Level Set and Phase Field Methods: Application to Moving Interfaces and Two-Phase Fluid Flows

Maged Ismail
Claremont Graduate University

Abstract

Level Set and Phase Field methods are well-known interface-capturing techniques for approximation of moving interfaces. In the first part of this project, a recently developed sharp-interface phase-field method [1] is implemented in Matlab and used to implicitly capture normal interface motion and interface advection by an external velocity field. The interface evolution equation is solved numerically using an explicit finite difference scheme. In the second part, a conservative level set method, introduced by Olsson and Kreiss [2], is used for simulation of incompressible two-phase flows. Several two-phase test-cases are considered, including rising bubble and falling droplet. Differences between the current methods and the original level set method are highlighted.

1. Introduction

Problems involving moving interfaces or boundaries arise in a wide range of applications. Examples include simulation of two-phase flows, melting and solidification, and combustion and reacting flows. The two main approaches to characterize moving interfaces are interface tracking (Eulerian) and interface capturing (Lagrangian) techniques. In interface tracking techniques the moving interface is represented and tracked explicitly using marker particles distributed evenly on the interface; whereas in interface capturing techniques such as level set and phase-field methods, the interface is implicitly captured by a certain level set or isocontour of a globally defined scalar function. Major drawbacks of the (explicit) interface tracking techniques include the difficulty in handling topological changes, such as merging curves, without some ad-hoc manner, and their inability to produce desired "entropy solutions" around sharp corners. Therefore interface capturing techniques have become more widely used, especially in applications where topological changes and/or sharp corners are present.

A brief overview of the level set method is given here (for more detailed description of the method the reader is referred to the texts [3, 4]). The method, which has been introduced by Osher and Sethian [5], is a simple and versatile method for computing and analyzing the motion of an interface in two or three dimensions. It is based on introducing a continuous auxiliary function (level set function), ϕ, over the whole global domain, and embedding of the interface as the zero level set of this higher dimensional function. The level set function can be advected with a background flow field, thus propagating the interface exactly as the zero level set of ϕ.

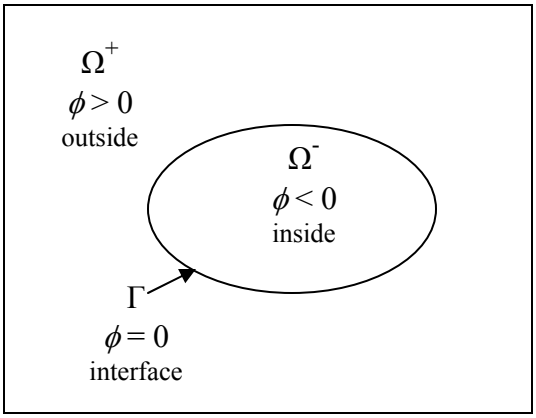

Fig. 1. Level set representation of the interface Γ.

Consider a simple closed curve Γ(*t*), enclosing Ω^- in a region Ω, and moving in two dimensions (see Fig. 1). In the level set formulation, the interface is represented as the zero level set of ϕ with the boundary defined as

$$\Gamma = \left\{ \boldsymbol{x} \in \Omega \,|\, \phi(\boldsymbol{x}, t) = 0 \right\} \tag{1.1}$$

The level set function $\phi(\boldsymbol{x}, t)$ is customarily defined as a signed distance function to the interface, so

$$\phi(\boldsymbol{x}, t) = \begin{cases} +d(\boldsymbol{x}, t) & \boldsymbol{x} \in \Omega^- \\ 0 & \boldsymbol{x} \in \Gamma \\ -d(\boldsymbol{x}, t) & \boldsymbol{x} \in \Omega^+ \end{cases} \tag{1.2}$$

where

$$d(\boldsymbol{x}) = \min_{\boldsymbol{x}_\Gamma \in \Gamma} \left(\left| \boldsymbol{x} - \boldsymbol{x}_\Gamma \right| \right) \tag{1.3}$$

and the evolution of ϕ for a velocity field $\boldsymbol{u}(\boldsymbol{x},t)$ is governed by

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0 \tag{1.4}$$

The above formulation can handle topological changes in the evolving interface Γ naturally; the position of the interface at time *t* is given by the zero level set $\phi(\boldsymbol{x}, t) = 0$ of the evolving level set function. This set does not need to be connected and can break and merge as *t* advances. Moreover geometrical quantities, such as the unit normal $\boldsymbol{n}$ and the curvature κ can be easily determined using the level set function ϕ.

The unit normal at any point of the front is given by

$$\mathbf{n} = \nabla\phi / |\nabla\phi| \tag{1.5}$$

and the mean curvature of the interface can be found by taking the divergence of the unit normal vector to the front,

$$\kappa = \nabla \cdot \boldsymbol{n} = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) \tag{1.6}$$

It should be noted that level set function ϕ will not necessarily remain a signed distance function, under the evolution of the advection equation, even though it is initialized as such. Typically, a reinitialization step, Eq. (1.7) below, is performed to keep ϕ as a signed distance function near the interface while keeping the original zero level set unchanged. In other words, a given a level set function ϕ, at time t, is reinitialized to a signed distance function by integrating the following partial differential equation to steady state

$$\frac{\partial\phi}{\partial\tau} = sign(\phi_0)(1 - \nabla\phi) \tag{1.7}$$

here τ is an artificial time. Eq. (1.7) is solved using initial conditions $\phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x})$, where

$$sign(\phi) = \begin{cases} 1 \text{ if } \phi > 0 \\ 0 \text{ if } \phi = 0 \\ -1 \text{ if } \phi < 0 \end{cases} \tag{1.8}$$

In the next two sections, two variants of the original level set method, namely the sharp-interface phase-field method and the conservative level set method, are described and used for simulation of moving interfaces and incompressible two-phase flows.

2. Part I: Sharp-Interface Phase-Field Method

The sharp-interface phase-field method is a general interface capturing technique that has been recently introduced by Sun and Beckermann [1]. A hyperbolic tangent phase-field profile is enforced across the interface, and the zero phase-field contour is used to implicitly capture the interface. The method shares some of the advantages of the original level set method, e.g. the ability to handle topological changes naturally. Moreover it maintains a hyperbolic tangent phase-field profile of constant thickness normal to the interface, and thus it eliminates the need for a separate reinitialization step, as for the signed distance function in the level-set method.

2.1 Mathematical Formulation

The same general interface advection equation, Eq. (1.4), is used here to derive the phase-field equation. However, we denote ϕ here as the phase-field function.

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0 \tag{2.1}$$

The velocity field can be split into a normal interface speed, u_n and an interface velocity due to external advection, $\boldsymbol{u}_e$,so that

$$\boldsymbol{u} = u_n \boldsymbol{n} + \boldsymbol{u}_e \tag{2.2}$$

where $\boldsymbol{n} = \nabla\phi / |\nabla\phi|$ is the unit normal vector to the interface, as defined earlier.

Then, the normal interface speed, u_n, can be further decomposed into a constant and a curvature dependent terms as follows

$$u_n = a - b\kappa \tag{2.3}$$

The curvature can be calculated from the phase-field function using

$$\kappa = \nabla \cdot \boldsymbol{n} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) = \frac{1}{|\nabla \phi|} \left[\nabla^2 \phi - \frac{(\nabla \phi \cdot \nabla)|\nabla \phi|}{|\nabla \phi|} \right] \tag{2.4}$$

The phase-field function ϕ is taken to vary normal to the interface according to

$$\phi = -\tanh\left(\frac{n}{\sqrt{2}W} \right) \tag{2.5}$$

where W is a parameter that controls the width of the hyperbolic tangent profile and n is the coordinate normal to the interface. After substituting into Eq. (2.1) and simplifying (for details see [1]), the following phase-field equation can be obtained

$$\frac{\partial \phi}{\partial t} + a|\nabla \phi| + \boldsymbol{u}_e \cdot \nabla \phi = b \left[\nabla^2 \phi + \frac{\phi(1-\phi^2)}{W^2} - |\nabla \phi| \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right] \tag{2.6}$$

2.2 Numerical Implementation

The phase-field evolution equation, Eq. (2.6), is solved numerically using the finite-difference method on a Cartesian grid in a rectangular domain. The numerical algorithm is implemented in Matlab. The time integration is done using simple forward Euler (explicit) scheme. More details of the numerical implementation are provided in Appendix A.

2.3 Numerical Experiments and Results

2.3.1 Interface motion with constant normal speed

In the case of an interface exclusively moving with a constant normal speed, such that $u_n = a$ = const. and $\boldsymbol{u}_e = 0$, Eq. (2.6) is reduced to

$$\frac{\partial \phi}{\partial t} + a\left|\nabla \phi\right| = b\left[\nabla^2 \phi + \frac{\phi\left(1-\phi^2\right)}{W^2} - \left|\nabla \phi\right| \nabla \cdot \left(\frac{\nabla \phi}{\left|\nabla \phi\right|}\right)\right] \tag{2.7}$$

Since we don't consider curvature-driven interface motion in this test-case, the coefficient b in the above equation is just a numerical parameter that controls the relaxation behavior of the phase-field profile and the smoothing of interface singularities. The values used here for this numerical parameter, are based upon the values recommended by Sun and Beckermann [1], taking into consideration that the phase-field equation is solved here in dimensional form, whereas the dimensionless form is used in [1].

We consider here the problem of a periodic cosine curve propagating with normal speed of unity (i.e. $u_n = a = 1$). The initial curve is parameterized by

$$\Gamma(0) = \left[1-s, \left(1+\cos 2\pi s\right)/4\right], \quad 0 \le s \le 1 \tag{2.8}$$

where $\Gamma(t)$ denotes the curve at different times t.

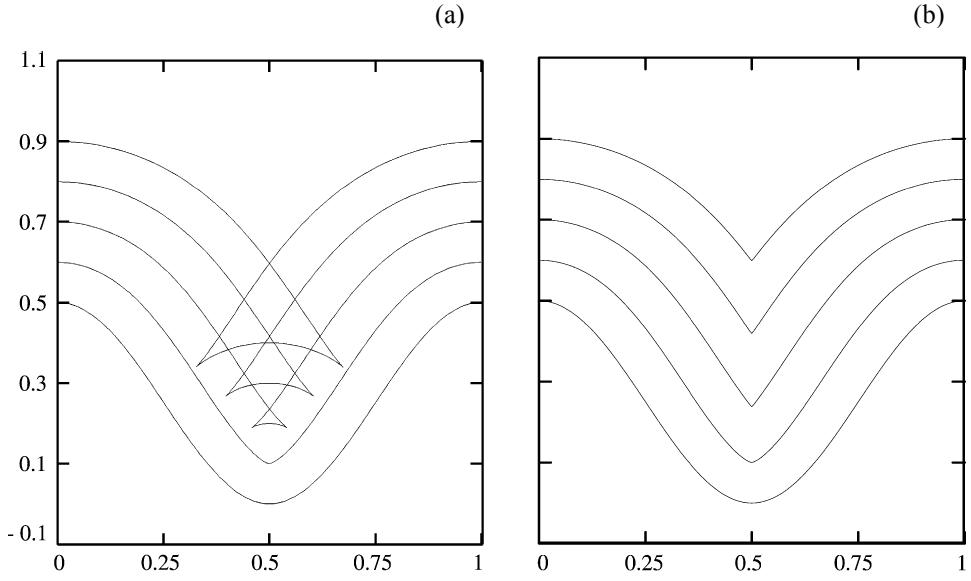

Fig. 2. Analytical solutions for the propagating cosine curve test problem (bottom curve is for $t = 0$): (a) swallowtail solution, and (b) entropy solution. (Reproduced from [1].)

As the curve propagates (see Fig. 2), it soon develops a sharp corner at the center, and thus the derivative is not defined there. This problem has two possible analytical solutions. One possibility is the "swallowtail" solution, which is formed by letting the front pass through itself. However, if the moving curve is regarded as a physical interface separating two regions, this solution wouldn't be meaningful. A more physically reasonable weak solution can be obtained through the following "entropy condition": If the front is viewed as a burning flame, then once a particle is burnt it stays burnt. This alternate weak solution is shown in Fig. 2 (b).

Eq. (2.7) is solved numerically on a rectangular domain of dimensions [0, 1] in x and [-0.1, 1.1] in y using a uniform mesh of 100x120 grid points. The spatial grid size is taken fine enough so that a smooth interface can be obtained ($\Delta x = \Delta y = 0.01$). The time step of the simple forward Euler method is set so that the Courant–Friedrichs–Levy (CFL) stability condition, Eq. (A.4), is satisfied. Periodic boundary conditions are enforced on the x-axis (i.e. at $x = 0$ and $x = 1$), while Dirichlet boundary conditions are enforced on the y-axis (i.e. at $y = -0.1$ and $y = 1.1$). Computed $\phi = 0$ contours are shown in Fig. 3 below, for two different values of the parameter b. The other numerical parameter of Eq. (2.7), W, which is a measure of the width of the hyperbolic tangent profile, is taken here to be constant. It is apparent from Fig. 3 that the solution for the case where $b = 0.01$ is in much better agreement with analytical weak solution. Generally, a careful choice of b has to be made. Increasing b would result in better enforcement of the hyperbolic tangent phase-field across the interface. On the other hand, larger values of b would blow up the discretization errors and numerical dissipation from the right-hand side of Eq. (2.7).

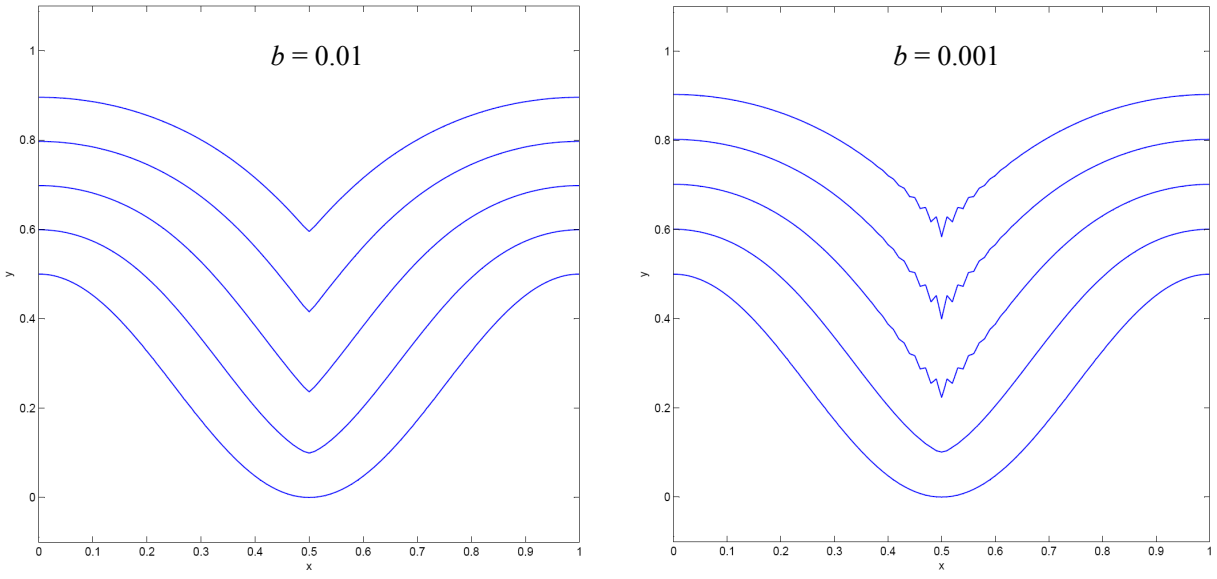

Fig. 3. Calculated $\phi = 0$ contours for the propagating cosine curve test problem using a 100 x 120 mesh; left panel: $b = 0.01$; right panel: $b = 0.001$.

2.3.2 Interface motion due to an external flow field

In this test case the diagonal translation of a circle across a unit square domain is considered. The circle, with a radius $R = 0.15$, is initially centered at (0.25, 0.25), and the phase-field is initialized so that $\phi_0 = 1$ inside the circle and $\phi_0 = -1$ outside the circle. A uniform and constant external velocity field is imposed over the whole computational domain, such that the x and y components of the external velocity are $u_e = 1$ and $v_e = 1$, respectively. 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.

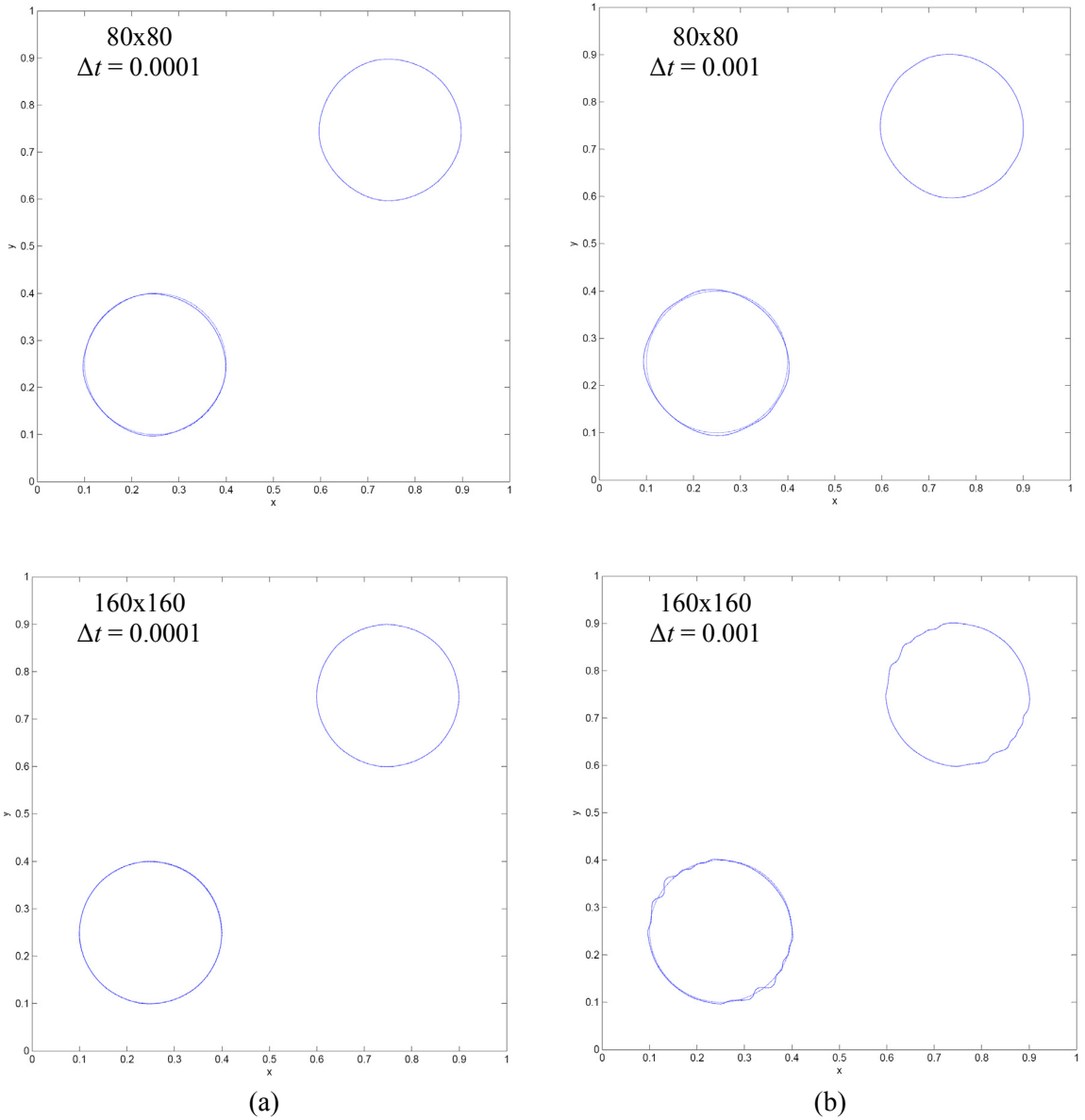

Fig. 4. Calculated $\phi = 0$ contours for diagonal translation of a circle at half domain translation and after return to the initial position using $b = 0.01$(dashed line: initial interface contour; upper panels: 80x80 mesh; lower panels: 160x 160 mesh): (a) $\Delta t = 0.0001$, and (b) $\Delta t = 0.001$.

The optimum value for the numerical parameter b recommended by Sun and Beckermann [1] is used $(b \approx \Delta x)$. Computed $\phi = 0$ contours at half domain translation and after return to the initial position are plotted in Fig. 4 for two meshes and two different time step values. Fig. 4 (a) shows a slight improvement in the accuracy by refining the spatial grid from 80x80 to 160x160. However, if the time step is not small enough, numerical instabilities could arise if the mesh is refined, which can be observed in Fig. 4 (b).

3. Part II: Conservative Level Set Method for Two-Phase Flows

One of the major drawbacks of the level set method is that it is not conservative; incorrect loss or gain of area (volume in three dimensions) might occur especially in regions where the evolving surface has high curvature. Recently a conservative level set method has been developed by Olsson and Kreiss [2] to overcome this deficiency. The conservative level set method, which is implemented in the finite-element based solver COMSOL Multiphysics, is used here to simulate incompressible two-phase flows, where gravity and surface tension effects are included.

3.1 Mathematical Formulation

In the conservative level set formulation [2], the level set function ϕ is represented by a smeared Heaviside function (Eq. 3.1), instead of the signed distance function that is used in the original level set method. So, the level set function changes smoothly across the interface from 0 to 1, and the interface is defined by the 0.5 isocontour of ϕ.

$$\phi = H_{sm}\left(\phi_{sd}\right) = \begin{cases} 0, & \phi_{sd} < -\varepsilon \\ \dfrac{1}{2} + \dfrac{\phi_{sd}}{2\varepsilon} + \dfrac{1}{2\pi}\sin\left(\dfrac{\pi\phi_{sd}}{\varepsilon}\right), & -\varepsilon \le \phi_{sd} \le \varepsilon \\ 1, & \phi_{sd} > \varepsilon \end{cases} \tag{3.1}$$

where

$$\left|\phi_{sd}\left(\boldsymbol{x}\right)\right| = d\left(\boldsymbol{x}\right) = \min_{\boldsymbol{x}_\Gamma \in \Gamma}\left(\left|\boldsymbol{x} - \boldsymbol{x}_\Gamma\right|\right) \tag{3.2}$$

The parameter ε determines the thickness of the region where ϕ goes smoothly from 0 to 1 and is taken to be of the same order as the size of the mesh elements.

The advection equation of the standard level set method, Eq. (1.4), does not contain any diffusive term, and it is therefore unstable to solve numerically. Moreover, the initially smooth transition across the interface might either smear out or sharpen. To stabilize the equation and to maintain the same smooth transition between 0 and 1, Eq. (1.4) is replaced by

$$\frac{\partial\phi}{\partial t} + \boldsymbol{u}\cdot\nabla\phi = \gamma\left[\nabla\cdot\left(\varepsilon\nabla\phi - \phi\left(1-\phi\right)\frac{\nabla\phi}{\left|\nabla\phi\right|}\right)\right] \tag{3.3}$$

where γ is a parameter that determines the amount of reinitialization or stabilization and must be carefully tuned for each specific problem. If γ is too small, the thickness of the interface might not remain constant, and oscillations in ϕ could appear because of numerical instabilities. On the other hand, if γ is too large, the interface moves incorrectly.

However, for incompressible flows the velocity field is divergence free, that is

$$\nabla \cdot \boldsymbol{u} = 0 \tag{3.4}$$

and thus the volume (surface for 2D problems) bounded by the interface should be conserved if there is no inflow or outflow through the boundaries. To obtain exact numerical conservation, it is important to note that it is possible to rewrite Eq. (3.3) in the conservative form

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \left(\boldsymbol{u} \phi \right) = \gamma \left[\nabla \cdot \left(\varepsilon \nabla \phi - \phi \left(1 - \phi \right) \frac{\nabla \phi}{\left| \nabla \phi \right|} \right) \right] \tag{3.5}$$

Using the above conservative form exact numerical conservation of the integral of ϕ can be obtained.

For incompressible two-phase flows, the level-set equation Eq. (3.5), is coupled with the Navier-Stokes equations:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{3.6-a}$$

$$\rho \left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{t}} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\nabla p + \nabla \cdot \left(\mu \left(\nabla \boldsymbol{u} + \left(\nabla \boldsymbol{u} \right)^T \right) \right) + \rho \boldsymbol{g} + \sigma \kappa \delta \boldsymbol{n} + \boldsymbol{F} \tag{3.6-b}$$

where $\boldsymbol{g}$ is the gravitational acceleration, σ is the surface tension, and ρ and μ are the density and viscosity, respectively. The density and viscosity varies smoothly over the interface according to

$$\rho = \rho_1 + \left(\rho_2 - \rho_1 \right) \phi \tag{3.7}$$

$$\mu = \mu_1 + \left(\mu_2 - \mu_1 \right) \phi \tag{3.8}$$

where ρ_1 , ρ_2 and μ_1 , μ_2 are the densities and viscosities of the two fluids, respectively.

Before solving the governing equations, Eq. (3.5) and (3.6), the level set function has to be initialized such that it varies smoothly from 0 to 1 across the interface. This is done by letting ϕ_0 be 0 on the interior of the interface and 1 on its exterior, and then the following initialization equation is solved to steady state

$$\frac{\partial \phi}{\partial \tau} = \gamma \left[\nabla \cdot \left(\varepsilon \nabla \phi - \phi \left(1 - \phi \right) \frac{\nabla \phi}{\left| \nabla \phi \right|} \right) \right] \tag{3.9}$$

where τ is an artificial time.

3.2 Numerical Implementation

CMOSOL Multiphysics, a finite element based solver, is used to solve the governing equations. The conservative level set method of Olsson and Kreiss [2] has recently been implemented within the Level Set Two Phase Flow application mode. The application mode models the momentum and mass balances by the Navier-Stokes equations, Eq. (3.6), with volume forces for surface tension and gravity. The conservative level set method [2] is adopted to implicitly capture the position of the fluid-fluid interface.

3.3 Numerical Experiments and Results

3.3.1 Rising bubble

As a first test case, we consider the problem of a submerged oil bubble that travels up through a heavier fluid (water) and finally merges with the oil already residing at the top of the container. Three different regions exist initially: the oil bubble, the oil at the top of the container, and the water surrounding the bubble. The level set function is initialized by letting ϕ_0 equals 1 in the water and 0 in the oil, and then solving Eq. (3.9) to steady state, so that ϕ varies smoothly from 0 to 1 across the interface. Fig. 5 depicts the resulting initialized level set function. The container is cylindrical with a diameter of 0.01 m and a height of 0.015 m. Both geometry and initial data are axisymmetric.

The oil studied in this example has viscosity 0.0168 Pa·s and density 880 kg/m^3. For water viscosity is $1.04 \cdot 10^{-3}$ Pa·s and density 997 kg/m^3. The driving force for the bubble's movement is the difference in density between oil and water. The oil is lighter than water and the bubble therefore starts to rise due to gravity.

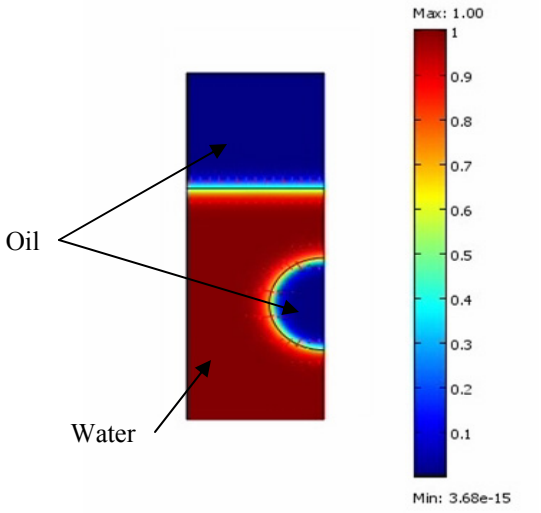

Fig. 5. Rising bubble: surface plot of initial level set function.

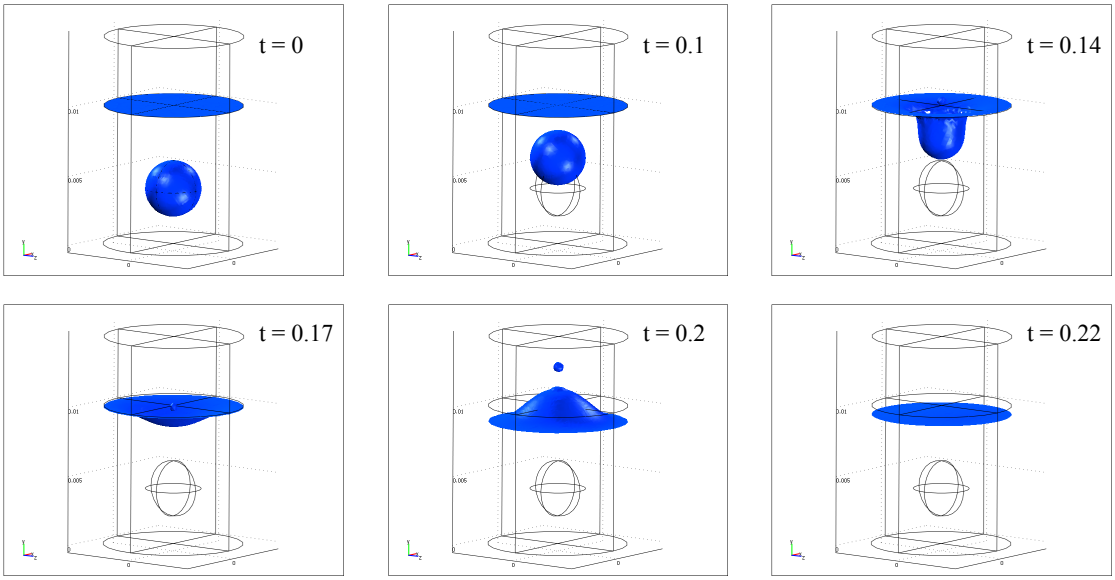

Fig. 6. Rising bubble: snapshots of the oil-water interface at different times.

No-slip conditions, $\boldsymbol{u} = 0$, are used at the top and bottom of the cylinder. Slip conditions are used at the vertical walls in order to allow the fluid interface to move along these walls.

To create 3D images of the fluid interface, the 2D axisymmetric results are revolved around the vertical axis. The snapshots of Fig. 6 show how the bubble travels up through the water and merges with the oil above. Upon coalescence of the bubble with the fluid interface a small water droplet is formed, which then falls back due to gravity. As evident in Fig. 7, exact numerical mass conservation is achievable using the current method.

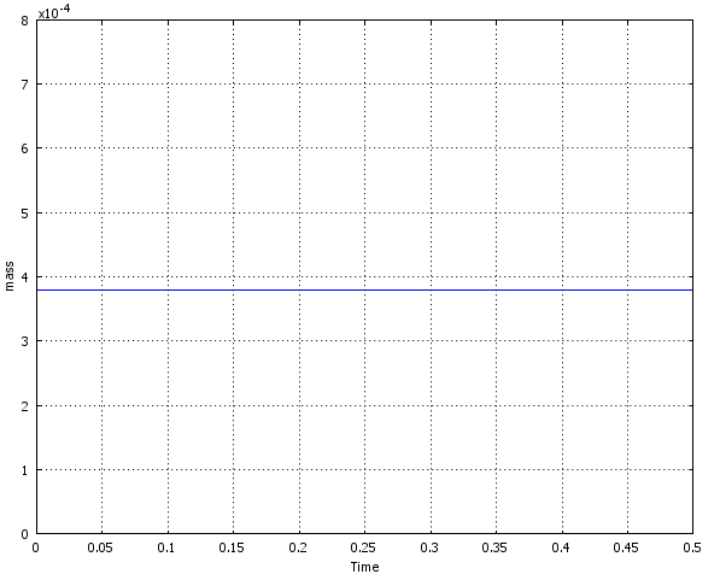

Fig. 7. Mass conservation for the rising bubble case.

3.3.2 Falling droplet

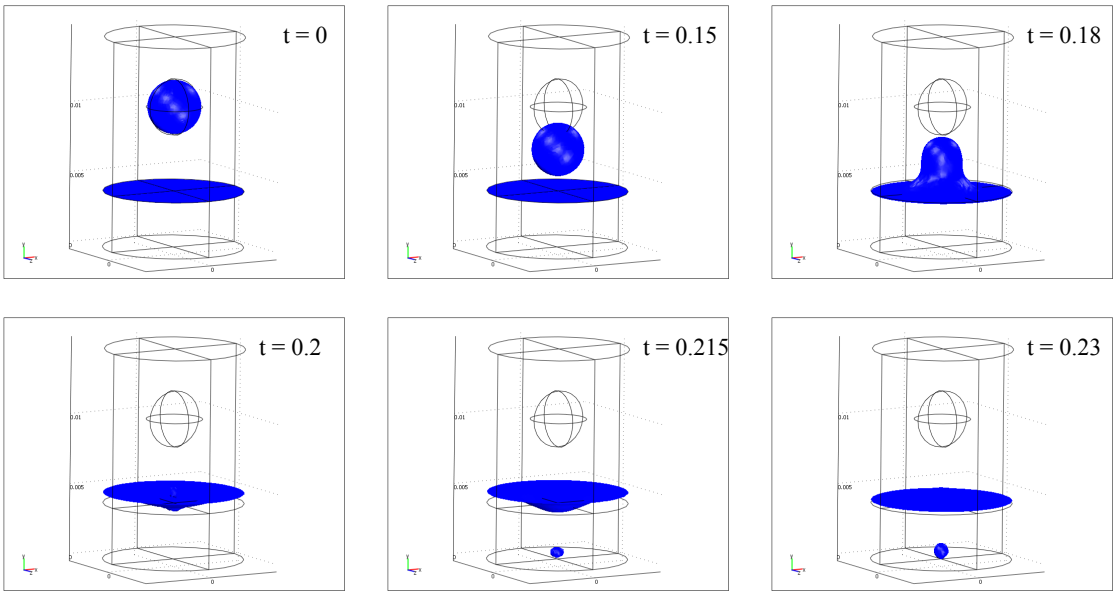

Fig. 8. Falling droplet: snapshots of the oil-water interface at different times.

Finally, we consider the case of a water droplet falling upon an oil-water interface. The properties of the fluids and the boundary conditions are the same as for the rising bubble case. The location of the fluid interface at different times is shown in Fig. 8. As the droplet falls through the oil, its shape remains spherical because of the surface tension between the water and oil. As the droplet hits the oil-water interface, it splashes and creates waves on the surface, which are finally damped.

4. Conclusions

Two recently developed variant techniques of the standard level set method were investigated. The sharp-interface phase-field method [1] was implemented in Matlab using finite difference discretization, and was used for capturing propagating interfaces due to normal interface motion and external velocity fields. This approach avoids the need for separate reinitialization step, which is normally needed in standard level set method, but the implementation of the numerical scheme is considerably more complex and the computational cost is relatively high. In the second part of the project, the conservative level set method [2] was adopted for simulation of incompressible two-phase flows. Good mass-conservation was achieved without loosing the simplicity of the original level-set method. However both methods include non-physical numerical parameters that need to be carefully tuned for each specific problem.

Acknowledgements

The author would like to thank Prof. Ali Nadim and Prof. Darryl Yong for helpful discussions and comments. The use of some of the Matlab m-files from [6] after proper modification is acknowledged. Support from Claremont Graduate University and Keck Graduate Institute is highly appreciated.

Appendix A

The numerical implementation of the finite-difference method for the phase-field equation is presented here for the two-dimensional case. The discretization formulas, recommended by Sun and Beckermann [1], has been followed in most cases, in an attempt to reproduce some of their results.

The phase-field equation is descritized on a square grid using a spatial step size of Δx. For the Laplacian of the phase-field, a 9-point finite-difference stencil is adopted, i.e.,

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

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

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

For the mean curvature, $\kappa = \nabla\cdot\left(\nabla\phi/|\nabla\phi|\right)$, the following formula is adopted

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

The third-order Hamilton-Jacobi essentially non-oscillatory (HJ ENO) scheme is used to calculate the numerical fluxes, $\phi_x^-, \phi_x^+, \phi_y^-$, and ϕ_y^+, for the hyperbolic term $\boldsymbol{u}_e \cdot \nabla\phi$ in the phase-field equation. Here, the superscripts – and + denote backward and forward difference, respectively, and the subscripts *x* and *y* denote partial derivatives with respect to the spatial coordinates. The basic idea behind the HJ ENO scheme is to use a high-order accurate polynomial to reconstruct ϕ and then differentiate it to get an approximation to the partial derivatives of ϕ. A different such polynomial is constructed at each grid point. The key to the algorithm is to choose the neighboring points for the interpolation so that we are not interpolating across steep gradients. For instance, when calculating (ϕ_x^-) at node (*i,j*), the third-order accurate HJ ENO scheme uses a subset of $\left\{\phi_{i-3,j}, \phi_{i-2,j}, \phi_{i-1,j}, \phi_{i,j}, \phi_{i+1,j}, \phi_{i+2,j}\right\}$ to get the smoothest possible polynomial interpolation of ϕ. For more details about the HJ ENO, the reader is referred to [3].

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 Courant–Friedrichs–Levy (CFL) stability condition for Eq. (2.6) is given by

$$\Delta t\left[\frac{\left|a\phi_x/\left|\nabla\phi\right|+u_e\right|}{\Delta x}+\frac{\left|a\phi_y/\left|\nabla\phi\right|+v_e\right|}{\Delta x}+\frac{10b}{3\left(\Delta x\right)^2}\right]<1 \qquad \text{(A.4)}$$

References

1. Y. Sun, C. Beckermann, Sharp interface tracking using the phase-field equation, J. Comput. Phys. 210 (2007) pp. 626-653.

2. E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (2005) pp. 225-246

3. S. Osher and R. Fedkiw, Level set methods and dynamic implicit surfaces, Springer-Verlag, Berlin (2003).

4. J. Sethian, Level set methods and fast marching methods, Cambridge University Press, Cambridge (1999).

5. S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithm based on Hamilton–Jacobi formations, J. Comput. Phys. 79 (1988) pp. 12-49.

6. B. Sumengen, A Matlab toolbox implementing Level Set Methods, unpublished, (2004).