J Sci Comput (2008) 35: 330–349
DOI 10.1007/s10915-008-9188-6

# A Local Semi-Implicit Level-Set Method for Interface Motion

**David Salac · Wei Lu**

Received: 18 June 2007 / Revised: 21 September 2007 / Accepted: 22 January 2008 /
Published online: 13 February 2008


**Abstract** This paper proposes and implements a novel hybrid level set method which combines the numerical efficiency of the local level set approach with the temporal stability afforded by a semi-implicit technique. By introducing an extraction/insertion algorithm into the local level set approach, we can accurately capture complicated behaviors such as interface separation and coalescence. This technique solves a well known problem when treating a semi-implicit system with spectral methods, where spurious interface movements emerge when two interfaces are close to each other. Numerical experiments show that the proposed method is stable, efficient and scales up well into three dimensional problems.

**Keywords** Interface motion · Surface diffusion · Level set approach · Semi-implicit scheme · Localized treatment · Hybrid algorithm

## 1 Introduction

The level set method has recently become an invaluable tool for investigating the motion of interfaces in a wide variety of systems and situations. For example, the method has successfully been employed to investigate electromigration [7], epitaxial growth [2] and evolving fluid interfaces [16]. While useful, the basic level set method is hampered by high computational costs, especially in situations concerning surface diffusion. To relieve this constraint, two different classes of approaches have been developed. The first class, known as the local level set method, aims to reduce the overall computational cost by localizing the level set calculation [14]. The second class aims at developing semi-implicit scheme to increase the temporal stability, allowing for larger time steps to be utilized compared to explicit methods [17]. In this paper we propose a novel hybrid level set method that combines the numerical efficiency of the local level set approach with the temporal stability afforded by a semi-implicit technique, and an extraction/insertion algorithm to accurately capture complicated behaviors such as interface separation and coalescence.

D. Salac · W. Lu (✉)
Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA
e-mail: weilu@umich.edu

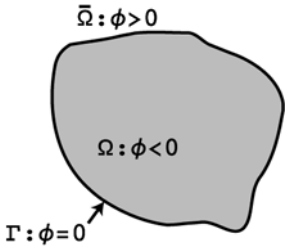


**Fig. 1** A schematic of the level set representation. The body is defined by negative values of the function $\phi$ while the interface is the zero level set of the function

First introduced by Sethian and Osher [12], the central concept of the level set approach is to describe an interface, $\Gamma$, implicitly by embedding it into a function of higher dimensionality, $\phi$. The interface is given by the zero level-set of $\phi$, i.e.

$$\Gamma(t) = \{\mathbf{x} | \phi(\mathbf{x}, t) = 0\}, \tag{1}$$

where $\mathbf{x}$ is a position vector and $t$ is time. To describe a body that occupies the space $\Omega(t)$ we can define the level set function $\phi$ such that

$$\phi(\mathbf{x}, t) \begin{cases} < 0 & \text{in } \Omega(t), \\ = 0 & \text{on } \Gamma(t), \\ > 0 & \text{in } \bar{\Omega}(t) \end{cases} \tag{2}$$

with $\bar{\Omega}(t)$ indicating the region outside the body, as shown in Fig. 1.

Using this level set formulation the normal of the interface, $\mathbf{n}$, is given by

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}. \tag{3}$$

The positive normal direction points outward from $\Omega(t)$, i.e. from the region of negative to positive $\phi(\mathbf{x}, t)$.

The curvature, $\kappa$, is given by the divergence of the normal [14],

$$\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}. \tag{4}$$

The curvature is positive for a convex surface of $\Omega(t)$.

While any function that satisfies (2) can be a valid choice for $\phi$, it is advantageous to choose specific forms with known good properties. Here we use the signed distance function. By definition, a signed distance function describes the shortest distance from any given point in space to the closest interface, with a positive value on one side of the interface (outside) and a negative value on the other side (inside).

The motion of the interface carries the associated $\phi(\mathbf{x}, t)$ field in a way similar to a flow that carries mass. Denote the interface velocity by $\mathbf{v}$ and consider a control volume, it is easy to show that

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0. \tag{5}$$

The interface velocity can be decomposed into components normal ($v_n \mathbf{n}$) and tangential ($v_t \mathbf{t}$) to the interface. As $\mathbf{t} \cdot \nabla \phi = \mathbf{0}$, (5) can be rewritten in the form of the standard evolution

equation [14],

$$\frac{\partial \phi}{\partial t} + v_n |\nabla \phi| = 0. \tag{6}$$

The interface motion is determined by calculating the normal velocity of the interface and advancing the level-set function using (6). In general the normal velocity of the interface can be caused by various thermodynamic forces via processes such as surface diffusion. Examples include surface tension [12, 17], elasticity [15] and electrostatic interactions [7]. In this paper we focus on interface motion driven by surface energy, a mechanism that exists in all interface systems and is computational challenging due to its dependence on interface curvature.

Generally speaking, the chemical potential of an atom on an interface, $\mu$, is position dependent. Atoms can diffuse on the interface from one region to another to reduce the chemical potential. This surface diffusion is important in many physical systems and known to cause morphology changes [10, 11]. Surface diffusion is characterized by mass conservation. The normal velocity of the interface can be expressed by $v_n = M\nabla_s^2 \mu$, where $M$ is a surface mobility term [7]. The surface Laplacian is defined as $\nabla_s^2 = \nabla_s \cdot \nabla_s$, where $\nabla_s = \nabla - \mathbf{n}\partial_n$ is the surface gradient and $\partial_n = \mathbf{n} \cdot \nabla$ [17]. One can easily put $M$ into the time scale without losing generality, so we assign $M = 1$. The chemical potential on the surface is the curvature multiplied by the surface energy density, $\gamma$, which may be spatially dependent. Thus the normal velocity is given by [3, 17]

$$v_n = \nabla_s^2(\gamma\kappa). \tag{7}$$

Several researchers, such as Chopp and Sethian [3], Smereka [17], and Khenner et al. [6] have investigated methods to solve interface motion by diffusion. The main difficulty in many of these methods (see [3, 6]) relates to the fact that (6) is extremely stiff. In fact, solving this level set evolution equation is analogous to solving the fourth-order differential equation $\phi_t = -\phi_{xxxx}$. Any explicit time-discretization method will require the time step to scale as $(\Delta x)^4$, where $\Delta x$ is the grid spacing. Clearly this is a very stringent condition. One possible route to reduce the overall computational time is to use a local level set method, which restricts calculations to a small region around the interface [14]. However, this approach does not remove the stringent time step constraint.

Ideally an implicit method would be utilized to advance the level set as this would remove any time step restriction. The nonlinear nature of (6) makes the fully implicit method difficult, if not impossible, to accomplish. An alternative approach was proposed by Smereka where a *semi*-implicit method was utilized to increase the overall stability of the algorithm. Using this scheme, time steps as large as $15000(\Delta x)^4$ have been demonstrated. While successful in ameliorating the time step requirement, the current semi-implicit method used a global smoothing scheme. This scheme would introduce non-physical interface motion especially when two surfaces are close. Such spurious motion could be detrimental in simulating some phenomena such as interface coalescence. The scheme also faces challenges to scale up to large three-dimensional problems since it is carried out over the entire simulation domain.

In the following we propose a novel hybrid level set method that combines the numerical efficiency of the local level set approach with the temporal stability afforded by a semi-implicit method while accurately capturing interface coalescence by an extraction/insertion algorithm. The plan of this article is the following. The numerical scheme is developed in Sect. 2. Section 3 presents both two- and three-dimensional results demonstrating the capability of the proposed method. Section 4 summarizes the key features and future developments.

## 2 The Local Semi-Implicit Level Set Method

In this section we begin with general semi-implicit schemes for the level set method and then move on to local implementation and level set extraction. This is followed by a discussion of techniques we have employed to ensure mass conservation. Algorithms necessary for the scheme are included in the appendices.

### 2.1 Semi-Implicit Schemes for the Level Set Method

A semi-implicit scheme for (6) can be obtained by adding bilaplacian stabilization terms:

$$\frac{\partial \phi}{\partial t} + \eta \nabla^4 \phi = \eta \nabla^4 \phi - v_n |\nabla \phi|, \tag{8}$$

with $\eta$ a positive constant. By writing the time differential to first order accuracy, implicitly calculating the left hand bilaplacian, and explicitly calculating the right hand terms we obtain the discrete form of (8),

$$\left(1 + \Delta t \eta \nabla^4\right) \delta \phi^{n+1} = -\Delta t v_n \left|\nabla \phi^n\right|. \tag{9}$$

Here $\Delta t$ is the time step, $\phi^n$ is the level set function at time $t$, $\phi^{n+1}$ is the level set function at time $t + \Delta t$, and $\delta \phi^{n+1} = \phi^{n+1} - \phi^n$ is the level set change at the current time step. The bilaplacian term acts as a smoothing operator applied to the explicit scheme, suppressing the unstable high wave number modes [17]. Compared to explicit methods this approach allows for the use of larger time steps without a loss of stability. Similar concepts have been applied to non-conserved level set methods [15] and phase-field models [8, 19].

Previous applications of the semi-implicit method were limited to periodic domains, where the Fast Fourier Transform (FFT) technique was used to solve (9). This approach faces challenges in three-dimensional problems since the FFT quickly becomes inefficient as the system size increases [4]. Consequently, the FFT-based semi-implicit schemes are constrained to two-dimensional or small-scale three-dimensional simulations. Additionally, studies have shown that the use of the FFT can introduce spurious interfacial motion [17]. In the following we propose an efficient non-FFT scheme that eliminates non-physical motion and scales up well in three-dimensions.

A major difficulty in solving (9) using a non-FFT scheme is that the system of equations from discretization of (9) is neither diagonally dominate nor compact. Due to this fact a typical iterative scheme may not converge. We have found that by integrating the Biconjugate Gradients Stabilized Method (BI-CGSTAB) [21] and a bilaplacian stencil with isotropic discretization error [13] into our local semi-implicit level set scheme, we obtain convergence in all situations.

### 2.2 The Local Level Set Method

The key idea of the local level set scheme is to avoid evolving (6) in the whole domain since only the zero-level set, which defines the interface, directly relates to physical motion. Instead, we can define a calculation tube enclosing the interface and calculate interface dependent quantities such as curvature and velocity in this small tube only. This scheme significantly reduces the amount of computation. To do so, we choose a constant $\beta > 0$ which is on the order of the grid spacing. The calculation tube $\Lambda_\beta$ is defined by all grid points within a distance of $\beta$ from the interface.

The integration of local level set and semi-implicit scheme is achieved in the following way. We calculate the interfacial curvature and surface Laplacian in the calculation tube $\Lambda_\beta$. These values are then used to update the level set within the calculation tube by the semi-implicit scheme in (9). By applying the Biconjugate Gradients Stabilized Method (BI-CGSTAB) [21] and a bilaplacian stencil with isotropic discretization error [13] to advance (9) in each time step we have achieved nice convergence and accurate results. However, when the calculation involves two interfaces close to each other, we have observed interfacial distortions and poor temporal stability. This phenomenon can be understood by looking into the stencils obtained by discretizing (7) and (9). Both equations depend on fourth-order derivatives. A second-order discretization of the derivative results in a stencil which is, at minimum, 5 grid points wide. If there is not enough separation between two interfaces, the level set calculation of the two physically separate interfaces can numerically affect each other. The crosstalk can lead to non-physical interface motion. To better under the phenomenon, Fig. 2 shows two close interfaces whose calculation tubes overlap. Physically the motion of two interfaces is independent of each other before they contact. However, the sharing of some grid points (black points) in calculating interface quantities such as curvature causes the two interfaces to numerically sense the existence of each other. This phenomenon makes

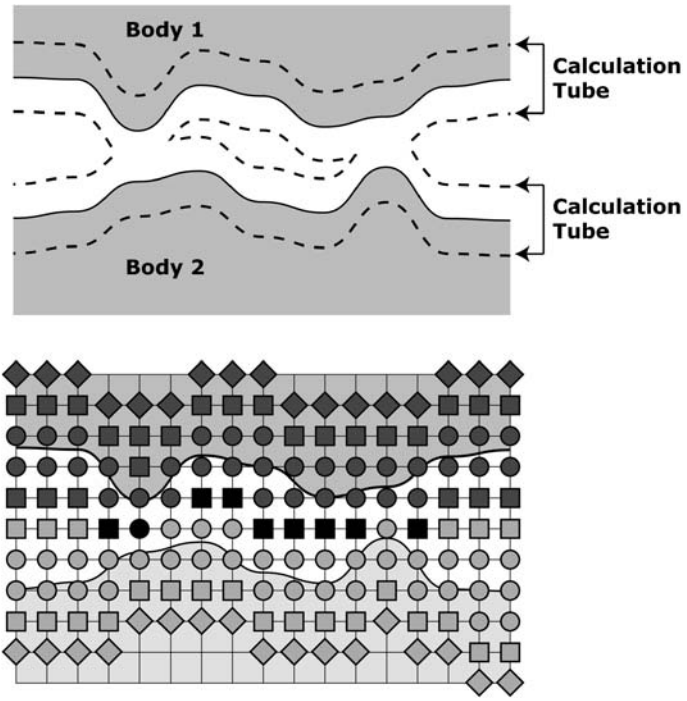


**Fig. 2** (**a**) When two interfaces are close to each other, their calculation tubes can overlap. (**b**) Illustration of crosstalk between two interfaces before they physically contact due to the sharing of some grid point values. Grid points are marked with icons of different shapes to indicate their distance to the interface. *Circle*: within one grid; *square*: within two grids; *diamond*: within three grids. Grid points with dark shading are associated with the upper interface, which means that the calculation of the interface quantities involves these points. Grid points with light shading are associated with the lower interface. Note that the *black points* are involved in the calculation of both interface quantities such as curvature. Thus the two physically separated interfaces sense each other and crosstalk emerges

the level set method unable to accurately capture processes such as interface coalescence or separation. In the following we propose a level set extraction/insertion approach to resolve this issue.

### 2.3 Level Set Extraction

Here we propose an extraction/insertion approach as shown in Fig. 3 to deal with close interfaces. We first identify distinct bodies, which are then extracted into individual, temporary level set functions. The interface motion of each body is calculated separately. The updated interfaces are inserted back into the original level-set function. In the following we elaborate on the process of body identification and extraction.

Consider a system consisting of multiple bodies described by a level set function. From the level set we know whether a region is interior ($\phi < 0$) or exterior ($\phi > 0$), but do not

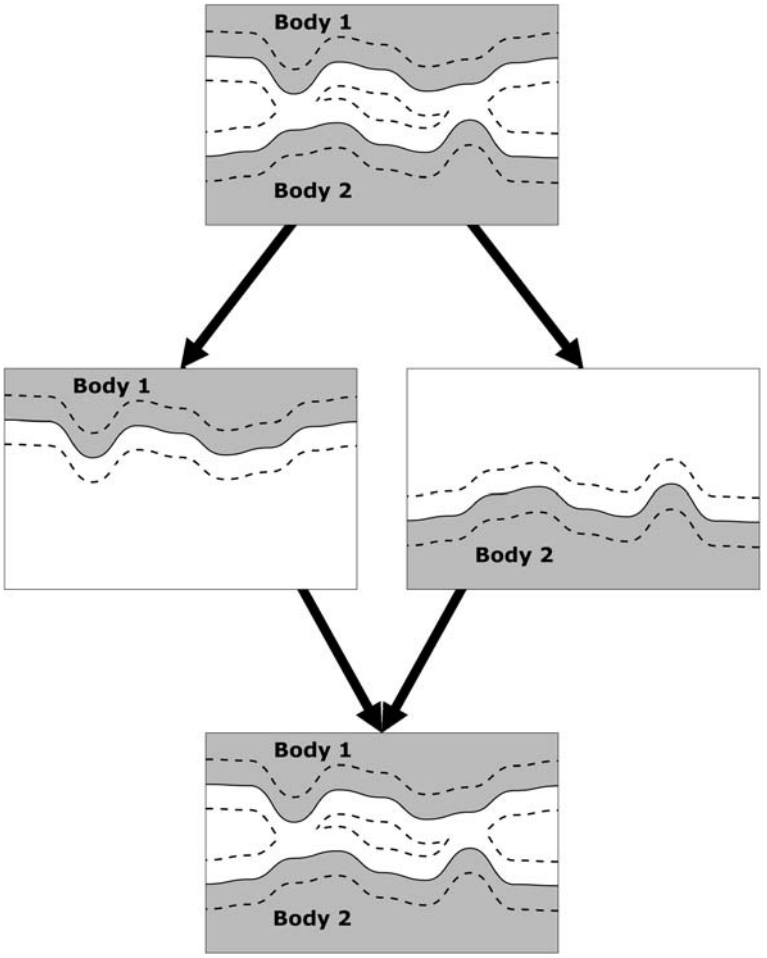


**Fig. 3** Illustration of the extraction/insertion approach. The *top* figure is the original level set function at time $t$. The two figures in the *middle* show the extracted bodies to calculate separately the updated interface locations. In the *bottom* figure the updated locations are inserted back into the original level set function at time $t + \Delta t$

know explicitly the number of distinct bodies or which grid points belong to which body. We begin by marking all grid points as NB (No Body). We scan all the NB points to identify any interior grid point ($\phi < 0$). Define the first found interior grid point as a seed point and mark it A, which means that the point belongs to a unique body A. Scan for all interior grid points directly next to a point marked A, and mark them A as well. Continue the process until the search results in zero hit. We refer to this process as finding all of the "connected" points within body A. Among the set of NB points scan for another interior grid point. This seed point belongs to a different body and is marked B. In a similar manner as identifying body A, we can identify all of the "connected" points within body B. Continue to determine other unique bodies until there is no interior grid point left in the set of NB. This procedure can identify all unique bodies in the computation domain since there must exist at least one exterior grid point between different bodies.

The level set algorithm also requires information of the exterior grid points within the calculation tube $\Lambda_\beta$ of each interface. The following is the procedure. We begin by considering all exterior grid points directly next to an interface. For each of these exterior points, scan its surrounding grid points to determine whether the surrounding points are inside any body. If all surrounding grid points except those marked NB are only inside one body, then we mark this exterior grid point as "independent" and associate it to that body. On the other hand, the surrounding grid points may be inside two or more bodies, i.e. this exterior grid point is next to two or more unique bodies. In that case, we can not associate this exterior grid point to any single body. We mark these types of grid points as "dependent". After all exterior grid points directly next to an interface are marked as either "independent" or "dependent", we move to the exterior grid points directly next to this first set of exterior grid points. This process is repeated until all exterior grid points within the calculation tube $\Lambda_\beta$ have been marked. An example of this delineation is shown in Fig. 2, where "dependent" grid points are marked black while "independent" grid points are colored according to which body they are associated to. Note that interior grid points of a body are associated to that body only and thus "independent".

After body identification and all grid points within the calculation tubes have been associated to a body or marked as dependent, we can begin the extraction procedure. For each unique body we extract the independent grid points associated to this body and put the original level set values into a temporary level set function, $\tilde{\phi}$. At dependent grid points directly next to the extracted body we explicitly calculate the distance to the interface using a technique shown in Adalsteinsson and Sethian [1]. We then construct a temporary signed distance function at grid points not associated to this extracted body but within $\beta$ grid points of its interface using a third-order, Weighted Essentially Non-Oscillatory (WENO3) upwind reinitialization scheme (see Appendix). Note that the interface of $\tilde{\phi}$ corresponds to that of the original body while the values away from the interface may differ from the original level set function. Any values intrinsic to the interface such as curvature are not affected by this extraction process.

We use $\tilde{\phi}$ to calculate the curvature information at points directly next to the extracted interface with an approach similar to [17]. The curvature information is then extended by at least 2 grid points away from the interface using an extension procedure (see Appendix). Using the extended curvature it is possible to calculate $\tilde{v}_n = \nabla_s^2(\gamma(\mathbf{x})\tilde{\kappa})$ at grid points next to the interface. This process is followed by an extension of $\tilde{v}_n$ to the rest of the calculation tube $\tilde{\Lambda}_\beta$. We then calculate $\Delta t\tilde{v}_n|\nabla\tilde{\phi}|$ in $\tilde{\Lambda}_\beta$. The next step is to solve for $\delta\tilde{\phi}^{n+1}$ by (9) using the BI-CGSTAB/isotropic bilaplacian method. Here we solve for $\delta\tilde{\phi}^{n+1}$ in a smaller tube around the interface, $\tilde{\Lambda}_\alpha$, using the points in $\tilde{\Lambda}_\beta$ but outside of $\tilde{\Lambda}_\alpha$ as boundary conditions. This technique allows for the application of an iterative method to solve the semi-implicit

system of equations. The smaller tube is defined by all grid points within a distance of $\alpha$ from the interface, where $\beta \geq \alpha + n_\nabla$ and $n_\nabla$ depends on the specific discretization form used for the bilaplacian smoothing term in (9). The parameter $\alpha$ should be large enough so that the interface stays within $\tilde{\Lambda}_\alpha$ during one time step or designated number of time steps. On the other hand, a large $\alpha$ also means wider calculation tube and thus more computation. Thus one need to balance the two conditions when deciding $\alpha$.

Now we have obtained the updated interface position for each extracted body. In the following we put this information back into the original level set function. For independent grid points within $\tilde{\Lambda}_\alpha$ we can simply set $\delta\phi^{n+1} = \delta\tilde{\phi}^{n+1}$ and update the level set value such that $\phi^{n+1} = \phi^n + \delta\phi^{n+1}$. Dependent grid points are treated differently since they are shared by two interfaces. Our approach is to extend the velocity calculated from independent grid points to these dependent points. We then update their values using a first-order explicit discretization of the level set evolution equation (6).

Recently Macklin and Lowengrub [9] developed an approach to calculate the curvature at points close to two different interfaces. This approach is more accurate than the standard treatment and can alleviate some of the spatial errors associated with merging interfaces. However, the approach does not alleviate any time step restrictions. If a semi-implicit scheme with a bilaplacian stabilization term is utilized, the crosstalk between bodies would still occur, resulting in spurious motion of interfaces. The proposed level set extraction approach allows the implementation of semi-implicit schemes, leading to both accurate capture of interface merging and fast computation.

### 2.4 Discretizations

Unless otherwise noted all spatial derivatives are approximated by second-order central difference functions. The curvature of the level set is written in two-dimensions as

$$\kappa = \frac{\phi_{xx} + \phi_{yy}}{(\phi_x^2 + \phi_y^2 + \varepsilon)^{1/2}} - \frac{\phi_x^2\phi_{xx} + \phi_y^2\phi_{yy} + 2\phi_x\phi_y\phi_{xy}}{(\phi_x^2 + \phi_y^2 + \varepsilon)^{3/2}}, \tag{10}$$

and in three-dimensions as

$$\kappa = \frac{\phi_{xx} + \phi_{yy} + \phi_{zz}}{(\phi_x^2 + \phi_y^2 + \phi_z^2 + \varepsilon)^{1/2}} - \frac{\phi_x^2\phi_{xx} + \phi_y^2\phi_{yy} + \phi_z^2\phi_{zz} + 2(\phi_x\phi_y\phi_{xy} + \phi_x\phi_z\phi_{xz} + \phi_y\phi_z\phi_{yz})}{(\phi_x^2 + \phi_y^2 + \phi_z^2 + \varepsilon)^{3/2}}, \tag{11}$$

where $\varepsilon$ is a small parameter to ensure that the denominator does not equal zero. The surface Laplacian of the curvature is calculated in a way similar to Smereka [17]. First consider the surface gradient of the curvature term, $\nabla_s(\gamma\kappa) = \nabla(\gamma\kappa) - \mathbf{n}\partial_n(\gamma\kappa)$. In component form this can be written as

$$\begin{aligned}\nabla_s(\gamma\kappa) &= (\gamma\kappa)_x\mathbf{e}^x + (\gamma\kappa)_y\mathbf{e}^y + (\gamma\kappa)_z\mathbf{e}^z \\ &\quad - \big(n^x(\gamma\kappa)_x + n^y(\gamma\kappa)_y + n^z(\gamma\kappa)_z\big)\big(n^x\mathbf{e}^x + n^y\mathbf{e}^y + n^z\mathbf{e}^z\big) \\ &\equiv A\mathbf{e}^x + B\mathbf{e}^y + C\mathbf{e}^z.\end{aligned} \tag{12}$$

Here $\mathbf{e}^x$, $\mathbf{e}^y$ and $\mathbf{e}^z$ denote unit vectors in the $x$-, $y$-, and $z$-directions, while $n^x$, $n^y$, and $n^z$ are the unit normal directions of the interface obtained using central differences. Then by computing the surface divergence of (12) the surface Laplacian of the curvature is given by

$$\begin{aligned}\nabla_s^2 &= A_x + B_y + C_z - n^x\big(n^xA_x + n^yA_y + n^zA_z\big) - n^y\big(n^xB_x + n^yB_y + n^zB_z\big) \\ &\quad - n^z\big(n^xC_x + n^yC_y + n^zC_z\big).\end{aligned} \tag{13}$$

To calculate $|\nabla\phi|$ in (9) we utilize a third-order Weighted Essentially Non-Oscillatory (WENO3) upwind method (see Appendix). Upwind schemes are entropy-satisfying and attempt to take information from behind the moving front [16]. The key idea is to choose the smoothest possible derivative to at least third order accuracy by weighing multiple possible stencils [5].

Finally we discuss about the how to choose the form for the bilaplacian stabilization term in (9). The simplest form would be to use central differences to calculate all the directional derivatives, resulting in an stencil with anisotropic discretization error [13]. We have found that using this discretization is not optimal for two reasons. First, the anisotropic nature of the stencil can influence the motion of the interface, as the leading order error terms in the discretization depend on the underlying grid. Second, the resulting system of equations using this stencil is difficult to solve using iterative approaches due to the extremely diagonally weak system. Instead we utilize an isotropic stencil which aids in the stability of the simulation and allows for the use of iterative methods to solve the semi-implicit scheme. In particular we utilize the fourth second-order isotropic three-dimensional bilaplacian stencil given by Patra and Karttunen [13]. Here the stencil size remains at $5 \times 5 \times 5$, but the points utilized and their weights are different from the standard second-order central difference scheme. The advantages of such a method include having isotropic discretization error and a system which is not as diagonally weak as the standard method. Using this stencil we set $n_\nabla = 4$ to determine the outer calculation tube.

### 2.5 Mass Conservation

One issue with the level set method is mass conservation. Theoretically the level set method is mass conserving. In practice numerical discretization can introduce large mass change during the course of a simulation, particularly during the reinitialization process. To counter this problem we have modified a global mass correction scheme [22] for the proposed level set method. Before the simulation begins we calculate the total mass of each body in the simulation using a mollified Heaviside function,

$$H(\phi) = \begin{cases} 0 & \phi < -\psi, \\ \frac{\psi+\phi}{2\psi} + \frac{1}{2\pi}\sin(\frac{\pi\phi}{\psi}) & |\phi| \le \psi, \\ 1 & \phi > \psi, \end{cases} \tag{14}$$

where $\psi$ is a finite thickness on the order of the grid spacing. The mass of a body $l$ is simply a summation over the points contained in and near the body,

$$M_l = \sum_{i,j,k} V_{ijk} H(-\phi_{ijk}), \tag{15}$$

where $V_{ijk}$ is the volume of the element containing the grid point $\mathbf{x}_{ijk}$ and $\phi_{ijk}$ is the level set value at that point. After updating the level set for one time step we iterate over all the bodies in the system and perform a mass correction step. This procedure begins by defining a mass correction factor, $M_l^{\text{cor}} = (M_l^0 - M_l^\tau)/M_l^0$, where $M_l^0$ is the original mass of body $l$ and $M_l^\tau$ is mass of the same body at pseudo time $\tau$. We then solve the following pseudo-time differential equation for each body,

$$\frac{\partial\phi_l}{\partial\tau} = -M_l^{\text{cor}}, \tag{16}$$

until a prescribed accuracy is achieved. The accuracy of mass conservation at pseudo-time $\tau$ is measured by $|M_l^0 - M_l^\tau|/M_l^0$. In the equation $\phi_l$ denotes the level set grid points in and near body $l$. In practice the calculation domain of (16) is body $l$ plus the grid points directly next to the interface but outside the body. We solve (16) with a variable time step $\Delta\tau = \lambda\Delta x$, where $\lambda$ is chosen through a standard line search to ensure that the mass error always decreases. The updated level set is given by $\phi_l^{n+1} = \phi_l^n - \lambda\Delta x M_l^{\text{cor}}$. This algorithm typically converges to a tolerance of $10^{-6}$ within 5 to 10 iterations. This mass correction method solves an issue of the global correction schemes, where a body may receive incidental correction when mass change actually happens in another body. The mass correction step adjusts the interface location to ensure mass conservation of the enclosed body. The flat region of an interface which does not move (where $v_n = 0$) may change its location slightly as a result of this process. If the mass correction step is taken frequently, the induced movement will be small and not affect the physics. In our simulations the mass correction was performed in each time step of the interface motion.

### 2.6 The Algorithm

The algorithm described in previous sections is shown here in the complete form.

**Step 0.** Initialize the level set function $\phi^0$ to the signed distance function and calculate the initial mass of each body.

**Step 1.** Mark all grid points within $\beta$ grid spacing of the interface and find extraction information.

**Step 2.** Iterate over the number of bodies in the system.

a. Extract the current body using the independent grid points associated with the body.
b. Reinitialize all grid points within a distance of $\beta$ grid spacing of the extracted interface to a signed distance function, $\tilde{\phi}$. The level set function at grid points directly next to the interface is calculated explicitly as discussed in Sect. 2.3.
c. Set $\tilde{\kappa} = 0$ everywhere.
d. Calculate $\tilde{\kappa}$ at grid points next to the extracted interface using (10) or (11).
e. Extend $\tilde{\kappa}$ from the interface by at least 2 grid points using an extension algorithm (see Appendix C).
f. Set $\tilde{v}_n = 0$ everywhere.
g. Calculate $\tilde{v}_n = \nabla_s^2(\gamma(\mathbf{x})\tilde{\kappa})$ at grid points next to the extracted body using (12) and (13).
h. Extend $\tilde{v}_n$ to $\tilde{\Lambda}_\alpha$ and $\tilde{\Lambda}_\beta$. Save $\tilde{v}_n$ at independent grid points in $\Lambda_\alpha$ for use in Step 3.
i. Calculate $\Delta t\tilde{v}_n|\nabla\tilde{\phi}|$ using upwind WENO3 in $\tilde{\Lambda}_\alpha$ and $\tilde{\Lambda}_\beta$.
j. Solve (9) using BI-CGSTAB and an isotropic bilaplacian discretization for $\delta\tilde{\phi}^{n+1}$. Use $\tilde{\Lambda}_\beta$ as boundary conditions for $\tilde{\Lambda}_\alpha$.
k. At independent grid points in $\Lambda_\alpha$ set $\delta\phi^{n+1} = \delta\tilde{\phi}^{n+1}$.

**Step 3.** Extend $v_n$ from independent grid points in $\Lambda_\alpha$ to dependent grid points in $\Lambda_\alpha$ and the grid points in $\Lambda_\beta$.

**Step 4.** At $\Lambda_\beta$ and dependent grid points in $\Lambda_\alpha$ calculate $\delta\phi^{n+1} = -\Delta t v_n|\nabla\phi^n|$ using a WENO3 upwind scheme.

**Step 5.** Update $\Lambda_\alpha$ and $\Lambda_\beta$ with $\phi^{n+1} = \phi^n + \delta\phi^{n+1}$.

**Step 6.** Correct for any mass gain or loss by solving $\frac{\partial\phi_l}{\partial\tau} = -M_l^{\text{cor}}$.

**Step 7.** One time step has been completed. Return to Step 1 and repeat.

## 3 Results

In this section we present results of interface motion using the algorithm described above. All results were computed on square or cube grids with uniform grid spacing. We set $\alpha = 4$ and $\beta = 8$ to define the computation tubes. Following [17] the stabilization coefficient was chosen to be $\eta = 0.5$. With these parameters the BI-CGSTAB method solves the semi-implicit equation (9) within 40 iterations to a 2-norm tolerance of $10^{-8}$. Mass correction was performed in each time step to an accuracy of $10^{-6}$. We start by considering a representative shape evolution problem and demonstrate convergence of the proposed method. Then we look into a series of two- and three-dimensional simulations.

Consider the interface motion and shape evolution of an ellipse, as shown in Fig. 4. The grid size is $64 \times 64$ with grid spacing of $\Delta x = 0.05$ and time step of $\Delta t = 0.0001$. The large curvature at the tips of the ellipse drives the surface there to move inward and eventually

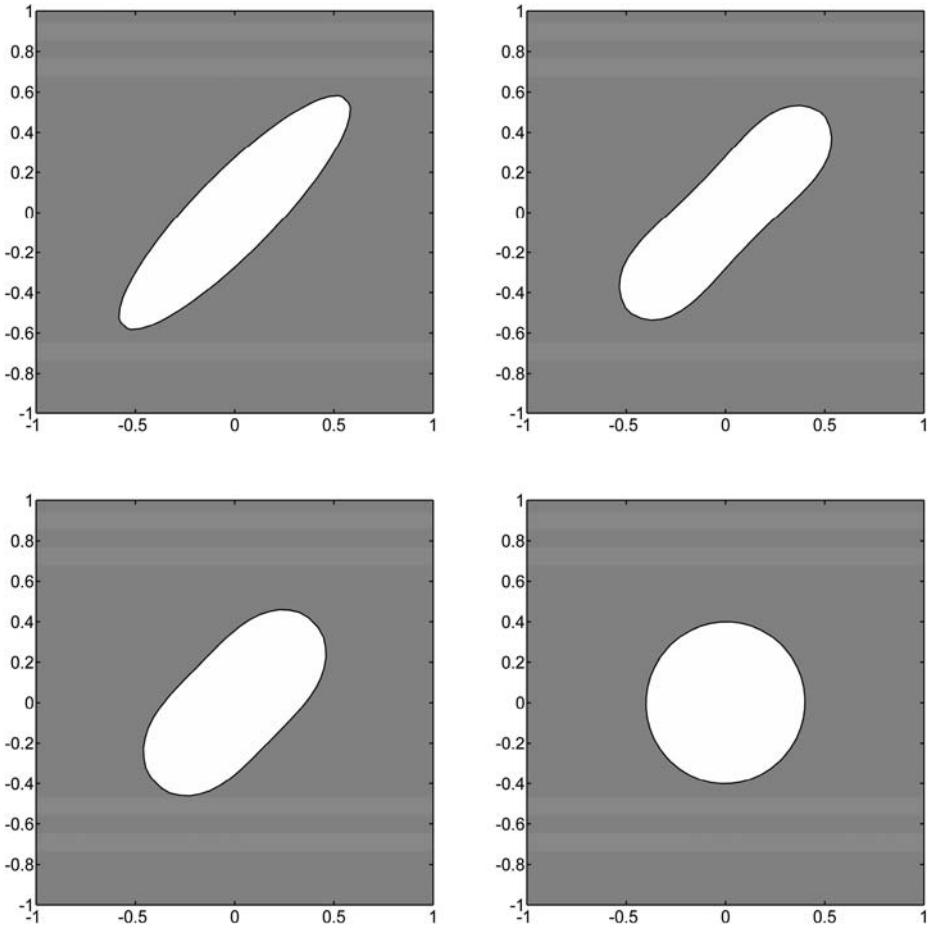


**Fig. 4** Shape evolution of an ellipse via surface diffusion. Computation with grid size $64 \times 64$, grid spacing $\Delta x = 0.05$ and time step $\Delta t = 0.0001$. Results shown for time (**a**) 0, (**b**) 0.0012, (**c**) 0.005, (**d**) 0.03

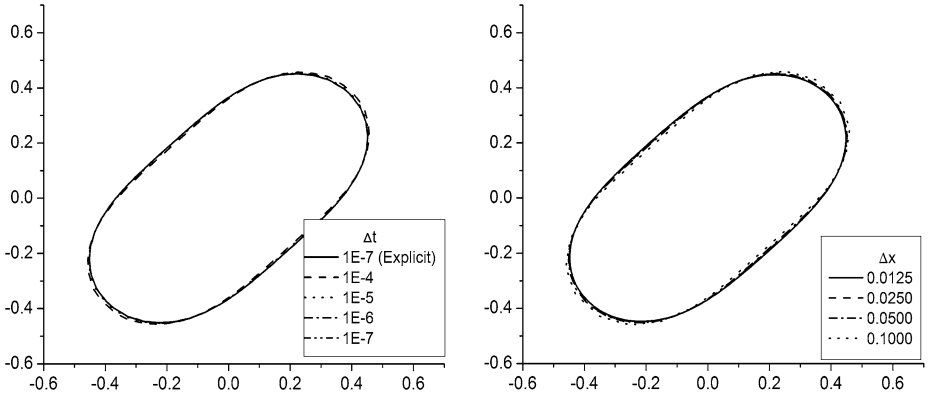


**Fig. 5** Convergence check. (**a**) Temporal convergence. Computation with grid size $64 \times 64$, grid spacing $\Delta x = 0.05$. The time step of the local semi-implicit method varies from $\Delta t = 1 \times 10^{-7}$ to $\Delta t = 1 \times 10^{-4}$. The results are also compared to an explicit surface-diffusion simulation for a time step of $\Delta t = 1 \times 10^{-7}$. (**b**) Spatial convergence. Results shown for a time step of $\Delta t = 1 \times 10^{-7}$. Grid spacing and associated grid size are: $\Delta x = 0.0125$ ($256 \times 256$), $\Delta x = 0.025$ ($128 \times 128$), $\Delta x = 0.05$ ($64 \times 64$), and $\Delta x = 0.1$ ($32 \times 32$)

the body evolves into a circular shape. In this simulation we have $\Delta t/(\Delta x)^4 = 16$, a large improvement over the explicit scheme where $\Delta t/(\Delta x)^4 \approx 0.25$ [3]. In fact, we can further increase the time step by using a larger $\alpha$ and $\beta$ (the width of the computation tubes). Here the constraint is not the stability of the algorithms, but the width of the calculation tube since we need to ensure that the interface does not leave $\Lambda_\alpha$ in a time step. In the extreme case, when the tube is the whole domain, we expect to achieve time step as large as the global semi-implicit scheme where $\Delta t/(\Delta x)^4 \approx 10^3$ [17]. However, wider calculation tubes increase the amount of computation in each time step. Thus there is optimized choice of the calculation tube from computational point of view. From physical point of view, in many cases we do not need extremely large time step due to the requirement of time resolution. Computational optimization should be considered together with the physical problem to determine whether choosing a larger calculation tube and time step, or smaller calculation tube and time step.

Figure 5 demonstrates the temporal and spatial convergence of the proposed local semi-implicit method. We calculated the case in Fig. 4 with various time step and grid spacing to $t = 0.005$. The time convergence appears to be very quick compared to the global semi-implicit scheme. Note that the time steps shown in Fig. 5a represent a wide range of $\Delta t/(\Delta x)^4$ conditions from $0.016 \sim 16$, indicating the accuracy and stability of the proposed method. We have also compared the local semi-implicit scheme to a fully explicit method without mass correction. The local semi-implicit scheme matches well with the explicit approach. Figure 5b shows quick spatial convergence.

The method can handle bodies with large positive and negative curvatures very well. Figure 6 shows the evolution of a star-like shape to a circle. The motion shows the expected behavior. We have let this simulation run up to a time of $t = 0.5$ with no change in the resulting circle. Similarly, we have evolved a split-annulus type shape shown in Fig. 7. Here an annulus is bisected along a diagonal and the ends are rounded, with the interfaces separated by only 1.5 grid spacing. Due to the use of our local semi-implicit scheme, we do not see any attraction between the ends of the shapes. Both shapes evolve into the expected circles.

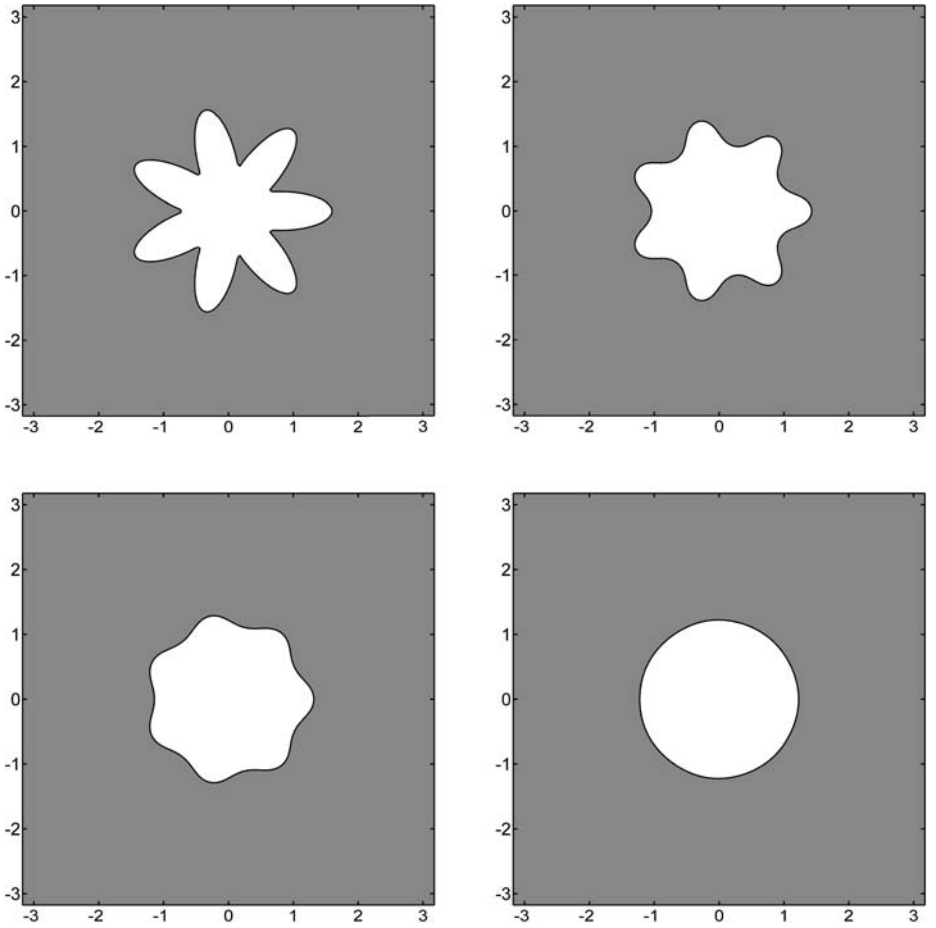


**Fig. 6** Evolution of a shape with highly positive and negative curvatures. Computation with grid size $128 \times 128$, grid spacing $\Delta x = 0.05$ and time step $\Delta t = 0.0001$. Results shown for time (**a**) 0, (**b**) 0.005, (**c**) 0.0075, (**d**) 0.015

We have run the simulation till a long time (up to $t = 1.0$) and did not observe any change in the equilibrium shapes.

A major advantage of the proposed method is to accurately capture interface coalescence. To demonstrate this, consider a system containing an ellipse and a circle, as shown in Fig. 8. The expected motion is for the ellipse to evolve towards an equilibrium circular shape as that in Fig. 8. The circle is expected to maintain its shape until the ellipse impedes on the boundary of the circle. In the global semi-implicit method, spurious movement of the circle's interface was observed during the simulation [17]. This effect was attributed to the use of the FFT to solve the semi-implicit evolution equation. In contrast, the proposed local semi-implicit scheme does not have this spurious motion. The two shapes are able to be in very close proximity without any incorrect interfacial movement. The computation accurately shows that it is not until the ellipse impedes on the circle do the two coalesce into a single body.

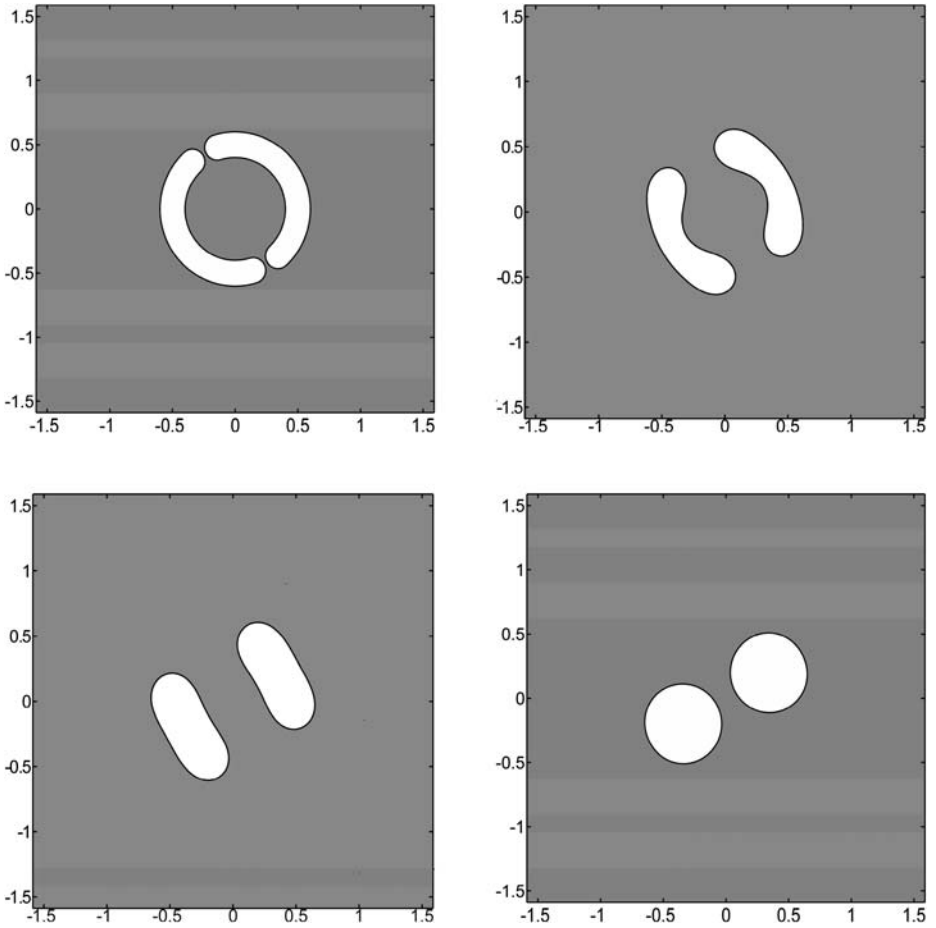


**Fig. 7** Evolution of an annulus bisected along a diagonal, with the interfaces separated by only 1.5 grid spacing at the beginning. Our local semi-implicit scheme accurately captures the motion without any spurious attraction between the ends of the two shapes. Both shapes evolve into the expected circles. Computation with grid size $128 \times 128$, grid spacing $\Delta x = 0.025$ and time step $\Delta t = 5 \times 10^{-6}$. Results shown for time (**a**) 0, (**b**) 0.001, (**c**) 0.0025, (**d**) 0.01

The method scales up well in three-dimensions. The evolution of an irregular star shape is shown in Fig. 9. The resulting shape is a sphere which is energetically favorable. The simulation captures the effect of curvature on the evolution dynamics. Figure 10 shows the pinching off of a dumbbell-like shape. Here the largest curvature exists at the connection between the center tubular part and the spherical ends. Diffusion causes the tubular part to narrow and finally break. This simulation demonstrates the importance of kinetics. The system evolves into a state of two equal spheres rather than a single larger sphere, although the latter would have smaller surface area. Energy alone cannot determine the equilibrium structure.

Finally we present the evolution of a highly irregular body with little symmetry, as shown in Fig. 11. After some time of evolution small openings begin to close off. Figure 11b shows a hollow eight-sided shape. This shape then splits into two separate parts as shown in

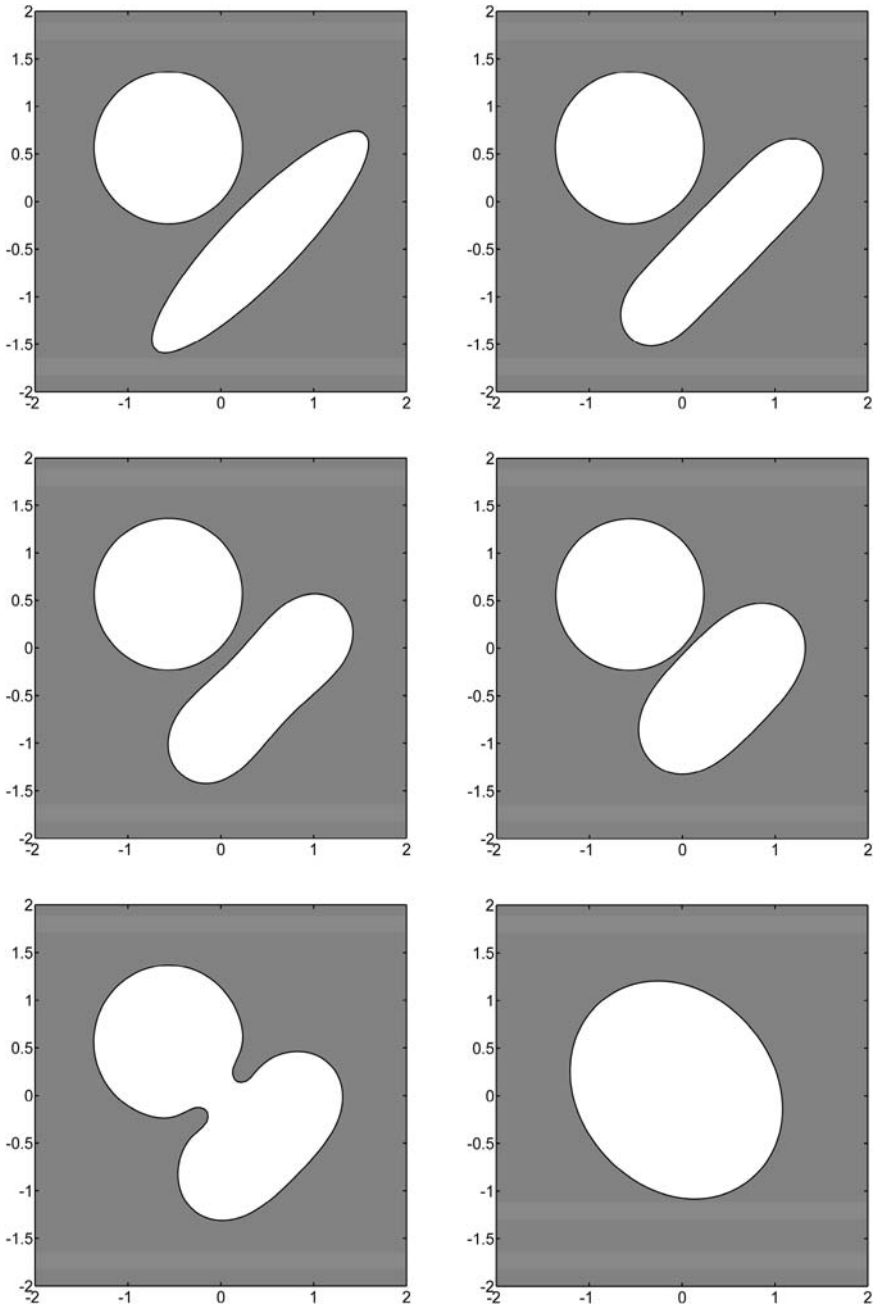

**Fig. 8** The proposed method accurately captures the coalescence of two bodies. Computation with grid size $128 \times 128$, grid spacing $\Delta x = 0.05$, and time step $\Delta t = 0.0001$. Results shown for time (**a**) 0, (**b**) 0.01, (**c**) 0.04, (**d**) 0.085, (**e**) 0.09, (**f**) 0.5

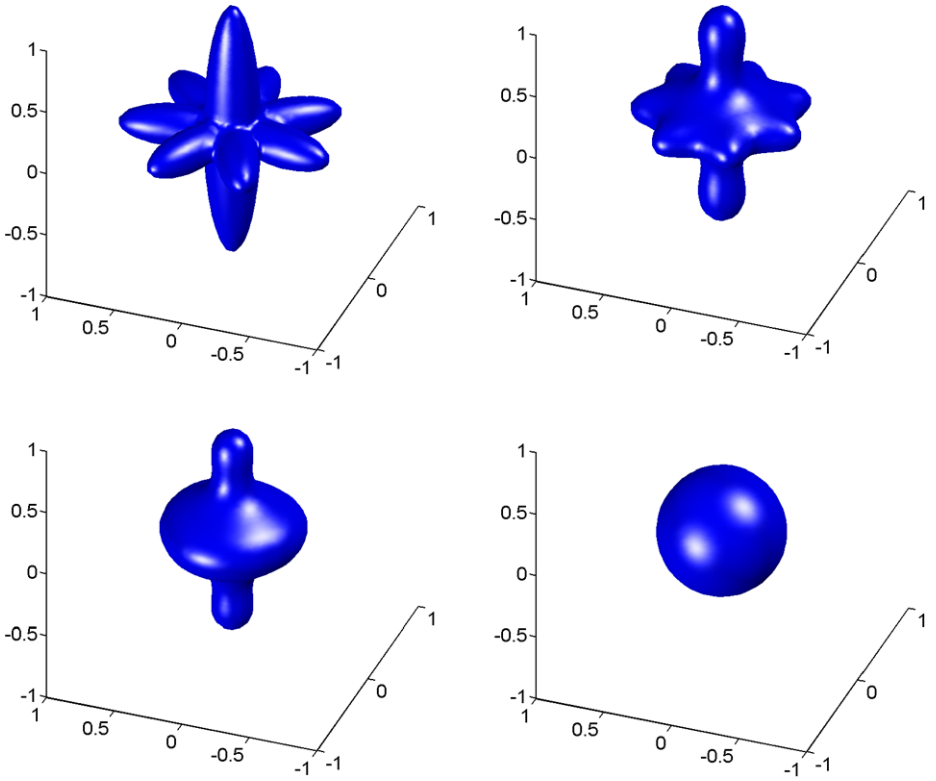


**Fig. 9** Evolution of a star shape in three dimensions. Computation with grid size $64 \times 64 \times 64$, grid spacing $\Delta x = 0.025$ and time step $\Delta t = 5 \times 10^{-5}$. Results shown for time (**a**) 0, (**b**) 0.005, (**c**) 0.01, (**d**) 0.015

Fig. 11c and d. Due to the non-symmetry of the initial configuration these two parts are not equal in mass or shape. They both evolve toward spheres. During the evolution they come into contact and eventually merge into a single sphere.

## 4 Summary

In this paper we have introduced a novel local semi-implicit level set method for surface diffusion and interface motion. This method reduces the computational work by only performing calculations in a small tube around the interface and allowing large time step due to the application of a semi-implicit scheme. By integrating the scheme with the Gradients Stabilized Method and bilaplacian stencil with isotropic discretization error, we are able to use an iterative approach to solve the resulting set of equations. The use of an extraction/insertion approach during evolution allows us to accurately capture complicated interface motion such as coalescence and separation, eliminating spurious interfacial movements seen in the global semi-implicit scheme. Numerical results demonstrate that we can use time steps several orders larger than those in explicit methods. In future work we plan to include multiple energetic driving forces and parallelize the algorithm to allow for simulations of very large systems.

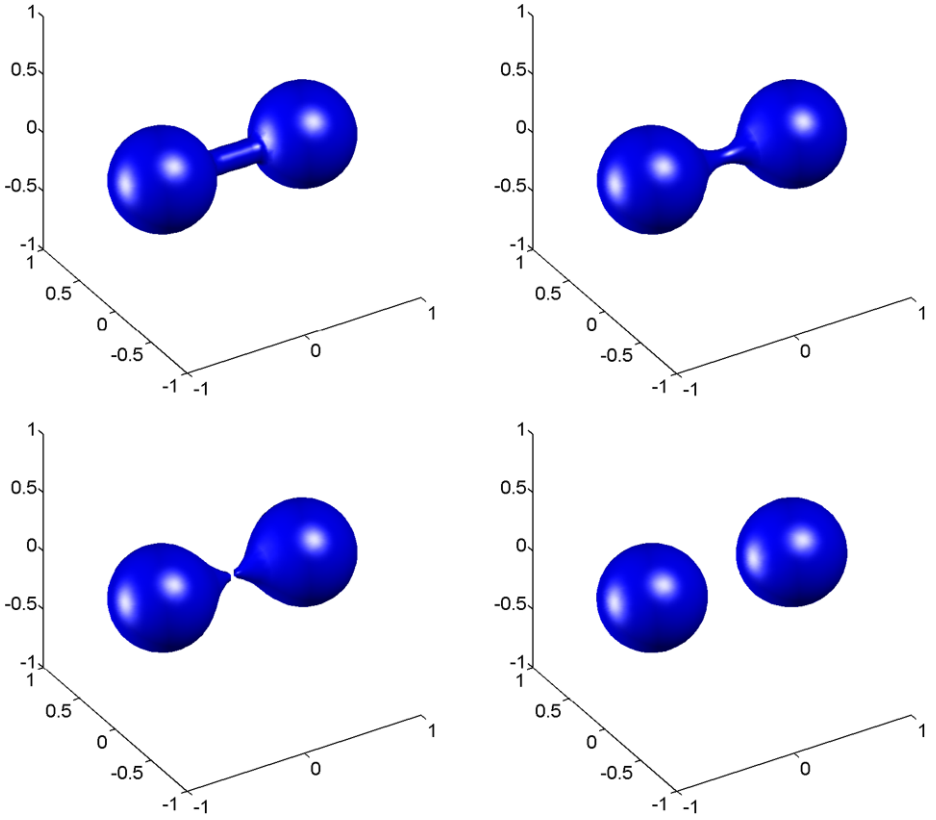

**Fig. 10** Pinching off of a dumbbell-like shape in three dimensions. Computation with grid size $64 \times 64 \times 64$, grid spacing $\Delta x = 0.05$ and time step $\Delta t = 0.0001$. Results shown for time (**a**) 0, (**b**) 0.001, (**c**) 0.0016, (**d**) 0.01

**Acknowledgements** The authors acknowledge financial support from National Science Foundation CAREER Award No. DMI-0348375.

## Appendix

This section will outline some of the basic algorithms needed for this work.

### A. Upwind Derivatives

Upwind schemes were first introduced by Osher and Sethian to investigate Hamilton-Jacobi type equations, of which the level set evolution equation is one example [12]. The concept is to use the direction of information flow (e.g. the interface velocity) to decide the form of the spatial derivative. The method utilized in this work is also known as Godunov's method and can be written as

$$v_n|\nabla\phi| = \max(v_n, 0)\nabla^+ + \min(v_n, 0)\nabla^-, \tag{17}$$

where

$$\nabla^+ = \left[\max(\phi_x^-, 0)^2 + \min(\phi_x^+, 0)^2 + \max(\phi_y^-, 0)^2 + \min(\phi_y^+, 0)^2 + \max(\phi_z^-, 0)^2 + \min(\phi_z^+, 0)^2\right]^{1/2} \tag{18}$$

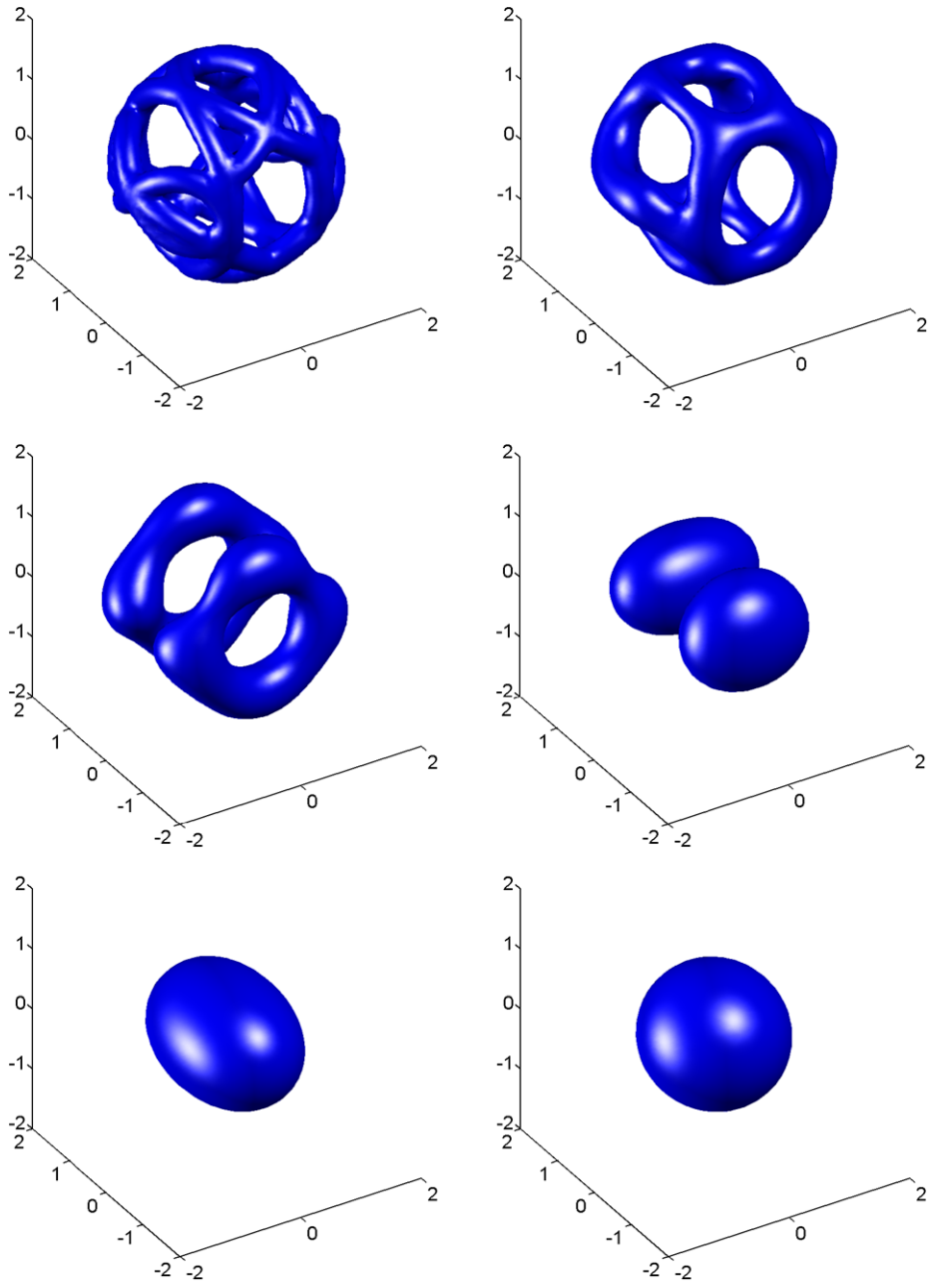


**Fig. 11** The evolution of a complex, non-symmetric body. Computation with grid size $64 \times 64 \times 64$, grid spacing $\Delta x = 0.1$ and time step $\Delta t = 0.0001$. Results shown for time (**a**) 0, (**b**) 0.01, (**c**) 0.05, (**d**) 0.2, (**e**) 0.5, (**f**) 1.0

and

$$\nabla^- = \left[\max(\phi_x^+, 0)^2 + \min(\phi_x^-, 0)^2 + \max(\phi_y^+, 0)^2 + \min(\phi_y^-, 0)^2 + \max(\phi_z^+, 0)^2 + \min(\phi_z^-, 0)^2\right]^{1/2}. \tag{19}$$

Here $\phi_x^-$ denotes a backward difference approximation to the spatial derivative in the $x$-direction and $\phi_x^+$ is the forward difference approximation.

B. Reinitialization Algorithm

Reinitialization of the level set is needed whenever we extract a body to the temporary level set function. We achieve reinitialization by propagating the distance information off of the interface. Imagine a particle moving normal to the interface with a constant speed of 1. The travel time of the particle is equal to its distance from the closest interface point. Mathematically, this can be modeled by solving a differential equation in the form of [20]

$$\frac{\partial \phi}{\partial \tau} + \operatorname{sgn}\left(\phi^0\right)(1 - |\nabla \phi|) = 0, \tag{20}$$

where $\phi^0$ is the original value of the level set function which defines the interface, $\phi$ is the updated level set function, and $\operatorname{sgn}(x)$ is that standard sign function. Here $\tau$ is a pseudo-time. If this equation is solved to the steady-state, the domain of interest is reinitialized to be a signed distance function.

To solve (20) we utilize the upwind derivatives in Appendix A and a WENO3 approximation [5]. While the details of WENO3 spatial approximations are beyond the scope of this article, it is sufficient to say that the scheme chooses a one-sided derivative based on the smoothest possible choice among multiple stencils. To integrate (20) in time we utilize the optimal third-order, four step strong-stability-preserving method [18], with a time step of $\Delta\tau = \Delta x$ where $\Delta x$ is the grid spacing.

C. Extension Algorithm

To evolve the level set it is necessary to extend quantities which are only defined on the interface to nearby regions. We achieve this by solving the following hyperbolic equation, [14]

$$\frac{\partial S}{\partial \tau} + \operatorname{sgn}(\phi)\frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla S = 0, \tag{21}$$

for a predetermined amount of time, where $S$ is the quantity to be extended. Here $\tau$ is a pseudo-time and $\operatorname{sgn}(\phi)$ is the standard sign function.

This particular form of extension is a specific case of a general Hamilton-Jacobi equation and thus we can utilize upwind schemes to solve the extension equation. In general, numerical accuracy is not an overriding factor, as long as $S$ remains unchanged at the zero level-set and is extended off the interface in a sensible and controlled manner. Thus we utilize a first-order upwind scheme with a discrete forward Euler time step [17].

For all the grid points that we wish to extend to we define a mollified sign function in place of the exact form,

$$\operatorname{sgn}(\phi) = \begin{cases} -1 & \phi < -\psi, \\ \frac{\phi}{\psi} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\psi}\right) & |\phi| \le \psi, \\ 1 & \phi > \psi, \end{cases} \tag{22}$$

where $\psi$ is a finite thickness on the order of the grid spacing. Values of $\nabla\phi/|\nabla\phi|$ are calculated using standard center differences while those for $\nabla S$ are calculated with first-order upwind derivatives.

## References


1. Adalsteinsson, D., Sethian, J.A.: The fast construction of extension velocities in level set methods. J. Comput. Phys. **148**, 2–22 (1999)
2. Chen, S., Merriman, B., Kang, M., Caflisch, R.E., Ratsch, C., Cheng, L.T., Gyure, M., Fedkiw, R.P., Anderson, C., Osher, S.: A level set method for thin film epitaxial growth. J. Comput. Phys. **167**, 475–500 (2001)
3. Chopp, D.L., Sethian, J.A.: Motion by intrinsic Laplacian of curvature. Interfaces Free Bound. **1**, 1–18 (1999)
4. Frigo, M., Johnson, S.G.: The design and implementation of FFTW3. Proc. IEEE **93**, 216–231 (2005)
5. Jiang, G.S., Peng, D.P.: Weighted ENO schemes for Hamilton-Jacobi equations. SIAM J. Sci. Comput. **21**, 2126–2143 (2000)
6. Khenner, M., Averbuch, A., Israeli, M., Nathan, M.: Numerical simulation of grain-boundary grooving by level set method. J. Comput. Phys. **170**, 764–784 (2001)
7. Li, Z.L., Zhao, H.K., Gao, H.J.: A numerical study of electro-migration voiding by evolving level set functions on a fixed Cartesian grid. J. Comput. Phys. **152**, 281–304 (1999)
8. Lu, W., Salac, D.: Programmable nanoscale domain patterns in multilayers. Acta Mater. **53**, 3253–3260 (2005)
9. Macklin, P., Lowengrub, J.: An improved geometry-aware curvature discretization for level set methods: Application to tumor growth. J. Comput. Phys. **215**, 392–401 (2006)
10. Mazor, A., Srolovitz, D.J., Hagan, P.S., Bukiet, B.G.: Columnar growth in thin-films. Phys. Rev. Lett. **60**, 424–427 (1988)
11. Mullins, W.W.: Theory of thermal grooving. J. Appl. Phys. **28**, 333–339 (1957)
12. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed—algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**, 12–49 (1988)
13. Patra, M., Karttunen, M.: Stencils with isotropic discretization error for differential operators. Numer. Methods Partial Differ. Equ. **22**, 936–953 (2006)
14. Peng, D.P., Merriman, B., Osher, S., Zhao, H.K., Kang, M.J.: A PDE-based fast local level set method. J. Comput. Phys. **155**, 410–438 (1999)
15. Salac, D., Lu, W.: Design nanocrack patterns in heterogeneous films. Nanotechnology **17**, 5185–5191 (2006)
16. Sethian, J.A., Smereka, P.: Level set methods for fluid interfaces. Annu. Rev. Fluid Mech. **35**, 341–372 (2003)
17. Smereka, P.: Semi-implicit level set methods for curvature and surface diffusion motion. J. Sci. Comput. **19**, 439–456 (2003)
18. Spiteri, R.J., Ruuth, S.J.: A new class of optimal high-order strong-stability-preserving time discretization methods. SIAM J. Numer. Anal. **40**, 469–491 (2002)
19. Suo, Z., Hong, W.: Programmable motion and patterning of molecules on solid surfaces. Proc. Natl. Acad. Sci. U.S.A. **101**, 7874–7879 (2004)
20. Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible 2-phase flow. J. Comput. Phys. **114**, 146–159 (1994)
21. van der Vorst, H.A.: Bi-Cgstab—a fast and smoothly converging variant of Bi-Cg for the solution of nonsymmetric linear-systems. SIAM J. Sci. Statist. Comput. **13**, 631–644 (1992)
22. Yap, Y.F., Chai, J.C., Wong, T.N., Toh, K.C., Zhang, H.Y.: A global mass correction scheme for the level-set method. Numer. Heat Transf. B Fundam. **50**, 455–472 (2006)