from top to bottom.

of our level set calculations starting with two circles, one large and one small, that form a ring containing all the solute particles. In the relaxed system, the solvent is inside the ring region of solute molecule. These examples demonstrate that our level set method captures different local minima of the free-energy functional.

A two-rod system. This system consists of 18 solute particles with 9 evenly distributed on one line and the other 9 evenly distributed on a different line. These two lines are parallel. The parameters in the model are: $\gamma = 0.00035$, $\rho_w = 0.5$, all $\varepsilon = 0.0062$, and all $\sigma = 0.1$. We consider different values of the distance d between these two lines. For each of the distance d, we consider both loose-wrap and tight-wrap initial surfaces.

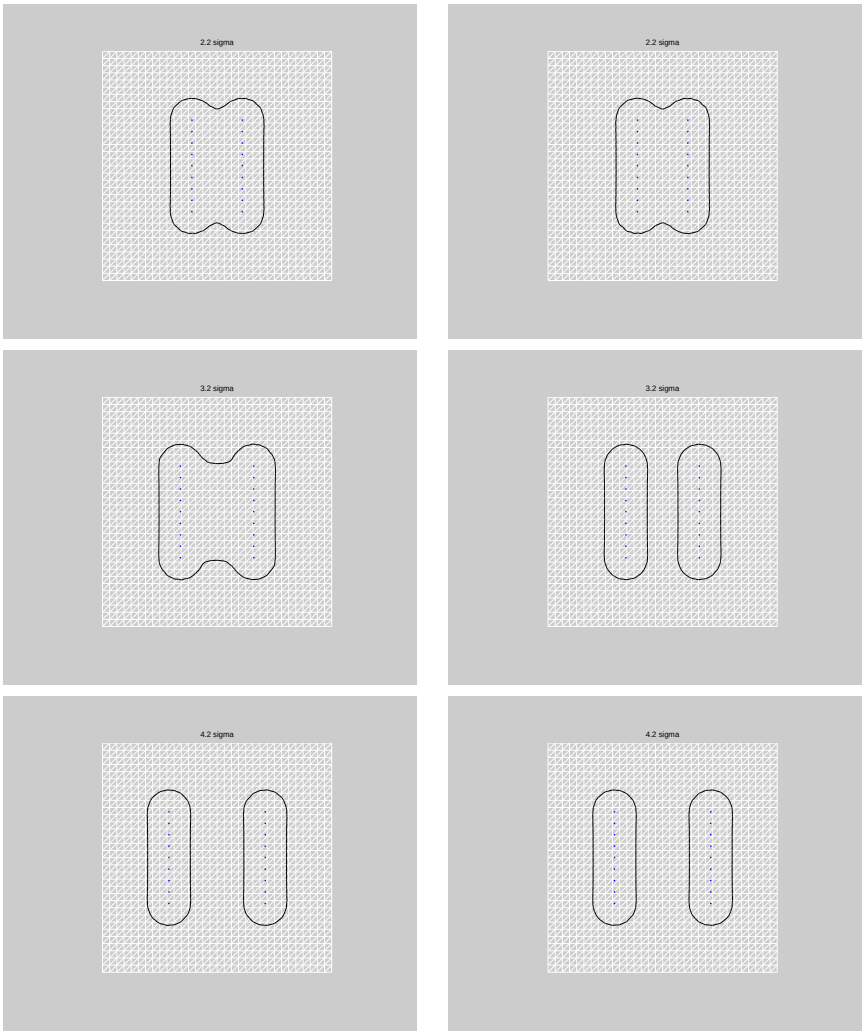

Figure 23: Top: $d = 2.2\sigma$. Middle: $d = 3.2\sigma$. Bottom: $d = 4.2\sigma$. The left ones are relaxed from initially loose wraps and the right ones are from initially tight wraps.

Fig. 23 shows the result of our level set calculations. When the distance is small, the free-energy minimizing surface is a one-component surface that wraps all the 18 particles. When the distance is large, the free-energy minimizing surface is a two-component surface with each component wraps all the particles on the same rod, and the system is a

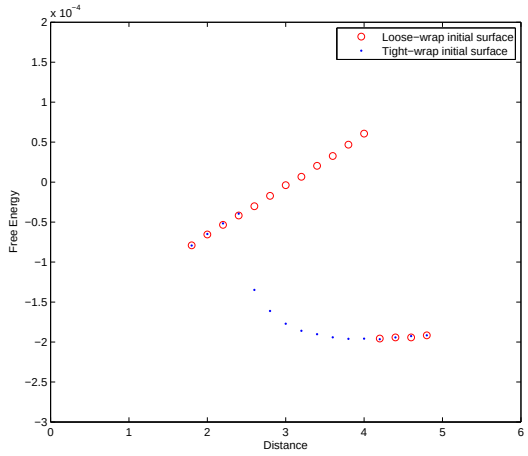

Figure 24: Free energy vs. distance d for the two-rod system.

two-rod molecule with each one a linear polymer. In both cases, the free-energy minimizing surface seems to be unique, regardless of loose-wrap or tight-wrap initial surfaces. When the rod separation distance is neither too small nor too large, there are two local minimizers of the free energy functional: a one-component large surface wrapping all the particles, corresponding to the loose initial surface, and a two-component surface with each component wrapping tightly the particles on one line. The one-component surface shows the dewetting phenomenon: inside the surface there is no solvent.

Fig. 24 plots the free energy values vs. the distance d. It is rather clear that when the distance d is in the range between $2.4\sigma \sim 4\sigma$, the system has two local minimizers, one corresponding to an initial tight wrap and the other to loose wrap. The free energy of the one-component equilibrium surface is larger than that of the two-component tight-wrap surface. This means that dewetting is a local but not global minimizer of the underlying system.

8 Conclusions

We have introduced a finite element based level set method for numerically evolving surfaces. The key feature of this method is the construction of an interface-fitted mesh and its unrefinement with respect to a fixed base mesh in each time step of evolution. An advantage of using such interface-fitted meshes is that a high accuracy can be achieved in solving field equations together with interface conditions. Another advantage of our method is to relatively easily handle interfaces with complex geometries.

We have developed new level set techniques for the extension of normal velocity and the reinitialization of level set function. The main idea is to solve Laplace's or Poisson's equations on both sides of the interface with suitable interface conditions. Approximate solutions to such elliptic interface problems can be obtained by iteration methods with a

few iterations. We have also designed a simple, finite element based scheme for computing the curvature. Our analysis and numerical tests show a second-order convergence of this scheme.

We have used our method to solve the classical solidification problem and the new molecular solvation problem. We have reproduced the previous numerical results of dendritic solidification. We have also captured some key properties of variational solvation of molecules. In addition, our numerical tests have shown high accuracy of these techniques.

Our method needs to be much refined and improved, particularly in two aspects.

i. *Combined with the adaptive finite element method.* When an interface-fitted mesh is initially constructed, it is often of poor quality in terms of the mesh regularity. Using the adaptive finite element method to solve the field equations, one can simultaneously improve the quality of the mesh. A possible difficulty is to keep nodes on the interface unchanged.

ii. *Extension to three-dimension.* In principle, this can be carried out. But much implementation work is needed.

Acknowledgments

B. Li and J. Shopple were supported by the US National Science Foundation (NSF) through the grant DMS-0451466 and DMS-0811259, and by the US Department of Energy through the grant DE-FG02-05ER25707. B. Li was also supported by the NSF Center for Theoretical Biological Physics (CTBP) under the NSF grant PHY-0822283 and by the grant Award Number R01GM096188 from the National Institute of General Medical Sciences (NIGMS), the National Institutes of Health (NIH). The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIGMS or NIH. The authors thank Dr. Li-Tien Cheng for helpful discussions.

References

[1] J. Che, J. Dzubiella, B. Li, and J. A. McCammon, Electrostatic free energy and its variations in implicit solvent models, J. Phys. Chem. B., 112 (2008), 3058–3069.
[2] S. Chen, B. Merriman, S. Osher, and P. Smereka, A simple level set method for solving Stefan problems, J. Comput. Phys., 135 (1997), 8–29.
[3] L.-T. Cheng, J. Dzubiella, J. A. McCammon, and B. Li, Application of the level-set method to the implicit solvation of nonpolar molecules, J. Chem. Phys., 127 (2007), 084503.
[4] L.-T. Cheng, Y. Xie, J. Dzubiella, J. A. McCammon, J. Che, and B. Li, Coupling the level-set method with molecular mechanics for variational implicit solvation of nonpolar molecules, J. Chem. Theory. Comput., 5 (2009), 257–266.
[5] K. P. Deckelnick, G. Dziuk, and C. M. Elliott, Computation of geometric PDEs and mean curvature flow, Acta. Numer., 14 (2005), 139–232.

[6] D. A. Di Pietro, S. Lo Forte, and N. Parolini, Mass preserving finite element implementations of the level set method, Appl. Numer. Math., 56 (2006), 1179–1195.
[7] J. Dzubiella, J. M. J. Swanson, and J. A. McCammon, Coupling hydrophobicity, dispersion, and electrostatics in continuum solvent models, Phys. Rev. Lett., 96 (2006), 087802.
[8] J. Dzubiella, J. M. J. Swanson, and J. A. McCammon, Coupling nonpolar and polar solvation free energies in implicit solvent models, J. Chem. Phys., 124 (2006), 084905.
[9] M. Fried, A level set based finite element algorithm for the simulation of dendritic growth, Comput. Visual. Sci., 7 (2004), 97–110.
[10] Y. Gong, B. Li, and Z. Li, Immersed-interface finite-element method for elliptic interface problems with non-homogeneous jump conditions, SIAM J. Numer. Anal., 46 (2008), 472–495.
[11] S. Groß, V. Reichelt, and A. Reusken, A finite element based level set method for two-phase incompressible flows, Comput. Visual. Sci., 9 (2004), 239–257.
[12] S. Osher, and R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, 2003.
[13] S. Osher, and J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys., 79 (1998), 12–49.
[14] J. A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge, 1999.
[15] A.-K. Tornberg, Interface Tracking Methods with Application to Multiphase Flows, PhD thesis, KTH, Stockholm, Sweden, 2000.
[16] A.-K. Tornberg, and B. Engquist, A finite element based level set method for multiphase flow applications, Comput. Visual. Sci., 3 (2000), 93–101.
[17] H. Xie, K. Ito, Z. Li, and J. Toivanen, A finite element method for interface problems with locally modified triangulations, Contemp. Math., 466 (2008), 179–190.
[18] X. Xie, and Z. Li, A finite element method for elasticity interface problems with locally modified triangulations, Int. J. Numer. Methods., 2009, in press.