Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 220 (2007) 626–653

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Sharp interface tracking using the phase-field equation

Y. Sun, C. Beckermann *

Department of Mechanical and Industrial Engineering, The University of Iowa, College of Engineering, 2412 Seamans Center, Iowa City, IA 52242, USA

Received 27 June 2005; received in revised form 19 May 2006; accepted 21 May 2006
Available online 10 July 2006

Abstract

A general interface tracking method based on the phase-field equation is presented. The zero phase-field contour is used to implicitly track the sharp interface on a fixed grid. The phase-field propagation equation is derived from an interface advection equation by expressing the interface normal and curvature in terms of a hyperbolic tangent phase-field profile across the interface. In addition to normal interface motion driven by a given interface speed or by interface curvature, interface advection by an arbitrary external velocity field is also considered. In the absence of curvature-driven interface motion, a previously developed counter term is used in the phase-field equation to cancel out such motion. Various modifications of the phase-field equation, including nonlinear preconditioning, are also investigated. The accuracy of the present method is demonstrated in several numerical examples for a variety of interface motions and shapes that include singularities, such as sharp corners and topology changes. Good convergence with respect to the grid spacing is obtained. Mass conservation is achieved without the use of separate re-initialization schemes or Lagrangian marker particles. Similarities with and differences to other interface tracking approaches are emphasized.

MSC: 65C20; 76M20; 80A22

Keywords: Phase-field method; Interface tracking; Curvature; Interfacial flows

1. Introduction

Numerical tracking of interface motions has been a major area of interest in computational physics over the past two decades. Applications include the simulation of multiphase flows with and without phase change, solidification and melting, solid-state transformations and other multi-material problems. Sometimes, a moving grid is employed where the interface is a boundary between two sub-domains of the mesh [1]. Most interface tracking methods, however, use a fixed numerical grid. The most popular Eulerian techniques are the level-set [2–4] and volume-of-fluid (VOF) ([5] and references therein) methods. The key ideas behind the level-set method are the Hamilton–Jacobi (HJ) algorithm for solving the advection equation for a signed distance function to the

---

* Corresponding author. Tel.: +1 319 335 5681; fax: +1 319 335 5669.
E-mail address: becker@engineering.uiowa.edu (C. Beckermann).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.05.025

interface, i.e., the level-set function, and the re-intialization scheme for reconstructing the level-set function as the interface evolves [4]. A high-order essentially non-oscillatory (ENO) scheme was developed [6–8] to provide accurate approximations to the numerical fluxes in the HJ algorithm. One shortcoming of the level-set method can be an inadvertent loss of mass. Attempts to improve mass conservation and accuracy have led to different re-initialization schemes [4]. The main advantage of the VOF method is that it conserves mass accurately even for a coarse numerical grid. One challenge in the VOF method is the calculation of the interface curvature from volume fractions. Elaborate algorithms have been developed to overcome this difficulty [9,10]. Lagrangian markers/particles, which are essential in front tracking approaches [11], are sometimes also used in conjunction with the VOF [12] and level-set [13] methods to improve numerical accuracy.

Diffuse interface methods have become popular tools for physical modeling of multiphase systems with and without flow [14]. Among them, the phase-field method has emerged as a widely used technique to numerically simulate complex interfacial pattern formation processes [15]. Diffuse interface models are built on the notion that the interface between the phases is not a sharp boundary, but has a finite width and is characterized by rapid but smooth transitions in the density, viscosity and other physical quantities. In phase-field models, a non-conserved order parameter, the phase field ϕ, is introduced to describe the phase transition. It has constant values in the bulk phases (e.g., in this paper, $\phi = 1$ in one bulk phase and $\phi = -1$ in the other) and varies smoothly across the diffuse interface region ($-1 < \phi < 1$) in a hyperbolic tangent or similar fashion. The propagation equation for the phase field (i.e., the phase-field equation) and the relevant conservation equations are derived from thermodynamically consistent theories of continuum phase transitions that account for the gradient energy across the diffuse interface. The most appealing feature of the phase-field method is that all governing equations can be solved over the entire computational domain without any *a priori* knowledge of the location of the interfaces. Interface tracking is completely avoided and topology changes are handled naturally without the need for any special procedures. Even though interface normals and curvatures are not explicitly evaluated, the phase-field method is especially well suited for problems in which the interface motion depends on gradients of an external field normal to the interface and on the local curvature of the interface.

The objective of this study is to develop a general method for tracking sharp interfaces that is based on the numerical solution of a phase-field-like propagation equation. The resulting method should be viewed in the same spirit as the level-set method. Instead of avoiding the tracking of an interface, as in the traditional use of the phase-field method, it is solely used for that purpose. The interface is no longer viewed as being diffuse and no reference is made to thermodynamics of continuum phase transitions. However, the basic structure of the propagation equation for the phase field is preserved. The $\phi = 0$ contour is used to locate the "sharp" interface. The phase field ϕ still varies in a hyperbolic tangent fashion normal to the interface over a thin, but numerically resolvable region. Interface normals and curvatures can thus be easily calculated from the phase field, but this is generally not necessary for the solution of the phase-field equation (unless the counter term described below is needed to cancel back out curvature effects). The present method for interface tracking is valid for completely general interface motions, including those that do not involve phase change, curvature-driven motion, or any physics at all. When solving a physical problem, it does not need to (but can) be coupled to the continuum conservation equations that are used in the phase-field method [14–16] and other diffuse interface approaches (e.g., [17,18]). The present phase-field equation can also be solved in conjunction with sharp interface formulations of the conservation equations, as is sometimes done when using the level-set method for interface tracking [19–21].

In the traditional use of the phase-field method, the phase field is an order parameter that is not conserved. Furthermore, in the presence of interface curvature the phase-field equation always accounts for curvature-driven motion, which is not often recognized. In order to use the phase-field equation as a more general interface tracking equation, it thus becomes necessary to modify it to allow for cases where the phase field must be conserved (as in immiscible, incompressible two-phase flow) and where the interface motion is not driven by curvature. For such cases, Folch et al. [22] introduced a so-called "counter term" in the phase-field equation to cancel the curvature effect at the leading order, so that the phase field becomes non-relaxational and strictly conserved for a divergence-free flow field. Then, the usual hyperbolic tangent ϕ profile normal to the interface is still obtained as the solution of the phase-field equation, but it is simply advected by the interface motion and not modified by curvature effects. This method is adopted in the present study for cases that do not involve curvature-driven interface motion. The only other studies identified in the literature where the counter term is used are by Biben, Misbah and coworkers [23–25]. None of these previous studies are directly concerned with tracking of sharp interfaces. A counter term of a similar nature was introduced in a phase-field-like method for interface tracking that was developed previously by one of the present authors and coworkers [26].

In addition to introducing an interface tracking technique based on the phase-field equation, the present study also serves to clarify the differences and similarities between the phase-field method and other approaches, in particular the level-set method. This is important in view of the fact that the phase-field method is often obscured by complex thermodynamic derivations [15] (see also Appendix B) and is plagued by difficulties associated with the coupling of the phase-field equation to other continuum equations (as exemplified by the issues surrounding the so-called thin-interface limit [27]; see also Appendix A). By decoupling the phase-field equation from the other equations, its merits for pure interface tracking can be better assessed. For example, the numerical issues associated with the inclusion of arbitrary interface motion driven by an external velocity field can be investigated in detail. While several studies have considered interfacial flows using phase-field methods [22–25,28–33], they have generally not evaluated the numerical issues associated with the discretization of the advection term in the phase-field equation or the extension to conserved phase fields. It is shown below that the same ENO scheme as used in the level-set method can be successfully employed to discretize the advection fluxes. However, since the phase-field equation naturally maintains a hyperbolic tangent ϕ profile of constant thickness normal to the interface during motion, no separate re-initialization equation, as for the signed distance function in the level-set method, needs to be solved. Very recently, Olsson and Kreiss [18] developed a conservative level-set method for two-phase flow that utilizes a phase-field-like (sinusoidal) ϕ profile of constant thickness normal to the interface and, hence, has some similarity with the present method. However, that method relies on a two-step advection/artificial compression procedure that is quite different from the phase-field approach.

The use of the phase-field equation for general interface tracking problems is demonstrated in the present study for a variety of interface motions, including a constant normal interface speed, curvature-driven motion and passive advection by arbitrary external flow fields. Classic numerical tests, such as the propagation of a cosine curve with a constant interface speed, collapse of dumbbells under mean curvature, diagonal translation of a circle, rotation of Zalesak's slotted disk and deformation of a circle with a single vortex, are used to examine the present method for interface tracking. Many of these test cases involve interface topology changes (e.g., pinch-offs) or singularities (e.g., sharp corners). Detailed numerical studies are performed to quantify the accuracy of the phase-field method in capturing the exact interface location, conservation of mass and the convergence rate with respect to grid spacing and other model parameters.

In Section 2, the present phase-field equation for sharp interface tracking is derived. The ability of the phase-field function to represent a stationary sharp interface and approximate the curvature of an interface is investigated in Section 3. In Section 4, the method is tested for the case of a constant normal interface speed. Numerical tests for curvature-driven interface motion and interface motion due to external flow fields are presented in Sections 5 and 6, respectively. Comparisons are provided between the present phase-field model and other interface tracking techniques. The conclusions are summarized in Section 7. Appendix A provides an example of how explicit expressions for the interface velocity can be obtained for a certain physical problem. Appendix B shows that the present equation for sharp interface tracking is indeed equivalent to a thermodynamically derived phase-field equation. Appendix C summarizes the numerical methods and error measures used in this study.

2. Model equations

The present derivation of the phase-field equation starts from the same general interface advection equation as used in other Eulerian techniques for interface tracking, i.e.,

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0 \tag{1}$$

where ϕ is the phase field, $\boldsymbol{u}$ is the velocity of the interface and t is time. In the same manner as in the level-set literature [3,4], the velocity is decomposed into several parts. First, the interface velocity is split into a normal interface speed, u_{n} and an interface velocity due to external advection, $\boldsymbol{u}_{\mathrm{e}}$, as

$$\boldsymbol{u} = u_{\mathrm{n}}\boldsymbol{n} + \boldsymbol{u}_{\mathrm{e}} \tag{2}$$

where $\boldsymbol{n} = \nabla\phi/|\nabla\phi|$ is the unit vector normal to the interface. Eq. (1) can hence be rewritten as

$$\frac{\partial\phi}{\partial t} + u_{\mathrm{n}}|\nabla\phi| + \boldsymbol{u}_{\mathrm{e}}\cdot\nabla\phi = 0 \tag{3}$$

Second, the normal interface speed, u_{n}, is further decomposed into parts that are independent of and proportional to the interface curvature, κ, i.e.,

$$u_{\mathrm{n}} = a - b\kappa \tag{4}$$

where the (variable) coefficients a and b have units of m/s and m^2/s, respectively. The coefficient b must generally be positive [3,4]. Appendix A provides an example of how these two coefficients can be obtained for a physical problem that is governed by a relatively complex set of interface conditions. Substituting Eq. (4) into Eq. (3) results in

$$\frac{\partial\phi}{\partial t} + a|\nabla\phi| + \boldsymbol{u}_{\mathrm{e}}\cdot\nabla\phi = b\kappa|\nabla\phi| \tag{5}$$

The interface curvature can be expressed as a function of the phase field via

$$\kappa = \nabla\cdot\boldsymbol{n} = \nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) = \frac{1}{|\nabla\phi|}\left[\nabla^2\phi - \frac{(\nabla\phi\cdot\nabla)|\nabla\phi|}{|\nabla\phi|}\right] \tag{6}$$

As in Beckermann et al. [16], the following kernel function for the variation of ϕ normal to the interface is now introduced

$$\phi = -\tanh\left(\frac{n}{\sqrt{2}W}\right) \tag{7}$$

where W is a measure of the width of the hyperbolic tangent profile (i.e., ϕ varies from -0.9 to 0.9 over $3\sqrt{2}W$) and n is the coordinate normal to the interface. Eq. (7) is motivated by the equilibrium ϕ profile obtained in thermodynamically derived phase-field models [15]. For systems out of equilibrium, Eq. (7) represents the leading order solution for ϕ. Other kernel functions are possible (e.g., a sinusoidal function [18,34]), but are not investigated further here. Using Eq. (7), the normal derivatives of ϕ can be expressed as

$$|\nabla\phi| = -\frac{\partial\phi}{\partial n} = \frac{1-\phi^2}{\sqrt{2}W} \quad \text{and} \quad \frac{(\nabla\phi\cdot\nabla)|\nabla\phi|}{|\nabla\phi|} = \frac{\partial^2\phi}{\partial n^2} = -\frac{\phi(1-\phi^2)}{W^2} \tag{8}$$

Substituting the second relation in Eq. (8) into Eq. (6) leads to the following expression for the curvature term:

$$\kappa = \frac{1}{|\nabla\phi|}\left[\nabla^2\phi + \frac{\phi(1-\phi^2)}{W^2}\right] \tag{9}$$

Substituting Eq. (9) into Eq. (5) yields

$$\frac{\partial\phi}{\partial t} + a|\nabla\phi| + \boldsymbol{u}_{\mathrm{e}}\cdot\nabla\phi = b\left[\nabla^2\phi + \frac{\phi(1-\phi^2)}{W^2}\right] \tag{10}$$

Eq. (10) is one of the versions of the phase-field equation for interface tracking investigated in this study. It includes normal interface motion, curvature-driven motion and external advection. Note that Eq. (10) is a parabolic-type partial differential equation. The limit of vanishing curvature-driven interface motion, when $b = 0$ and Eq. (10) becomes hyperbolic, is treated below. Another version can be obtained by substituting the first relation in Eq. (8) for $|\nabla\phi|$ into Eq. (10), resulting in

$$\frac{\partial\phi}{\partial t} + a\frac{1-\phi^2}{\sqrt{2}W} + \boldsymbol{u}_{\mathrm{e}}\cdot\nabla\phi = b\left[\nabla^2\phi + \frac{\phi(1-\phi^2)}{W^2}\right] \tag{11}$$

In Eq. (11), the hyperbolic term $a|\nabla\phi|$ is converted into a nonlinear term in ϕ. Hence, Eq. (11) may be easier to implement numerically than Eq. (10). More importantly, it is shown in Appendix B (for $\boldsymbol{u}_{\mathrm{e}} = 0$) that, with the substitution for $|\nabla\phi|$, Eq. (11) is equivalent to certain thermodynamically derived phase-field equations. A comparison of the use of Eqs. (10) and (11) for $\boldsymbol{u}_{\mathrm{e}} = 0$ is provided in Section 4.

It should be noted that the special form of the right-hand side of Eq. (10) or (11) is the primary reason for the unique nature of the phase-field method. As demonstrated in Section 3, for a stationary interface (when the left-hand side vanishes) solving $\nabla^2\phi + \phi(1-\phi^2)/W^2 = 0$ yields the hyperbolic tangent profile, Eq. (7), of width $\sim W$ across the interface. This is true even in the absence of interface curvature. If instead the expression for the curvature given by Eq. (6) were directly substituted for κ in Eq. (5), the ϕ profile would be undetermined. Clearly, the width parameter W plays an important role in the phase-field equation. In the context of the present study, W should be viewed as a purely numerical parameter and the method used to choose a suitable W is explained in Section 3. Also note that the $\nabla^2\phi$ term on the right-hand side of Eq. (10) or (11) will serve to smooth singularities in the interface shape; this is demonstrated in Section 4.

The case of no curvature-driven interface motion is treated using the counter term approach originally introduced by Folch et al. [22]. The curvature term on the right-hand side of Eq. (10) is subtracted back out as follows:

$$\frac{\partial\phi}{\partial t} + a|\nabla\phi| + \boldsymbol{u}_\mathrm{e}\cdot\nabla\phi = b\left[\nabla^2\phi + \frac{\phi(1-\phi^2)}{W^2} - \kappa|\nabla\phi|\right] = b\left[\nabla^2\phi + \frac{\phi(1-\phi^2)}{W^2} - |\nabla\phi|\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right] \tag{12}$$

Note that in view of Eq. (9), the right-hand side is nothing but $b(\kappa|\nabla\phi| - \kappa|\nabla\phi|)$. However, Eq. (12) is different from simply setting $b = 0$. This can be best understood by considering the case of a flat interface where $\kappa = 0$. In that case, the last term in Eq. (12), i.e., $\kappa|\nabla\phi|$, vanishes and the remaining terms on the right-hand side (i.e., $\nabla^2\phi + \phi(1-\phi^2)/W^2$) yield the hyperbolic tangent ϕ profile as the stationary solution of Eq. (12). If instead the coefficient b was set to zero, Eq. (12) would become a hyperbolic equation that advects *any* function ϕ with the interface velocity $\boldsymbol{u} = a\boldsymbol{n} + \boldsymbol{u}_\mathrm{e}$. The interplay of the left- and right-hand sides of Eq. (12), however, relaxes an arbitrary initial phase field to a hyperbolic tangent profile across the interface and then sustains this profile during interface motion. While the use of the counter term $\kappa|\nabla\phi|$ in the absence of curvature-driven interface motion may seem awkward at first, it is shown below that it provides accurate results. The reader is referred to Folch et al. [22] for a rigorous mathematical analysis of the counter term in the phase-field equation. They show that for a finite interface width, it cancels out the original curvature term (i.e., $\nabla^2\phi + \phi(1-\phi^2)/W^2$) at the leading order. It is important to note that in the absence of curvature-driven interface motion, the coefficient b no longer represents any physical quantities. In Eq. (12), b is a purely numerical parameter. It controls both the relaxation behavior of the ϕ profile and the smoothing of interface singularities through the dissipative nature of the $\nabla^2\phi$ term in multiple dimensions. Numerical results that show the effect of different choices of the coefficient b on the accuracy of the present interface tracking method, including mass conservation, and a method for choosing an optimum b are presented in subsequent sections.

A direct connection to the level-set method can be made by rewriting Eq. (12) in terms of a signed distance function to the interface, $\psi(=n)$, given by $\phi = -\tanh(\psi/\sqrt{2}W)$. Such a substitution is equivalent to the nonlinear preconditioning technique for the phase-field method introduced by Glasner [35]. Then, Eq. (12) becomes

$$\frac{\partial\psi}{\partial t} + a|\nabla\psi| + \boldsymbol{u}_\mathrm{e}\cdot\nabla\psi = b\left[\nabla^2\psi + \frac{1}{W}(1-|\nabla\psi|^2)\sqrt{2}\tanh\left(\frac{\psi}{\sqrt{2}W}\right) - |\nabla\psi|\nabla\cdot\left(\frac{\nabla\psi}{|\nabla\psi|}\right)\right] \tag{13}$$

Now it can be seen that the only difference to the level-set method is the right-hand side of Eq. (13). The right-hand side serves to maintain and, at singularities, smooth the signed distance function during interface motion. With respect to the former purpose, the right-hand side can be thought of as an integrated re-initialization scheme for the ψ field [35]. The advantage of using preconditioning is that higher order derivatives of ψ are independent of W, and therefore one can expect smaller discretization errors than with using Eq. (12). This is demonstrated as part of the numerical tests in subsequent sections. Note that for the signed distance function ψ $(=n)$, $|\nabla\psi| = \partial\psi/\partial n = 1$. A comparison of using $|\nabla\psi| = 1$ in the second term on the left-hand side of Eq. (13) versus discretizing $|\nabla\psi|$ using central differences (see Appendix C) is provided in Section 4.

In the present study, the phase-field equation is solved numerically using a standard explicit finite-difference method together with a fourth-order ENO scheme for the hyperbolic term $\boldsymbol{u}_\mathrm{e}\cdot\nabla\phi$. The details of the numerical implementation are provided in Appendix C. This appendix also discusses the constraints on the time step and the coefficient b in the absence of curvature-driven motion, and explains the error measures utilized in the following numerical tests.

3. Stationary interfaces

The above phase-field equations are first examined for stationary interfaces in order to illustrate how the location of the sharp interface is obtained from the solution of the phase-field equation and how interface curvature is approximated. The effects of grid spacing and nonlinear preconditioning are also investigated.

In the absence of interface motion (i.e., $a = 0$ and $\boldsymbol{u}_\mathrm{e} = 0$) and for a planar interface (i.e., $\kappa = 0$) in one dimension, Eq. (12) reduces to the following dimensionless form:

$$\frac{\partial \phi}{\partial t'} = \frac{\partial^2 \phi}{\partial x'^2} + \phi(1 - \phi^2) \tag{14}$$

where $t' = t/(W^2/b)$ and $x' = x/W$. Numerical results at steady state (i.e., $t' \to \infty$) are shown in Fig. 1(a) for grid spacings of $\Delta x' = 1$ and $\Delta x' = 0.5$. The initial condition is taken to be a step function, i.e., $\phi = 1$ for $x < 0$ and $\phi = -1$ for $x \geqslant 0$. It can be seen that, at steady state, ϕ indeed varies in a hyperbolic tangent fashion from 0.9 to -0.9 for $-1.5\sqrt{2} < x' < 1.5\sqrt{2}$. A measure of the total width of the hyperbolic tangent profile, l_i, is therefore given by $l_i = 3\sqrt{2}W$. For $\Delta x' = 1$, there are four grid points within l_i and the computed phase-field profile shows some deviations from the exact solution at steady state, $\phi^\mathrm{ex} = -\tanh(x'/\sqrt{2})$. For the finer grid spacing of $\Delta x' = 0.5$, there are eight grid points within l_i and the numerical solution overlaps with the exact solution within the thickness of the lines in the figure. However, for this example, the variation of ϕ away from

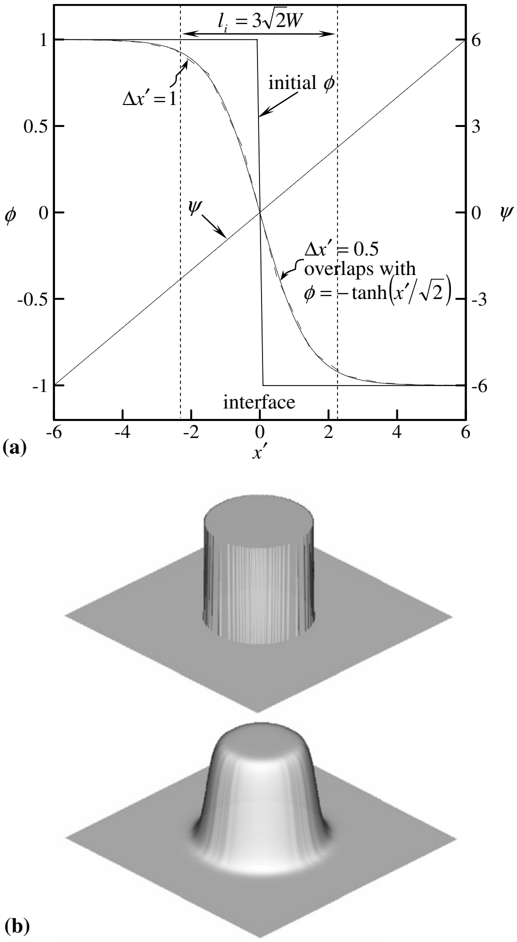


Fig. 1. Phase-field profiles for stationary interfaces: (a) planar interface, and (b) 2-D circle (top panel: initial condition; bottom panel: solution at steady state).

the interface is of little consequence. For both grid spacings, the computed location of the interface (i.e., where $\phi = 0$) is exactly at $x' = 0$, and all error norms are exactly equal to zero. This is always true for the present example as long as an even number of grid points is employed. Fig. 1(a) also shows the calculated variation of the signed distance function, ψ, which is linear as expected. Again, the computed $\psi = 0$ location is exactly at $x' = 0$. Note that ψ continues to change away from the interface, whereas ϕ approaches constant values (i.e., ± 1). The fact that the phase-field function is constant away from the interface considerably simplifies its initialization (i.e., it can be initialized as a step function), compared to the level-set function.

In multiple dimensions, the stationary interface version of Eq. (12) can be written as

$$\frac{\partial \phi}{\partial t'} = \nabla'^2 \phi + \phi(1 - \phi^2) - |\nabla' \phi| \nabla' \cdot \left(\frac{\nabla' \phi}{|\nabla' \phi|} \right) \tag{15}$$

Eq. (15) is solved numerically in two dimensions for a circle of radius $R' = R/W$. Thus, this example illustrates how well the phase-field equation can approximate an interface of a given curvature ($\kappa = 1/R$). The initial condition is a step profile for ϕ, as illustrated in the upper panel of Fig. 1(b), with $\phi = 1$ and $\phi = -1$ inside and outside of the circle, respectively. The computed phase-field distribution at steady state (when ϕ changes by less than 10^{-9}) is shown in the lower panel of Fig. 1(b). The $\phi = 0$ contour (mid-way up the profile) in this figure represents the calculated interface location, and it is compared to the exact circle in the following.

Fig. 2 shows the variation of the calculated error norms L_1, L_2, and L_∞ (see Appendix C) with the dimensionless grid spacing $\Delta x'$ for a circle of radius $R' = 20$. In this simple example, all three error norms show approximately the same behavior. For $\Delta x' < 1$, a convergence rate with respect to $\Delta x'$ of at least fourth order can be observed. For $\Delta x' > 1$, no meaningful results are obtained. With $\Delta x' = 0.5$, corresponding to eight grid points inside l_i, all error norms are about 10^{-4}. Since $\Delta x' = \Delta x/W$, the results in Fig. 2 imply that a certain minimum number of grid points are needed to accurately resolve the hyperbolic tangent ϕ profile across the interface. Due to the rapid convergence rate with respect to $\Delta x'$, a grid spacing between $0.5W$ and $0.25W$ should provide sufficiently accurate results for most practical purposes.

Fig. 3 shows the calculated L_1 norm as a function of the dimensionless radius of the circle, R'. Results are shown for $\Delta x' = 0.5$ and 0.25. If R' is too small (e.g., 2), the results become highly inaccurate because the ϕ profiles normal to the interface overlap inside the circle (see inset). For R' increasing from 2 (corresponding to $R = 4\Delta x$) to about 5 ($R = 10\Delta x$), the error decreases very rapidly. At $R' = 5$, the ϕ profiles no longer overlap inside the circle (see inset) and the error norms start to level off. Hence, the radius of curvature should be at least equal to the width of the hyperbolic tangent ϕ profile, i.e., $R \geqslant l_i = 3\sqrt{2}W = 4.2W$, in order to avoid overlapping of the ϕ profiles. The error norms keep decreasing for $R' > 4.2$, but at a relatively slow rate. Therefore, as long as overlapping is prevented, the error is close to the minimum value that can be expected for a given grid spacing (e.g., $\sim 10^{-4}$ for $\Delta x' = 0.5$). Further improvements in the accuracy are primarily

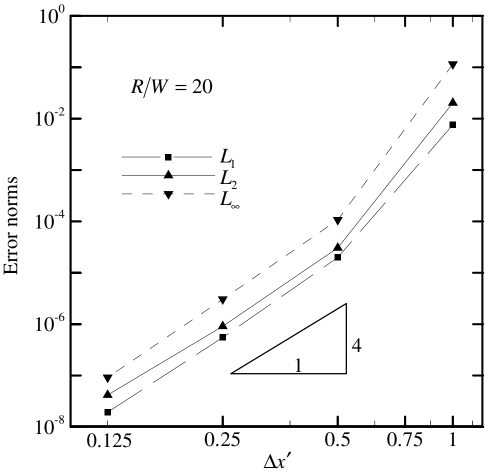


Fig. 2. Calculated error norms as a function of the dimensionless grid spacing for a stationary circle of radius $R/W = 20$.

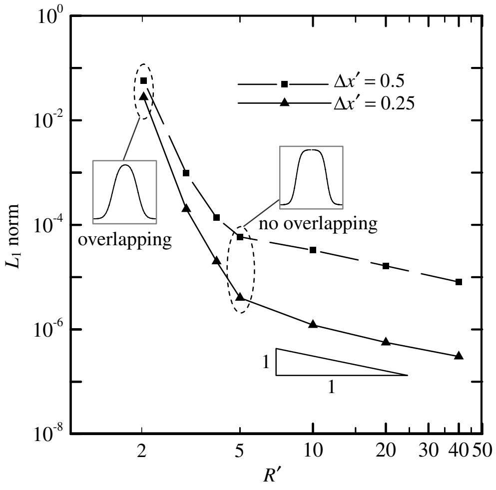


Fig. 3. Calculated L_1 error norm for two grid spacings as a function of the dimensionless radius of a stationary circle.

obtained by decreasing $\Delta x'$. As shown in both Figs. 2 and 3, decreasing $\Delta x'$ from 0.5 to 0.25 results in a decrease in the error for $R' > 4.2$ of almost two orders of magnitude.

The need to specify the width of the ϕ profile across the interface through the parameter W, is an important feature of the phase-field method. In the present context of sharp interface tracking, W is a purely numerical parameter. A similar parameter exists in the level-set method, i.e., the thickness of the region where re-distancing is applied in the re-initialization scheme [36]. The above results allow for the following general conclusions to be drawn regarding the selection of W. According to Fig. 2, the need to accurately resolve the ϕ profile results in the constraint that W needs to be larger than the grid spacing, i.e.,

$$W > \Delta x \tag{16}$$

Due to the fourth-order convergence rate with respect to W, there is generally no need to employ a W that is greater than $2\Delta x$ to $4\Delta x$ (see Fig. 2). Unless otherwise noted, $W = 2\Delta x$ (or $\Delta x' = 0.5$) is used in all subsequent tests. It is important to keep W as small as possible (relative to Δx) in order to avoid overlapping of ϕ profiles for a curved interface, as discussed in connection with Fig. 3. If R is the local radius of curvature of an interface, an upper constraint on W is given by

$$W < R/4.2 \tag{17}$$

which corresponds to $R' > 4.2$ in Fig. 3. The constraints given by Eqs. (16) and (17) combine into the requirement that the grid spacing Δx should be less than about $0.2R$, i.e., about five grid points are needed to accurately resolve a radius of curvature. This is not unreasonable for any interface tracking method.

Before proceeding, it is useful to examine the effect of preconditioning on the choice of W. The preconditioned version of Eq. (15) can be written as

$$\frac{\partial\psi'}{\partial t'} = \nabla'^2\psi' + (1 - |\nabla'\psi'|^2)\sqrt{2}\tanh\left(\psi'/\sqrt{2}\right) - |\nabla'\psi'|\nabla'\cdot\left(\frac{\nabla'\psi'}{|\nabla\psi'|}\right) \tag{18}$$

where $\psi' = \psi/W$. For the stationary circle example from above, Fig. 4 shows a comparison of the L_1 error norm obtained with and without the use of nonlinear preconditioning, i.e., Eq. (18) versus Eq. (15), as a function of the dimensionless grid spacing. With preconditioning, the results converge at a fourth-order rate up to at least $\Delta x' = 4$. Recall that without preconditioning, no meaningful solutions are obtained for $\Delta x' > 1$. However, the results with preconditioning become highly inaccurate for $\Delta x' > 2$ (the error norm is greater than 10^{-2}), and a dimensionless grid spacing much greater than unity cannot be recommended. For $\Delta x' = 1$ an improvement in the accuracy by about a factor of five is obtained due to preconditioning. This constitutes the main advantage of preconditioning and is consistent with the findings in Glasner [35] and Ramirez and Beckermann [37]. Note from Fig. 4, however, that for $\Delta x' \leqslant 0.5$ the differences in the accuracy are negligibly small and preconditioning

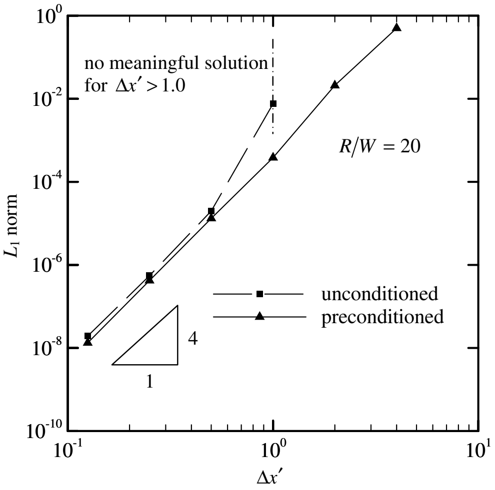


Fig. 4. Comparison of the calculated L_1 error norm with and without nonlinear preconditioning, as a function of the dimensionless grid spacing for a stationary circle.

offers no advantage. Nonetheless, a more robust behavior of the method for large dimensionless grid spacings can be of sufficient advantage in certain applications to justify the use of preconditioning.

4. Interface motion with a constant normal speed

For an interface moving exclusively with a constant normal interface speed, i.e., $u_n = a = \text{const.}$, $b = 0$ and $\boldsymbol{u}_e = 0$, Eq. (12) reduces to the following dimensionless form:

$$\frac{\partial\phi}{\partial t'} + |\nabla'\phi| = b'\left[\nabla'^2\phi + \phi(1-\phi^2) - |\nabla'\phi|\nabla'\cdot\left(\frac{\nabla'\phi}{|\nabla'\phi|}\right)\right] \tag{19}$$

where $\nabla' = \nabla/W$, $t' = t/(W/a)$, and $b' = b/(Wa)$. Since curvature-driven interface motion is not considered, the coefficient b' is a purely numerical parameter that controls the relaxation behavior of the phase-field profile and the smoothing of interface singularities (see Section 2). The advection term in Eq. (19), i.e., $|\nabla'\phi|$, is discretized using the central difference scheme (see Appendix C). Following the discussion leading to Eq. (11), the $|\nabla'\phi|$ term can also be evaluated as $|\nabla'\phi| = (1-\phi^2)/\sqrt{2}$; then, Eq. (19) becomes

$$\frac{\partial\phi}{\partial t'} + \frac{1-\phi^2}{\sqrt{2}} = b'\left[\nabla'^2\phi + \phi(1-\phi^2) - |\nabla'\phi|\nabla'\cdot\left(\frac{\nabla'\phi}{|\nabla'\phi|}\right)\right] \tag{20}$$

The interface tracking equations given by Eqs. (19) and (20) are tested and compared in the following for the classic cosine curve propagation problem suggested by [38]. Preconditioned versions of these equations are also examined.

Consider an interface given by the periodic initial cosine curve [3]

$$\gamma(0) = [1-s, (1+\cos 2\pi s)/4] \tag{21}$$

propagating with a normal speed of unity (i.e., $u_n = a = 1$). In Eq. (21), $\gamma(t)$ denotes the curve at different times t and $0 \leqslant s \leqslant 1$. As shown in Fig. 5, the interface soon develops a sharp corner at the center. Once this corner develops, the normal is ambiguously defined. One possible solution is that the curve passes through itself generating a double-valued swallowtail, as shown in Fig. 5(a). Fig. 5(b) illustrates an alternative weak solution that can be obtained through a Huygen's principle construction [38]. This construction represents the physically meaningful solution if the curve is an interface separating two phases. As shown in Fig. 5(b), the wave front always corresponds to the "first arrivals", and the "tail" present in Fig. 5(a) is removed. The Huygen's principle construction is the exact solution used below in evaluating numerical errors. Obviously, the numerical resolution of the sharp corner represents the main challenge in this test case.

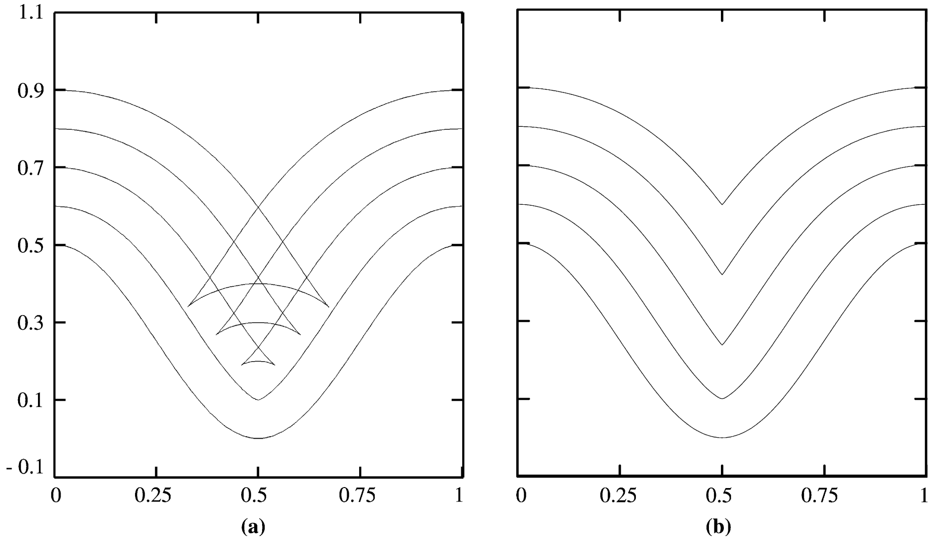
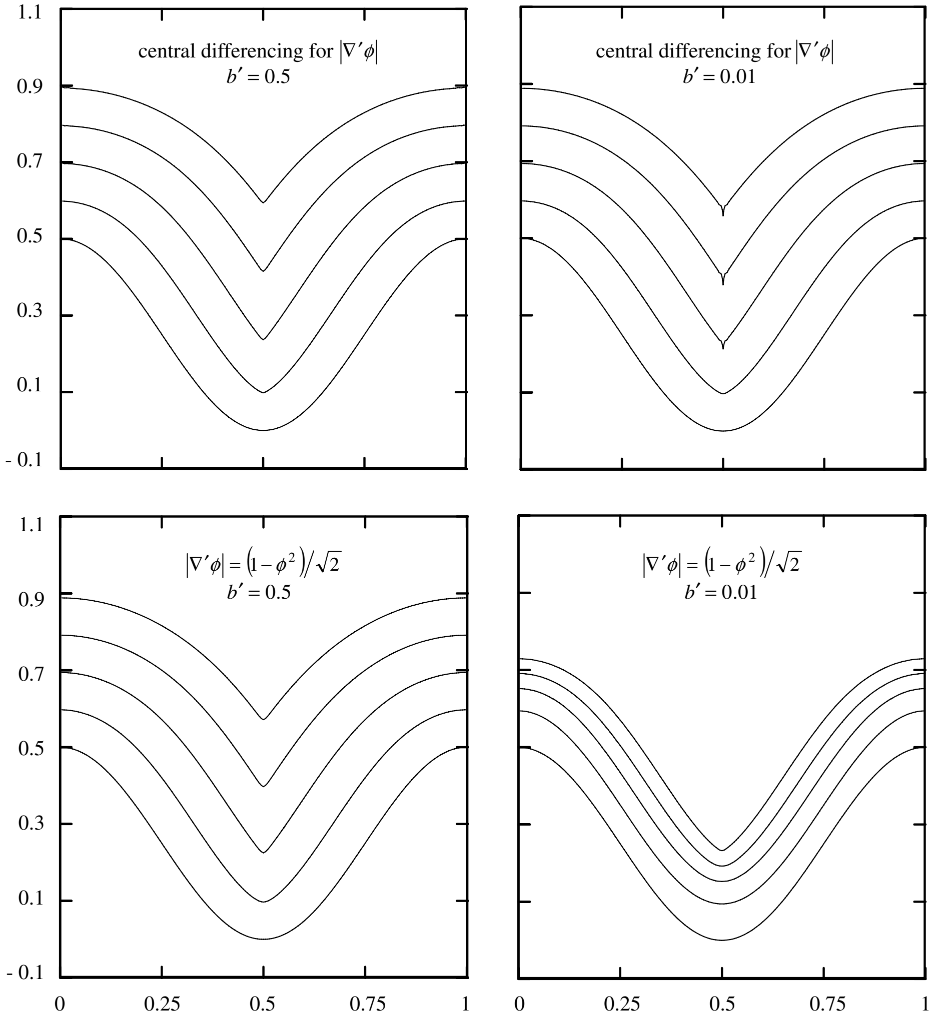


Fig. 5. Analytical solutions for the propagating cosine curve test problem at five different times (bottom curve is for $t = 0$): (a) swallowtail solution, and (b) Huygen's principle construction after Sethian [38].

Numerical solutions of Eqs. (19) and (20) are obtained on a rectangular domain of dimensions $[0, 1]$ in x and $[-0.1, 1.1]$ in y using a uniform mesh of 100×120 grid points. As discussed in Section 3, $\Delta x'$ is chosen equal to 0.5, which fixes the value of W. Numerical tests indicate that a Courant number of $\Delta t'/\Delta x' = 0.1$ is sufficient for the results to be considered converged with respect to the time step. With these choices for $\Delta x'$ and $\Delta t'$, the upper limit on the coefficient b' is given by Eq. (C.5) as $b' < 1.2$.

Fig. 6 shows computed $\phi = 0$ contours at the same times as for the exact solution in Fig. 5. Fig. 6(a) and (b) (left and right sides, respectively) are for $b' = 0.5$ and $b' = 0.01$, respectively, while the upper and lower panels are the results obtained using Eqs. (19) and (20), respectively. Clearly, the results for $b' = 0.5$ are in much better agreement with the exact solution than for $b' = 0.01$. For $b' = 0.5$ (Fig. 6(a)), a slight rounding of the corner can be observed, but otherwise the curve propagates at close to the correct speed. When the advection term is evaluated using $(1 - \phi^2)/\sqrt{2}$, as in Eq. (20), the curve propagates slightly behind the exact solution (lower panel of Fig. 6(a)). This effect is highly amplified for $b' = 0.01$ (lower panel of Fig. 6(b)), where the curve can be seen to travel far behind the exact solution and the corner is severely rounded. For such a low b', the hyperbolic tangent profile across the interface is not well maintained during interface motion; hence, using $(1 - \phi^2)/\sqrt{2}$ to propagate the interface leads to inaccurate local interface velocities. On the other hand, direct discretization of the advection term $|\nabla'\phi|$, as when using Eq. (19), does not produce this deleterious effect for small b'. As can be seen in the upper panel in Fig. 6(b), the interface propagates at approximately the correct speed, even though b' is very small (0.01). However, a well-known numerical instability occurs when discretizing $|\nabla\phi'|$ using the central difference scheme and using a very small b' [3]: the interface starts to develop a wiggle at the location of the corner (upper panel of Fig. 6(b)). For larger b', as in the upper panel of Fig. 6(a), this instability is successfully prevented. In summary, for a sufficiently large coefficient b', the present form of the right-hand side of Eq. (19) or Eq. (20) works well to cancel out curvature-driven interface motion and to suppress instabilities at corners, while maintaining the hyperbolic tangent profile during interface motion. For the same value of b', direct central difference discretization of the advection term $|\nabla'\phi|$ [Eq. (19)] results in more accurate interface propagation than with substituting $(1 - \phi^2)/\sqrt{2}$ for the advection term [Eq. (20)]. For a coefficient b' that is too small or zero, a more complex discretization method for the advection term on the left-hand side would have to be employed (such as the ENO scheme).

Fig. 7 shows the corresponding results obtained with nonlinear preconditioning of Eqs. (19) and (20). For $b' = 0.5$, as shown in Fig. 7(a), the results are somewhat improved relative to those without preconditioning (Fig. 6(a)): the corner that develops at later times is sharper and the interface propagation is more accurate. Comparing the upper and lower panels in Fig. 7(a), with preconditioning the interface velocities are accurately

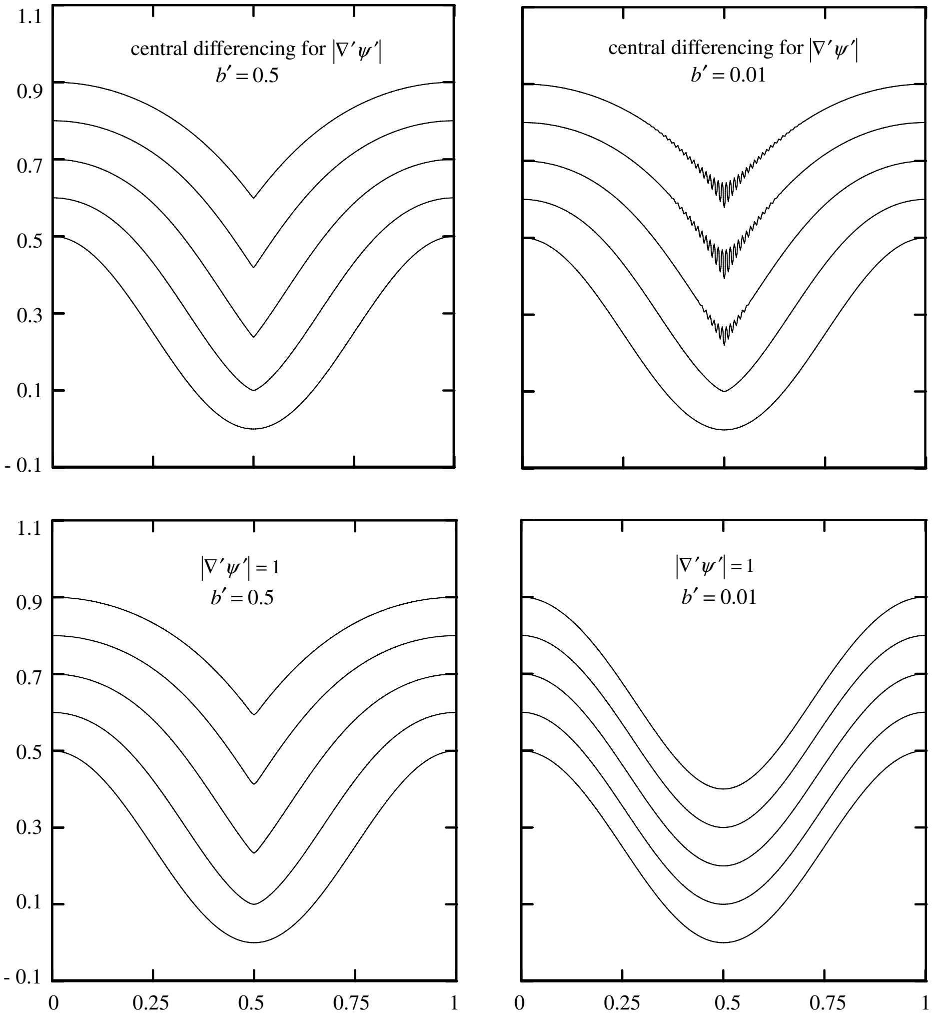


(a) **(b)**

Fig. 6. Calculated $\phi = 0$ contours for the propagating cosine curve test problem without preconditioning and a 100×120 mesh [upper panels: central differencing for $|\nabla'\phi|$, Eq. (19); lower panels: $|\nabla'\phi| = (1-\phi^2)/\sqrt{2}$, Eq. (20)]: (a) $b' = 0.5$, and (b) $b' = 0.01$.

calculated with both central differencing for the advection term $|\nabla'\psi|$ (upper panel) and using $|\nabla'\psi| = 1$ (lower panel); the same is true even for $b' = 0.01$, as can be seen from Fig. 7(b). Hence, preconditioning is of particular advantage when evaluating the norm of ϕ. However, even with preconditioning, the use of a coefficient b' that is too small (i.e., 0.01) introduces large errors at the location of the corner. As shown in the upper panel of Fig. 7(b), for central differencing of the advection term numerous small-scale wiggles appear at the corner. With $|\nabla'\psi| = 1$ (lower panel of Fig. 7(b)), a severe rounding of the corner results from the use of a b' that is too small.

The issue of finding an optimum value for the coefficient b' in the absence of curvature-driven interface motion is further investigated in Fig. 8. In this figure all three error norms (see Appendix C) for the $\phi = 0$ contour at $t' = 10$ are plotted against b'. Only results from calculations based on Eq. (19) are shown; the trends are the same when using Eq. (20) or preconditioning. It can be seen that the error norms decrease at a second-order rate with increasing b'. However, for b' greater than about 0.2 the error norms essentially cease to decrease and, as expected, no stable solution is obtained beyond the upper limit of $b' = 1.2$. The flattening of the decrease in the error norms for $b' > 0.2$ can be explained as follows. Increasing b' generally results in better enforcement of the hyperbolic tangent phase-field profile across the interface and, hence, more accu-



(a) **(b)**

Fig. 7. Calculated $\phi = 0$ contours for the propagating cosine curve test problem with preconditioning and a 100×120 mesh (upper panels: central differencing for $|\nabla'\psi'|$; lower panels: $|\nabla'\psi'| = 1.0$): (a) $b' = 0.5$, and (b) $b' = 0.01$.

rate approximation of the norm and curvature. On the other hand, larger b' enhance any discretization errors and numerical dissipation from the right-hand side of Eq. (19); recall that in the absence of curvature-driven motion, the right-hand side of Eq. (19) should vanish at first order. In practice, Fig. 8 shows that any choice of b' within about one order of magnitude below the upper limit given by the Courant–Friedrichs–Levy (CFL) condition [Eq. (C.5)] yields L_1 and L_2 error norms that are approximately equal to the minimum value that can be expected for $\Delta x' = 0.5$ (i.e., $\approx 10^{-4}$ as shown in Fig. 2). Note that in Fig. 8 the L_∞ error norm is substantially higher than the other two norms. This reflects the increased local error in the computed interface shape near the corner due to wiggles or excessive rounding. Since the L_∞ error norm shows a minimum at approximately $b' = 0.5$, this value for the coefficient b' can be regarded as an optimum value for cases where curvature-driven interface motion is absent.

5. Curvature-driven interface motion

Using the phase-field equation for interface tracking is particularly straightforward when curvature-driven interface motion is present (since the counter term is not needed and b is no longer a numerical parameter).

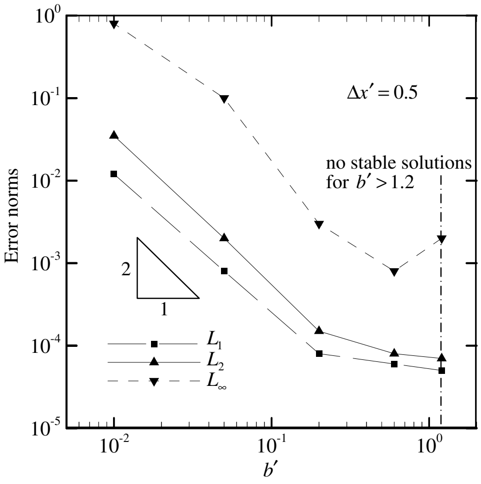


Fig. 8. Calculated error norms at $t' = 10$ as a function of the coefficient b' for the propagating cosine curve test problem [Eq. (19) and a 100×120 mesh].

Focusing on the case where $u_n = -b\kappa$ by setting $a = 0$ and $\boldsymbol{u}_e = 0$ in Eq. (10), the phase-field equation can be written in dimensionless form as

$$\frac{\partial \phi}{\partial t'} = \nabla'^2 \phi + \phi(1 - \phi^2) \tag{22}$$

where $t' = t/(W^2/b)$. Eq. (22) avoids the direct calculation of the interface normal and curvature, as is typical for the phase-field method. Recall from the discussion in Section 2 that the right-hand side of Eq. (22) has a different effect than just modeling curvature-driven interface motion; the right-hand side maintains the hyperbolic tangent ϕ profile even in the absence of curvature. Note that the absence of W from Eq. (22) is solely due to the non-dimensionalization employed; W is still an important numerical parameter in this example (see below). Eq. (22) can be expected to work well in the presence of interface singularities and topology changes because of the dissipative nature of the $\nabla'^2\phi$ term (in multiple dimensions), as already discussed. This is illustrated next using the well-known example of the collapse of three-dimensional dumbbells under curvature (with $b = 1$) after Sethian [39].

Consider the three-dimensional dumbbell shown in Fig. 9(a). It is made up of two spheres, each of radius 3, that are connected by a cylindrical handle of radius 1.5. Using this dumbbell as the initial interface shape, Eq. (22) is solved on a $20 \times 7 \times 7$ domain using, as a base case, a uniform mesh of $200 \times 70 \times 70$ grid points. The parameter W is again chosen according to $\Delta x' = 0.5$. For simplicity, only a 7-point finite-difference stencil is used for the discretization of the $\nabla'^2\phi$ term, instead of the 27-point stencil that would result from an extension of Eq. (C.1) to three dimensions. The computed evolution of the dumbbell in its diagonal cross-section is shown in Fig. 9(b). The interface is plotted every 1000 time steps, until the handle becomes small; then, the interface is shown every 100 time steps. Since the initial shape is only piece-wise continuous, the sharp corners are quickly smoothed out as the surface of the dumbbell moves inward [39]. The handle narrows as the surface shrinks, until it pinches off and the dumbbell separates into two pieces. These two pieces continue to shrink, while acquiring a more spherical shape.

The results of a grid convergence study for this case are shown in Fig. 10. The figure shows the calculated pinch-off time as a function of the number of grid points in the direction parallel to the handle (i.e., in the x direction). The number of grid points in the other directions was adjusted to maintain a uniform, square mesh. Also included in the figure are the results of Sethian [39] who used the level-set method. It can be seen that as the mesh is refined to 250 grid points in the x direction, the pinch-off time converges to about $t = 1.12$. Above 140 grid points, the convergence rates of the phase-field and level-set methods are approximately the same. The accuracy of the phase-field method deteriorates more quickly for grid numbers below 140. This can be attributed to overlapping of ϕ profiles for curved interfaces, as discussed in connection with Fig. 3. In the

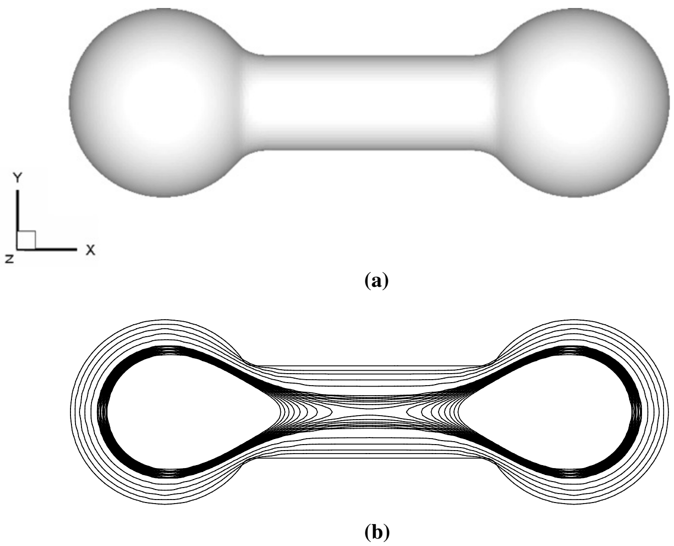


Fig. 9. Evolution of a 3-D dumbbell due to curvature-driven interface motion using a $200 \times 70 \times 70$ mesh: (a) initial shape and (b) cross-section showing interface contours at later times.

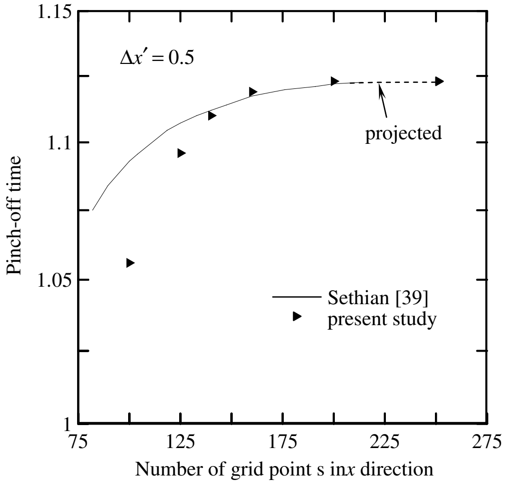


Fig. 10. Calculated pinch-off times as a function of the number of grid points in the x direction for the dumbbell test problem of Fig. 9; the results of Sethian [39] are based on the level-set method.

present convergence study, increasing the grid spacing Δx, while keeping $\Delta x' = \Delta x/W$ constant at 0.5, results in an increasing W. Ultimately, the width of the ϕ profile becomes larger than the local radii of curvature, R, and the condition given by Eq. (17) is violated. This leads to the same kind of rapid increase in the error as observed in Fig. 3 for $R' < 4.2$.

An extension of this example is the collapse of a four-armed dumbbell under curvature, as studied by Chopp and Sethian [40] using the level-set method. The initial interface shape is shown in the upper-left panel of Fig. 11. Eq. (22) is solved using the same conditions as in the previous example but with a mesh consisting of $100 \times 100 \times 35$ grid points. Computed interfaces are shown in Fig. 11 at three times that are close to the pinch-off. As the surface of the dumbbell collapses under its curvature, the four handles pinch-off, leaving a separate closed surface in the center. This "pillow" is formed because the curvature of each handle is larger than that of the saddle joints [40]. Once the pillow becomes separated, it quickly collapses to a point and finally disappears. It can be verified that the results shown in Fig. 11 match very well with those in Fig. 6 of Chopp and Sethian [40].

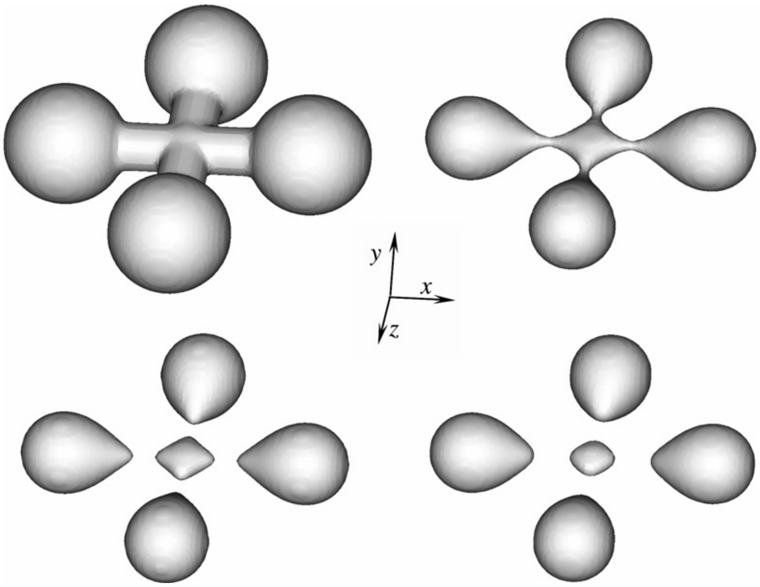


Fig. 11. Calculated collapse of a four-armed dumbbell under curvature using a 100 × 100 × 35 mesh.

The above examples demonstrate that the phase-field method is especially well suited for complex curvature-driven interface motions, including those that involve topology changes. Eq. (22) can be implemented numerically in literally a few lines of code. This simplicity of the phase-field method for curvature-driven interface motion is likely the primary reason for its popularity in phase-change applications [15], where such motions are usually present.

6. Interface motion due to external flow fields

In this section, the phase-field equation is tested for passive interface advection by various external flow fields, $\boldsymbol{u}_\mathrm{e}$. Uniform translations and solid body rotations are widely used to test interface tracking algorithms [9,12,41,42]. An acceptable tracking method should translate and rotate smooth bodies without significant distortion of the interface. Also, for a divergence-free flow field (as in incompressible flows) volume (or mass) should be conserved rigorously. More challenges arise when corners and sharp edges are advected in these velocity fields. Sometimes, strong and non-uniform vorticity fields are used to significantly distort the shape of the interface and induce topology changes [12,41]. In the present study, diagonal translation of a circle across a square domain is used as a first test case. Then, Zalesak's slotted disk [43] with sharp corners is used in a rotation test. Finally, a single vortex is used in a test to deform an initially circular shape, spinning and stretching it into a filament that spirals toward the vortex center [44].

In the absence of normal interface motion, Eq. (12) can be written in dimensionless form as

$$\frac{\partial\phi}{\partial t'} + \boldsymbol{u}'_\mathrm{e}\cdot\nabla'\phi = b'\left[\nabla'^2\phi + \phi(1-\phi^2) - |\nabla'\phi|\nabla'\cdot\left(\frac{\nabla'\phi}{|\nabla'\phi|}\right)\right] \tag{23}$$

where $t' = t/(W/u_\mathrm{max})$, $\boldsymbol{u}'_\mathrm{e} = \boldsymbol{u}_\mathrm{e}/u_\mathrm{max}$, and $b' = b/Wu_\mathrm{max}$, where u_max is the maximum external velocity. Eq. (23) is solved numerically in two dimensions for all of the following test cases. The fourth-order Convex ENO (CENO) scheme of Liu and Osher [45] is used to calculate the numerical fluxes for the hyperbolic term, $\boldsymbol{u}'_\mathrm{e}\cdot\nabla'\phi$, as described in Appendix C. Since the test cases in this section do not involve curvature-driven motion, b' is just a numerical parameter, as described earlier. All results presented in this section are obtained using a dimensionless grid spacing of $\Delta x' = 0.5$ and a dimensionless time step of $\Delta t' = 0.05$. Based on Eq. (C.5), the upper limit of b' is then equal to 1.2. Mass conservation is assessed by integrating the volume fraction, φ, over the entire solution domain. The relation between the volume fraction, φ, and the phase field, ϕ, is given by $\phi = 2\varphi - 1$.

6.1. Diagonal translation of a circle

In this test case, a circle of radius $R = 0.15$ is diagonally translated across a unit square domain. The circle is initially centered at (0.25, 0.25), and the phase field is initialized to $\phi = 1$ and $\phi = -1$ inside and outside of the circle, respectively. A uniform and constant velocity field, $u'_e = v'_e = 1.0$, is imposed everywhere in the domain. The circle is translated until it is centered at (0.75, 0.75); then, it is returned to its initial position by inverting the velocity field instantaneously. The circle should not change its shape as a result of the translation. Errors are measured after the circle returns back to its initial position.

The computed $\phi = 0$ contours both after the half domain translation and at the final position of the circle are plotted in Fig. 12 for two different values of b' and two mesh sizes. Fig. 12(a) shows that with $b' = 0.5$, the shape of the circle is well maintained and almost overlaps with the initial circle (dashed line) after one full domain translation. A slight improvement in the accuracy can be observed with refining the mesh from 80×80 (upper panel) to 160×160 (lower panel) grid points. On the other hand, as shown in Fig. 12(b), for $b' = 0.01$ the interface fluctuates and a significant loss of mass can be observed for both the 80×80 and 160×160 meshes. This indicates that the present form of the right-hand side of the phase-field equation together with a suitable value for the coefficient b' (i.e., 0.5) works well in maintaining the circle during translation. The cases with $b' = 0.01$ illustrate that without the right-hand side, even when a high-order advection scheme and a relatively fine grid are used, good results are generally not obtained. Recall from the discussion

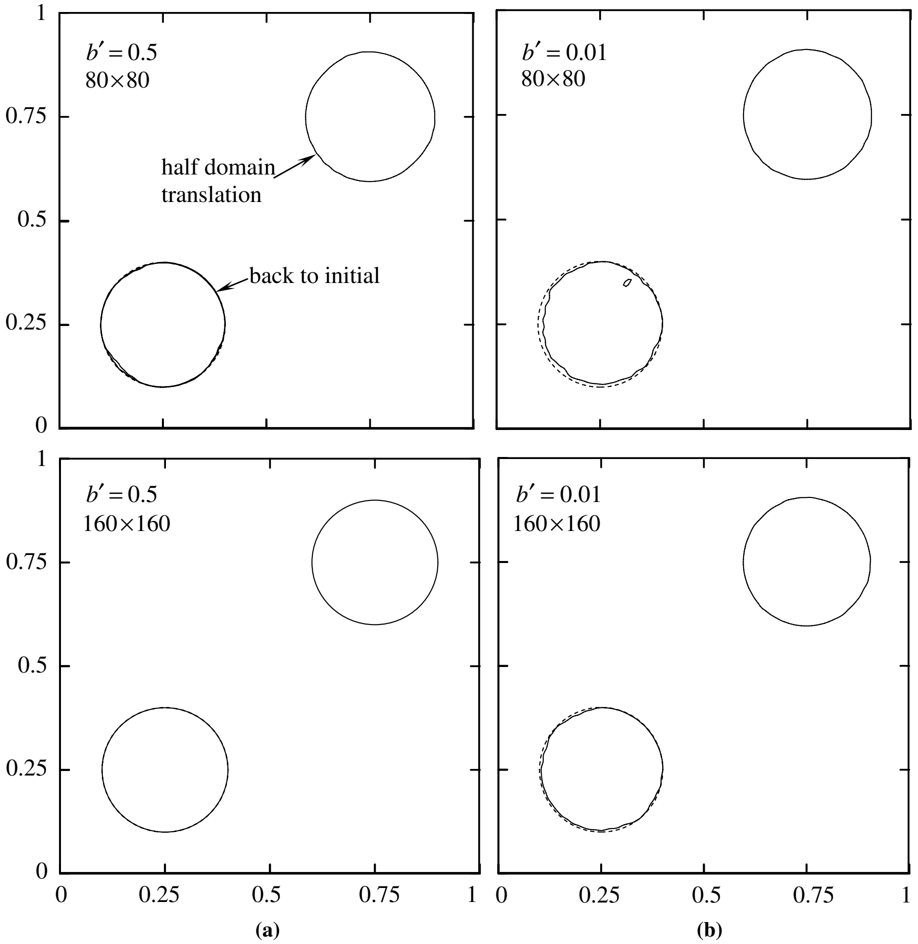


Fig. 12. Calculated $\phi = 0$ contours at half domain translation and after return to the initial position for diagonal translation of a circle (dashed line: initial interface contour; upper panels: 80×80 mesh; lower panels: 160×160 mesh): (a) $b' = 0.5$, and (b) $b' = 0.01$.

in Section 2 that for $b' = 0$ the present phase-field equation becomes identical to the level-set propagation equation. Hence, the right-hand side of the phase-field equation (for a reasonably large b') serves a similar purpose as the re-initialization scheme in the level-set method: it maintains the profile of the indicator function across the interface during motion. As shown in Section 4, the right-hand side of the phase-field equation also suppresses instabilities.

The various errors are evaluated more quantitatively in the following two figures. The L_1, L_2, and L_∞ error norms (see Appendix C) are shown as a function of b' in Fig. 13 for the 80×80 mesh. A behavior very similar to that for the propagating cosine curve in Fig. 8 can be observed. For b' approaching 0.1, the errors decrease at a second-order rate. For $b' > 0.1$, the errors are approximately constant at their respective minimum values. Slight increases in the errors are present for b' close to the maximum value of 1.2. Fig. 14 shows the mass conservation percentage (initial mass minus final divided by initial) as a function of b' for both the 80×80 and 160×160 meshes. The mass conservation error decreases rapidly as $b' = 0.1$ is approached and is relatively constant for b' between 0.1 and the maximum value of 1.2. Again, $b' = 0.5$ appears to be an optimal choice. Conservation of mass improves significantly with mesh refinement. For $b' = 0.5$ and the 160×160 mesh, mass conservation after 16,000 time steps is achieved to within 0.2%.

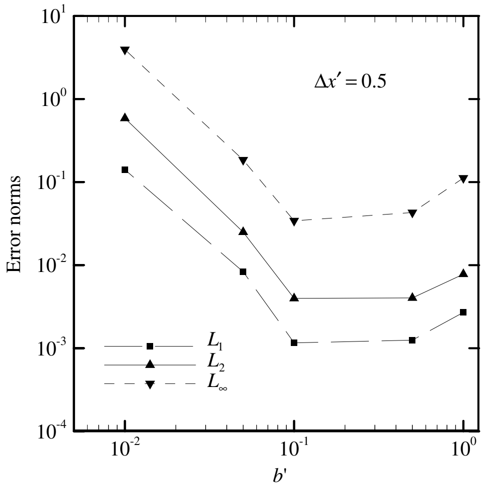


Fig. 13. Calculated error norms for a 80×80 mesh as a function of the coefficient b' for diagonal translation of a circle (see Fig. 12).

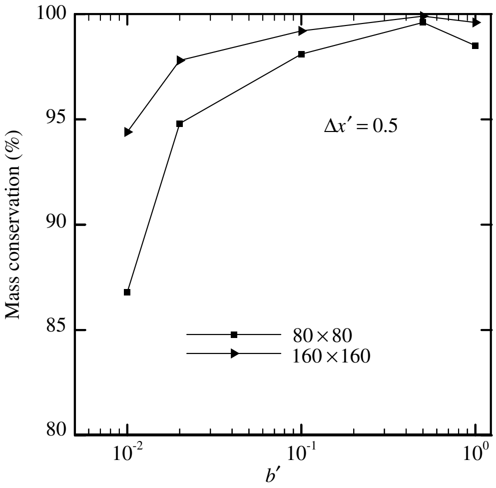


Fig. 14. Calculated mass conservation for 80×80 and 160×160 meshes as a function of the coefficient b' for diagonal translation of a circle (see Fig. 12).

6.2. Solid body rotation of a slotted disk

In the next test case, Zalesak's slotted disk [43] is rotated around the center of a unit square domain using a uniform vorticity field given by $u'_e = 2y - 1.0$ and $v'_e = 1.0 - 2x$ (see Fig. 15). The disk of radius $R = 0.15$ and slot width $H = 0.05$ is initially centered at (0.5, 0.75). The phase field is initialized to $\phi = 1$ and $\phi = -1$ inside and outside the disk, respectively. Errors are measured after the disk returns back to its initial position.

Interface contours after half and full domain rotations are shown in Fig. 15 for the same two values of b' and two mesh sizes as in Fig. 12. Fig. 15(a) shows that for $b' = 0.5$ the shape of the slotted disk is generally well maintained during rotation. For the coarser grid (upper panel), some errors can be observed at most locations along the interface. The results are improved with the finer mesh (lower panel); here, only a slight rounding of the corners occurs, while all other parts of the interface are well preserved. It can be verified that the results in the lower panel of Fig. 15(a) are close to the best results reported for this test case in the literature [4,46]. On the other hand, for $b' = 0.01$, as shown in Fig. 15(b), the advected interface becomes highly deformed and heavily jagged. For the 80 × 80 mesh (upper panel), the interfaces from the two sides of the slot are starting to merge. The results do not improve much when using the finer mesh of 160 × 160 grid points. For reference, the calculated error norms and mass losses for the four cases shown in Fig. 15 are provided in Table 1 (Cases 2a, 2b, 4a, and 4b). It can be seen that mass loss is not a significant problem when using $b' = 0.5$, i.e., the mass loss is equal to 1.7% for the 80 × 80 mesh and 1.1% for the 160 × 160 mesh. For $b' = 0.01$, on the other hand, the mass increases by more than 6% and 13% for the 80 × 80 and 160 × 160

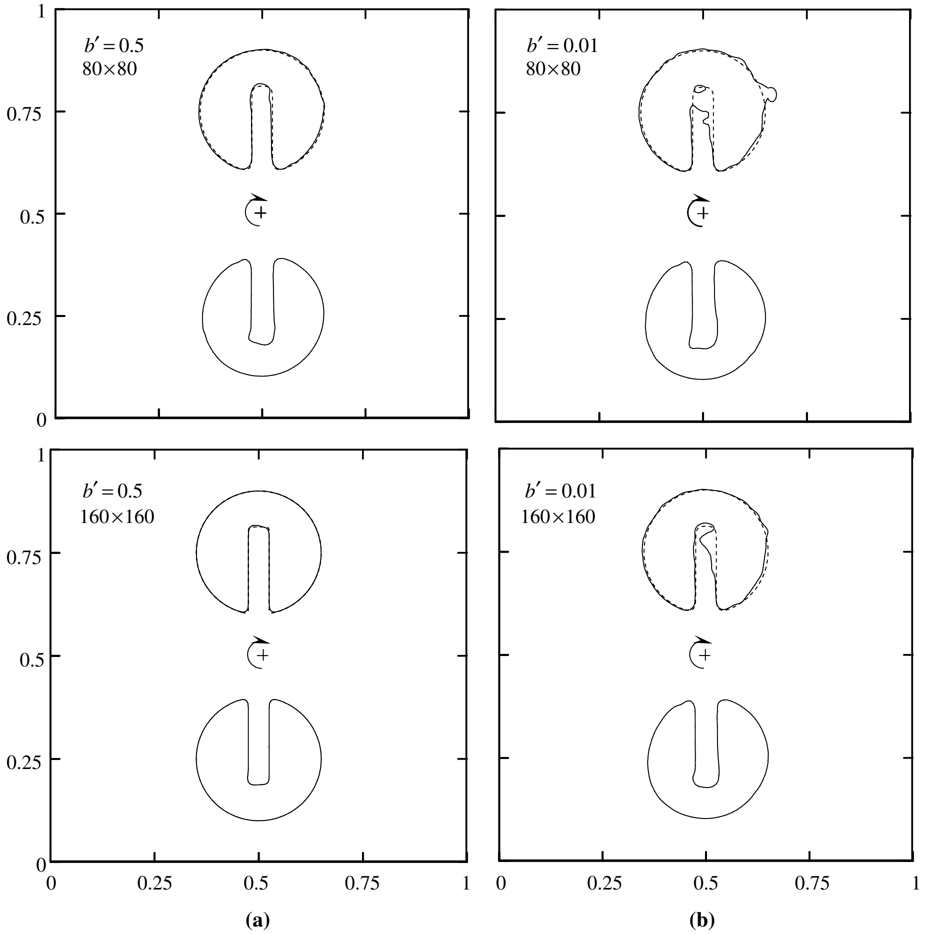


Fig. 15. Calculated $\phi = 0$ contours after half and full domain rotation for Zalesak's slotted disk test problem (dashed line: initial interface contour; upper panels: 80 × 80 mesh; lower panels: 160 × 160 mesh): (a) $b' = 0.5$, and (b) $b' = 0.01$.

Table 1
Error norms and mass losses for Zalesak's slotted disk test

Case	Grid	Present phase-field method						Level-set [13]	Particle level-set [47]
		b'	$\Delta x'$	L_1	L_2	L_∞	Mass loss (%)	Mass loss (%)	Mass loss (%)
1	50×50	0.5	0.5	1.52×10^{-2}	5.23×10^{-2}	0.523	3.3	100	3.09
2a	80×80	0.01	0.5	0.133	0.489	6.211	−13.4	N/A	N/A
2b		0.5	0.5	4.91×10^{-3}	1.74×10^{-2}	0.241	1.7		
3	100×100	0.5	0.5	2.83×10^{-3}	1.01×10^{-2}	0.168	1.4	−5.3	1.07
4a	160×160	0.01	0.5	6.71×10^{-2}	0.149	2.974	−6.1	N/A	N/A
4b		0.5	0.5	8.27×10^{-4}	3.21×10^{-3}	7.93×10^{-2}	1.1		
5	200×200	0.5	0.5	5.71×10^{-4}	2.24×10^{-3}	5.72×10^{-2}	0.8	0.54	0.22

meshes, respectively. In summary, while the slotted disk test illustrates that sharp corners can be the cause of strong numerical instabilities, the use of an optimal coefficient b' (i.e., 0.5) results in good performance of the present method. The use of a high-order advection scheme alone, without the right-hand side of Eq. (23), does not suppress instabilities and results in large mass losses.

In order to provide a quantitative comparison with the level-set method, additional runs of the slotted disk test were performed for mesh sizes of 50×50, 100×100, and 200×200. The calculated error norms and mass losses are also shown in Table 1 (Cases 1, 3, and 5). The mass losses are compared to those reported by Enright and coworkers [13,47] for two different level-set methods: the standard level-set method (with re-initialization) and a semi-Lagrangian particle level-set method. The most significant finding from this comparison is that for the two coarsest meshes (50×50 and 100×100), the present phase-field method has a similar accuracy as the particle level-set method of Ref. [47] (about 3% and 1% mass loss, respectively). As is well known, the standard level-set method [13] suffers from significant mass loss problems for such coarse meshes (100% and −5.3%, respectively). For the finest mesh (200×200), on the other hand, all three methods perform reasonably well (less than 0.8% mass loss), with the particle level-set method showing the best result (0.22% mass loss). This indicates that further improvements in the present phase-field method are possible by combining it with a semi-Lagrangian particle technique.

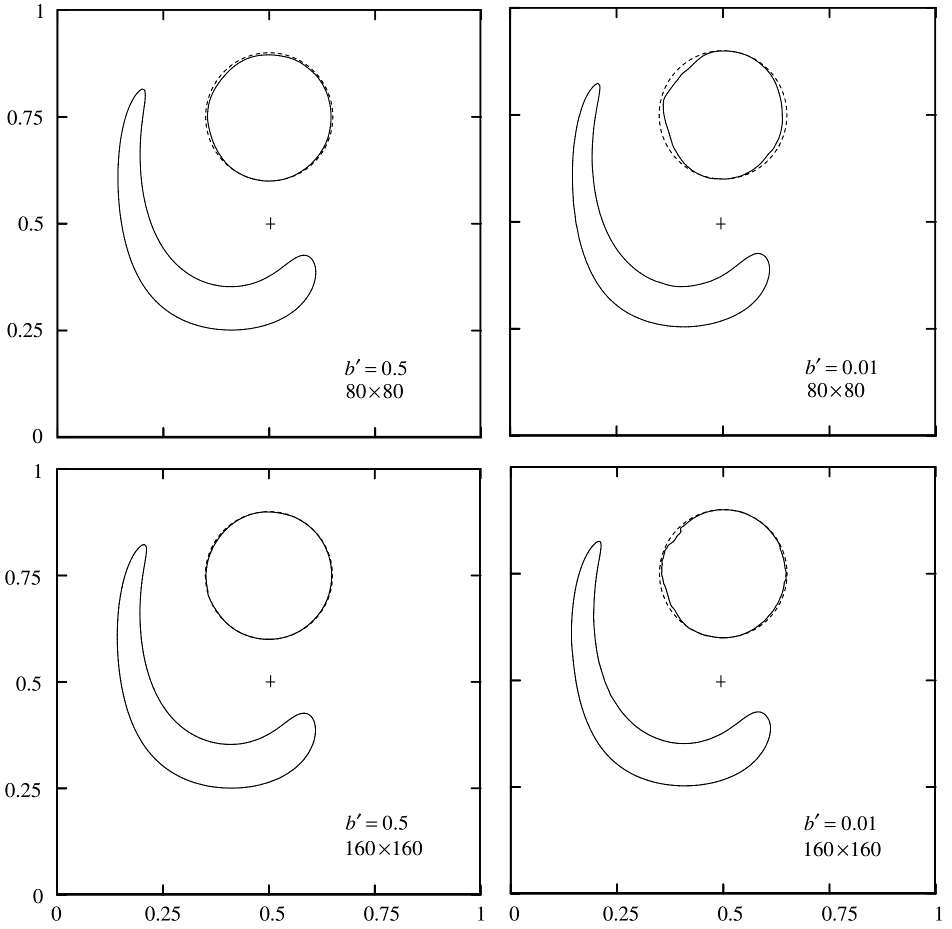
6.3. Deformation of a circle by a single vortex

The final case examined here is the single vortex test of Bell et al. [44]. In this test, a circle is deformed with a velocity field defined by the stream function

$$\Psi = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos(\pi t/T) \tag{24}$$

As shown in Fig. 16 by the dashed line, initially the circle has a radius of 0.15 and is centered at (0.50, 0.75) in a unit square domain. The phase field is initialized to $\phi = 1$ and $\phi = -1$ inside and outside the circle, respectively. The advection by the vorticity field causes the circle to evolve into a filament that spirals toward the vortex center at (0.5, 0.5). Following LeVeque [48], the temporal cosine term, $\cos(\pi t/T)$, is used to reverse the flow. The interface is back at its initial location at $t = T$, at which time the errors can be evaluated. Common reversal times used in the literature are $T = 2.0$ and $T = 12.0$ [12], and both of these values are examined in the following.

For $T = 2.0$, computed $\phi = 0$ contours at $t = 1.0$ and 2.0 are shown in Fig. 16 for the same two values of b' and two mesh sizes as in Figs. 12 and 15. As in these previous figures, the results for $b' = 0.5$ (Fig. 16(a)) can be considered good. For $b' = 0.01$ (Fig. 16(b)), the final circle at $t = T$ is heavily distorted and a loss of mass becomes apparent. The grid refinement provides a significant improvement for $b' = 0.5$, but appears to be of little value for $b' = 0.01$ since both of the results for this value of b' are rather inaccurate. The calculated error norms and mass losses for the four cases of Fig. 16 are provided in Table 2 (Cases 2a, 2b, 4a, and 4b). Based on the error norms, a convergence rate of about second order is achieved. For $b' = 0.5$, the mass loss is reasonably small for both meshes (2.2% and 0.9% for the 80×80 and 160×160 meshes, respectively). For $b' = 0.01$, the mass loss becomes unacceptably large regardless of the mesh. These observations are similar to those made in connection with the previous test cases.



(a) **(b)**

Fig. 16. Calculated $\phi = 0$ contours at maximum deformation ($t = 1.0$) and after return to the initial position ($t = 2.0$) for the single vortex test with $T = 2.0$ (dashed line: initial interface contour; upper panels: 80×80 mesh; lower panels: 160×160 mesh): (a) $b' = 0.5$, and (b) $b' = 0.01$.

Table 2
Error norms and mass losses at $t = 2.0$ for the single vortex test with $T = 2.0$

Case	Grid	Present phase-field method						VOF-PLIC [5]	VOF + marker [12]
		b'	Mass loss (%)	L_1	L_2	L_∞	L_1^{VOF}	L_1^{VOF}	L_1^{VOF}
1	64×64	0.5	3.2	6.38×10^{-3}	4.09×10^{-2}	0.483	5.23×10^{-4}	5.85×10^{-4}	2.69×10^{-4}
2a	80×80	0.01	10.3	2.94×10^{-2}	9.97×10^{-2}	3.071	N/A	N/A	N/A
2b		0.5	2.2	3.41×10^{-3}	1.84×10^{-2}	0.237			
3	128×128	0.5	1.3	8.51×10^{-4}	3.99×10^{-3}	6.44×10^{-2}	1.17×10^{-4}	1.31×10^{-4}	5.47×10^{-5}
4a	160×160	0.01	4.1	6.27×10^{-3}	3.43×10^{-2}	0.438	N/A	N/A	N/A
4b		0.5	0.9	4.42×10^{-4}	1.76×10^{-3}	3.53×10^{-2}			
5	256×256	0.5	0.6	2.22×10^{-4}	8.82×10^{-4}	1.71×10^{-2}	4.97×10^{-5}	N/A	1.36×10^{-5}

In order to compare the results for this test case to those available in the VOF literature, additional runs with mesh sizes of 64×64, 128×128, and 256×256 were made. The calculated error norms and mass losses are provided in Table 2 (Cases 1, 3, and 5). Two different VOF methods are considered: the VOF-PLIC (piecewise linear interface calculation) method of Rider and Kothe [5] and the mixed Lagrangian marker and VOF method of Aulisa et al. [12]. The comparison is based on the L_1^{VOF} error norm defined in Appendix C. The mass loss is negligibly small in the VOF methods. It can be seen that the present L_1^{VOF} error norms are of about the same magnitude as those reported for the VOF-PLIC method [5]. It should be noted, however, that the present phase-field technique avoids the relatively complex interface reconstruction procedures needed in the VOF method for the evaluation of interface normals and curvatures. Table 2 also shows that the L_1^{VOF} error norms for the more intricate mixed marker and VOF method of Ref. [12] are smaller by about a factor of two, regardless of the mesh.

Larger shape changes occur for an evolution time of $T = 12.0$, as shown in Fig. 17. Computed interface contours, using $b' = 0.5$ only, are provided in Fig. 17(a) and (b) for $t = 1.0$ and 2.0, respectively. Results are included for 160×160 and 320×320 meshes. The solid and dashed lines are for calculations without and with nonlinear preconditioning, respectively. Fig. 17 shows that, once a filament is generated, a finer grid helps to better resolve the thin "tail". For the same mesh size, nonlinear preconditioning also improves the results, as expected. The mass conservation errors are given in Table 3 for mesh sizes of 128×128, 160×160, 256×256, and 320×320. It can be seen that preconditioning reduces the mass loss by 2.1% for the 128×128 mesh, by 1.8% for the 160×160 mesh, and by 0.9% for the 256×256 and 320×320 meshes. On the other hand, more significant improvements of 7.3% and 3.8% without preconditioning and 6.1% and 2.9% with preconditioning are obtained when refining the grid by a factor of two in each direction from 128×128 to 256×256 and 160×160 to 320×320, respectively. For the 320×320 mesh with preconditioning, mass conservation is achieved to within 98.8%. Even for this mesh some mass loss occurs, because the thin filament tail becomes under-resolved.

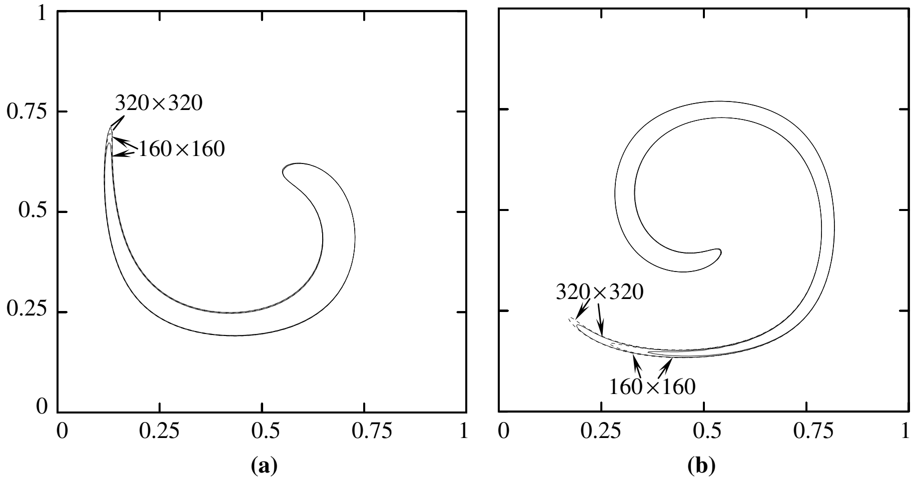


Fig. 17. Calculated $\phi = 0$ contours for the single vortex test with $T = 12.0$ using 160×160 and 320×320 meshes and $b' = 0.5$ (solid lines: without preconditioning; dashed lines: with preconditioning): (a) $t = 1.0$, and (b) $t = 2.0$.

Table 3
Mass losses at $t = 2.0$ for the single vortex test with $T = 12.0$ and $b' = 0.5$

Grid	Preconditioning	Mass loss (%)
128 × 128	No	10.4
128 × 128	Yes	8.3
160 × 160	No	5.9
160 × 160	Yes	4.1
256 × 256	No	3.1
256 × 256	Yes	2.2
320 × 320	No	2.1
320 × 320	Yes	1.2

7. Conclusions

A general interface tracking technique has been proposed that is based on the numerical solution of various forms of the phase-field equation. The method enforces a hyperbolic tangent phase-field profile of constant thickness across the interface, which avoids the need for a separate re-initialization or artificial compression scheme. Its dissipative properties make it well suited for cases that involve interface singularities. While the phase-field method is most easily applied to cases with curvature-driven interface motion, a previously developed counter term is used in those cases that do not involve such motion. The issues of finding a suitable width, W, of the hyperbolic tangent profile and an optimum coefficient b' in the absence of curvature-driven interface motion are addressed in a general manner. It is shown that for interface motion driven by arbitrary external flow fields, the fourth-order CENO scheme often used in the level-set method provides an effective numerical approximation of the advection fluxes in the phase-field equation. The effects of nonlinear preconditioning and other modifications to the standard phase-field equation are also investigated. The method is tested for a number of well-documented interface motions, including some that involve corners, sharp edges and topology changes. It is found that the phase-field method provides good numerical accuracy, mass conservation, and convergence properties. Its ease of implementation and efficient nature should make the method a popular technique for a large variety of interface tracking applications. Further improvements are possible by combining the present phase-field approach with particle or VOF methods, as has been done for the level-set method [13,47,49].

Acknowledgments

This work was supported by the US National Science Foundation under Grant No. DMR-0132225. Comments by Prof. H.S. Udaykumar of The University of Iowa are highly appreciated.

Appendix A. Determining *a* and *b* for a Stefan problem

As mentioned in Section 1, the present method for interface tracking can be used in conjunction with both sharp and diffuse interface (continuum) equations describing the physics of a problem. Even in a sharp interface approach, it is not always clear how the coefficients a and b in Eq. (4) for the normal interface velocity should be expressed if the problem is governed by a complex set of coupled interface conditions. The meaning of the coefficients a and b is even less clear in the context of diffuse interface approaches, such as the phase-field method, where sharp interface conditions are usually not considered. This appendix explains how explicit expressions for a and b can nonetheless be obtained. The specific example considered is a solidification heat flow problem (i.e., a Stefan problem) involving the Stefan and Gibbs–Thomson interface conditions.

For solidification of pure materials from the melt, the normal interface speed is controlled by the temperature gradients on each side of the interface according to the Stefan condition

$$u_{\mathrm{n}} = D\left(\left.\frac{\partial\theta}{\partial n}\right|_{\mathrm{s}} - \left.\frac{\partial\theta}{\partial n}\right|_{\mathrm{l}}\right) \tag{A.1}$$

where $\theta = (T - T_{\mathrm{m}})/(L/c_p)$ is the dimensionless temperature, T_{m} is the melting temperature, L is the latent heat of fusion, c_p is the specific heat at constant pressure, D is the thermal diffusivity, and subscripts s and l denote the solid and liquid phases, respectively. Comparing Eq. (A.1) to the general form of the normal interface speed equation, $u_{\mathrm{n}} = a - b\kappa$, i.e., Eq. (4), it can be seen that the entire right-hand side of Eq. (A.1) should be assigned to a and $b = 0$. However, this simplistic approach overlooks the fact that in many solidification problems, the interface motion depends sensitively on the local curvature of the interface. This curvature dependence arises from the (extended) Gibbs–Thomson condition for the interface temperature, θ_i, i.e.,

$$\theta_i = -d_0\kappa - \beta u_{\mathrm{n}} \tag{A.2}$$

where $d_0 = \sigma T_{\mathrm{m}} c_p/L^2$ is the capillary length, σ is the surface tension, and β is an interface kinetic coefficient. Eq. (A.2) can be solved for the normal interface speed as

$$u_{\mathrm{n}} = -\frac{\theta_i}{\beta} - \frac{d_0}{\beta}\kappa = a - b\kappa \tag{A.3}$$

so that a comparison with Eq. (4) gives $a = -\theta_i/\beta$ and $b = d_0/\beta$. However, Eq. (A.3) is of little practical use, because the interface temperature, θ_i, is not known *a priori*. More importantly, the kinetic coefficient, β, is extremely small in real solidification systems and, in fact, most analyses of solidification problems assume $\beta = 0$, so that Eq. (A.3) becomes singular. Thus, the normal interface speed is indeed controlled by the normal temperature gradients, i.e., Eqs. (A.1) and (A.2) should only be used to obtain the interface temperature. However, the temperature gradients at the interface depend strongly on the interface temperature, which in turn is a function of the interface curvature and velocity. Thus, fully implicit sharp interface approaches to Stefan problems would require an iterative procedure to simultaneously determine the interface temperature (or curvature) and velocity, while solving the unsteady heat equation in the solid and liquid regions. In an explicit approach, the interface temperature could simply be updated at the end of each time step [19–21].

Explicit expressions for the coefficients a and b can also be obtained in the context of diffuse interface approaches, as shown the following for the Stefan problem. This derivation illustrates that a set of sharp interface conditions can be satisfied exactly in a diffuse interface method. In a diffuse interface solution, the temperature varies continuously and smoothly across the diffuse interface, as illustrated in Fig. A.1. This temperature variation is obtained from the solution of a continuum heat equation that accounts for the latent heat release inside the diffuse interface (see, for example, [15]). Fig. A.1 shows that the temperature inside the diffuse interface, θ, is generally different from the temperature of the sharp interface, θ_i. In view of Fig. A.1, the difference in the temperature gradients at the interface can be approximated as

$$\left(\frac{\partial\theta}{\partial n}\bigg|_{\mathrm{s}} - \frac{\partial\theta}{\partial n}\bigg|_{\mathrm{l}}\right) \sim \frac{\theta_i - \theta}{W} \tag{A.4}$$

where W is, as before, a measure of the width of phase-field variation across the interface. Substituting Eq. (A.1), Eq. (A.4) becomes

$$\frac{\theta_i - \theta}{W} = A\frac{u_{\mathrm{n}}}{D} \tag{A.5}$$

where A is a constant of proportionality. Substituting Eq. (A.2) for θ_i into Eq. (A.5) yields

$$\theta = -d_0\kappa - \left(\beta + \frac{AW}{D}\right)u_{\mathrm{n}} \tag{A.6}$$

which can be compared directly to Eq. (A.2). It can be seen that the temperature variation inside the diffuse interface yields an effective kinetic coefficient equal to AW/D. Solving Eq. (A.6) for u_{n} yields

$$u_{\mathrm{n}} = \frac{-\theta}{\beta + AW/D} - \frac{d_0}{\beta + AW/D}\kappa \tag{A.7}$$

A comparison with Eq. (4) gives

$$a = \frac{-\theta}{\beta + AW/D} \quad \text{and} \quad b = \frac{d_0}{(\beta + AW/D)} \tag{A.8}$$

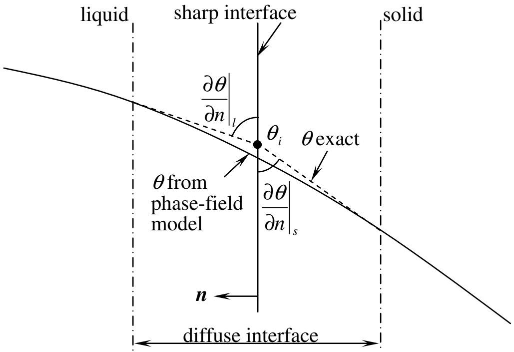


Fig. A.1. Schematic illustration of the thin-interface analysis of the Stefan problem in the context of the phase-field method.

Now, the expression for a no longer contains the unknown interface temperature θ_i, but the temperature θ which is known from the solution of the continuum heat equation. More importantly, for a finite width W, the kinetic coefficient β can be allowed to vanish without causing any singularity. The constant of proportionality A is of the order of unity and its exact value can be determined analytically [27]. Substituting Eq. (A.8) into Eq. (10) or (11), and solving the resulting phase-field equation together with a diffuse interface version of the heat equation, causes the solution to satisfy the Stefan and Gibbs–Thomson interface conditions exactly, even though the interface is diffuse. The above result for the coefficients a and b is equivalent to that obtained by Karma and Rappel [27] from a matched asymptotic analysis of the coupled phase-field and heat equations; this analysis is commonly referred to as the thin-interface limit. A thin-interface analysis for two-phase flow problems can be found in Refs. [16,33,34].

Appendix B. Relation to free energy based phase-field models

In certain classes of thermodynamically derived phase-field models, the phase field ϕ is governed by the Allen–Cahn equation, which guarantees a decrease in the total free energy with time, as in [15]

$$\frac{\partial\phi}{\partial t} = -\frac{1}{M}\frac{\delta F}{\delta\phi} \tag{B.1}$$

where M is the positive interface mobility, F is the free energy functional, and δ denotes the functional derivative $\delta/\delta\phi = \partial_V/\partial\phi - \nabla[\partial_V/\partial(\nabla\phi)]$, where the subscript V denotes the functional density. Taking as an example the solidification of a pure substance, F can be written as

$$F = \int_V \left[f(\phi,\theta) + \frac{\varepsilon^2}{2}|\nabla\phi|^2 \right] \mathrm{d}V \tag{B.2}$$

where f is the bulk free energy density per unit volume as a function of ϕ and temperature θ, and ε is a gradient energy coefficient that is related to surface tension. The free energy density, f, can be expressed as $f = g + \lambda p\theta$, where g is a double-well function in ϕ with the two minima corresponding to the bulk solid and liquid phases, p is an interpolating function, and λ is a coupling constant. Substituting Eq. (B.2) into (B.1) yields

$$M\frac{\partial\phi}{\partial t} = \varepsilon^2\nabla^2\phi - \frac{\partial g}{\partial\phi} - \lambda\frac{\partial p}{\partial\phi}\theta \tag{B.3}$$

One possible choice for the double-well function is $g = h(-\phi^2/2 + \phi^4/4)$, where h is the height of the double-well [50]. The interpolating function, p, can be chosen as $p = \phi - \phi^3/3$, such that $\partial p/\partial\phi = 1 - \phi^2$ [51]; other relations have been investigated as well [50,52]. With $W = \varepsilon/\sqrt{h}$ [15], Eq. (B.3) can then be rewritten as

$$\frac{\partial\phi}{\partial t} + \frac{\lambda}{M}(1-\phi^2)\theta = \frac{\varepsilon^2}{M}\left[\nabla^2\phi + \frac{\phi(1-\phi^2)}{W^2}\right] \tag{B.4}$$

In the absence of external advection ($\boldsymbol{u}_\mathrm{e} = 0$), and taking $b = \varepsilon^2/M$ and $a = \sqrt{2}\lambda\theta W/M$, Eq. (B.4) is equivalent to Eq. (11).

Additional explanations regarding thermodynamically derived phase-field models for simulating solidification, including alloys, crystalline anisotropy, convection, multiple phases, grain boundaries and other effects, can be found in the review of Boettinger et al. [15].

Appendix C. Numerical implementation and error measures

For simplicity, the numerical implementation is only presented for the two-dimensional case. The extension to three dimensions is straightforward, and a three-dimensional interface tracking example is presented in Section 5.

The phase-field equation is discretized on a square grid with a spacing equal to Δx. For the Laplacian of the phase field, a 9-point finite-difference stencil commonly used in phase-field simulations [37,53] is adopted, i.e.,

$$\nabla^2\phi_{i,j} = \frac{2(\phi_{i+1,j}+\phi_{i,j+1}+\phi_{i-1,j}+\phi_{i,j-1}-4\phi_{i,j})+0.5(\phi_{i+1,j+1}+\phi_{i+1,j-1}+\phi_{i-1,j+1}+\phi_{i-1,j-1}-4\phi_{i,j})}{3\Delta x^2} \tag{C.1}$$

where i and j are the node indices. For the norm of ϕ, $|\nabla\phi|$, a central difference scheme is used, i.e.,

$$|\nabla\phi|_{i,j} = \frac{1}{\Delta x}\sqrt{\frac{(\phi_{i+1,j}-\phi_{i-1,j})^2}{4}+\frac{(\phi_{i,j+1}-\phi_{i,j-1})^2}{4}} \tag{C.2}$$

For the mean curvature, $\kappa = \nabla\cdot(\nabla\phi/|\nabla\phi|)$, the method used in Echebarria et al. [53] is adopted here as

$$\begin{aligned}\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)_{i,j} = \frac{1}{\Delta x}\Bigg(&\frac{\phi_{i+1,j}-\phi_{i,j}}{\sqrt{(\phi_{i+1,j}-\phi_{i,j})^2+(\phi_{i+1,j+1}+\phi_{i,j+1}-\phi_{i+1,j-1}-\phi_{i,j-1})^2/16}}\\ &-\frac{\phi_{i,j}-\phi_{i-1,j}}{\sqrt{(\phi_{i,j}-\phi_{i-1,j})^2+(\phi_{i-1,j+1}+\phi_{i,j+1}-\phi_{i-1,j-1}-\phi_{i,j-1})^2/16}}\\ &+\frac{\phi_{i,j+1}-\phi_{i,j}}{\sqrt{(\phi_{i+1,j+1}+\phi_{i+1,j}-\phi_{i-1,j+1}-\phi_{i-1,j})^2/16+(\phi_{i,j+1}-\phi_{i,j})^2}}\\ &-\frac{\phi_{i,j}-\phi_{i,j-1}}{\sqrt{(\phi_{i+1,j-1}+\phi_{i+1,j}-\phi_{i-1,j-1}-\phi_{i-1,j})^2/16+(\phi_{i,j}-\phi_{i,j-1})^2}}\Bigg)\end{aligned} \tag{C.3}$$

The fourth-order HJ ENO scheme is used to calculate the numerical fluxes, $(\phi_x^-)_{i,j}$, $(\phi_x^+)_{i,j}$, $(\phi_y^-)_{i,j}$, and $(\phi_y^+)_{i,j}$, for the hyperbolic term $\boldsymbol{u}_\mathrm{e}\cdot\nabla\phi$ in the phase-field equation. Here, ϕ_x and ϕ_y denote the partial derivatives of ϕ with respect to the spatial coordinates x and y, respectively. The superscripts $-$ and $+$ denote backward and forward differencing, respectively. The sign of the velocity, u_e, determines the use of either $(\phi_x^+)_{i..j}$ or $(\phi_x^-)_{i..j}$, whichever is in the upwind direction. The original ENO scheme [6,7] uses the "smoothest" possible polynomial interpolation to approximate the fluxes. For example, to calculate $(\phi_x^-)_{i,j}$ by a second-order scheme, one has to choose the smaller of the two possible divided differences obtained from the stencils $\{\phi_{i-2},\phi_{i-1},\phi_i\}$ and $\{\phi_{i-1},\phi_i,\phi_{i+1}\}$. The ENO philosophy of picking exactly one of the candidate stencils may be overkill in smooth regions where data are well behaved [54]. Weighted ENO (WENO) schemes [54,55] choose convex combination weights of the ENO approximates to improve accuracy in smooth regions. In the present study, the CENO scheme [45] is implemented. The scheme chooses the divided difference value "closest" to the previous order flux. This means, to choose a nth-order flux $(\phi_x^-)_{i,j}^{(n)}$ between n candidate stencils $\{\phi_{i-n,j},\phi_{i-n+1,j},\ldots,\phi_{i,j}\},\{\phi_{i-n+1,j},\phi_{i-n+2,j},\ldots,\phi_{i+1,j}\},\ldots,$ and $\{\phi_{i-1,j},\phi_{i,j},\ldots,\phi_{i+n-1,j}\}$, the convex combination of these n candidates "closest" to $(\phi_x^-)_{i,j}^{(n-1)}$ is taken. The CENO scheme reduces to lower order automatically at discontinuities, while maintaining higher order in smooth regions [56].

The simple forward Euler method is used for the time discretization of the phase-field equation. With the present temporal and spatial discretization schemes, the CFL condition for Eq. (12) is given by

$$\Delta t\left[\frac{|a\phi_x/|\nabla\phi|+u_\mathrm{e}|}{\Delta x}+\frac{|a\phi_y/|\nabla\phi|+v_\mathrm{e}|}{\Delta x}+\frac{10b}{3(\Delta x)^2}\right]<1 \tag{C.4}$$

where u_e and v_e are the x and y components of the external velocity $\boldsymbol{u}_\mathrm{e}$, respectively, and Δt is the time step. The factor 10/3 in the last term on the right-hand side of Eq. (C.4) stems from the use of a 9-point stencil for $\nabla^2\phi$; a 5-point stencil would result in the usual factor of 4. In the presence of curvature-driven interface motion, the coefficient b is a physical parameter, and Eq. (C.4) can be used directly to determine the maximum time step for a given Δx. In the absence of curvature-driven interface motion, on the other hand, b is a purely numerical parameter and Eq. (C.4) can be used to determine a constraint on the coefficient b. Defining the Courant number (i.e., the ratio between the time step and the time required for the interface to be advected by one grid spacing) as $Cr = \max(|a\phi_x/|\nabla\phi|+u_\mathrm{e}|,|a\phi_y/|\nabla\phi|+v_\mathrm{e}|)\Delta t/\Delta x$, it follows that:

$$b < \frac{3}{10}\frac{\Delta x^2}{\Delta t}(1 - 2Cr) \tag{C.5}$$

As shown in Section 3, the grid spacing Δx must be chosen equal to or smaller than W in order to accurately resolve the hyperbolic tangent profile. Without curvature-driven motion, the interface is purely advected, and hence the time step can be determined solely based on the Cr number. Typically, a value of $Cr = 0.1$ provides sufficient accuracy. With Δx and Δt chosen, Eq. (C.5) is then used to calculate the upper limit for b.

To evaluate the accuracy of the present interface tracking algorithm, errors are calculated by measuring the distance between the calculated zero phase-field contour and the exact location of the interface (or the results from the finest mesh). Let $\phi_{i,j}$ be the calculated value of ϕ at the grid point (x_i, y_j), as illustrated in Fig. C.1. If ϕ changes its sign when moving to one of the neighbor points, i.e., if one of the four products $\phi_{i,j}\phi_{i+1,j}$, $\phi_{i,j}\phi_{i-1,j}$, $\phi_{i,j}\phi_{i,j+1}$, and $\phi_{i,j}\phi_{i,j-1}$ is less than or equal to zero, the grid point (x_i, y_j) is called "next" to the $\phi = 0$ contour. The coordinates of the point (x_n, y_n) on the $\phi = 0$ contour associated with (x_i, y_j) (Fig. C.1) are obtained from $x_n = x_i + n_x\psi_{i,j}$ and $y_n = y_j + n_y\psi_{i,j}$, where $\psi_{i,j}$ is the signed distance for (x_i, y_j) and is obtained from $\phi_{i,j}$ via $\psi = -\sqrt{2}W\tanh^{-1}\phi$. The interface normal, $\boldsymbol{n} = \nabla\phi/|\nabla\phi|$, is evaluated using the method of Sethian [3], where the average of all four possible normals is taken to approximate the normal at a corner. The following error norms are then calculated for all grid points k that are next to the $\phi = 0$ contour,

$$E^{L_1} = \frac{\sum_{k=1}^{N}\left|y_{n,k} - y_{n,k}^{\mathrm{ex}}\right|}{N} \tag{C.6}$$

$$E^{L_2} = \sqrt{\frac{\sum_{k=1}^{N}\left|y_{n,k} - y_{n,k}^{\mathrm{ex}}\right|^2}{N}} \tag{C.7}$$

$$E^{L_\infty} = \max\left|y_{n,k} - y_{n,k}^{\mathrm{ex}}\right| \tag{C.8}$$

where N is the total number of grid points next to the $\phi = 0$ contour. As shown in Fig. C.1, the point (x_n, y_n^{ex}) is on the exact interface contour, $\phi^{\mathrm{ex}} = 0$. If $\phi^{\mathrm{ex}} = 0$ is described by the function $y^{\mathrm{ex}} = f(x)$, it follows that $y_n^{\mathrm{ex}} = f(x_n)$. Otherwise, y_n^{ex} is estimated using the results obtained from the finest mesh. Therefore, the error is the distance parallel to the y axis between points (x_n, y_n^{ex}) and (x_n, y_n), as shown in Fig. C.1.

In order to quantitatively compare the present method to results reported in the VOF literature [5,12], the following definition of the L_1 error norm is also used:

$$E^{L_1^{\mathrm{VOF}}} = \sum_{ij} A_{i,j}\left|\varphi_{i,j} - \varphi_{i,j}^{\mathrm{ex}}\right| \tag{C.9}$$

where $A_{i,j}$ is the area of the grid cell corresponding to node i, j and $\varphi = (\phi + 1)/2$ is the volume fraction. Note that the L_1^{VOF} norm is evaluated for all grid points in the solution domain, including the grid points far away

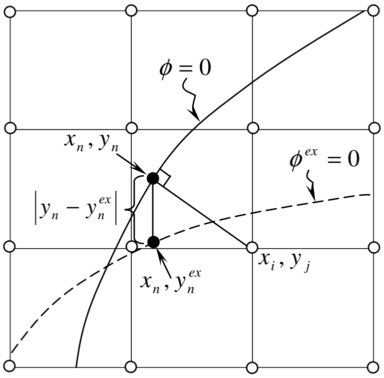


Fig. C.1. Schematic illustration of method used to calculate the error in the interface location for the grid point (x_i, y_j); the solid line is the computed $\phi = 0$ contour and the dashed line is the exact interface contour ($\phi^{\mathrm{ex}} = 0$); the error is defined as the distance between $(x_{n,}, y_n)$ and (x_n, y_n^{ex}), i.e., $|y_n - y_n^{\mathrm{ex}}|$.

from the interface where the error is naturally zero; the L_1, L_2, and L_∞ error norms, on the other hand, are calculated only for the grid points next to the $\phi = 0$ contour, making them independent of the size of the domain. Also, the L_1, L_2, and L_∞ error norms are based on the distance between the calculated $\phi = 0$ and exact $\phi^{\mathrm{ex}} = 0$ contours, and this distance can be larger than unity. The L_1^{VOF} error norm is based on volume fractions, which are always between 0 and 1.

References

[1] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Ann. Rev. Fluid Mech. 31 (1999) 567.

[2] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithm based on Hamilton–Jacobi formations, J. Comput. Phys. 79 (1988) 12.

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

[4] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2002.

[5] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112.

[6] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439.

[7] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, J. Comput. Phys. 83 (1989) 32.

[8] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high-order accurate essentially non-oscillatory schemes III, J. Comput. Phys. 131 (1997) 3.

[9] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2004) 465.

[10] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2002) 400.

[11] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25.

[12] E. Aulisa, S. Manservisi, R. Scardovelli, A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows, J. Comput. Phys. 188 (2003) 611.

[13] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (2002) 83.

[14] D.M. Anderson, G.B. McFadden, A.A. Wheeler, Diffuse-interface methods in fluid mechanics, Ann. Rev. Fluid Mech. 30 (1998) 139.

[15] W.J. Boettinger, J.A. Warren, C. Beckermann, A. Karma, Phase-field simulation of solidification, Ann. Rev. Mater. Res. 32 (2002) 163.

[16] C. Beckermann, H-J. Diepers, I. Steinbach, A. Karma, X. Tong, Modeling melt convection in phase-field simulations of solidification, J. Comput. Phys. 154 (1999) 468.

[17] L. Tan, N. Zabaras, A level set simulation of dendritic solidification with combined features of front-tracking and fixed-domain methods, J. Comput. Phys. 211 (2006) 36.

[18] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (2005) 225.

[19] S. Marella, S. Krishnan, H. Liu, H.S. Udaykumar, Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations, J. Comput. Phys. 210 (2005) 1.

[20] H. Liu, S. Krishnan, S. Marella, H.S. Udaykumar, Sharp interface Cartesian grid method II: a technique for simulating droplet interactions with surface of arbitrary shape, J. Comput. Phys. 210 (2005) 32.

[21] Y. Yang, H.S. Udaykumar, Sharp interface Cartesian grid method III: solidification of pure materials and binary solutions, J. Comput. Phys. 210 (2005) 55.

[22] R. Folch, J. Casademunt, A. Hernández-Machado, Phase-field model for Hele-Shaw flows with arbitrary viscosity contrast. I. Theoretical approach, Phys. Rev. E 60 (1999) 1724.

[23] T. Biben, C. Misbah, A. Leyrat, C. Verdier, An advected-field approach to the dynamics of fluid interfaces, Europhys. Lett. 63 (2003) 623.

[24] T. Biben, C. Misbah, Tumbling of vesicles under shear flow within an advection-field approach, Phys. Rev. E 67 (2003) 031908.

[25] J. Beaucourt, F. Rioual, T. Seon, T. Biben, C. Misbah, Steady to unsteady dynamics of a vesicle in a flow, Phys. Rev. E 69 (2004) 011906.

[26] I. Steinbach, C. Beckermann, B. Kauerauf, Q. Li, J. Guo, Three-dimensional modeling of equiaxed dendritic growth on a mesoscopic scale, Acta Mater. 47 (1999) 971.

[27] A. Karma, W.-J. Rappel, Phase-field model for computationally efficient modeling of solidification with arbitrary interface kinetics, Phys. Rev. E 53 (1996) R3017.

[28] D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling, J. Comput. Phys. 155 (1999) 96.

[29] L.M. Pismen, Mesoscopic hydrodynamics of contact line motion, Colloid Surf. A 206 (2002) 11.

[30] C. Liu, J. Shen, A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method, Physica D 179 (2003) 211.

[31] V.E. Badalassi, H.D. Ceniceros, S. Banerjee, Computation of multiphase systems with phase field models, J. Comput. Phys. 190 (2003) 371.

[32] P. Yue, J.J. Feng, C. Liu, J. Shen, A diffuse-interface method for simulating two-phase flows of complex methods, J. Fluid Mech. 515 (2004) 293.
[33] Y. Sun, C. Beckermann, Phase-field simulation of two-phase micro-flows in a Hele-Shaw cell, in: A.A. Mammoli, C.A. Brebbia (Eds.), Computational Methods in Multiphase Flow III, WIT Press, UK, 2005, pp. 147–157.
[34] Y. Sun, C. Beckermann, Diffuse interface modeling of two-phase flows based on averaging: mass and momentum equations, Physica D 198 (2004) 281.
[35] K. Glasner, Nonlinear preconditioning for diffuse interfaces, J. Comput. Phys. 174 (2001) 695.
[36] P. Gómez, J. Hernández, J. López, On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows, Int. J. Numer. Meth. Eng. 63 (2005) 1478.
[37] J.C. Ramirez, C. Beckermann, Examination of binary alloy free dendritic growth theories with a phase-field model, Acta Mater. 53 (2005) 1721.
[38] J.A. Sethian, Curvature and the evolution of fronts, Commun. Math. Phys. 101 (1985) 487.
[39] J.A. Sethian, Numerical algorithms for propagating interfaces: Hamilton–Jacobi equations and conservation laws, J. Diff. Geom. 31 (1990) 131.
[40] D.L. Chopp, J.A. Sethian, Flow under curvature: singularity formation, minimal surfaces, and geodesics, Exp. Math. 2 (1993) 235.
[41] W.J. Rider, D.B. Kothe, Stretching and tearing interface tracking methods, in: 12th AIAA CFD Conference, AIAA 95-1717, 1995.
[42] M. Rudman, Volume-tracking methods for interfacial flow calculations, Int. J. Numer. Meth. Fluids 24 (1997) 671.
[43] S.T. Zalesak, Fully multi-dimensional flux corrected transport algorithms for fluid flow, J. Comput. Phys. 33 (1979) 335.
[44] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (1989) 257.
[45] X.-D. Liu, S. Osher, Convex ENO high order schemes without field-by-field decomposition or staggered grids, J. Comput. Phys. 142 (1998) 304.
[46] J. López, J. Hernández, P. Gómez, F. Faura, A volume of fluid method based on multidimensional advection and spline interface reconstruction, J. Comput. Phys. 195 (2004) 718.
[47] D. Enright, F. Losasso, R. Fedkiw, A fast and accurate semi-Lagrangian particle level set method, Comput. Struct. 83 (2005) 479.
[48] R.J. LeVeque, High-resolution conservation algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (1996) 627.
[49] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301.
[50] A. Karma, W.-J. Rappel, Quantitative phase-field modeling of dendritic growth in two and three dimensions, Phys. Rev. E 57 (1998) 4323.
[51] R.F. Almgren, Second-order phase field asymptotics for unequal conductivities, SIAM J. Appl. Math. 59 (1999) 2086.
[52] Y.-T. Kim, N. Provatas, N. Goldenfeld, J. Dantzig, Universal dynamics of phase-field models for dendritic growth, Phys. Rev. E 59 (1999) R2546.
[53] B. Echebarria, R. Folch, A. Karma, M. Plapp, Quantitative phase-field model of alloy solidification, Phys. Rev. E 70 (2004) 061604.
[54] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200.
[55] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202.
[56] H.S. Udaykumar, L. Tran, D.M. Belk, K.J. Vanden, An Eulerian method for computation of multimaterial impact with ENO shock-capturing and sharp interfaces, J. Comput. Phys. 186 (2003) 136.